SLAC – PUB – 4254
March 1987
M

AN INTRODUCTION TO REAL TIME GRAPHICAL TECHNIQUES FOR ANALYZING MULTIVARIATE DATA

JEROME H. FRIEDMAN, JOHN A. MCDONALD AND WERNER STUETZLE
Stanford Linear Accelerator Center
and
Department of Statistics
*Stanford University, Stanford, California, 94305**

ABSTRACT

Orion I is a graphic system used to study applications of computer graphics – especially interactive motion graphics – in statistics. Orion I is the newest of a family of "Prim" systems whose most striking common feature is the use of real-time motion graphics to display three-dimensional scatterplots. Orion I differs from earlier Prim systems through the use of modern and relatively inexpensive raster graphics and microprocessor technology. It also delivers more computing power to its user; Orion I can perform more sophisticated real-time computations than were possible on previous such systems. We demonstrate some of Orion I's capabilities in our film: *Exploring Data with Orion I.*

Invited talk presented at the Asilomar Computing in
High Energy Physics Conference, Feb. 2-6, 1987,
Asilomar State Beach, California

---

* This research was supported in part by the Department of Energy under contracts DE-AC03-76SF00515 and DE-AT03-81-ER10843, the Office of Naval Research under contract ONR N00014-81-K-0340, and the U.S. Army Research Office under contract DAAG29-82-K-0056.

† Also published as Dept. of Statistics Orion 009, March 1982.

systems through the use of modern and relatively inexpensive raster graphics and microprocessor technology. It also delivers more computing power to its user; Orion I can perform more sophisticated real-time computations than were possible on previous such systems. We demonstrate some of Orion I's capabilities in our film: "Exploring data with Orion I".

PREFACE

This paper accompanies a film that demonstrates some programs written for the Orion I workstation. Orion I is an experimental computer graphics system, built in the Computation Research Group at the Stanford Linear Accelerator Center (SLAC) in 1980-81. It is used to develop applications of interactive graphics to data analysis.

We intend this paper for an audience familiar with computer graphics, but not necessarily with statistics. We hope that it provides some perspective on the place of interactive graphics in statistics and an introduction to several areas of current research.

1. GOALS

1.1 DESCRIPTION AND INFERENCE

The goal of statistics is to analyze data. A data set consists of, say, n observations. For each observation, we measure p variables. It will be a useful idealization to assume that the values each variable takes on are reasonably represented by real, or floating point, numbers. Then each observation can be thought of as a vector in a p-dimensional real vector space. A data set is then a set of vectors. This type of data is typical of the field of statistics known as 'multivariate analysis'.

How we analyze a given data set is very dependent on context. Sometimes we are given data that arises from an experiment designed to answer a particular question. In

this case, we may have a great deal of prior knowledge about our data, and have confidence that we know what to expect in it. This is a situation in which statistical inference may be appropriate. An inference is a generalization from a given data set to some larger population (real or hypothetical) from which the data set is presumed to be a sample.

Often, however, we are given data to analyze about which we have very little prior knowledge. The goal then is to "just look at the data and see what is going on". Statisticians call this description or exploratory data analysis [12,10]. We aim to describe features of the particular data set at hand and not draw (formal) conclusions about larger populations. There are two parts to description: exploration, to discover interesting features of our data, and summary, to report what we have discovered.

In order to make inferences, we need a probability model of the process by which our data set is sampled from the larger population. Our inferences will only be reasonable if the probability model is a reasonable approximation of the way the data is generated. In order to construct a reasonable probability model, we must have a great deal of prior knowledge about the data. A common and serious flaw in standard statistical practice is the use of inappropriate models.

In classical multivariate analysis, one typically assumes that observations are independent samples from a multivariate Gaussian distribution. The Gaussian distribution models only ellipsoidal shaped point clouds; it is inappropriate when the data set seems to come from a distribution that has more a complicated shape. In our experience, most real data sets, especially those with many (more than three) variables, have some features that are clearly not Gaussian. The Gaussian distribution, and other very similar models that characterize classical multivariate analysis, have received a great deal of attention from theoretical statisticians. This interest is not because they are reasonable models of the generation of any data. Rather, the Gaussian distribution is favored because it is easy to analyze mathematicallly.

Description is a more primitive, and fundamental problem than inference. If inference is to be done successfully, then we must use good models. In order to get enough prior knowledge about a problem to construct a reasonable model, we must, at some time, look at data.

1.2 THE VALUE OF INTERACTIVE GRAPHICS

In statistics, graphical methods are used primarily for description. Our research, which emphasizes interaction and real-time motion, concentrates on the exploratory part of description. We want to develop methods that allow us to discover, understand, and summarize the multivariate "structure" of our data set. Since, in general, we have little reliable prior knowledge about our data, we are especially interested in methods that allow us to discover unanticipated kinds of structure in data. We also want to be able to resond to unexpected circumstances in an intelligent way.

This is where interactive graphics has much to contribute. First of all, computer graphics can quickly expose different views of data to human perception, so that the data analysist can detect many kinds of patterns in the data. The analysist can interpret apparent patterns using his knowledge of the context in which the data arose. Also, the analysist can select directions for further exploration using results so far.

Thus, interactive graphical methods combine human talents for perception of patterns and judgement using the full context of a problem with a machine's ability to do rapid and accurate computation.

1.3 AN EXAMPLE OF STRUCTURE

We have intentionally left "structure" undefined. What we mean by "structure" is any apparent pattern or interesting feature in a graphical (or numerical) description of data. Structure is a subjective perception. Quantitative measures of specific aspects of structure can be very useful, but we need to avoid too restrictive definitions.

In developing new graphical methods, it is useful to have some examples of structure in mind. We discuss next a simple kind of structure that that is illustrated in our film: "Exploring data with Orion I".

A typical, and simple, thing that we would like to know about a set of data is the following: Does the data set naturally seperate into groups or clusters?

In the film, we look at a data set presented by Harrison and Rubinfeld[9,1]. They measured 14 variables for each of 506 census tracts in the Boston Standard Metropolitan Statistical Area. They were interested in examining the dependance of housing price (represented by the median value of owner-occupied houses) on air pollution (represented by nitrogen oxide concentration). The remaining 12 variables measured other quantities thought to influence housing prices, such as crime rate, average number of rooms per house, etc.

Harrison and Rubinfeld tried to determine the dependance of housing price on pollution by forming a prediction rule (a linear regression) for median housing value as a function of nitrogen oxide concentration and the other 12 variables. The effect of pollution alone on housing price was presumed to be reflected in the partial dependance of the prediction rule on nitrogen oxide.

In the exploration movie, we do not consider explicitly the dependance of housing price on pollution. We concentrate instead on clustering.

Before fitting any prediction rule, it is natural to consider whether it is appropriate to fit one rule for all the data. If the data set seperated in a natural way into distinct and internally homogeneous groups (or clusters) then we would want to consider fitting different rules for each group.

Statisticians have developed many algorithms for clustering data. These algorithms usually rely on a notion of distance in the data space to partition a data set into isolated clumps. Observations are considered similar if they are close together and dissimilar if they are far apart. A cluster is a group of points that are close to each other and far from any other points.

With Orion I, we can use other criteria besides seperation to partition a data set. In particular, we can use subjective perception of patterns to define natural groupings. A data set may naturally divide into two groups, which follow clearly distinct patterns. Yet the difference between the groups may not be easily summarized by any natural measure of distance.

In the film, we show how the Harrsion-Rubinfeld data can be divided naturally into several groups. The major division turns out to be between urban and suburban-rural census tracts. However, the urban and suburban tracts do not form isolated clumps. Instead, in certain views, they lie in intersecting, perpendicular planes. Because the planes intersect, the groups are not isolated in any distance measure. But the seperation in the two groups is obvious when seen.

2. METHODS

The really challenging problem in multivariate data analysis is to discover and understand structure involving more than three variables at once.

Suppose our data set involves only a single variable. Then we can see most of what we want with a histogram.

We look at two dimensional data with a conventional scatterplot. In conventional scatterplots, observations are represented by points, which are plotted at horizontal and vertical positions corresponding to the values of two variables.

To look at three dimensional data, we draw a three dimensional version of the scatterplot. Real-time motion graphics makes it possible to draw pictures that appear three dimensional. We subject the data to repeated small rotations and display the projection of the rotated data as points on the two dimensional screen. If we can compute and display rotations fast enough (>10 per second) then we get an illusion of continous motion. Apparent parallax in the motion of the points provides a convincing and accurate perception of the shape of the point cloud in the three dimensional space.

In a conventional scatterplot, it is easy to read off aproximate values of the two variables for any particular point. In moving, three dimensional scatterplots estimating the values of the three variables for any particular point is difficult, if not impossible. However, perceiving the shape of the point cloud is easy. Because we are interesting in detecting patterns, we are much more interested in overall shape than in individual points.

There is no completely satisfactory method that lets us look at more than three variables at a time. Three basic approaches to many (more than three) dimensional structure are being studied with Orion I. They are:

1. Higher dimensional views: we try to represent as many variables as possible in a single picture.

2. Projection Pursuit: we try to find a low dimensional picture that captures the structure in the many dimensional data space.

3. Multiple Views: we look at several low dimensional views simultaneously; by making connections between the low dimensional views we hope to see higher dimensional structure.

2.1 HIGHER DIMENSIONAL VIEWS

One way to see high dimensional structure is to try to invent pictures that show as many dimensions at a time as possible.

One of the simplest ways to add dimensions to a picture is through color. We start with a three-dimensional scatterplot. We can add a fourth variable to the picture by giving each point in the scatterplot a color that depends on the value of a fourth variable. With an appropriately chosen color spectrum, we can easily see simple or gross dependence of the fourth variable on position in the three dimension space. Our ability to perceive distinctions in color does not compare to our ability to perceive position in space; we should expect to miss subtle or complicated relationships between a color variable and three position variables. Color works best for a variable that takes on only a small number of discrete values.

There are many more tricks that let us add dimension to a picture. For example, we can represent each observation in the scatterplot by a circle, rather than by a simple point. The radius of the circle can depend on the value of a fifth variable.

In a simple scatterplot, observations are represented by featureless points. We add dimension to the picture by

replacing points with objects that have features, such as color, size, and shape. These features can be used to represent variables in addition to those represented by a point's position in the scatterplot. These "featurefull" objects, sometime called glyphs, are well known [8].

With each new dimension, the glyphs become more complicated and the picture becomes more difficult to interpret. We need more experience to determine which are good ways of adding dimension to glyphs and to understand the limitations of each method.

2.2 PROJECTION PURSUIT

The basic problem with adding dimension to a single view is that the picture quickly becomes impossible to understand. An alternative is to restrict our picture to a few (3 or less) dimensions and then try to find low dimensional pictures that capture interesting aspects of the multivariate structure in our data. This is the basic idea of projection pursuit. [4,5]

In general, we could consider any mapping from the many dimensional data space to to a low dimensional picture. The projection pursuit methods that have been developed so far restrict the mappings considered to orthogonal projections of the data onto 1, 2, or 3 dimensional subspaces of the data space. More general versions of projection pursuit would include methods similar to multidimensional scaling [8].

The original projection pursuit algorithm [7] used a numerical optimizer to search for 1 or 2 dimensional projections that maximized a "clottedness" index, which was intended as a measure of interesting structure.

Automatic projection pursuit methods have been developed for more well defined problems: non-parametric regression, non-parametric classification, and non-parametric density estimation. In these problems, we build up a model that summarizes the apparant dependance in our data set of a response on some predictors. Interesting views are those in which the summary best fits or explains the response.

We are, of course, not restricted to a single low dimensional view. Models of a response will usually be constructed from several views.

Interactive versions of the automatic projection pursuit methods are being developed for Orion I. The system allows a user to manually immitate the Rosenbrock search strategy [11] used by the numerical optimizer in the automatic versions. However, a human being can search to optimize subjective criteria, using perception and judgement, instead of having to rely on a single, numerical measure of what constitutes an interesting view.

2.3 MULTIPLE VIEWS

This approach is inspired by M and N plots of Diaconis and Friedman [2]. A two and two plot is one kind of M and N plot; two and two plots are used to display four-dimensional data. To make a two and two plot, we draw two two-dimensional scatterplots side by side; the scatterplot show different pairs of variables. We then connect corresponding points in the two scatterplots by lines. To get a picture that is not confusing, we will often not draw all the lines. Diaconis and Friedman give an algorithm for deciding which lines to draw.

Our idea is a modification of the above. Instead of connecting corresponding points by lines, we draw corresponding points in the same color.

Briefly, this is how the program works. On the screen there are two scatterplots, side by side, showing four variables. There is also a cursor on the screen in one of the two scatterplots. The scatterplot that the cursor is in is the active scatterplot. We can move the cursor by moving the trackerball. Points near the cursor in the active scatterplot are red. Points at an intermediate distance from the cursor are purple. Points far from the cursor are blue. Points in the non-active scatterplot are given the same color as the corresponding point in the active scatterplot. The colors are continuously updated as the cursor is moved. We can also move the cursor from one scatterplot to the other, changing which scatterplot is active.

Using color instead of line segments to connect points has a disadvantage; it is not as precise at showing us the connection between a pair of points representing the same observation. However, we are not usually very interested in single observations. More often, we want to see how a region in one scatterplot maps into the other scatterplot; the combination of local coloring and the moveable cursor is a good way of seeing regional relationships in the two scatterplots.

Another advantage of color over line segments is that it is possible to look at more than two scatterplots at once. Connecting corresponding points with line segments in more than two scatterplots at a time would produce a hopelessly confusing picture. With color, on the other hand, it is no more difficult to look at three or more scatterplots at once than it is to look at two.

3. MACHINERY

3.1 PRIM-9

Orion I is the youngest descendant of a graphics system called Prim-9, which was built at SLAC in 1974 [4]. Prim-9 was used to explore up to 9 dimensional data. It used real time motion to display three dimensional scatterplots. Through a combination of projection and rotation, a user of Prim-9 could view an arbitrary three dimensional subspace of the 9 dimensional data. Isolation and masking were used to divide a data set into subsets.

The computing for Prim-9 was done in a large mainframe (IBM 360/91) and used a significant part of the mainframe's capacity. A Varian minicomputer was kept busy transferring data to the IDIIOM vector drawing display. The whole system, including the 360/91, cost millions of dollars. The part devoted exclusively to graphics cost several hundreds of thousands of dollars.

3.2 OTHER PRIMS

Successors to Prim-9 were built at the Swiss Federal Institute of Technology in 1978 (Prim-S) and at Harvard in 1979-80 (Prim-H). [3]

Prim-H improves on Prim-9 in both hardware and software. It is based on a VAX 11/780 "midi-" computer and an Evans and Sutherland Picture System 2 (a vector drawing display). It incorporates a flexible statistical package (ISP). However, the system has a price tag still over several hundreds of thousands of dollars. The VAX is shared with perhaps two dozen other users. Computation for rotations is done by hardware in the Evans and Sutherland. No demanding computation can be done on the VAX in real-time.

3.3 ORION I

There are two ways in which the Orion I hardware is a substantial improvement over previous Prim systems: price and computing power. The total cost for hardware in Orion I is less than $30,000. The computing power is equivalent to that of a large mainframe computer (say one half of an IBM 370/168) and is devoted to a single user. The hardware is described in detail by Friedman and Stuetzle [6].

The basic requirement for real time motion graphics is the abiity to compute and draw new pictures fast enough to give the illusion of continuous motion. Five pictures per second is a barely aceptable rate. Ten to twenty times per second gives smoother motion and more natural response for interaction with a user.

Orion I and the earlier Prims use real time motion to display three dimensional scatterplots. We can view three dimensional objects on a two dimensional display by continuously rotating the objects in the three dimensional space and displaying the moving projection of the object onto the screen. In a scatterplot, the object we want to look at is a cloud of points. A typical point cloud will contain from 100 to 1000 points. So our hardware must be able to execute the viewing transformation, which is basically a multiplication by a 3x3 rotation matrix, on up to 1000 3-vectors ten times per second. The system must also be able to erase and draw 1000 points ten times per second.

The important parts of the hardware are:

1. a SUN microcomputer, based on the Motorola MC68000 microprocessor. The SUN microcomputer is a MULTIBUS board developed by the Stanford Computer Science Department for Stanford University Network. It is the master processor of the Orion I system. It controls the action of the other parts of the system and handles the interaction with the user. The SUN is programmed mostly in Pascal; a few critical routines for picture drawing are in 68000 assembly language.

2. a Lexidata 3400 raster graphics frame buffer, which stores and displays the current picture. The picture is determined by a "raster" of 1280 (horizontal) by 1024 (vertical) colored dots, called pixels. There are 8 bits of memory for each pixel which, through a look up table, determine the setting (from 0 to 255) of each of the three color guns (red, green, blue). Thus, at any time, there may be 256 different colors on the screen, from a potential palette of 2**24. The Lexidata 3400 contains a microprocessor that is used for drawing vectors, circles, and characters. Color raster graphics devices like the Lexidata are cheaper and more flexible than the black and white line drawing displays used in earlier Prim systems.

3. an arithmetic processor called a 168/E. The 168/E is basically an IBM 370/168 cpu without channels and interrupt capabilities. It was developed by SLAC engineers for the processing of particle physics data and has about half the speed of the true 370/168. Because it has no input/output facilities, it is strictly a slave processor. The 168/E serves us as a flexible floating point or array processor. It is programmed in FORTRAN.

A programmer of Orion I works on:

4. the host, an IBM 3081 mainframe. The host is used only for software development and long term data storage. Programs are edited and cross-compiled or cross-assembled on the 3081. Data sets are also prepared on the 3081. Programs and data are downloaded to the SUN board and the 168/E through a high speed serial interface that connects the MULTIBUS and the 3081 (by emulating an IBM 3277 local terminal). Strictly speaking, the 3081 is not considered a part

of the Orion I workstation; it does not have an active role in any of the interactive graphics on the Orion I.

A user of Orion I deals most of the time with:

5. a 19 inch color monitor that displays the current picture.

6. a trackerball, which is a hard plastic ball about 3 inches in diameter set into a metal box so that the top of the ball sticks out. The ball can be easily rotated by hand. The ball sends two coordinates to the SUN computer. In Orion I, the trackerball provides the angles of a rotation; the apparent motion of a point cloud on the screen mimics the motion of the trackerball under a user's hand. The trackerball has six switches, which are used for discrete input to programs.

The part of the system that currently limits what can be done in real time is the Lexidata frame buffer. Its speed of erasing and redrawing pictures does not keep up with the computational speed of the SUN board and the 168/E. In fact, the 168/E can execute our most demanding real time computation, a sophisticated smoothing algorithm, at much more than ten times a second. This smoothing algorithm [5] is much more demanding than simple rotation.

It is also worth noting that rotations can be done in real time by the SUN board alone. Thus a system could be built with all the capabilities of earlier Prim systems using only a SUN board or some other 68000 based microcomputer and without a 168/E. Such systems are now (Feb. 1982) commercially available for about $25,000 for basic hardware and about $40,000 for a complete, stand alone system with hard disk, UNIX-like operating system, resident high level language compilers, etc.

REFERENCES

1. Belsey, D.A., Kuh, E., and Welsch, R.E. Regression Diagnostics 1980.

2. Diaconis, P., and Friedman, J.H. "M and N Plots", Tech. Report #151, Dept. of Statistics, Stanford University, April, 1980.

3. Donoho, D., Huber, P.J., and Thoma, H. The Use of Kinematic Displays to Represent High Dimensional Data, Research Report #PJH-5, Dept. of Statistics, Harvard University, March 1981.

4. Fisherkeller, M.A., Friedman, J.H., and Tukey, J.W. Prim-9, An Interactive Multidimensional Data Display and Analysis System. Proc. 4th International Congress for Stereology, 1975

5. Friedman, J.H. and Stuetzle, W., "Projection Pursuit Regression", JASA v. 76, 1981.

6. Friedman, J.H. and Stuetzle, W. "Hardware for Kinematic Statistical Graphics", Tech. Rep. #Orion00, Dept. of Statistics, Stanford University, Feb. 1982.

7. Friedman, J.H. and Tukey, J.W. "A projection pursuit algorithm for exploratory data analysis", IEEE Trans. Comput. C-23 pp. 881-890, 1974.

8. Gnanadesikan, R. Methods for Statistical Data Analysis of Multivariate Observations, 1977.

9. Harrison, D. and Rubinfeld, D.L. "Hedonic Prices and the Demand for Clean Air", Journal of Environmental Economics and Management, 5, 1978.

10. Mosteller, F. and Tukey, J.W. Data Analysis and Regression, 1977.

11. Rosenbrock H.H., "An automatic method for finding the greatest or least value of a function", Comput. J. 3, 1960.

12. Tukey, J.W. Exploratory Data Analysis, 1977.