XGobi: Interactive Dynamic Data Visualization in the X Window System

DEBORAH F. SWAYNE [1], DIANNE COOK [2] and ANDREAS BUJA [3]

March 5, 1998

XGobi is a data visualization system with state-of-the-art interactive and dynamic methods for the manipulation of views of data. It implements 2-D displays of projections of points and lines in high-dimensional spaces, as well as parallel coordinate displays and textual views thereof. Projection tools include dotplots of single variables, plots of pairs of variables, 3-D data rotations, various grand tours, and interactive projection pursuit. Views of the data can be reshaped. Points can be labeled and brushed with glyphs and colors. Lines can be edited and colored. Several XGobi processes can be run simultaneously and linked for labeling, brushing, and sharing of projections. Missing data are accommodated and their patterns can be examined; multiple imputations can be given to XGobi for rapid visual diagnostics. XGobi includes an extensive on-line help facility.

XGobi can be integrated in other software systems, as has been done for the data analysis language S, the geographic information system (GIS) ArcView™, and the interactive multidimensional scaling program XGvis.

XGobi is implemented in the X Window System™ for portability as well as the ability to run across a network.

Key Words: Data visualization, statistical graphics, interactive graphics, dynamic graphics, linked views, brushing, data rotations, grand tours, projection pursuit, parallel coordinate displays.

1 Preliminaries

This article is intended to be the published standard reference to the XGobi system. The article gives an overview of the functionality of the system as well as a discussion

[1]Deborah F. Swayne is Senior Technical Staff Member, AT&T Labs – Research, 180 Park Ave, P.O. Box 971, Florham Park, NJ 07932-0971.
dfs@research.att.com, http://www.research.att.com/~dfs/

[2]Dianne Cook is Assistant Professor, Department of Statistics, 323 Snedecor Hall, Iowa State University, Ames, IA 50011.
dicook@iastate.edu, http://www.public.iastate.edu/~dicook/

[3]Andreas Buja is Technology Consultant, AT&T Labs – Research, 180 Park Ave, P.O. Box 971, Florham Park, NJ 07932-0971.
andreas@research.att.com, http://www.research.att.com/~andreas/

of its design. Detailed descriptions of XGobi's functionality are in the extensive on-line help facility. Case studies of its use can be found in earlier papers (Buja, Cook, Swayne 1996 , Koschat and Swayne 1996).

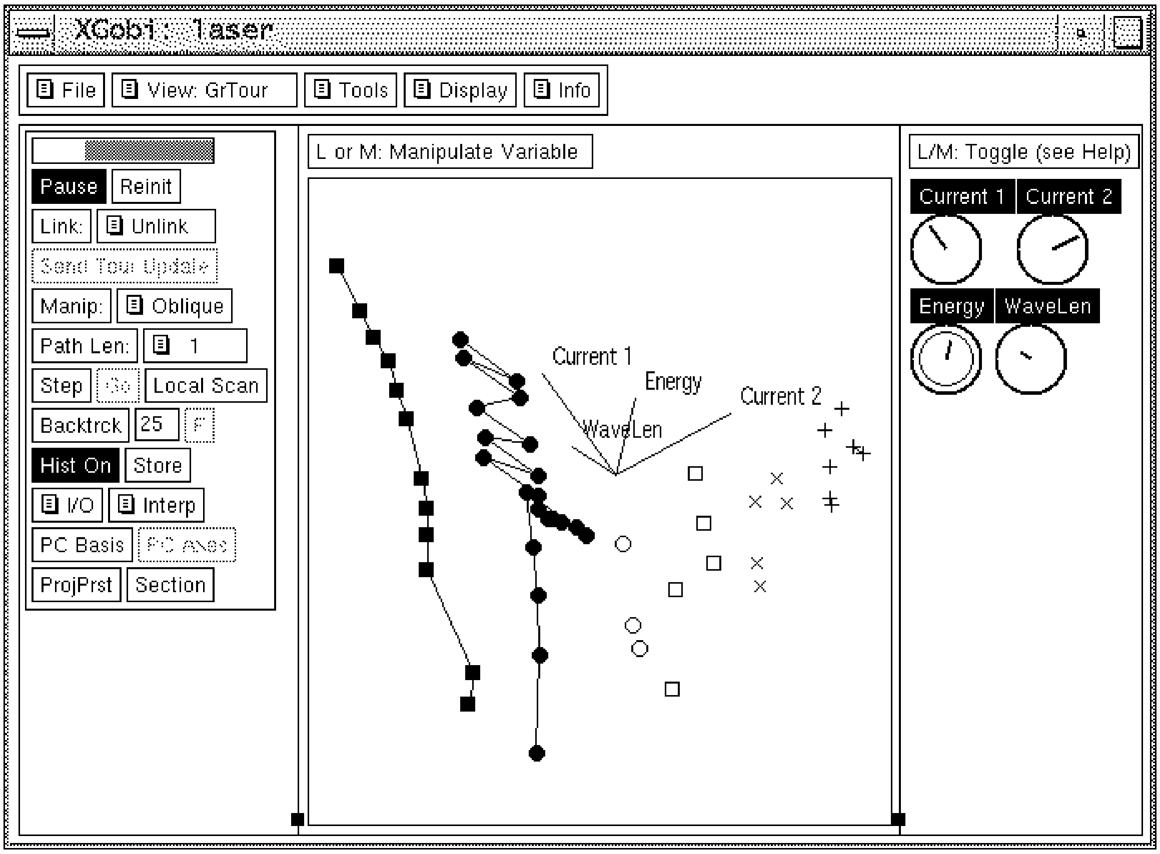

Figure 1: Layout of an XGobi window. The data are 64 measurements of four variables on a laser (Buja et al. 1996). All four variable labels are highlighted, indicating that they all contribute to the current projection of the data. Brushing has been used, and six different glyphs are in use. In addition, line segments have been added to emphasize an irregularity in the data.

XGobi can be used for simple tasks with virtually no instruction. All that a user needs is a cursory knowledge of the developments in interactive statistical graphics of the last 15 years, as well as a willingness to experiment and use the on-line help facility. Help windows for any part of an XGobi window can be popped up with a right mouse click. This allows a user to get going quickly with simple tasks such as querying data with brushing and labeling operations. It is more efficient, though, to acquire up front a basic understanding of the overall functionality of the system. For the greatest success, extensive experience is required; there is nothing easy about a system that implements many concepts and many dozens of features.

We first describe XGobi's layout and functionality and defer the discussion of its design to the end. Figure 1 shows an XGobi window, made using the version that was available in 1997.

2 Layout and functionality

2.1 The major functions

In the 1997 redesign of XGobi, we adopted widespread conventions and organized the major functions in pull-down menus that can be accessed from buttons stretching across the top of the window. Following these conventions, the buttons are named "File", "View", "Tools", "Display" and "Info".

As expected, the "File" button opens a menu for selecting input/output functions as well as exiting. The output functions include saving the colors of points and lines obtained through brushing operations.

"View" opens a menu for selecting from XGobi's interactive graphics operations:

- 1-D dotplots,
- 2-D XY-plots,
- 3-D rotations,
- grand tours with projection pursuit,
- correlation tours with projection pursuit,
- axis scaling,
- brushing,
- identification,
- line editing, and
- moving points.

The unifying feature of these operations is that they take over the display area and dictate the mouse operations for directly manipulating the view. Examples are mouse sweeps to control 3-D rotations, or moves of the brush when brushing. These operations also determine the content of the control panel on the left of the window (see below). The "View" button shows the currently selected "View" mode, here: "Gr-Tour" for grand tour. Exactly one mode is selected at all times. "View" modes are discussed in section 4.

"Tools" gives access to

- window clones,
- smooths and their smoothing parameters,

- subsetting functions for systematic and random subsampling,
- functions for jittering variables,
- a parallel coordinate display,
- a list of all variables (columns, attributes),
- a list of all cases (rows, records),
- functions that deal with missing values, and
- functions for viewing precomputed multiple imputations.

The division between "View" functions and "Tools" is somewhat arbitrary. The "View" functions determine the mouse interactions in the display area, while none of the "Tools" does. Furthermore, "View" functions populate the left hand control panel in the XGobi window, while "Tools" create their own permanent control panels or views of the data in separate windows. For example, the imputation "Tool" generates an imputation control window, shown in the top right of Figure 2 below, and the parallel coordinate "Tool" generates a parallel coordinate window, also shown in Figure 2.

"Display" allows users to determine whether to plot points, lines and/or axes.

"Info" guides the user to the on-line help facility and to general information about XGobi, including papers.

2.2 The display area

The display area in the center of the XGobi window shows the current 1-D or 2-D projection of the data. The user can decide whether to show points, lines and variable axes by toggling items in the "Display" menu. The display area responds to mouse actions according to the current "View" mode. At the bottom is a mouse documentation line that indicates the effects of dragging or clicking the mouse in the display area.

2.3 Variable widgets

At the right is the variable widget panel, in which each variable (column) of the data is represented by one labeled box, called a "variable widget". A highlighted label indicates that the variable can contribute to the current plot. The line inside the circle below the label provides information about the magnitude and direction of that contribution. In Figure 1, the variables named *Current 1*, *Current 2* enter strongly into the current projection, while *Energy* enters less and *WaveLen* even less. The lines in the circles are partly redundant with the variable axes shown in the display area. This redundancy permits us to remove the variable axes in situations where the variable axes interfere with the data display, as is often the case in the tour modes where many variables may contribute to a projection.

Variable widgets also have two important functions as input devices:

- Users can select variables for viewing by clicking on the variable circles. The exact response to clicks depends on the current "View" mode representing the type of projection: 1-D dotplot, 2-D XY-plot, 3-D rotation, grand tour, or correlation tour. A mouse documentation line below the variable widgets indicates the current response to left (L) and middle (M) clicks.
- Users can select variable transformations by depressing the mouse on a variable label and popping up a menu of transformations, including log, a few powers and roots, and discretization to a binary variable, among others.

2.4 Control panel

On the left side of the window is a panel of controls for the functions that are available in the current "View" mode. In Figure 1 the control panel shows the operations available in grand tour mode. The controls in these panels are too numerous to list here, but with a basic understanding of the "View" modes, the user will usually be able to get sufficient information from the help facility.

3 File: input/output

XGobi recognizes a host of input files by filename extension: ascii data (*x.dat*), binary data (*x.bin*), case or row labels (*x.row*), variable or column labels (*x.col*), point glyphs (*x.glyphs*), point colors (*x.colors*), flags for erasing points (*x.erase*), lines (*x.lines*), line colors (*x.linecolors*), variable groups (*x.vgroups*), X resources (*x.resources*), missing flags (*x.missing*), multiple imputations (*x.imp*). Only an ascii or binary data file is compulsory. For details about the file formats, see the help facility or, better, the man page (part of the XGobi distribution), or the third author's XGobi web page.

The "File" menu provides for input of some of these files, and for output through two types of save operations:

- adding files to the current file set, such as saving a colors file (*x.colors*) after interactively brushing the points with colors;
- creating a new file set with fewer or permuted cases and/or variables.

4 View modes

In this section we discuss each view mode in general terms. Again, details of use can be found through on-line help.

4.1 Dotplots

XGobi includes a 1-D dotplot display based on lateral jittering. We wanted a method based on the dot plot so that it would be consistent with XGobi's presentation of cases as points and hence allow interactive identification of cases with labels. The dotplot method we adopted was proposed by Tukey and Tukey (1990) and called "textured dot strips." They use a lateral jittering method that is partly constrained and partly random. This results in a fairly smooth spreading of the points, which minimizes the artifacts seen in purely random jittering and focuses the user's eyes on changing patterns in data densities.

As in all modes, the plotted variable is chosen by clicking the respective variable widget. The subordinate control panel for the dotplot mode allows cycling through all possible 1-D dotplots at a speed chosen by the user.

4.2 XY-plots

XY plots show the usual Cartesian display of two variables, chosen with a left and a middle click on the respective variable widgets. The control panel for the XY plot mode allows cycling through all possible 2-D XY plots of variable pairs at a speed chosen by the user.

4.3 3-D rotations

This is the familiar 3-D rotation in the space spanned by three variables. Controls exist for speed, pause, and rocking, among others. Three types of rotation can be chosen:

- rotation around a fixed vertical variable,
- rotation around a fixed horizontal variable, or
- rotation around oblique axes.

The rotation can be controlled in a direct manipulation fashion by dragging the mouse in the display area. No depth cues are provided at this point.

A subtle issue arises in the interactive selection of variable triples that define the 3-space: When selecting a new variable, it is not clear which among the three previously selected variables to replace. We solved the problem by marking the candidate for replacement with a small circle. To replace a variable, the user first clicks on one of the currently selected variables to designate it as the target for replacement, then clicks on the new variable. The replacement marker is visible only when the mouse hovers over the variable widgets.

4.4 Grand tours with projection pursuit and manual tours

The grand tour is a generalization of 3-D rotations whereby dynamic 2-D projections of 3-space are replaced with dynamic 2-D projections of p-D space, where p can be as large as the viewer's brain can handle — or larger. The original grand tour (Asimov 1985) is meant as an automatic animation for viewing all of data space. In the present implementation it is much more: Users can select arbitrary variable subspaces by clicking on variable widgets; they can pause, backtrack, control speed, do scans of local neighborhoods, alternate touring with exploratory projection pursuit, manually drag any variable into or out of the projection by dragging the mouse in the display area, freeze variables so that their contributions to the projection stay fixed during the tour, and link tours running in different XGobi windows.

Simple clicks on variable widgets toggle the respective variables in and out of the projected subspace. Again, we encountered a human interface issue: If simple clicks already have a function, what kind of click operation should be used to select the variable to be manipulated in manual operation? Inelegantly, we resorted to a modifier: Depressing <Shift> and clicking on a variable widget specifies the manipulated variable (<Shift>-L or <Shift>-M). Dragging the mouse on the display area has the effect of rotating the data in the space spanned by the current 2-D projection and the drag variable. Similarly, depressing <Ctrl> and clicking on a variable widget will freeze that variable's contribution in the tour.

The wealth of operations that go with grand tours, projection pursuit guided tours, as well as manual tours, is too great to be covered here. See Cook, Buja, Cabrera and Hurley (1995), Buja, Cook and Swayne (1996), and Cook and Buja (1997) for more.

4.5 Correlation tours with correlation pursuit and manual tours

The correlation tour is to the grand tour what canonical correlation analysis is to principal component analysis: Both the grand tour and principal components have a symmetric view of the variables; in contrast, the correlation tour and canonical variates require a division of variables into two groups, called X-variables and Y-variables, respectively. The difference is:

- Correlation tours plot linear combinations of the Y-variables against linear combinations of the X-variables.
- Grand tours plot two orthogonal linear combinations of *all* variables against each other.

The correlation tour has controls that are analogous to those of the grand tour: Users can select spaces of X-variables and spaces of Y-variables with left (L) and middle (M) clicks on variable widgets; they can pause, backtrack, control speed, alternate touring with a form of projection pursuit that optimizes correlation, manually drag any pair of variables into or out of the projection by dragging the mouse in the display area, and freeze variables in their current contribution to the projection.

Because simple left and middle clicks on variable widgets toggle variables in and out of the X- and Y-spaces, we implemented selection of variables for manual operation again with a modifier key: Depressing <Shift> while clicking left (L) or middle (M) selects the manual X- or Y-variable, respectively. Dragging the mouse in the display area will then amount to a genuine 4-D rotation whereby a rotation of the horizontal axis is combined with a rotation of the vertical axis. For more background, see Buja, Asimov, Hurley and Donald (1988) and Hurley and Buja (1990), as well as the references cited in the preceding subsection. Also, the pictures of sections 3 and 5 in Buja, Cook and Swayne (1996) were generated with manual operation of the correlation tour.

4.6 Scaling of axes and variables

Axis scaling is provided in a separate "View" mode in which pan, zoom and the aspect ratio of plots can be changed interactively and dynamically. Buttons can be depressed to scroll, shrink and stretch axes continuously. The same can be achieved in a direct manipulation mode by dragging the mouse on the display area and depressing the left (L) button for panning or the middle (M) button for scaling.

Another selection provided in this mode is the type of variable standardization. Currently variables can be centered and scaled according to (1) range, (2) mean and standard deviation, and (3) median and MAD. Scaling according to the range is the default as is the convention in computer graphics, if only to fit all data into the window.

Note that variable scaling differs from axis scaling in systems that provide general projections of data: As in principal component analysis, variables need to be made commensurate, that is, when variable units are pounds, miles, days, it needs to be decided how many inches in virtual data space each unit is to take. In mathematical terms, data space has to be provided with a Euclidean metric, which is what variable scaling does. Axis scaling then determines the shape of the viewing window into the projection plane, as well as the aspect ratio of the horizontal to the vertical axis.

Frequently we encounter the requirement to scale variables identically, especially when the variables have the same units. This is accommodated by the ability to define variable groups in a special input file (*x.vgroups*). Variables in the same group are scaled and transformed identically.

4.7 Linked brushing of points and lines

In brush mode, a user can manipulate a rectangular "paintbrush" to change the glyphs and/or colors of points under the brush, and/or the colors of lines. Similarly, the user can brush to erase points. The brush is moved by clicking or dragging the left (L) mouse button, and reshaped by dragging the middle (M) mouse button. Following Becker and Cleveland (1987), XGobi has both persistent (sticky) and transient (non-sticky) brushing.

The real power of brushing stems from linking multiple XGobi windows: As a user brushes points and/or lines in one XGobi window, the changes are automatically reflected in all other "linked" XGobi windows. By default, windows are linked when their data have the same number of cases. This adds a degree of flexibility because it allows users to do linked brushing when datasets have the same cases but different variables; it introduces small risks in rare circumstances when two unrelated datasets have by chance the same number of cases and should not be linked. Linking, however, can be selectively disabled for point glyphs, point colors, line colors, and erasing.

It is possible to link XGobi windows with data having different numbers of cases by using the *x.nlinkable* file, which specifies that the first *nlinkable* points (cases) should be considered available for linking.

Point glyphs, point colors, line colors, and erase flags can be saved to or read from files.

4.8 Linked identification of points with labels

Identifying cases is done by sticking labels to points. When the mouse moves over the display area, the point nearest to the mouse shows a transient label. A mouse click makes the label persistent. By default, identifying is linked across XGobi windows under the same criterion as brushing, and linking can be disabled on demand.

Case labels can be passed to XGobi in a file, but in the absence of such a file, labels showing the sequential case (row) number are used.

4.9 Line editing

Line editing mode is for interactively adding and deleting lines that connect points in the data. In "Add" mode, one attaches a line by dragging the left mouse from one point to another. In "Delete" mode, the line nearest to the mouse is highlighted; a left click deletes the line.

Data for line drawing can be passed to XGobi in a file (*x.lines*). In the absence of the file, the default is that the points are connected with lines in sequential order, which is handy when the order of the cases reflects time order of the data.

4.10 Moving points

Points or groups of points can be manually moved by dragging them with the mouse. Points are always moved parallel to the current projection plane in data space. This capability, together with brushing and line editing, allows users to literally draw pictures in high-dimensional spaces.

5 Tools

The "Tools" menu contains functions that can be activated without changing the "View" mode. Selections from "Tools" do not change the mouse interactions in the display area. They generate permanent new windows containing controls for data manipulations or other types of views of the data.

5.1 XGobi Clone

Selecting this tool opens a new XGobi window with its own process using the current data and the current features, such as glyphs, colors, and labels.

This is especially useful if XGobi has been started from within another software package where the data is stored in that package, such as from within S or ArcView™. In this case, cloning is the simplest way, and perhaps the only way, to have multiple linked views of the same set of data.

ArcView is a trademark of Environmental Systems Research Institute, Inc.

5.2 Smooths

Simple smooths can be added to enhance the scatterplots. A running mean or median smoother is computed either for all the data or for each color group separately. A slider provides interactive control over the width of the smoothing window.

5.3 Subsetting

The main purpose of subsetting is to provide increased efficiency of interaction for data sets with large numbers of cases. This control window permits users to subset the data by

- specifying a block of contiguous cases (rows, records),
- randomly sampling a specified number of cases without replication (the *-subset* command line argument applies this method at startup),
- selecting every n'th case in consecutive order,
- selecting the set of cases labeled in "Identify" mode or highlighted in the case list window,
- selecting all cases with the same label (assuming that there are multiple labels denoting groups of cases).

5.4 Jittering

Visual jittering can be achieved by adding small random numbers to the data. The amount of jittering can be interactively controlled, and both uniform and normal random numbers are available. Jittering is useful for scatterplots of categorical (especially binary) variables to prevent overstrike. Linked brushing as well as rotation of jittered categorical variables can be surprisingly effective (Buja, Swayne, Cook 1996).

5.5 Parallel coordinate window

XGobi provides on demand an auxiliary window with a parallel coordinate display. In parallel coordinate displays, p variables are represented by p axes, plotted parallel and side by side; cases are represented by profiles, each consisting of $p-1$ lines that connect the p variable values of a case across the p axes. See Figure 2 for an example in XGobi's implementation. For references see Inselberg (1985) and Wegman (1990).

Parallel coordinate displays were implemented in XGobi due to a need to display longitudinal, repeated measures and panel data as functions of time (Koschat and Swayne 1996). In such data, the variables arise from a single variable which is observed multiple times on a collection of cases or individuals. Parallel coordinate displays then simply amount to multiple time series plots, if the variables are shown in time order.

In general, parallel coordinate displays require a decision about the order in which to line up the axes. For multiple time series data, this order is dictated by time. For general data, no natural order may exist and an arbitrary decision is required. In XGobi, it is the order of the variables in the data file.

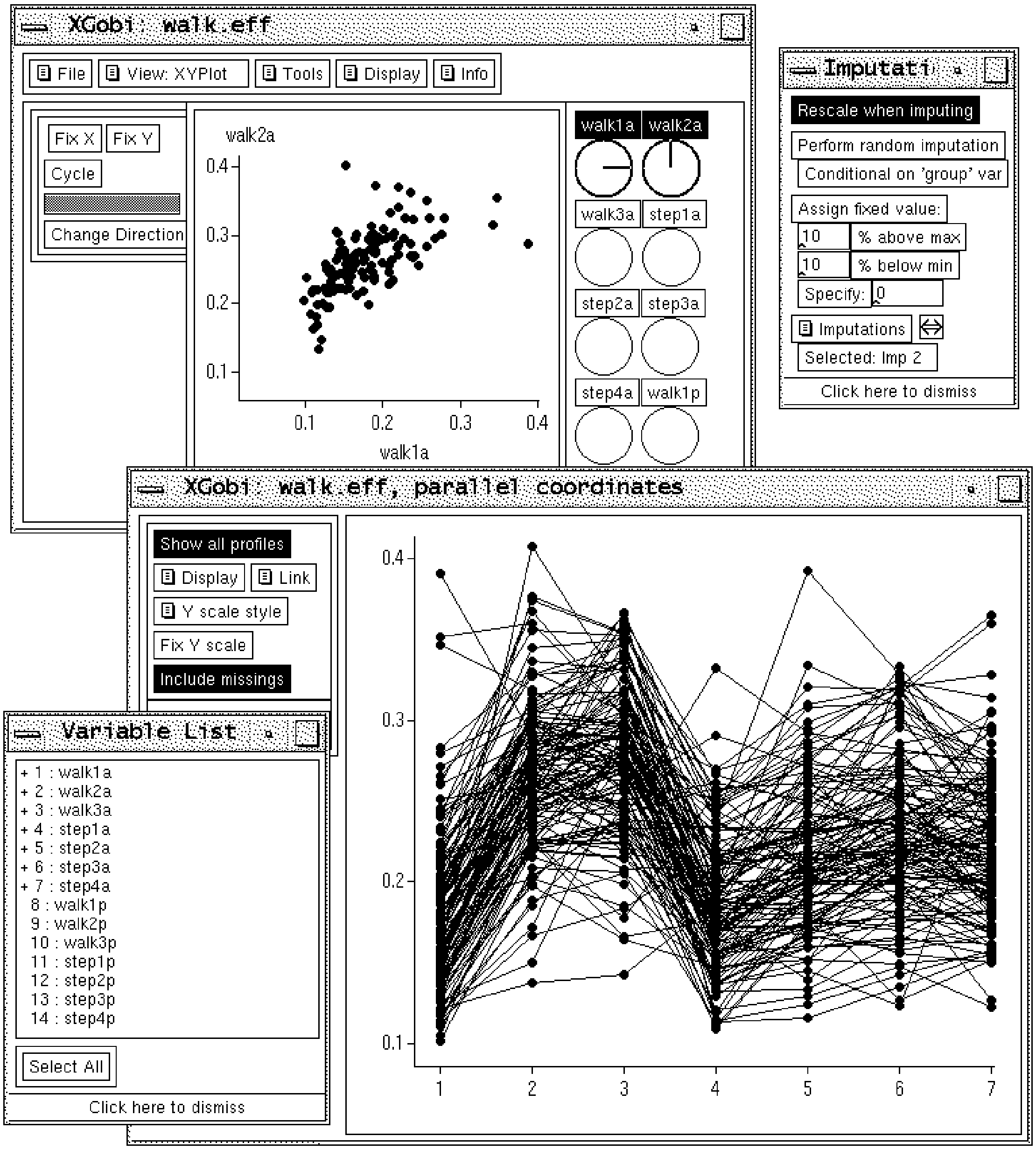

Figure 2: An XGobi window with its parallel coordinate window. Also shown are two control panels: imputation controls (top right) and variable selection controls for the parallel coordinate display (bottom left). Seven out of fourteen variables are marked in the Variable List, and only those seven are included in the parallel coordinate display.

When there is a need to limit the parallel coordinate display to a smaller subset of the variables, variable selections can be made in the "Variable List" window (accessible from "Tools"), as has been done in Figure 2.

Parallel coordinate displays suffer from the "spaghetti problem", by which we mean the effect that even a small amount of data — such as 300 cases — can swamp the display beyond recognition. Even when completely overcrowded, however, parallel coordinate displays retain their usefulness if embedded in an interactive system with highlighting operations. Highlighting intelligible subsets of cases in parallel coordinates is an efficient way of comparing values across many variables and detecting local correlations.

For historical reasons, XGobi's parallel coordinate display is a slave to the primary XGobi window, which is why the parallel coordinate display is not equipped with interactive point and click operations. It responds, however, to identification and color brushing in the primary XGobi window through linking. It may be mildly inconvenient that we do not provide direct manipulations on the parallel coordinate display, but their effects can be easily emulated by equivalent manipulations on dotplots and XY-plots in the primary window.

The operation of the parallel coordinate window is in two ways:

- **Lens mode — viewing few cases in isolation**: One starts with a blank display area and selects individual cases with the "Identify" operation in the primary window. In this mode, the parallel coordinate display acts as a "lens" into small interactively selected subsets of the data; it can hence be used for data sets as large as XGobi can handle (a few 10,000 depending on the hardware; see below).
- **Global mode — viewing all cases**: The display shows all cases. One performs color brushing in the primary window in order to highlight subsets of interest. This mode is generally less useful for very large data sets due to the spaghetti problem as well as insufficient drawing speed.

Users can easily toggle back and forth between the two modes with a click on the button marked "Show all profiles". The default is the first mode (starting up with a blank parallel coordinate display), the reason being that on very large data sets the display might clog up the system if applied to all cases.

5.6 Variable list window

Currently, this window is solely a control panel for variable selection in the parallel coordinate display. By clicking or dragging on the list of variable names, one can add or remove variables from the parallel coordinate display. In future releases of XGobi, the functionality of this window may be extended.

5.7 Case list window

This is an auxiliary window showing a scrollable list of all case labels. The window is linked to the primary XGobi window and the parallel coordinate display: Highlighting cases in the list, with clicking or dragging, causes their labels to appear in the XGobi window, and their profiles to be drawn in the parallel coordinate display (when in lens mode). The case list window has two modes of operation: a transient mode in which only one case is highlighted at any time, and a persistent mode in which arbitrary numbers of points can be highlighted.

5.8 Missing data XGobi

Missing value patterns are multivariate: Each case can have missing values on a different set of variables. In order to examine patterns in missing values and uncover correlations between missingness and covariates, users can launch a new XGobi window which displays jittered binary dummy data of the same size as the original data and where the high value indicates missingness. XGobi recognizes missing data in one of two ways:

- coded as "NA" or "na" or "." in the data file (*x.dat*);
- coded as a data matrix in an auxiliary file (*x.missing*) of the same size as the data; a non-zero value such as 1 indicates a position of a missing value.

The latter file exhibits a couple of creative aspects:

- In positions where the indicator file shows a missing value, the data file may still contain a value.
- The indicator file can contain values other than 0 and 1. Any value other than 0 marks the corresponding data as missing or otherwise exceptional.

This has a few interesting uses:

- "Missing" may mean "existing but disqualified or otherwise unreliable." In this case, the data file allows one to retain these values for exploration, and at the same time they are marked as exceptional in the indicator file.
- "Missing" may really be "censored." In this case, the data file allows one to retain the censored values for analysis and the indicator file marks their positions.
- In applications with more than one type of missingness or censoring, the indicator file may contain more than two levels. For example, 1 could mean left censoring and 2 right censoring.

Functionality related to missing values is the topic of Swayne and Buja (1997).

5.9 Imputation controls for missing data

In order to display missing data, some form of imputation is necessary. We provide three functions related to imputation:

- **Imputation of fixed values**, either directly specified by value, or in terms of a percentage above the maximum or below the minimum of each variable. This is convenient for plotting cases with missing values outside the variable ranges on the margins of the scatterplots.
- **Random imputation** on a per-variable basis. For each variable, missing values are imputed by randomly sampling existing values. This is essentially non-parametric imputation using the model of independent variables and randomness of missings. Each mouse click generates a new imputation. The method is meant for graphically checking whether the dependence structure should be used in more sophisticated imputation procedures.
- **Random imputation conditional** on a categorical variable. The categorical conditioning variable can be defined by brushing operations, or it can be imported in a color or glyph file. This corresponds to imputation under the assumption that, conditional on the categories, the variables are independent and the missings are random.
- Users can place **multiple sets of imputations** in specific data files which XGobi recognizes (*x.imp*). The imputations are to be computed elsewhere, either with custom code or in a modeling package. In XGobi, users can cycle through the supplied imputations and diagnose their adequacy graphically. (The precise format of *x.imp* files is in the man page as well as the help facility for the imputation control window.)

More about imputation functionality can be found in Swayne and Buja (1997).

6 Use of XGobi on large data

XGobi can be used on data of a size that is not usually considered within the reach of interactive data visualization. There exist two meanings of "large": large N (number of cases, records) and large P (number of variables, attributes). The problems posed by each are different and need to be discussed separately.

6.1 Large N

Data with large numbers of cases can cause problems in several ways:

- **Ascii input** may delay startup, sometimes to an intolerable degree.
- **Realtime processes** such as data rotations and brush dragging may get slow.
- **Overplotting** may cause loss of information in the views.

Each of these obstacles can render a viewing system useless. Each obstacle, however, can be countered with partial remedies that make XGobi useful for N up to about 500,000 cases on state-of-the-art computers at the time of writing. Interactive operation when N is in excess of 10,000, however, requires a certain amount of adjustment from the user:

- XGobi can be started with a command line argument (*-subset* n) that randomly subsamples the data to be displayed. The subsetting menu in "Tools" can be used to change the sample size and generate new samples during the session. The possibility of repeatedly and quickly generating subsamples takes some of the edge off the drawback of viewing only part of the data.
- In order to avoid ascii input, XGobi allows users to input binary data files (*x.bin* as opposed to *x.dat*), which may be created externally or else within XGobi, after enduring a long startup from an ascii file. All other files must be input in ascii, but most do not cause significant delays. Two notable exceptions are color files and case label files (*x.colors* and *x.row*), which should be avoided for sizable data. The absence of a color file can be partly remedied with a hack: Add an additional categorical variable that codes the color groups, and brush these groups interactively.
- Realtime processes such as data rotations slow down proportional to N. Favorable proportionality constants can be achieved in some cases: Rotation speeds can be improved by using single pixel points; complex glyphs are more expensive to draw. Brushing performance can be improved by brushing in discrete steps with clicks, as opposed to dragging of the brush with the mouse depressed. The brush should be reshaped while its action is temporarily turned off ("Brush Off").
- Overplotting can be minimized by choosing single pixel points and large windows.

6.2 Large P

Data with large numbers of variables do not strain the computational resources, but they may cause cognitive overload in the viewer. If P is 100, say, the viewer needs to have a good idea of where to look, or some form of numerical aggregation or dimension reduction outside of XGobi may be necessary. When the number of relevant variables is reduced to a few dozen, XGobi generally proves useful.

Sometimes the number of variables can be greater than a few dozen, for example, when the variables have simple meanings, as in multiple time series or multiple periodograms, where the variables are measurements of the same quantity at different times or frequencies. In these situations, parallel coordinates are a natural display method: They act as multiple time series plots or multiple periodogram plots.

A powerful graphical interface for rapid selection of variables is provided by the variable widgets. We have had applications with 128 variables and a corresponding number of widgets shown in an XGobi window. The obvious problem here is that the widgets take up too much screen space to be shown at all times, even if the window is sized to fill the whole screen. This problem is partly solved by allowing users to trade off space between the three major areas of the window — control widgets on the left, display area in the middle, and variable widgets on the right — by dragging the small square grips at the bottom between the areas.

An additional method for better packing the available space in an XGobi window is provided by options in the resource file (*x.resources*): There, it is possible to set parameters that reduce the size of the variable widgets (see the man page).

7 Design of the user interface

It is common place that users are unwilling to spend much time on learning and memorizing a tool; they may use it only occasionally and, unlike a touch typist, they rarely get to the point of automated operation. Therefore the design of XGobi's user interface has received a great deal of our attention. For basic operations, we tried to avoid complicated mouse operations as much as possible, and we incorporated extensive on-line help, accessible by clicking (R) on most content areas of interest.

Simplicity of user interface design is in tension with another goal for XGobi: that of providing us, the designers, with a test bed for new visualization and interaction methodology. For example, our interest in high-dimensional dynamic projections led us to include grand tour and projection pursuit methods whose inherent complexity is hard to reconcile with a simple user interface. In addition, the lack of experience with human controls for some of these methods leads to a two-stage process, whereby we initially implement an overly complex interface which later gets simplified after experience has been gained. Currently we can say that, for example, basic variable selections and deselections in grand tours are easily handled even by the uninitiated user. Productive use of this tool is another matter and depends heavily on data analytic sophistication.

8 Implementation

XGobi is written in C, using X Window System programming libraries. It runs wherever X runs: on workstations and PCs running various dialects of UNIX. It is also possible to run XGobi on a UNIX server and display it on a client computer: either a UNIX client or a PC running Microsoft Windows or the Macintosh operating system.

The X Window System provides interprocess communication that can be used for linking of multiple views, a functionality which is essential for data visualization. Through named locations in memory, different processes can share data. Using this method for linked brushing, for example, one XGobi process writes vectors of glyph shape and color to the designated memory locations, then a second process detects that a change has occurred and reads the new values.

9 Integration of XGobi in other software systems

There exist two mechanisms for embedding XGobi in other software:

- XGobi can be called with an external system call and a file interface.

- XGobi can be called as a C function.

In the first method, the data of interest are written out to temporary files; XGobi is then invoked with an external system call and reads in the temporary files. In the second method, a C program must be written to initialize the data structures to be passed to the XGobi function.

The first approach was chosen for embedding XGobi in the S data analysis environment (Becker, Chambers and Wilks 1988), and also for cloning XGobi (see "Tools" above).

The second approach was applied in XGvis, an interactive multidimensional scaling (MDS) system (Littman, Swayne, Dean and Buja 1992). It uses XGobi for algorithm animation of MDS optimization, and it allows users to tour MDS configurations in arbitrary dimensions.

Another application of this type is described by Klein and Moreira (1994) who link XGobi to an image manipulation system in order to explore Landsat and other spatial images.

An extension of this approach was used to establish an interface between XGobi and the ArcView geographic information system (Symanzik, Majure, and Cook 1996). A custom program calls XGobi as a function and communicates with ArcView using so-called Remote Procedure Calls (RPC).

Current work by Symanzik et al. (1997) puts the RPC mechanism directly into XGobi, which will allow it to communicate with any other software that supports RPC protocols.

See Swayne, Cook and Buja (1991) for further discussions of XGobi integration.

10 History

XGobi is the descendent of an earlier system called "DataViewer" written by the third author during the second half of the 1980s at the University of Washington in Seattle and at Bellcore. This predecessor system was an experimental prototype programmed on a Symbolics Lisp machine. It implemented linked brushing and labeling, various tour methods, parallel coordinates and several types of Andrews curves, as well as visual permutation test ideas, partly documented in Buja, Asimov, Hurley and McDonald (1988). Hurley's thesis (Hurley and Buja 1990) brought the culmination of this work by integrating tours and multivariate analysis.

In early 1989, when the X Window System was becoming the windowing standard on Unix workstations, the present authors teamed up with the intention of continuing visualization research in a hopefully more successful programming and graphics environment. These efforts were generously supported by Jon Kettenring whose unobtrusive guidance was crucial to this endeavor. Originally, the new system was called XDataViewer or XDV, but repeated collisions with names already in use forced us finally to choose an oddball name: XGobi.

In 1996 and early 1997, a rush of new development led to the current release, with the redesign of the user interface, the addition of manual controls for tours, features for missing values, a true parallel coordinate display, a new input/output scheme, among others.

11 On-line Resources

The distribution version of XGobi is available from Statlib at CMU. More up to date releases can be downloaded as shar files from

http://www.research.att.com/~andreas/xgobi/
ftp://ftp.research.att.com/dist/xgobi/

For further literature about XGobi, see the authors' web pages,

http://www.public.iastate.edu/~dicook/
http://www.research.att.com/~andreas/

and for videos see the ASA Statistical Graphics Section's video lending library:

http://www.amstat.org/sections/

Acknowledgments

We'd like to thank the following contributors of XGobi code: Bobby Hall, Nancy Hubbell, Michael Littman, Jürgen Symanzik, Philip G. Jones, James Brook, Inna Megretskaia, and Sigbert Klinke.

References

[1] Asimov, D. (1985), "The grand tour: a tool for viewing multidimensional data," *SIAM J. Sci. Statist. Computing* **6** 1, pp. 128–143.

[2] Becker, R. A., Chambers, J. M., Wilks, A. R. (1988), *The New S Language*, Pacific Grove, CA: Wadsworth & Grove.

[3] Becker, R. A., and Cleveland, W. S. (1987), "Brushing scatterplots," *Technometrics* **29**, pp. 127-142; reprinted in Cleveland and McGill, eds (1988), *Dynamic Graphics for Statistics*, Belmont, CA: Wadsworth, Inc., pp. 201-224.

[4] 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 and M. E. McGill, Belmont, CA: Wadsworth, pp. 277-308.

[5] Buja, A., Cook, D., and Swayne, D. F. (1996), "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics* **5**, pp. 78–99. A companion video tape can be borrowed from the ASA Statistical Graphics Section. lending library.

[6] Cook, D., Buja, A., Cabrera, J., and Hurley, H. (1995), "Grand tour and projection pursuit," *J. of Computational and Graphical Statistics* **2** 3, pp. 225–250.

[7] Cook, D., and Buja, A. (1997), "Manual controls for high-dimensional data projections," *J. of Computational and Graphical Statistics*, to appear.

[8] Hurley, C., and Buja, A. (1990), "Analyzing high-dimensional data with motion graphics," *SIAM Journal on Scientific and Statistical Computing*, **11** 6, pp 1193–1211.

[9] Inselberg, A. (1985), "The plane with parallel coordinates," The Visual Computer 1, New York: Springer, pp. 69–91.

[10] Koschat, A. M., and Swayne, D. F. (1996), "Interactive graphical methods in the analysis of customer panel data" (with discussion), *Journal of Business and Economic Statistics*, **14** 1, pp. 113–132.

[11] Klein, R., and Moreira, R. I. (1994), "Exploratory analysis of agricultural images via dynamic graphics," Technical Report, Laboratório Nacional de Computacão Científica – LNCC, Rio de Janeira, Brazil.

[12] Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992), "Visualizing the embedding of objects in Euclidean space," *Computing Science and Statistics: Proc. of the 24th Symp. on the Interface*, Fairfax Station, VA: Interface Foundation of North America, Inc., pp. 208–217.

[13] Swayne, D. F., Cook, D., and Buja, A. (1991), "XGobi: Interactive dynamic graphics in the X Window System with a link to S," in *ASA Proc. of the Section on Statistical Graphics*, pp. 1–8.

[14] Swayne, D. F., and Buja, A. (1997), "Missing data in interactive high-dimensional data visualization," *Computational Statistics*, to appear.

[15] Symanzik, J., Majure, J., and Cook, D. (1996), "Dynamic graphics in a GIS: a bidirectional link between ArcView 2.0 and XGobi," *Computing Science and Statistics: Proc. of the 27th Symp. on the Interface*, Fairfax Station, VA: Interface Foundation of North America, Inc., pp. 299–303.

[16] Symanzik, J., Kötter, T., Schmelzer, S., Klinke, S., Cook, D., Swayne, D. F. (1997), "Spatial Data Analysis in the Dynamically Linked ArcView/XGobi/XploRe Environment," *Computing Science and Statistics, Proc. of the 29the Symp. on the Interface*, Fairfax Station, VA: Interface Foundation of North America, Inc.

[17] Tukey, J. and Tukey, P. (1990), "Strips Displaying Empirical Distributions: I. Textured Dot Strips," Bellcore Technical Memorandum.

[18] Wegman, E. J. (1990), "Hyperdimensional data analysis using parallel coordinates," *Journal of the American Statistical Association*, **85** 411, pp. 664-675.