An Exploratory Approach to Mathematical Visualization

Daryl Hepting
`dhepting@cs.sfu.ca`

Weiming Cao
`wcao@math.sfu.ca`

Robert D. Russell
`rdr@math.sfu.ca`

Simon Fraser University
8888 University Drive
Burnaby, British Columbia
Canada V5A 1S6

Abstract

Hanson *et al.* [4] described mathematical visualization as “the art of creating concrete understanding from abstract mathematical objects and their transformations.” Such artistry may be difficult to achieve because it demands both mathematical and graphical knowledge in one person, or excellent communication between members of a group of individuals with specific talents. Furthermore, it may also require the ability to exceed the limitations of one’s own experience. This paper describes a computer-aided visualization system which aids users in finding meaningful imagery amidst the range of alternatives.

Introduction

Consider that any images produced by a visualization process are visual representations which illustrate some underlying concept or data. Any particular visual representation is a combination of *components* from several *categories* which may (depending on the task) include camera, lighting, material, and geometry. Selection of a particular visual representation involves a selection of a component from each category. As a combination, each visual representation is a point in the space of all possible visual representations. When there are many categories each containing many components, the space becomes huge.

Computers have proven to be a great aid to the creative human endeavour of visualization. Computer-generated images have provided access to a variety of data which may have remained impenetrable by pre-graphical techniques. Considerable visualization research has focused on developing new components and categories for particular applications. Such a focus may neglect effective new visual representations which may be found by choosing new combinations within an already defined space.

The effectiveness of any image for any particular user will depend on the interpretation which the user gives to the image. A user is likely to find more meaning if they have been actively involved in creating that imagery: “doing” is more powerful than “watching”.

More and more research now deals with the question of how to productively involve the user in the visualization process. In particular, this paper describes an approach which allows users to search the space of available alternatives by iteratively selecting from examples and refining the results until a meaningful visual representation is found. After a review of previous work in this area, we describe the new system and give examples of how it can be used effectively in the context of mathematical visualization.

Motivation

This work has evolved from a somewhat traditional model of the interaction between researchers in mathematics (providing the problem) and those in visualization (providing the images). Key limitations of this model are highlighted by the following questions:

- *how do I realize the image I have in mind?* Mathematicians may lack the means to articulate their ideas in forms which can be directly translated to images. The necessary translation is usually done through an iterative process of describing, generating, and evaluating images.
- *given this image, what other alternatives do I have?* A particular visual representation which is effective in one situation may not be so effective in others. Without a knowledge of the available alternatives, it may not be easy to find new visual representations for the problem.

Such difficulties present themselves in the more general context of human-computer interaction. Norman [9] viewed this interaction in terms of cycles of action and evaluation. He identified the gulf of execution between the user's intentions and the system's commands; and the gulf of evaluation between the system's representation and the user's expectations.

A Presentation Tool (APT) [7] is one of the first systems intended to provide computer support for the development of visual representations. It used an artificial intelligence approach to shield the viewer from the immensity of the space of possible visual representations. Although this approach serves an important role, its effectiveness is limited because it removes humans from the essential task of interpretation.

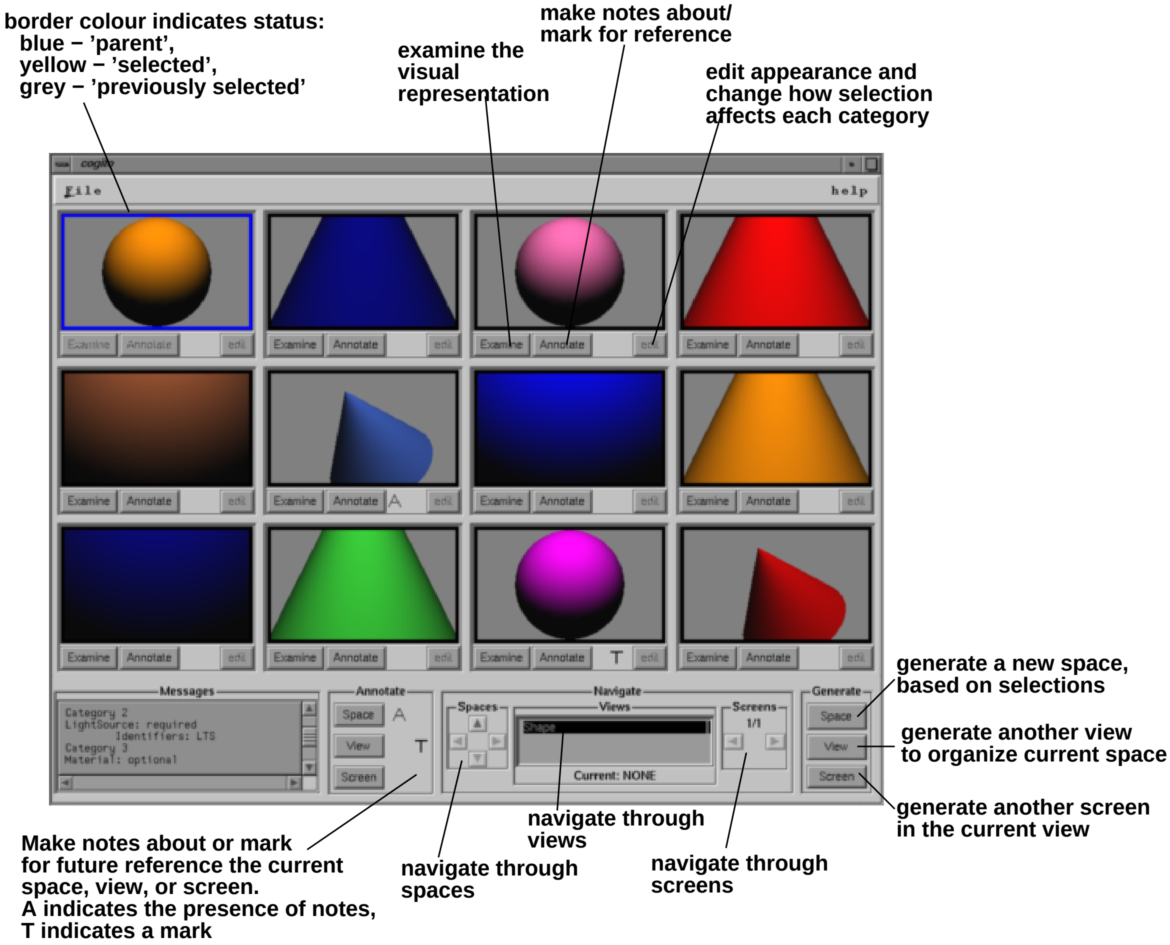
SageTools [12] seeks to provide advice to designers through static examples of previously successful graphics, and some facility for reuse of those graphics. The CCAD [6] is more powerful in its capabilities for exploring alternative representations, though it requires at least a partial description of the desired output. Design Galleries [8] is still more powerful in its capabilities for exploring the parameter space of very well-defined problems. Each of these systems relies to some extent on a more or less complete description of the problem solution, which may serve to limit the range of alternatives which are considered.

Researchers have observed that, especially in design, problems and solutions coevolve [3]. One usually does not, perhaps cannot, have a clear statement of the problem without an idea of the solution.

Rittel [11] described the difficulty in the following way: "you cannot understand the problem without having a concept of the solution in mind; and that you cannot gather information meaningfully unless you have understood the problem but that you cannot understand the problem without information about it." The advice is to begin, regardless of whether one is actually at the beginning.

AVS (Application Visualization System) [15] allows users to create complete visualizations from components, using a dataflow model to connect them. The SIV (Spreadsheet for Visualization) [2] adopts a familiar analogy to the task of visualization, and programming is done within the spreadsheet model. These systems require the user to program, even in very simple terms. The space of alternatives available for exploration in these schemes is implicitly limited by the user's own experience.

Sims' artificial evolution approach gives a more explicit sense of the range of available alternatives, though it does not provide much of the fine control that would allow particular regions to be searched more thoroughly. It engages users in an iterative process of evaluating a variety of sample images and selecting those which are aesthetically appealing. The user can direct the development of the images without concern for the means used to create them.



```
Category 2
LightSource: required
       Identifiers: LTS
Category 3
Material: optional
```

border colour indicates status:
blue – 'parent',
yellow – 'selected',
grey – 'previously selected'

examine the visual representation

make notes about/ mark for reference

edit appearance and change how selection affects each category

generate a new space, based on selections

generate another view to organize current space

generate another screen in the current view

Make notes about or mark for future reference the current space, view, or screen.
A indicates the presence of notes,
T indicates a mark

navigate through views

navigate through spaces

navigate through screens

Figure 1: An annotated image of the cogito display, working on a sample application

Description

Sims' application of artificial evolution answers the concern about execution in an appealing way. In visualization applications, aesthetics is not the most important concern when examining images. However, because each individual may find a different interpretation for an image, support for the use of subjective criteria is essential. Therefore, this notion of artificial evolution with a direct manipulation interface has been used as the basis for a new system, called cogito.

To find a meaningful visual representation, the space of alternatives must be searched. Even with relatively few components and categories, an exhaustive search becomes impractical. To organize a search of this space, adoption of a problem-solving approach can be helpful. Following Campbell's evolutionary epistemology [1], problem-solving is often described as an iterative process comprising steps of combination, selection, and retention. Such a process is made accessible through a flexible visual interface, the core of which is inspired by Sims' system for artificial evolution in computer graphics [13]. The interface (see Figure) displays a subset of available representations (sampled according to the selected organization of the search space), generated from the current data, with which the user can interact.

We reject Mackinlay's use of artificial intelligence as protection for the user because we regard the user as someone whose intellect can manage the enormity of the space of all available alternative visual representations and whose input is essential with regards to the subjective nature of the evaluations needed.

A large space need not cause problems if it can be structured in a meaningful way: the phone book for a large urban centre is an example. What constitutes a meaningful structure may not be the same for all users, who may perceive the space to be either clue-rich (“homing”) clue-poor (“Klondike”) [10]. Giving the user the ability to place different structures on the same space increases the chances that the search will conclude with a positive outcome.

Cogito aids creativity in the problem-solving process by making available to the user the entire space of combinations possible from the components provided. It also provides, through views, the means to structure and examine the space according to a range of criteria. The user sees the current space, with the current structural view, one screen at a time. Cells, which display individual visual representations and permit certain operations on them, comprise each screen.

The approach espoused by cogito encourages invention and discovery by facilitating the combination of ideas, namely components. The user indicates desirable components or complete visual representations by selection (done by clicking directly on the desired cell). Once the user is satisfied with the selections made on a particular space, a new space consistent with those selections is generated by a genetic approach which performs crossover operations amongst selected combinations. Successive generations can be used to either narrow or expand the search space, depending on the needs of the user. In this way, the space of all available visual representations can be effectively navigated to achieve meaningful results.

The possibility that the selected alternatives will come from combinations generated outside the user’s experience is a very powerful aspect of this approach. It is therefore much more likely that fruitful combinations will be explored using this system.

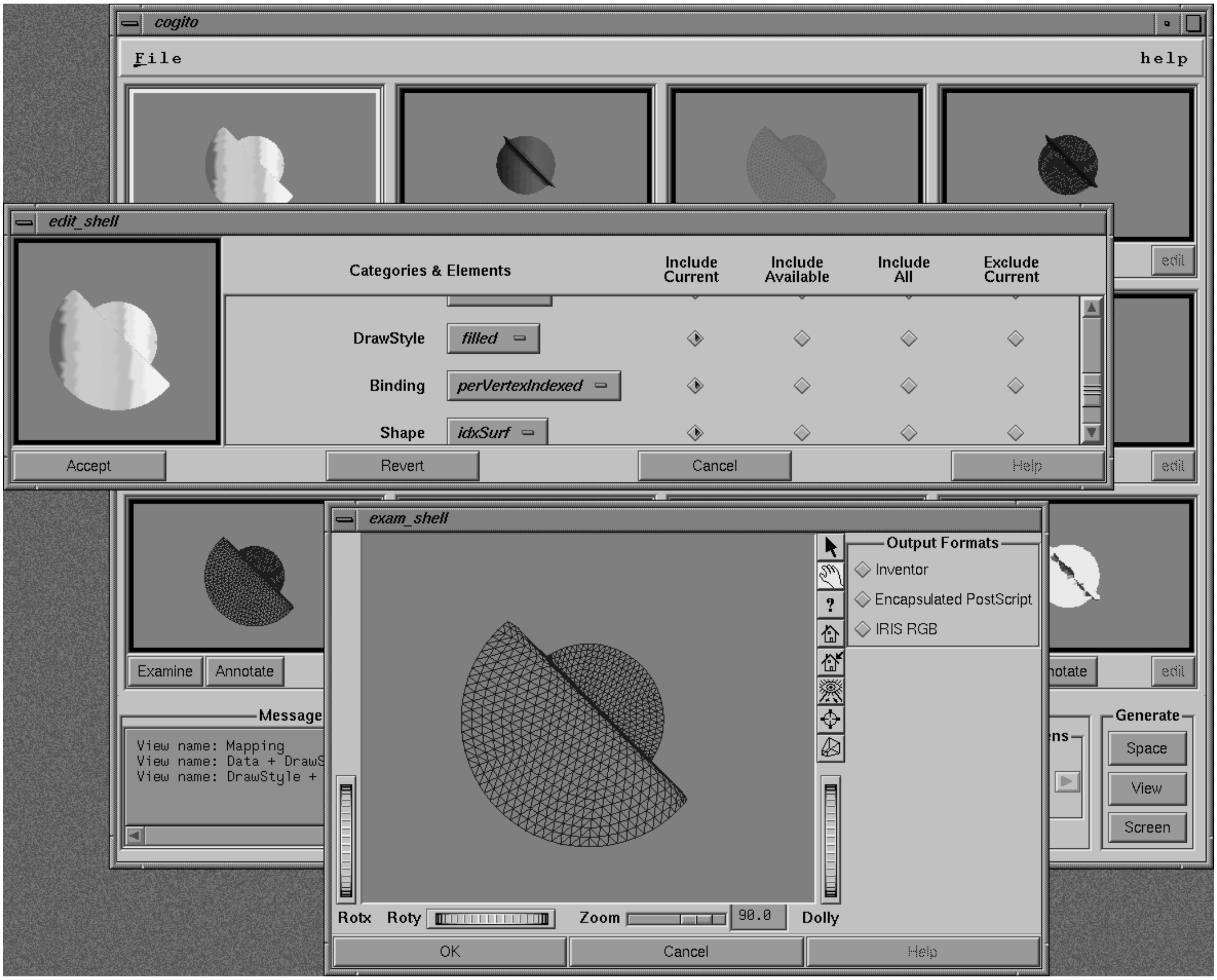
A good strategy for exploration of the whole search space is to pursue a particular direction with each search and perform many searches. Left without user guidance, the system will present images taken from the entire search space. In this case, however, a higher proportion of the alternatives presented may be deemed unusable; and since only a relatively small number of alternatives are sampled and displayed, useful ones may be missed.

Programming is still required in support of the cogito system, but it differs from the traditional model in two ways: it is done separately from the interaction during which images are evaluated; and it is done in terms of components instead of complete representations. Mathematicians and programmers work together to define the abstractions which will be used to work with the problem and then to build and assemble the relevant components and categories. The mathematician alone then works directly with the cogito system to find the visual representations meaningful for the problem. These two steps may be performed iteratively as understanding of the problem develops.

The cogito system promotes the notion of process over end-result. It creates a map of the user’s exploration in the space of alternative visual representations, which can provide the basis for new directions in searching of the space. In this way, support is provided for one’s notion of the solution to develop and be refined over the course of a single session or several subsequent sessions. And it provides a firm basis for sharing one’s insights with others.

Visualization of Moving Mesh Numerical Methods

Adaptive methods are an important class of techniques in the numerical solution of partial differential equations (PDEs). For most scientific and engineering problems, the associated PDEs have large variations in their solution behaviors. These variations indicate that different numerical treatments should be adopted in the different solution regions. For instance, small elements are necessary to capture the fine structure of the solution in regions where the solution changes rapidly, and large elements can be used to reduce



DrawStyle: filled
Binding: perVertexIndexed
Shape: idxSurf
Zoom: 90.0

```
View name: Mapping
View name: Data + DrawS
View name: DrawStyle +
```

Figure 2: Screen capture from cogito working on the moving mesh data

the overall computational cost where the solution does not change much. One may generate the solution-adapted meshes by solving a system of mesh generation equations. A fundamental task for numerical analysts is to understand the mesh adaptation procedure. Theoretical analysis is applicable in simple cases, but for the more general cases it remains a complicated issue [14].

Cogito is applied in this field to help the numerical analysts to understand the mesh adaptation procedure, and to design more efficient and effective adaptive solution methods. Here we use a set of data, obtained from the numerical computation of a particular PDE, which represents the solutions of the physical problem and the values of the mesh-generation monitor function, quantities which reflect the expectation of mesh distribution at different times [5].

Several solution features are of great interest to numerical analysts, including: sizes of elements, mesh density, aspect ratio of elements, and speed (both magnitude and direction) of mesh points. For numerical analysts, it is desirable to start from one or more of the above features, proceed to suitable pictures which correspond to certain critical times, and then explore other aspects related to these pictures. We apply cogito to display a handful of pictures initially representing various but limited combinations of the solution features. The user may select one or more of these features, examine the corresponding pictures, and make furthur selections. This interactive procedure facilitates the tasks which numerical analysts perform to understand the mesh adaptation procedure; to examine the fine structures of certain critical solutions; to discern the regions where the numerical method encounters difficulties; and ultimately to design better numerical methods for the solution of scientific and engineering problems.

Conclusions

We claim that this exploratory approach to visualization is effective in fostering discovery. It allows each user the freedom to create meaningful visual representations, in a way which does not require programming of the system. The direct-manipulation style of interaction largely removes the requirement of translation while maintaining access to available alternative representations.

The emphasis on the process of discovery, and the annotation tools which support it, means that this system's approach will provide a firm foundation for communication of visual representations and insights to colleagues and still broader communities.

References

[1] D. T. Campbell. Blind variation and selective retention in scientific discovery. *Pyschological Review*, pages 380–400, 1960.

[2] E. H. Chi, P. Barry, J. Riedl, and J. Konstan. A spreadsheet approach to information visualization. In *IEEE Information Visualization Symposium 1997*, 1997.

[3] G. Fischer and B. Reeves. Beyond intelligent interfaces: Exploring, analyzing, and creating success models of cooperative problem solving. *Journal of Applied Intelligence*, 1:311–332, 1992.

[4] A. J. Hanson, T. Munzner, and G. Francis. Interactive methods for visualizable geometry. *Computer*, 27(7):73–83, 1994.

[5] W. Huang and R. D. Russell. A high dimensional moving mesh strategy. *Appl. Numer. Math.*, 26:1–14, 1997.

[6] S. Kochhar. Ccad: A paradigm for human-computer cooperation in design. *IEEE Computer Graphics and Applications*, 14(3), 1994.

[7] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2), 1986.

[8] J. Marks et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Computer Graphics: SIGGRAPH '97 Conference Proceedings*, 1997.

[9] D. Norman. *The Design of Everyday Things*. Doubleday/Currency, New York, 1990.

[10] D. N. Perkins. Insight in minds and genes. In R. J. Sternberg and J. E. Davidson, editors, *The Nature of Insight*, pages 495–533. MIT Press, Cambridge, MA, 1995.

[11] H. W. J. Rittel. Second-generation design methods. In *Developments in Design Methodology*, pages 317–327. Wiley and Sons, 1984.

[12] S. F. Roth, J. Kolojejchick, J. Mattis, and M. Chuah. Sagetools: An intelligent environment for sketching, browsing, and customizing data-graphics. In *Proceedings CHI'95 Human Factors in Computing Systems*. ACM Press, 1995.

[13] K. Sims. Artificial evolution in computer graphics. In R. J. Beach, editor, *Computer Graphics: SIGGRAPH '91 Conference Proceedings*, pages 319–328. ACM Press, 1991.

[14] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation*. North-Holland, New York, 1985.

[15] C. Upson et al. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.