Interactive Methods for Visualizable Geometry

Andrew J. Hanson
Department of Computer Science
Indiana University, Bloomington, IN 47405

Tamara Munzner
The Geometry Center
Minneapolis, MN 55454

George Francis
Department of Mathematics
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Title Page Abstract

Interactive computer graphics methods provide new insights into the world of pure geometry.

1 Introduction

Mathematical visualization is the art of creating a tangible experience with abstract mathematical objects and concepts. While this process has been a cornerstone of the mathematical reasoning process since the times of the ancient geometers, the advent of high-performance interactive computer graphics systems has opened a new era whose ultimate significance can only be imagined.

Typical geometric problems of interest to mathematical visualization applications involve both static structures, such as real or complex manifolds, and changing structures requiring animation. In practice, the emphasis is on manifolds of dimension two or three embedded in three or four-dimensional spaces due to the practical limitations of holistic human spatial perception — it is extremely challenging to construct intuitively useful images of anything more complicated! General approaches to visualizing N-dimensional spaces are at best piecemeal, so that algebraic manipulations often remain our most powerful tool for high dimensions. Nevertheless, despite the apparent limitations of visual representations, their utility is far from being completely exploited; we may still gain significant intuitive value by pushing our visual understanding of relatively simple geometric objects as far as our imagination can take us.

Our goal is to show the nature of the interrelationship between mathematics and computer science, especially computer graphics. In this article, we adopt for the most part a computer scientist's perspective on the progress, techniques, and prospects of mathematical visualization, emphasizing those areas of 3D and 4D geometry where interactive paradigms are of growing importance. Without demanding detailed mathematical expertise of the reader, we present a selection of the domains with which we are familiar, describe some of the critical visualization problems involved, and discuss how various researchers have approached the solution of these problems.

The article begins with some general background and then turns its attention to some of the visualization methods that have been used to bring computer graphics technology to bear on mathematical problems of low-dimensional topology and geometry. The concluding sections discuss the system design philosophies of various research groups and prospects for the future. Examples of computer-generated images are supplied throughout, and separate sidebars are devoted to a brief glossary, sources of additional background information on visualizable mathematics, and an overview of selected film and video animations concerned with mathematical visualization.

2 Pictures and Mathematics

Mathematics has a long tradition of interest in visualization methods. The explosive development of geometry in the late nineteenth century was accompanied by intense activity in creating plaster and wire models as well as pedagogical illustrations. The reader need only consult such classic works as *Anschauliche Geometrie* by Hilbert and Cohn-Vossen to see the influence of this visual approach to mathematics; it is worth noting that the English translation [6], *Geometry and the Imagination*, of the German title only barely does justice to the complex nuances of the German *anschaulich*, which involves other connotations, including "vivid," "graphic," "intuitive," "demonstratable," "perceivable in the mind's eye," or perhaps *visualizable*: hence our choice of the phrase *Visualizable Geometry* in the title of this paper.

Despite the fact that pictures scratched on napkins and blackboards never ceased to play an important part in mathematical creativity and intuition, the mathematical literature has been predominantly algebraic for most of the twentieth century. There are good reasons for this: for one thing, it is easy to abuse pictures and convince oneself of false arguments that would not stand up to a formal proof; for another, many interesting current questions involve the properties of spaces so complex that we only know how to treat them algebraically.

It is no accident that the emergence of computer graphics, and especially interactive computer graphics, as a communication medium has coincided with renewed interest in visual mathematics. Many mathematicians harbor the hope that computer graphics technology will have a significant positive influence on the progress of mathematics.

Goals and Methods of Mathematical Visualization If graphics is to contribute to mathematics, we must next ask: What are the principal tasks of mathematical visualization? Like other visualization domains, it must address several levels of problems:

- **Information Content.** Images, animations, and interactive systems involving displays of mathematical objects must possess intrinsically non-trivial information content and reveal this information in some way to the viewer. Since each viewer sees the world differently, the information display should also take into account the beliefs, assumptions, and perceptual preconceptions carried by the viewer.
- **Teaching**. To carry out effective research, one must master what has gone before. Worthwhile visualization techniques attempt to convey a useful and accurate knowledge of the subject material. Computer-assisted methods should enhance breadth, depth, and learning speed compared to standard teaching methods.
- **Insight**. Visual ways of representing objects that the viewer has only studied algebraically should offer previously unattained insights, help expose general principles, and should suggest fruitful conjectures.
- **Results**. The final objective is to prove interesting new results, thereby extending the corpus of known mathematics.

These are lofty and difficult goals, as significant mathematical insights and results are unusual no matter how they are obtained.

Creating mental models. How can we expect computer graphics and interactive graphics methods to help attain these goals? The real power of computer graphics lies in its ability to accurately represent objects for which physical models are difficult or impossible to build, combined with its ability to allow the user to interact with *simulated worlds*. To understand how significant these features are, consider this: our entire common-sense knowledge of the physical world exists *only in our mental models* – we have watched and interacted with the immensely complex laws of physics governing such objects as water, sand, and trees, and have developed, through interaction, a mental picture that enables us to predict with some accuracy what is physically reasonable. Successful systems should include the capability of exploiting old models — creating pictures of unknown domains that exploit our *existing* mental models and perceptions — and creating new models — enabling the development of *new classes* of mental worlds.

3 Problems in Visualizable Geometry

We now turn to a discussion of selected visualization tasks and the approaches that have been used to carry them out. We first consider the depiction of surfaces or 2-manifolds, beginning with familiar surfaces such as the Klein bottle; we are led immediately to consider surfaces in 4D space as well as in 3D because the self-intersections of many classical surfaces in 3D disappear when they are represented in 4D coordinates. We then proceed to a discussion of volumes or 3-manifolds, methods for exploiting 4D lighting models, and the visualization of non-Euclidean geometry. Finally, we turn to surfaces that obey geometric constraints such as minimizing an energy functional, and then to the question of surfaces that change in time subject to the constraint that the curvature remains finite at every point; deformations obeying this constraint, the so-called regular homotopies, play a special role in the development of descriptive topology.

3.1 Manifolds

What is a manifold?

A central visualization problem in pure geometry is to create pictures of manifolds situated in space, as well as images showing how they might look from the inside.

Informally speaking, surfaces are 2-manifolds and volumes are 3-manifolds. A few examples of 2-manifolds constructed by gluing and twisting strips of paper may help clarify the definition in the Glossary. When we glue the ends of a strip of paper together after a twist, we get a (one-sided) surface called a Möbius band. If we glue opposite pairs of edges of a square together, we get the 2-torus, a 2-manifold that looks like the surface of a donut. A manifold is *embedded* when it is situated in a space without self-intersections or singularities. When we give one of the edges of a square a twist before gluing, we get a Klein bottle, which, being a closed, one-sided surface, cannot be embedded in 3-space. We can embed it in 4D space, but its projection to 3D must be self-intersecting. When we glue volumes instead of surfaces, we construct 3-manifolds. Gluing together the walls of a cube in opposite pairs yields a 3-torus. If we twist one volume before gluing, we obtain a 3-dimensional Klein bottle.

Surfaces in 4D

During the seventies, Thomas Banchoff and his collaborators pioneered a mathematical visualization program at Brown university. They achieved interactive, real-time geometrical visualizations of surfaces projected from 4D to 3D with a custom-built matrix multiplier and a fast-refresh vector graphics display for wire-frame modelling. They also explored techniques such as enhancing depth perception by rotating the resulting 3D object at a constant angular velocity, and produced many animations of classical objects such as projective planes and tori projected from 4D to a 3D graphics depiction. In Figure 1, we show an image of a classic Klein bottle generated by this group that exhibits two separate visualization methods: the use of a color code to indicate 4D depth and a surface rendered with alternating transparent ribbons to reveal some internal structure of the self-intersecting surface.

At the University of North Carolina/Chapel Hill, the Fourphront system by David Banks is another interactive system for the study of surfaces in 4D [1]. This system, which runs on the high-speed massively parallel Pixel-Planes 5 graphics engine, provides user control of transparency, depth cueing, intersection highlights, and 2-sided paint as well as supporting user control of rotation and translation. An illustration of Fourphront's alternative approach to the Klein bottle display is shown in Figure 2.

Another influential development was Apéry's presentation in 1984 of a long sought parametric equation for Boy's surface, a classic immersion of the projective plane in 3D whose purely topological description dates to the turn of the century. Apéry also formulated the "Romboy homotopy" a smooth deformation between his own parametrization of Boy's surface and Steiner's Roman Surface, another projective plane whose parametrization as a 4D surface had been known for a century. George Francis and his collaborators Donna Cox and Ray Idaszak exploited this work to create an animation, "The Etruscan Venus," whose title refers to the suggestive shape of a singular Klein bottle that appears in the deformation. An additional deformation leads to a new, highly symmetrical immersion of the Klein bottle dubbed "Ida," which graced the cover of the August 1989 issue of IEEE *Computer* . The animation editor used to produce the Etruscan Venus and Ida has evolved into the highly interactive real-time illiView collection of mathematical animators, produced principally by students taught by George Francis in the Renaissance Experimental Laboratory at the National Center for Supercomputing Applications (NCSA) at the University of Illinois. Figure 3 shows an example of a current illiView visualization.

In 1987, the Geometry Supercomputer Project was created at the University of Minnesota, and subsequently evolved to become the Geometry Center, a National Science and Technology Research Center, in 1991. The Center has served as a focal point for a number of efforts in mathematical visualization; a widely used system distributed by the Geometry Center is *Geomview* [9], a very general surface viewer developed by Stuart Levy, Tamara Munzner, and Mark Phillips. Its built-in functionality can be extended by customized user programs, called *external modules.* While Geomview is fundamentally a 3D viewer, the 4DView external module by Daeron Meyer supplied with Geomview accepts 4D data points, allows the user to change the 4D viewpoint, and includes tools for creating 4D slices. Another external module, NDView by Olaf Holt and Stuart Levy, interacts with objects of dimension 4 and higher using multiple projections into families of 3D subspaces. A typical Geomview surface display is shown in Figure 15.

A 4D viewer with a different philosophy is the MeshView program designed by Hui Ma and Andrew Hanson of Indiana University. This system supports the Geomview 4D surface mesh data format, but in addition provides a high-speed mouse-driven 4D rotation interface and a utility for locating particular points on a projected surface relative to the abstract parametric mesh coordinates. Figure 4 illustrates a MeshView display of the $n = 4$ case of a closed-form construction developed by Andrew Hanson for representing the complex "Fermat" equations $(z_1)^n + (z_2)^n = 1$; the surface is projected from 2 complex dimensions to 3 real dimensions from any desired viewpoint. Large families of complex surfaces can be displayed interactively in MeshView using this technique.

Knotted Surfaces in 4D

An important special case of surfaces in 4D is the subject of knotted surfaces. While closed curves are knottable in 3D, smooth curves (whether or not they are thickened) can always be untied without self intersection in 4D. However, surfaces *can* be knotted in 4D. Some surfaces in 4D appear to be knotted but are really unknotted. They can be "untied" in principle by a series of deformations developed by Dennis Roseman during which the surface does not develop self-intersections; such deformations are examples of *isotopies*.

The important topological as well as graphical problem is that of determining which *a priori* characteristics of an apparently knotted surface guarantee that it is isotopic to another surface; of particular interest is determining whether a surface is isotopic to an embedded sphere, and thus unknotted.

Examples of strategies for understanding these issues range from analyzing 4D slices of the surface and projecting them to 3D, as shown in Figure 5, to showing cutaway interiors or providing above-below crossing markings on the self-intersections of the 3D projection, as shown in Figure 6. The latter may be thought of as a special case of the color coded 4D depth method in Figure 1; Roseman has also experimented with the use of varying 4D-depth-keyed *texture sizes* to appeal to "near" and "far" visual preconceptions, illustrated in Figure 5. In the next section, we note a method involving thickened surfaces that automatically provides occlusion cues on the 3D image of the 4D structure; these occlusions are analogous to the 3D occlusion cues observed in 2D images of curves thickened into tubes.

3-manifolds

When we extend our domain from 2-manifolds to 3-manifolds, we are confronted with the problems of volume visualization. The traditional surface visualization approach is to embed the 2-manifold in 3-space and let the user fly around in the empty spaces, viewing the manifold from the outside. This is harder for 3-manifolds, but still feasible if one can do rapid volume rendering. In essence, one projects from 4D to 3D, treating space as a photosensitive medium that one can also fly through. In [5, 4], Hanson, Pheng, and Cross introduce "outside viewer" techniques that allow interaction with 4D-lit, thickened 2-manifolds as well as moderately complex tessellated 3-manifolds.

Charlie Gunn's imaging system, Maniview [3], an external module of Geomview, takes an alternative approach that dates to Bernard Riemann, the founder of manifold theory. The viewer is placed *inside* the 3-manifold, with no notion of an embedding in some ambient, higher-dimensional space. This is an elegant, mathematical solution because it avoids artifacts of any particular embedding. One can interact with an environment that is 3-dimensional, just like our familiar 3-space, but with surprises. For example, the barbershop experience of sitting between two parallel mirrors is similar to being inside certain 3-manifolds, provided you ignore images of yourself that face you. What seems subjectively like being inside a vast, repeating volumetric tiling of space is objectively the gluing of one wall to another to form a 3D cylinder. Conceptually, the "insider's view" obtained by this approach is an infinite tessellation of space. Of course, the tessellation drawn by Maniview must be finite, but the combination of a large tessellation radius with light attenuation yields a convincing picture. In Figure 11, we see Maniview's representation of life inside a 3-dimensional Klein bottle.

Viewing with 4D Light.

A perceptual capability finely honed in humans is the ability to perceive 3D shape from 2D shading information such as one might see in a photograph or painting. Traditionally, surfaces projected from 4D to 3D have been illuminated by 3D lighting models to generate the rendered shading. We can generalize this procedure and systematically compute the properties of shaded images of illuminated 4D objects in an attempt to recover some of this intuitive perception. The image of a 2D world is a projection to 1D film, 3D worlds project to 2D film, and 4D worlds project to 3D film, a volume filled with points of light. Volumes differentially reflect 4D light to give changing shades in the projected 3D volume image just as faces of a 3D polyhedron reflect 3D light to give different shades in the 2D image plane.

A common technique for viewing 1D curves in 3D graphics is "tubing," which thickens each point on a curve by adding a disk; the boundary or outer skin of this solid fiber is a finite cylindrical surface that can now be rendered by standard methods; 3D shading and occlusion cues can now be computed directly. In [5], Hanson and Heng propose an analogous technique for 4D shading: thicken a surface embedded in 4D by adding a shiny circle at each point, illuminate with 4D light, depth buffer the projection to a 3D volume image, and volume render. The 4D depth buffer in principle produces precisely the same type of characteristic occlusion cues that 3D rendering produces for tubes. However, the full method using volume imaging, 4D occlusion calculations, and a final volume rendering step is very time consuming. In [4], Hanson and Cross introduce new techniques that are fast enough to use for interactive 4D visualization in virtual reality environments. In Figure 12, we compare the low-resolution, time-consuming, full volume image of the thickened surface to the fast approximation that computes a texture map for an ideal, infinitesimally thickened surface illuminated by 4D light and then projected from 4D to 3D.

Non-Euclidean Geometry

Mathematicians in the nineteenth century showed that it was possible to create consistent geometries in which Euclid's Parallel Postulate was no longer true. Absence of parallels leads to spherical, or elliptic, geometry; abundance of parallels leads to hyperbolic geometry. By mid-century the English mathematician Arthur Cayley had constructed analytic models of these three geometries that had a common descent from projective geometry, which one may think of as the formalization of the renaissance theory of perspective. Cayley's construction is in fact ideal for programming interactive navigations of nonEuclidean geometries [10].

Many manifolds are naturally suited for hyperbolic or spherical, rather than Euclidean, geometry. Although the formulas for computing distance and angles in these geometries differ from Euclidean geometry, they can be built into mathematical visualization systems by hand. Translations, rotations, and dot-products for shaders and illumination must also be handled differently in the non-Euclidean geometries. In Figure 7, we see a partial tessellation of regular right-angled dodecahedra in the three built-in Geomview models of hyperbolic 3-space. In the virtual model shown in Figure 7a, the user can fly around to see the tessellation from the insider's point of view. The other two models are from the outsider's point of view, and we can see the position of the insider's camera marked by the blue X.

In the projective (Beltrami-Klein) model of hyperbolic space, shown in Figure 7b, geodesics (the paths of light rays) are Euclidean straight lines, while angles are non-Euclidean in character. In the conformal (Poincaré) model, on the other hand, angles are measured by Euclidean protractors, but light rays travel along circular arcs as in Figure 7c. Figure 8, from the animation *Not Knot* (Sidebar B), shows us what an insider in an interesting hyperbolic 3-manifold would experience. This 3-manifold is obtained by gluing the faces of a right-angled dodecahedron (an impossibility in Euclidean space) in a way analogous to our imaginary barbershop.

The projective models of non-Euclidean geometry can be represented by 4×4 real matrix transformations on homogeneous coordinates that are serendipitously supported by today's computer graphics transformation hardware and software [10]. The conformal model, however, cannot be implemented using 4×4 real matrices, so standard computer graphics hardware does not suffice. Furthermore, the lines between edge endpoints as well as the faces themselves are curved, thus requiring subdivision of polygon edges and faces into small line segments and polygons in order to draw them as curves in computer graphics. Thus, the graphics in the conformal model is considerably slower than in the other two models.

In all cases, however, the standard built-in illumination computations are implicitly Euclidean; correct rendering of surface shading requires custom software shaders that use alternative inner products for computing distances and angles in non-Euclidean geometry. Such shaders have been implemented and shown in [3] to impose only moderate performance penalties in comparison to the built-in Euclidean ones.

Such interactive tools for the exploration of non-Euclidean spaces show clearly how computer graphics can allow humans to experience worlds that otherwise would not be accessible. Direct manipulation of non-Euclidean rotation and translation allows us to develop an intuition for non-Euclidean behavior that we would be hard put to gain in any other way. For research mathematicians, it provides a microscope for investigating the diverse world of three-dimensional manifolds.

Not all technical problems in geometrical visualization stem from complicated manifolds, dimensions higher than three, or the challenge of representing non-Euclidean geometry. Problems imposing strong geometric constraints on the evolution of surfaces in ordinary 3D space are also of great potential interest. The remaining parts of this section deal with two important domains of this type: minimal surfaces — surfaces that minimize curvatures or energy functionals — and regular homotopies — manifold deformations whose curvatures remain finite at every point. The latter treatment focuses on a particularly fascinating regular homotopy, the sphere eversion.

3.2 Minimal Surfaces

The study of *minimal surfaces*, also known as *optimal geometry*, is a branch of *differential geometry*, because the methods of differential calculus are applied to geometrical problems. One of the oldest questions here is: “What is the surface of smallest area spanning a given contour?” The question is nontrivial despite the fact that every physical soap film appears to know the answer. Unbordered minimal surfaces have the property that each point is the center of a small patch that behaves like a soap-film relative to its boundary contour. From the point of view of local geometry, a minimal surface is equivalently described as one that is equally bent in all directions so as to have zero average curvature, e.g., a saddle shape.

The field of minimal surfaces has been one of the success stories of mathematical visualization: insights gleaned from computer graphics tools have led directly to concrete results and theorems.

Previously unknown and certainly unexpected minimal surfaces were found by David Hoffman and his collaborators at GANG, the Center for Geometry, Analysis, Numerics, and Graphics at the University of Massachusetts in 1985. They first used their MESH computer graphics system to find these surfaces, and then later proved their existence with fully rigorous mathematics. This truly excited the minimal surface community and piqued their interest in computer graphics. In Figure 10, we show new surfaces recently given by Hoffman, Wei, and Karcher [7]; the one-hole surface is the first complete, properly embedded minimal surface of finite topology and infinite total curvature to be found since the discovery of the helicoid in the eighteenth century.

The “Minimal Surfaces Team” of the Geometry Center consists of mathematicians and applied mathematicians modeling equilibrium and growth shapes of surfaces such as occur in soap bubbles and crystals. It includes Jean Taylor of Rutgers, Fred Almgren of Princeton, Ken Brakke of Susquehanna University, and John Sullivan of the University of Minnesota. Recently, Brakke's Surface Evolver was instrumental in finding a counter-example to an 1887 conjecture of Lord Kelvin. A partition of space into equal volume cells was found with less interface area than one conjectured by Lord Kelvin to be minimal in 1887. The Surface Evolver has also been used outside of pure mathematics, by an engineer at Martin-Marietta to aid in the design of rocket-fuel tanks where surface tension is the only force available to guide fuel to the intake valve in low-gravity conditions (see Figure 9). Since the end of the 1980's, other groups in Berlin and Bonn working mostly in the area of minimal/optimal surfaces have developed systems tailored to their visualization needs that have led to significant results [11].

Software Systems. Many problems in optimal geometry require specialized software systems because there is often no explicit parametrization of a desired minimal surface. Here we take note of some of the specific software systems being used in minimal surface research.

One tactic for generating minimal surfaces is to evolve a given initial surface to minimize *energy*, such as surface tension. Ken Brakke's Surface Evolver [2] is an interactive program that evolves a surface toward minimal energy by a gradient descent method. The energy in the Evolver can be a combination of factors such as surface tension, gravitational energy, squared mean curvature, user-defined surface integrals, or knot energies. The user can interactively modify the surface to change its properties or to keep the evolution well-behaved. The Evolver was originally written for one and two dimensional surfaces, but it can handle higher dimensional surfaces with some restrictions on the features available. A limitation on the Evolver is the requirement that it be given an initial combinatorial structure.

Another approach is taken by the University of Massachusetts MESH system, which generates triangulations of parametric surfaces defined by conformal mappings from a two-dimensional domain to a three dimensional range. It deals correctly even with highly non-uniform mappings, for example where points on the domain map to infinity on the range, employing an incremental process starting at the origin of the domain and repeatedly adding new triangles to the perimeter of a growing region.

Ulrich Pinkall's geometrical graphics group at the Technical University of Berlin has chosen the software product AVS as its primary visualization tool. This system allows the researcher to chain together independent modules into complex computational networks. With a family of standard and user-generated modules, this group conducts research on the visualization of optimal surfaces, such as H-surfaces (surfaces of constant mean curvature), and on discrete dynamical models for quantum systems. This work has led to new results such as the discovery of the simplest soliton.

A group at the University of Bonn led by Konrad Polthier has created its own visualization environment, known as GRAPE, for their research into minimal surfaces and related differential systems. GRAPE (GRAphical Programming Environment) reflects an object-oriented approach that encourages users to create specific geometric objects, and includes features that expedite the creation of animations.

3.3 Homotopies and the Sphere Eversion Problem

There is a mathematical phenomenon, the *homotopy*, that lends itself particularly well to real-time interactive computer animation. Mathematically speaking, the notion of a homotopy spans a continuum of sophistication. At one end are the familiar, rigid Euclidean motions of translation, rotation and reflection; at the other are exotic metamorphoses of surfaces, such as *sphere eversions*, whose complexity resists holistic comprehension, and thus challenges computer graphics in a unique way. Simply put, one wishes to interact with the temporally extended homotopy as easily as with rigid objects.

Current hardware and graphics libraries deal well with objects in 3-space that do not change their shape during a rigid motion. Animating mild deformations that alter object shape without losing recognizable identity requires ingenuity and good technique, but is not intrinsically difficult. Non-linear interpolation between two given forms, such as *morphing*, is a familiar example of a less trivial homotopy that does generate animation problems. A topologist's (regular) homotopy, however, tends to be much more complicated than morphing. Turning a sphere inside out without tearing or excessively creasing its virtual fabric (*everting* it) is the paradigm example of such a homotopy. If a rendered teapot is the classical subject of computer graphics, sphere eversion is the “teapot” of visualizable geometry.

During an *eversion* the surface must be permitted to pass through itself. If either of the two constraints of continuity and regularity on a regular homotopy is relaxed, then eversion becomes trivial mathematically, though a graphical depiction may remain difficult. When both constraints are enforced, the problem has remained a challenge into this, the fourth decade since Smale proved the existence of an eversion. The collection of explicit examples has grown steadily over the years, and we discuss those that are most relevant to the present paper below.

In the early seventies, Nelson Max digitized Charles Pugh’s wire mesh models of the stages in Bernard Morin’s sphere eversion. Central to this eversion is an immersion of the sphere with symmetric but very complicated self-intersections. The homotopy simplifies this in stages until an embedded sphere is reached. There are two ways of proceeding that differ by an easily programmed symmetry. Reversing the one and following the other everts the sphere. With the technology of the time, Max could interact in real-time only with animated wire-frames of the homotopy, so that his film with fully rendered surfaces (see Sidebar B), had to be generated painstakingly frame-by-frame.

Mathematicians as well as computer animators require analytic expressions that parametrize homotopies. The former are obliged to mistrust purely qualitative depictions on logical grounds, while the latter find analytic representations far preferable to huge hand-generated data bases. Morin devised the first parametrizations of his eversion in the late seventies.

The power to manipulate a homotopy in real-time using a mouse did not appear until the eighties, when John Hughes used a Stardent graphics computer to realize an interactive parametrization of Morin’s eversion. Like Max, he began with polyhedral models, but ones with very few vertices. Using techniques from Fourier analysis, he converted these first to power series in the frequency domain, and then mathematically manipulated the results so that their inverse transforms produced a fast and beautifully smooth eversion, a frame of which is shown in Figure 13.

More recently, François Apéry realized the Morin-Denner eversion as an illiView interactive animation, pictured in Figure 14. This polyhedral homotopy, influenced by a polyhedral Möbius band of Ulrich Brehm (who also inspired the trefoil knotbox in Figure 3), has the minimum number of vertices theoretically possible. It thus also solves an optimization problem. With the help of an illiView team, Apéry was also able to use an experimental smooth parametrization to accomplish the Morin-Apéry homotopically minimal sphere eversion.

A truly new sphere eversion based on an idea of William Thurston is the focus of the Geometry Center video *Outside In*, discussed in Sidebar B, and illustrated in Figure 15. From a mathematical viewpoint, the parametrization of this homotopy comes closest to Smale’s original concept. The basic idea is that for any eversion there is another homotopy in an associated, higher dimensional manifold, which shadows it in an imperfect way. The equations for this *doppelgänger* are easy to find. Thurston solved the problem of producing an actual eversion from the higher dimensional “shadow” homotopy.

4 Design Philosophies for Geometry Visualization

The selection of visualization systems discussed in this paper represents a multitude of design philosophies. Due to time and resource limitations, the developers of these systems have settled on different trade-offs among a host of issues: ease of implementation, generality, domain-specialization, extensibility, exploitation of architecture-specific features, inter-architecture portability, interactivity of the user interface, simplicity of the programmer's interface, code customizability, integration with other programs, and Internet distributability. We shall discuss the choice of emphasis for several software systems whose details the authors know well.

The philosophy of the Brown University mathematical visualization group is to create online "interactive books" which incorporate interactive graphics demonstrations into a hypertext system for teaching calculus and differential geometry. The *fnord* system, developed at and distributed by Brown, is used for the graphics. The commercial *dynatext* hypertext system is the basis for the interactive books, which have been integrated into the mathematics curriculum at Brown and are spreading to other universities. Besides reading text, the student can click on words to get definitions, follow links to related subjects, and interact with a multitude of 3D visualizations whose parameters can be manipulated to help understand a wide array of concepts. A typical example permits the user to "fly" on a curve in 3-space, keeping track of all the differential geometry (e.g., curvature and torsion) continuously as the curve is traversed. The "interactive book" approach of situating interactive graphics within a written context is an appealing educational paradigm.

Geomview [9] and the Surface Evolver [2] can be widely distributed because of their generality, extensibility, and portability; they were designed to accommodate user-defined tasks not built into the original system. Both reflect considerable effort devoted to making them available to broad community of users; Evolver will run on any system with a C compiler, while Geomview was until recently limited to SGI workstations and systems running NeXTStep. A limited X-windows version that runs on a wider variety of workstations has recently become available.

Geomview has a multilevel interface including mouse-driven 3D interaction, control panels, and an interpreted command language. Its programming interface is built on an object-oriented library, but the command language can be used for high-level run-time communication with external modules written in any language. While this text-based communication approach is flexible, it does not suffice for applications that require massive data transfer or highly interactive custom mouse manipulation.

While the Surface Evolver can produce graphics in a number of formats, it uses Geomview as its preferred graphics output server and concentrates mainly on computation. The Evolver text interface permits flexible specification of user data as well as interactive user control of a large number of parameter settings that can be changed while monitoring the output. The system already supports a wide variety of application-specific energy programs, and custom code may be added by the user as well.

Like the Evolver and Geomview, GRAPE from the University of Bonn features a machine-independent interface for graphics that runs on a wide variety of workstations. While not in the public domain, it is non-commercial and the developers will distribute libraries to scientific sites. GRAPE is an object-oriented programming environment for developing applications, not an application itself.

The University of Massachusetts MESH system has focused on specific research at GANG rather than wide distribution. MESH features a mouse-based direct manipulation philosophy; but instead of hardwiring the mouse functionality or providing a control panel approach where a mouse mode is chosen from a relatively small set of fixed possibilities, MESH emphasizes dynamic mouse binding. Hierarchical popup menus are used to set up a mapping from mouse motion to parameter change on the fly. Mouse motion can then be used to easily vary any of an extremely wide variety of parameters.

The illiView collection of RTICAs (real-time interactive computer animators) for SGI workstations exemplifies another interactive design philosophy. Users of illiView are expected to understand, modify and, on occasion, rewrite from scratch the code for their particular RTICA. A typical RTICA is a single C program that uses SGI's GL-library to bring mathematical phenomena to life. Mouse motions and keys control all conceivable parameters in tandem, requiring substantial dexterity on the part of the illiViewer. To avoid breaking visual concentration, illiView avoids popup menus and control panels. Numerical and the status information can be incorporated into a heads-up display in the single window.

The Indiana University 4D lighting software system [4] illustrates another contrasting approach. Here a particular piece of high performance hardware (the SGI Reality Engine) is chosen and is exploited to achieve unusual effects for unique applications. To simulate a world of surfaces lit by 4D light, for example, the hardware's unique texture-mapping capabilities are used to compute in real-time a texture-map representing the current reflection map of the 4D light; the intensities are fixed to the 3D geometry because the 3D space is the "film" on which 4D images are projected.

The Fourphront system, which relies on the custom-built testbed Pixel-Planes 5 and multiple 3D joysticks, is even further along the hardware-specific continuum than the previous example. The SGI Reality Engine, while an extremely high-end machine, is nevertheless a commercial product. Pixel-Planes 5, in contrast, is the unique result of the research program in high-speed parallel rendering hardware at UNC-Chapel Hill; it has been used by UNC software developers for experiments with exotic input and output devices and related virtual reality applications. Fourphront has sought to extend the envelope of viable user interaction techniques, sidestepping the restrictions of commercially available hardware.

Mathematical Virtual Reality. Emphasis on speed and the no-control-panel, direct manipulation philosophy characterizing the last several systems becomes very important when trying to produce workable mathematical visualizations in virtual reality environments. Demonstrations from both the Illinois and Indiana groups now run in the duplicatable room-size CAVE environment pioneered by the Electronic Visualization Laboratory of the University of Illinois at Chicago. If it is possible to induce new mathematical perceptions by immersing the user in intuition-building interactive worlds of this sort, then these developments are potentially of great importance.

5 Prospects for the Future

Mathematical visualization, like other areas in the rapidly developing field of visualization science, is still defining itself. We have presented examples of interactive workstation systems for visualizing pure geometry as well as noting the importance of these systems for developing precomputed animations (see Sidebar B). Many of these techniques can be adapted to the emerging virtual reality medium as interactive performance continues to improve.

Furthermore, there are clearly many areas of development that are appropriate for the participation of computer scientists with skills in interactive interface design, computer graphics, efficient algorithms, and perhaps data management. One can also conclude from looking at the images we have presented that graphic arts and design skills also have a unique role to play in improving the quality of graphical communication. Yet clearly the active participation of the end-users, the mathematicians themselves, is just as critical as in other visualization problems, and possibly more so, since the subject material is so complex that research into visualization merges rapidly with the mathematics itself. Our purpose here has been to provide a look at some major developments and trends in this fascinating field, and to try to build a bridge between the computer science and mathematics cultures by summarizing problems of mutual interest, hopefully with the result of generating more activity in this style of research in both communities.

Acknowledgments

This work was supported in part by NSF grant IRI-91-06389 (Indiana University) and NSF grant DMS-89-20161 (The Geometry Center: The National Science and Technology Center for the Computation and Visualization of Geometric Structures, at the University of Minnesota.). The authors gratefully acknowledge the following people for contributing synopses of their work that were integrated into various sections of the paper, and, most importantly, for sharing with us many of the beautiful images that they have generated in the course of their own research: Fred Almgren, Thomas Banchoff, David Banks, Kenneth Brakke, Robert Cross, Celeste Fowler, Charlie Gunn, Pheng Heng, David Hoffman, Olaf Holt, John Hughes, Hui Ma, Mark Phillips, Dennis Roseman, and Jean Taylor.

References

[1] David Banks, "Interactive Display and Manipulation of Two-Dimensional Surfaces in Four Dimensional Space," in 1992 *Symposium on Interactive 3D Graphics*, pp. 197–207 (ACM, 1992).

[2] Kenneth A. Brakke, "The Evolver System," Experimental Mathematics **1**, No. 2 (1992). Available by anonymous ftp from `geom.umn.edu`.

[3] Charlie Gunn, "Discrete Groups and Visualization of Three-Dimensional Manifolds," in Proceedings of SIGGRAPH 93 (Anaheim, CA, August 1–6, 1993), pp. 255–262. In *Computer Graphics* Proceedings, Annual Conference Series (ACM SIGGRAPH, New York, 1993).

[4] A.J. Hanson and R.A. Cross, "Interactive Visualization Methods for Four Dimensions," in Proceedings of *Visualization '93*, San Jose, CA, Oct 25–29, 1993, pp. 196–203 (IEEE Computer Society Press, Los Alamitos, CA, 1993).

[5] A.J. Hanson and P.A. Heng, "Illuminating the Fourth Dimension," Comp. Graphics and Appl., **12**, No. 4, pp. 54–62, (IEEE Computer Society Press, Los Alamitos, CA, 1992).

[6] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination*, Chelsea, New York, 1952.

[7] M. Callahan, D. Hoffman, and J. Hoffman, "Computer graphics tools for the study of minimal surfaces," Comm. ACM **31**, pp. 641–661 (1988); D. Hoffman, F. Wei, and H. Karcher, "Adding handles to the helicoid," Bull. of the Amer. Math. Soc. **29**, No. 1, pp. 77–84 (July, 1993).

[8] N. Max and W. Clifford, "Computer animation of the sphere eversion," Computer Graphics, Vol. 9, No. 1, pp. 32-39 (ACM, 1975).

[9] Mark Phillips, Silvio Levy, and Tamara Munzner, "Geomview: An Interactive Geometry Viewer," in the Computers and Mathematics column of the *Notices of the Amer. Math. Soc.* **40**, No. 8, pp.985–988 (October 1993). Available by anonymous ftp from `geom.umn.edu`.

[10] Mark Phillips and Charlie Gunn, "Visualizing Hyperbolic Space: Unusual Uses of 4×4 Matrices," in 1992 *Symposium on Interactive 3D Graphics*, pp. 209–214 (ACM, 1992).

[11] Ulrich Pinkall and Konrad Polthier, "Computing Discrete Minimal Surfaces and Their Conjugates," Experimental Mathematics **2**, No. 1 (1993).

Glossary

1. **Manifold**. A generalization of N-dimensional space in which a neighborhood of each point, called its *chart*, looks like Euclidean space. The charts are related to each other by Cartesian coordinate transformations and comprise an *atlas* for the manifold. The atlas may be non-trivially connected; there are round-trip tours of a manifold that cannot be contracted to a point. The surface of a donut, called a *torus*, is a familiar non-trivial 2D manifold.

2. **Submanifold, ambient space**. A submanifold is a subset of a manifold, its ambient space, for which each point has a chart in which the submanifold looks like a linear subspace of lower dimension. A common knot is a 1-dimensional submanifold of its 3-dimensional ambient space.

3. **Homotopy**. A continuous deformation of a mathematical object which preserves its topological integrity but may develop self-intersections and even worse singularities. There is a homotopy that takes a teapot to a torus (a sphere with a hole). There is another deforming it to a point.

4. **Isotopy**. A homotopy of an object produced by a deformation of the ambient space, so therefore the object cannot develop new self-intersections. The deformation of the teapot to a torus is an isotopy, but the deformation to a point is not.

5. **Embedding.** The parametrization of a submanifold by means of a standard model. A knotted sphere in 4-space is an embedding of the familiar round sphere. Whitney's theorem says that an N-dimensional manifold is guaranteed to have an embedding in Euclidean $2N$-space.

6. **Immersion.** A locally (but not globally) smoothly invertible mapping of one manifold into another. The image may have self-intersections; the figure-8 is an immersion of the circle in 2D.

7. **Minimal Surface**. A surface that locally has the smallest area given a particular topological shape for it, and possibly, constrained by a fixed boundary (soap-films) or prescribed behavior at infinity.

8. **Steepest Descent Method**. A particular way of guiding an isotopy of an embedded surface to one which minimizes a function that measures its shape. Moving down the gradient of the area function often terminates at a minimal surface.

A Background for Further Reading

The exploitation of pictorial representations in mathematical problems is attracting new interest, as described in the main body of the paper. The reader who is interested in more general literature exhibiting the development of this subject area over the last 15 years is invited to consult other books and collections on the subject. Among these, the authors particularly recommend the following:

1. T.F. Banchoff, *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions*, Scientific American Library, New York, 1990. This is general book, accessible to people with a moderate level of mathematical interest; the graphics are excellent, and the exposition very readable.

2. D.W. Brisson, Ed., *Hypergraphics: Visualizing Complex Relationships in Art, Science and Technology*, AAAS Selected Symposium **24**, Westview Press, 1978. This collection contains many of the early seeds of the current work in visualizing geometry; this contains a considerable amount of mathematics as well as graphics.

3. Gerd Fischer, *Mathematische Modelle/ Mathematical Models*, Vols. I and II, Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 1986. This book includes an exhaustive survey of classical models of mathematical shapes. It is worth noting that perhaps the most significant change in capability enabled by computer graphics is the new ability to *animate* models such as those in Fischer's book in response to a user's actions.

4. G.K. Francis, *A Topological Picturebook*, Springer-Verlag, New York, 1987. This book is primarily a mathematical survey that phrases its material in terms of "descriptive topology" with the goal of resurrecting our nineteenth century fascination with mathematical pictures.

5. J.R. Weeks, *The Shape of Space*, Marcel Dekker, New York, 1985. Weeks' brief book is a gem of clarity and mathematical insight, and yet is sufficiently complete that it has been used as the basis for courses on topology for secondary school teachers. Weeks also has developed an advanced computer program *SnapPea*, for creating and studying hyperbolic 3-manifolds, available by anonymous ftp from `geom.umn.edu`.

Finally, we remind the reader of the classic *Geometry and the Imagination*, by Hilbert and Cohn-Vossen [6], which has served to inspire generations of professional and amateur mathematicians.

B Mathematical Videos

The emergence of computer graphics has sparked a renewed interest in visual mathematics in the form of films and videotapes. Many of these efforts have made use of interactive graphics.

For the past 15 years, the yearly focal point of the computer graphics world has been Siggraph, the annual computer graphics meeting of the Association for Computing Machinery. The new, refereed, computer animations shown in both the Animation Screening Room and in the evening Electronic Theater, stringently refereed technical paper and panels sessions, and huge exhibit floor now draw in excess of 30,000 people. Many of the animations mentioned below have been shown at Siggraph, which showcases animations from the science, art, and entertainment communities.

Nelson Max was one of the pioneers of mathematical visualization movies with his groundbreaking Topology Films project in the early 1970's, when computer graphics was first used to make mathematical movies despite the primitive state of hardware and software. His films on curves, *Regular Homotopies in the Plane, Parts 1 and 2*, were made using a computer-controlled oscilloscope to plot individual points and multiple exposures to create the image of a dynamically moving curve. His classic film *Turning A Sphere Inside Out* [8] captures Bernard Morin's concrete description of Smale's surprising but abstract theorem that a sphere can be turned inside out ("everted") in 3-space (see 3.3). Max created this epic of mathematical visualization on many computers at the nodes of the far-flung ARPANET. Later, he and Banchoff used their common visual insight to prove that every sphere eversion has a quadruple point.

Other early efforts in this area that received wide attention were Thomas Banchoff's interactive, real-time geometrical visualization studio at Brown University. In the late 1970's and early 1980's, he and his associates produced computer animated films of 4-dimensional objects such as the award winning "Hypercube," "The Veronese Surface," and wireframe versions of his recent video animation, "The Hypersphere: Foliations and Projections." Several of these animations were shown at early Siggraph conferences.

At Indiana University, a scientific visualization effort led by Andrew Hanson has focused on finding new ways to represent and visualize Riemann surfaces, on rendering techniques using 4D light, and on the development of corresponding interactive methods (see 3.1); this group has produced a variety of short animations, including three shown at Siggraph, "Visualizing Fermat's Last Theorem," "FourSight," and "knot^4," and three others at IEEE Visualization conferences.

George Francis of the University of Illinois at Urbana-Champaign has worked on a number of mathematical animations. Francis, Donna Cox and Ray Idaszak of the National Center for Supercomputing Applications created the animation *The Etruscan Venus*, shown at Siggraph 88. A variety of other short video productions have been created by Francis and illiView teams as well. Francis recently teamed with Louis Kauffman of the University of Illinois at Chicago Mathematics Department and Dan Sandin of The Electronic Visualization Laboratory at the University of Illinois at Chicago to produce *Air on the Dirac Strings*, shown at Siggraph 93.

Finally, the Geometry Center has produced a wide range of animations. At one end of the spectrum are short videos intended for use in lieu of a computer demonstration in a lecture. In the middle of the spectrum are longer animations by individuals and small groups. Examples of such productions are *Twisting and Turning in Four Dimensions* by Dennis Roseman, who has made several animations on 3D and 4D knots shown at mathematical conferences, and *Computing Soap Films and Crystals*, distributed by the American Mathematical Society, produced by the "Minimal Surfaces Team" of pure and applied mathematicians led by Jean Taylor of Rutgers and Fred Almgren of Princeton. At the other extreme are high-quality productions aimed at the general public, which require years of effort. These include *Not Knot,* a guided tour of hyperbolic geometry (see 3.1) and knot theory directed by Charlie Gunn and Delle Maxwell (see Figure 8), and *Outside In*, which focuses on the sphere eversion problem (see 3.3 and Figure 15), directed by Silvio Levy, Delle Maxwell, and Tamara Munzner.

To sum up, there is today a lively activity in the area of producing mathematically oriented computer graphics animations that are shown at computer graphics conferences and mathematics seminars, and to the general public.

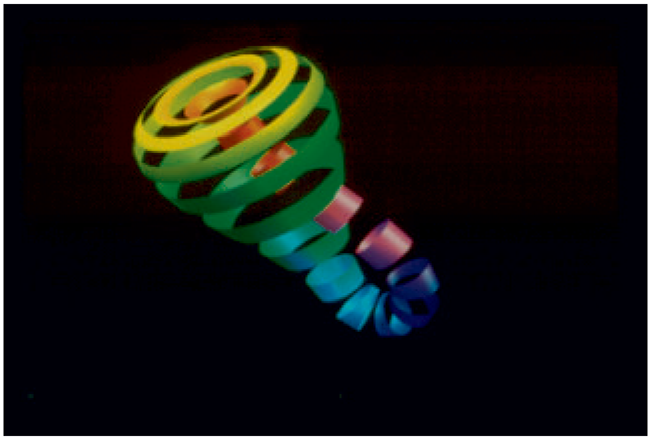

Figure 1: Image of a Klein bottle with color coded 4D depth and ribbon slicing to reveal interior structure. (T. Banchoff and N. Thompson, Brown University.)

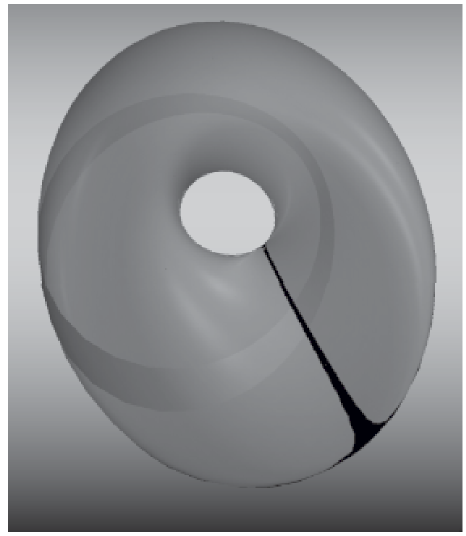

Figure 2: This semi-transparent Klein bottle displayed by Fourphront contains a Möbius band (opaque). The projection of the surface to 3D contains an intersection line, highlighted in black. (D. Banks, University of North Carolina and Langley Research Center.)

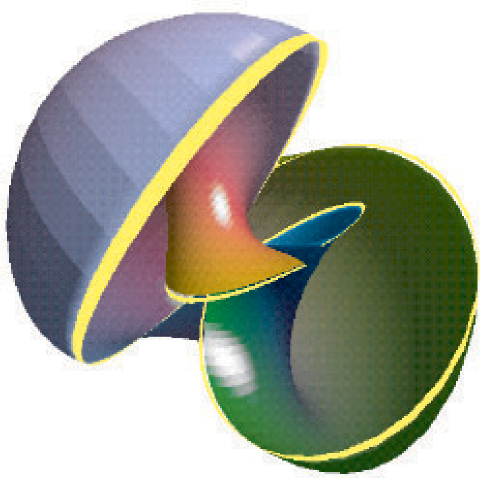

Figure 3: A snapshot from an interactive sequence showing the deformation of a Möbius band with 3 half-twists into Ulrich Brehm's *trefoil knotbox*. (G. Chappell, G. Francis and C. Hartman, University of Illinois at Urbana-Champaign.)

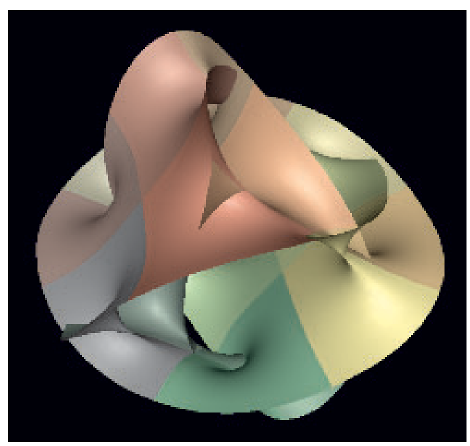

Figure 4: The $n = 4$ Fermat surface projected to 3D using the MeshView interactive 4D viewer. The colors encode the relative complex phase of different patches of the surface. (H. Ma and A. Hanson, Indiana University.)

Figure 5: Left: An application of slicing, projection, and color-coding to exhibit the properties of a knotted sphere in 4D space. Right: A cutaway view of a knotted sphere with 4D-depth-dependent texture map density. (D. Roseman, University of Iowa.)

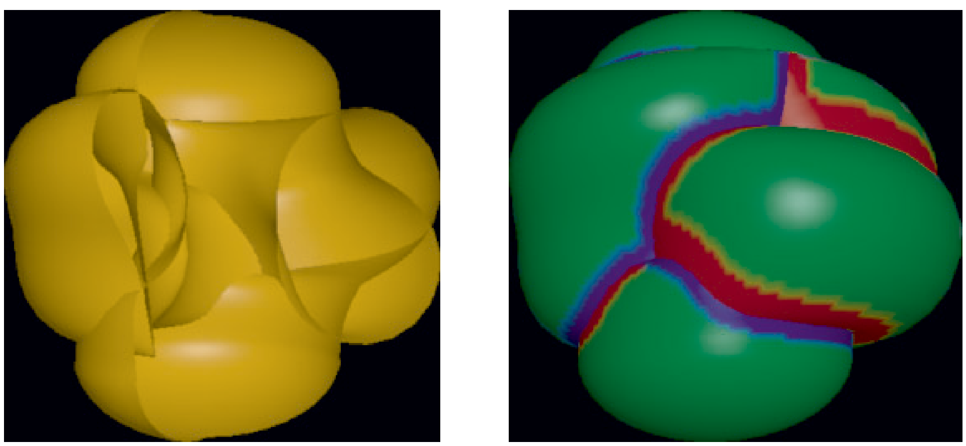

Figure 6: Illustration of cutaway and above-below crossing markings for the twist-spun trefoil, a surface embedded 4D that, surprisingly, is actually not knotted. (Frames from the video animation "knot^4," Indiana University.)

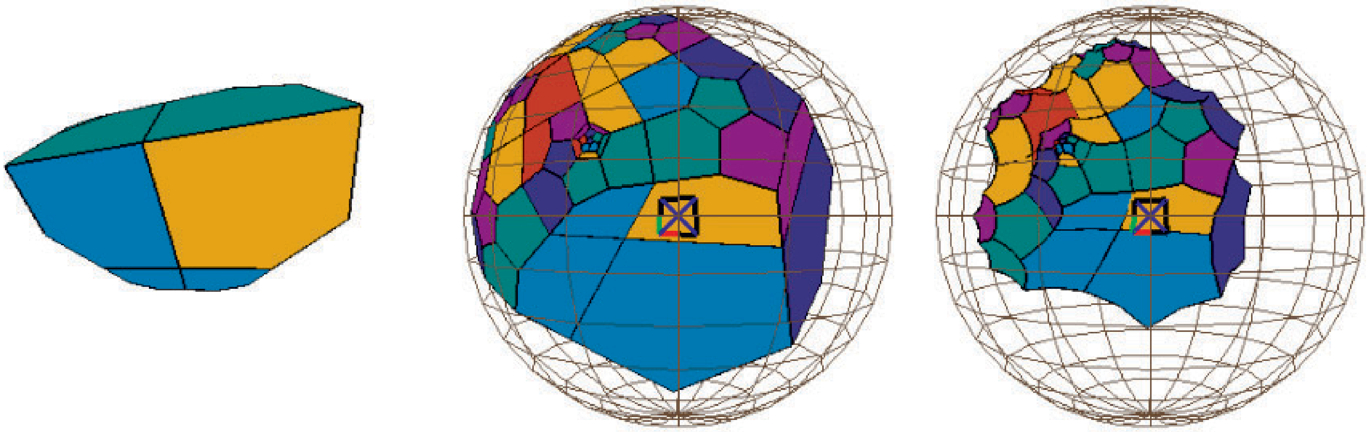

Figure 7: A partial tessellation of regular right-angled dodecahedra in the three built-in Geomview models (virtual, projective, and conformal, respectively) of hyperbolic 3-space. (Geometry Center.)

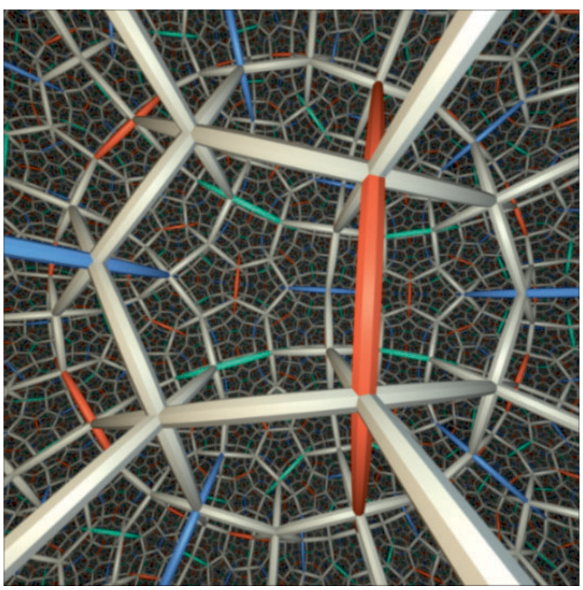

Figure 8: A view of the tessellation of hyperbolic space by regular right-angled dodecahedra. (Frame from the movie “Not Knot”; C. Gunn, Geometry Center.)

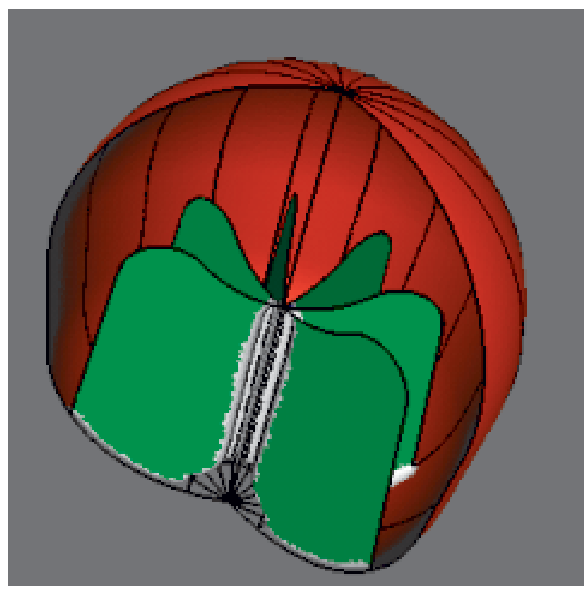

Figure 9: Evolver/Geomview display of spacecraft fuel tank; the fuel surface tension computed with the Evolver is the only force available to guide fuel to the intake valve in low gravity. (K. Brakke, Susquehanna University.)

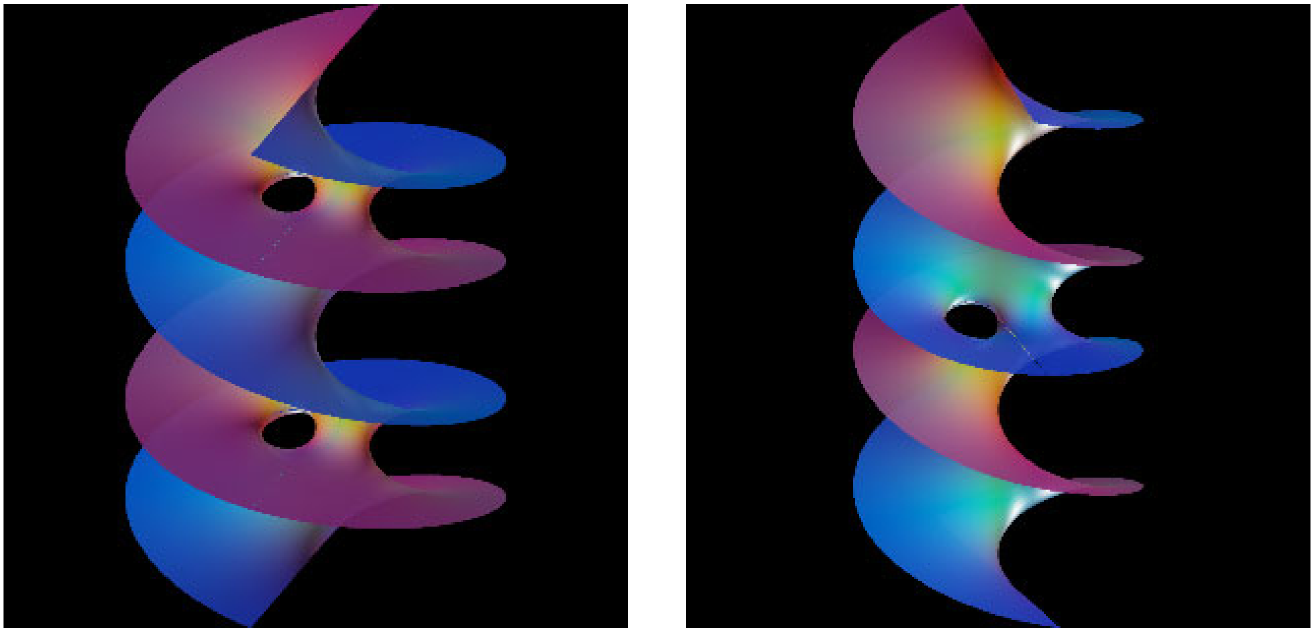

Figure 10: The minimal surfaces $\mathcal{W}_1$, periodic with quotient genus one, and $\mathcal{H}e_1$, which is nonperiodic of genus one. These surfaces were discovered with the aid of computer graphics techniques. (D. Hoffman, University of Massachusetts, Amherst.)

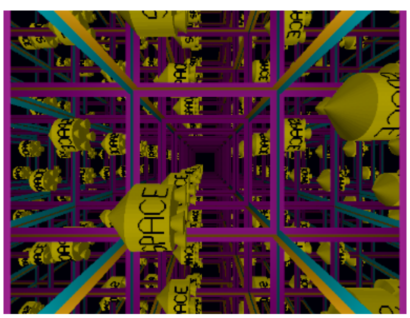

Figure 11: Using Maniview, an external module of Geomview, the user can fly around in the insider's view of the 3D Klein bottle. Note that spaceships in alternating columns are mirror reversed. (C. Fowler and C. Gunn, Geometry Center.)

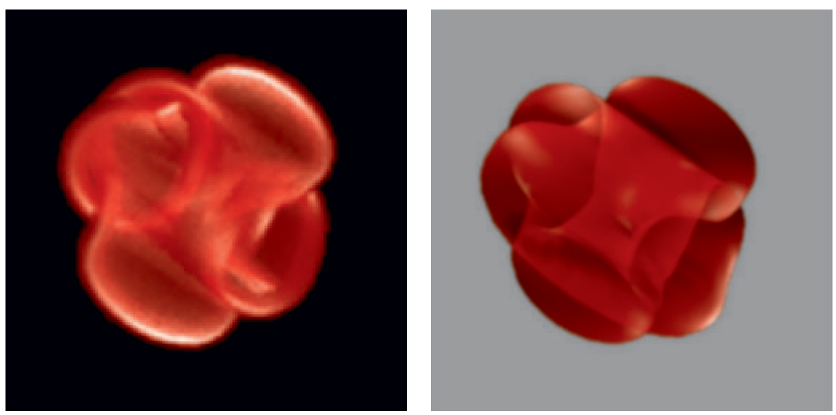

Figure 12: Images of 4D-lit knotted spheres including depiction of 4D occlusion in the 3D projection; this is similar to the self-occlusions of a knotted rope in 3D projected to a 2D image. The left image is a time-consuming volume rendering, while its equivalent on the right can be manipulated using real-time methods. (R. Cross, A. Hanson, and P. Heng, Indiana University.)

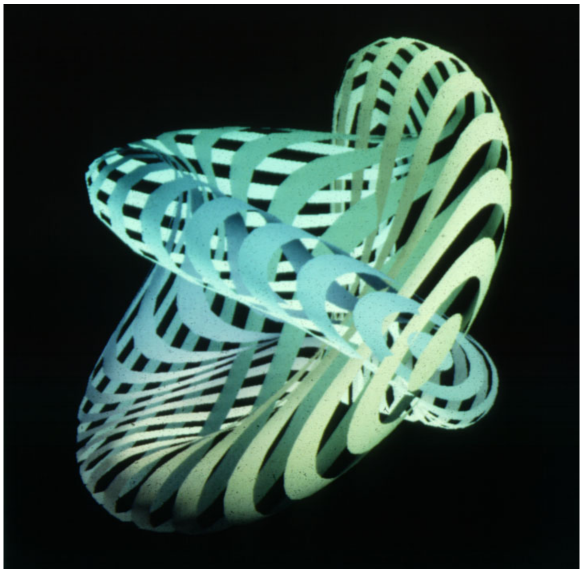

Figure 13: A frame from John Hughes' real-time interactive animation of Bernard Morin's eversion of the sphere realized on a Stardent graphics computer. (J. Hughes, Brown University.)

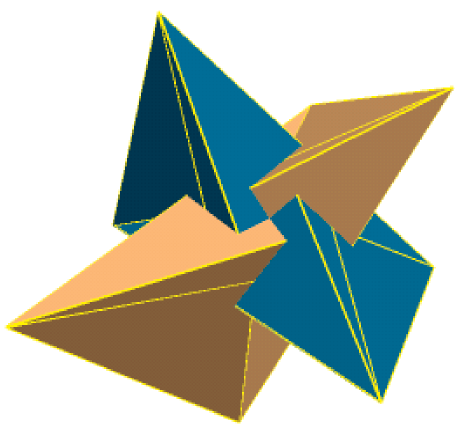

Figure 14: This image, the *cuboctahedral sphere eversion*, is a piece-wise linear realization of this classical sphere eversion homotopy due to Morin, Denner and Apéry. (F. Apéry, G. Francis, C. Hartman and G. Chappell, University of Illinois at Urbana-Champaign.)

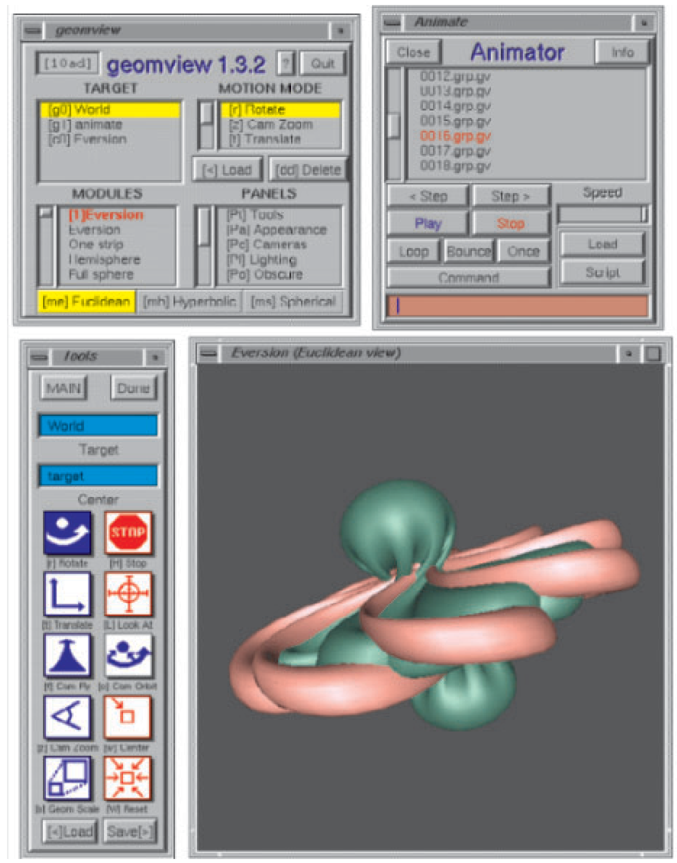

Figure 15: A frame from the film in progress "Outside In," a sphere halfway through the Thurston eversion. In this example of a typical Geomview application, the Animator external module is being used to control a "flipbook" of the animation. The main "geomview" control panel at the upper left controls the viewing state, invokes modules, and brings up other control panels for control of lighting, object appearance, and so on. The basic mouse driven motion controls for changing the user's view are in the "Tools" panel on the left. (Geometry Center.)