CPSC-533C
Information Visualization
Project Report

VisCMD:
Visualizing Cloud Modeling Data

Quanzhen Geng (#63546014) and Jing Li (#90814013)
Email: gengqz@hotmail.com
larrybcca@yahoo.ca

(Master of Software Systems Program, CICSR/UBC)

April 29, 2003

Table of Contents

Abstract

In this project, a cloud modeling dataset visualization software called VisCMD is developed. This software includes three parts: (1) data parser, (2) GUI (Graphical User Interface), and (3) Data visualization. The data visualization is the core part of VisCMD.

The data visualization part has three components that form a complete set of data visualization functions: (1) 1D/2D plot, (2) 3D visualization, and (3) parallel coordinates for multi-variable data visualization.

The software is implemented with Java and based on existing visualization libraries and tools such as VisAD, ncBrowse and parvis.

VisCMD is tested with a large cloud model output dataset provided by EOS/UBC and proved to be successful and meets all its requirements.

The 3D visualization package developed here can also be easily extended to other fields such as medical use.

1. Problem Statement

1.1 The Cloud Model and output data

The Department of Earth and Ocean Sciences of the University of British Columbia is presently using a cloud model with a high-resolution of 256x256x90 (25m) in North-South (6375m), West-East (6375m), and Altitude (2225m) directions.

The output data amount for each time-step is [256x256x90 real-numbers/variable] x 4 (bytes/real-Number) x 12 variables ~ 270MB. If we integrate the model for several days and save the model output data in a short time interval (e.g., 1 hour), the data amount will be very large. In addition, the model output data is voluminous, complex and inherently both three-dimensional and dynamic, and has multi-variables. It is nearly impossible to the atmospheric scientists to quickly analyze and interpret these model output datasets without using good data visualization tool.

1.2 Dataset

For this project, we use the three-dimensional cloud dataset from the high-resolution cloud model output provided by Atmospheric Research Group of the Department of Earth and Ocean Sciences of the University of British Columbia. The dataset is a three dimensional, multi-variable gridded data archived in the netCDF (Network Common Data Format) format.

1.3 Cloud model output data visualization requirements

According to the characteristics of the cloud model output data, the visualization tool must be able to do (1) 3D visualizations, (2) multi-variable visualizations. Also, this visualization tool should enable the user dynamically manipulate the rendering object, such as zooming, slicing, changing colors, selecting data range for visualizing, etc.

2. Related works

2.1 2D data visualization software used in atmospheric sciences

In atmospheric science, there are certainly many full-featured tools available for visualizing data in two dimensions. NCAR (National Center for Atmospheric Research) Graphics is a Fortran and C based software package for scientific visualization. It used to be very popular in the past two decades. Now, NCL (NCAR Command Language) becomes more popular. NCL is a programming language designed specifically for the analysis and visualization of data. Another very popular visualization software is called GrADS (The Grid Analysis and Display System) developed by COLA (Center for Ocean-Land-Atmosphere Studies). GrADS is an interactive desktop tool that is used for easy access, manipulation, and visualization of earth science data. However, all these data visualization software are just for 1D/2D plotting.

2.1 3D data visualization libraries used in atmospheric sciences

There are also programs for visualizing scientific data in three dimensions. Vis5D, developed at Space Science and Engineering Center at Madison of Wisconsin is a system for interactive visualization of large 5-D gridded data sets such as those produced by numerical weather models. One can make isosurfaces, contour line slices, colored slices, volume renderings, etc of data in a 3-D grid, then rotate and animate the images in real time. There's also a feature for wind trajectory tracing, a way to make text annotations for publications, support for interactive data analysis, etc. A new data visualization library, VisAD (Visualization for Algorithm Development), developed at the same place as Vis5D, is a Java component library for interactive and collaborative visualization and analysis of numerical data. The Visualization ToolKit (VTK) is another library for data visualization. It is an open source, freely available software system for 3D computer graphics, image processing, and visualization used by thousands of researchers and developers around the world. VTK consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. However, using these visualization libraries require scientists to do a lot of programming. Also these libraries tend to be complex and hard to extend to the special purposes of scientists.

2.3 What scientists want?

But scientists want visualization software that is ready to use, no need to program, and specifically designed for their purpose. Our purpose of this project is to develop software that can be used specifically for visualizing large cloud modeling dataset in three dimensions. Also, to facilitate the study of the relationships of clouds with other variables, we intend to implement a parallel coordinate visualization of cloud dataset.

2.4 Current status of developing 3D visualization software for atmospheric scientists

At present, several people and institutions are developing the 3D visualization software that does not need any programming of atmospheric scientists. For example, NOAA (National Ocean Atmosphere Administration) HPCC Program is now proposing a project to develop 3-D Visualizations with a Networking netCDF Data File Browser-ncBrowse.

3. Goals and objectives

3.1 Goals

The goal of the project is to develop the software that can be used specifically for visualizing large cloud modeling output data in three dimensions and visualizing the cloud properties (multi-variables) in parallel coordinates.

3.2 Objectives

The project tends to achieve 3 to 4 of the following objectives:

(1) General 3D display and navigation of the cloud body, which includes
  a) Display the cloud in 3D
  b) View the cloud in different directions
  c) Zoom in/zoom out the whole cloud body.

(2) Dynamic slicing of the volume rendering of cloud body and cloud properties, for example, the temperature and humidity inside the cloud

(3) Zoom a mouse-selected region

(4) Display the animation of the cloud development in a time period

(5) Parallel coordinates visualization of cloud data

4. Design

According to the goals and objectives of the software, Figure 1 shows our general design of VisCMD. VisCMD includes three parts, i.e.,

(1) Data parser, which reads the model output dataset in netCDF format, convert the data into the format needed by VisAD (flat field) and parallel coordinates plotting (STF-Simple Table Format).

(2) Graphics user interface, which enables the user input and select various parameters to control the data rendering, and also display the data information on for user to see what variables and which part of the variables are now being plotted.

(3) Data Visualization is the core part of the project. This part has three functions, i.e., (1) 1D/2D plot, (2) 3D volume rendering, and (3) parallel coordinates for multi-dimensional (multi-variable) data visualizations.

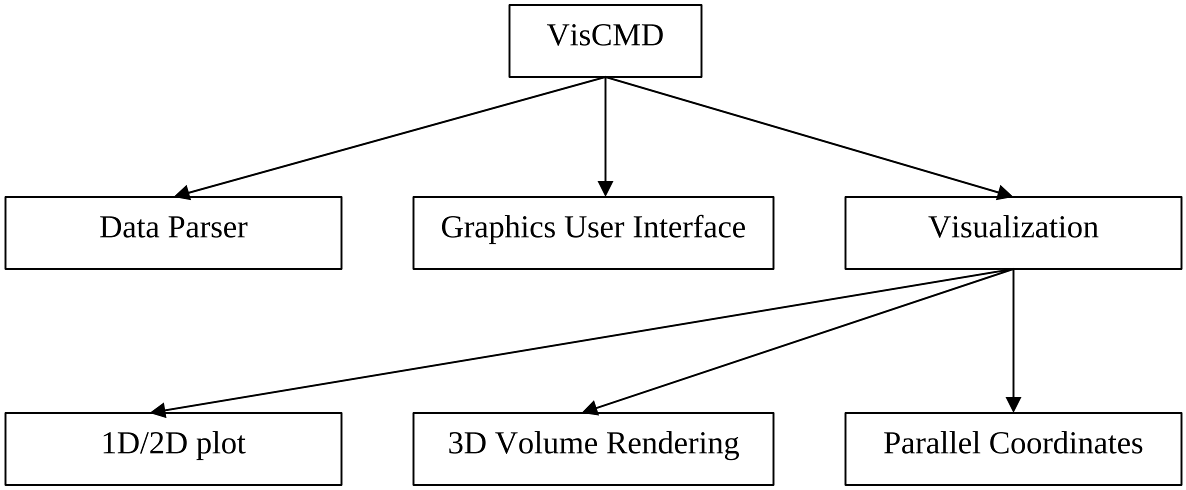


Figure 1 . The architecture of VisCMD

5. Implementation

5.1 Software Tools:

Development of the software is on based on VisAD (Visualization Algorithm Development) and existing implementations of parallel coordinates. For each part, the software libraries used can be briefly described as following:

(1) Data parser: UCAR data manipulation libraries: *Multiarray, NetCDF*

(2) Graphics User Interface: *Java Swing*

(3) Visualization:

- (a) 1D/2D plot: *ncBrowse (a netCDF file browsing tool written in pure Java)*
- (b) 3D rendering: *VisAD (a visualization library written in pure Java)*
- (c) Parallel Coordinates: *parvis (a parallel coordinates visualization tool in Java)*

The following is a brief description of the major software libraries used in the implementation:

ncBrowse is a Java tool that provides flexible, interactive graphical displays of data and attributes from a wide range of netCDF data file conventions.

VisAD (Visualization for Algorithm Development) is a Java component library for interactive and collaborative visualization and analysis of numerical data.

parvis is a tool for parallel coordinate (PC) visualisation of multidimensional data sets, as first described in [Inselberg 1981]. It is a flexible, reusable user-interface component compliant to the Java Swing and Java Beans standards, and can perform state of the art

PC visualisation and provide the user with the necessary means of visual interaction with the data set.

The Scientific Graphics Toolkit (SGT) facilitates easy development of platform independent, Java applications to produce highly interactive, flexible, publication quality, object oriented graphics of scientific data. Features include user settable or automatically scaled axes, sophisticated, automatically self-scaling time axes, labels as movable, customizable objects, automatic generation of legends to explain the data being displayed, and many more.

5.2 Implementation details:

The following describes our implementation details:

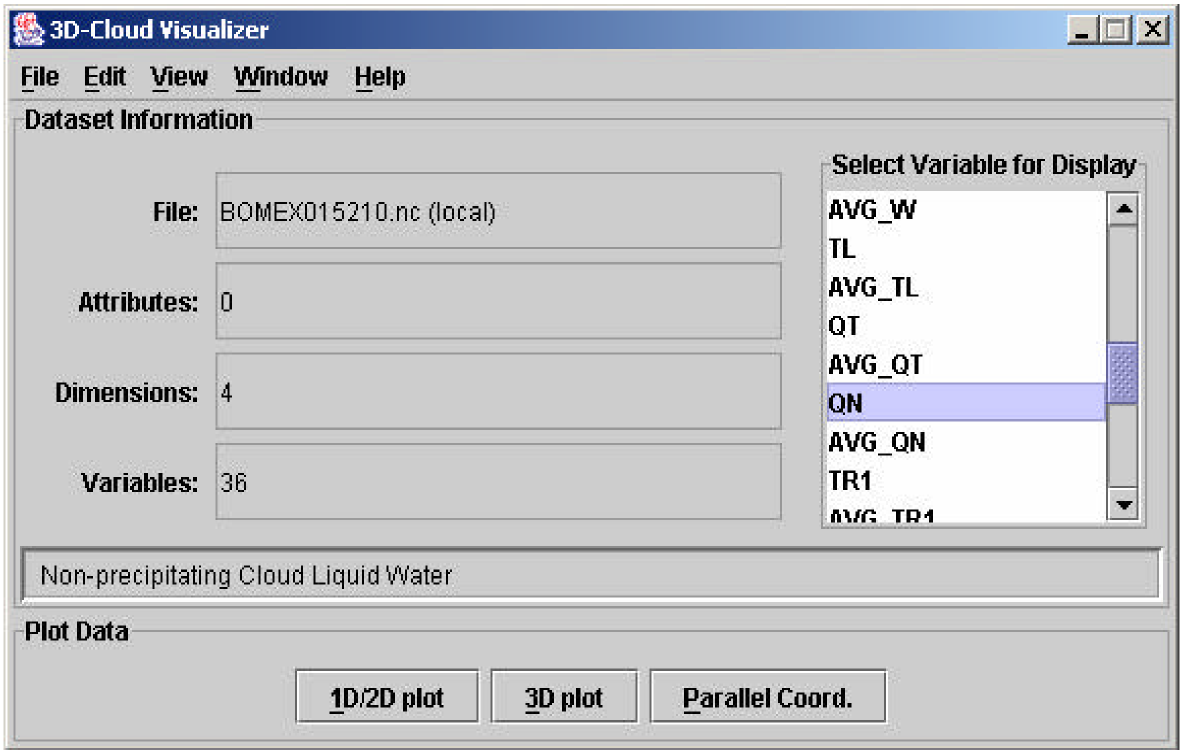
(1) We wrote a data parser with UCAR libraries Multiarray, netCDF
(2) We wrote a set of Graphical User Interface (GUI) with Java Swing
(3) We implemented the whole package of 3D visualizations with VisAD libraries.
(4) We modified the ncBrowse tool, added some classes to connect ncBrowse with our data parser and GUI, and integrate ncBrowse into our software VisCMD by creating an object of ncBrowse in VisCMD. Integration of ncBrowse into VisCMD enables the software be able to plot 1D/2D graph, and thus enhanced the completeness of the software.
(5) We modified the parvis tool, added some classes to connect parvis tool with our data parser and GUI, and integrate parvis into our software VisCMD by creating an object of parvis in VisCMD. Integration of parvis into VisCMD enables the software be able to plot multi-dimensional (or multi-variable) dataset with parallel coordinates.

5.3 Programming language and platform:

The programming language we used to implement VisCMD is Java and we developed the software on Windows 2000.

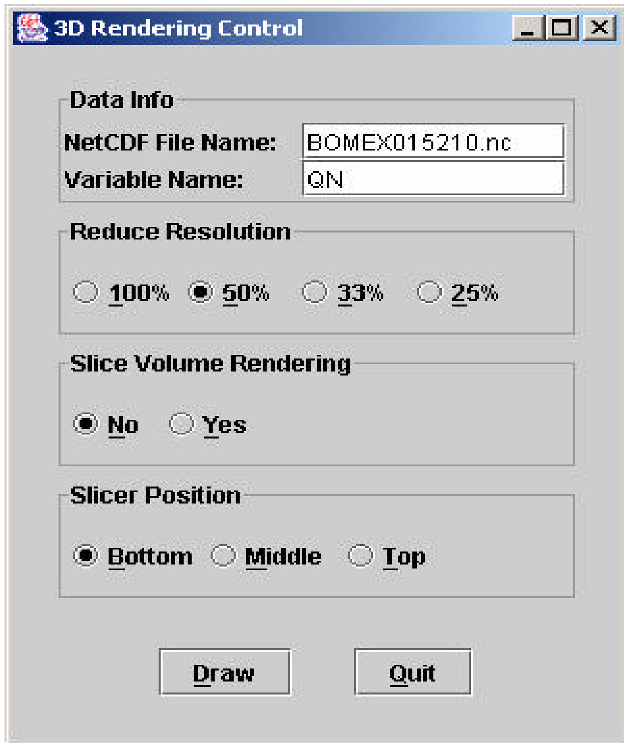
6. Screenshots

Screenshot 1: Choose and open a NetCDF file



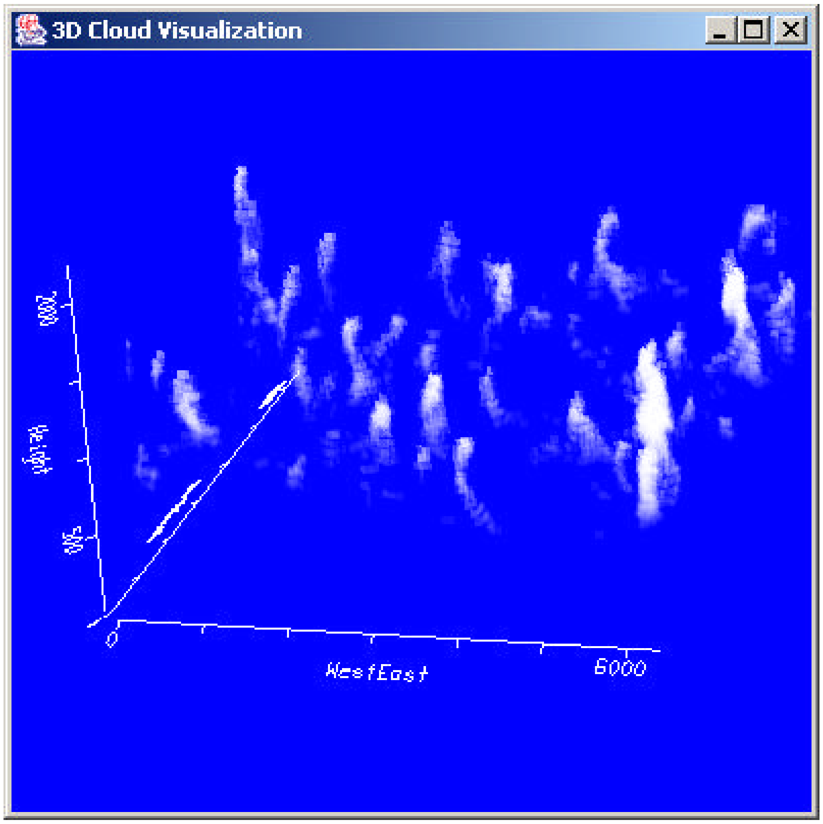
Screenshot 1 is the starting window of the VisCMD. It enables the user to choose a NetCDF file, select a variable from the variable list and choose to display with 1D/2D, or 3D rendering or parallel coordinates.

Screenshot 2: 3D rendering control



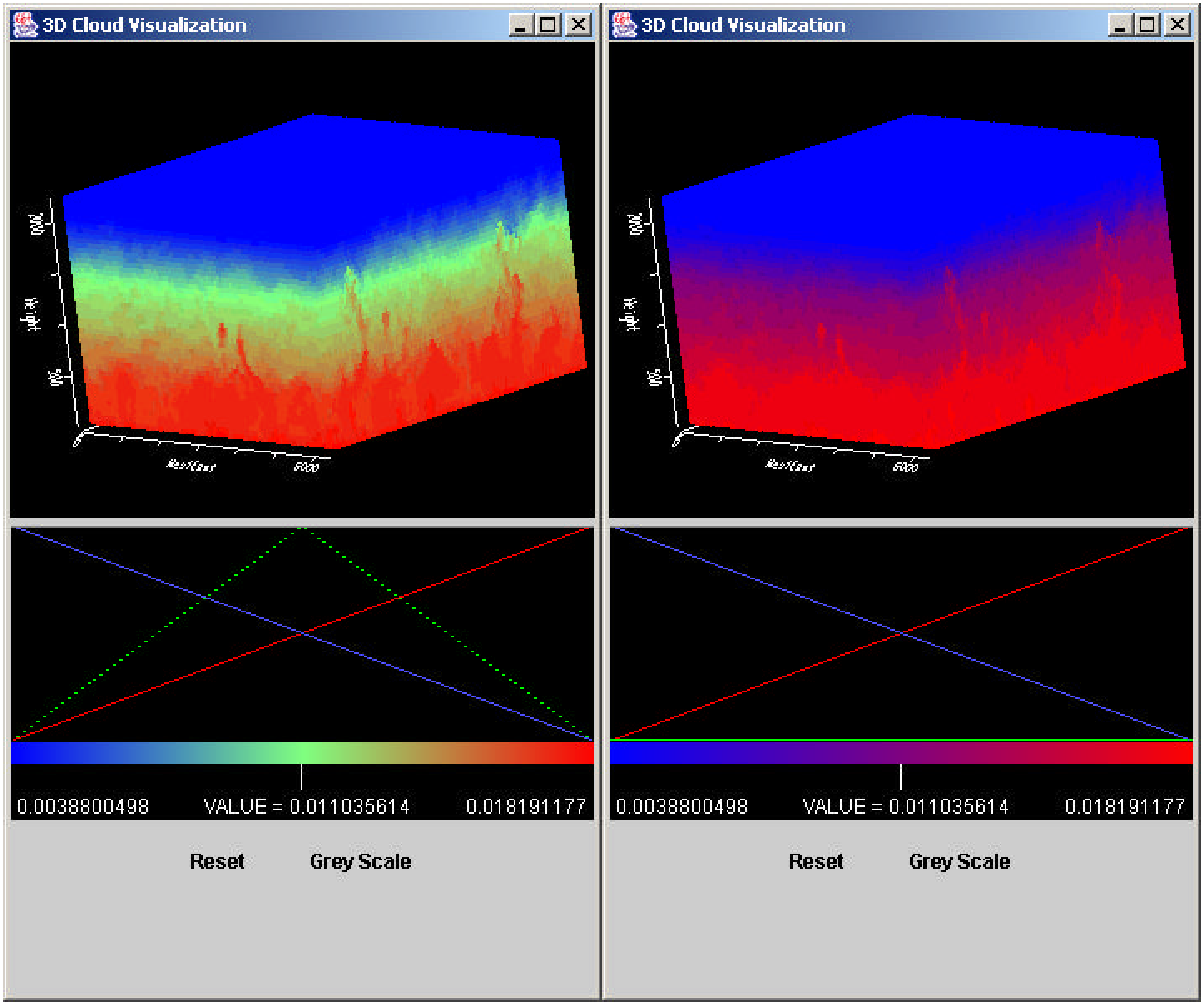
Screenshot 2 is the main control window for 3D volume rendering. It will appear when the user chooses to draw the 3D plot. Because the high resolution of the dataset, drawing the 3D plot will use a large amount of memory. So, to reduce the use of memory, the user can choose to reduce the data resolution to 50%, 33% and 25%. The volume rendering also has a function, which lets the user to make dynamical slicing of the volume rendering. The user can choose to draw the plot with or without a slicer. If a slicer is choosen, the user can also choose to put the slicer at the bottom, in the middle or on the top of the volume rendering.

Screenshot 3: 3D cloud display and navigation



Screenshot 3 show the general 3D display of the cloud body. The user can easily navigate (rotate, move, zoom in and zoom out) the 3D plot.

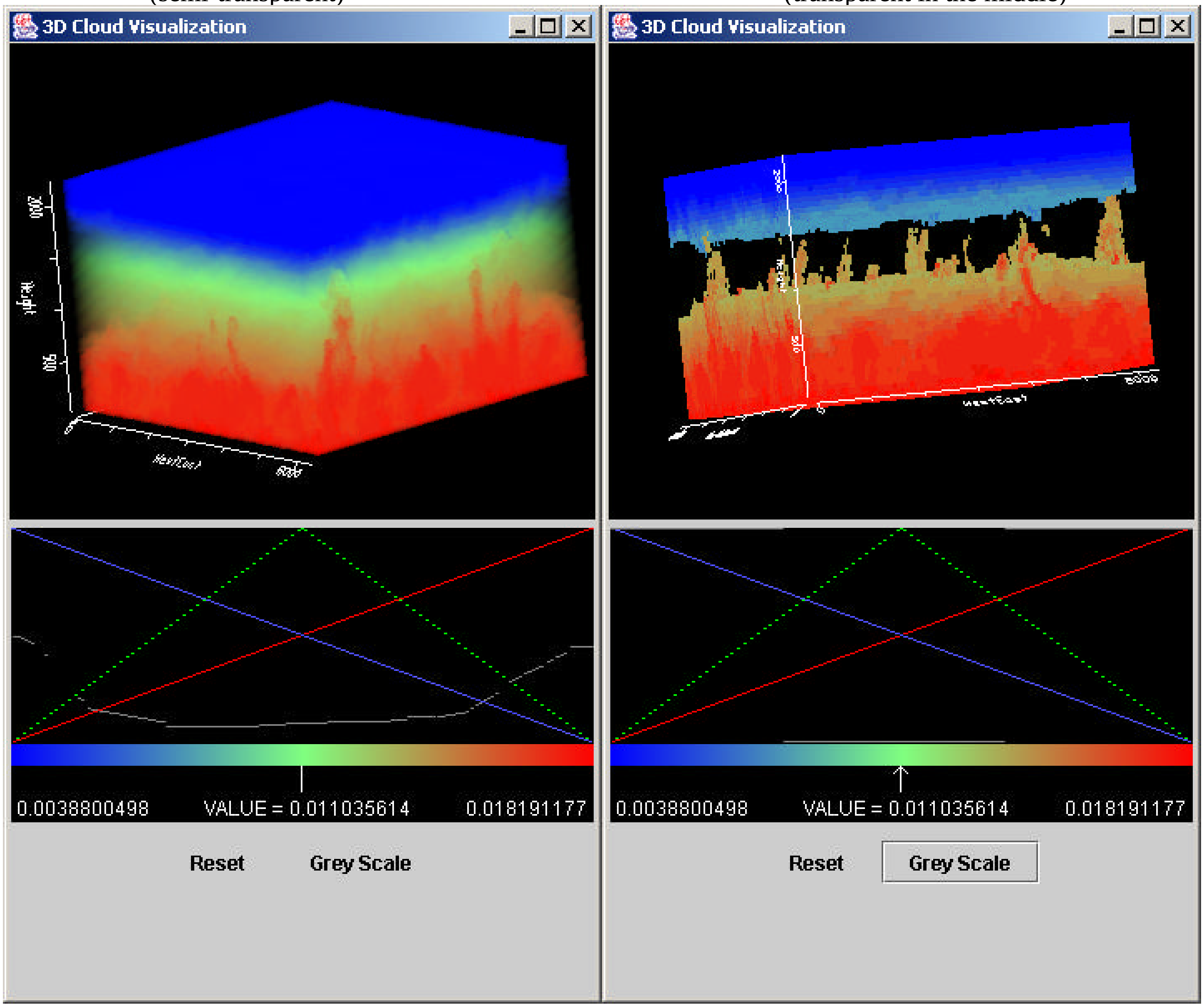
Screenshot 4: Dynamically change the colors (water vapor in the air)



Screenshot 4 shows the dynamical change of the color of the 3D volume rendering. The lower part of each figure is a color widget that can be used to change the components of RGB colors and the Alpha value. The right hand slide shows when the user set the green color component into zero, the volume rendering changed into blow to red color map.

__Screenshot 5: Dynamically change the Alpha value__ (water vapor in the air)

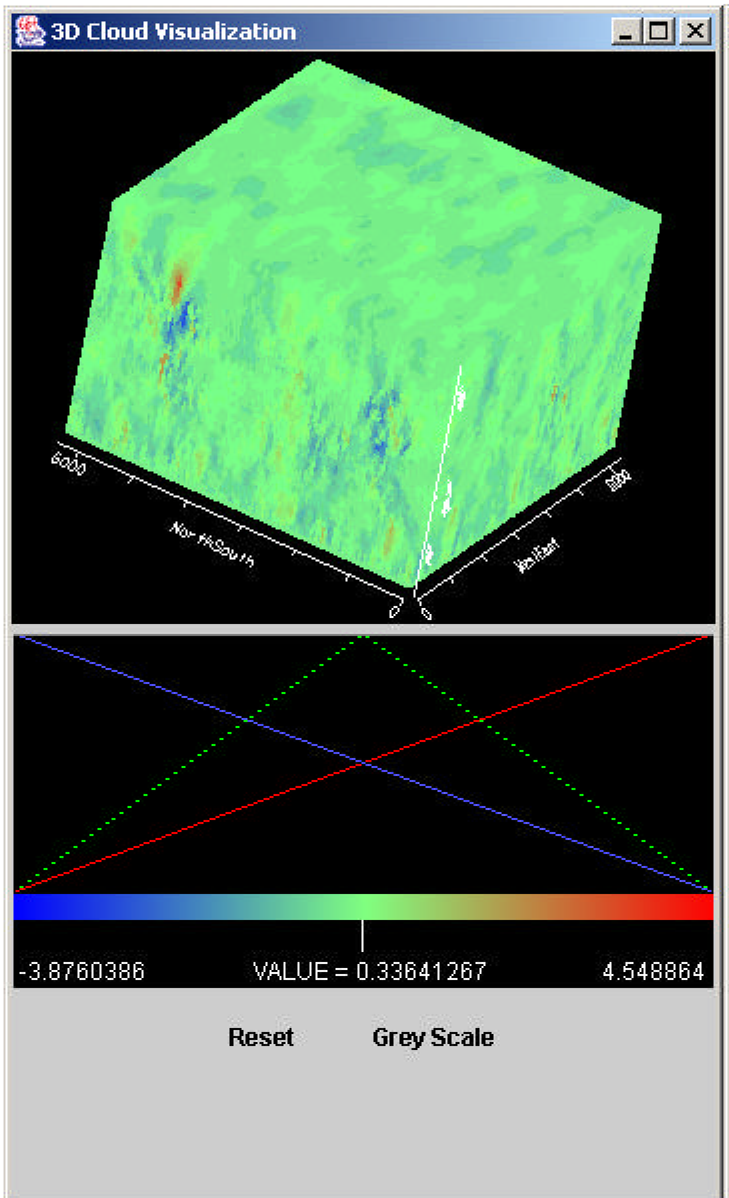
(semi-transparent) (transparent in the middle)



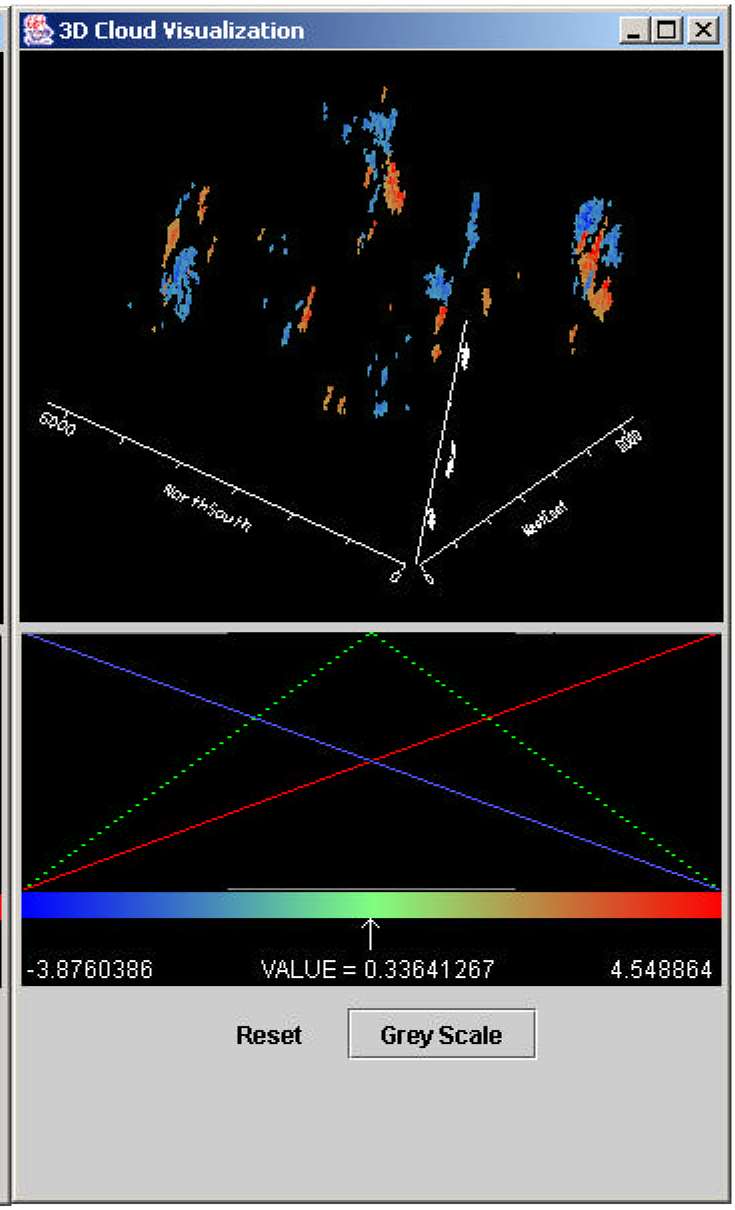
Screenshot 5 shows the effects of changing the Alpha value on the 3D plot. The left figure shows that we can make the 3D rendering semi-transparent. And the right figure shows that the user can make some part of the data range transparent by setting the Alpha value of that part of data into zero. By doing so, we can clearly see the inside of the data.

Screenshot 6: Dynamically change the Alpha value (upward flow)

(opaque)

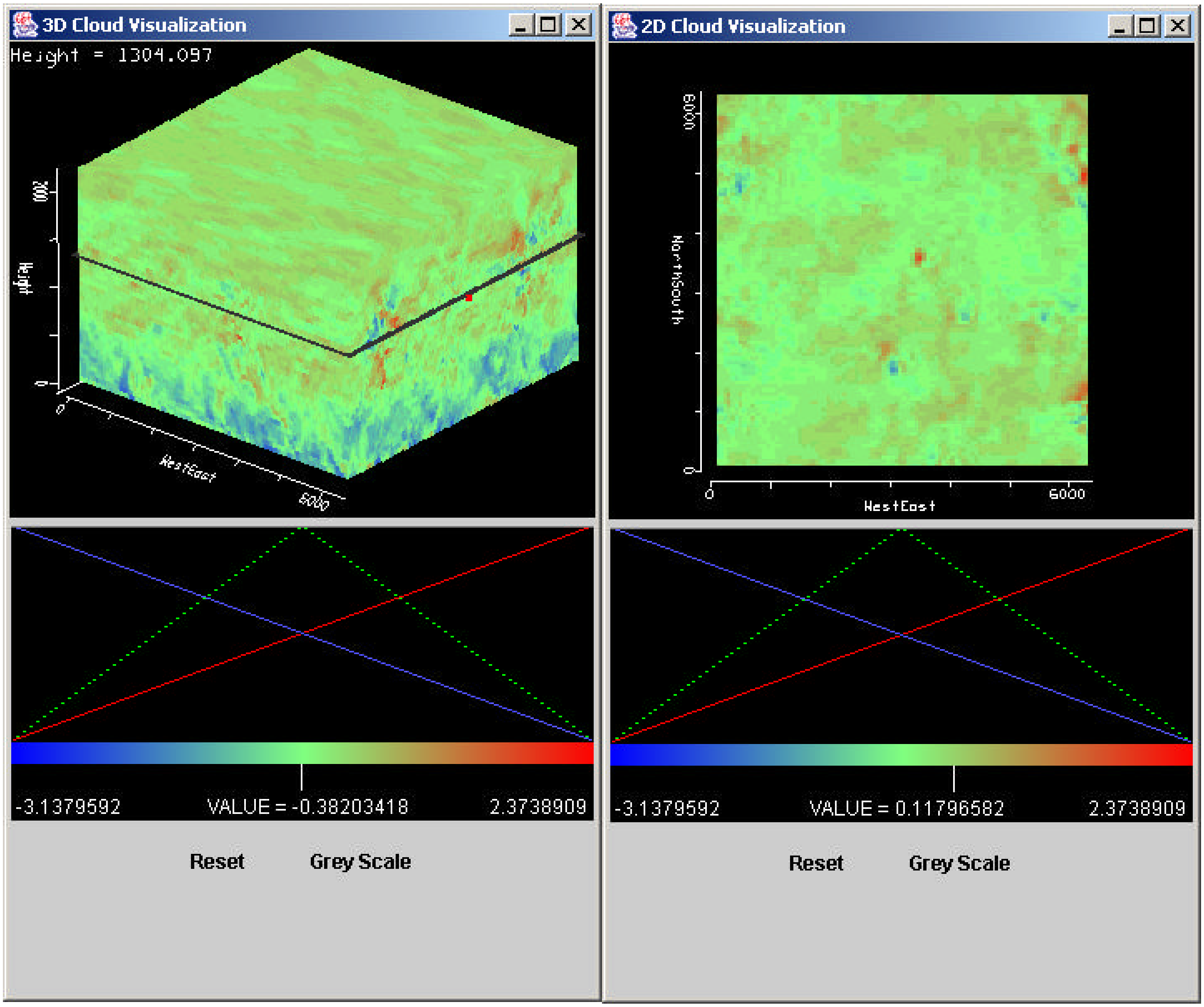


(Middle data values transparent)



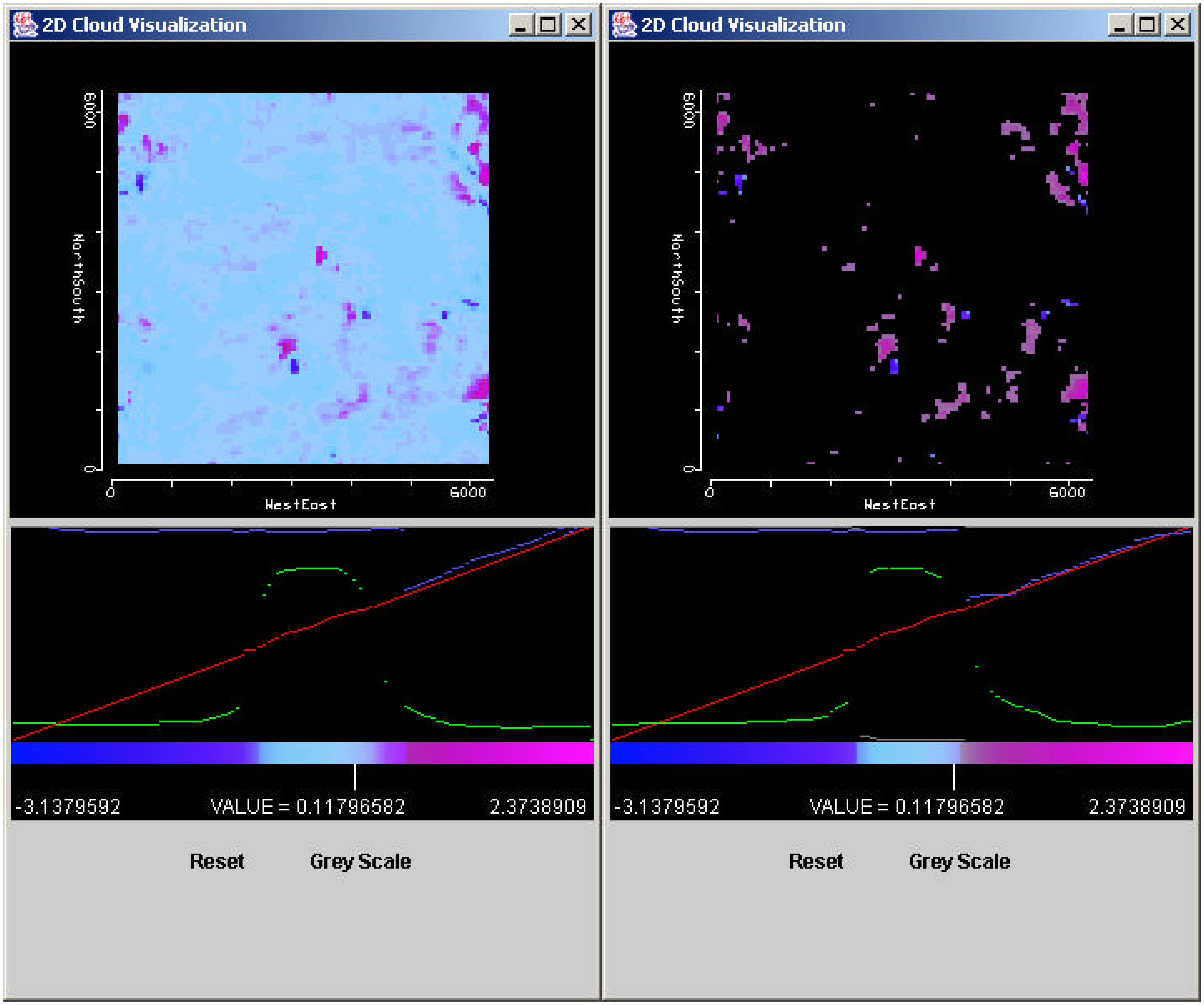
Screenshot 6 shows another example of the use of setting the Alpha value to let the plot become transparent and let the user see only the most important part of the data.

Screenshot 7: Dynamical slicing (upward flow)



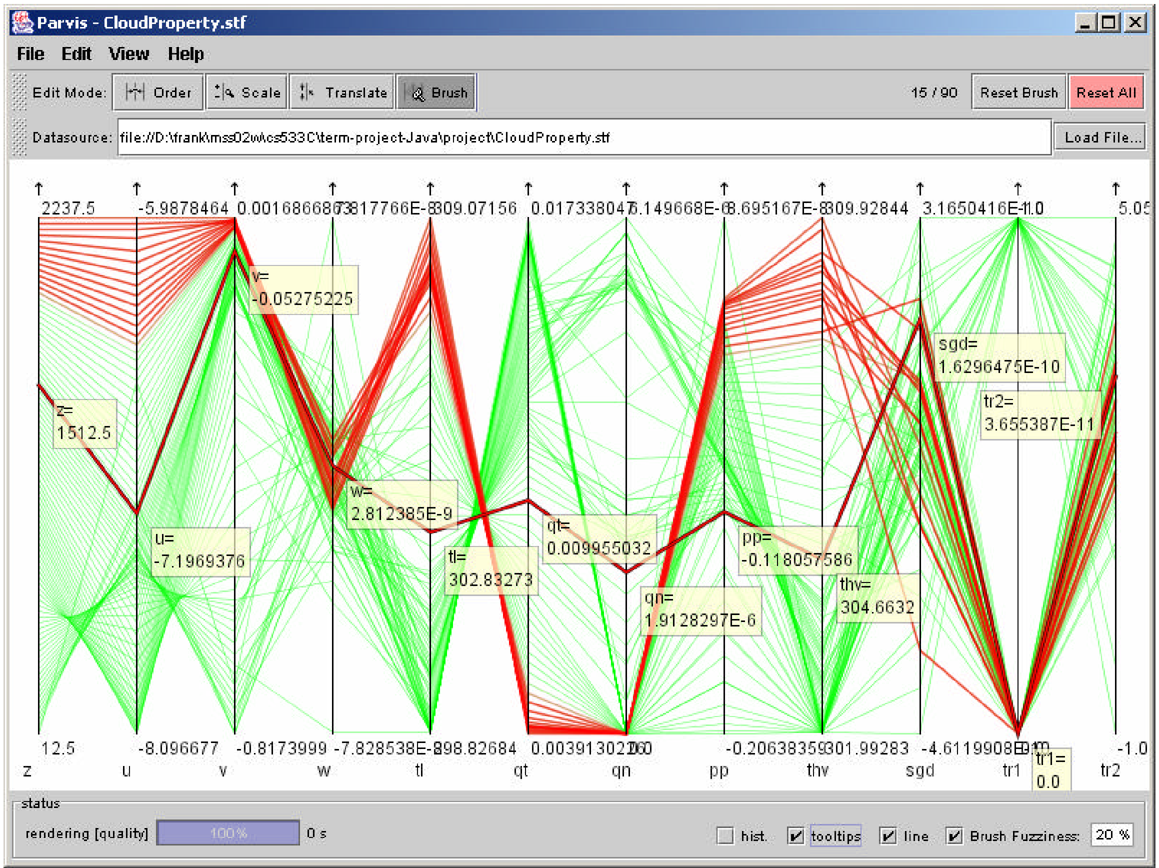
Screenshot 7 shows the dynamical slicing of the 3D volume rendering. On the left figure we can see a gray slicer and a red dot which is a handle to move the slicer. By moving the slicer, we can get a slice of the volume rendering to display on the left figure. The left 2D plot will change dynamically with the movement of the slicer. This feature is important because it enables the user to see the inside of the volume rendering slice by slice.

Screenshot 8: Dynamically change the color and Alpha (2D slice) (meridional flow)



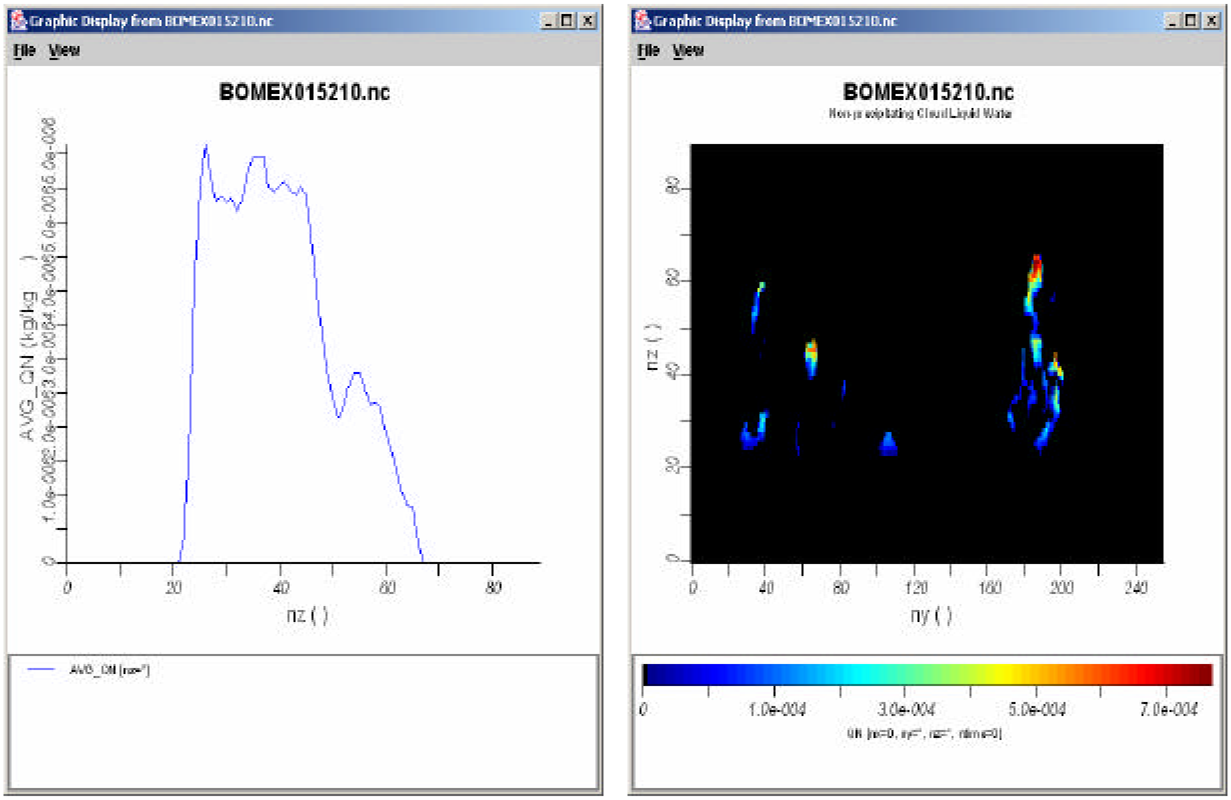
Screenshot 8 shows that we can also dynamically change the color and Alpha value of the sliced piece.

Screenshot 9: Parallel Coordinates (with parvis)



Screenshot 9 shows the display of the 12 variables of the cloud model output data with the parallel coordinates techniques. The 12 variables are altitude (z), zonal wind (u), meridional wind (v), vertical motion (w), liquid water temperature (tl), total water vapor (qt), cloud (qn), pressure (pp), etc. With help of the parallel coordinates, we can clearly see the relationships of these variables, which cannot be easily seen from the traditional 2D line plots. In addition to the general parallel coordinates display, it also has the features such as brushing.

Screenshot 10: display data in 1D & 2D plot (with ncBrowse)



Screenshot 10 show the 1D and 2D display of the cloud modeling data, with the libraries provided with ncBrowse.

7. Results

We have developed software called VisCMD, which is specifically designed and implemented for visualizing large cloud-modeling dataset. The VisCMD has a user-friendly Graphical User Interface and a complete set of Information Visualization functions, i.e.,

(1) 1D/2D plotting
- 1D/2D graph,
- zooming,
- movable graph title

(2) 3D visualizations
- general 3D display and navigation (rotate, move, zoom in and zoom out)
- dynamically set colors
- dynamically set Alpha values (transparency)
- dynamical slicing

(3) Parallel coordinates
- General parallel coordinates display
- Brushing
- Change the order of axis
- Remove and add axis
- histogram

8. Lessons learned

3D volume rendering and N-dimensional parallel coordinates are the core visualization parts in the VisCMD. In 3D volume rendering, the proper colour RGB settings can let the atmospheric scientists perceive the whole cloud model. But selecting an effective colour scheme is difficult and colour-blinded people must be considered. Overusing colours could make the visualization cluttered and downgrade the readable data patterns. Therefore we finally provide the colour and alpha widget tool to let the user decide the colour settings and transparency. On the other hand, zooming is a good visualization technique to reveal more information, but may not quite useful for the atmospheric scientists because of the possibilities of losing some contexts and increasing clutter, especially in 1D/2D plotting (even though we implemented here).

Parallel coordinates technique is very helpful in analyzing the relationships among multiple variables. But selecting appropriate variables could be difficult and the user has to spend some time to figure out which set of variables are useful and need further analysis.

Using the legacy code (the existing libraries and software tools) achieves fast software development, but the flexibility and performance are significantly affected by those existing tools. Sometimes this may become one of the weaknesses for our software applications developed.

9. Evaluation

The software was tested with the cloud model output dataset provided by the Department of Earth and Ocean Sciences of the University of British Columbia. The results show that VisCMD can let the user interpret the cloud modeling data much better and faster.

The VisCMD can be directly used to visualize the cloud model output data in the Department of Earth and Ocean Sciences (EOS) of UBC to improve the currently used graphics software. At present, EOS/UBC uses software developed with Python to display their cloud modeling data. The software is not very convenient to use because (1) the variables are hard coded inside the program and the user cannot easily select a variable to display. If the model changes the output variables, the software should also be changed; (2) The user cannot dynamical manipulate the 3D rendering; (3) there is no a visualization technique that can be used to visualize multi-variables at the same time. With VisCMD, the user can easily select a variable to display with the friendly GUI and dynamically manipulate the 3D volume rendering and quickly visualize, analyze and interpret the model output data. Also, the user can display multi-variables and analyze their relationships with the parallel coordinates visualization techniques.

The strengths of the software are (1) a user friendly Graphical User Interface (GUI), (2) a complete set of data visualization functions, from 1D/2D plot to 3D visualization and to N-dimensional parallel coordinates, (3) the user can easily manipulate the 3D rendering dynamically (e.g., the dynamical slicing, dynamically change color and Alpha (transparency) values), (4) the parallel coordinates technique can let the user make multi-variable display and more easily find the relationships of these variables.

The weakness of the software are (1) Parallel coordinates display and 3D volume rendering are not connected with each other. So we can not observe how parallel coordinates are dynamically changing while the position of the slicer is changing in 3D volume rendering. However, this function may be useful for perceiving other visual data patterns when observing the multiple relations between 3D space and cloud properties; (2) Due to the limitation of some libraries we used, every time we redraw the 3D volume rending, we need to restart this software program from the beginning, even though we implemented something like “DestroyObject”; but other functions work well without any problem; (3) Volume rendering needs a lot of memory and may be slow even for only one-time step data sets (we presently solved this by reducing the resolutions). But if large amount of multiple time-step data sets are processed, the performance of the VisCMD may suffer a lot. We believe these weaknesses can be solved soon in the future.

Bibliography

[1] Lakshmanan, V. and T. Vaughan, 2003: A Four-Dimensional Multiple-Source Weather Information System for Algorithms and Visualization, *Report of National Severe Storms Laboratory & University of Oklahoma.*

[2] Hibbard, B. , 2000: Confessions of a Visualization Skeptic, *Computer Graphics*, **Vol. 34, No.3**.

[3] McCaslin, P.T., P. A. McDonald and E. J. Szoke, 2000: 3D Visualization Development at NOAA Forecast Systems Laboratory, *New Visualization Techniques*, **Vol.34, No.1**.
[4] Joe Kniss and Charles Hansen, 2002: Volume Rendering Multivariate Data to Visualize Meteorological Simulations: A Case Study, *The Visualzation Symposium 2002* (12/2001).
[5] Fua, Y.-H., Ward, M. O., and Rundensteiner, E. A., 1999: Hierarchical Parallel Coordinates for Visualizing Large Multivariate Data Sets, *IEEE Visualization '99*.
[6] Alfred Inselberg and Bernard Dimsdale, 1990: Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. *IEEE Visualization '90*.
[7] Brewer, C. A. 1999: Color Use Guidelines for Data Representation, *Proceedings of the Section on Statistical Graphics, American Statistical Association*, Alexandria VA. pp. 55-60.
[8] Ware, C., 1999: Information Visualization-Perception and design (Chapter 3: Lightness, Brightness, Contrast, and Constancy; Chapter 4: Color). Morgan Kaufmann Publishers, 272pp.
[9] Vis5D : http://www.ssec.wisc.edu/~billh/vis5d.html
[10] VisAD : http://www.ssec.wisc.edu/~billh/visad.html
[11] GrADS: http://grads.iges.org/grads/grads.html
[12] NCL: http://ngwww.ucar.edu/ncl/
[13] NCAR Graphics: http://ngwww.ucar.edu/ng4.3/
[14] parvis: http://home.subnet.at/flo/mv/parvis/
[15] ncBrowse: http://www.epic.noaa.gov/java/ncBrowse/
[16] HTTPClient: http://www.innovation.ch/java/HTTPClient/
[17] Scientific Graphics Toolkit for creating interactive graphics applications http://www.epic.noaa.gov/java/sgt/
[18] UCAR: http://www.unidata.ucar.edu/community/committees/metapps/docs/ucar/
[19] http://tmap.pmel.noaa.gov/~callahan/project_management/projects/GODAE_APDRC/IPRC-PMELprogresst.htm
[20] ncBrowse: OPeNDAP Server Access and 3-D Graphics http://ams.confex.com/ams/annual2003/techprogram/paper_56599.htm
[21] 3-D Visualizations with a Networking netCDF Data File Browser: ncBrowse FY 2002 Proposal to the NOAA HPCC Program
[22] http://www.cio.noaa.gov/hpcc/projects/200217.html