VISUAL MATH : A FANTASY FOR THE FUTURE OF EDUCATION

Ralph Abraham
Peter Broadwell
Brian Beach

University of California at Santa Cruz

June 7, 1979

VISUAL MATH : A FANTASY FOR THE FUTURE OF EDUCATION

Ralph Abraham
Peter Broadwell
Brian Beach

University of California at Santa Cruz

1. Introduction:

The computer revolution, like a river of molten lava, courses along the cold banks of an indifferent population. Furthest removed from the action, yet destined for total transformation,is the educational system. What form it will finally take is a subject for fantasy.

Some fantasies have already materialized. The mainstream of these, CAI, while of great interest to the computer industry and the experimentalists of cognitive psychology, has never made a serious impress on the teaching establishment. Its great monuments, such as Plato and the Baltimore Learning Center, are wonderful creations. But, here is a different fantasy.

2. Visual versus verbal modes:

Since Euclid at least, the mathematical community has wrestled with the problems of describing visual concepts verbally. This indirect strategy works with some persons, and creates rage in others. The key axioms of our Visual Math Project are the following:

I. Visual/verbal translation

-- is difficult
-- develops in childhood
-- difficulties contribute to math anxiety and rage
-- is learnable.

II. Color video computer graphics will

-- provide a new medium for direct visual communication of geometric concepts
-- develop visual/verbal translation skills and thus math capability
-- increase the success of the math curriculum
-- decrease math anxiety and rage.

Clearly these assumptions are testable. Cognitive psychologists have recently discovered visual modes of cognition, and interest themselves in this ageless struggle of mathematicians and geometers.

3. On Diffusion:

Another prejudice, more political than psychological, affects our project:

> III. Transportable software for affordable stand-alone hardware will diffuse widely through society, without the necessity of restructuring its institutions.

Specifically, we have taken aim at the teaching of math in schools and universities, and have in view the potential of the new media to:

- -- increase the popular appreciation of mathematics, and its life-positive applications to, science, and technology
- -- enlarge the world-wide pool of scientifically informed persons
- -- open doors to the technical professions.

4. On Hardware:

Our axioms dictate hardware decisions in the current market place, and also point toward new products just around the corner. Two schemes are currently practical:

- -- System A: a color video graphics terminal with a serial link to a micro-computer with floppy disk.

These systems have bimodal screens: alphanumeric mode, or graphics mode, or both (overlayed). In this category several configurations are possible. These may be classified by resolution and price (see Table 1.)

- -- System B: an alphanumeric terminal with a serial link to a microcomputer with floppy disk, and a parallel link from the microcomputer to a color video graphics (output only) device.

These are also classified by resolution and price, but available systems are not yet affordable on our scale. Within a year or two, such a system should be possible in the same price range as (A) above (see Table 1.)

5. On Languages:

Our axioms dictate software decisions as well. We want a software system which is compatible with the hardware schemes described above, diffusable, and organized as a tree of options referred to a central root of operating system and utility programs. Thus, 16K-BASIC, UCSD-PASCAL, and TINY-C are our current target languages, while UNIX BASIC, PASCAL, and C are our development languages. The central root of our software system -- a universal graphics utility program with limited editing features called SIGMA, which we will describe below -- is easily adapted to (and currently runs on) any of the hardware choices currently available.

6. Current Status:

We now have a "Prototype Visual Math Learning Center", with two color video graphic terminals -- a Ramtek 6200A and a Tektronix 4027 -- connected to the university Computer Center time-sharing system (PDP 11/70 with UNIX). A Hewlett Packard color plotter is on the way. These machines have been purchased with funds from the University of California Task Force on Instructional Improvement, and gifts from Ramtek, Tektronix, and Hewlett-Packard. Public access to the system is provided by several Tektronix calligraphic terminals(4006, 4016, 4051) and plotters. We have several grant proposals outstanding to NSF and DHEW to implement the Visual Math Learning Center on a larger scale.

7. The User's View:

The system itself is best understood from the user's point of view. The user sits at a graphics terminal in one of the public terminal rooms, types GRAPHICS on the keyboard, and receives a welcome message, then a menu, shown in Figure 1. Each choice produces a new menu (Figures 2A, B, and C) except EXIT (which logs the terminal off the network) and HELPER (which prints documentation on the screen). The options under TOOLBOX (Figure 2A) are all basic visual math tools, intended for more experienced users. We also use them extensively in preparing movies and learning games. The choices under CAVE (Figure 2B) are old movies, saved for demonstrations or nostalgia. The learning games in ARCADE (Figure 2C) are interactive in a limited way, and are intended for computer novices. Each aims at exploring a single math concept. First the user reads an introductory message, and answers simple questions such as: "What function do you want to integrate?" (Typing "?" brings hints and sample responses to the screen, and an input parser makes perfect punctuation unnecessary). Next, a computer graphic movie, with labels, running commentary, flash-backs, etc. - unrolls on the screen. For example, a sequence of stills from a movie on integration is shown, in Figure 3. Finally, there is a choice to return to the first (interactive) stage for another experiment, or to exit, back to the ARCADE menu (Figure 2C).

8. The Future:

The format of these learning games has just begun its evolution, and we are constantly seeking to improve it. Soon we will be working with a visual cognition psychologist, Kristina Hooper, toward more sophisticated and successful displays. We also want to increase the interactive options, and to use digital voice synthesis to replace the silent movie titles, which distract from the graphics. The feedback from users, and from cognitive psychological experiments will be the basis for further evolution of this learning medium. And hopefully, some student users will create fantastic new learning games, leaving them behind for our menu, and future users.

9. Software Overview:

Software is the bottleneck in generating computer graphic images. We have tried two distinct ways of designing our software.

First, and most traditional, was a package of functions, integrated into each user program. To draw axes, for example, one called the function FNA% (with appropriate arguments to place tick marks, etc., as desired). Calling the function would generate axes on the screen, directly. If the complete function package was appended to a program, it allowed fairly high level control over the output. One problem we encountered was size. With all the functions in a program, there was not space for much else. Another problem was understandability of the programs. The function names were cryptic codes, with a rather obscure relation to their action, ie.: FNQ% (which drew a cube). As a result, programs written this way quickly became unmanageable. Also, you had to be a real "computer nut" to enjoy learning how all the functions fit together, which ones could be left out to give more room, and so on.

Our second approach has been much more productive. Taking the stand that graphics design is most natural in problem space, we broke the image generation-processes in two parts. One is totally concerned with generating the image in the problem space, the other with translating from this general problem space to machine language. In this system a program that wants axes displayed prints the word "axes" into a file. Then a second program (SIGMA) reads this file and translates from the word "axes" to axes on the screen.

The advantages of this two step system are numerous:

-- application programs become very simple

This interpreter (SIGMA) is rather high level. It takes real numbers as coordinates for objects in two or three dimensional space. It then maps these into machine coordinates according to user-specified scaling, rotation, and positioning information. It also allows for convenient repetition of common shapes, user-defined shapes, and placing textual information within the image. Choice of viewpoint and type of projection are also selectable in the three dimensional case.

-- its commands are ASCII character strings which can either be in a file or entered by the user.
-- easy to edit picture files with text editors
-- 3-D picture files can be viewed in perspective from different (or

moving) viewpoints

-- compute bound images (hidden surface removal for example) can be displayed as quickly as very simple ones.

The slow part is done separately, generating a picture file, then SIGMA quickly turns this into an image.

Having this high level interpreter allows the application programs to focus on the user's problems. For example, plotting a function becomes simply a loop to generate its values over the desired range.

-- machine dependence becomes isolated in SIGMA.

It has proved easy to adapt it to new machines and no other software is affected.

-- application programs are essentially language independent.

We only noticed this after we switched operating systems from RSTS to UNIX and had multiple languages available. Most languages support the creation of ASCII text files so the application programs can be written in whatever language is best suited or most familiar to the programmer.

The main disadvantage is the lack of dynamic interaction with the image. User input is solicited, then an image is generated. Keeping this process short allows significant interaction however and the benefits in simplicity more than make up for the restriction in most applications. In exceptional cases more interaction is obtainable by useing code from SIGMA as in the traditional package first described. We have done this only for games and a freehand drawing program.

10. <u>Postscript</u>:

Many readers will note the parallels between our project, and the Physics Computer Development Project at the Irvine campus of the University of California. This system is farther advanced than ours in many ways, but has more CAI flavor than our laboratory scheme, and it is primarily verbal. What other related projects are under way? We would be happy to hear about them. We are willing to share software and experience with other groups. And finally we are keenly interested in influencing the manufacture of a hardware package suited to our project.

Table 1. Representative Hardware in 1979

Maker	Apple	Ramtek	Tektronix	Ramtek
Model	II	6110	4027	6300
Resolution	Low	Medium	High	UltraHigh
Pixels	128x128	320x240	640x480	1024x1024
Colors	8	8	8	16
Price	3000	8000	15000*	30000*

* = includes 5000 for accessory cpu

Figure 1

The available catagories are:

```
arcade   - Some math-learning experiments
cave     - Storehouse of historical visual math movies
games    - Some fun graphics games
toolbox  - Library of graphics utility programs
helper   - easy access to some helpful documentation
exit
```

Figure 2A

The available tools are:

```
plot      - plot a function of your choice
follow    - follow a vector field
sigma     - draw pictures line by line
hist      - plot a histogram with your data
scat      - plot a scatter diagram with your data
graph     - plot a 3-D surface that you define
points    - draws lines between points you enter
polar     - plot a function in polar coordinates
curve     - draws a curve defined in parametric form
exit
```

Figure 2B

The old movies we have in this cave consist of :

```
int.sub    - A display of graphical integration.
dif.sub    - A display of graphical diferentiation.
saddle.sub - A nice three-dimensional saddle.
wink.sub   - A curriouse vector field.
exit
```

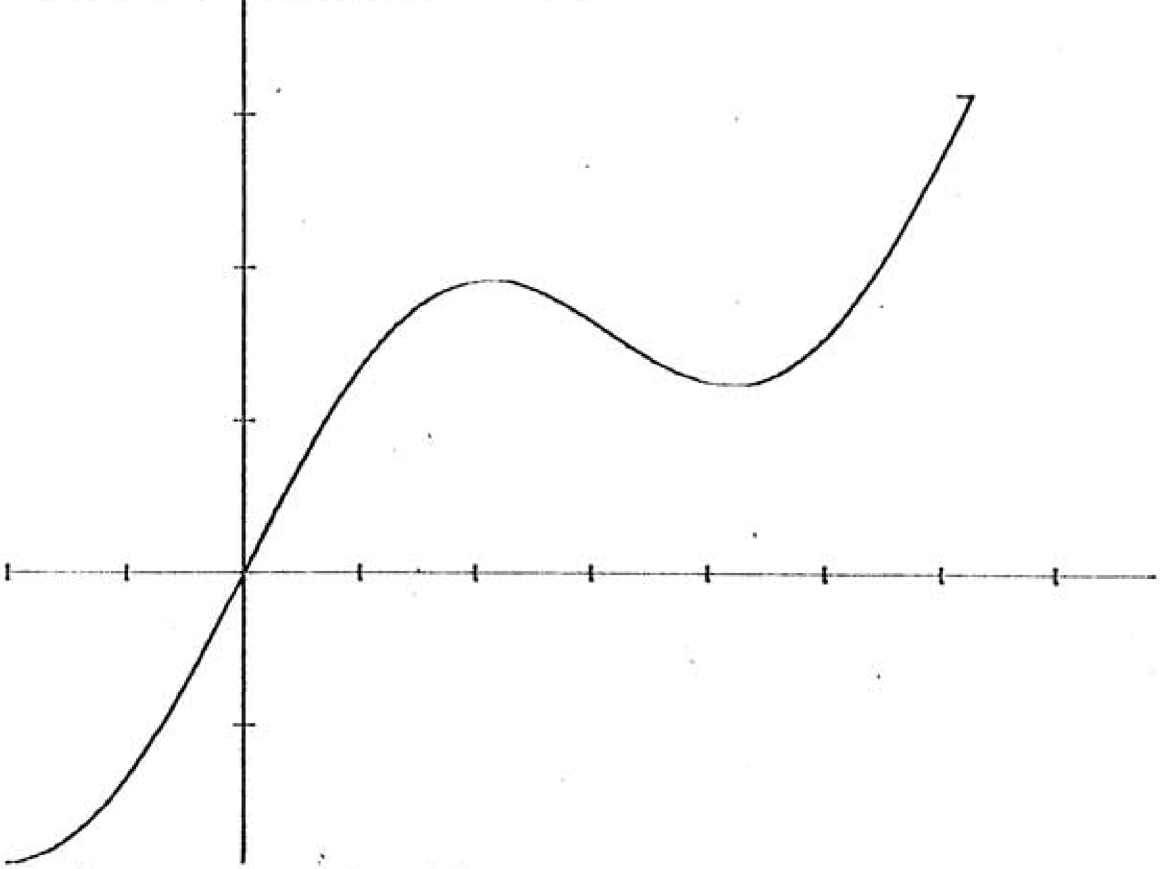
Figure 2C

The available Arcade experiments are:

```
calc   - draws the first and second derivitives of
         any function
dif    - generates a movie of graphical differentiation
         of the function you enter
int    - generates a movie of graphical integration
         of the function you enter
line   - experiment with the formulae Y = (M) X + B
sketch - generates a movie of the process of
       curve sketching using the curve you
       enter as the target
exit
```

Figure 3A

Your function is : sin(X) + X/2
The X domain is from -2.000 to 6.283



The lower sum using 3 partitions is : 1.076
The upper sum using 3 partitions is : 16.625
The difference between the sums is : 15.550

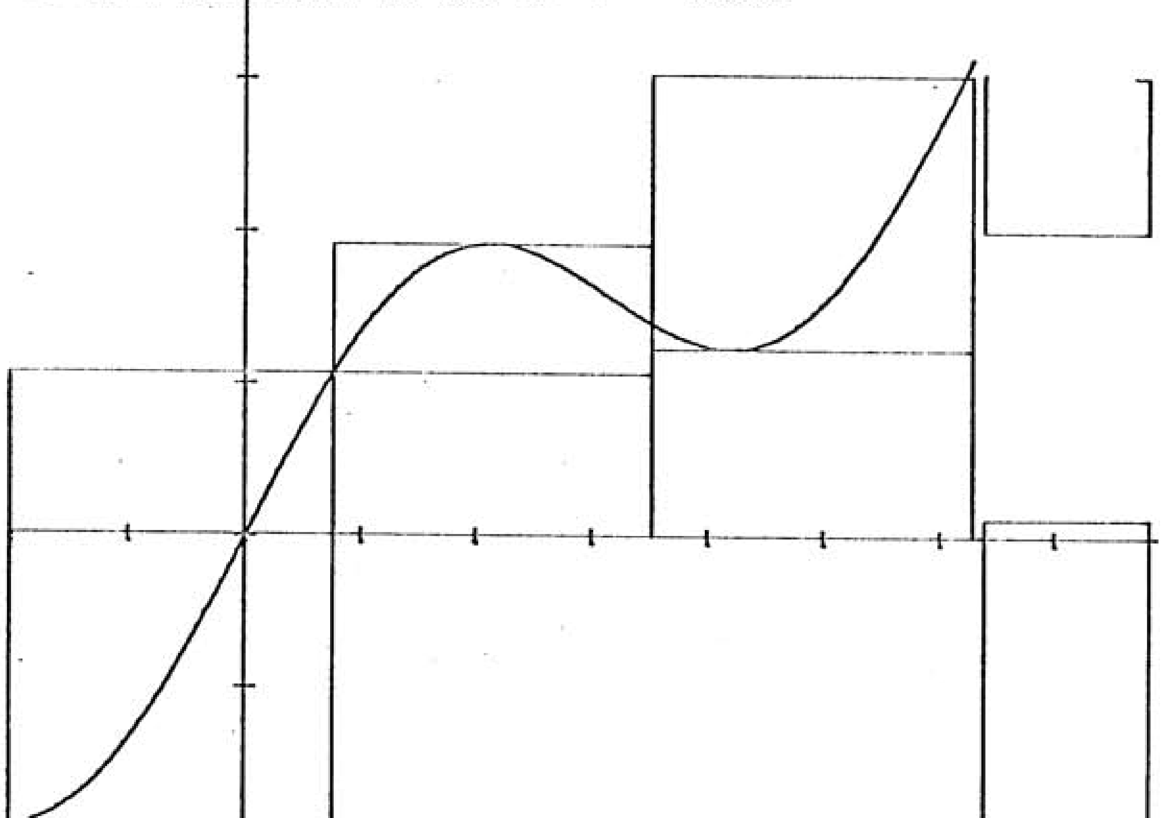
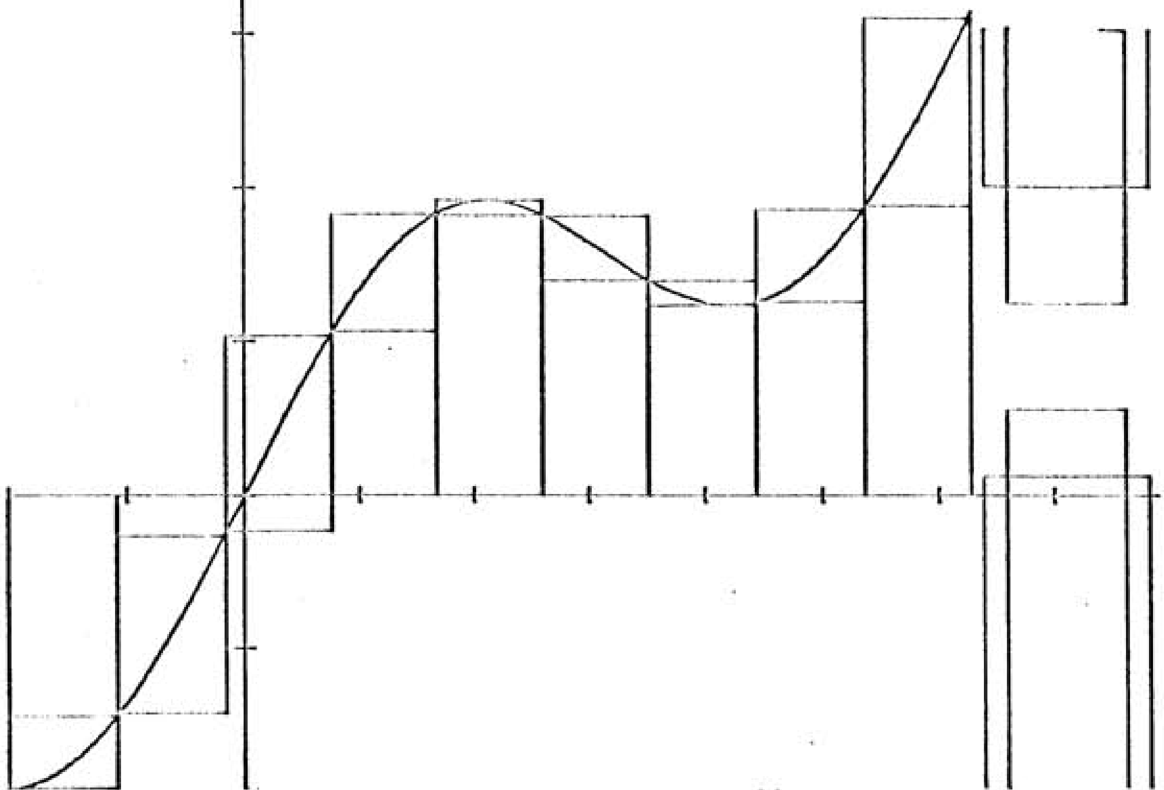


Figure 3B

The lower sum using 9 partitions is : 4.689
The upper sum using 9 partitions is : 10.341
The difference between the sums is : 5.652



The lower sum using 21 partitions is : 6.244
The upper sum using 21 partitions is : 8.442
The difference between the sums is : 2.199
Using even more partitions I get that
the area under this curve is : 7.280