User's Manual for XGobi, a Dynamic Graphics Program for Data Analysis Implemented in the X Window System (Release 2)

Deborah F. Swayne, Bellcore
Dianne Cook, Rutgers University and Bellcore
Andreas Buja, Bellcore

XGobi brings contemporary dynamic graphics for statistics to the workstation environment. Dynamic graphical methods offer a data analyst the power of motion, high interaction and rapid response for discovering and understanding higher-dimensional relationships among variables. For example, XGobi has provided insights within Bellcore to software engineers analyzing data on switch memory allocation and to market researchers studying telephone usage data.

XGobi's plots are all variants of the scatterplot. These scatterplots can be single-variable strip plots, pairwise plots, or linear combinations of three or more variables. The plots can be set in motion: three-variable rotation or higher dimensional rotation known as the grand tour. In addition, a method called the projection pursuit guided tour is provided for directing the tour toward interesting views of the data. The scatterplots can be transformed and manipulated in many ways. Groups of points can be distinguished with a method called brushing or identified with row labels, either in one window at a time or using two or more linked windows. A plot can be moved and reshaped, and variables can be transformed interactively. An on-line help system is available which contains most of the information in this manual.

XGobi is implemented in the X Window System™, so it is widely portable. It has been tested on workstations manufactured by both Sun Microsystems, Inc. and the Digital Equipment Corporation, and on some high-end personal computers as well. [1]

XGobi is designed to be used as a complement to existing analytical tools. For example, one can use XGobi and the S data analysis environment simultaneously, making it possible to use each program's unique functions on the same data.

This memo is an updated version of an XGobi user's manual that was issued in July, 1990.

X Window System is a trademark of MIT.

[1] It is the policy of Bellcore to avoid any statements of comparative analysis or evaluation of products or vendors. Any mention of products or vendors in this document is done where necessary for the sake of scientific accuracy and precision, or to provide an example of a technology for illustrative purposes, and should not be construed as either a positive or negative commentary on that product or vendor. Neither the inclusion of a product or vendor in this document, nor the omission of a product or vendor, should be interpreted as indicating a position or opinion of that product or vendor on the part of the authors or of Bellcore.

Contents

1 Introduction

Multidimensional data arise in many disciplines within the telecommunications industry. Scientists and engineers, as well as statisticians, are faced with the task of analyzing complex data on network traffic, software performance, telephone usage and device characteristics. These data can come as experimental measurements, simulation results, or survey responses.

Interactive and dynamic graphical methods are very useful in the analysis of multidimensional data. Such methods enable a data analyst to directly manipulate graphical displays of the data on a computer screen, with nearly instant response. A user can quickly become familiar with a new set of data by scanning sequences of scatterplots, interactively rescaling plots and using the mouse to identify interesting points.

Furthermore, possible relationships among three or more variables can be discovered and explored. For instance, rotation may reveal high-dimensional curvature or multidimensional outliers, and brushing can help an analyst spot clusters and study their behavior. This information can be used to guide the use of other graphical and analytic techniques.

XGobi is a program for data analysis which combines many graphical tools for the display and manipulation of scatterplots in a highly interactive direct manipulation format. Its functionality presently includes the following:

- one-variable textured dot plots;
- two-variable scatterplots, to which connected lines can be added;
- three-dimensional rotation around an arbitary axis with trackball control, including the ability to save coefficients of rotation;
- grand tour rotation: a smooth randomized sequence of two-dimensional projections in order to explore a higher-dimensional point cloud;
- projection pursuit guided tour: a method for helping the grand tour to find interesting views of the data;
- brushing: the ability to change the color or plotting character of a point or a group of points, including the ability to transmit that information to other windows, and to save indicator vectors of brushing information;
- moving and reshaping of the data in the plot region;
- identification: displaying the case label or row number of an interesting point by positioning the cursor near it, including the ability to cause the same point to be labelled in other windows;

- variable transformation;
- on-line help.

XGobi has been written so that it can be linked to New S (Becker et al. (1988)) or it can be run as a stand-alone program. When run in conjunction with S, it accepts a matrix in S data format as input and writes its output as S vectors. Otherwise, it reads and writes ASCII files.

The X Window System was chosen as a software implementation environment for XGobi because it is the most widely used window system for UNIX® workstations, and it offers good portability across workstations. X has also been implemented on some high-end personal computers. In recent years, X has been chosen as a development environment by several of Bellcore's owner-client companies. XGobi has been tested on workstations manufactured by Sun Microsystems, Inc. and the Digital Equipment Corporation, among others, and is expected to work wherever X works.

In this memo, we present a more detailed description of each of XGobi's capabilities and a manual for its use. For further reading on the use of dynamic graphics for statistics, see Cleveland and McGill (1988).

2 Initiating XGobi

XGobi can be executed at the UNIX shell level or from within S. When it is initiated at the shell level, XGobi can accept an ASCII file as an argument or read from standard input, and it writes output to an ASCII file. Within S, XGobi is initiated using an S function, and the first argument to the S function is an S matrix. In this case, XGobi writes out data by creating files in the S data format.

2.1 At the shell level

Execute XGobi with the name of a file as an argument:

xgobi datafile

Each line in the file must contain one row of the input data matrix, and lines must be separated by carriage returns. Columns, or variables, can be separated by any number of tabs or spaces. (If the file containing the data matrix has a name of the form datafile.dat, it can be passed to xgobi as datafile; that is, without the .dat suffix.)

UNIX is a registered trademark of UNIX System Laboratories.

XGobi also accepts standard input, which allows a user to select certain rows or columns to plot using UNIX commands. For example, columns can be selected using 'cut' and piping the output to XGobi:

cut -f1-10 datafile | xgobi

Rows can be specified using 'sed' or 'head' and piping the output to XGobi:

head -1500 datafile | xgobi

To find variable and case labels, XGobi looks for labels files in the directory where **datafile** is found. Variable (column) labels must be in a file named datafile.col (or datafile.column, datafile.collab, or datafile.var), where datafile is the name of the file containing the matrix. Case (row) labels must be in a file named datafile.row (or datafile.rowlab or datafile.case). There should be one label per line, and the label can include blanks. If variable or case labels are not supplied, default labels using column or row numbers are used. If XGobi is reading from standard input, it does not look for any labels files, but uses the default values.

Variables can be grouped so that all variables in a group are scaled and transformed together. You might want to group variables which use the same unit of measure, such as number of telephone calls per unit time, or any other variables which you want to compare directly. In that case, if one of the variables is transformed, it seems natural that every variable in the same group should be similarly transformed. (See section 4.2 and 5.6.6 for other references to variable grouping.)

Upon initialization, XGobi looks for a file named datafile.vgroups which is constructed as follows: there is a single row of integers, one integer for each variable in the matrix. Variables to be grouped together are assigned the same integer. For example, a file containing the row 1 2 3 3 1 is used with a data matrix of five columns. The first and fifth variable will be grouped, as will the third and fourth. If datafile.vgroups is not found each variable is scaled individually and any transformation applied to that variable affects no other variable.

By default, the scaling used is minimum-maximum; that is, the data are scaled into the plotting window using the minimum and maximum values of each variable or variable group, in such a way that the midpoint of the variable is at the center of the plotting window, and no points fall outside the plotting window. Two other scaling methods can be specified with command line arguments. To scale using mean and standard deviation, type

xgobi -std msd datafile

To use median and median absolute deviation (MAD), use **mmd** instead of **msd**. If one of those two scaling methods is chosen, the number of standard deviations (or MADs) from the mean (or median) to be contained within the plotting window can be specified as well, by adding **-dev x**, where **x** is a real number between 0 and 100. (The view of the data can be rescaled at any time once XGobi is running; see section 5.6.6.)

Other command line arguments are those used by standard X programs; read about them in documentation for the X Window System such as Jones (1989). A frequently used argument is "display," which allows a user to run an X application on one machine and display the output on another. It is used as follows:

xgobi -display machine_name:0 datafile

2.2 From within S

At the S command line, type

> **xgobi(matrix)**

All the capabilities described in the previous section for use on the UNIX command line are also available for the xgobi S function; the S help page in the Appendix describes the arguments to the xgobi S function which allow these capabilities. The data matrix can be an S expression, as in this example:

> **xgobi(cbind(spd,pow,freq))**.

3 Note on terminology

There are some common shorthand expressions in use to describe mouse behavior. "Click left (or middle or right)" on a button or region means position the mouse cursor on that button or region, and then press and release the left (or middle or right) mouse button.

"Selecting" a button or mode usually means clicking left on that button or on the button displaying that mode's name. If it means clicking middle or right, that should be clear from the context. A selected button is highlighted, meaning it is displayed with its foreground and background colors reversed. Sometimes "deselecting" is used to mean clicking on a button that is already highlighted.

"Dragging the mouse" means pressing one of the mouse buttons and holding it down while moving the mouse itself.

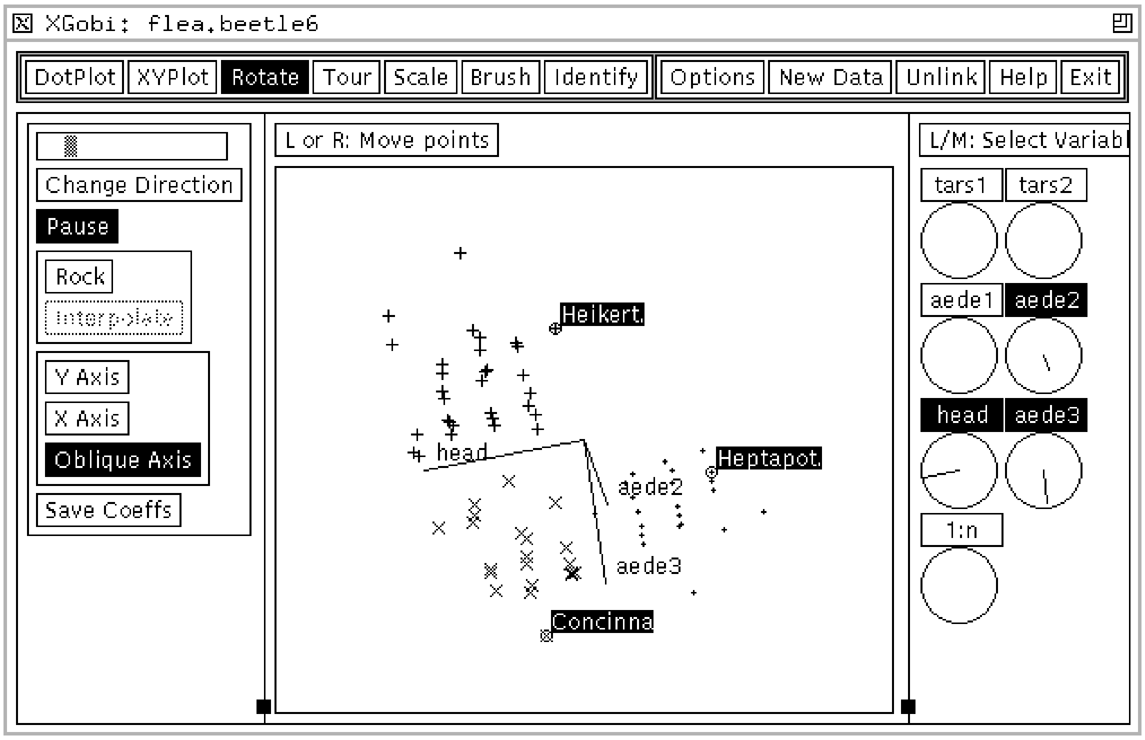

Figure 1: XGobi window showing rotation and brushing

4 Layout

When you initiate an XGobi process, an XGobi window appears on your screen, or on whatever other screen you designate. It has a row of buttons across the top which we call the main panel or the main control panel.

The rest of the window is divided into three sections, as you can see in Figure 1: the variable selection panel on the right, the plotting window in the center, and a control panel on the left. The partitioning of the space among these three elements, called "panes," can be adjusted interactively using the small rectangles, called "grips," that appear at the bottom of the partitioning lines. Here is an explanation of their use, taken from the documentation for the Athena Widget Set (Peterson (1991)): "When the mouse pointer is positioned on a grip and pressed, an arrow is displayed that indicates the pane that is to be resized. When keeping the mouse button down, the user can move the grip . . . left and right . . . This, in turn, changes the size of the pane."

Readers who are familiar with the work of Buja, Hurley, and McDonald (Buja et al. (1988), Hurley and Buja (1990)) will recognize XGobi's layout and much of its functionality. Many modes and features in XGobi are reimplementations in C and the X Window System of designs originally prototyped on a Symbolics LISP machine.

4.1 Main panel

The main panel is the row of buttons across the top of an XGobi window. This panel is divided into two sections. The buttons on the left (from **DotPlot** to **Identify**) represent XGobi modes, and they are mutually exclusive. For example, if **Scale** is selected, then selecting **Brush** causes **Scale** to be deselected. Each of these modes also has an associated control panel which is displayed to the left of the plotting window when it is selected.

The buttons on the right of the main panel (from **Options** to **Exit**) are miscellaneous top level controls. All the buttons in the main panel are described in Section 5.

The first group of buttons in the main panel has two functional subgroups. The leftmost buttons, **DotPlot**, **XYPlot**, **Rotate** and **Tour**, are all used to change the plotting mode; that is, the dimensionality and other characteristics of the plot and the behavior of the variable selection panel. Selecting one of the rightmost buttons, **Scale**, **Brush** or **Identify**, does not affect the plotting mode, but rather makes certain functions available for the current plot.

Figure 1 was produced by first selecting **Rotate** and then rotating the data to produce the displayed view. **Brush** was then selected: this didn't alter the view, but allowed the displayed plot to be brushed so that three glyph types are present. **Identify** was selected next; this also didn't alter the view, but allowed three case labels to be attached. Upon reselecting **Rotate**, the user could then rotate the plot while retaining the effects of brushing and identification.

4.2 Variable selection panel

At the right of the window is a set of labeled circles, called variable selection circles, each of which is associated with one variable. There is one extra variable, which may hold different data throughout an XGobi session; its behavior is described in section 4.2.1.

The variable selection circles act as buttons which respond to mouse clicks by selecting and deselecting variables for plotting. A selected variable is indicated by a highlighted label above the corresponding variable selection circle. In Figure 1, the variables *aede2*, *head* and *aede3* are selected.

The behavior of the variable selection circles depends on the plotting mode, and is described in the corresponding sections of this manual. Above the variable panel is a bit of text to remind the user of the mouse actions to perform to select and deselect variables. In Figure 1, the text is truncated; the full label reads "L/M: Select Variable," meaning that either the left or the middle mouse button is used to select a new variable. (Clicking on a variable circle or most other buttons with the right button actives XGobi's on-line help; see section 5.12 for further details.)

XGobi can be either in a one-variable plotting mode (initiated by selecting **DotPlot**), a two-variable plotting mode (**XYPlot**), a three-variable mode (**Rotate**), or a mode in which two, three or more variables can be plotted (**Tour**). In all plotting modes, the lines in the variable circles are parallel and proportional in length to the axes which can be displayed in the plot window.

In **DotPlot**, a horizontal line is drawn in the variable circle of the "extra" variable, and a vertical line in the circle of the variable being plotted.

In **XYPlot**, a horizontal line is drawn in the variable circle of the X variable and a vertical line in that of the Y variable.

In the **Rotate** mode, if rotation is about the **Y axis**, one Y variable is plotted against a linear combination of two X variables. A vertical line is drawn in the variable circle of the Y variable; horizontal lines are drawn in the variable circles of the two X variables, and they are proportional in length to the coefficients of the two X variables. If rotation is about the **X axis**, one X variable is plotted against a linear combination of two Y variables, and the horizontal lines appearing in the Y variable circles are proportional in length to the coefficients of the two Y variables. If **oblique** rotation is chosen, the lines in the three variable circles are usually oblique, as they are in Figure 1, and they indicate the length and orientation of the associated axes.

In **Tour** mode, the lines in the variable circles are usually oblique, and again parallel and proportional in length to the plotting axes; see Section 5.4 for further description.

Clicking left on a variable label brings up a menu of transformations (for example, **Inverse**, **Natural Log** and **Square Root**) that may be performed on that variable and other members of its variable group. A mark appears next to the name of the current transformation. To apply a transformation, drag the mouse down the menu until the desired transformation is highlighted and then release it. If data in the selected variable or variable group are outside the range of the transformation, it is not be applied and an error message appears in the window from which XGobi was initiated.

4.2.1 The extra variable

When XGobi is initiated, the extra variable, the last one in the variable selection panel, contains the row numbers of the data and is labeled "1:n." When **DotPlot** is selected, it is used to contain the values used by the texturing algorithm to spread the data laterally, and the label becomes "txtr." When another plotting mode is selected, the extra variable reverts to the row numbers.

If brushing has been performed, and different colors or glyphs exist, and the **Save in var** button in brushing is selected, then each group defined by brushing acquires a group number and those group numbers are stored in this variable. The label then becomes "group." If colors and glyphs are reset using the brushing **Reset Menu** and **Save in var** is selected again, the variable reverts to the row numbers. (It isn't possible to use the **Save in var** function on a plot that has been defined in the **DotPlot** mode.)

4.3 Plotting window

In the center of the XGobi window is the plotting window which always contains a scatterplot, with the possible addition of line segments connecting the points. In some XGobi modes, such as **Brush** or **Scale**, there are a few words of instruction displayed above the plotting window. These instructions describe the mouse operations that are effective within the plotting window while that mode is selected.

5 Main control panel buttons

5.1 DotPlot

This button initiates a one-variable plotting mode, displaying the data in a vertical textured dot strip, using a method that was proposed by Tukey and Tukey (1990). Textured dot strips spread the data laterally by amounts that are partly constrained and partly random, resulting in a fairly smooth spreading of the points and minimizing artifacts of the plotting method, such as stripes, clusters, or gaps.

When you select **DotPlot**, the last variable in the variable selection panel is used to contain the lateral spreading data, the "texturing" variable. Its label becomes "txtr." Only the middle button is effective for variable selection in this mode, since the "texturing" variable is held fixed.

Left click	always ignored
Middle click	
the "txtr" variable	ignored
the active variable	a new textured dot plot is generated
an inactive variable	this variable replaces the active variable

Selecting the active variable repeatedly causes new textured dot plots to be generated. These differ from one another due to the partially random nature of the texturing algorithm.

From the **DotPlot** mode, you can enter **Scale**, **Brush**, or **Identify** without losing a textured dot plot, and the last variable, "txtr," is unchanged. However, if you select **XYPlot**, **Rotate** or **Tour**, the variable will change: it is then used to contain the row numbers of the data.

5.2 XYPlot

This is the mode that XGobi is in upon initiation; it is a two-variable plotting mode. The control panel to the left of the plotting window contains the controls used in cycling through a sequence of plots. Select **Cycle** to activate this process. Selecting **Fix X** or **Fix Y** forces one of the axes to be held constant; otherwise, the plots cycle over all Y variables for the current X, and then move to the next X, and so forth. The scrollbar allows control of the delay between plots: moving the scrollbar to the right lengthens the delay.

The following table describes the results of different clicks in the variable selection panel:

Left click	
an inactive variable	this variable replaces the active X
the active X variable	ignored
the active Y variable	the active Y and X are interchanged
Middle click	
an inactive variable	this variable replaces the active Y
the active X variable	the active X and Y are interchanged
the active Y variable	ignored

The same variable selection rules apply when a two-variable scatterplot appears in the plotting window during brushing, scaling, or case identification. The plot in Figure 2 is in a two-variable mode, as is indicated by the two highlighted variables, Time and Flow, in the variable panel. The plot has been scaled and connected line segments have been added.

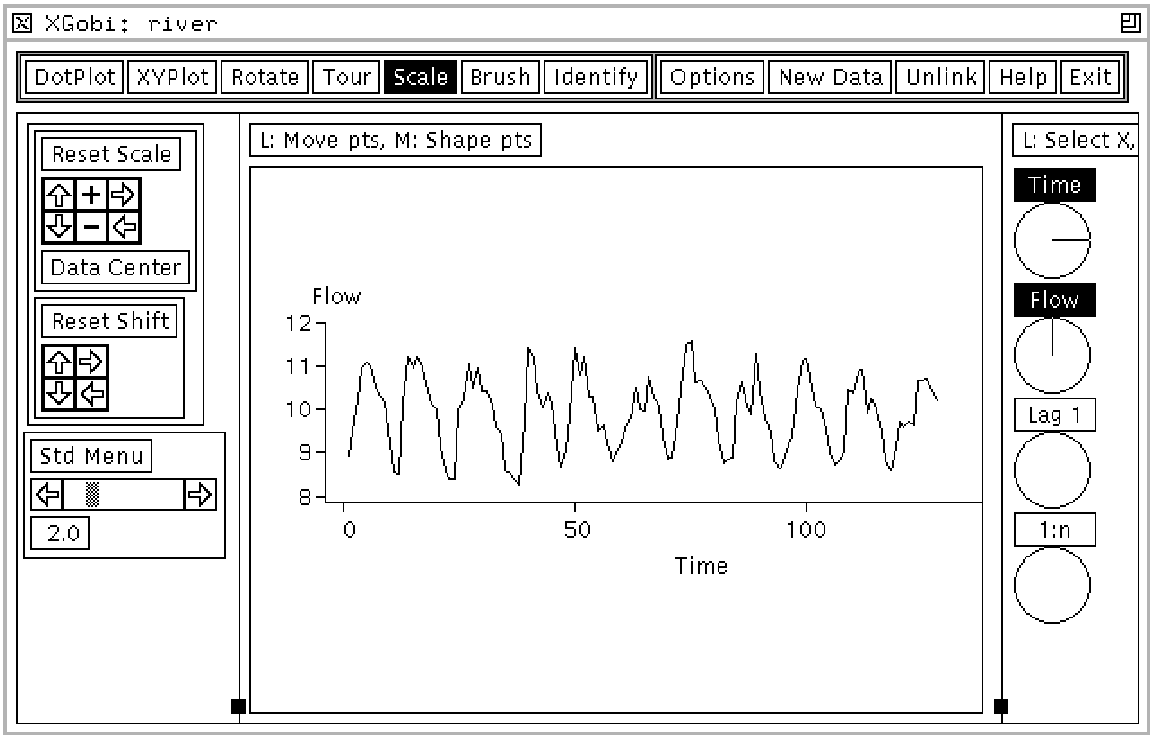

Figure 2: XGobi window showing a scaled two-variable plot with connected lines

5.3 Rotate

Rotation can be around three types of axes: the vertical Y axis, the X axis in the plane of the screen, or any oblique axis. For each type, a small circle, a *target*, is drawn in one of the variable circles to indicate that this variable will be affected by the next mouse click. (The *target* is only drawn if the cursor is in the variable selection panel.) Variable selection is different for each type.

Y Axis:

Left click	
an inactive variable	this becomes the new *target* X variable
active *target* X	ignored
active non-*target* X	becomes active *target* X variable
active Y variable	Y becomes *target* X ; *target* X becomes Y
Middle click	
an inactive variable	this variable replaces the active Y

active *target* X	Y becomes *target* X; *target* X becomes Y
active non-*target* X	Y becomes *target* X; *target* X becomes Y
active Y variable	ignored

X Axis:

Identical to Y axis behavior described above, except read Y for X, and X for Y.

Oblique Axis:

Left click or Middle Click

an inactive variable	this becomes the new *target* variable
active *target* variable	ignored
active non-*target* variable	this becomes active *target* variable

The same variable selection behavior continues during brushing, scaling and case identification when the current plot was defined during the **Rotate** mode.

In all three types of rotation, the cursor on the plot window changes to a hand indicating that trackball controls allow interactive plot manipulation. Dragging the left or middle mouse button in the plot window causes a rotation of the three-dimensional point cloud. When rotation is about a fixed horizontal or vertical axis, mouse movements cause rotation around that axis. During **Oblique Axis** rotation, mouse movement both causes rotation about the current axis and causes a new axis to be chosen.

When a user selects **Rotate**, a control panel appears to the left of the plotting window, as shown in Figure 1, and its elements are described here.

5.3.1 Speed scrollbar

The first element in the rotation panel is a scrollbar which is used to control the apparent speed of rotation. (In fact, it adjusts the angle of rotation between successive replots of the data.) To control the scrollbar, click the mouse where you would like the new speed setting to be. Dragging the vertical bar causes continuous adjustments.

5.3.2 Change Direction

Clicking left on this button reverses the direction of rotation.

5.3.3 Pause

Select **Pause** to halt rotation; deselect it to restart.

5.3.4 Rock and Interpolate

The states set by these two buttons are mutually exclusive, that is, selecting one causes the other to be deselected.

While the plot is rotating, selecting **Rock** causes the rotation to change direction after each ten subsequent replots. When **Rock** is selected, increasing the speed using the scrollbar causes the angle described by the rocking motion to be wider.

Selecting **Interpolate** causes the plot to rock between a plot of Y versus one X variable and Y versus the other X variable, if Y axis rotation is selected. (It causes the plot to rock between a plot of X versus on Y variable and a plot of X versus the other Y variable, if X axis rotation is selected.) During oblique axis rotation, it is not possible to select **Interpolate**.

5.3.5 Axis selection

Y Axis: Rotation is around the vertical axis.
X Axis: Rotation is around the horizontal axis in the plane of the screen.
Oblique Axis: Rotation is around any oblique axis. The axis is selected manually by dragging the mouse in the plot window with the left or middle button held down.

5.3.6 Save Coeffs

Click here to save the coefficients of rotation for the current view. Rotation stops and a window pops up that asks you to type in a file name. Position the cursor inside the text window to have your typing acknowledged; erase typed characters using your usual backspace operation. When you have entered the file name, hit return.

If XGobi is initiated from the shell, a two-column ASCII file is created. The first column contains the coefficients of the variables plotted on the X axis and the second column those on the Y axis.

If XGobi is initiated within S, a vector in the S data format is created. This vector can be transformed into a matrix with the same format as that of the two-column ASCII file described in the previous paragraph by using the following S command: **array(coefs, c(2,ncols))**, where coefs is the vector of coefficients and ncols is the number of columns in the current data matrix.

5.4 Tour

This button initiates a grand tour using the first three variables in the data matrix. The labels of the included variables are highlighted. (If the data matrix you supply has only one variable, the **Tour** button is not active.)

The grand tour method used here successively samples planes in p-space, where p is the number of variables presently selected, and connects the planes by moving along a geodesic interpolation path between them. The grand tour algorithm is as described in Buja et al. (1989). Motion is smooth between the starting and ending planes, but when the ending plane is reached there is a slight delay while the next randomly selected ending plane and consequent path are calculated. Allowing the grand tour to run without uninterruption, the viewer sees an ever-changing sequence of projections of the data in p-space.

The XGobi window in Figure 3 is in the **Tour** mode, exploring a six-dimensional space. (The user has been running projection pursuit guided tour, which is described later in this manual.)

A variable may be added to or removed from the tour by clicking on the corresponding variable selection circle with the left or middle mouse button. If at any time only two variables are selected, motion stops at a pairwise plot of the two, and it is not possible to remove any more variables. The bars drawn inside the variable circles are parallel and proportional in length to the axes for the current plot, and convey information about magnitude of the coefficients of each variable. That is, the horizontal projection of the variable bar corresponds to the X coefficient of that variable, and the vertical projection corresponds to the Y coefficient.

5.4.1 Speed scrollbar

Like the speed scrollbar in **Rotate,** this scrollbar allows the apparent speed of the grand tour to be adjusted. To control the scrollbar, click the mouse where you would like the new speed setting to be. Dragging the vertical bar causes continuous adjustments.

5.4.2 Pause

Click here to cause the motion of the tour to stop; click again to restart the motion.

5.4.3 Reinit

Clicking on this button resets the **Tour** coefficients using the currently selected variables. The plot becomes a pairwise plot of the first two selected variables in the variable selection panel, and the tour proceeds from that point.

5.4.4 Step and Go

When **Step** is active, the tour pauses when it reaches an ending plane. Clicking on **Go** generates one new ending plane, and the tour to travels to it. These two buttons are helpful for understanding the mechanism of the grand tour.

5.4.5 Local Scan

When **Local Scan** is selected, the current plane is chosen to be the "anchor" plane. The tour continues to choose new ending planes, but returns to the anchor plane after visiting each new plane.

5.4.6 Backtrck

As tour runs, a history file of the base planes passed is saved. When **Backtrck** is selected the tour runs backward through this path. Simultaneously, the number in the box to the right of the **Backtrck** button changes to display the number of the plane in the history file that the tour is approaching.

When the tour has backtracked to the first plane in the history file, it turns around and moves forward through the history file. This is indicated by the label immediately to the right of **Backtrck** which changes from **B** (backward) to **F** (forward). Similarly, when the tour has reached the end of the history file, it reverses direction again and begins to move backward through the recorded planes, and the label to the right of **Backtrck** correspondingly changes to **B**. When **Backtrck** is deactivated the tour begins again from the current projection, overwriting the history file from this position onward.

During **Backtrck** it is not possible to add or remove variables, since the purpose is to retrace old steps.

5.4.7 B/F

This button indicates the direction of the backtracking, and clicking on it changes the backtracking direction. It is only activated when **Backtrck** is on.

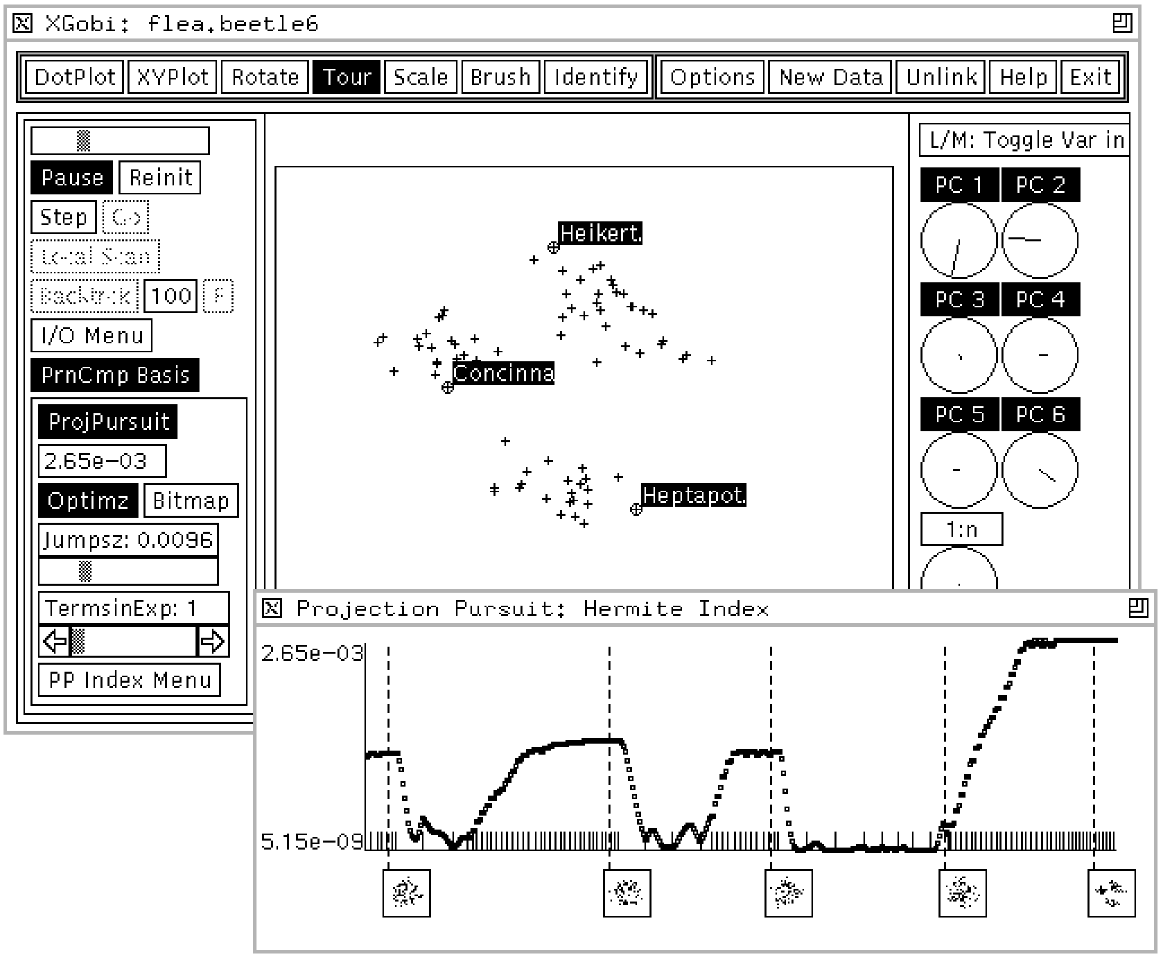

Figure 3: XGobi window showing projection pursuit guided tour and identification

5.4.8 I/O Menu

Save Coeffs: Like the **Save Coeffs** function in **Rotate,** this writes out the coefficients for the current view of the data into an ASCII file as a two column matrix. The first column contains the X coefficients of each variable and the second contains the Y coefficients. If XGobi is running from within S, these coefficients are written out as a vector in the S format. This vector can be transformed into a matrix with the same format as that of the two-column ASCII file by using the following S command: **array(coefs, c(2,ncols))**, where coefs is the vector of coefficients and ncols is the number of columns in the current data matrix.

Save History: This allows a history file created during backtracking to be saved, soliciting a file name from the user.
Read History: This allows one of the history files created during backtracking to be read in so that a prechosen tour path can be viewed, or a previous session continued.

5.5 Projection Pursuit Guided Tour

The grand tour is an exciting way to visually explore data, searching for interesting structure in high-dimensional space. One criticism made of the grand tour is that the chance of finding such structure becomes small as the dimensionality of the data increases. The projection pursuit guided tour in XGobi uses the gradient of a projection pursuit function to steer the tour toward interesting views, increasing the chance that a user will discover them. More detailed explanations can be found in Cook et al. (1991).

Selecting the **ProjPursuit** button while in **Tour** mode first turns on the **PrnCmp Basis** button, sphering the data by principal components, and creates a window in which the projection pursuit index for the current two dimensional view of the data is plotted. As the tour progresses the index is plotted continuously and its value is displayed directly beneath this button.

The XGobi window in Figure 3 shows a projection pursuit guided tour session that has been running for several minutes. The Projection Pursuit window contains a plot of the projection pursuit index against time.

The implemention of projection pursuit in XGobi invites user intervention at many points, as the descriptions of the **Optimz** button and the two scrollbars will make clear. In addition, a menu allows a user to select different projection pursuit functions, each of which may find different interesting views.

5.5.1 PrnCmp Basis

Selecting this button performs a principal components transformation of the active variables in the data. When projection pursuit is selected, it is automatically activated: the 2-dimensional projection pursuit indices are designed for sphered data, and currently this sphering is done by principal components.

There is no restriction in XGobi against using the projection pursuit guided tour with unsphered data, but it may behave strangely, in particular with the two polynomial-based projection pursuit functions (Legendre and Hermite).

5.5.2 Optimz

When this button is selected, the tour follows the gradient of the projection pursuit function to find more interesting views of the data. In this mode it is usual to keep control of the **Jumpsz** scrollbar to alter the frequency at which derivatives are calculated (the "jump size"), and thus control the optimization of the projection pursuit function.

For example, keep the jump size reasonably large (that is, keep the jump size scrollbar at about the center of its range) when starting **Optimz** from an arbitrary view so that large jumps toward the nearest local maximum can be taken. When the tour seems close to a local maximum, reduce the jump size to keep it in the local neighborhood.

When the guided tour reaches a local maximum, it may turn itself off. It does this when the derivatives are all very close to zero, meaning a better new direction cannot be found. This can happen at a local maximum, but it can also happen at low values if the function is flat. To turn the tour back on, turn **Optimz** off.

When **Optimz** is not selected, the projection pursuit index is still being plotted, but the guided tour is not taking place. Instead the viewer is watching a grand tour, with new directions of rotation being chosen at random.

The projection pursuit window in Figure 3 shows the history of several minutes of use of the guided tour. The function starts at a local maximum, and then suddenly the index drops: here the user turned **Optimz** off for a while and allowed a random tour to proceed. A few minutes later, the user turned **Optimz** on again, and a second local maximum was reached. This process was repeated twice more before a much larger maximum was reached. This maximum corresponds to the view of the data in the plotting window, which shows three distinct clusters.

5.5.3 Bitmap

Selecting this button when projection pursuit is running causes a miniature copy of the current view of the data to be drawn on the projection pursuit window. Figure 3 contains several examples.

5.5.4 Jumpsz label and scrollbar

The first scrollbar controls the tour "jump size;" that is, the distance between the starting and ending planes. Its effect is to control the frequency of the derivative calculations. (Thus it has no effect when **Optimz** is turned off.) When the jump size is large the tour is allowed to move further before a new direction, based on the derivative, is calculated. The associated label displays the current jump size.

You may want a small jump size when the projection pursuit index is increasing, and a large one (at about the middle of the scrollbar) to cause the tour to move away from a a region that is not interesting to you or when a local maximum has been reached.

To control the scrollbar, position the cursor about where you'd like the new jump size to be, then click the left or middle mouse button. Moving the vertical bar with the mouse button held down adjusts the jump size interactively.

5.5.5 TermsinExp or WinWidth label and scrollbar

There are two types of indices currently implemented. One type is based on density estimation using expansions in orthogonal polynomials, and the other is based on kernel density estimations. In each of these methods there is a parameter which it is desirable to alter from time to time, and this is done by means of this scrollbar. For the polynomial-based indices, the scrollbar controls the number of terms used in the expansions, and for the kernel density estimation methods, it controls the window width of the kernel; the label above the scrollbar changes accordingly.

Changing the number of terms included in the expansion has a strong effect on the behavior of the polynomial functions, causing them to reach local maxima for distinctly different views of the data. When the window appearing in Figure 3 was saved, there was only one term in the expansion.

To control the scrollbar, position the cursor about where you'd like the new number of terms in the expansion (or window width) to be, then click the left or middle mouse button. Moving the cursor with the mouse button held down adjusts the number of terms in the expansion (or window width) interactively. For finer control, click left on one of the arrows at either end of the scrollbar.

5.5.6 PP Index Menu

Use this menu to choose among four projection pursuit functions. The Legendre and Hermite indices are based on the expansion of density functions in terms of orthogonal polynomials; the Friedman-Tukey and Entropy indices are based on kernel density estimates.

The **Legendre** index is constructed by inverting the density through a normal cumulative distribution function, and using the Legendre polynomials for the expansion. This expansion is then compared with a uniform density on the interval [-1,1] by an L_2-distance. This is a good general index, and fast to compute. This index was proposed by Friedman (1987).

The **Hermite** index is constructed using Hermite polynomials for the same expansion as underlies the Legendre index. This expansion is compared to a standard normal density using an L_2-distance. This index works best when few terms are used in the expansion. The idea of using Hermite polynomials instead of Legendre polynomials was proposed by Hall (1989).

The **Friedman-Tukey** index is similar to the original projection pursuit index proposed by Friedman and Tukey (1974). It is based on an L_2-norm of a local kernel density estimate.

The **Entropy** index (Jones and Sibson (1987)) is an extension of the Friedman-Tukey index constructed using the negative entropy of a kernel density estimate.

5.6 Scale

The **Scale** button activates a set of operations for scaling and shifting the plot inside the plotting window. The **Scale** control panel, which can be seen in Figure 2, is to the left of the plotting window and has three subpanels. The controls in the upper two subpanels allow users to interact directly with the plot, changing its size and shape in a smooth, continuous way.

The controls in the lower subpanel allow users to choose among three methods for standardizing the view of the data. The default view standardization method used in XGobi simply scales the data into the plotting window using the minimum and maximum values of each variable or variable group. Assuming that the user has not scaled or shifted the plot, this method scales each variable in such a way that the midpoint of the variable is at the center of the plotting window, and no points fall outside the plotting window.

The other two view standardization methods scale the data so that the mean (or median) of the data lies at the center of the plotting window, and only the points smaller than some multiple of the standard deviation (or the median absolute deviation) fall inside the plotting window.

As an illustration of the use of these alternative view standardization methods, consider a set of data with a few very large outliers. In this case, the minimum-maximum method causes the bulk of the data to be plotted in a small clump and the outliers to be spread at the edges of the plotting window. A more useful view of the data can be easily obtained by restandardizing the view of the data using the median and the median absolute deviation (MAD).

Similar views can be obtained by using the interactive scaling and shifting controls, but restandardizing the view can be simpler and faster.

5.6.1 Reset Scale

This button restores the shape of the plot to a square region adjusted to the smallest dimension of the rectangular plotting window.

5.6.2 Scale controls

The six buttons appearing below the **Reset Scale** button all change the scale when they are pressed:

$\uparrow$	expand the data in the vertical direction
$+$	expand the data in horizontal and vertical directions equally
$\rightarrow$	expand the data in the horizontal direction
$\leftarrow$	contract the data in the horizontal direction
$-$	contract the data in horizontal and vertical directions equally
$\downarrow$	contract the data in the vertical direction

In addition, the data can be expanded and contracted by dragging middle inside the plotting window. This facility allows the reshaping of the data in arbitrary ways. The direction and extent of the change in scale depends on the direction and distance the user moves the cursor.

The plot in the Figure 3 has been reshaped using the Scale controls.

5.6.3 Data Center/Plot Center

This button toggles between two types of scaling methods. When the label is **Data Center**, the data are scaled from the center (midrange) of the data. This preserves the shape of the scatterplot. When the label is **Plot Center**, the data are scaled from the center of the plotting window. This facilitates shape changes. For example, when the data are all to one side of the center of the window, the points farthest from the center expand out or contract in faster than points near the center.

5.6.4 Reset Shift

This reset button re-positions the data around the center of the plotting window.

5.6.5 Shift controls

The four buttons appearing below the **Reset Shift** button all change the data translation when they are pressed:

$\uparrow$ move the plot up
$\rightarrow$ move the plot down
$\leftarrow$ move the plot left
$\downarrow$ move the plot right

In addition, the plot can be shifted by dragging left inside the plotting window, allowing the user to "grab" the point cloud and move it around.

5.6.6 Std Menu

This menu allows a user to choose among three different methods to standardize the view of the data. These choices apply to all the data, not just to the currently selected variables; that is, if new variables are subsequently selected for plotting, the same standardization method will be used.

Min/Max: Scale each variable so that the midpoint of the variable (or of the variable group to which it belongs) is at the center of the plotting window and no points fall outside the plotting window (assuming that the data have not been scaled or shifted).

Mean/StdDev: Scale each variable so that the mean of the variable (or of the variable group to which it belongs) is at the center of the plotting window and points within a multiple of the standard deviation fall inside the plotting window (assuming that the data have not been scaled or shifted).

Median/MAD: Scale each variable so that the median of the variable (or of the variable group to which it belongs) is at the center of the plotting window and points within a multiple of the median absolute deviation (MAD) fall inside the plotting window (assuming that the data have not been scaled or shifted).

5.6.7 View Standardization scrollbar and label

This scrollbar affects the second two view standardization methods. It adjusts the scaling of the plot by increasing or decreasing the number of deviations (either standard deviation or median absolute deviation) contained within the plotting window.

To control the scrollbar, position the cursor about where you'd like the new deviation to be, then click the left or middle mouse button. Moving the vertical bar with the mouse button held down adjusts the deviation interactively. For finer control, click left on one of the arrows at the ends of the scrollbar.

5.7 Brush

This button activates a set of operations used in brushing. Brushing can be performed with one single XGobi process running, but it is most interesting and useful to perform brushing with more than one XGobi process active, each displaying a different view of the same data. When brushing operations are performed in any one XGobi window, the new current state is instantly broadcast. Any other XGobi window that reads the broadcast information promptly updates its own window to reflect the changes. In order to respond to brushing events, an XGobi process must be plotting data with the same number of rows as the broadcasting window, but it need not be in the **Brushing** mode itself. (In fact, machine response is best if the passive window is in a **Pause** state or in the **XYPlot** mode, consuming few computing cycles.) In order to prevent the sending or receiving of brushing information, select **Unlink** in the main panel.

In Figure 1, the three clusters have been brushed so that each one is drawn using a different glyph.

The instruction label that appears above the plotting window is there to remind you how to use the brush, which is a rectangle drawn in the current brushing color. When you click left or middle with the cursor inside the plotting window, the brush jumps to the cursor. Then drag middle to change the size or shape of the brush, or drag left to move the brush within the plotting window. (If you lose the brush by pulling it outside the plotting window, you can grab it again if you press the left or middle button while the cursor is inside the plotting window.)

5.7.1 Brush On

When the **Brush On** button is highlighted, moving the brush over a plotted point causes that point to change its color or plotting character (called a glyph). If the brush is turned off, the brush can be freely moved across the plotting window and it does not change the points.

5.7.2 Color and Glyph Menus

Below the **Brush On** button are the **Color Menu** and the **Glyph Menu** buttons. Press one of the menu buttons and hold the mouse button down to activate the menus. Drag the mouse and release when the label for the color or glyph you want is highlighted.

(The **Color Menu** is only active if you are using a color workstation or personal computer.)

5.7.3 Brushing modes: persistent, transient, undo

These are three different brushing modes.

Persistent: When you brush a point, it retains its new color or glyph when the brush has moved away.
Transient: As the brush moves off a point, it returns to the color and glyph it had before the brush covered it.
Undo: When **Undo** is activated, moving the brush over a point that was painted during the preceding **Persistent** brushing action restores that point to the state it had just before it was brushed.

5.7.4 Erase and Complement

Erase: When **Erase** is activated, the brush is used to cause points to become invisible; that is, they are simply not plotted. This is always a persistent operation.
Complement: Clicking here causes erased points to become visible and visible points to become erased.

There is no **Undo** operation for erasing, but by using a combination of **Complement** and **Erase**, a user can recover particular points that were previously erased. In addition, the **Reset Menu** (see section 5.7.6) has a button that allows all erased points to be restored.

5.7.5 Send Update

If you initiate another XGobi window for a set of data, and you want that window to acquire the brushing characteristics of an existing window, then click on **Send Update** in the older window. This causes the new window to acquire all the information about color, glyph and visibility of the old window.

5.7.6 Reset Menu

Press and hold to access the reset choices presented in this menu.

Reset brush size	Reset the brush to its default size.
Reset colors	Restore all points to the default color.

Reset glyphs	Restore all points to the default glyph.
Restore erased points	Make all points visible.

5.7.7 Save in file

Click here to save the brushing glyph and color information. A popup window appears that solicits a file name. Position the cursor inside the text window to have your typing acknowledged; erase typed characters using your usual backspace operation.

If XGobi is initiated from the shell, two files are created: fname.glyphs and fname.colors, where "fname" is the name you typed into the popup window. (Only fname.glyphs is created if you are using a monochrome display.) Fname.glyphs contains integers corresponding to glyphs, where the integers are those that appear in the **Glyph Menu**. Fname.colors contains color names. If XGobi is initiated within S, two vectors in the S data format are created; they are named in the same way.

If you let fname be the name of the data you're currently viewing, then when XGobi is re-initiated with this same data, the points will be automatically painted using the glyphs and colors in the stored files. Similarly, these brushing files can be created by hand using an editor.

5.7.8 Save in var

Click here to store in the extra variable the groups created by brushing. Each subset of points that shares a common color and glyph forms a group, and each group is represented in the data as an integer between 1 and the number of groups. After you have saved the group information in that variable, its label changes to "group."

The group information will remain in the variable until either **DotPlot** is selected or **SaveInVar** is re-executed.

5.8 Identify

This mode is used to display identifying labels near points in the plotting window. The labels are taken from the file of row or case labels supplied by the user; if the file is not present, the row number of each point is used. To see these labels, simply move the cursor inside the plotting window. The label of the point nearest the cursor is displayed.

Identification in one window is instantly reflected in another window if both XGobi processes are viewing data matrices with the same number of rows. Just as it does during brushing, selecting the **Unlink** button prevents the sending and receiving of identification information. See the description of **Brush** for further explanantion of linking XGobi windows.

To cause a label to become "sticky," click left when the target label is printed. The printing style changes and the label now remains printed as the cursor moves off, and even remains printed as you leave the **Identify** mode. It is possible to rescale or rotate data, and the sticky labels will continue to be printed next to their associated points. In Figure 1, three points have associated sticky labels.

To cause a label to become "unsticky," return to the **Identify** mode and click left again when the target point is nearest the cursor. It is also possible to restore all labels to unsticky status by clicking on the **Remove Labels** button.

5.9 Options

Click here to pop up and down a menu of drawing options; each option is applicable to all plotting modes.

Add Axes: Select to add axes to the plot.
Show Points: Select to have points plotted; deselect to keep points from being plotted. You may want to deselect this button if you are adding connecting lines to the plot.
Show Lines: Select to add segments that connect all points in the order in which they occur in the data file.

You can change the default values for these options in your XGobi resource file. The defaults are:

*showAxes:	True
*showPoints:	True
*showLines:	False

5.10 New Data

Click on this button to read in a new data file together with its associated labels, groups, and brushing files. (**Tour** history files must be read in separately, while **Tour** is activated.) This is done by means of a popup window which solicits the name of the file to read. (This has not been implemented from within S.)

5.11 Unlink

Select **Unlink** to isolate a particular XGobi window. In this mode, it neither broadcasts nor receives brushing or identification information.

5.12 Help

Click on **Help** for an explanation of XGobi's on-line documentation system. To obtain help for a specific XGobi function, select the widget in question with the right mouse button. (The term "widget" is X jargon, and it refers to a sub-window with some associated functionality. In XGobi, help is available for buttons, scrollbars, the variable selection panel and the plotting window, but not for labels or arrows as of this writing.) This action generates a text window which explains the use and meaning of the widget in question. The help window will remain until you dismiss it by selecting the "Click here to dismiss" button at the bottom of the help window.

5.13 Exit

Close down the XGobi window and terminate the XGobi process.

6 Resources

For X programs, resource files can be used to control size, color, layout, labels and various other program features. The current version of XGobi does not require a resource file, but you may want to manipulate some resources yourself. If so, create a directory called $HOME/app-defaults and create a file named XGobi in that directory. If you already use an XUSERFILESEARCH shell variable, then you know how to add your app-defaults directory to that variable. If you do not, then the simplest course is to set the shell variable XAPPLRESDIR to be $HOME/app-defaults/. For the following examples, type these lines just as they appear here – with the exception that the amount of white space after the colon is not significant.

These lines set the initial width and height of the plotting window and variable selection panel. (The variable selection panel automatically acquires the height of the plotting window.) All numbers below are given in pixels.

```
*XGobi*PlotWindow.width:      300
*XGobi*PlotWindow.height:     300
*XGobi*VarPanel.width:        300
```

These lines can be used to set the diameter of the variable circles and the length of the variable labels.

*XGobi*VarWindow.width:	36
*XGobi*VarLabel.width:	36

The colors used in brushing can be overridden in a private resource file as well. There are ten colors, and they are specified by setting "*brushColor0" through "*brushColor9" to colors by name, such as Red or SlateBlue. The list of colors that can be used can be obtained by executing **showrgb**, **xcol** or **xcolors**.

You might want to set the font used in the buttons and labels or the font used in the plotting window. These can be specified using "*Font" and "*plotFont," respectively. The list of available fonts can be obtained by executing **xfontsel** or **xlsfonts**.

The default plotting glyph can be set as well, using the resource "*glyphId." Any of the integers in the **Glyph Menu** can be used.

Various other sizes, colors, and distances of separation can be set in a private resource file. The default resources are contained in the source file xgobi.c.

6.1 Use of the -name argument

You might want to adjust resources differently for each of several sets of data you are working with. Use the -name command line argument in combination with a private resource file to accomplish that. For example, here are two groups of named resource settings:

usa*brushColor0:	Red
usa*brushColor1:	DarkOrange
usa*brushColor2:	Orange
usa*brushColor3:	Gold1
usa*brushColor4:	SpringGreen2
usa*brushColor5:	Green
usa*brushColor6:	DarkTurquoise
usa*brushColor7:	DodgerBlue1
usa*brushColor8:	Orchid
usa*brushColor9:	Maroon
places*VarLabel.width:	48
places*VarWindow.width:	48
places*title:	XGobi: Places Rated Data

The first group of lines overrides the default brushing colors in favor of a spectrum of colors when the value of "name" is "usa." The second group causes the variable circles and labels to be larger than their default settings and changes the string appearing in the title bar when the value of "name" is "places."

The value of "name" is defined on the command line, for example:

xgobi -name places data/places

If running XGobi from inside S, the name argument is given as a character string, that is:

> **xgobi(placesrated, name = "places")**

Notice that the name argument does not have to be the same as the file name. It needs only to be the same as the prefix of the resources.

7 Future work

Current work on XGobi includes extensions to the "connected lines" function. We've written a rudimentary line editor, and we allow users to provide a file which specifies which points should be connected by lines. We plan to add the "line brushing," allowing lines to be drawn in different colors or line types.

We're also adding the ability to copy the contents of the plotting window into a postscript file for printing.

Currently, erased points are simply not drawn. We plan to allow users to specify that erased points should be excluded from all calculations, from scaling to the projection pursuit guided tour.

8 Acknowledgements

We would like to thank our colleagues at Bellcore and Rutgers who have offered suggestions and feedback about the design of XGobi, especially Allen McIntosh for his invaluable expertise in C language programming and the X Window System. The first XGobi brushing code was written by Bobby Hall during the summer of 1989, the first axis code was written by Nancy Hubbell during the summer of 1990, and Michael Littman wrote the first code to control scaling by mouse actions in the plotting window.

9 Appendix: S Help Page

xgobi	XGobi: A Dynamic Graphics Program for Data Analysis	**xgobi**

xgobi(matrx, collab, rowlab, std='mmx', dev=2.0, vgroups=NULL, colors=NULL, glyphs=NULL, display=NULL, name=NULL, mach=NULL)

ARGUMENTS

`matrx` Any numeric matrix

`collab` Optional character vector of column labels; the default is dimnames(matrx)[[2]]. If no default exists, xgobi constructs its own defaults.

`rowlab` Optional character vector of row labels; the default is dimnames(matrx)[[1]]. If no default exists, xgobi constructs its own defaults.

`std` Optional string; which standardization of view to use. Default is '**mmx**,' minimum-maximum scaling, in which the view is centered at the midpoint of the data, and all the data fits inside the plotting window. Alternatives are '**msd**,' in which the plot is centered at the mean of the data and **dev** standard deviations fit inside the plotting region, or '**mmd**,' in which the plot is centered at the median and **dev** median absolute deviations are shown inside the plotting window.

`dev` Optional numeric scalar; the number of standard deviations (if '**msd**' is chosen) or median absolute deviations (if '**mmd**' is chosen) that will be plotted inside the plotting window. Default is 2.

`vgroups` Optional integer vector, used to assign columns to groups for transformation and axis scaling. This vector must contain one integer for each variable. Columns to be grouped together should share the same integer. Default is the vector 1:(ncol(matrx)).

`colors` Optional character vector, used to supply colors to be used on startup; the default is that all points are the same color.

`glyphs` Optional integer vector, used to supply glyphs to be used on startup; the default is that all points are drawn with the same glyph.

`display` Optional character string, identifying the monitor on which to display the xgobi window. The default is '**machine:0.0**' where 'machine' is the name of the user's workstation. See documentation for xgobi or for X.

`name` Optional character string which defines the '**-name**' argument used by X. See documentation for xgobi, or for X.

`mach` Optional characracter string identifying the machine architecture in use. This is useful on networked file systems as a way to indicate where to find the xgobi executable. Examples are '**sun4**' or '**dec3100**;' default is '**sun3**.'

VALUE

The xgobi S function executes a call to the C program of the same name, an

interactive statistical graphics program which runs under the X Window System, and returns control of the S shell to the user.
XGobi can be used to create vectors of brushing information and rotation coefficients; see the documentation for XGobi for details.

REFERENCE

"User's Manual for XGobi, a Dynamic Graphics Program for Data Analysis Implemented in the X Window System," Swayne, D. F., Cook, D., and Buja, A., Bellcore Technical Memorandum, 1991.

CONTACT

D. F. Swayne

EXAMPLES

```
xgobi(x)
xgobi(cbind(x,y,z), name='laser')
xgobi(x, collabels, rowlabels)
```

References

Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988). *The New S Language: A Programming Environment for Data Analysis and Graphics.* Wadsworth & Brooks/Cole, Pacific Grove, CA.

Buja, A., Asimov, D., and Hurley, C. (1989). Methods for Subspace Interpolation in Dynamic Graphics. Bellcore Technical Memorandum.

Buja, A., Asimov, D., Hurley, C., and McDonald, J. A. (1988). Elements of a Viewing Pipeline for Data Analysis. In Cleveland, W. S. and McGill, M. E., editors, *Dynamic Graphics for Statistics*, pages 277–308. Wadsworth & Brooks/Cole.

Cleveland, W. S. and McGill, M. E. (1988). *Dynamic Graphics for Statistics.* Wadsworth & Brooks/Cole.

Cook, D., Buja, A., and Cabrera, J. (1991). Direction and Motion Control in the Grand Tour. In *Proceedings of the 23rd Symposium on the Interface.* Springer/Verlag.

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

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23:881–889.

Hall, P. (1989). Polynomial Projection Pursuit. *Annals of Statistics*, 17:589–605.

Hurley, C. and Buja, A. (1990). Analyzing High-Dimensional Data with Motion Graphics. *SIAM Journal of Scientific and Statistical Computing*, 11(6):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.

Jones, O. (1989). *Introduction to the X Window System.* Prentice-Hall, Englewood Cliffs, New Jersey.

Peterson, C. D. (1991). *Athena Widget Set – C Language Interface, X Window System.* Massachusetts Institute of Technology and Digital Equipment Corporation. X Version 11, Release 5.

Tukey, J. and Tukey, P. (1990). Strips Displaying Empirical Distributions: I. Textured Dot Strips. Bellcore Technical Memorandum TM-ARH-015809.