Technical Report

A Prototype Web Portal for Interactive 3D Radar Data Access

(Jianting Zhang, Last Modified 5/4/2004)

Introduction......................................................................................................................... 2
Related Work ...................................................................................................................... 5
Design and Implementation ................................................................................................. 7
Conclusion and Future Work............................................................................................... 10
References.......................................................................................................................... 11
Introduction

Web access to meteorological data has gained increasing interests during the past years. While many web sites, commercial or academic, provide near real time radar images, most of them are pre-generated in an ad-hoc manner. They do not allow advanced 2D-based interactions, such as zoom/pan/query; and do not support 3D visualization and analysis, such as iso-surface based 3D radar products. In this study, we focus on interactive 3D radar data access over the Web.

Existing work on web-access of meteorological data can be roughly classified into three categories. The first category explores Java Applet technique. Java Applets are developed and then embedded into HTML pages. When these Applets are executed at the client side, they communicate with the servers that host them. Data at the server sides are retrieved on demands and transferred to client sides for further processes, such as visualization and analysis. The advantage is that once the Applet and the data are transferred to client sides, they become part of the client side computing environment and users’ requests can generally be answered quicker. The GUI components can be embedded in the Applets and thus allow much flexible and efficient interactions. The disadvantage is that it usually takes a long time to download an Applet and the data when they are big over a narrow bandwidth connection. In addition, running a complex Applet consumes considerable resources at the client machine and there could be security problems. By putting some commonly used functions into plug-ins, it is possible that these plug-ins need only to be installed once at the client sides and sever for multiple Applets. This makes it similar to the techniques in the third category as discussed shortly.
However, the application specific Applets still need to be downloaded on the fly which make it different from the techniques in the third category.

The second category techniques use standard HTML components. They generate images based on the requests specified in the URLs and/or hidden fields of forms. The advantage is that, since they are based on standard HTML, they can support a broader range of web browser and does not require any plug-ins. Apart from the display, they do not consume any client resources and do not impose security problems. The disadvantage is that images need to be generated multiple times which means significant communication overheads between clients and servers when there are user interactions. In addition, the GUI components need client side scripts for communication which is not as flexible as that in the Applet based techniques.

The techniques in the third category can be treated as the combination of that of the above two categories. The servers generate files (other than the images defined in HTML) on demands and send them back to the client side. If the file types are known to the clients, stand alone programs or plug-ins can be invoked to handle the files. The programs and the plug-ins are only needed to install once for all the files of the same types which is clearly more beneficial than the Applet based techniques. However, the cost of managing the programs and the plug-ins at the client side can be expensive when they are extensively used.

Many projects providing web-access to meteorological data use ESRI ArcIMS ([HREF 1]), whose technique falls into the second category. However, it only work for 2D data that are similar to geographical data in nature. For web-access to 3D meteorological data, to the best of our knowledge, most existing work propose to use
VRML which falls into the third category. Although it is possible to use Applet based technique for 3D data access over the web, such as using 3D Java APIs (not necessarily Java3D from SUN), with the exception of IDV which we will discuss shortly, we are not aware of such applications in meteorology. The possible reason maybe due to its technical complexities. In fact, even for 2D data Web-access, techniques from the second category seem dominate over that of the first one.

We are particular interested in providing 3D weather radar access using the techniques in the second category, i.e., dynamically generating images at the server sides and rendering them to client sides. In addition to the advantages discussed above, we observe that many 3D weather radar products, such as iso-surfaces, requires hardware supports which may not be available at the client sides. Indeed, computational power of graphics hardware has been improved dramatically in recent years. Currently the top-of-the-line consumer graphics processors has 63 million transistors, while modern CPUs such as AMD AthlonXP and Intel Pentium 4 have only 37.5 and 55 million transistors, respectively (Sun, 2003). The sever side high-performance graphic hardware has the capability of fast generating images and the shift of computation resource requirements from client side to server side can be well justified.

For the rest of this report, we first overview the existing applications of web-access to meteorological data and analysis the techniques explored according to our classification. We then describe the design and implementation of the prototype web portal for interactive 3D radar data access for iso-surface based visualization. Finally it is the conclusion remarks and future work directions.
Related Work

A Web-based mapping tool being developed at the Goddard Space Flight Center Earth Science Distributed Active Archive Center (GES DAAC) explores Java applet technique (Pollack, 2003) The tool allow users to view and select data sets of Region of Interest (ROI). It can also generate data sets of ROI on-the-fly for downloading. The QPE package (used internally) developed at the NSSL provides web access to NEXRAD data, both single radar and radar mosaic and related GIS data sets as background. It uses a Java plug-in from WDT for 2D display which also provide client side scripts for communications between GUI components. The elevation scans are displayed individually.

Several projects uses the Commercial-Off-The-Shelf (COTS) ArcIMS package from ESRI. A Web-portal to real-time observations and forecasts for estuaries and the coastal ocean developed at the National Ocean Service’s Cost Survey Development Laboratory is reported in (Kelley, 2003). The Motion Tracker developed by NOAA’s Office of Marine and Aviation Operations (OMAO) is reported in (Merati, 2003). It provides near-real time mapping of moving objects, both research vessels and marine mammals. The WEBMap Calculator project at the Pacific Marine Environmental Laboratory (PMEL) extends the simple display/query features of ArcIMS to inter and intra layer calculations with layers of the data served in the map server (Merati, 2004). The National Weather Service (NWS) has begun exploring the use of Internet Mapping Service technology to disseminate hydrometerological weather data as a way to improve their services (Settelmaier, 2004). Built on top of ArcIMS, the data layers include in the system are Hurricane information, Warnings and Watches, Satellite Imagery,
Precipitation, Flooding, Wind and background information. Including Radar data is among their future work.

A project on VRML-based visualization of GIS data for a marine sanctuary is reported in (Vance, 2003). VRML files are created using varies software modules (we are not sure whether it is online or offline.) A web page using a Java applet and the VRML External Authoring Interface (EAI) allows the user to change the VRML scene graph depending on the choice of the data. The technique explored is an extension from the techniques in the third category. The project reported in (Hermann) uses the Live Access Server (LAS), a web server for earth science data sets, to call a back-end routine to create VRML for iso-surface based visualization. This routine is written using the VTK API and takes as input a 3D netCDF data file, the variable name, the region and a file size compression target for iso-surface decimation.

The VisAD package has gained increasing popularities in meteorological community. Designed to exploit distributed object technology to provide improved data sharing and collaborative applications to scientists and others with numerical data, VisAD provides much more functions than a web portal. However, the thin-client schema (as in a web portal) is preferred to the fat-client schema (as in a full functional distributed computing environment like VisAD) for scientists who are not experienced with information technology. The Unidata’s Integrated Data Viewer (IDV), built on top of VisAD (Murray, 2004), has a web-start interface which allows user to start the application using a web-browser. Although it might be easier for users to launch the program, the excessive computation resource demands required at the client side (at least
1.2GHz processor and 512M RAM) makes it a less idea candidate for easy web-access to radar data ([HREF 3]).

**Design and Implementation**

Our design philosophy follows that of ESRI ArcIMS ([HREF 1]) and Mapinfo MapExtreme ([HREF 2]), i.e., generating images at the server side and rendering image to the web browsers at the client side. In addition to the discussions in the introduction section, the following observations further justify our preference:

1. We aim at providing iso-surface based 3D radar data visualization. The number of data points in a complete volume (9 elevation scans in VCP 21 mode) is about \( \frac{3}{4} \) million. Although there is only about 10% of the points has meteorologically meaningful values, converting them to VRML files for visualization alone might be too much. Even worse, using the marching cube algorithm, a triangle in the generated iso-surface requires not only vertex coordinates, but also additional data such as normal vector, color and texture. The huge amount of data might be too overwhelming for VRML-based radar data iso-surface visualization over the Web. On the contrary, no matter how large the data might be at the client side, once converted to a reasonable sized image, the data volume can be quite small and suitable for transport over the Internet.

2. When users change their view parameters, such as time period, ROI, view angles and iso-values, it is true that the images have to be regenerated and retransmitted. However, we believe that only a small portion of the parameter space are interested to users and the parameter space they actually want to
explore are even less. Thus it is possible that the total overheads of regeneration and retransmission is less than that of VRML-based technique.

3. Hardware acceleration can be used in image generation at the server side and does not client side to have graphic hardware. This is desirable for users with low-ends equipment configuration and even more desirable for hand held devices in mobile computing.

4. For our solution, users deals with standard HTML GUI, rather than ad-hoc interfaces as in Java applet based solution. We believe it is more suitable for domain experts that are less experienced in IT.

The architecture of our design is illustrated in Fig. 1.

![System Architecture Diagram]

Fig. 1 System Architecture

We implement a proof-of-concept prototype using Java, its extensions and third party packages. The Web Server for Applications is implemented as a Java Server Socket. It listens to a predefined port and accept URL requests. It parses the URL and extracts parameters necessary for 3D radar data visualization. Our current implementation only supports changing iso-values. A full functional implementation will support changing data parameters (such as data set, ID of virtual volume and iso-value) and view parameters (such as rotation/translation/scale factors). These parameters are passed to the visualization module to generate image and the web server will send the image to the client browser. An alternative to using URL to pass parameter is using
HTML form fields, such as input textbox, checkbox, spin buttons, dropdown list, etc. They can provide a better graphic user interface.

The Iso-Surface Based Visualization Module is implemented as a Java thread. A new thread will be generated for each URL request. The module is built on top of VisAD package. Given a data set name and an ID of virtual volume, the module extracts a virtual volume and interpolate the data points in the virtual volume into a VisAD Gridded3DSet. For a given iso-value, a contour map as a ScalarMap can be added to the display window, in addition to the x/y/z axis. The DisplayImplJ3D class of VisAD can extract a BufferedImage of Java using the off-screen rendering capability of Java3D. The BufferedImage then can be encoded into a JPEG image using JPEGCodec and JPEGImageEncoder from Java Advanced Imaging (JAI) and return to it to the Web Server for Applications. Note different from ESRI ArcIMS, the image is handled in server side memory without having to write out to hard disk which is more efficient. Due to the overlapping nature of virtual volumes (an elevation scan can participate up to nine virtual volumes), it is possible to implement a cache in the module.

Currently we use operating system file system as the data server, i.e., the visualization module and the data are located at the same machine. However, it is possible to put the radar data into distributed data servers. We are also planning to write back the user requests into a relational database. We can perform data mining to discover the patterns of visualization for more efficient and effective web services of 3D radar data.

A screen snapshot showing our prototype system is given in Fig. 2.
Fig. 2 A Prototype Screen Snapshot for Web Access of Iso-Surfaces

Conclusion and Future Work

Providing Web access to 3D radar data for advanced visualization is desirable. In this study, we propose to extend the image based web data access scheme for 2D geographical data to 3D radar data access. Compared to Java applet and VRML based method, it is advantageous in terms of technical complexity, efficiency, easy of use at the client side and exploring graphic hardware acceleration at the server side. We implement prototype using Java, its extensions Java3D and JAI and third party visualization package VisAD for proof of concept. More developments are underway.
References


