Visualization in Geometry Research and Education

Position Statement for
Scientific Visualization Environments:
A Workshop at Visualization '91

August 30, 1991

Tamara M. Munzner
munzner@geom.umn.edu

Mark B. Phillips
mbp@geom.umn.edu

The National Science and Technology
Research Center for Computation and
Visualization of Geometric Structures
(The Geometry Center)

University of Minnesota
1300 South Second Street
Minneapolis, MN 55454
(612) 626-0888

Pure mathematics is an area which has remained largely unchanged by computers. In a broad sense, the goal of the mathematician is to explore and discover new mathematical theories using rigorous logic. This involves such mental processes as recognizing patterns, making generalizations, and following hunches, tasks for which computers are generally not very good. Current mathematics papers and books usually contain very few or no illustrations and consist largely of statements of results followed by rigorous proofs. This is true even in geometry, where the very objects of study are frequently visual. This style of mathematics grew out of lessons learned in the last century about how intuition, especially pictures, can be misleading. Mathematicians of the late 19 and early 20 centuries sought to put the subject on a rigorous foundation, and this began a trend towards abstraction which continues today.

This flow towards abstraction has the advantage of providing a strong foundation of clearly stated assumptions upon which to build theories. The process of formalizing an intuitive concept also serves as a guard against incorrect reasoning. It lacks, however, the ability to convey in a transparent way the intuitive ideas behind the theory. Although most mathematical theories begin with a simple, intuitive idea, this idea is frequently lost amidst the formalism of the theory. Mathematicians frequently draw diagrams at the blackboard or on scratch paper while working out theories, and commonly converse with each other using general, descriptive phrases, but these aspects of their work rarely make it into the final product. Consequently, research papers frequently fail to convey the very intuitive ideas which led to the theory they are describing.

Readers are left to themselves to reconstruct the diagrams and intuitions which led to the theory. This trend has also influenced mathematics education and is probably at least somewhat responsible for alienating the public and giving mathematics a reputation as a difficult and unpopular subject.

At the Geometry Center we are trying to change this. In particular, we are exploring and developing ways in which computers can be used in geometry research and education. One of the most exciting aspects of this involves using computer graphics to guide our intuition. The computer can serve well as an information organizer, allowing us to organize, experiment with, and interpret complex examples which would be unmanageable by hand. It can also serve as a window into theoretical worlds, allowing us to see and experience mathematical phenomena to an extent impossible in the "real world".

Most mathematicians, however, are still unaccustomed to the idea of using the computer as a tool for investigating mathematical ideas. The primary reason for this is that doing so in the past involved either writing extensive programs oneself, or learning to use a complicated existing program. Furthermore, until very recently most existing visualization software (e.g. CAD systems) was designed for engineering or industrial users and was not suitable for displaying the abstract geometric objects which arise in the study of mathematics.

One of the tools that we are developing at the Geometry Center to fill this gap is our "Object Oriented Graphics Language" (OOGL). It is a C graphics library with an expandable set of object-oriented primitives, together with MinneView, an interactive 3D viewer. We are seeking to create an environment in which mathematical objects such as functions, manifolds, knots, groups, minimal surfaces, automatic structures, and dynamical systems can be managed and viewed in a natural way. Our interest also extends beyond the objects themselves to the space around them. There are many spaces of mathematical interest besides our familiar "real world" of Euclidean 3-space. To visualize other geometries, such as hyperbolic space, we must redefine the meaning of distance and motion. To explore 4 dimensional objects we must allow 4 dimensional points and provide ways to move in 4-space.

The above examples illustrate the mathematician's need for flexibility and control — a need as yet unmet by scientific visualization systems. The "what-if's" of math and science are not identical: most scientific visualization systems concentrate on the display of huge amounts of data in conjunction with computationally intensive transformations. This data often comes from either from real-world measurements or computer simulations. In contrast, users at the Geometry Center are interested in transforming abstract mathematical definitions into visual representations. To support this, we want a very flexible environment where control over the "laws of physics" is provided. The path of a light ray or the motion of an object can be redefined according to the mathematical space of interest. Our data types must be flexible enough, for example, to allow us to make a hyperbolic camera or multidimensional points.

The goal of OOGL is to free mathematicians to concentrate on the math, without thinking about the graphics, so MinneView is designed to be the graphical front-end for mathematical programs. Ease of use is crucial not only for the user interface but also for the programmer's interface. Our audience of mathematicians should not have to spend more than an afternoon to figure out how to hook up their programs to our visualization environment.

The current version of MinneView, which runs only on SGI workstations, is available via anonymous ftp from geom.umn.edu (128.101.25.31). It is in the file pub/MinneView-0.1.tar.Z. This file contains an executable version of the viewer and several sample data files.

Mark Phillips's Experience in Visualization Software

My experience in visualization software began as a graduate student at the Center for Interactive Computer Graphics at Rensselaer Polytechnic Institute. I developed an interactive program for computing and displaying trajectories of 3D systems of ordinary differential equations. This was in 1983-84; the software ran on an E&S PS300 vector display terminal.

In the summer of 1984 I worked at General Electric R&D in animation. I wrote an interactive Coons patch modeler and used it to make a movie showing the unfolding of a surface in space.

In 1984 I began graduate work in mathematics at the University of Maryland. I specialized in hyperbolic geometry and developed graphics software for drawing objects from hyperbolic space and complex hyperbolic space on Sun workstations. At first these program were menu-driven, but the need for a powerful command language forced me to look into other interface techniques. After several experiments I decided to interface Mathematica to the program, essentially treating Mathematica as a sophisticated front-end for my specialized graphics display programs. This result proved to be the first really comfortable software environment for investigating the mathematical ideas of my research. (For a more detailed explanation of this software system, see the article by me in the second issue of the Mathematica Journal (Volume 1, Issue 2), entitled "Using Mathematica as an Interface to an Interactive Graphics Program".)

I believe that one of the most important issues in the development of visualization software is the integration of easy-to-use graphical front-ends with more powerful and flexible command languages. Most systems succeed at only one of these two styles of user-interfaces. For mathematicians it is especially important that a visualization system provide both, although some operations may only need to be done once or twice, others may need to be executed many times with varying parameters in the context of a series of experiments.

During my graduate work I made several visits to the Geometry Center in Minneapolis and ported my hyperbolic geometry software to IRIS workstations. My colleagues and I made several graphics videotapes which we have used in presentations at mathematics conferences.

During my graduate school years at the University of Maryland I also wrote, as part of my graphics software project, a general-purpose 3D wire-frame viewing program for the Suntools window system on Sun workstations. This program, which I called "viewwld" (a rather awkward abbreviation for "view world"), reads an ASCII list of line segments and points in space and displays the picture in a window. It provides an interactive control panel with buttons for turning the picture around, zooming in and out, etc. Although viewwld's performance can't be compared to what is possible on today's graphics workstations (no shaded or hidden-line images, etc), it proved to be very popular at the time (1988) because it quickly displayed simple to moderately complicated line drawings on widely available low-end Sun workstations. I distributed it on the Internet and received email from people using it all over the world.

I received my PhD in mathematics from the University of Maryland in 1990, and spent the next year teaching in the math department at the University of Richmond in Richmond, VA. During this year we experimented with Mathematica as an aid in teaching calculus; Mathematica's graphics capability formed an integral part of this effort.

I joined the Geometry Center's permanent staff in July of 1991 and have been concentrating on the OOGL project since then. We currently have a version which runs on IRIS workstation, and we are in the process of expanding it and making it easily portable to other platforms.

Tamara Munzner's Experience in Visualization and Graphics

My first involvement in computer graphics came as a summer intern for ETA Systems, Control Data's erstwhile supercomputer subsidiary. I was involved in porting GK2000 and DI3000 graphics libraries to the CYBER205 Supercomputer.

I was an intern at the Geometry Center last summer, when I added RenderMan snapshot capability to MinneView and extensions to OOGL in response to user requests.

I returned to the Center this spring after graduating from Stanford University with a BS in Computer Science and am one of the developers of the new versions of OOGL and MinneView.