

What Students Should Know About Technology: The Case of Scientific Visualization¹

Jeffrey S. Friedman² and Andrea A. diSessa³

Starting with the focal question, “what should students know about technology?” we describe and illustrate a way of designing educational technology that is strongly informed by empirical studies of how students actually understand and use a technology. We also have theoretical aspirations in developing what we hope to be general principles that can, along with empirical data, orient design.

The type of technology used to illustrate this design methodology is scientific visualization software, in which spatially distributed data is given form as adjustable and often highly suggestive visual displays. Our primary contention is that what students need to know about this software is precisely those aspects of it that define it as a system of representations. More generally, we advocate representation as an important instructional target, and we examine what students know that can be enhanced by appropriate technology and learning activities.

KEY WORDS: Educational technology; scientific visualization; representation.

INTRODUCTION

What should students know about the complex technologies that they use as tools for learning? A common stance is that the technology should be transparent and effortless to use. Students should “see through it” directly to the world or to the concepts of the domain (Hancock, 1995). For instance, calculators free students from thinking about algorithms so that they can focus on developing problem solving skills. Understanding how a calculator calculates is not evidently relevant for learning these skills. Light probes attached to graphing computers save students from the tedium of manually reading a light

meter, recording these values, and then plotting them on a graph so that they can spend more time discovering a quantitative relationship between light intensity and distance. Why should students need to know anything about how the light probe works, or how the computer treats the data to turn it into a graph? At best, it seems, students only need to know how to use these technologies.

We take an opposing stance. Technology, in some cases, should *not* be transparent. We will be considering a case where there is theoretical and empirical support for the proposition that understanding a particular part of the workings of the technology is critical to learning with it. Furthermore, what students should know about the technology and what the technology itself should show or hide is not, in general, an a priori matter. It must be informed by consideration of what students actually do with it and what they learn. During design phases, we must do the best we can at envisioning future use and learning. And after design, or in formative stages, empirical work is almost always critical in revising guesses for what should be shown and explained. In this work, we feel we learned what students needed to know

¹The work in this paper was supported, in part, by a grant from the National Science Foundation, A. diSessa, PI, RED-9553902. This work was also supported by grants from DODD, C. Pennypacker, PI, N66001-95-C-8622 and National Science Foundation, C. Pennypacker, PI, ESI-9554161. The opinions expressed are those of the authors, and not necessarily those of the NSF or the Department of Defense.

²Graduate Group in Math and Science Education, University of California, Berkeley.

³Graduate School of Education, University of California, Berkeley, California.

about the technology only by careful attention to their accomplishing—or failing to accomplish—the embedding educational tasks.

In approaching the problem of what students should know about technology, we want to illustrate a general orientation towards the design of educational technology. We feel much technology design is ad hoc. In addition to a stronger empirical component, introduced above, we emphasize principles and activities. Principles represent theoretical commitments in design. Our orientation towards activity stems from the belief that technology can only be evaluated together with, and in the context of, the educational activities in which students are intended to learn.

We begin with three heuristic principles that define, in part, a philosophy of technology design, including the design of (a) artifacts, (b) student activities using those artifacts, and (c) instruction about the artifact. These principles may not seem controversial, but later refinements and the priority we give these principles over alternatives are more distinctive.

1. Design for the best contact with the most important scientific ideas. As science educators, we hope that students will use the technology for learning something of enduring value about science.
2. Build on students' strengths: Foster uses that involve productive engagement of rich student knowledge and creativity. This is a familiar constructivist orientation. However, identifying particular prior student knowledge and how to build from it is a contribution.
3. Students should find learning activities personally engaging. Several ingredients may contribute to a sense of personal engagement, including opportunities to use prior knowledge, the satisfaction resulting from accomplishment, and a feeling one is learning something of importance. In our work, we have put a strong premium on design that can engage student interest in extended, personalized projects.

We are interested in applying these principles to a particular kind of technology, namely computer programs that create visual displays of spatially distributed data. Such software includes many “scientific visualization” tools to view, for example, color-coded images of the earth taken from satellites, or, what is most relevant here, electronically transformed images of astronomical objects. Visualization software is being used more and more in efforts to improve

science education, typically at the high school level (Greenberg *et al.*, 1993; Barstow and Barenfeld, 1996; Asbell-Clarke and Barclay, 1996, Gordin and Pea, 1995).

Most scientific visualization software can be classified as representational. That is, its purpose is to present a visual form as a surrogate for data. The visual representation is intended to facilitate inferences about the data or phenomenon that is represented. Noting that scientific visualization software is representational is critical, and our central claims all revolve around it. In particular, we will try to show that representational software, and scientific visualization tools in particular, should not be transparent. Instead, they should show users relevant aspects of their representational structure: In order to use this software effectively, students need to understand how it works as a representational device—that is, as a mechanism designed to produce adjustable representations. Although there are other things that students may need to know about this kind of software, we will show that its nature as a representational device is not optional.

Anticipating later parts of this paper, we specialize our beginning principles to the present context and to address the issue of what students should know about scientific visualization technology:

1. Design to support understandings of representational aspects of scientific visualization technology. As we will illustrate, to use this software effectively students need to consider representational aspects of the technology. In particular, we will try to show that representational software, and scientific visualization tools in particular, should not be transparent. Instead, they should show users relevant aspects of their representational structure. Proposing that students think about the representational aspects of scientific visualizations goes beyond the practical importance of using the technology effectively. We claim that representation, itself, is an important scientific idea. In the next section, we elaborate this claim.
2. Design to take advantage of students' resources for creating and interpreting representational displays (i.e., the scientific visualizations). As we will illustrate, intuitive perceptual interpretive capabilities are the central prior resources on which we seek to build. Technology design must be sensitive to

these resources, and activities should specifically address their development.

3. Design for personal engagement and success. As we will show, our students generated an impressive variety of interpretive methods. We will identify aspects of the design of technological resources and learning activities that can help students build on their capabilities for interpreting. This may lead to personalization and investment. Although we will not emphasize it here, allowing students to design *new* representations is an additional avenue of creative engagement.

We view design as an iterative process in which these principles will be further specified. Based upon empirical findings, we anticipate elaborating upon particular ways that technology and activities may support thinking about representational aspects of scientific visualization tools, ways that students use their own constructive resources, and ways that technology and activities may support engagement.

*Long-Range Learning Objective:
Meta-Representational Competence*

In this paper, we propose students should understand the representational aspects of a particular technology. Over the long term, we also hope students will learn something more generally about how representations work. Although learning generally about representations is not emphasized in this paper, it is an important part of our long-term agenda, which we explain in this section.

Knowing about how representations work, in general, is what we call meta-representational competence (MRC). MRC comprises, for example, abilities to design and critique representations, to understand how they work, and to learn to operate new representations quickly and effectively. MRC is not traditionally very visible among the repertoire of scientific skills slated for students to learn in school. Neither is representation prominent in canonical lists of important scientific ideas. However, we believe this relative invisibility is unwarranted and should be corrected. This is no small issue, and we cannot argue it extensively here. But, consider the following:

- Scientists have always and still continue to develop representations in order to do their work. Graphing, tables, and so on, have become so standard and commonplace that we

forget there was a time when these forms did not exist.

- In addition to substantial classes of representations, scientists are always inventing simple or more extensive modifications of existing representational forms. Even graphs appear in many variations that are tuned and judged to be adapted to particular circumstances. A perusal of research journals, or even of textbooks, shows a wide range of variation and also significant invention.
- Computers have introduced a huge, new premium on MRC. Not only are new forms of representation made possible, but also individual representational forms are much more complex and adjustable. Scientific visualization tools, themselves, are a perfect example. If students want to get in on the modern game of doing science, they will need to become accomplished with selecting and using a broader range of more complex representational devices.

Considering MRC to be an instructional goal immediately raises the issue of what prior knowledge students might have about representations, as in principle 2, above. While it is easy to imagine the answer to this question is “not much,” especially in view of lack of attention in current instruction and even well-documented incompetencies and misconceptions concerning standard representations (Leinhardt, Zaslavsky and Stein, 1990), prior work of our project suggests the opposite. Students at least as young as sixth grade show, in some contexts, remarkably rich and effective MRC. We provide a very brief review here.

In a sixth grade classroom we discovered that students had a wide range of excellent resource that aided them in the design and critique of scientific representational forms (diSessa, *et al.*, 1991). Students were asked to design paper and pencil representations that were effective in conveying information about motion. Over a sequence of four days, students produced a remarkable set of representations. They produced well-founded criticisms, and extended their line of invention into standard representational forms, such as graphs. This activity has been replicated at several educational levels, including high school, and we have continued to find rich prior resources that are spontaneously engaged by the task (Sherin, 1997).

More recently, we have discovered other ways

in which students seem to exhibit substantial MRC. In this case, high school students were using computer image processing to create and interpret representational displays of data (Friedman, 1996). The students generated a wide range of innovative interpretive methods to glean information from image displays. Interpretive methods are the means by which representational forms are used in particular contexts to infer information about the target of the representation. Interpreting representations falls mainly under the “learning to operate new representations quickly and effectively” portion of our definition of MRC. We argued in Friedman (1996) that students’ abilities to invent methods arose out of their substantial intuitive perceptual capabilities.

A particular subcomponent of MRC, which will emerge here as important, is students’ inclination towards and capabilities for critically evaluating intuitive or commonsense interpretations. Skillful visual data analyses require sensitivity to the fact that looks may be deceiving. Thus, commonsense interpretations should be neither uncritically accepted nor rejected, but they should be evaluated for their usefulness.

We have identified MRC here as, potentially, a valuable and achievable goal of instruction in and of itself. We have cited earlier work to suggest that there may be significant prior resources on which to build MRC. Later stages of this paper will elaborate on particular prior resources that students have with respect to developing more and better MRC. It also turns out that critical evaluation is an important part of what students need to do to complete typical tasks involving scientific visualization software effectively, so this will constitute a special component of MRC that is essential to develop.

Summary and Organization of Paper

The heart of this paper is the intent to illustrate a more accountable process of technology design. We are advocating an iterative process that involves empirical work at each stage. Also, we are attempting to illustrate a deeper theoretical accountability by proposing and illustrating a set of principles that guide our design. Finally, we claim that technology design should be guided and accompanied by activity design. Technology does not stand alone.

In the next sections, we describe some preliminary empirical work that motivated attention to representational aspects of image processing, which is an

important kind of scientific visualization tool. After a short section describing how this kind of technology works, we look at some preliminary studies of students using a computer image processing tool for creating and analyzing displays of digitized astronomical data. These studies will illustrate categories of understandings of the technology and their consequences for effectively using the technology. The last part of the paper describes a second iteration of design, a new visualization software system and activities designed to better support thinking about representational aspects of the technology.

IMAGE PROCESSING AS A REPRESENTATIONAL DEVICE

This section does double duty. First, we explain general features of the software that students used in empirical work so that it is easier for readers to understand what students were doing and why. But, in addition, we will be describing this software so as to make its features as a representational device evident. This is important in making our suggestions about what students need to understand about the technology generalizable.

Image processing software is a particular kind of scientific visualization tool. The term “scientific visualization tool” does not have a precise definition, but it is helpful to distinguish two categories: aids for developing understanding and tools for analyzing data. Simulations, such as interactive microworlds and 3-d simulations of molecular structures, are often intended as a direct aid to conceptual development.

Falling in the second category, image-processing tools are commonly used by scientists in the process of visual data analysis (Lillesand and Kiefer, 1994). Their distinctive usefulness derives from the fact that these tools can create visually suggestive displays of data. Skillfully adjusting the image processing settings, which control the way the data is displayed, can help the scientist see patterns and relationships in the data. In similar ways, the students discussed here use image processing as a tool for data analysis. Image processing tools are clearly representational in the sense relevant to this paper.

More specifically, image processing can be considered to be a representational device in the following manner. A common model of representation is that it involves three components (Palmer, 1977). The represented world, or referent, consists of whatever the representation is about, what is being repre-

sented. The representing world is configured into particular representations to convey information about the represented world. Finally, there is some mapping or set of rules that determines which attributes of the referent are depicted and how they are depicted in the representing world.

Referring to Fig. 1, image processing consists of two representations: the data file represents the night sky and the display represents the data file. For our purposes the link between the display and data file is, by far, the more important. In the case of astronomical data, the data files are often referred to as image data files and the displays are often referred to as image displays. Although some of the technical details and terminology is specialized for astronomy, the fundamental representational features discussed here are common to most image-processing technologies.

The image data file represents the area of sky in the telescope's field of view. An image data file simply consists of an array of numbers (Lillesand and Kiefer, 1994, p. 26). These data files are sometimes referred to as spatially organized data or "raster" data. The mapping is given by the deceptively "simple" fact that data values are proportional to brightness in the corresponding patch of sky. The telescope, CCD camera and supporting hardware and software provide this mapping. The CCD is comprised of an array of linearly sensitive light detectors, so if two numbers in a data file differ by a factor of 3, then the amount of light detected by the corresponding CCD cells differed by a factor of 3. The absolute

numbers depend on several factors, and are not usually meaningful. This part of the representational chain from object to image has proven, in general, less problematic for the students in our studies compared to the mapping from data file to image.

The image display represents the image data file. The image processing software provides the mapping from the data file to the display on the computer screen. Image displays are comprised of an array of picture elements, pixels, where each pixel represents the value of the corresponding array element stored in an image data file. For instance, the pixel at location row 2, column 3 in the image display represents the value of the array at location row 2, column 3 in the image data file.

The software creates the image display by mapping the data file (the referent) to the image display (the representation), which is controlled by user-accessible settings. For example, a user can select a color palette, which is simply an ordered list of colors or shades of gray that will appear in the image. Exactly which colors in the palette correspond to which data numbers is determined by the MIN (Minimum) and MAX (Maximum) numerical settings. A datum with value MIN will appear as the first color in the palette, and one with value MAX will appear as the last color. Intermediate colors will be uniformly distributed over the intermediate data values. (Values less than MIN will appear as the first color in the palette, and values greater than MAX will take on the last color.) With a color palette that consists of shades of gray and appropriate MIN and MAX set-

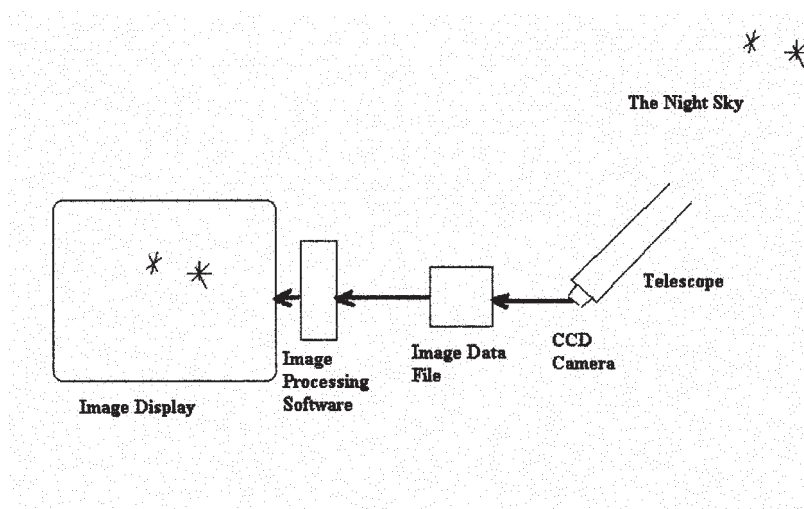


Fig. 1. Image processing technology is a representational device.

ting, the display will look like an ordinary black and white picture of the night sky. A smaller gap between MIN and MAX will generally enhance contrast, as smaller differences in data values will correspond to a given change in color in the color palette. See Fig. 2 for several displays of an image data file of the moon.

To summarize, the image display is a representation of the data file, where the image processing software creates the mapping between the data file and the display. The user may modify the mapping in various ways. The point of the software is to allow users to inspect the image and draw inferences about the data, and hence also about the objects represented.

In addition to image displays, image-processing

software can be used for creating ancillary representations of data files. A particularly useful tool plots a graph of brightness counts along a slice through the image, as in Fig. 2.5. The user specifies a slice by pressing and dragging the mouse.

It is probably evident already that interpreting an image involves reasoning across the representational map. Thus, understanding precisely this aspect of the technology is critical to effective interpretation. Our empirical work, below, bears this out. In particular, students frequently believe they can directly see attributes of objects in the images, without reasoning via the map. This is an entirely comprehensible generalization from everyday life, where our normal interpretive strategies transparently (without explicit rea-

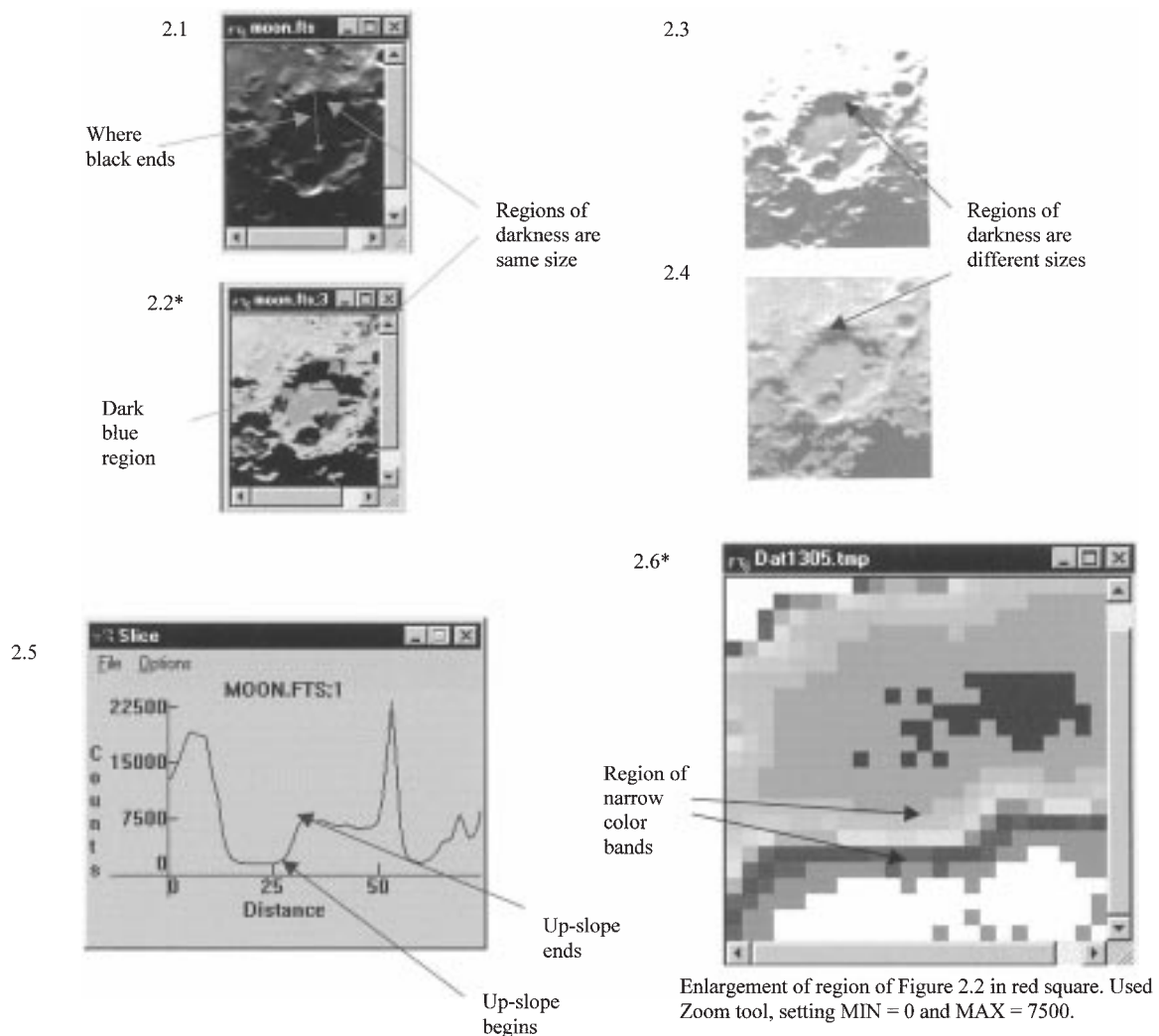


Fig. 2. Image displays and slice graph of the moon. Arrows point to possibly useful visual attributes for locating the edge of the shadow. *See Color Plate.

soning) give us access to knowledge about the world. But, in the case of representational artifacts, such apparent transparency is misleading. In order to evaluate suggestive interpretations critically, students must explicitly consider the relevant representational features of the software.

IMAGE PROCESSING ACTIVITIES IN A SCIENCE CLASSROOM

We intend to illustrate that understanding representational aspects of technology can play a critical role in students' building on and extending their native capabilities for interpreting visual representations. These results will illustrate how empirical findings have helped us to specialize our initial principles, particularly 2 and 3.

In the following examples, students analyze image displays of a region of the moon, such as those in Fig. 2. As described below, the students had just used the image-processing tool pictured in Fig. 3 (see Fig. 4, color plate, for color version) to create similar displays. Since we will be presenting examples where students experience difficulties when interpreting displays as representations, we encourage the reader to reflect on how well the display in Fig. 3 supports thinking about the representational aspects of the technology: the representation, referent, and mapping. For instance, note that values in the data file can only be viewed one at a time. Fig. 3 shows an array value count = 2293, at row $x = 2$ and column $y = 5$.

Our analysis of the students' thinking uses a simple model of the process of visual data analysis. As shown in Fig. 5, the student begins with a task, such as finding the edge of a shadow on the moon. Then the student focuses attention on a visual attribute, such as noticing where the blackness ends, as indicated in Fig. 2.1. Next, the student interprets the visual attribute. (The term visual attribute is really a stand-in for the more general term "description." People make descriptions of an image and infer descriptions of the data or referent. Attributes are simply a special type of description, a simple predicate.) For example, "That (where the blackness ends) is the edge of the shadow."

The student may end here, or may continue, reinterpreting the visual attribute. For instance, the student might critically evaluate the interpretation, "This may not be exactly the edge because the display depends upon the image processing settings." Or the

student may choose to create another display or attend to another visual attribute that might be more useful for locating the edge.

The model allows us to localize students' intuitive perceptual capabilities (e.g., what they know about interpreting particular visual attributes) and when students might apply knowledge of the technology (e.g., for critically evaluating a suggestive interpretation, or for selecting new displays with more useful visual attributes). The textual annotations in Fig. 2 show some of the visual attributes identified by students when trying to locate the edge of the shadow. Below, we will illustrate how selecting the most useful visual attributes and interpreting them correctly require thinking about the representational aspects of the displays.

We begin the empirical study proper by describing the data sources for our analyses. Next, we provide a more detailed description of the data and task and illustrate how an expert student might proceed. In particular, we describe how an expert student might use knowledge of representational aspects of the technology for interpreting visual attributes. Then we present several examples of students' analyses. These examples illustrate a range of expertise, as measured against the proposed expert.

Data Source: Students, Classroom, and Activities

The examples presented below were extracted from transcripts of hour-long, videotaped discussions in which pairs of students analyzed digital images of the moon. The students were enrolled in a high school physics and astronomy class, which was participating in the Hands-On Universe Project (Asbell-Clarke and Barclay, 1996). This outreach program provided the image processing tool and data. Approximately six weeks of a two-semester class were devoted to activities in which students use image processing to analyze digital images of astronomical objects (Friedman, 1996).

These data-analytic activities entail making calculations and inferences from digital images of astronomical objects. As elaborated below, the students use the image-processing tool to create and modify displays of the digital images. The activities include calculating sizes (of craters on the moon, solar flares, a fireball on Jupiter created by the impact of a fragment of the Shoemaker-Levy comet, etc.), calculating brightness of stars, distinguishing stars from galaxies, and identifying asteroids. Many of these activities

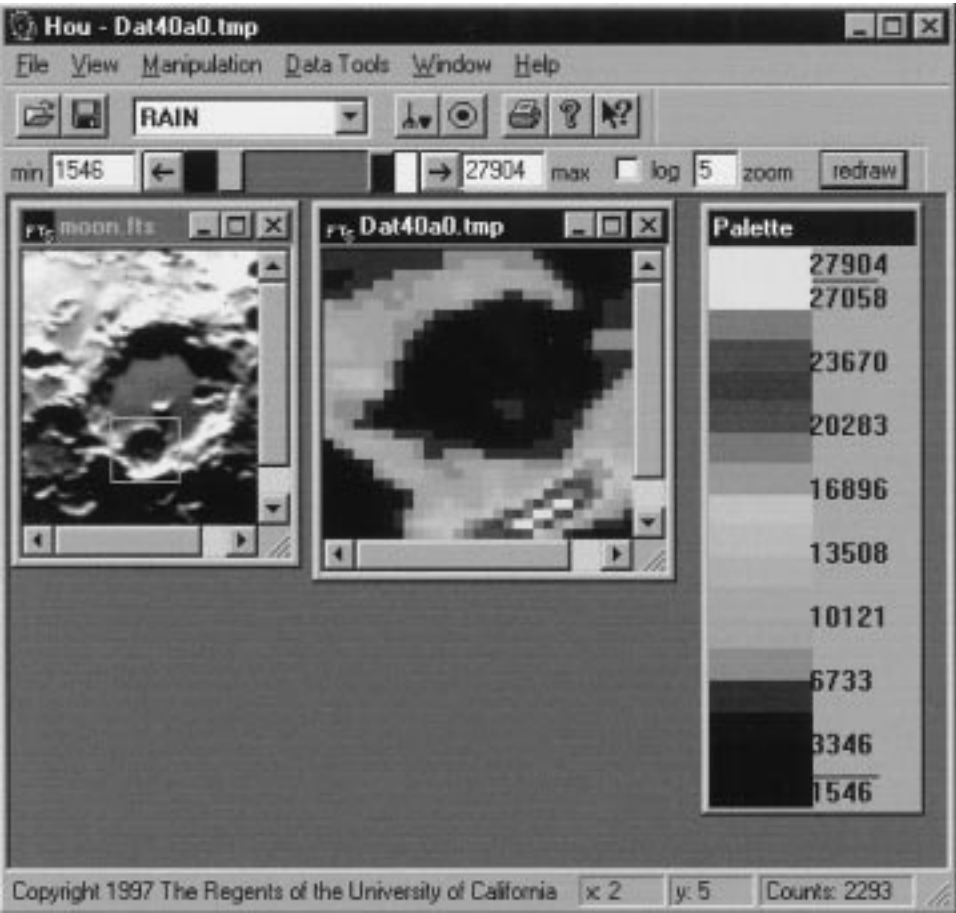


Fig. 3. Image processing tool used in preliminary study. See Color Plate. Reprinted with permission by Hands-On Universe Project.

have common subtasks, namely locating edges (of moon shadows, planets, stars, and galaxies) and identifying changes (in the brightness of an exploding star or the location of an asteroid).

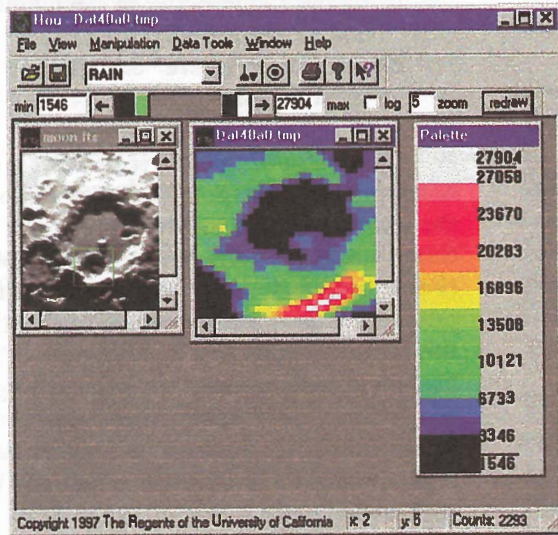
The following examples are extracted from discussions that take place following a lab activity where students used the image processing tools to analyze digital images of moon. In these discussions, students work in pairs and are given a set of pictures and data analytic tasks. They also had access to a computer. The pictures are familiar to the students because they had just created similar looking pictures using the image processing tools. A researcher (I) presents the task, and acts mainly as an observer, occasionally prompting verbalization or focusing attention on issues that had arisen. At times, the researcher prompts students to think about using the computer, answers procedural questions about the task, and occasionally

provides information about the objects depicted in the tasks.

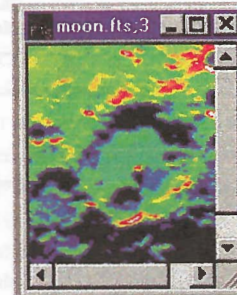
The Tasks and Expert Student

This section introduces the tasks and illustrates ways an expert student might use knowledge of representational aspects of the technology. In particular, the expert might use mapping knowledge and slice graphs for interpreting visual attributes of image displays (the representation) in terms of the data (referent). Table I summarizes the expert interpretations of visual attributes described in this section. The next section compares the students' interpretations to the expert's. As we will illustrate, students attend to most of the proposed visual attributes and consider most of the proposed representational aspects of the tech-

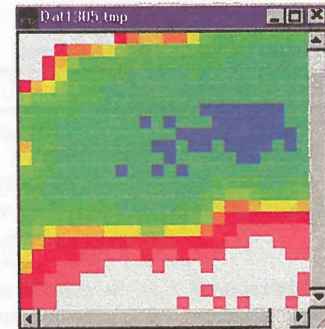
3C



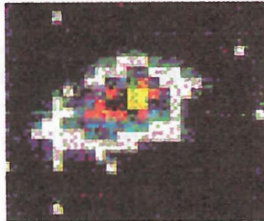
2.2C



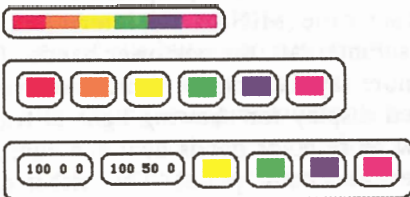
2.6C



11C



8C



9C

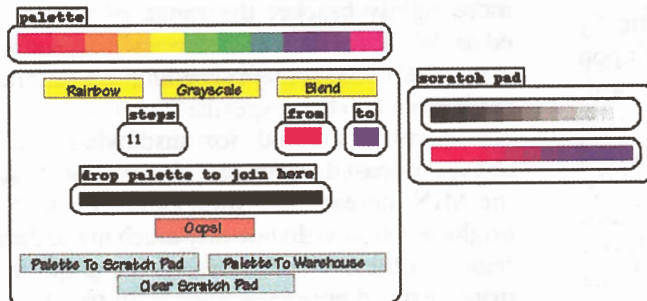


Fig. 4. Color Plate.

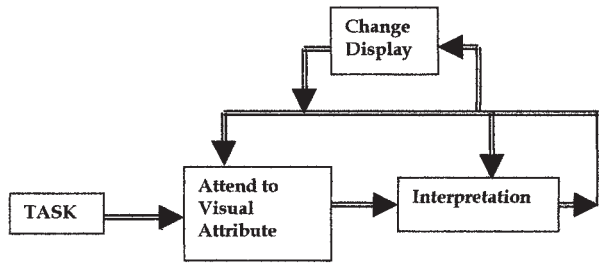


Fig. 5. Model for visual data analysis.

Table I. Expert’s Interpretations of Visual Attributes

Visual Attribute	Expert Interpretation
Black region	Region where counts < MIN
Where black ends	Black where counts < MIN, Grey where counts > MIN
Width of color bands	Data values increases more rapidly where width of color bands is narrower
Sudden increase in area of blackness	MIN setting is a little greater than values of counts in the unshadowed region

nology. In other words, collectively, students have most of the resources needed by the proposed expert. We will also illustrate how the expert/student comparison has helped us identify students’ difficulties. Anticipating the final sections of this report, the identified difficulties informed the design of our new scientific visualization tool and embedding activities.

The task entails analyzing image displays of the moon similar to those in Fig. 2. Students were to identify the edge of the large shadow cast by the crater wall and then to compare the height of the crater wall to a mountain in the middle of the crater (the “central peak”).

Better edge detection methods systematically apply some criterion for edge detection based upon the distribution of light near the edge. Why? As with other edge or boundary detection tasks, it is usually not possible to locate an exact edge. Light detected at the edges of shadows tends to change gradually. This can be seen in the slice graph of Fig. 2.5. Thus, the best displays for edge detection tasks most clearly show the distribution of light near the edge. Some reasonable alternative criteria for locating the edge are (consult the figure) where:

- a. Brightness begins to increase
- b. Brightness stops increasing

- c. In the range between beginning of increase and where the increase stops
- d. At point where brightness is increasing most rapidly.

Using any of the grayscale displays in Fig. 2, it is tempting to interpret the **black region** literally as the shadow and to locate the edge at the location where **the black ends** and the **shades of gray begin**. (For emphasis, we put visual attributes in bold). The proposed expert student would realize that although this may be a good first approximation, the extent of blackness depends upon the MIN setting. The region of blackness represents values less than the MIN. Thus, if the MIN were set to a higher value, a larger region would be displayed as black and the apparent edge would move.

It is much less tempting to interpret a colorized display literally, since it does not look realistic. Still, we find that some students are tempted to interpret the “**darker**” **color bands** as representing the shadow. But the expert would realize that the different color bands have an arbitrary relation to brightness, depending on the settings of the MIN and MAX and the sequence of colors in the color palette. The color bands do show the distribution of light at the shadow’s edge better than the grayscale display since the color hues are more perceptually distinct than the shades of gray. In particular, the **narrow band of blue** in Fig. 2.2 (see also color plate) indicates a rapid rate of change of values. The proposed expert would reason that since the (MIN, MAX) interval is divided into equal subintervals, the narrower bands of color represent more rapid change of values. An even better colorized display for showing light distribution would show more color bands near the edge, as in Fig. 2.6 (see also color plate). This effect can be achieved by setting the MIN and MAX values to more tightly bracket the range of values near the edge. We note that this elaborate reasoning is not easy. We believe this represents an ambitious but achievable level of expertise.

Another method for displaying the rate of change is based upon the observation that raising the MIN increases the area of blackness. Since the brightness counts do not vary much in the illuminated region of the crater floor (see slice graph in Fig. 2.5 from distance approximately = 30 to 50), when the MIN is raised to a certain threshold value, the region of blackness will suddenly become much greater, and the display will appear as if the entire floor of the crater were in shadow. If the MIN is set to the greatest

value before this **sudden increase in blackness**, then the area of blackness corresponds closely to the edge.

The expert student might use the slice graphs to help interpret image displays in terms of the data. Slice graphs clearly show the distribution of light, but only for the selected portion of the data file. For instance, the slice graph in Fig. 2.5 represents the data values along the slice, from top to bottom, on the image display in Fig. 2.1. An expert might use the slice graph for determining MIN and MAX values that would bracket values near the edge. For example, using the slice graph of Fig. 2.5, the brightness counts near the edge (from distance approximately = 25 to 30) increase within the interval (MIN to MAX) approximately 0 to 7500. With the proper settings, the colorized display can show as much detail in fluctuations of light values as the slice graph. For example the region of narrow color bands on the image display in Fig. 2.6 (see also color plate) corresponds to the upward sloping part of the slice graph indicated in Fig. 2.5.

Student Work

The following analysis compares students' interpretations of visual attributes and applications of their understandings of the technology to the expert's. First, we present examples where students approach expertise in the ways they consider representational aspects of the technology when interpreting a display's visual attributes and designing new displays. These examples illustrate the first two specialized principles: focusing on representational aspects of the technology and capitalizing on student competence. Next, we present examples where students did not consider representational aspects of the technology and consequentially misinterpreted visual attributes. Understanding why students have difficulties is a critical step in the argument. We want to show that learning about representations—and, more particularly, learning about representational aspects of the technology—plays a critical role in students' building on their native interpretive capabilities.

Anticipating the final section of this paper, we are interested in designing the new technology and instructional activities in ways that will prompt students to use their knowledge of representational aspects of the technology. We also seek to identify factors that may hinder students from effectively applying their representational knowledge. Empiri-

cally determined factors that hinder and prompts that can help inform our new design.

Chris and Arthur: Applying Knowledge of Representational Aspects of the Technology. Chris and Arthur approach the proposed expertise. They consistently interpret visual attributes of display in terms of the data. They use mapping knowledge and eventually slice graphs to help critically evaluate their interpretations and to help design displays that would be more useful for the edge-detection task. They also touch upon a fundamental characteristic of all representations: resolution.

We now turn to the transcript. Referring to the grayscale display in Fig. 2.1, Chris initially interprets the display as if it were just a picture. At this point, we don't believe he has any representational intent, but only means to describe the display. The students are attending to the **contrast in brightness** and the **transition from black to gray**.

C: so, where does the shadow end? It ends right there (*pointing to where black ends*)

A: Yea. That's where I say it ends.

C: I think it ends right there, too.

Responding to Arthur's request for an explanation, below, Chris applies mapping knowledge, values below the MIN are displayed as black, to interpret a visual attribute, a **black region**, in terms of the numerical data. We categorize such informed explanations as a kind of critical evaluation.

A: How do you know it?

C: Well, you would figure that looking at it . . . this is where you see the darkness starts ending. . . . These points right over here are obviously below the MIN. So they show up as black and when there starts being pigmentation (*gray shades*), then that's where the shadow ends.

In the next segment, the students consider a way to create a more useful display for edge detection. They are still attending to the region of blackness and using their mapping knowledge. The interviewer (I) gives the students two more image displays of the same moon data, similar to Figs. 2.3 and 2.4. The apparent length of the shadow varied because the MIN and MAX settings differed. This prompted the students to consider a way to change the display, based upon their knowledge of the mapping. But they are misled by a mistaken assumption that the brightness counts of the shadow were exactly zero.

Chris's first statement, below, makes it clear that "shadow," to this point, has meant "dark region on

the display.” (The problem in communication is understandable since the interviewer speaks as if there is one place the shadow ends, but Chris knows the edge of darkness depends on the MIN and MAX.)

A: Is this the same picture?

I: It's the same image file.

C: Well, are you asking . . . if we decreased the MIN, if the shadow would be different? Or are you just asking with this setting as it is, where does it end?

I: I'm asking where the shadow really ends on the moon.

C: Really ends on the moon? Well, if it really ends on the moon, you'd have to decrease the MIN to zero to pick up all possible

A: (*interrupting*) pixel settings

C: . . . but with this, all counts below 610 (the current setting of the MIN) will not register

A: It's like all points above 26249 (*the current setting of the MAX*) will

C: (*interrupting*) appear white.

I: So, how would setting the MIN to zero . . . help?

C: Well, that would give you (*the*) actual image of the moon, I think.

Chris seems uncertain. When prompted, he elaborates:

C: I think so. [I think that would give you the actual image.] But I don't think we could actually prove that's the way it is on the moon. Even if we did it on the computer.

I: And why is that?

C: Because we don't have an actual reflecting image of the moon. We just have what the camera picked up for us.

Chris correctly notes that he and Arthur have no way to determine the CCD's sensitivity, “what the camera picked up.” In other words, the students have no way of exploring the CCD's mapping of the reflected light striking the CCD and the data values stored in the data file. However, the students could inspect the data values to test their assumption that the shadow corresponds to zero values. So far, they have not done so.

So far the students have attended to the black region and applied their knowledge that counts less than MIN are displayed as black. In the next segments, prompted to think about other methods, the students use more knowledge to interpret other visual attributes: color hues and different shades of gray. They correctly interpret **color hues** as represent-

ing a range of values that depends upon the settings of the MIN and MAX. But they do not figure out how this might be helpful for finding the edge, partly because they still think there is an exact edge at the border between zero and nonzero data values. In this next segment, the students use the grayscale and color displays in Figs. 2.1 and 2.2 (see color plate for color version of Fig. 2.2):

I: Can you think of anything else (*other methods for finding the edge*)?

C: Well, you could change the colors on it (*referring to color display*).

I: how would that help?

C: Well, that might—it wouldn't actually help. But with gray, it's a little harder to distinguish differences. And if you made the MIN and MAX smaller so that you could make a better distinction between the two different places. Because right now, the colors are spread out over such a large number of counts that it's hard to tell.

Note that the students are touching upon the idea of resolution by thinking about how a display can distinguish differences in brightness. But unlike the expert, they do not consider how increasing the resolution can be used for creating color bands that show rate of change. Most likely, they are still mistakenly assuming that there is an exact edge defined by where the brightness counts of zero end:

I: Which of those methods seems to be the best?

C: What do you think (*asks A*)?

A: I think setting it to zero.

C: I would have to agree.

As the students continue to think of new methods, they eventually think of using the slice tool and discover their mistaken assumption:

I: Can you think of any other methods you might use?

C: Well, we've got the MIN/MAX. We've got the color.

A: Setting it to zero.

C: . . . We've got putting the MIN and MAX closer together. . .

A: or we could use the slice.

C: . . . The slice tool, yes! . . . The slice will actually tell where the zero (*i.e. the shadow*) ends.

I: Maybe this will help you. (*Hands slice graph to students.*)

C: Right. The slice will cut across it (*motioning along the slice from top to bottom*). And when it reaches—Well. See, there's no absolute shadow.

Chris had discovered that the brightness values in the shadow are not zero, “there’s no absolute shadow.” Thus he has skillfully used the slice graph to help critically evaluate his interpretation of the image display.

Summarizing, although Chris and Arthur never considered an “expert” rate-of-change interpretation, they consistently considered representational aspects of the technology as they designed and interpreted displays. Initially, their interpretations were flawed because of a mistaken assumption that brightness counts corresponding to the shadow were zero. But they discovered their mistake with the help of a slice graph.

These excerpts illustrated several factors that prompt students to apply understanding of representational aspects of the technology. First, Arthur’s request for an explanation of an interpretation prompted Chris to apply mapping knowledge. On several occasions, asking the students to compare displays or methods prompted them to step back from interpreting a display and think about representational characteristics of displays. For example, when comparing two grayscale displays, Chris considered the mapping from data values to display, “if it really ends on the moon, you’d have to decrease the MIN. . . .” When comparing grayscale and color display, Chris considered the resolution of the representations, “it’s a little harder to distinguish differences, and if you made the MIN and MAX smaller so that you could make a better distinction. . . .”

It seems to us that these students have ample knowledge of representational aspects of the technology and would continue designing displays, inventing methods for interpreting them, and critically evaluating their interpretations if they had more time. So it seems the best way to help these students is to design engaging visualization tools and activities that will sustain their interest. In fact, comments such as Chris’s, “we don’t have an actual reflecting image of the moon,” have encouraged us to design activities where students analyze digital images of models that they build themselves. We think this might be an engaging way for students to learn about the “represented world” and explore the relationships between image displays, the data file and the represented world.

Diane and Sam: Not Applying Knowledge of the Technology. Compared to Chris and Arthur’s nearly expert knowledge of the technology, Diane and Sam’s knowledge is correct, but limited. This limita-

tion at least partly explains why these students abandon a potentially useful visual attribute in favor of a less useful and incorrectly interpreted visual attribute.

Initially, referring to the colorized display in Fig. 2.2 (see also color plate) Diane locates the edge of the shadow at the **transition from the purple hue to the brighter color hues**. (Purple appears as blue in Fig. 2.2.) Sometimes, prompting the students to be precise initiates a critical evaluation, but not here:

D: Yeah, the peak. But this purple (*blue in Fig 2.2*) part’s like, it’s part of the shadow, but it’s like the lighter part. . .

I: Uh huh. How precisely do you think you can locate it? The end of it. Do you think you’re. . .

S: Um.

When prompted directly to think about the technology, the students demonstrate that they do have relevant knowledge, but they do not apply it to the edge-detection task:

I: Think it’s exactly there? And do you think if you got on the c(*omputer*)

D: (*interrupting*) Maybe not.

I: Maybe not?

D: No, because it depends on how many, how do you like, counts; how many. . . It depends on how many counts the computer counts for that color.

Diane seems to understand that colors correspond to data values or “counts,” although she has expressed it backwards, “counts . . . for that color.” In fact, the computer assigns colors for a range of counts. But then, without any consideration of the technology, Sam insists upon interpreting the dark region as the “whole shadow.” Referring to Fig. 2.2, he locates the shadow’s edges at the **transition between dark and bright color hues**:

S: I still think that’s the whole shadow right there.

D: Yeah. Yeah.

S: And it just starts there, and ends there. And this might be a little bit lighter (*pointing to purple region*), but it’s still part (*of the shadow*).

Analysis of the transcript does not reveal exactly why Sam did not apply relevant knowledge of the technology. One possible explanation is that his understanding of the mapping was too limited to be effectively applied. This suggests ways that the software could be redesigned to support Sam’s process of interpretation. For instance, a color key that showed the range of data values corresponding to

each color might have helped. (The software design described later in the paper has exactly this function: clicking on a color in a palette can show the range of values it encodes.)

Ben and Alice: Applying a Misunderstanding of the Technology. The final example illustrates how mistaken understanding of the technology may interfere with appropriate representational interpretations. Comparing the grayscale displays⁴ (Figs. 2.3 and 2.4), Ben judged the display with the sharper edge, the **sudden transition in brightness**, would be better for locating the edge than the less “defined” edge, the **gradual transition in brightness**. Apparently, Ben thought the image-processing tool can “define” an image as a camera can focus. Ironically, searching for a sharp edge loses all the referent information that is helpful in finding a good approximation of the edge.

B: I think that this is the clearest picture of it (*Fig. 2.3*), because this one (*Fig. 2.4*), it kind of goes on, but it's hard to tell where a definite edge is. But this one (*Fig. 2.3*) it's really clear. . .

A: Because there is more of a contrast (*Fig. 2.3*) for this one in that area (*pointing to edge of shadow*)

B. Right.

I: So, before you go on, can you think of any way the image processing tool on the computer can help you answer this question?

B: We could do something like what was done here, where you really define the edge of the shadow.

The idea of “focusing” (to show a sharp edge) is one of several mistaken understandings that some students developed as they used the image-processing tools. These mistaken understandings indicate that students are not thinking of the display as a representation of data. This is understandable, since the data file is almost invisible in the software. As shown in Fig. 3, the image-processing interface only shows one data value: (count = 2293) at location (x = 2, y = 5). This suggests one way to limit such mistaken understandings: make the data file more salient.

Summary Remarks

To summarize, the above examples are illustrative of our principles and suggest that the proposed

⁴The actual displays used by students more clearly showed a sharp, discontinuous edge for Figure 2.3 and a more gradually brightening edge for Figure 2.4.

expertise is a challenging yet achievable goal for students. Altogether, the examples illustrate students’ rich intuitive perceptual resources, where students attend to a variety of visual attributes. Table II lists the students’ interpretations of visual attributes. This is typical of the larger data corpus from which the examples of this section were drawn. Compared to the expert student, these include both correct and flawed interpretations. But taken as a whole, students’ many and varied interpretations of visual attributes are evidence of substantial interpretive capabilities.

The examples illustrate the making good use of interpretive capabilities depends upon considering representational aspects of the technology. Consequences for not considering representational aspects of the technology include:

- Students may reject or not identify potentially useful visual attributes
- Students may misinterpret visual attributes

The analysis also identifies some factors that prompt students to apply their knowledge of the technology:

- Explaining an interpretation
- Comparing representations
- Considering precision

Finally, the analysis identifies factors that may limit effective applications of knowledge of the technology:

Table II. Student’s Interpretation of Visual Attributes	
Brightness contrast: Transition from black to gray (Fig. 2.1)	So, where does the shadow end? It ends right there.
Black region (Fig. 2.1)	Points over here are obviously below the MIN.
Color hue, shade of gray (Fig. 2.2)	Colors spread out over a large number of counts. If you made the MIN and MAX (interval) smaller, you could make a better distinction.
Comparative contrast: Gradual transition in brightness (Fig. 2.4) Sudden transition in brightness (Fig. 2.3)	I think that this is the clearest picture, because there is more of a contrast.
Brightness contrast: Transition from dark hues (black and purple) to bright hues (Fig. 2.2)	that’s the whole shadow.

- Limited knowledge of the technology itself
- Mistaken understandings of the technology
- Limited or mistaken understandings of the referent (e.g., that shadows are not uniform zero data)

These findings have informed our design of the new visualization tool and activities. The final sections of this paper will describe activities where students use the new technology.

DESIGN OF TECHNOLOGY AND ACTIVITIES FOR LEARNING ABOUT REPRESENTATIONS

In the previous sections, we presented empirical work that motivated attention to representational aspects of the technology. The students were using an already existing image-processing tool. This section turns to the design phase, illustrating how we took those ideas (and our principles) into account in the design of some new technology and activities. As a reminder, our principles are:

1. Design for the best contact with the most important scientific ideas. In our case, this includes learning specifically about representation.
2. Build on students' strengths. In our case, this means building on their creative abilities at interpreting and (below) designing visual displays.
3. Design for activities that students find personally engaging.

Technology design is an extremely complex process, so we can only show illustrative aspects of what we did.

Consonant with principle 1, we describe our new software as a metarepresentational tool. While we will illustrate with astronomical images, the tool is meant to provide resources for building many kinds of representations of spatially distributed data. The tool is constructed as an extension to the Boxer computer environment (diSessa, Abelson, and Ploger, 1992). Boxer is a system with a broad range of facilities, including programming, in which developers, teachers and students can use, modify and extend provided resources. Because of Boxer's properties, what we produced should more properly be described as a tool kit (diSessa, 1997) than as a tool since it is easily modifiable and extendible. This is important

in providing technological support for student initiative, as in principle 3.

Basic Architecture

Consonant with a focus on activities, we began our design by creating not only a specification for the technical properties of the system, but also a database of activity types that we felt were representative of the range of things students and teachers should be able to do with our software. For these purposes, activities don't need to be spelled out in detail for we only need to make sure the software is technically capable of supporting such activities. Our list included standard image processing functions, as illustrated in the work above, but also much more general capabilities, such as dynamic processing of the data, creation of adjunct tools for directly manipulating and displaying data, and the ability to overlay graphical objects that interact with the data. (Such objects might be, for example, creatures walking on terrain represented by altitudes, or they might be annotations that help make the display comprehensible in representational terms). We alternated specifying technical components of the system and then checking their sufficiency to deal with our list of activities. Our final design database included full technical specification, the list of activities, and sketches of how each activity could be carried out with the software.

The basic element in our design is called a landscape box. The metaphor "landscape" arises from the fact that a canonical task for which one would use a landscape box is to show values of some attribute (temperature, altitude) over some terrain. In order to make representational features of the software highly salient, we took care to do justice to each of the three basic components of representation in landscapes. First, we have a computational layer of "information to represent." This layer is a two-dimensional array of elements, called patches, which correspond, generally, to places in space at which measurements (e.g., of light intensity) may be made. Figure 6 shows a simple numerical array of patches such as might be used for image processing. Figure 7 shows a single patch containing more complex data, in this case, a slot for altitude and slots for components of gradient (representing direction and magnitude of "uphill"). (In the future, we intend that slots may contain non-numerical data, for example, a "category" slot might indicate whether that portion of

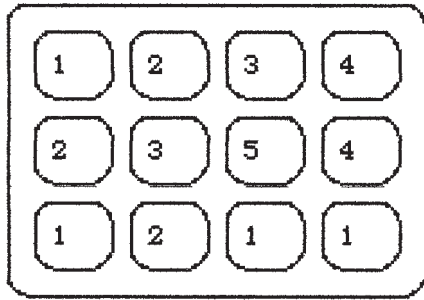


Fig. 6. The landscape representation of data (referent) is shown in a two-dimensional array. Each patch (small boxes) contains exactly one datum.

the landscape corresponds to land or water, city or countryside.)

The second aspect of our landscape system is the graphical representation of this data layer. In Boxer, all graphics boxes (of which landscapes are an example) have two “sides.” On one side, one sees a picture. By “flipping” the box, one sees the data that defines the picture. In the case of landscapes, the flip side of the graphical presentation is an adjustable display of the matrix of patch data. (In the current implementation, the display of data must be explicitly built. This compromise was accepted in order to reduce effort in programming before we felt empirically satisfied with the rest of the design.)

The third essential component of a representation is the mapping between the data and the graphical representation. In our preliminary design, we wanted students to write (or inspect and modify) simple programs that convert the data in a patch into a color. For reasons of efficiency, our first implementation does not allow that generality, but, instead, colors are selected automatically from a palette by linear interpolation between Minimum and Maximum data settings, similar to what we explained for image processing, above. In practice, we found this an excellent compromise. Landscapes worked fast

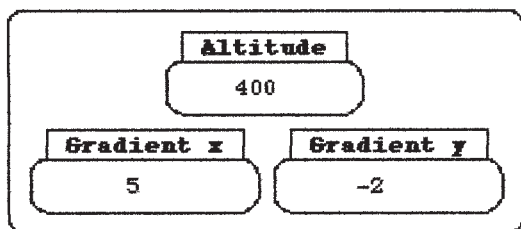


Fig. 7. A more complex patch of data, including multiple slots.

enough to be practical, yet students had plenty of freedom for their own creativity. Since the mapping part of the representation is somewhat more distributed (among the palette, MIN and MAX settings), and since the algorithm of color selection is invisible, we felt we had to augment activity focus on the mapping to compensate, which we describe below.

MIN and MAX are boxes with those names (Boxer variables) that also appear on the flip side of a landscape. Palettes, once again, appear on the flip side of a landscape, and have their own “elegant” (see below) representation. On the graphics side of a palette, one sees the sequence of colors in the palette. On the data side, one sees concretely the sequence of colors as they are represented as basic data objects in Boxer. Boxer color data objects are themselves graphics boxes that show a swatch of color, and their data flip sides contain three numbers, which represent the percentage of red, green and blue (RGB) in the color. Figure 8 (see color plate for color version) shows the graphics side of a palette, then the flip (data) side, and finally, the first two colors are flipped to show RGB percentages.

The reason we describe the palette design as elegant is that it uses very simple, standard Boxer features (e.g., the flip side of graphics boxes) to allow students and teachers complete access to the structure. Students can flip the palette, re-arrange, delete or add colors, or edit individual colors using only basic Boxer “cut and paste,” “flipping,” and similar ideas that were likely learned in early experience with Boxer. One can even write a simple program to generate palettes. Even though we wrote tools to make generating and modifying palettes easy, students still often worked directly and concretely on the palettes, themselves.

Compared to the image processing tool we used in formative work, this one gives much higher prominence to the data and the mapping, more comparable

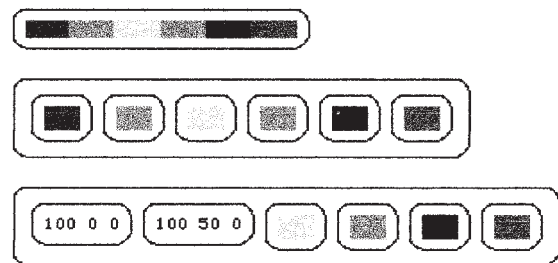


Fig. 8. A palette; the flipped (data side of the) palette; and showing the data from the first two colors. See Color Plate.

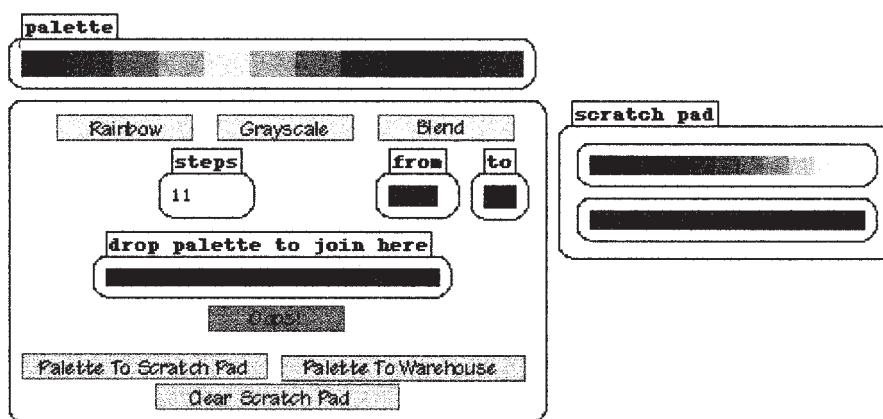


Fig. 9. A palette making tool. See Color Plate.

to the graphical representation. Students also have much more direct access to data and map and much more flexibility in changing them (e.g., palettes in the other software were simply one of many things one selects from global menus). The connection between a landscape and its data and mapping are a little more direct in our design in that one can see relevant parts of these inside (on the flip side of) the landscape itself, rather than in some not-obviously-connected part of the interface. We expected these properties to enhance both understanding of all three components of representation and also to provide more flexible and personal use. Flexibility means that teachers and materials developers, as well as students, can adapt the software more easily to particular things they want to do.⁵

More Tools

In designing the course in which we first intended to use landscapes, we decided to add some further tools to facilitate student activities that we hoped would be engaging and helpful. Figure 9 (color plate, color version) shows a palette-making tool. A formative study other than described above convinced us that designing and working with color palettes would

be a rich task for students (Azevedo, 1998). In order to facilitate working with palettes, the palette tool simplifies frequent tasks. The top row of buttons produces standard palettes, either “rainbow” (red, orange, yellow, green, . . .), gray-scale, or blended between two polar colors (from and to). Steps determines the number of colors that appear in the standard palettes. You can concatenate one palette onto the end of the existing one by dragging and dropping the palette-to-add into the drop palette to join here box. The Oops button undoes previous operations, and there is a scratch pad to save temporary palettes. Once you have a palette you like, you can press a button to save it in a global warehouse, which is available at any time for use with any landscape. Drag and drop is used to facilitate many operations. One can drag a palette onto a landscape to change the landscape’s palette. One can drag a color within a palette into a different position, or delete it by dragging it off the palette. Individual colors can also be dragged from a palette to any color box, or inserted into a palette with the reverse operation.

A second tool facilitated comparison of graphical presentation with underlying data. The “data peeker” shows a small portion of the underlying data while the landscape is still showing its graphical side.

Figure 10 shows a typical setup of tools to investigate astronomical images. The bracketed square in the middle of the landscape image shows the region selected for the data peeker; it can be dragged around in the image, and data is constantly updated. MIN and MAX are easily adjusted with slider controls, and a slice graph is shown. The “ball and stick” icon on the left of the graph can be dragged across the graph, showing the value of the graph at each abscissa while

⁵For example, since landscapes are both general and programmable, one could use them to represent compound data (such as selectively showing barometric pressure, temperature and wind over some terrain); dynamic changes in data (such as computing wind from some model of its relation to barometric pressure); alternate “overlay” representations (such as showing wind as vectors); and showing “object/environment” interactions (such as the effects of wind on a graphical airplane flying through a landscape).

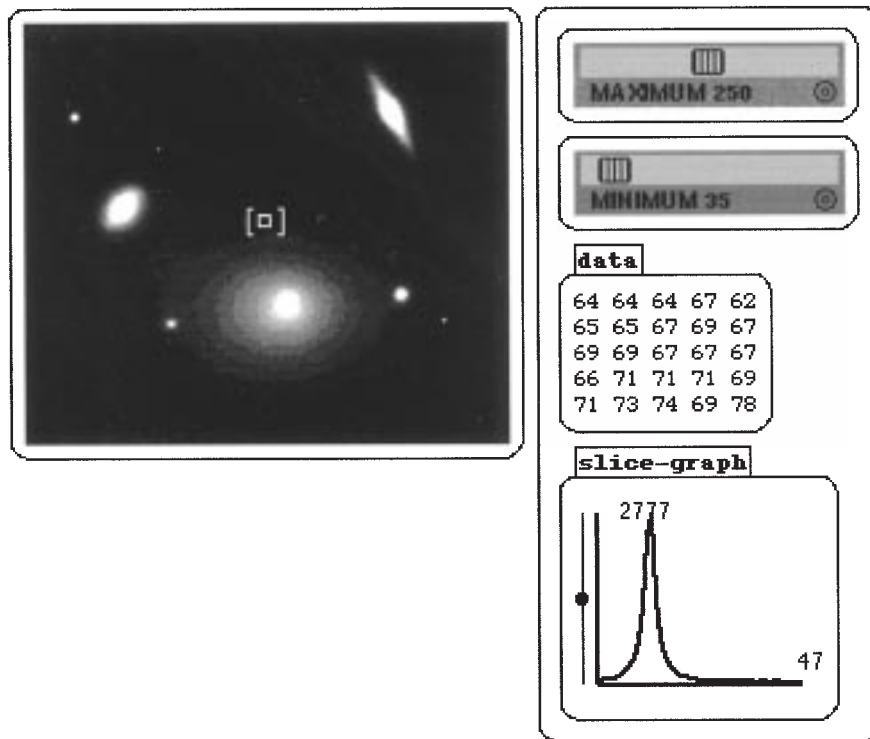


Fig. 10. Typical setup to investigate astronomical images.

crosshairs shows the location in the landscape from which the value is drawn.

Activities

Guided by our principles and previous empirical work, we designed the following sequence of assignments for students before they begin engaging astronomical tasks such as those we reported on above. Partly, these activities were designed to teach about representational aspects of the landscape tool. Partly, they were designed to engage students' creative abilities for interpreting and designing visual displays.

1. We started by asking students to design paper and pencil representations of a model physical landscape. Our experience has been that students invent a significant range of representational forms, and good discussions of the properties of these may be engaged. In addition, we have discovered that the idea of color-coding height almost always spontaneously emerges from this activity (Azevedo, 1998).
2. We capitalize on students' suggestion of color coding by asking them to select colors that do a good job showing altitude data in a given array of altitude numbers. We limit them to fewer colors than numbers to force the issue of grouping of data under a single color (resolution).
3. Following these paper and pencil activities, we introduce a simplified setup of a Boxer landscape, which had only a few patches and in which the map to color is explicitly represented as a program.
4. The first full-fledged use of the landscape system is an exercise in which students are given data in the form of a landscape representing altitudes of a fictitious archeological site in Egypt. The site contains two pyramids and an obelisk. The students must locate and measure height of the features of the site using only the graphical landscape representation. The task is especially difficult because only a small palette is allowed. Later in the activity, students are introduced to the data pecker in order to check their graphical work. Resolution is an important issue since, for example,

the smaller pyramid is not even visible with the first settings of MIN and MAX (the pyramid all fits within the range of values associated with one color). Skillful use of MIN and MAX settings is required.

5. Students take pictures of themselves with a simple digital camera. The data is imported into a landscape, and students are asked to do such things as bring out maximal details in their eyes and to color their hair green. In this case, they should be very familiar (in principle) with the data, and the task is somewhat reversed in that visual effect, rather than determining the data, is at issue.
6. In preparation for tasks such as examining images of the moon to determine relative heights of craters and mountains, students are given physical materials to form into “experimental landscapes.” The landscapes are then photographed and imported into Boxer landscapes, where students test their hypotheses about how to examine the data to determine relative height.

AN EXAMPLE

How well did our technology and activity design (and, implicitly, our principles) work? We cannot provide a scientifically sound evaluation in a small space. However, we want to illustrate the sorts of things we found in our first course using the above-described software and activities. The course was run during the summer of 1997 in the Berkeley Graduate School of Education Academic Talent Development Program. It involved 10 students, ranging from seventh to 11th grade, over six weeks, two sessions of three hours each week. The activities and software described above were used for a bit less than half the time of the course. We do not claim the students or time involved were “typical,” and hence we aim mainly to show what is possible.

Overall, we felt our students acquired an excellent sense of representational aspects of the computer tools, how they work and how to use them. Errors and difficulties of the sort we illustrated in transcripts above were rare and were not, as far as we could see, the result of any limitation or misunderstanding of the computer tools. On the other hand, we observed that some students continued to make errors typical of those documented in the literature concerning

standard representations, so we have no convincing evidence of MRC transfer, yet.

Concerning principle three, engagement, students made many creative uses of the tools beyond what we scripted into exercises. This often entailed substantial thinking about representational ideas, as illustrated by one student’s final project, described below. Amy’s use of the technology supported substantial thinking about resolution and about contours of constant value, a useful representational convention. We use this project also to indicate the general level of representational sophistication of students in the course. Amy showed many, if not all, of the attributes of our proposed expert student during her project.

Amy’s project entailed creating displays that would clearly show the difference between two images of the same patch of sky. The images were made three weeks apart. In the latter image, a supernova, an extremely bright exploding star, appears as a “new star” in one of the galaxies. Amy used the palette maker to mimic contour-lined representations she had seen in a scientific report. Figure 11 (color plate, color version) displays a research astronomer’s and one of Amy’s final representations, side by side. In Amy’s display, the yellow region in the galaxy distinctly represents the supernova on the right side of the galaxy. As she explained in her written report:

The advantages of . . . contours are that they show the change . . . in their context. . . They are easy to use and they show detail.

Amy’s report chronicles her thought process leading to her final representation. In particular, she was not satisfied with the first palette she made:

This palette didn’t go over too well. It lacked the details that I wanted to be shown, but did give me an idea of where I should go from here.

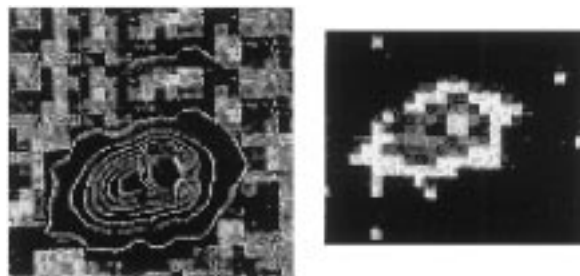


Fig. 11. Scientist’s and Amy’s Representations of Supernova 970.
See Color Plate.

Using the data-peeker tool, Amy established the range of light intensity in the galaxy:

Using the data-peeker tool, we are able to find the ranges of light intensity in the supernova and pre-supernova images. (In these images, we are focusing on the oval shape on the left side of the image . . . SN-970 is a real supernova. Wow!) Knowing the light intensity range makes using the MIN and MAX for contrast much easier.

Then, using the palette-maker tool over several hours, she designed many palettes and evaluated the displays she made with them. She was satisfied only when she created a display with as much detail as the data and technology would permit:

After playing around with the palettes some more, adding in new colors where they are needed, adding more steps. . . I developed a SNboxer-Contour that was tight down to the pixel. . .

Note that Amy was not provided any specific criteria for evaluating her displays. She developed her resolution criterion, “tight down to the pixel”—meaning showing the relevant information in every pixel—while designing and evaluating palettes. By varying the MIN and MAX settings, changing the sequence of colors in a palette, varying the number of colors in a palette, and even creating new color hues, she explored a wide range of ways to vary the mapping. Although, of course, we have not proven it here, we believe such sustained thinking will frequently result in deepening understandings of representations.

Amy was obviously enthusiastic about her project throughout. She explained that she was attracted to the project because she wanted to “work with something real.” However, it seemed evident to us that her interest was also substantially sustained by using the technology for designing palettes and evaluating displays.

SUMMARY

In this paper, we have tried to describe and illustrate a more than usually accountable practice of technology design for educational purposes. The process is iterative and is both empirically and theoretically grounded at each stage. Empirical grounding in cognitive studies of the use of technology establishes plausible learning goals, difficulties experienced by students, and clearer criteria for evaluating the technology. Theoretical grounding, in our case, amounts to establishing, refining, and testing (as much as possi-

ble) principles that articulate general commitments of designers. In addition, we have emphasized the importance of activities at all phases of design and evaluation. Activity design is part of technology design, and evaluating technology, we claim, should be in the context of relevant educational activities. Although this paper is about one type of software, we believe this kind of design can be valuable more broadly.

The particular design described here concerns the first generation of a new, meta-representational software tool. The design was based upon a general set of constructivist principles. The fundamental idea is that a learning technology and embedding activities should be designed to help students build on what they already know, including native capabilities and common sense knowledge. These principles were specialized, based upon empirical studies of students who used image-processing software for visual analysis of spatially distributed data. These studies revealed that student understandings and misunderstandings of the technology itself played a substantial role in their thinking.

Returning to our original question, what should students know about the technology? Our answer is that students should know about the representational aspects of the technology. This is embedded in our specialized design principles: design technology and activities to support thinking about the representational aspects of the technology. The specialized design principles and data from our study guided the design of new representational software. We provide evidence that such representational thinking will help students better utilize their intuitive perceptual capabilities for purposes of visual data analysis. More speculatively, we hope students will learn something about representations generally. We refer to this idea as meta-representational competence.

The main empirical part of this paper aimed at showing students’ strengths and weaknesses at interpreting visual displays. Students specifically attend to many important and informative visual attributes. And the study showed that at least some students can integrate these with understanding of the representational aspects of the technology so as to produce cogent interpretations of visually presented data. On the other hand, the students in our first study also showed limitations that we believed were due to weaknesses in the technology design—specifically in limited access to representational properties of it—and in learning activities that could highlight representational aspects of the software. Student weak-

nesses included attending only to the attributes of the display, rather than their representational import, inability to critically evaluate “evident” interpretations (e.g., the edge must be “right there,” at a discontinuity), inability to use knowledge of the technology for representational ends, and direct misunderstandings of how the technology worked and what it was capable of (e.g., sharp edges mean “focused” and provide the best evidence on where an edge is).

The second part of the paper illustrated how we used this data and our specialized principles in designing software that would, most particularly, make its own structure as a representational device salient. We also designed a series of activities that supplemented software design in:

- capitalizing on student strengths and interests
- opening up important (representational) understandings of the technology
- focusing, more generally, on important representational ideas, like resolution and arbitrariness (of color meaning and of boundaries between colors).
- highlighting empirically determined-to-be-difficult ideas
- facilitating student progress into extended, personally meaningful projects

These activities and software were incorporated into a summer course. Overall, we felt students in the course acquired an excellent sense of representational aspects of the computer tool. A description of one student’s final project illustrated how designing activities and software according to our principles can support sustained thinking about representations, productively using perceptual capabilities, and can engender a high level of engagement.

REFERENCES

- Asbell-Clarke, J. and Barclay, T. (1996). Discovering the scientist within. *Hands-On! (TERC)* 19(2): 4,5,17.
- Azavedo, F. S. (1998). Inventing mapping: Meta-representational competence for spatially distributed data. Paper presented at annual meeting of the American Education Research Association, San Diego, California.
- Barstow, D. and Berenfeld, B. (1996). Data Visualization as an Essential Component of Telecollaborative, Inquiry-based Science Learning. Paper presented at annual meeting of the American Education Research, New York.
- diSessa, A. A. (1987). Artificial worlds and real experience. In Lawler, R., and Yazdani, M., (Eds.), *Artificial Intelligence and Education*, Ablex, Norwood, New Jersey, pp. 55–77.
- diSessa, A. A., Hammer, D., Sherin, B., and Kolpakowski, T. (1991). Inventing graphing. *Journal of Mathematical Behavior* 10: 117–160.
- diSessa, A. A. (1992). Images of learning. In De Corte, E., Linn, M. C., Mandl, H., and Verschaffel, L. (Eds.), *Computer-Based Learning Environments and Problem Solving*, Springer-Verlag, Berlin, pp. 19–40.
- diSessa, A. A. (1995). The many faces of a computational medium. In diSessa, A., Hoyles, C., Noss, R., and Edwards, L., (Eds.), *Computers and Exploratory Learning*, Springer-Verlag, Berlin, pp. 337–359.
- diSessa, A. A. (1997). Open toolsets: New ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.), *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education, Vol. 1*. Lahti, Finland, pp. 47–62.
- Friedman, J. S. (1996). Image Processing in a Science Classroom: A Constructivist Perspective on the Role of Prior Knowledge. Paper presented at annual meeting of the American Education Research, New York.
- Gordin, D. N., and Pea, R. D. (1995). Prospects for scientific visualization as an educational technology. *Journal of the Learning Sciences* 4(3): 249–279.
- Greenberg, R., Kolvoord, R. A., Magisos, M., Strom, R. G., and Croft, S. (1993). Image processing for teaching. *Journal of the Learning Sciences* 2: 469–480.
- Hancock, C. (1995). The medium and the curriculum: Reflections on transparent tools and tacit mathematics. In diSessa, A., Hoyles, C., Noss, R., and Edwards, L. (Eds.), *Computers and Exploratory Learning*, Springer-Verlag, Berlin, pp. 221–240.
- Leinhardt, G., Zaslavsky, O., and Stein, M. K. Functions, graphs, and graphing: Tasks, learning, and teaching. *Review of Educational Research* 60(1): 1–64.
- Lillesand, T. M. and Kiefer, R. W. (1994). *Remote Sensing and Image Interpretation*, Chapter 1. John Wiley and Sons, New York.
- Palmer, S. (1977). Fundamental aspects of cognitive representation. In Rosch, E., and Lloyd, B. B. (Eds.), *Cognition and Categorization*, Erlbaum, Hillsdale, New Jersey, pp. 259–303.
- Sherin, B. L. (1998). The elements of representational design. Paper presented at the annual meeting of the American Education Research Association, Chicago.