A Reprocessing Tool for Quantitative Data Analysis in a Virtual Environment

E. J. Griffith, M. Koutek, and F. H. Post
Data Visualization Group
Delft University of Technology
Netherlands
{E.J.Griffith, M.Koutek, F.H.Post}@ewi.tudelft.nl

T. Heus and H. J. J. Jonker Department of Multi-Scale Physics
Faculty of Applied Sciences
Delft University of Technology
Netherlands
{T.Heus, H.J.J.Jonker}@tnw.tudelft.nl

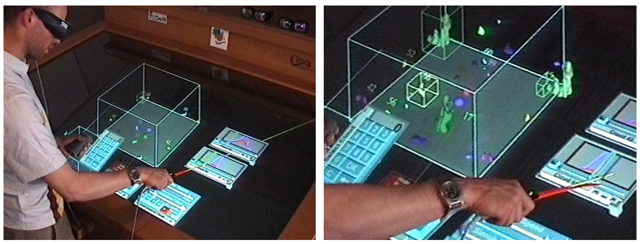

Figure 1: The exploratory VE in use.

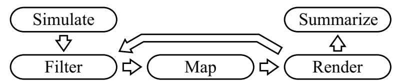

Figure 2: The analysis cycle as described by Upson et al. in [12]. Simulation data is filtered, mapped, rendered, and studied yielding insight, which can be used to initiate a new round of filtering.

ABSTRACT

This paper presents an approach to help speed up and unify the exploration and analysis of time-dependent, volumetric data sets by easily incorporating new qualitative and quantitative information into an exploratory virtual environment (VE). The new information is incorporated through one or more expedited offline "reprocessing" steps, which compute properties of objects extracted from the data. These objects and their properties are displayed in the exploratory VE. A case study involving atmospheric data is presented to demonstrate the utility of the method.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Virtual Reality; I.3.8 [Computer Graphics]: Applications

General Terms: Design, Experimentation

Keywords: Virtual Reality, Data Visualization

1. INTRODUCTION

Many researchers make use of simulations to analyze various phenomena. These simulations often produce time-dependent data sets, which are growing larger, and the time required to extract meaningful results from the data is also increasing. Virtual reality (VR) can help scientists make sense out of such data sets, but most VR visualization software is designed for specific problems and often lacks integration into the larger data analysis process.

This paper proposes an approach to help speed up and unify the analysis and exploration of large, time-dependent data sets. This is accomplished through the combination of an exploratory virtual environment (VE) and a data processing tool. The VE provides standard manipulation and data probing tools for exploring the 3D, multivariate data, and it incorporates quantitative and qualitative information generated by the data processing tool (Figure 1). The data processing tool generates this information by using a simple expression parsing grammar to perform a variety of computations on the data. Furthermore, the same data processing tool can be reused during an arbitrary number of "reprocessing" steps to generate new output data for inclusion in the VE. Currently this takes place offline after examining the data in the VE, but we plan to incorporate it as an interactive, real-time component of the VE in the future.

The remainder of this paper is organized as follows. We discuss related work in the next secion. In Section 3, we give an overview of our approach. In Section 4, we present our software system and the salient points in its evolution from our previously developed VE exploration tool for life-cycle studies [5]. In Section 5, we describe how this work relates to the atmospheric research in our cooperative project. In Section 6, we conclude and discuss our plans for creating a fully interactive system from the current one.

2. BACKGROUND AND RELATED WORK

Upson et al. [12] and Springmeyer et al. [11] characterize the data analysis process. Upson et al. describe the process as largely a filter, map, render loop (Figure 2), the insight of which drives further iterations of the loop. Springmeyer et al. further break the process down into four components, which encompass a variety of research tasks: analyzing representations of the data, performing calculations, maneuvering through and in the data, and expressing the ideas

VRST'06, November 1–3, 2006, Limassol, Cyprus.

gleaned from the process. They derive a set of five functional requirements for data analysis software from these requirements: allow interactive, quantitative exploration; assist in maintaining records of sessions; link materials from different stages of a study; simplify navigation requirements; and provide support for culling large data sets. We try to incorporate the salient points from these models in our software.

A variety of exploratory data visualization environments exist. One Popular is AVS [12]. Much VR visualization work has also been done. See [2, 13] for several examples. However, there are still open problems in visualization. Some listed by Johnson [8] are: effective visualization of time-varying, multivariate data; effective interaction tools for 3D data exploration; and application integration within the overall problem solving environment. With our method, we attempt to address these challenges by more closely coupling the data processing and the exploratory VE.

In a process of "selective visualization", van Walsum [14] selects important parts of the data based on the results of derived data computed by parsing expressions. Henze [6] extended this idea to include exploration of the derived data in user-defined 2D coordinate spaces linked to the original data. In our approach, a similar expression parsing facility is used to calculate new data, but in our case the new data are imported in the VE, so the full range of visualization and exploration techniques is available to the user.

Our "reprocessing" bears some similarities to computational steering. See [9] for some early examples. A critical difference, though, is that computational steering affects the result of the simulation, whereas our approach works with the results of a completed simulation run.

3. METHOD OVERVIEW

Our previous Cloud Explorer application [5] supported a process shown in the top of Figure 3. This linear approach culls the data set by having scientists identify the portions of the data containing interesting objects, but the narrow focus of the preprocessing and VE make it difficult for the scientists to gain further insight into the data.

Our newly proposed pipeline incorporates five steps:
Step 1: Simulation
Step 2: Data (re)processing
Step 3: Data visualization
Step 4: Repeat Steps 2 and 3 as needed
Step 5: Generate quantitative results.
This approach is shown in the bottom of Figure 3. Besides generalizing away from clouds, the most important change is the data "reprocessing" cycle with the optional jump from Step 4 to Step 2. With this, the new pipeline closely parallels the Upson et al. [12] analysis cycle (Figure 2).

The initial data processing phase identifies important objects in the data. The lifespans and bounding boxes of these objects are determined and saved to disk. Optionally, new copies of the raw data are created containing only the data set contained in object bounding boxes. This can greatly speed up later processing of the data as scientists are most interested in objects and their immediate surroundings.

The data reprocessing step is the key new step in the method. The data processing tool can be used to generate quantitative data based on user supplied expressions. The initial processing step lets the reprocessing proceed quite quickly. The new information can then be directly loaded into the VE. This allows researchers to spend more time in

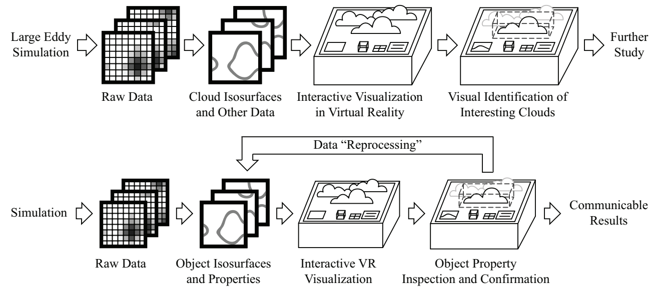

Figure 3: Top: This pipeline was supported by our original Cloud Explorer application [5]. Bottom: This represents our newly proposed pipeline, which incorporates a "reprocessing" cycle.

<input path="**/data/simulation001**">
<variable name="**ql**" file="**ql.001**" />
<time steps="**2000**" />
<grid x="**256**" y="**256**" z="**160**" />
</input>

Figure 4: This example snippet of the input section from a processing specification file describes the grid size, the number of time steps, and the name and location on disk of one simulation variable.

the VE, while also making that time more productive.

The last step is the creation of communicable results, in the form of plots, numbers, or other representations. Presentation quality results often require precision, flexibility, and functionality beyond that which is sufficient for data exploration. This is currently left as a post-processing step for more specialized tools. In the future, however, we plan to include more of this functionality in the VE.

4. SOFTWARE SYSTEM

Our new software can be seen as an extension and generalization of our Cloud Explorer application [5].

4.1 Preprocessing Generalization

Cloud Explorer was designed for our atmospheric research project. While successful, it was not broadly applicable because of restrictions placed on the data format and nature.

In the new system, most of these restrictions have been eased or eliminated. The data can now be periodic in any direction, or it can be completely aperiodic. The objects in the data are now identified by a user specified threshold, and they no longer need to have manifold surfaces. Also, more allowances are now made for varying file formats.

The data processing program loads a processing specification file, which is in extensible markup language (XML) [1] format. This file is broken into three sections, which describe the input data (the simulation data), the desired output data, and mappings from the input data to the output data. Figure 4 illustrates an example input section.

4.2 Preprocessing Extension

The first extensions to the data processing are the creation of subvolumes and downsampled volumes. The downsampled volumes are for slicing tools. The subvolumes are

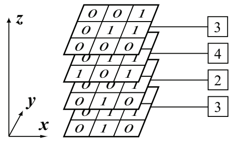

Figure 5: An illustration of a vector value. Here, each value in the vector represents the sum of all values in the corresponding x-y voxel plane.

used for per object data calculations, and they each contain an object and the volume around it in a particular time step.

For data calculations, we have implemented a small expression grammar. It supports simple mathematical expressions involving functions and scalar, vector, and multidimensional matrix values. A set of functions has been defined to support various operations. These include summing and averaging data; basic image processing functions; detecting maxima, minima and extents; and subvolume extraction. Gradient operations are planned for future versions.

In the processing specification file, the user supplies the input variables, some extra information, and a set of expressions describing the mapping from input data to output data. In addition, the user lists the output data to save.

The data processing program generates two types of data. The first is a per object per time step scalar value, which allows 2D plots over time. The second is a per object per time step vector value. The vectors are filled with one scalar value for each voxel plane in a subvolume along a coordinate axis (Figure 5). Taking the example of the z axis, an image can be generated where the pixel coordinates represent time and vertical height in the subvolume and the pixel color or intensity represents the value for that voxel plane at that point in time. Figure 8 illustrates such z vector images.

Consider the following set of expressions:

$$\begin{aligned}
bin_grid &= \mathbf{bin}(grid) \\
volume &= \mathbf{sum}(bin_grid) \\
ql_vec &= \mathbf{sumdim}(\mathbf{sumdim}(grid, 0), 0)\ / \\
&\quad\ \mathbf{sumdim}(\mathbf{sumdim}(bin_grid, 0), 0)
\end{aligned}$$

Here, *grid* is the input subvolume, which has had all non-object voxels set to 0. The **bin** function converts all non-zero voxels to a 1, and the **sum** function sums the values of all voxels in *bin_grid*. Thus, *volume* is the object volume in voxels. **sumdim** sums along the specified dimension, reducing the dimensionality by one. In this case, the effect is to set *ql_vec* equal to the average liquid water value at each height in the subvolume. See Figure 8 for an example of the result.

The final extension was the "reprocessing". For reprocessing, the processing description file is updated and the data processing program run again. The newly generated data is added to the existing data for inclusion in the VE.

4.3 Cloud Explorer Expansion

To complete the generalization and expansion process, we updated the Cloud Explorer interface with new widgets, added new data probing tools, added various windows, and included a slicing plane. We now use the IntenSelect interaction metaphor in the environment to make selection and manipulation tasks easier, especially with small widgets. See [4] for a description of the widgets and the IntenSelect technique. Figure 6 compares the old and new interfaces.

One addition of note is the new graph window. For each

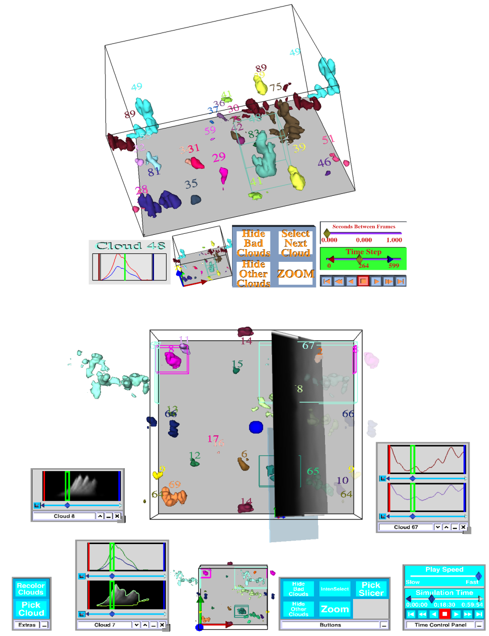

Figure 6: Top: Old Cloud Explorer interface. Bottom: New application interface, featuring windows, a slicing plane, and the new graph window.

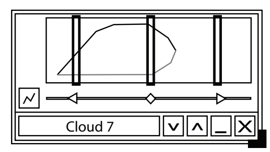

Figure 7: An illustration of the graph window.

selected object, up to five, a graph window is displayed with information about the object. Figure 7 illustrates the window and Figures 6 and 8 show examples of the actual windows. The window is divided into one or more plot display areas, which each display one or more overlayed user selected plots. The window can be resized, and plot display areas can be added or removed in each graph window. The current time step, and the playback timespan are also highlighted on the plots. Using these graph windows, scientists can make correlations between object behaviors over time.

5. CASE STUDY

In our research, we use a Large-Eddy Simulation (LES) of the Atmospheric Boundary Layer (ABL) containing shallow cumulus clouds. LES is a simulation method where the ruling Navier-Stokes equations are solved up to a certain scale, while the influence of smaller scale (turbulent) motion is approximated via a statistical model. The simulations generate a $256 \times 256 \times 160$ output grid every third time step for several variables: temperature, amount of liquid and gaseous water, buoyancy, and air velocity. For more details about LES see [3] and [10], and for more details about our experiments and previous work, see [5] and [7].

In our analysis, we first identified interesting clouds for further study, which was the focus of our previous work [5, 7]. In short, cloud isosurfaces, bounding boxes, and volume

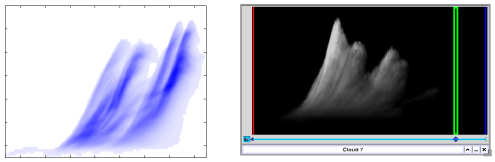

Figure 8: Left: Horizontally integrated liquid water vs. height and time. This plot was generated via a special tool. Darker areas indicate a higher concentration of liquid water. Each point represents the average amount of liquid water at a given height in the cloud at a particular time step. Right: An equivalent plot generated by our new software and displayed within the VE.

information were interactively visualized in the Cloud Explorer application, which enabled the visual identification of 40 interesting clouds for further study.

Next, we studied the interesting clouds. This proceeded iteratively, with each analysis inspiring the next. We started with simple properties (e.g. how the cloud base and top moved vertically over time) indicative of interesting behavior. This led to an examination of liquid water amounts at various heights in the clouds due to its influential role in cloud dynamics. Surprisingly, the liquid water seemed to be concentrated in pulses that rose with time (Figure 8). We confirmed this by performing the same analysis with buoyancy, which led us to discover a correlation between the amount of water beneath the clouds and pulse onset. The consistency of these observations across the clouds we studied let us say that this was a generic phenomenon rather than specific to a particular cloud.

This iterative advancement of ideas is at the core of our proposed system. The initial object identification remains largely unchanged, but the analytical data generated during the later steps is something that our new software supports. The height of the cloud base and cloud top are scalar values, represented by the maximum and minimum extent of the object in a given time step, which can be calculated by the data processing. The horizontal integration of liquid water and buoyancy can be calculated with the vector calculations supported by the data processing. Figure 8, illustrates the original liquid water integration plots and the equivalent plots generated by the data processing as seen from within the VE. The slicing plane can be used to examine the amount of water beneath the clouds. Looking at the integration plot of liquid water while using the slicing plane shows the build up of liquid water preceding the appearance of the pulses in the clouds.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an approach geared towards helping scientists working with large, time-dependent data sets reach results more swiftly. The approach uses the idea of "reprocessing" the data and incorporating new quantitative information into a scientific visualization virtual environment. This allows researchers to spend more time in the VE, while also making that time more productive.

We described the expansion and extension of our previous software to meet the objectives of the proposed method. We restructured the data processing to handle a variety of data sets, and we incorporated a small expression grammar to generate quantitative data from the data sets. Lastly, we updated the VE to incorporate this new quantitative data, among other improvements. We presented a case study relating how our research into cloud life-cycles progressed, and we described how our new software supports each step of the exploration process we went through.

In the future, we would like to move toward a VE capable of supporting virtual experiments, which would allow scientists to develop and new hyphotheses within the VE. To do this, plan to do three things. First, we plan to refine and polish our software through practical application. Secondly, we plan to integrate the "reprocessing" in real-time in the VE. Finally, we would like to introduce further enhancements such as particle tracing, object surface properties, and intelligent large data handling capabilities.

7. ACKNOWLEDGMENTS

We would like to thank the Netherlands Organization for Scientific Research (NWO) for providing project funding. We would also like to thank Gerwin de Haan for his valuable insight and comments on the paper.

8. REFERENCES

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (XML) 1.0 (third edition). http://www.w3.org/TR/REC-xml/, February 2004.

[2] C. Cruz-Neira, et al. Scientists in wonderland: A report on visualization applications in the CAVE virtual reality environment. In *IEEE Visualization*, pages 59–66, October 1993.

[3] J. Cuijpers and P. Duynkerke. Large-eddy simulation of trade wind cumulus clouds. *J. Atmos. Sci*, 50:3894–3908, 1993.

[4] G. de Haan, E. J. Griffith, M. Koutek, and F. H. Post. Hybrid interfaces in ves: Intent and interaction. In R. Hubbold and M. Lin, editors, *Virtual Environments 2006*, pages 109–118, May 2006.

[5] E. J. Griffith, F. H. Post, M. Koutek, T. Heus, and H. J. J. Jonker. Feature tracking in VR for cumulus cloud life-cycle studies. In E. Kjems and R. Blach, editors, *Virtual Environments 2005*, pages 121–128, October 2005.

[6] C. Henze. Feature detection in linked derived spaces. In *IEEE Visualization*, pages 87–94, 1998.

[7] T. Heus, H. Jonker, E. Griffith, and F. Post. Lifecycle analysis of cumulus clouds using a 3D virtual reality environment. In *Boundary-Layers and Turbulence*, 2006.

[8] C. Johnson. Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, 2004.

[9] B. McCormick, T. DeFanti, and M. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6), November 1987.

[10] A. P. Siebesma, et al. A large eddy simulation intercomparison study of shallow cumulus convection. *J. Atmos. Sci.*, 60(10):1201–1219, 2003.

[11] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In *IEEE Visualization*, pages 235–242, October 1992.

[12] C. Upson, et al. The application visualization system: a computational environmentfor scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, July 1989.

[13] A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, and R. M. Simpson. Immersive VR for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, 2000.

[14] T. van Walsum. *Selective Visualization Techniques for Curvilinear Grids*. PhD thesis, Delft University of Technology, December 1995.