

Interactive Exploration of Scientific Data Using Virtual Reality Methods

Oliver Kreylos*

Bernd Hamann*

Terry J. Ligocki†

Wes Bethel†

1 Introduction

A lot of scientific research these days is performed by gathering and analyzing hideous amounts of data. No matter if this data is the result of real-world measurements or of numerical simulations, extracting information from it becomes more and more difficult as the amount of data grows. Thus, scientific visualization is becoming a major tool of research: The human visual system is unparalleled in its capacity to see patterns or detect features in vast amounts of data.

1.1 Visual Exploration

In this paper I give an overview of a special area of scientific visualization: visual exploration. In this area, a user (generally a scientist) is confronted with unknown data from an experiment or simulation, and the task is to understand the phenomenon behind the data by examining the data.

For example, a data set might be the result of a numerical simulation of air flow around an aircraft wing, and the task is to find out whether the wing design fulfills the required aerodynamic properties and how the wing design could be improved. In a standard visualization system, this task would be handled in about the following way:

1. Place some visualization primitives (e. g., stream lines) at points considered "interesting."
2. Generate an image.
3. If the image does not reveal anything exciting, repeat from step 1.

One can immediately see that there are several problems with this approach. First, a user has to specify at which points to place primitives, and specifying positions in 3-space using a 2D input device (e. g., a mouse) is awkward. Second, the process of image generation can be time-consuming, and with placement of primitives essentially being a trial-and-error process, the overall time spent on analysis can grow prohibitively large. Even worse, merely seconds of delay between placing a primitive and seeing the resulting image can be enough to render intuitive exploration of a data set almost impossible. Third, even when an image reveals some features, it is only a 2D projection of a 3D phenomenon and can be misleading as to the feature's spatial relation.

1.2 Benefits of Virtual Reality Techniques

Virtual reality can attack all three of these problems, making the exploration process much more productive. There are probably more contradicting definitions of virtual reality in existence than there are VR researchers; for the purposes of

this paper, every system is considered VR if it at least offers the following features:

- True 3D (stereoscopic) display with head tracking, i. e., the display is dependent on a user's true eye position in space
- Six-degree-of-freedom (6-DOF) input devices
- Interactivity, meaning that the response time of the system to user actions is small enough to seem immediate ($\approx 1/10$ s)

With a system like this, the basic exploration loop shown above stays the same; but with immediate response, a user is now able to move a visualization primitive directly through the 3D data set until something interesting happens. The immediate update of the displayed image, leading to an animation of the visualization primitive's behaviour, gives valuable clues as to where to move the primitive next to home in to a feature.

1.3 Related Work

Virtual reality data visualization is not a new area; the most important contributions so far have been made by NASA scientists: The NASA Virtual Wind Tunnel developed by Bryson and Levit [1] is a visualization system for vector-valued data, and Meyer and Globus [2] describe an interactive isosurface generator for scalar-valued data.

2 Virtual Reality Data Exploration

In this section we describe the two main classes of 3D data sets, and introduce two virtual reality exploration systems we implemented for them.

2.1 Scalar-valued Data Sets

Scalar-valued data sets associate a single real data value with each point in space, e. g., the air temperature in a room, the air pressure of air flow around a wing or the gravitational potential in interplanetary space.

The most commonly used visualization techniques for scalar-valued data sets are volume rendering [3] and the generation and rendering of isosurfaces [4]. To generate the latter in a standard visualization system, a user would specify a certain data value, and the system would then generate a set of surfaces connecting all points in the data set that have the specified value. The standard algorithm for isosurface generation, the Marching Cubes algorithm [5], typically generates millions of small triangles approximating the isosurfaces.

A virtual reality system offers several benefits for isosurface visualization:

- Instead of specifying the isovalue by numerical input, a user can just specify a point in space she is interested in, and the system will calculate the data value at that point and create the appropriate isosurface.

*Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis; {kreylos,hamann}@cs.ucdavis.edu

†Visualization Group, Lawrence Berkeley National Laboratory; {tjligocki,ewbethel}@lbl.gov

- The generated surface can then be inspected by either the user physically moving around the surface, or by moving the surface in respect to physical space.

The main problem to overcome is lack of interactivity: Generating an isosurface takes time on the order of the number of points in the data set; on a typical graphics workstation, this takes several seconds even for smaller data sets. The way to solve this problem is to not generate a complete isosurface, but only a part of it close to the point selected by the user. The isosurface part is grown outwards from the selected point by performing a breadth-first search of all grid cell intersected by the isosurface, starting from the cell containing the selected point. The generation of such a *seeded* isosurface can be stopped at any point, allowing to enforce immediate response by allocating a render budget. The render budget can be prescribed by the user or fixed; in our application, we use a budget of 1/10s to guarantee immediate response. The seeded isosurface algorithm adds more benefits for virtual reality exploration:

- Its fixed response time allows animation of isosurfaces as the user moves the point of interest through the data set.
- The locality of the generated surface highlights the area the user is currently investigating and aids exploration of the whole data set by suppressing surface parts far away from the current area.
- The method is independent from data set size. As the data sets grow, relatively smaller regions are processed, but the amount of detail shown stays about the same.

Our visualization system implements the described isosurface generation algorithm, and offers inspection of the generated surface by moving the data set with respect to the physical world. A user can “grab” the data set and zoom/move/rotate it intuitively using a 6-DOF input device, or she can “fly through” the data set by pointing the input device in the intended direction of flight and controlling movement velocity with a pressure-sensitive joystick mounted on the input device.

2.2 Vector-valued Data Sets

Vector-valued data sets associate a 3D vector with each point in space, e.g., the velocity vector of air flowing around a wing.

Vector-valued data sets are most commonly visualized by placing *primitives* at certain points in space. These primitives can range from simple straight line segments visualizing local vector directions (“hedgehogs”) over integral curves (stream lines and other kinds of pasta) to complicated glyphs visualizing multiple properties, e.g., direction, rotation and divergence, simultaneously. The problems with all these are similar: A user has to find placement points that reveal interesting features of a data set, and the spatial relations of the primitives have to be understood from the resulting images.

Again, virtual reality methods come in to save the day by offering 3D input, 3D display and immediate response. As we found out, especially immediate response sets a virtual reality system apart from a more traditional one; in the former, a user is guided towards interesting placement points by the animation of the visualization primitives as the placement point is moved through space, whereas in the latter finding good points is essentially a trial-and-error process.

The visualization system we implemented offers the following kinds of pasta (primitives):

- Stream lines, “spaghetti” showing the direction of a flow
- Stream ribbons, “fettucine” showing the direction and rotation of a flow
- Stream tubes, “macaroni” showing the direction, rotation and divergence of a flow
- Stream brushes, tufts of “capelli d’angelo” helpful in finding vortices and turbulent regions

The visualization result can be inspected in the same way as in the scalar-value visualization system; additionally, this system provides menu interaction to choose the current primitive from the provided set. The implemented menu system works similar to a standard 2D popup menu; a user can open the menu with an input device gesture at any convenient point in space.

3 Conclusion and Future Work

Early experiments have shown that virtual reality systems are extremely helpful for exploring 3D (scalar- and vector-valued) data sets and offer a new level of insight and intuitive understanding.

Virtual reality systems also create new (and unexpected) problems: They are based on a certain suspension of disbelief on the user’s behalf in order to interact with a virtual environment in a natural way. Now, if the system displays behaviour inconsistent with those unspoken rules, this is not only very distracting for a user but can also induce reactions like motion sickness. In the isosurface exploration system, for example, it often happens that isosurface parts pop in and out of existence right in front of the user’s eyes when the input device is merely moved by a very small distance. This does not happen in the real world, and can startle a user and put strain on the user’s eyes. Another effect is that isosurfaces often curl around themselves and hide the “3D mouse cursor;” this can cause the user getting lost in the data set. One main direction of future research is to find ways to deal with these perceptual issues.

References

- [1] Bryson, S. and Levit, C., *The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows*, in: Proc. of Visualization '91 (1991), San Diego, CA, pp. 17–24
- [2] Meyer, T. and Globus, A., *Direct Manipulation of Isosurfaces and Cutting Planes in Virtual Environments*, technical report CS-93-54 (1993), Brown University, Providence, RI
- [3] Drebin, R. A., Carpenter, L. and Hanrahan, P., *Volume Rendering*, in: Proc. SIGGRAPH '88 (1988), pp. 65–74
- [4] Bloomenthal, J., *Polygonization of Implicit Surfaces*, in: Computer Aided Geometric Design 5(4) (1988), pp. 341–356
- [5] Lorensen, W.E. and Cline, H.E., *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, in: Proc. of SIGGRAPH '87 (1987), pp. 163–169