Manual Controls for High-Dimensional Data Projections

Dianne COOK and Andreas BUJA

Projections of high-dimensional data onto low-dimensional subspaces provide insightful views for understanding multivariate relationships. This article discusses how to manually control the variable contributions to the projection. The user has control of the way a particular variable contributes to the viewed projection and can interactively adjust the variable's contribution. These manual controls complement the automatic views provided by a grand tour, or a guided tour, and give greatly improved flexibility to data analysts.

Key Words: Data visualization; Dynamic graphics; Grand tour; Multivariate analysis; Projection pursuit.

1. INTRODUCTION

This article builds on dynamic visualization methods for high-dimensional data using low-dimensional projections. Among these methods, the most familiar are 3-D data rotations, generated by displaying a continuous sequence of 2-D projections of 3-D data. From a statistical perspective it is rare to have data that are strictly 3-D, and so, unlike most computer graphics applications, the more useful methods for data analysis show projections from an arbitrary high-dimensional space. We collect such methods under the term "tours."

Tours involve views of high-dimensional (p) data in low-dimensional (d) projections. In his original paper on the *grand tour*, Asimov (1985) provided several algorithms for tour paths that could theoretically show the viewer the data "from all sides." A grand tour path is postulated to be dense in the set of all d-dimensional planes in p-space, meaning that, if the viewer could watch until the end of time, she would come arbitrarily close to every possible d-dimensional projection of the data.

Prior to this conceptual advance several methods were available for touring data. In the PRIM-9 system—for picturing, rotation, isolation, and masking in up to nine dimensions—rotation was achieved by horizontal and vertical interpolations between user-selected pairs of variable axes. None of the early rotation methods have the dense-

Dianne Cook is Assistant Professor, Department of Statistics, Iowa State Universty, Ames, IA 50011-1210; e-mail: dicook@iastate.edu. Andreas Buja is a Member of Technical Staff, AT&T Bell Laboratories, Florham Park, NJ 07932-0971; e-mail: andreas@research.att.com.

Journal of Computational and Graphical Statistics, Volume 6, Number 4, Pages 464–480

ness property of the grand tour, so they will systematically miss some fraction of the projection space. With the user control method in PRIM-9, though, the problem is rather the impossible nature of manually steering through the projection space.

There have been several recent developments in tour methods. Asimov's grand tour was originally proposed in a movie format, but computing hardware advances have spurred research into user interface issues as well (Buja, Asimov, Hurley, and McDonald 1988). With the proposed user interfaces came mechanisms for including or excluding a variable, pausing, speed control, and backtracking over the path. To make viewing more efficient, high-dimensional data space could be restricted to the space of several principal components or interpolating between orthogonal subspaces (Hurley and Buja 1990). Alternatively, more interesting projections could be given a higher probability of a visit than less interesting views during the tour path (Cook, Buja, Cabrera, and Hurley 1995). Underlying these expanded tour methods are tour path constructions whose mechanics and theory are based on subspace interpolation, the details of which are given in Buja, Cook, Asimov, and Hurley (in press).

A different line of research into tours was pursued by Wegman (1991) who introduced the notion of a "full-dimensional" grand tour using a p-dimensional parallel coordinate display to render the data in a continuously moving coordinate system. A full-dimensional grand tour can also be rendered in a $p \times p$ scatterplot matrix, providing the effect of monitoring many windows of activity simultaneously. Full-dimensional tours with both rendering methods are available in ExplorN (Carr, Wegman, and Luo 1996).

Full-dimensional tour algorithms are implicitly used in LispStat (Tierney 1991) and XploRe (Härdle, Klinke, and Turlach 1995) and, for a density grand tour by Scott (1995). In each case, only two or three dimensions are peeled from the moving coordinate system and used for display. In LispStat, the implementation lends itself to an interesting feature: While the grand tour progresses, 2-D projections are shown from the first two moving coordinates; when the grand tour is stopped, the third coordinate provides a 3-D subspace in which the data can be further spun with familiar 3-D controls. In this method the third dimension is as unpredictable as the touring coordinate system from which it is inherited.

In this article we propose manual controls for 1-D and 2-D projections of p-D data space by controlling the variable contributions in the projections. Similar to LispStat, we augment the current 2-D projection plane with a backdimension to create a 3-D space, but in contrast to LispStat, this backdimension is user-chosen, predictable, and interpretable.

Coming from a legacy of fully automated grand tours, the motivation for our work was to provide the user with more intimate controls over the views of the data. In a full circle of graphics research, these controls return us to the earliest implementation of manual tours, after several important developments in automated grand tours. Variable-centered manual controls were provided in PRIM-9 with rotation between pairs of variables. By fully integrating automatic grand tours and manually guided tours, we greatly increase the power of these tools in the hands of data analysts.

The proposed manual controls for 1-D and 2-D projections are described in Section 2. Examples of their use are given in Section 3. The implementation details are given in the Appendix. For illustration, we use the XGobi system (Swayne, Cook, and Buja in press) at each step of the discussion.

A competing variable-based manipulation approach is described in Duffin and Barrett

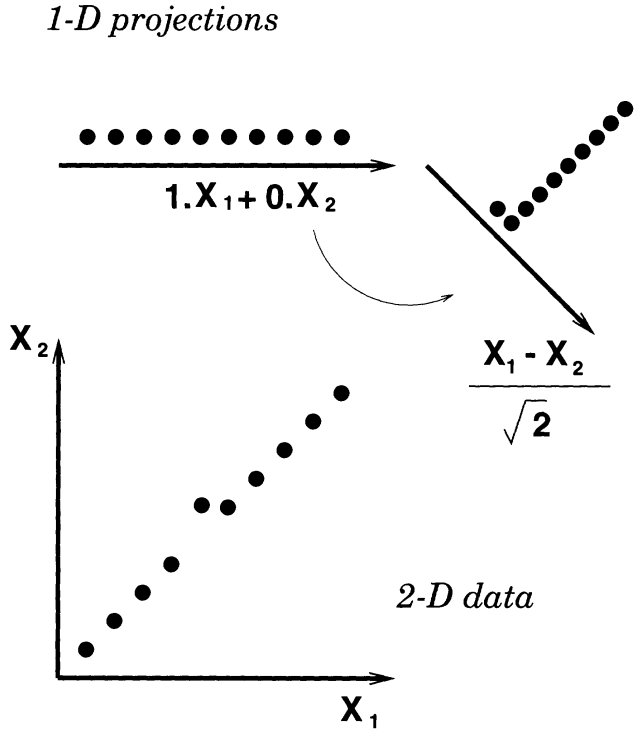

Figure 1. Looking at a 2-D data set with 1-D projections. Pulling X_2 into the projection reveals the out-of-line point.

(1994). Some of their methods appear to be a subset of ours, corresponding to what we call "radially constrained" manipulation in the following. These authors also allow "angular manipulation" but in a different manner than what is described here.

2. FUNCTIONALITY FOR CONTROLLING 1-D AND 2-D PROJECTIONS

The paradigm is that of pulling a variable in and out of the current projection. In what follows, this variable will be called the "manip variable" for short.

2.1 MANIPULATION OF 1-D PROJECTIONS

This is the simplest case. For 1-D projections, one constructs the 2-D plane spanned by the current projection vector on the one hand, and the basis vector corresponding to the manip variable on the other hand. Rotating the projection direction in this plane has the effect of pulling the manip variable in and out of the 1-D projection.

Figure 1 shows a toy example where manipulation of 1-D projections lets us discover structure: Some 2-D data have all points collinear except one. The projection onto variable 1 (X_1) does not reveal this out-of-line point, but if the second variable (X_2) is pulled into the projection, the structure is revealed in the projection $(X_1 - X_2)/\sqrt{2}$.

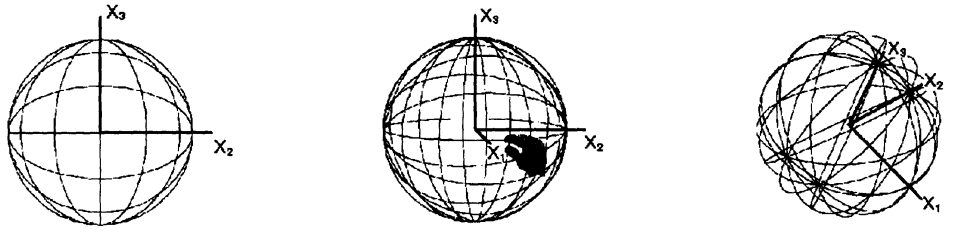

Figure 2. A schematic picture of trackball controls. The semblance of a globe is rotated by manipulating the contribution of X_1 in the 2-D projection.

2.2 MANIPULATION OF 2-D PROJECTIONS IN A GRAND TOUR

Two-dimensional manual control is analytically more involved but otherwise quite intuitive because we conceive of it as a 3-D spatial manipulation. (See the Appendix for algorithmic details.) Three-dimensional manipulation is akin to rotating a globe with the hand. Assume that the globe can be rotated arbitrarily—that is, it has no stand restricting its movement, and also that we have a coordinate system in $\mathbb{R}^3$ that rigidly follows the globe. Figure 2 shows a sequence of globe rotations made by manipulating the contribution of the first variable (X_1).

The present situation differs from conventional 3-D computer graphics only in that the 3-D manipulation space may be composed of linear combinations of more than three variables. We construct a 3-D subspace from the current 2-D projection plane by using the direction of the manip variable as a third axis, coming out of the screen (the plane, that is) at a generally oblique angle (see Fig. 3). The intuition is that we can pick up the variable axis like a lever and pull it around in a 3-D subspace.

In XGobi, these contols are naturally embedded in the grand tour mode, where arbitrary 2-D projections of arbitrarily high-dimensional variable spaces can be animated, subjected to projection pursuit, and manually controlled with mouse sweeps.

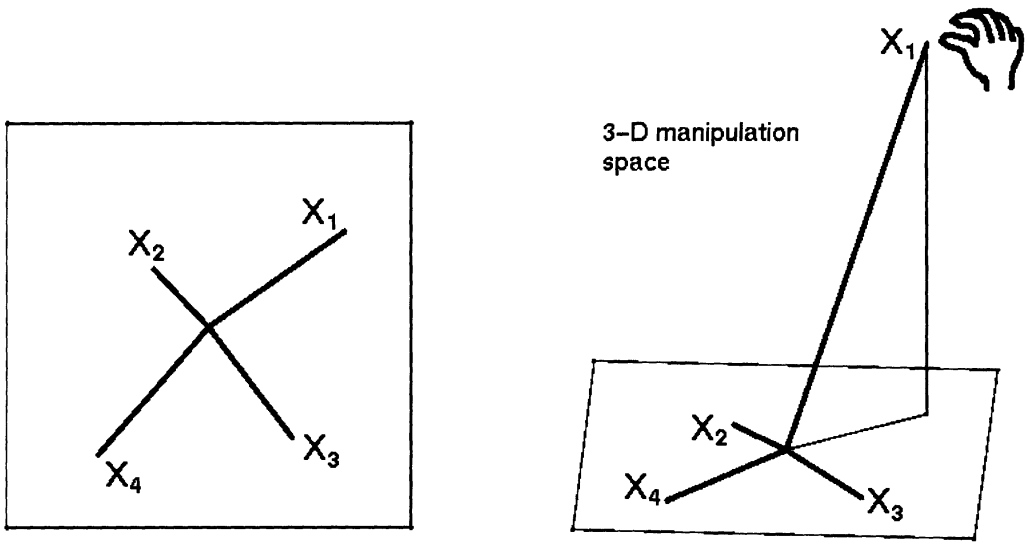

Figure 3. Constructing the 3-D manipulation space to manipulate the contribution of variable 1 in the 2-D projection.

On the side, we note that if the direction of the manip variable falls in the current projection plane, no 3-D space results and the 3-D manipulations cannot take place. An implementation has to intercept this singular case.

2.3 MANIPULATION OF TWO 1-D PROJECTIONS IN A CORRELATION TOUR

One-dimensional projections become considerably more interesting when two of them are simultaneously manipulated—one on the horizontal axis and one on the vertical axis. This is indeed possible in XGobi, and the natural place is in the so-called "correlation tour" mode (Buja et al. 1988). Correlation tours differ from grand tours the way canonical correlation analysis differs from principal component analysis: Variables are divided into X variables, shown in horizontal 1-D projection, and Y variables, shown in vertical 1-D projection. Correspondingly, we need to select *two manip variables*, a horizontal one and a vertical one. The simultaneous manipulation of both 1-D projections generally results in a genuine 4-D rotation, manually controlled. (Algorithmic details are again in the Appendix.)

For illustration, consider the following simple situation: Assume we are currently looking at the variables X_1 plotted horizontally and Y_1 plotted vertically. Assume further that the horizontal and vertical manip variables are X_2 and Y_2, respectively. We then subject the two-parameter family of pairs of 1-D projections

$$(c_\theta X_1 + s_\theta X_2, c_\phi Y_1 + s_\phi Y_2) \tag{2.1}$$

to manual control. If only θ is moved starting with $(\theta, \phi) = (0, 0)$, the effect is a 3-D rotation around the fixed Y_1 axis as θ varies but ϕ stays fixed. Similarly, moving ϕ translates into a 3-D rotation around the fixed X_1 axis. If, however, the parameters move identically, $\theta = \phi$, the manipulation results in a transformation of the (X_1, Y_1) plot into the (X_2, Y_2) plot as $\theta = \phi$ increase from 0 to $\pi/2$. If θ and ϕ vary in arbitrary ways, one can attain plots of essentially any linear combination of Y_1 and Y_2 against any linear combination of X_1 and X_2.

In general, when the starting projections are not X_1 and Y_1 but arbitrary linear combinations of $\sum_i a_i X_i$ and $\sum_j b_j Y_j$, respectively $\left(\sum_i a_i^2 = \sum_j b_j^2 = 1\right)$, simultaneous rotation should simply be considered as that—rotation of two 1-D projections. These manipulations become more intuitive once we think of them as ways of pulling in or pushing out variable contributions in both axes. Selecting X_k and Y_l as manip variables, the manipulation will result in changes in the relative contributions of a_k and b_l to their linear combinations.

2.4 A NOTE ON THE USER INTERFACE

Manual controls require a specification of the user interface. On today's computers, direct manipulation of graphical objects is mostly provided through mouse motions and mouse clicks. In the implementation of manual controls in XGobi we followed widespread conventions: Moving the mouse while depressing a button ("mouse sweep") translates into an action on the projection(s).

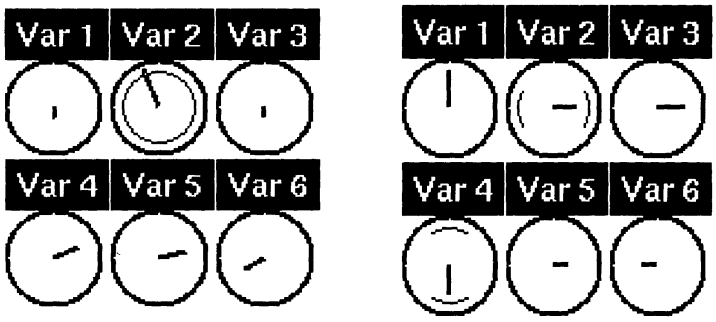

Figure 4. Indicating which of the variables are the ones to be manipulated (left) 2-D manipulation, Var 2, in a grand tour (right) 1-D manipulation, in both horizontal (Var 2) and vertical (Var 4) directions of a correlation tour.

Manipulation of a 2-D projection in 3-D in grand tour mode is implemented following the trackball paradigm (Foley, van Dam, Feiner, and Hughes 1990): Mouse sweeps on the plot have the same effect as rotating a globe with a corresponding sweep of the hand.

Manipulation of two 1-D projections in correlation tour mode is implemented in the obvious way: The horizontal component of a mouse sweep is proportional to the amount of rotation of the horizontal projection, and similarly the vertical component is proportional to the amount of rotation of the vertical projection.

Finally we need a way to select and change the manip variable(s). Variable selection in the grand tour and the correlation tour is done by clicking on the variable circles, so to select a manip variable we have added a shift modifier. In the grand tour, depressing <Shift> and clicking on a variable circle selects the manip variable, indicated by a thin, smaller circle within the variable circle. In the correlation tour, depressing <Shift> and clicking the left (right) mouse button changes the horizontal (vertical) manip variable, indicated by thin arcs within the variable circle. See Figure 4 for the appearance.

2.5 CONSTRAINED MANIPULATIONS

The manipulations described previously provide control over a variable's contributions to a projection in an arbitrary direction and magnitude. We call unconstrained manipulation *oblique* in the grand tour mode and *combined* in the correlation tour mode, for obvious reasons.

By constraining the quantities picked up from mouse sweeps, we obtain several useful constrained manipulations. In both grand tour and correlation tour, it is sometimes necessary to constrain the manipulations either in the *horizontal* or the *vertical* screen direction. Under these constraints, only the horizontal or the vertical component of a mouse sweep will be translated into a motion, leaving the other direction at rest.

In the grand tour, two more constrained modes are useful—*radial* and *angular*. Under the radial constraint, only the component in the direction of the projection of the current manip variable is picked up; that is, the manip variable will only extend or diminish in its current direction, maintaining its angle on the screen. Under the angular constraint, the manip variable's contribution stays fixed in size but it rotates on the screen following

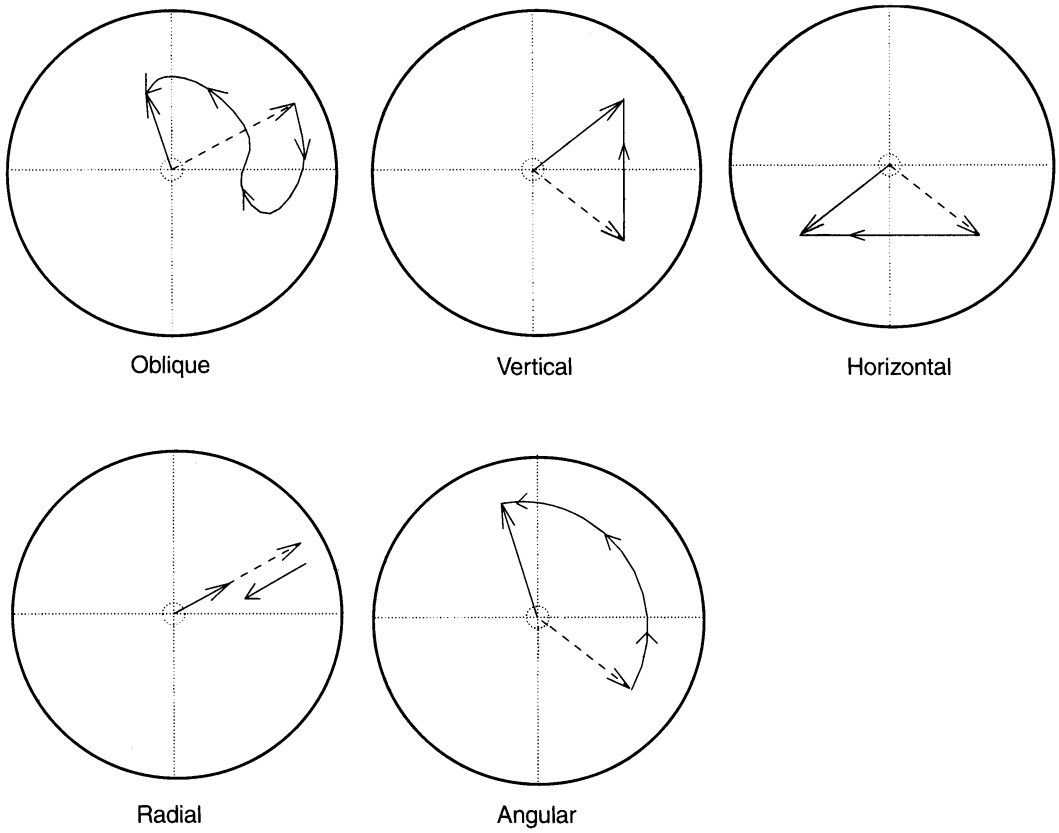

Figure 5. Constrained manipulation modes for 2-D projections. The dashed line represents the variable contribution to the projection before manipulation, and the solid line is the contribution after manipulation.

mouse sweeps around the center of the plot.

In the correlation tour, one more constrained mode is useful: *equal*, whereby only the component of the mouse in the 45-degree direction is picked up, moving the horizontal and vertical projections in the same amounts.

Figure 5 illustrates the constraints for the grand tour.

2.6 ARBITRARY LINEAR COMBINATIONS AS MANIP VARIABLES

Manipulation based on one coordinate axis is reasonably natural and has the advantage that the action can be quickly understood: A mouse sweep reduces or increases the magnitude and/or direction of one variable's contribution to the projection. But it may be unnecessarily rigid. In some situations it is desirable to manipulate several variables simultaneously. Doing the manipulation sequentially, variable by variable, is unrealistic.

The problem is easily solved by generalizing the manip variables to essentially arbitrary manip directions. That is, instead of a manip variable X_k, permit general manip linear combinations $X = \sum_j \alpha_j X_j$. This may be too much generality, but a few examples will illustrate situations in which this could be useful:

α	*Application*
$\frac{1}{\sqrt{p}}[1 \ldots 1]'$	rotates the view toward the multivariate mean
$\frac{1}{\sqrt{2}}[10 \ldots 0 - 1]'$	rotates the view towards the contrast of the first variable with the last
$\frac{1}{\|\|\hat{\beta}\|\|}\hat{\beta}$, where $\hat{\beta}$ is a least squares estimate in a regression model	rotates the view towards the estimated regression model projection

3. APPLICATIONS

The applications of interactive manipulation tools are surprisingly broad. We will discuss applications to projection pursuit, conditional plots, and multiple time series. Additional examples are described in Buja, Cook, and Swayne (1996).

3.1 PROJECTION PURSUIT

Projection pursuit as an exploratory tool is used to find revealing low-dimensional projections of high-dimensional data. The algorithm involves defining a criterion of interest (e.g., negative entropy), which is formulated for an arbitrary projection and uses optimization over the space of all possible projections of the data to find the global maximum. Finding local maxima is also important because from a data analysis perspective the most revealing projection does not necessarily correspond to the global maximum. More complete accounts of projection pursuit methods can be found in Diaconis and Freedman (1984), Huber (1985), Jones and Sibson (1987), and Friedman (1987).

In practice, when conducting projection pursuit the big challenge is optimizing the index in a manner that can be used for exploratory data analysis. We have argued in Cook et al. (1995) that monitoring the optimization visually provides more exploratory power than unmonitored optimization. In that article a projection pursuit optimization algorithm was animated which revealed some previously unknown features in a data set. Here we discuss how the manual controls in conjunction with projection pursuit optimization can be used to get a better understanding of multivariate structure. Having these controls available then would have saved us considerable time to reach the same conclusions.

In Figures 6 and 7 a demonstration of using the interactive controls is shown on the 7-dimensional particle physics data (Fisherkeller, Friedman, and Tukey 1974). This data set was extensively described in Cook et al. (1995) using the combination of grand tour and projection pursuit.

The two views in Figure 6 illustrate a problem that can arise when a purely numerical index is used to measure interestingness. The projection shown in the left side plot is the one that corresponds to a local maximum of the so-called central mass index (described in Cook et al. 1995). It is an interesting view because three fairly clean, but connected,

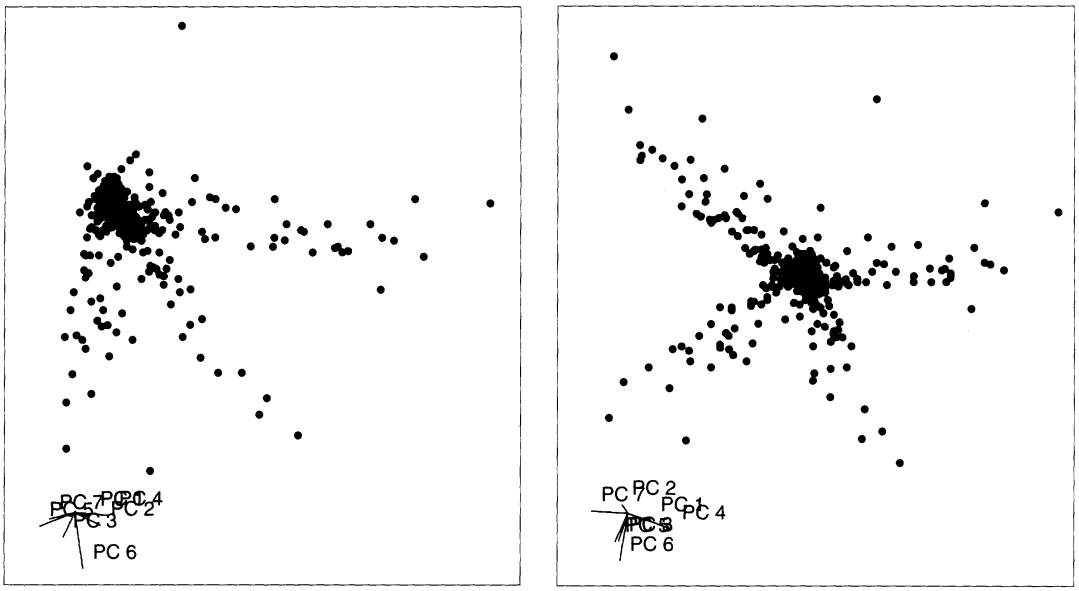

Figure 6. A projection corresponding to a maximum of projection pursuit index—central mass = .638 (left); A projection that is slightly more interesting in that there are four visible linear features—central mass = .610 (right).

clusters can be seen. The view of the data shown in the right side plot is a projection "close to" the maximum. It is actually more interesting than the view given by the maximum because four clusters are visible. Note that if one was conducting exploratory projection pursuit using a batch algorithm, rather than through a visual interface, this view would have been missed entirely.

Figure 7 illustrates how we might use manual controls to move from the three-cluster projection to the four-cluster projection. Principal component 7 is first pulled out a little. Principal component 5's contribution is manipulated from a downwards direction to a more horizontal contribution. Principal component 4 is moved to a slightly more vertical contribution. Principal component 2's contribution is increased to a positive horizontal position. Finally, reducing the contribution of principal component 1 produces the four-cluster view. So we "twisted" from one interesting view to another, which is impossible to achieve without manual controls. They allow us to understand how extensive/prominent each structure is (that is, easily seen or fairly hidden), how close two features are in data space, and how they relate to each other (for example, whether the four clusters are obtained by splitting one of the three clusters).

Other interesting questions that can be answered with manual controls are:

- If one variable is rotated out, does the structure remain intact? That is, is it possible to simplify the interpretation? (Morton [1989] attempted to answer this question by building a penalty term into the projection pursuit index to make the projection corresponding to the maximum more interpretable.)
- If another variable is rotated in, would the structure become "visually sharper"?

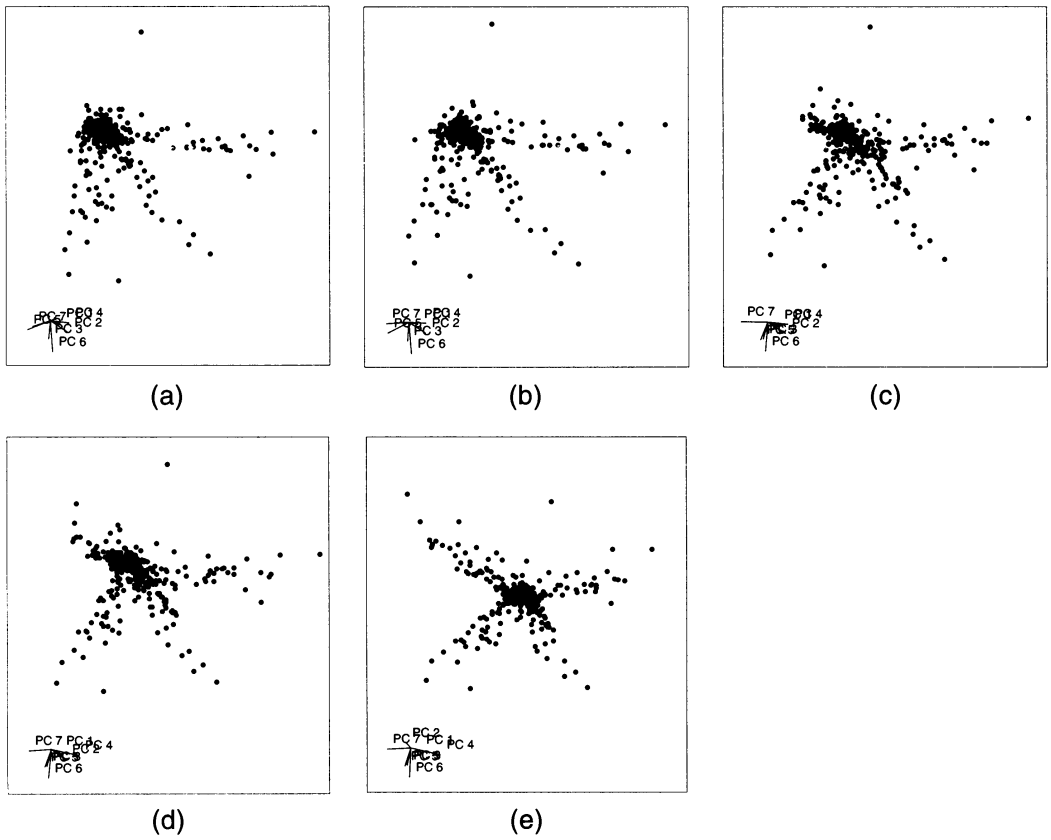

Figure 7. A sequence of intermediate projections betwen the three-cluster view and the four-cluster view of Figure 6. Principal component 7 is first pulled out a little. Principal component 5's contribution is manipulated from a downwards direction to a more horizontal contribution. Principal component 4 is moved to a slightly more vertical contribution. Principal component 2's contribution is increased to a positive horizontal position. Reducing the contribution of principal component 1 produces the four-cluster view.

3.2 TIPPING IN RESTAURANTS

The data used in this example were collected to study the factors that affect tipping in restaurants. They are taken from Bryant and Smith (1995). One foodserver recorded the total bill, total tip, sex of the person paying the bill, whether there was a smoker in the party, day of the week, night or day, and the size of the party, for all of his customers during an interval of two and a half months in early 1990. The restaurant, located in a suburban shopping mall, is one of a national chain that serves a varied menu. We use projection manipulation to create conditional plots with the binary variables.

In Figure 8 are three plots showing successive stages of the manipulation. The left plot displays tip vertically and the total bill horizontally. Points below the diagonal are the "low-tipping" customers. Notably, more customers give low tips than high ones! In the middle plot we have pulled the variable sex (the sex of the person paying the bill) into the projection horizontally. The plot of tip versus total bill for the males is left of the females. There is less spread for the females, essentially due to fewer higher total bills. The right plot shows the variable smoker (there is a smoker in the party) introduced into the projection vertically. Something very interesting can be seen: The variability of the smoker groups is huge compared to the nonsmokers; that is, the tip paid by smokers

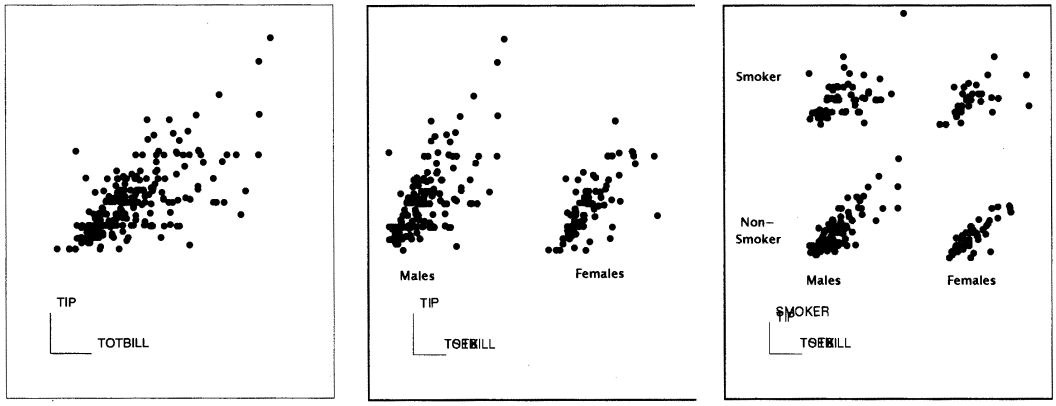

Figure 8. Sequential stages in the manipulation of the conditional variables SEX *and* SMOKER *into the projection: tip versus total bill (left); conditioned on sex (middle); conditioned on sex and smoker (right). Interestingly, the tip is very unpredictable when there is a smoker in the party, as evidenced by the huge variability in the upper plots. By contrast, female nonsmokers are meticulously consistent when it comes to calculating the tip, with the exception of three low tippers.*

is very unpredictable! It is fun to note that, with the exception of three low-tippers, the female nonsmokers are very consistent when it comes to calculating the tip.

Conditional plots are commonly used data analytic tools, generated by plotting several static plots. Here we have the same plotting power in a dynamic environment, where we can change the conditioning instantly and watch the interpolated views as the conditional plots are drawn out of the marginal plot. In this example, it was very useful to see the smokers pulled apart from the non-smokers because the motion dramatizes the variance differences. What is gained in dynamic action is countered by a lack of fancy plot additions—axes, overlaid regression/smooth lines—as may be found in specialized tools for conditional plots ("trellis displays," http://cm.bell-labs.com/cm/ms/departments/sia/-project/trellis/).

3.3 MULTIPLE TIME SERIES

When exploring multiple time series one often examines the lag relationships between series. This section discusses how the projection manipulation controls can be used for this purpose. The actions described in this example are familiar and well-established practices in exploratory time series analysis. A low-tech version is to hold up the individual series plots (plotted on identical time scales) to the light and slide the sheets of paper horizontally until the peaks and troughs roughly match. The electronic equivalent has been available in specialist software for some time (Unwin and Wills 1988), but for the case that such software is not readily available, we show how projection manipulation tools, surprisingly, can be applied for the same effect. It is necessary to preprocess the data, but this can be easily automated for arbitrary multiple time series. (An S function is available on the web page for this article, http://www.public.iastate.edu/~dicook/researc/papers/manip.html.)

The data has five time series measuring different aspects of pig production in the United Kingdom, with measurements taken quarterly over a 12-year period from 1967 to

1978. The data can be found in Andrews and Herzberg (1985). The series are as follows:

Series 1 Number of sows in pig for the first time; that is, a measure of intake into the breeding herd (GILTS).

Series 2 Ratio of all-pig price to all-fattener feed price (PROFIT).

Series 3 Ratio of sow and boar slaughter to total breeding herd size, that is, the removal of pigs from the breeding herd (SB).

Series 4 Number of clean pigs (meat) slaughtered (CP).

Series 5 Actual breeding herd size (HERDSZ).

We preprocess the data as follows:

1. Each series is standardized.

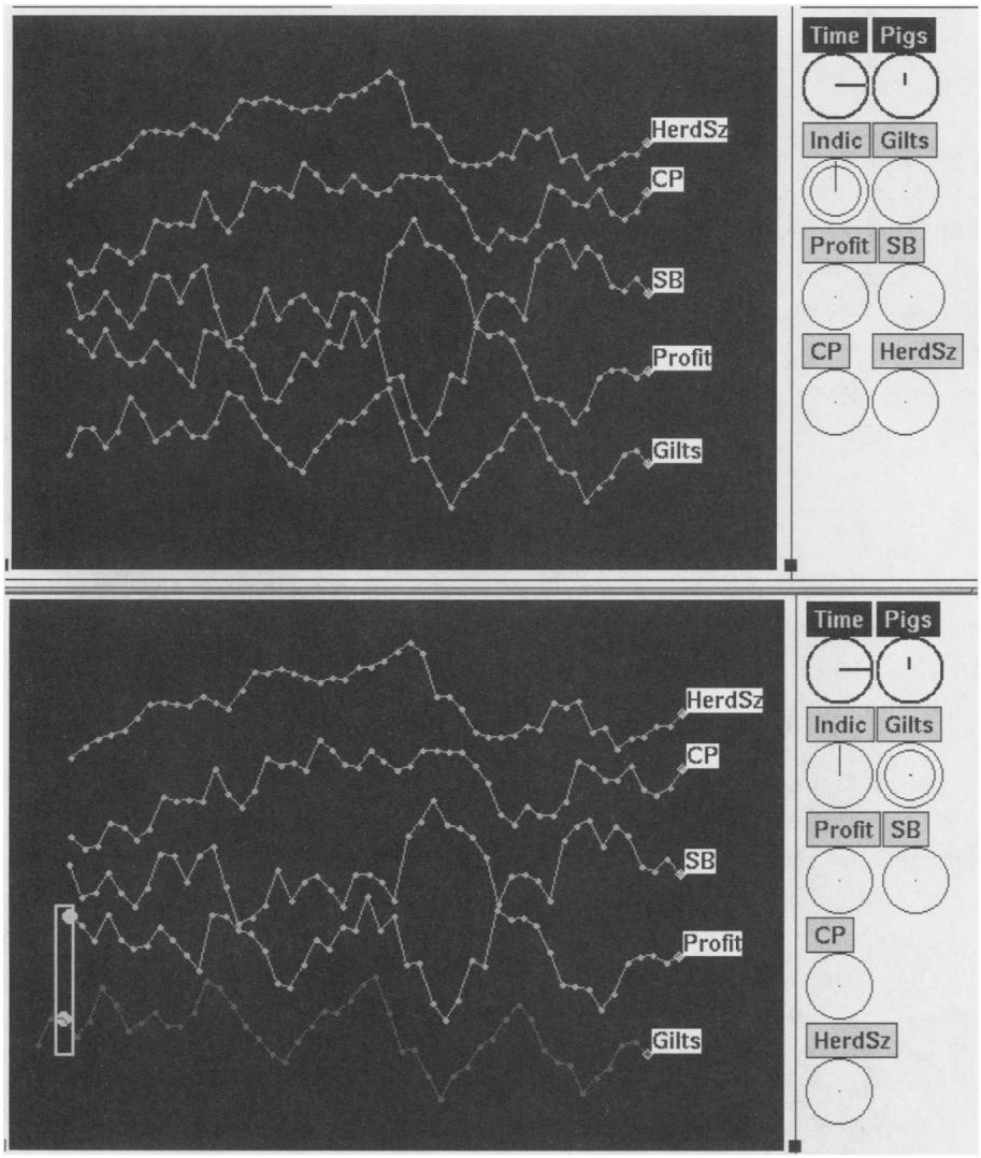

Figure 9. The series indicator variable Indic is manipulated into the vertical projection, thus separating the series, and the indicator for the Gilts series (red) is manipulated into the horizontal projection, thus lagging the series on the other series.

2. The five series are concatenated (***Pigs***):

$$GILTS_1, \ldots, GILTS_{48}, PROFIT_1, \ldots, PROFIT_{48}, \ldots, HERDSZ_1, \ldots, HERDSZ_{48}$$

3. New variables are created:
   (a) Series indicator (***Indic***): $\overbrace{1\ldots1}^{48}\overbrace{2\ldots2}^{48}\overbrace{3\ldots3}^{48}\overbrace{4\ldots4}^{48}\overbrace{5\ldots5}^{48}$
   (b) Lag indicators (five variables)

	$\overbrace{\quad}^{48}$	$\overbrace{\quad}^{48}$	$\overbrace{\quad}^{48}$	$\overbrace{\quad}^{48}$	$\overbrace{\quad}^{48}$
Gilts	$-1\ldots-1$	$1\ldots1$	$1\ldots1$	$1\ldots1$	$1\ldots1$
Profit	$1\ldots1$	$-1\ldots-1$	$1\ldots1$	$1\ldots1$	$1\ldots1$
SB	$1\ldots1$	$1\ldots1$	$-1\ldots-1$	$1\ldots1$	$1\ldots1$
CP	$1\ldots1$	$1\ldots1$	$1\ldots1$	$-1\ldots-1$	$1\ldots1$
Herdsz	$1\ldots1$	$1\ldots1$	$1\ldots1$	$1\ldots1$	$-1\ldots-1$

The ***Indic*** variable is used to separate the series by manipulating this variable into the vertical projecton, leaving the series sequentially plotted vertically (Fig. 9). Any one of the series can be lagged positively or negatively on the remaining series by manipulating the indicator variable for the series into the horizontal projection. For example, in Figure 9 the ***Gilts*** series has been lagged against the rest. We stopped shifting the series when the ***Profit*** peaks matched the ***Gilts*** peaks. The lag relationship looks to be about three-quarters of a year. This is interpretable given that the gestational period is about four months and clean pigs are usually slaughtered between four to six months of age.

4. DISCUSSION

Manual controls provide a fine tuning and searching mechanism for viewing high-dimensional data using tours. Grand tours and correlation tours provide overviews of data, guided tours search for more revealing views, and manual projection controls used in conjunction with these tools provide the necessary finesse to sharpen features and perform sensitivity analysis. In addition, creative use of manual controls on categorical data allows us to generate unexpected views, such as conditional plots, or lagged overlays of multiple time series.

The methods described in this article have been implemented in XGobi (Swayne et al. in press), with the exception of those in Section 2.6. We have concentrated on manual controls for 2-D projections and simultaneous control of two 1-D projections. These methods have been made relatively simple by reducing motions in high-dimensional data spaces to fairly intuitive 3-D and 4-D motions. It would be interesting to contemplate manual controls for 3-D projections of p-D spaces, especially given the rise of virtual reality technology where 3-D displays are the norm. Finally, one could entertain manual controls for full-dimensional tours as well.

A. APPENDIX: ALGORITHMS

A.1 1-D PROJECTIONS

The manipulation starts from a fixed but arbitrary 1-D projection, where

- $\boldsymbol{f}$ is a p-dimensional vector defining the current projection;
- $\boldsymbol{e} = \boldsymbol{e}_k$ is a p-dimensional vector of 0s with a 1 in the k-th position—k represents the variable we wish to manipulate;
- $\boldsymbol{X}_{(n\times p)}$ = data;
- dist_x = horizontal distance of the mouse sweep; and
- dist_y = vertical distance of the mouse sweep.

The steps (described for manipulation in the horizontal direction) are as follows:

1. First check if $||\boldsymbol{e}-\boldsymbol{f}|| <$ tolerance value. If this is true then $\boldsymbol{e}$ is for computational purposes the same as $\boldsymbol{f}$, and it is not possible to generate a 2-D manipulation space. An alternative would be to randomly select a new $\boldsymbol{e}$.
2. Orthonormalize $\boldsymbol{e}$ on $\boldsymbol{f}$ using a Gram–Schmidt step:
$$\boldsymbol{e} \leftarrow \boldsymbol{e} - < \boldsymbol{e}, \boldsymbol{f} > \boldsymbol{f}.$$
3. Preproject the data
$$(\boldsymbol{x}, \boldsymbol{z})_{n\times 2} \leftarrow \boldsymbol{X}_{n\times p}(\boldsymbol{f}, \boldsymbol{e})_{p\times 2}.$$
4. Initialize $\phi = 0$. *Now start capturing mouse sweeps.*
5. $$\phi = \phi + \frac{\text{dist}_x}{\text{size of plot region}}, \quad c_\phi = \cos(\phi), \quad s_\phi = \sin(\phi)$$
(Use dist_y instead of dist_x for vertical manipulation.)
6. Plot the data using
$$\boldsymbol{x} = c_\phi \boldsymbol{x} + s_\phi \boldsymbol{z},$$
7. and the projection coordinates using
$$c_\phi \boldsymbol{f} + s_\phi \boldsymbol{e}.$$

To keep the horizontal and vertical manipulations constrained to be equal set $\text{dist}_x^* = \text{dist}_y^* = (\text{dist}_x + \text{dist}_y)/2$ which corresponds to taking only the part of the mouse sweep that is in the $x = y$ direction.

A.2 2-D PROJECTIONS

Manpulation starts from a fixed but arbitrary 2-D projection of the p-dimensional data, where

- $\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2)$ is the $(p \times 2)$-dimensional orthonormal basis which defines the current projection;

Table A.1. Modifications to the Algorithm for Constrained Manipulation Modes

Constraint	*Algorithm modification*
Vertical	Set $c_\theta = 0$ and $s_\theta = 1$.
Horizontal	Set $s_\theta = 0$ and $c_\theta = 1$.
Radial	If the manip variable has no contribution to the projection, compute this from the first sweep by $(r_x, r_y) = (\text{dist}_x, \text{dist}_y)/\|\| \cdot \|\|$. The proportion of mouse sweep in the direction of the manip variable is captured by $(\text{dist}_x^*, \text{dist}_y^*) = ((\text{dist}_x, \text{dist}_y)'(r_x, r_y))(r_x, r_y)$. Then dist_x^* and dist_y^* replace dist_x and dist_y in the algorithm above.
Angular	Let ψ be the angle of the mouse movement around the center of the screen. Then rotate in the plane of the screen by setting $R = \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

- $\boldsymbol{e} = \boldsymbol{e}_k$ is a p-dimensional vector of 0s with a 1 in the k-th position - k represents the variable we wish to manipulate;
- $\boldsymbol{X}_{(n\times p)}$ = data;
- dist_x = horizontal distance of the mouse sweep; and
- dist_y = vertical distance of the mouse sweep.

The setup to begin interactive manipulation involves the following steps:

1. Orthonormalize $\boldsymbol{e}$ on $\boldsymbol{f}$ using a Gram-Schmidt step:

$$\boldsymbol{e} \leftarrow \boldsymbol{e} - < \boldsymbol{e}, \boldsymbol{f}_1 > \boldsymbol{f}_1 - < \boldsymbol{e}, \boldsymbol{f}_2 > \boldsymbol{f}_2.$$

   If $\|\boldsymbol{e}\| <$ tolerance value, then $\boldsymbol{e}$ is contained wholly within the plane spanned by $\boldsymbol{f}$. As a solution one could generate $\boldsymbol{e}$ randomly from the space of unit vectors in $\mathbb{R}^p$. If this is not the case, then $\boldsymbol{e}$ can be normalized.

2. Preproject the data into the space spanned by the $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{e})$:

$$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})_{n\times 3} \leftarrow \boldsymbol{X}_{n\times p}(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{e})_{p\times 3}$$

   and initialize the new 3-D orthonormal basis for this data space to be

$$(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d})_{3\times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

   Now you are ready to start capturing mouse sweeps.

3. Let

$$\phi = \frac{\text{length of mouse motion}}{\text{size of plot region}}, \quad c_\phi = \cos(\phi), \quad s_\phi = \sin(\phi)$$

$$c_\theta = \cos(\theta) = \frac{\text{dist}_x}{\text{length of mouse motion}}$$

$$s_\theta = \sin(\theta) = \frac{\text{dist}_y}{\text{length of mouse motion}}$$

which gives the rotation matrix

$$R = \begin{pmatrix} c_\theta^2 c_\phi + s_\theta^2 & -c_\theta s_\theta (1 - c_\phi) & -c_\theta s_\phi \\ -c_\theta s_\theta (1 - c_\phi) & s_\theta^2 c_\phi + c_\theta^2 & -s_\theta s\phi \\ c_\theta s_\phi & s_\theta s_\phi & c_\phi. \end{pmatrix}.$$

When R is premultiplied by $(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d})$ the rotated orthonormal basis results. The pronumeral ϕ can be interpreted as the angle from the initial position of $\boldsymbol{d}$ to the rotated position, and θ can be interpreted as the angle between the starting position of $\boldsymbol{h}$ to the rotated position.

4. To display the 2-D data projection post-multiply the preprojected data, $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, by the current $(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d})$:

$$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d}),$$

and plot the first two columns of the result.

5. The projection coordinates (from p-dimensions to 2) are obtained from post-multiplying the initial $(p \times 3)$-dimensional basis by $(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d})$:

$$(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{e})(\boldsymbol{h}, \boldsymbol{v}, \boldsymbol{d})$$

and use the first two columns of the result to get the horizontal and vertical contribution of each variable.

Small modifications provide several constrained manipulation modes given in Table A.1.

ACKNOWLEDGMENTS

We appreciate the suggestions by the associate editor and two reviewers, and thanks to Sunhee Kwon for a little background digging for us. The first author was partially supported by National Science Foundation grants DMS9632662 and DMS9214497.

ADDITIONAL RESOURCES

Additional resources associated with the paper—software, data sets—can be found at http://www.public.iastate.edu/~dicook/research/papers/manip.html.

[Received July 1996. Revised 1997.]

REFERENCES

Andrews, D. F., and Herzberg, A. M. (1985), *Data—A Collection of Problems from Many Fields for the Student and Research Worker*, New York: Springer-Verlag.

Asimov, D. (1985), "The Grand Tour: A Tool for Viewing Multidimensional Data," *SIAM Journal of Scientific and Statistical Computing*, 6, 128–143.

Bryant, P. G., and Smith, M. A. (1995), *Practical Data Analysis: Case Studies in Business Statistics*, Homewood, IL: Richard D. Irwin Publishing.

Buja, A., Asimov, D., Hurley, C., and McDonald, J.A. (1988), "Elements of a Viewing Pipeline for Data Analysis," in *Dynamic Graphics for Statistics,* eds. W. S. Cleveland, M. E. McGill, Monterey, CA: Wadsworth, pp. 277–308.

Buja, A., Cook, D., Asimov, D., and Hurley, C. (in press), "Dynamic Projections in High-Dimensional Visualization: Theory and Computational Methods," *Journal of Computational and Graphical Statistics.*

Buja, A., Cook, D., and Swayne, D. (1996), "Interactive High-Dimensional Data Visualization," *Journal of Computational and Graphical Statistics* 5, 78–99.

Carr, D. B., Wegman, E. J., and Luo, Q. (1996), "ExplorN: Design Considerations Past and Present," Technical Report 129, Center for Computational Statistics, George Mason University.

Cook, D., Buja, A., Cabrera, J., and Hurley, C. (1995), "Grand Tour and Projection Pursuit," *Journal of Computational and Graphical Statistics* 4, 155–172.

Diaconis, P., and Freedman, D. (1984), "Asymptotics of Graphical Projection Pursuit," *The Annals of Statistics,* 12, 793–815.

Duffin, K.L., and Barrett, W.A. (1994), "Spiders: A New Interface for Rotation and Visualization of N-Dimensional Point Sets," in *Proceedings of Visualization '94,* Los Alamitos, CA: IEEE Computer Society Press, pp. 205–211.

Fisherkeller, M., Friedman, J. H., and Tukey, J. (1974), "PRIM-9: An Interactive Multidimensional Data Display and Analysis System," Technical Report SLAC-PUB-1408, Stanford Linear Accelerator Center, Stanford, CA.

Foley, J.D., van Dam, A., Feiner, S.K., and Hughes, J.F. (1990), *Computer Graphics: Principles and Practice*, Reading, MA: Addison-Wesley.

Friedman, J.H. (1987), "Exploratory Projection Pursuit," *Journal of American Statistical Association,* 82, 249–266.

Härdle, W., Klinke, S., and Turlach, B. A. (1995), *XploRe: An Interactive Statistical Computing Environment*, New York: Springer-Verlag.

Huber, P. J. (1985), "Projection Pursuit," (with discussion), *The Annals of Statistics,* 13, 435–525.

Hurley, C., and Buja, A. (1990), "Analyzing High-Dimensional Data with Motion Graphics," *SIAM Journal on Scientific and Statistical Computing,* 11, 1193–1211.

Jones, M. C., and Sibson, R. (1987), "What is Projection Pursuit?" (with discussion), *Journal of the Royal Statistical Society,* Series A, 150, 1–36.

Morton, S. C. (1989), "Interpretable Projection Pursuit," Technical Report 106, Laboratory for Computational Statistics, Stanford University.

Scott, D. W. (1995), "Incorporating Density Estimation into other Exploratory Tools," in *ASA Proceedings of the Section on Statistical Graphics,* Alexandria, VA: American Statistical Association, pp. 28–35.

Swayne, D. F., Cook, D., and Buja, A. (in press), "XGobi: Interactive Dynamic Graphics in the X Window System," *Journal of Computational and Graphical Statistics.*

Tierney, L. (1991), *LispStat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics*, New York: Wiley.

Unwin, A. R., and Wills, G. (1988), "Eyeballing Time Series," in *ASA Proceedings of the Section on Statistical Graphics,* Alexandria, VA: American Statistical Association, pp. 263–268.

Wegman, E. J. (1991), "The Grand Tour in k-Dimensions," Technical Report 68, Center for Computational Statistics, George Mason University.