genuine masterpiece, a marvelous interactive program accompanied by a marvelous manual that gracefully explains the straight-forward computer mechanics involved and, more importantly, shows dynamic data analysis in action and chronicles the development of such displays. Alas, even in the MacSpin book, the graphics have the jaggies and murkies, too.

ADDITIONAL REFERENCES

AUDOUZE, J. and ISRAËL, G., eds. (1985). *The Cambridge Atlas of Astronomy.* Cambridge Univ. Press, Cambridge.

MAUNDER, E. (1904). Note on the distribution of sun-spots in heliographic latitude, 1874 to 1902. *Royal Astron. Soc. Monthly Notices* **64** 747–761.

TUFTE, E. R. (1988). *Envisioning Information.* Graphics Press, Cheshire, Conn.

Rejoinder

Richard A. Becker, William S. Cleveland and Allan R. Wilks

We would like to thank the discussants for their interesting comments. Our responses cover six areas: implementation of the methods of the paper, presentation of the graphs, the underlying software, the computing environment, brushing and rotation.

Since writing the paper we have studied dynamic methods in a color graphics environment—a Silicon Graphics IRIS 2400T workstation. Most of the original paper is based on an implementation of methods in a distributed processing system with an AT&T Teletype 5620 graphics terminal, which is monochrome. Our responses here will reflect more of the experience with the IRIS implementation.

1. IMPLEMENTATION

Experimental vs. Tested: Field Testing

Comments by Huber and Eddy made us realize that one deficiency in the paper is an explicit statement about whether the methods in Sections 2.1 to 2.6 are experimental or well tested. We have examined a large number of dynamic methods by field testing, which will be described shortly. With one exception, the methods of Sections 2.1 to 2.6 are those that we tested and judged to be useful tools for data analysis. (The one exception is advanced strategies for rotation control, which we only reported but did not test.) We strongly urge software developers to implement these methods in their software systems. Wainer is quite right—we tried many other ideas that did not work out.

Field testing a method means using it on a variety of data sets including those where data analysis is in progress. At the moment, because the amount of theory about data display is small, extensive field testing is the only way to effectively judge a graphical method. Armchair thinking is not enough. In 1982, Tukey (1987b) wrote the following about the development of a graphical method:

- now we try it [the graphical method] on diversified data—trying to understand when its performance is less than adequate,
- and then we try to understand something of what modifications would help,
- and then we try the modifications,
- and then we repeat the last 3 steps as needed!

There is no substitute for adequate iteration. *Such iteration is the original developer's obligation.*

It is particularly important to try out methods in settings where people are attempting to learn about the world from the data, and where the methodology is a means to an end. One cannot fully assess a method by using just old data sets no longer of interest to anybody and dredged up just to test the method. In the paper, though, we used familiar or easy-to-convey data sets because of space limitations.

NIH Means Not Implemented Here

In his first sentence, Eddy refers to "bias in favor of methods they and their colleagues have developed." "Developed" should be changed to "implemented." Many of the methods of the paper were invented by us. Many were invented elsewhere; we hope the extensive citations and bibliography make this clear. But except for the advanced strategies of Section 2.6, we discussed only methods that we implemented. This is as it must be. We could not write with much insight about methods that we did not field test, and we could not field test a method that we did not implement.

2. PRESENTATION

Excitement

We quite agree with Tukey that "paper versions of screens with highlighted points are rather weak and wan" and with Wainer that the static pages of a journal do not convey the excitement of dynamic graphics. We feel badly about this, but we can do something about it. If the IMS is willing to distribute it, we will provide a videotape that demonstrates the methods. (Perhaps we can call it the DYNAGRAFIX tape.)

ACM SIGGRAPH, the premier computer graphics group, now publishes the *SIGGRAPH Video Review* to provide a means of disseminating a wide variety of computer graphics. Perhaps it is time that statisticians did something along these lines, too.

The Mud

It is true that the scatterplot matrices of our examples do not contain residuals or transformed data, but this was not meant to imply that we do not think graphing them is of substantial value. To miss the value would be to miss some of John Tukey's most creative and important ideas about data display. (See Cleveland (1985) for a graph of (brain weight)—2/3(body weight).) In fact, the importance of graphing such derived variables is precisely what stimulated the first paragraph of Section 3.5. It is important to use these dynamic techniques within the context of a good data analysis system. Rather than burden the dynamic techniques with basic statistical computations, we expect that these will be carried out in a powerful data analysis environment.

Panels of the Scatterplot Matrix

The table vs. graph description of Tukey is nicely put. We doubt, though, whether having the empty panels running along the 45° line, or perpendicular to it, makes an appreciable difference.

Slopes

Tukey makes an excellent point about slopes. But we cannot, of course, avoid slope judgments altogether, and optimizing them when we cannot avoid them makes good sense. We also note that as soon as we plot the slopes of the sunspots we are quite likely to find ourselves studying the slopes of the slopes, so we would apply the 45° principle to such a graph.

Scales

Tick marks and tick mark labels in the empty panels, as Tukey suggests, are clearly needed.

Resolution

Huber has made important points about resolution on the scatterplot matrix and has given a sensible suggestion. In our implementation of brushing on the IRIS, the data analyst can use a pull-down menu to delete a variable shown on the scatterplot matrix or to add one not shown. For data analyses where resolution of the full matrix is a problem, we often spend a large fraction of the analysis time studying subsets of the variables.

Texturing

We look forward to seeing the work of Tukey, Tukey and Veitch.

Dynamic or High Interaction

Huber is right about the term "dynamic." It does not suggest, as it ought to, that the methodology we describe is more than "data moving," as Tufte nicely puts it, and is really data "being moved." Early versions of our paper used "high interaction," a more appropriate word, but as time went along we switched to the more conventional "dynamic" to avoid jargon. Meaning gave way ever so slightly to style. As distortions of this ilk go, it is not so bad.

Jaggedness Is Not Particularly Important

Tufte's concerns about jaggedness are appropriate and relatively easy to address in static, printed graphics. However, our paper concerns images that appear in real time on a computer display screen. Although we could readily produce higher quality plots (on a phototypesetter, for example), actual screen images are shown in the paper. These images show the jaggedness inherent in most current hardware.

A graphical display in science and technology should be judged mainly in terms of whether it conveys quantitative information and helps us learn about the world. It is unreasonable to produce displays that have a high quality appearance if the costs are too great; there is the cost in dollars to purchase and program extremely high resolution hardware, there is the cost of reduced information flow in a dynamic method, and there is the cost in analyst time in trying to get the best appearance of a graph. Because jaggedness does not generally interfere with our understanding of the data, it is relatively unimportant in interactive graphics; even a small increase in cost deters us from taking action when the benefit is small.

3. SOFTWARE

Simplicity of Software

Huber mentions that dynamic graphics has "a core of simple-minded, extremely useful techniques." We agree wholeheartedly. One of the difficult tasks in designing software is to find the core ideas and to keep them from being hidden by a clutter of options and features. This is the central idea of software tools as exemplified by the UNIX operating system (Ritchie and Thompson, 1978). Another interesting idea is that of the lifeboat principle in designing software: for each feature that is added, one must be thrown overboard. This leads to simple, yet general, software design.

Software Secrets

Wainer seems to imply that dynamic graphics programming is somehow beyond the realm of ordinary people. We insist it is not. It is a lot of hard work, and at present the effort put into dynamic graphics on one display device does not carry over to other devices. Nevertheless, anyone with sufficient desire and willingness to work should be able to experiment with these techniques.

Arithmetic

We agree with Tukey that special arithmetic can provide advantages over floating point arithmetic. Even with the availability of floating point coprocessors, integer arithmetic is likely to be faster; ultimately integers are used to address the screen on the graphics device.

4. ENVIRONMENTS

Distributed Model

It is interesting to contrast Eddy's statement that "It now seems clear that the next step is a distributed approach to interactive graphics" with Huber's "The distributed processor model is not only at a design disadvantage, but it also creates a software maintenance nightmare." We tend to side with Eddy; software design can be more subtle in the distributed approach, but the payoff is high.

Windows

Huber points out that modern workstations with multiple windows allow a data analyst to become thoroughly confused about what has been covered in an analysis session. Of course, the answer is not to outlaw windows and force the analyst to think in the traditional one-step-at-a-time manner, but to make use of the computer to monitor and record the analyst's actions. Later, the computer can use these audit trails to produce a well organized file describing what was done, as in a laboratory notebook. Much work needs to be done in this arena.

5. BRUSHING

Brush Shape

It is true that a rectangular brush makes computations easier. However, we did not pick a rectangular brush for computational convenience. To condition on one or two variables, the analyst must select rectangular regions whose sides are parallel to the coordinate axes. The brush is rectangular because that makes such selection easy and because conditioning is important and frequently used. When painting in the lasting mode, the brush shape has little effect on ease of use—we have used diamond-shaped brushes and could have implemented other shapes.

Roping

Roping is a natural idea—if given a scatterplot on a piece of paper, most of us would have little trouble using a pencil to draw a curve around a cluster. Roping is an analogue to this. Past implementations of roping did not produce a pencil-like instrument, but rather gave a pin-and-string method in which a polygon was formed by indicating vertices with a pointing device (Newton, 1978).

A roping implementation would have been within our capabilities. However, there are other problems with roping. There is no mechanism for unroping if we should accidentally rope-in more or less than desired. We can envision the need for a rope editor and much added complexity.

Highlighting

The highlighting issue, as Tukey has pointed out, is an important one. Plotting symbols of different colors provides the best highlighting of all. This, of course, was already known from experiences with static graphical methods (Cleveland and McGill, 1984).

Automatic Deletion

Fowlkes' operation was of the form "show me a normal probability plot with this data point removed from the sample." There would have been no reason to break it into two operations. However, in other settings, such as brushing, we are in agreement with Huber that deletion and redrawing should be separate operations. In fact, this is what we did in our brushing implementation on the 5620.

Undeleting

In Section 2.4 we describe shadow highlight. Although we did not say so explicitly in the paper, for our implementation on the Teletype 5620 the data analyst can also switch from the delete operation to a shadow delete operation; points that were previously deleted are turned on and points that were previously not deleted are turned off.

6. ROTATION

Control

We have to disagree with Huber on rotation control. We found method three (with stereo viewing) to be an extremely effective data analytic method. It provides delicate and accurate control that allows the data analyst to quickly get to a desired orientation of the point cloud. Also, method one is useful when studying how one variable depends on two others; it is very helpful to rotate about the axis of the dependent variable to help us keep the dependence structure firmly fixed in our minds. Of course, we could always use method two to do this by getting the dependent variable axis lined up with one of the screen axes, but this is a nuisance. In fact, having seven rather than three control selections adds very little additional burden to the system.

Perspective

Stereo viewing, as we stated in the paper, is an important enhancement. Without stereo, point clouds lose their three-dimensional feeling as soon as rotation stops. In our IRIS implementation, the routines that draw a stereo view also provide perspective and control of the distance to the center of the point cloud. At the moment we agree with the statement of Tukey that viewing from an infinite distance (i.e. no perspective) is sufficient, but this issue needs more study.

Dreibein

We have had no experience with a dreibein and so, have little to offer. We only wonder whether, as described by Tukey, the portrayal of more than three axes should be called a "vielbein."

ADDITIONAL REFERENCES

CLEVELAND, W. S. and MCGILL, R. (1984) The many faces of a scatterplot. *J. Amer. Statist. Assoc.* **79** 807–822.

RITCHIE, D. M. and THOMPSON, K. (1978). The UNIX time-sharing system. *Bell System Technical J.* **57** 1905–1946.

TUKEY, J. W. (1987b). Thoughts on the evolution of dynamic graphics for data modification display. In *The Collected Works of John W. Tukey* (W. S. Cleveland, ed.) **5**. Wadsworth, Pacific Grove, Calif.