Linking ArcView 3.0$^{\mathrm{TM}}$ and XGobi: Insight Behind the Front End

Jürgen Symanzik[1], Dianne Cook[2], Nicholas Lewin–Koh[2], James J. Majure[2], Inna Megretskaia[2]

[1] Utah State University, Department of Mathematics and Statistics, Logan, UT 84322–3900, USA

[2] Iowa State University, Department of Statistics, Ames, IA 50011, USA

symanzik@sunfs.math.usu.edu

Abstract

This paper presents aspects of the implementation of a bidirectional link between the Geographic Information System (GIS) ArcView 3.0$^{\mathrm{TM}}$ and the interactive dynamic statistical graphics program XGobi. We describe the main functionality of the link, the underlying Remote Procedure Call (RPC) mechanism, and internal data structures, and discuss topics such as security, concurrency, and linked brushing. We think that these topics are of particular interest to software authors intending to link similar software packages, and software users learning about strengths (and weaknesses) of the implementation of our link.

Keywords: Cartographic data; Dynamic statistical graphics; Geographic information system; Interprocess communication; Multivariate spatial data; Spatial cumulative distribution function; Spatial statistics; Spatially lagged scatterplot; Variogram–cloud plot.

1 Introduction

Geographic Information Systems (GISs) are used extensively for examining spatial data but, traditionally, they have few facilities for exploratory spatial data analysis. Likewise, dynamic statistical graphics software packages are used primarily for exploratory data analysis and have little of the sophisticated cartographic tools necessary for displaying geographic information. The lack of exploratory tools in a GIS is especially critical when there are multiple attributes measured at each spatial location, and the lack of cartographic tools in dynamic graphics software is critical in the presence of extensive data bases of geographic information.

To gain access to sophisticated dynamic graphics tools from an extensive GIS, we developed a bidirectional link between a GIS, ArcView, and a dynamic graphics program, XGobi (Swayne, Cook & Buja, 1998). ArcView is a widely–used

ArcView 3.0 is a trademark of Environmental Systems Research Institute, Inc.

and established GIS. XGobi is a dynamic graphics program which is especially powerful for exploring multivariate data. A seamless link is a natural way to obtain the desired capabilities, for several reasons: (1) ArcView has an established clientele but the graphical design makes it virtually impossible to build in the capabilities of XGobi, (2) XGobi's visual paradigms make it almost impossible to build in the sophisticated mapping tools available in ArcView, and (3) we use both packages in our own data analyses and the link is a way to streamline the sharing of information between the two programs. Both packages are unique and complementary. Their paths meet in the current climate of increasing amounts of multivariate geographically referenced information. ArcView has some rudimentary tools for establishing InterProcess Communication (IPC) through its Remote Procedure Call (RPC) mechanism, the basis for our link.

In general, seamless software links through IPC allow the rapid prototyping of cross–disciplinary methodology into complex software systems without having to re–implement existing individual software parts. This paper describes how RPCs were built into XGobi and the modifications made to ArcView which allowed the two packages to share data. It will be informative for software authors who wish to link other software packages and also to users wishing to learn about the strengths and weaknesses of our implementation.

The next section discusses the literature on geographic visualization and dynamic statistical graphics. In Section 3 we give a short summary of IPC standards and, in particular, the RPC mechanism. Section 4 provides an introduction to the capabilities of the link between ArcView 3.0 and XGobi. Section 5 describes our implementation in detail. We finish with a discussion in Section 6.

Our ArcView/XGobi link was developed on DEC™ alphastations, and tested on Sun™/Sparc™, SGI™, and Data General workstations. A current version of the software can be downloaded from the web at the URL

`http://www.public.iastate.edu/~arcview-xgobi/homepage.html.`

A description of the software from the user's point of view is given in Symanzik, Majure & Cook (1997c). Various pieces of this work have been documented in conference proceedings. This paper represents a comprehensive description of what is required to get an RPC link running.

2 Cartographic and Dynamic Statistical Graphics

Linking statistical plots with geography for analyzing spatially referenced data has been discussed widely in recent years. Monmonier (1988) describes a conceptual framework for geographical representations in statistical graphics and

DEC is a trademark of Digital Equipment Corporation.
Sun is a trademark of Sun Microsystems, Inc.
Sparc is a trademark of Sun Microsystems, Inc.
SGI is a trademark of Silicon Graphics, Inc.

introduces the term *geographic brushing* in reference to interacting with the map view. But geographic brushing does not only mean pure interaction with the map. In addition, this term has a much broader meaning, e. g., finding neighboring points and spatial structure in a geographic setting.

A good overview of dynamic graphics for multivariate data is given by Buja, Cook & Swayne (1996). In addition, many software solutions have been developed for exploring multivariate spatially referenced data. We describe a sampling of this work here.

In McDonald & Willis (1987), a grand tour (Asimov, 1985; Buja & Asimov, 1986) is linked to an image to assess the clustering of landscape types in the band space of a LandSat image taken over Manaus, Brazil. In Carr, Littlefield, Nicholson & Littlefield (1987) and Monmonier (1989), a scatterplot matrix is linked to a map view. In REGARD (Unwin, Wills & Haslett, 1990), map views are linked with histograms and scatterplots and, moreover, diagnostic plots for assessing spatial dependence are also available. Another exploratory system that links histograms and scatterplots with latitude and longitude (and depth) coordinates is discussed in MacDougall (1992). In Carr, Olsen & White (1992), (bivariate) ray–glyph maps have been linked with scatterplots. Klein & Moreira (1994) report on an interface between the image program MTID and XGobi, used for the exploratory analysis of agricultural images. DiBiase, Reeves, MacEachren, von Wyss, Krygier, Sloan & Detweiler (1994) provide an overview on existing multivariate (statistical) displays for geographic data. Some recent developments are the cartographic data visualizer, `cdv` (Dykes, 1996), where a variety of plots are linked with geography, the Space–Time–Attribute Creature/Movie, STAC/M (Openshaw & Perrée, 1996), that searches for patterns in GIS data bases under the control of a Genetic Algorithm, and an exploratory spatial analysis system in XLisp–Stat (Brunsdon & Charlton, 1996). The reader of these references should note that in most geographically influenced publications authors distinguish between *geographic (or cartographic) space* and *attribute (or data) space* but rarely use the statistical expression *variable* to relate to the latter one.

In addition to the ArcView/XGobi link, there are several other examples where GISs and (graphical) statistical packages have been linked. Williams, Limp & Briuer (1990) demonstrate how S and the GRASS GIS can be jointly used for archaeological site classification and analysis. Scott (1994) links STATA with ArcView and the spatial data analysis software SpaceStat has been linked with ARC/INFO™ (Anselin, Dodson & Hudak, 1993) and with ArcView (Anselin & Bao, 1996; Anselin & Bao, 1997). In Haining, Ma & Wise (1996), the designing of a software system for interactive exploration of spatial data by linking to ARC/INFO has been discussed, and in Zhang & Griffith (1997), a spatial statistical analysis module implemented in ArcView using Avenue has been discussed. Mathsoft (1996) describes the S+GISLink, a bidirectional link between ARC/INFO and S – PLUS®, and Bao (1997) describes the S+Grassland link

ARC/INFO is a trademark of Environmental Systems Research Institute, Inc.
S–PLUS is a registered trademark of StatSci, a division of MathSoft, Inc.

between S–PLUS and the Grassland GIS. Finally, a comparison of the operational issues of the SpaceStat/ArcView link and the S+Grassland link has been given in Bao & Anselin (1997).

3 InterProcess Communication Standards and Remote Procedure Calls

The main goal of our research was to establish a bidirectional link between ArcView (first versions 2.0 and 2.1, now version 3.0) and XGobi. This link utilizes Remote Procedure Calls (RPCs), an InterProcess Communication (IPC) feature available in ArcView. The use of RPCs is a programming technique where a process on the local system (*client*) invokes a procedure on a remote system (*server*). In this context, the term *request* is used to refer to the client's desire to execute a remote procedure and the term *response* is used for the result produced by the remote procedure (Stevens, 1990). RPCs are differently implemented on different hardware platforms (see Stevens, 1990, Chapter 18, for more details). It should be possible to use the ArcView/XGobi link on any hardware platform that supports Sun Microsystems' RPC mechanism in addition to DEC alphastations and Sun/Sparc, SGI, and Data General workstations that all conform to this RPC standard. Detailed information on this RPC mechanism can be found in Corbin (1991). Stevens (1990) describes other IPC standards such as pipes, FIFOs (named pipes), message queues, semaphores, shared memory, Berkeley sockets, and the System V Transport Layer Interface (TLI) IPC on UNIX® systems.

RPCs are a mature and robust IPC technology easily applied between ArcView and XGobi. Newer distributed software tools like the Parallel Virtual Machine (PVM, e. g., Bode, Dongarra, Ludwig & Sunderam, 1996; Alexandrov & Dongarra, 1998), the Distributed Component Object Model (DCOM, e. g., Eddon & Eddon, 1998), and the Common Object Request Broker Architecture (CORBA, e. g., Otte, Patrick & Roy, 1996) have not been explored for use in the ArcView/XGobi link. The reason is easy. In 1994, when we started work on this link, ArcView 2.0 supported only RPCs as IPC mechanism. Even now, at publication date, it would be extremely difficult to implement any additional IPC technology in ArcView without having access to the ArcView source code.

For linking geographic packages with statistical packages there are two existing approaches: *close coupling* and *loose coupling* (Goodchild, Haining & Wise, 1992). Close coupling relates to the fact that one package is calling other packages directly or that this package is calling user–written routines using one of the previously listed IPC facilities. Loose coupling is simpler: multiple processes simultaneously access (read and write) the same file(s) or they simply pass data files (ASCII or binary) among each other. The link between ArcView and SpaceStat (Anselin & Bao, 1996; Anselin & Bao, 1997) and the software system designed

UNIX is a registered trademark of UNIX Systems Laboratories.

by Haining et al. (1996) are examples for loose coupling. The S+Grassland link (Bao, 1997) and the ArcView/XGobi link are examples of close coupling.

Overall, the approach of linking existing statistical and geographic packages through close coupling has several advantages when compared to linking through loose coupling. Close coupling in general provides access to the full capabilities of both packages without re–developing sophisticated existing software. It represents the current trend towards *open systems* (Günther, 1998, p. 143). Moreover, linked brushing, and truly dynamic graphics capabilities, is practically feasible only through close coupling, and it considerably enhances the power of the software environment beyond the sum of the capabilities of the individual packages.

IPC facilities are likely to be incorporated in more statistical software packages over time to make them accessible in open systems. As an example, the latest revision of S, "Version 4", provides a so–called *connection class object* which has specific classes such as files, pipes, and FIFOs. The connection class can be extended towards other IPC connections as well (Chambers, 1997; Chambers, 1998). Another similar development that will likely gain more importance in the near future is the linking of statistical and non–statistical software that is accessible through the World Wide Web. The Method ManageMent system MMM (Günther, Müller, Schmidt, Bhargava & Krishnan, 1997) is such a web–based system that currently links three software packages, i. e., Mathematica, Matlab, and XploRe. "The Omega Project for Statistical Computing" (`http://www.omegahat.org/`) has the goal to provide open–source software for statistical applications and support for distributed statistical computing through the use of the CORBA (e. g., Otte, Patrick & Roy, 1996) standard.

4 The Capabilities of the ArcView/XGobi Link

After experimenting with linking a GIS and a statistical package by implementing a unidirectional link between ARC/INFO and XGobi in 1993 and early 1994 (Cook, Cressie, Majure & Symanzik, 1994; Symanzik, Majure, Cook & Cressie, 1994), we restarted the work using ArcView 2.0. ArcView 2.0 supported calls to external procedures via RPCs and enabled external programs to invoke ArcView 2.0 functions using RPCs in contrast to ARC/INFO. These features enabled us to implement the kernel for a bidirectional link between ArcView 2.0 and XGobi that is still central to the current version of the link.

Initially, we concentrated on one–to–one connections between ArcView and XGobi such as linking geographic location to (multivariate) attribute values and to empirical spatial cumulative distribution function (SCDF) values. Subsequently, we explored one–to–two and two–to–one linking to connect variogram–cloud plots (Chauvet, 1982; Cressie, 1984; Haslett, Bradley, Craig, Unwin & Wills, 1991; Haslett & Bradley, 1991; Bradley & Haslett, 1992), spatially lagged scatterplots (Cressie, 1984; Rossi, Mulla, Journel & Franz, 1992), and multivariate variogram–cloud plots (Majure & Cressie, 1998) to geography.

We call these five diagnostic plots *links* as well when used in the ArcView/XGobi context. When speaking of the *basic (or multivariate) link*, we refer to the feature of the ArcView/XGobi link that links geographic location to (multivariate) attribute values. Alternatively, we refer to the other features as *SCDF link*, *variogram–cloud link*, *spatially lagged scatterplot link*, and *multivariate variogram–cloud link*. This dual use of the term *link* is intentional. Although these five features are a built–in part of XGobi, they cannot be used as a stand–alone part within XGobi, i. e., they cannot be invoked through a menu option in XGobi. When using the term *link* in this paper, we refer to all five features of the ArcView/XGobi link and their underlying implementation using RPCs.

The linking between ArcView and XGobi allows us to simultaneously display spatial locations and concomitant geographic variables within the GIS while visualizing and exploring the corresponding data space within XGobi. The usefulness of the link has been highlighted for several different applications such as satellite imagery, forest health monitoring, and precipitation data (Cook, Majure, Symanzik & Cressie, 1996; Majure, Cook, Cressie, Kaiser, Lahiri & Symanzik, 1996a; Majure, Cressie, Cook & Symanzik, 1996c; Symanzik, Majure & Cook, 1996; Symanzik, Majure & Cook, 1997b; Symanzik, Megretskaia, Majure & Cook, 1997d).

Cook, Symanzik, Majure & Cressie (1997) presents examples (measures of livability in the United States, forest health data, and precipitation measurements) for all five features of the link. There also exist videos that demonstrate the use of the link (Majure, Cook, Cressie, Kaiser, Lahiri & Symanzik, 1995; Symanzik, Majure & Cook, 1995; Majure, Cook, Symanzik & Megretskaia, 1996b).

In the remainder of this section, we explain the general idea of linked brushing between ArcView and XGobi and provide two examples for the multivariate link and the variogram–cloud link.

4.1 Linked Brushing between ArcView and XGobi

In the multivariate and SCDF links, there is exactly one point in XGobi corresponding to exactly one spatial location in ArcView and vice versa. For each point/location brushed, the corresponding location/point is brushed as well. In contrast, for the variogram–cloud, spatially lagged scatterplot, and multivariate variogram–cloud links, every point in XGobi corresponds to exactly two locations in ArcView. Actually, there might be up to n^2 points in XGobi if there are n locations in ArcView. We assume we brush only one point/location at a time in either application. Then, we make use of the following strategy for linked brushing:

Brushing in XGobi: First we mark the point in XGobi with the selected color, glyph, and size. Then we determine in XGobi which are the two related spatial locations (using *Array of Pairs* described in Section 5.2.2) and pass this information to ArcView. Finally, we connect these locations in ArcView with a line of the selected color (provided by XGobi). If we brush a point in XGobi that relates to locations with identical spatial origin ($\boldsymbol{s}_i = \boldsymbol{s}_j$) in ArcView, we do not plot anything on the ArcView side.

Brushing in ArcView: This geographic brushing step can be described as a three phase process, where ArcView, XGobi, and again ArcView proceed. First, we brush the location in the ArcView map view (nothing visible happens) and pass the information to XGobi, which location has been brushed and which color, glyph, and size have been selected. Then, in XGobi, we determine all points that are related to this location (using *Brushing List 1* and *2* described in Section 5.2.2). There might be up to $2n - 1$ such points in XGobi. For each of these points, we assume it actually has been brushed in XGobi. We follow the instructions on brushing in XGobi, using the color, glyph, and size provided by ArcView. Finally, this will result in lines drawn in ArcView between the selected location and all locations that are not farther away than the cutoff distance d_{cut}.

It is important to note that in our application we redraw the selected theme in ArcView as graphics with the list of coordinates stored in a stack accessed by a hash function (a data dictionary in Avenue). After the data is initially passed from ArcView into XGobi, all brushing in ArcView occurs by manipulating graphics on the screen directly without accessing ArcView's theme table any more.

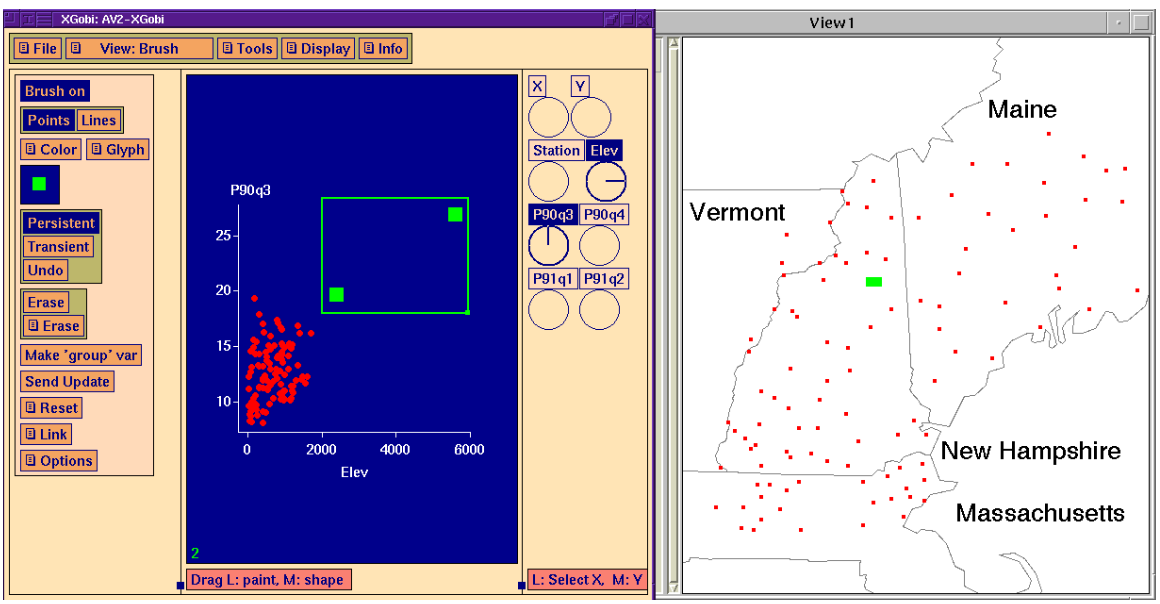

Figure 1: Example for the basic (or multivariate) link. The attribute data is displayed in XGobi (left) with a full complement of multivariate graphical tools available. This data is linked by brushing to the map view provided by ArcView (right). In the XGobi plot, two extremes on third quarter precipitation and elevation are painted as solid rectangles and they are shown to be located very close to one another in New Hampshire near the border with Maine in the ArcView map view.

4.2 Two Motivational Examples

We describe two examples which cover two of the five main features of the ArcView/XGobi link: the basic (or multivariate) link and the variogram–cloud link. The data in each case shows precipitation recordings in the northeastern United States, i. e., in Maine, New Hampshire, Vermont, and Massachusetts. Figure 1 shows an example of the basic (or multivariate) link, where all the attribute information is passed from ArcView into XGobi. The example shows the detection of outliers in the multivariate data space.

A trickier problem is the detection of spatial outliers, i. e., sites that have dramatically different attribute values from their close neighbors, but are not noticeably different in the overall range of the variables. A common approach to detecting spatial outliers is to use a variogram–cloud plot due to Cressie (1984). Here, $g_{11}(\boldsymbol{s_i}, \boldsymbol{s_j}) = \mid Z_1(\boldsymbol{s_i}) - Z_1(\boldsymbol{s_j}) \mid^{1/2}$ is plotted versus the Euclidean distance between the locations, $d(\boldsymbol{s_i}, \boldsymbol{s_j}) = \parallel \boldsymbol{s_i} - \boldsymbol{s_j} \parallel$, where $Z_1(\boldsymbol{s_i})$ represents the amount of precipitation measured at location $\boldsymbol{s_i}$ during the third quarter of 1990.

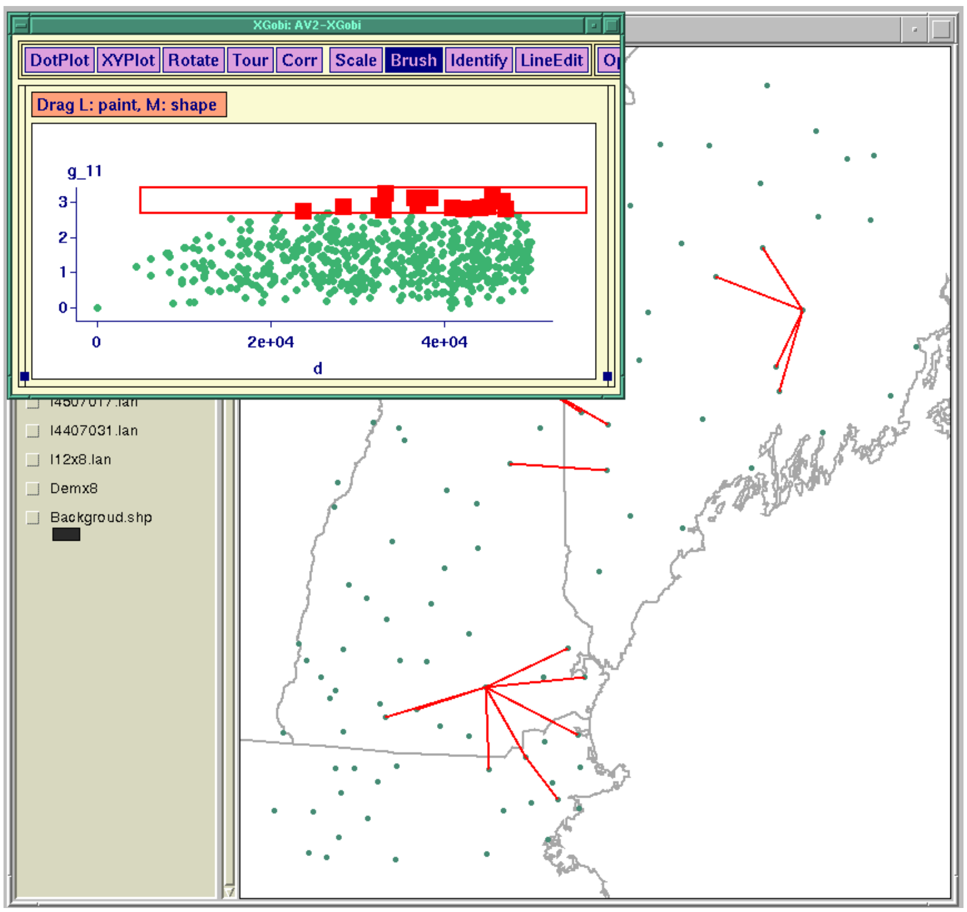

Figure 2: Example for the variogram–cloud link. In the upper left XGobi, we have brushed (using a solid rectangle) the highest values in the variogram–cloud plot ($g_{11}(\boldsymbol{s_i}, \boldsymbol{s_j})$) versus $d(\boldsymbol{s_i}, \boldsymbol{s_j})$). In the lower right ArcView map view, each pair of locations, related to a point that has been brushed in XGobi, has been connected by a line.

This approach is demonstrated in Figure 2. Each point in the variogram–cloud corresponds to two sampling sites in the map. High values of g_{11} (the precipitation difference between the sampling sites is large) are brushed in the upper left (XGobi) variogram–cloud plot and the corresponding pairs of sampling sites are identified by connecting lines in the lower right (ArcView) map view. Locations that have several lines connecting them to other sites are spatial anomalies, i. e., sites that are quite different from their neighbors. This display shows two such sites — one in Maine and one in New Hampshire. None of them was detected in the multivariate link because neither is an outlier in the global sense.

Note that a variogram–cloud plot is a way to explore spatial dependence and is usually the first step of estimating a variogram (e. g., Cressie, 1993). There are also tools for smoothing available in XGobi so the user can interactively smooth the variogram–cloud, giving the analyst a preliminary non–parametric estimate of the variogram.

5 Technical Implementation Details

This section describes the use of RPCs in the ArcView/XGobi link, including discussion on the additions to XGobi, modifications to ArcView, and security and concurrency issues.

5.1 RPCs in the ArcView/XGobi Link

From within ArcView, external remote procedures can be called and external programs can invoke internal ArcView functions via RPCs. To utilize these features, we initially wrote intermediate C code containing the RPCs and called XGobi as a subroutine, following the examples provided by the XGvis software system (Littman, Swayne, Dean & Buja, 1992) and the interface between the image program MTID and XGobi (Klein & Moreira, 1994). With the port to ArcView 3.0, the RPC mechanism became a permanent part of the XGobi source code.

ArcView has been modified for this application using its built–in Avenue programming language. All of the default ArcView functionality is available, with the addition of several operations that are necessary to handle the link. Specifically, ArcView has been modified to do the following: initiate an RPC server and client, initiate and pass data to the XGobi process, brush locations in the map view and request XGobi to brush the corresponding points, and process requests from XGobi to brush locations. Pseudo–code for ArcView and XGobi that describes this functionality in detail can be found in Symanzik et al. (1996, 1997b).

Both, ArcView and XGobi, have been set up as a server as well as a client. ArcView (as a client) can request action from XGobi (as the server). A procedure in the main event loop of XGobi has to be used to check for incoming RPC requests from clients (namely ArcView). Note that it is not possible to give XGobi the full functionality of an RPC server, which would require that it automatically processes all requests from its clients. Such an automation would never terminate and thus, no other user action within XGobi could be processed. Conversely, XGobi (as a client) sends requests to ArcView (as the server), asking ArcView to update its map view in accordance with the brushing and subsetting information of points done in XGobi. In the next section, we will describe how XGobi remote procedures can be called from within ArcView.

5.2 XGobi Additions

5.2.1 XGobi's RPC Structure

It was necessary to add 34 remote procedures to XGobi, that are executable through the XGobi RPC server, to obtain the functionality for the five different links. These remote procedures can be grouped into 6 classes: general use (function numbers 01–03), multivariate link (11–16), SCDF link (21–29), variogram–cloud link (31–35), spatially lagged scatterplot link (41–46), and multivariate variogram–cloud link (51–55). A client needs to know the function numbers. It does not use the full internal names. The full list of remote procedures can be found as an electronic appendix to this paper at

`http://www.math.usu.edu/~symanzik/papers/2000_jcgs/functdesc.ps.`

This list should also allow other software developers to establish RPC links to XGobi from their own programs.

We will describe the working of the multivariate link. To activate this link, we first call `RPC_Init_Data`(11), through which we pass the name of the data set, the number of columns (ncols), and the number of rows (nrows) from ArcView into XGobi. Many XGobi data structures are initialized here, internal memory to store the data (which is sent in a later call) is allocated, and the flag `is_init` is set. We call `RPC_Send_Colnames`(01) to pass the strings containing the names of the variables into XGobi. No flags are modified. If this remote procedure is not called, default variable names will be assigned. Also optionally, names for each observation can be passed into XGobi through `RPC_Send_Rownames`(02). The required next steps are `RPC_Send_Init_Data`(12), `RPC_Send_Init_Symbols`(13), and `RPC_Make_XGobi`(14), which sends the data matrix, the color/glyph of each point and lastly pops up the XGobi window, respectively. Linked brushing in the multivariate link is now supported through `RPC_Update_All_Symbols`(15) and `RPC_Update_Some_Symbols`(16).

The remote procedure calls return information to ArcView for error checking. This data consists of 5 character symbols: two digits represent the function number, a dot serves as a separator, and another two digits describe the result (error). ArcView uses a lookup–table of errors contained in the file `AV2XGErrTable` to provide feedback when problems occur. This file contains three columns of information. The first column contains the "*function number.error digits*". The second column contains the severity of the error. A "0" means success and is used only in combination with the error digits "00". An "R" indicates that incorrect numerical values have been received and ArcView should repeat (this is what the "R" stands for) its last request using correct data. A "W" indicates a warning. An "F" indicates a fatal error which is impossible to recover from so the link should be terminated — if possible directly from within ArcView or through user intervention. The third column contains the name of the remote procedure and a verbal description of the error. Both, name and description, are displayed in an ArcView message window in case of an "R", "W", or "F" condition.

The full `AV2XGErrTable` can be found as an electronic appendix to this paper at

`http://www.math.usu.edu/~symanzik/papers/2000_jcgs/AV2XGErrTable.`

To explain how it is used, we use a small subset of this file, i. e., some of the lines used by the multivariate link:

```
01.00 0 RPC_Send_Colnames: OK.
01.01 F RPC_Send_Colnames: Call RPC_Init_Data or RPC_Init_CDF_Data first.
01.02 F RPC_Send_Colnames: Submit number of columns many column names
  (separated by space).

11.00 0 RPC_Init_Data: OK.
11.01 F RPC_Init_Data: Submit a name, number of columns, and number of rows
  (separated by space).

12.00 0 RPC_Send_Init_Data: OK.
12.01 F RPC_Send_Init_Data: Call RPC_Init_Data first.
12.02 F RPC_Send_Init_Data: Submit (number of rows x number of columns)
  many values (separated by space).

16.00 0 RPC_Update_Some_Symbols: OK.
16.01 F RPC_Update_Some_Symbols: Call RPC_Send_Init_Symbols and
  RPC_Make_XGobi first.
16.02 F RPC_Update_Some_Symbols: Submit a list of number:color/glyph/size
  (separated by space).
16.03 R RPC_Update_Some_Symbols: Wrong value for number.
16.04 R RPC_Update_Some_Symbols: Wrong value for color/glyph/size.
```

For remote procedure 01 (`RPC_Send_Colnames`) the following problems can occur: (1) In case of error digits "00", everything worked OK. (2) For error digits "01", this remote procedure has been called before either remote procedure 11 or 21 has been called. (3) In case of error digits "02", too few column names representing the variables have been sent. Both cases are assumed to be fatal errors where the link should be terminated.

In the last few lines of the lookup–table reproduced above that relate to remote procedure 16 (`RPC_Update_Some_Symbols`), we see the use of the "R" condition. This tells ArcView to repeat its last request using correct data since an incorrect point identifier (error digits "03") or an incorrect value for the encoded color, glyph, and size (error digits "04") has been received. Even though this particular brush request failed, the link should still be operational and recovery from this error seems possible, most likely just by resending the data.

5.2.2 Modifications to XGobi's Data Structures

This section concentrates on XGobi's data structures specifically created to facilitate the variogram–cloud link, the spatially lagged scatterplot link, and the multivariate variogram–cloud link. There were no modifications to XGobi's internal data structures for the multivariate link and SCDF link. RPCs also provide access to XGobi's existing data structures: the data matrix and the color and brush vectors.

We will explain the new data structures based on the variogram–cloud link. This link displays the variogram–cloud plot and cross variogram–cloud plot in XGobi. Originally, we have an array of *Initial Data* passed from ArcView to XGobi. See Figure 3 for illustration purposes. We will take the number of spatial locations, n, to be 3, $\boldsymbol{s_i} = (x_i, y_i), \quad i = 1, \ldots, n$, are these locations in the 2–dimensional plane, and Z_1 and Z_2 (in the general case $Z_1, \ldots, Z_k$) are the measures taken at location $\boldsymbol{s_i}$. Note that Z_1 and Z_2 may be the same variable taken at different times, or they may be different variables. The cutoff distance, d_{cut} (1.1 in this example), is also passed from ArcView to XGobi. The link will provide the user with an estimate of the expected number of points in XGobi for a given d_{cut}. The larger the number of points displayed in XGobi, the slower the link is. However, performance considerably differs among hardware platforms. While old Sun/Sparc workstations needed more than 20 minutes to fully initialize the link in a complex scenario, the same scenario has been initialized in less than a minute on a SGI/Onyx workstation. After some experimentation with the software, users should know how many points can be handled reasonably well using their computer hardware.

The *Array of Pairs* is the main data structure used to support this part of the link. It consists of records corresponding to each pair $(\boldsymbol{s_i}, \boldsymbol{s_j})$ of locations satisfying the condition $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq d_{cut}$. Each record consists of the location indices (i and j), the angle between $\boldsymbol{s_j} - \boldsymbol{s_i}$ and the horizontal axis measured in degrees (A), the set $\gamma_{lm}, l, m = 1, \ldots, k$, where $\gamma_{lm}(\boldsymbol{s_i}, \boldsymbol{s_j}) = \mid (Z_l(\boldsymbol{s_i}) - Z_m(\boldsymbol{s_j})) \mid^{1/2}$, and $D(\boldsymbol{s_i}, \boldsymbol{s_j}) = \sqrt{\sum_{p=1}^{k} (Z_p(\boldsymbol{s_i}) - Z_p(\boldsymbol{s_j}))^2}$.

With these definitions in mind, a distinction among the terms variogram–cloud, cross variogram–cloud, and multivariate variogram–cloud can be expressed as follows: *Variogram–clouds* use γ_{ii}, *cross variogram–clouds* use γ_{ij}, $i \neq j$, and *multivariate variogram–clouds* use γ_{ij} back to back with γ_{ji}, $i \neq j$. The latter name intrinsically contains the word "cross" since γ_{ii} back to back with γ_{ii} would be nothing else but a mirror image of the positive x–axis mapped onto the negative x–axis.

In many situations, we will consider a plot of $\gamma_{lm}(\boldsymbol{s_i}, \boldsymbol{s_j})$ versus $d(\boldsymbol{s_i}, \boldsymbol{s_j}) = \mid\mid \boldsymbol{s_i} - \boldsymbol{s_j} \mid\mid = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ (given in Figure 2) to assess the spatial dependence before fitting a (cross–) variogram model. The corresponding variable names that appear in XGobi are called g_{lm} and d.

The data record in XGobi's *Array of Pairs* also contains A (angular distance), $\sin(A)$ and $\cos(A)$. In the geographic context, there often exists a directional

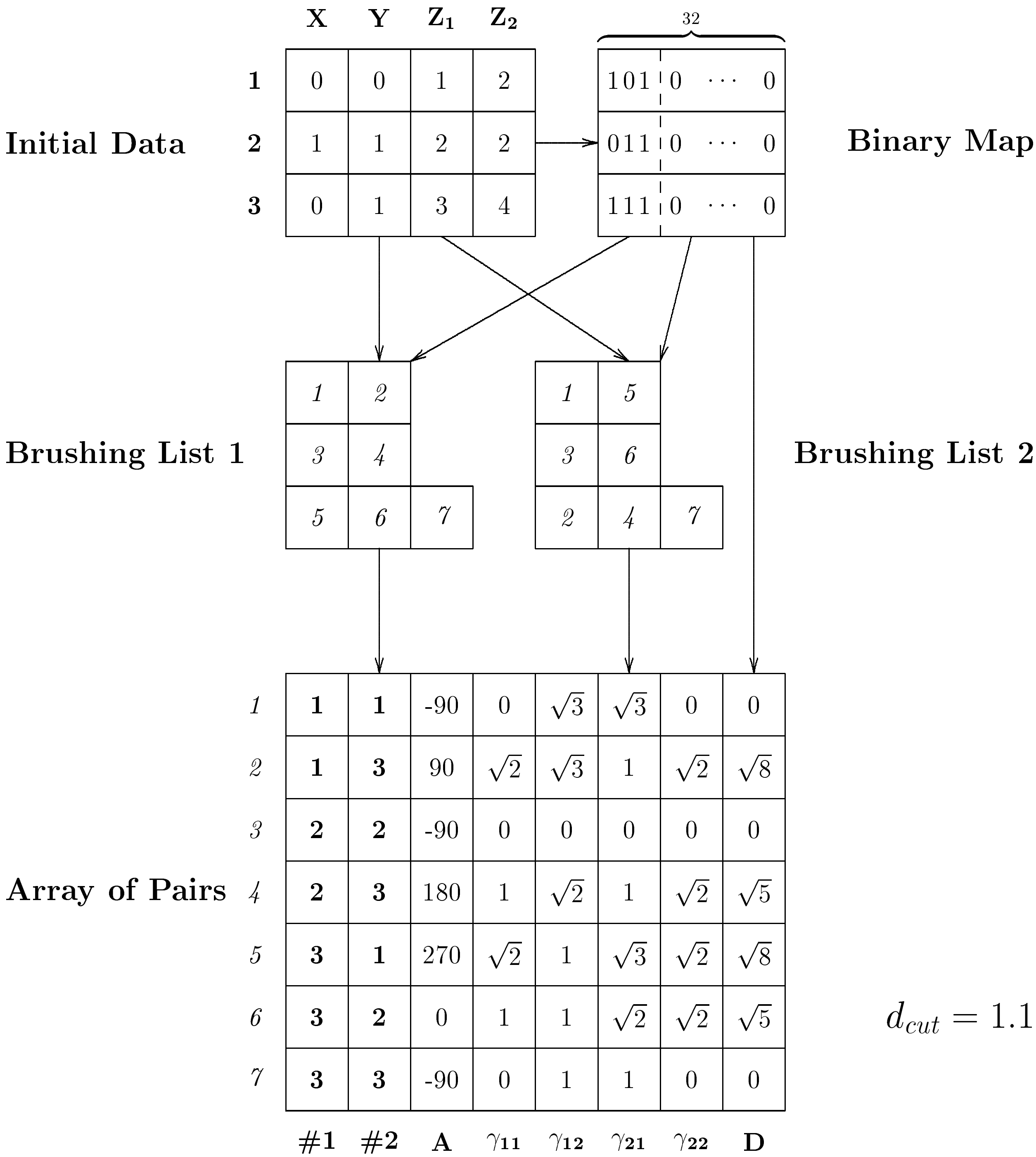

Figure 3: Data Structures used for the Variogram–Cloud Link are *Initial Data*, *Binary Map*, *Brushing List 1* and *2*, and *Array of Pairs*. In this example, we have $n = 3$ spatial locations and select a cutoff distance $d_{cut} = 1.1$. Numbers printed in bold (**1**, **2**, and **3**) relate to the initial data at spatial locations $\boldsymbol{s_1}$, $\boldsymbol{s_2}$, and $\boldsymbol{s_3}$ in ArcView. Numbers printed in italic (*1* through *7*) relate to points created for use in XGobi.

effect in the spatial dependence. A location $\boldsymbol{s_1}$ east of location $\boldsymbol{s_0}$ may have a considerable correlation with the values observed at location $\boldsymbol{s_0}$, but a location $\boldsymbol{s_2}$ at the same distance north of location $\boldsymbol{s_0}$ may not be correlated at all. Note that we handle indeterminate cases specially: When locations are identical ($i = j$), we assign $A = -90$. For observations from the same location that are accessed through different indices ($\boldsymbol{s_i} = \boldsymbol{s_j}, i \neq j$), we assign $A = -45$. So–called "directional variograms" are a feature of many software tools for spatial data analysis, such as Variowin (Pannatier, 1996). In the ArcView/XGobi link, we can interactively overlay directional variogram–cloud plots onto the overall variogram–cloud plot by brushing either in a dotplot of the angle A or in a scatterplot of $\sin(A)$ versus $\cos(A)$. For example, if we want to highlight the east–west direction, we might brush angles within the intervals (350°, 10°) and (170°, 190°). For the north–south direction, we might brush angles within the intervals (80°, 100°) and (260°, 280°). Because angle is a modular measurement, it is particularly convenient to display the scatterplot of sin(A) vs cos(A) and brush "around the circle". It is also more intuitive because this type of display can be read like a compass. Examples that feature the brushing of two angle classes can be found in Symanzik et al. (1997c, 1997d) and Cook et al. (1997).

A *Binary Map* is created upon initialization of the link. It is a 2–dimensional array of size $n \cdot \text{ceil}(n/32)$, which is large enough to provide information on all n^2 pairs of locations but small in memory size to faciliate speed and save storage. A "1" at the jth bit in the ith row indicates that the distance between $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ is smaller than d_{cut}. For example, in the first row, third column, the "1" indicates that the distance between $\boldsymbol{s_1}$ and $\boldsymbol{s_3}$ is smaller than $d_{cut} = 1.1$. Using the *Initial Data* array, we could see that the distance between $\boldsymbol{s_1}$ and $\boldsymbol{s_3}$ is 1. A "0" indicates that the distance between two locations is bigger than d_{cut}. For example, in the first row, second column, the "0" indicates that the distance between $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ is bigger than $d_{cut} = 1.1$. Actually, it is $\sqrt{2}$ based on the values in the *Initial Data* array. Information on about 32 pairs of locations is packed into one unsigned integer. Simultaneously with the creation of *Binary Map*, we count the number of locations that lie within the cutoff distance for every initial location. After we finish looking at *Initial Data*, we know exactly how many pairs that contain location $\boldsymbol{s_i}$ as a component are going to be in *Array of Pairs* for every $i = 1, \ldots, n$. This allows us to allocate memory for *Array of Pairs* and *Brushing List 1* and *2*. *Brushing List 1* and *2* contain information on linking from *Array of Pairs* to *Initial Data*. Each row of *Brushing List 1* and *2* relates to the corresponding row (and, thus, to a spatial location) in *Initial Data*, i. e., row 1 relates to $\boldsymbol{s_1}$, row 2 relates to $\boldsymbol{s_2}$, and row 3 relates to $\boldsymbol{s_3}$. *Brushing List 1* corresponds to all pairs of locations for which $\boldsymbol{s_i}$ is the first component in the pair. For example, its first row says that location $\boldsymbol{s_1}$ is the first component in the pairs (i. e., rows) *1* and *2* in *Array of Pairs*. *Brushing List 2* corresponds to all pairs of locations for which $\boldsymbol{s_i}$ is the second component in the pair. For example, its first row says that location $\boldsymbol{s_1}$ is the second component in the pairs (i. e., rows) *1* and *5* in *Array of Pairs*.

Using *Binary Map* we pick qualifying pairs of locations, add information to *Brushing List 1* and *2*, and fill the record for this pair in *Array of Pairs*. The initialization of *Brushing List 1* and *2* and *Array of Pairs* works as follows: For *Brushing List 1* and *2*, we know at each step exactly how many "cells" have been filled so far (there exists a special array of counters). We place the index number of the record used in *Array of Pairs* in the first free positions of *Brushing List 1* and *2* with respect to the two components of this pair and increment the corresponding counters by one. After we have gone through *Binary Map*, we have *Array of Pairs* and *Brushing List 1* and *2* completely filled.

Using these data structures, we can brush points/locations in two directions. If we brush point i in XGobi, we have to look at row i of *Array of Pairs* and know immediately which two spatial locations this entry consists of. Based on this information, we can draw a line that connects these two locations in the ArcView map view. Alternatively, brushing location $\boldsymbol{s_i}$ in ArcView corresponds to selecting row i of *Initial Data*. Also, the ith row of *Brushing List 1 (2)* contains all index numbers of the pairs that contain location $\boldsymbol{s_i}$ as first (second) component. Since these index numbers relate to rows in *Array of Pairs*, we know which points have to be brushed in XGobi. In continuation of our example, if we brush location $\boldsymbol{s_1}$ in ArcView, this translates to row 1 of *Initial Data*, and, finally, points related to pairs *1*, *2*, and *5* in *Array of Pairs* will be brushed in XGobi. See Section 4.1 for an explanation of the general idea of linked brushing between ArcView and XGobi.

The efficiency of our data structures is worth discussing. As an example, we assume there are $n = 128$ locations in ArcView initially. This would require an array of size $128^2 = 16384$ in XGobi if all possible pairs of locations would be used. We also assume that a cutoff distance d_{cut} has been selected such that only 1024, i. e., 1/16th of the possible number of points, are displayed in XGobi. Using the *Binary Map* approach, we reduce the total data array size to 1024 entries and need only $128^2/32 = 512$ unsigned integers as additional memory for *Binary Map* and 2×1024 entries for *Brushing List 1* and *2*. This can be neglected compared to a potential allocation of 16384 records of *Array of Pairs*. Each row of *Brushing List 1* and *2* is reasonably small ($1024/128 = 8$ in average) to allow efficient handling and brushing of points.

5.3 ArcView 3.0 Modifications

The implementation of the link in ArcView was accomplished by a combination of Avenue scripts, additions to the graphical user interface (GUI), and, in the latest version, the incorporation of all scripts into an extension that can be loaded by the user at runtime. In the context of the ArcView/XGobi link, ArcView has to do two things: (1) respond to user actions and request appropriate changes in XGobi, and (2) respond to RPC requests from XGobi and make the requested changes in the map view.

Figure 4 shows the *View* Document GUI used for the basic (or multivariate) link, the variogram–cloud link, the spatially lagged scatterplot link, and the multivariate variogram–cloud link. A slightly different GUI is used for the SCDF link.

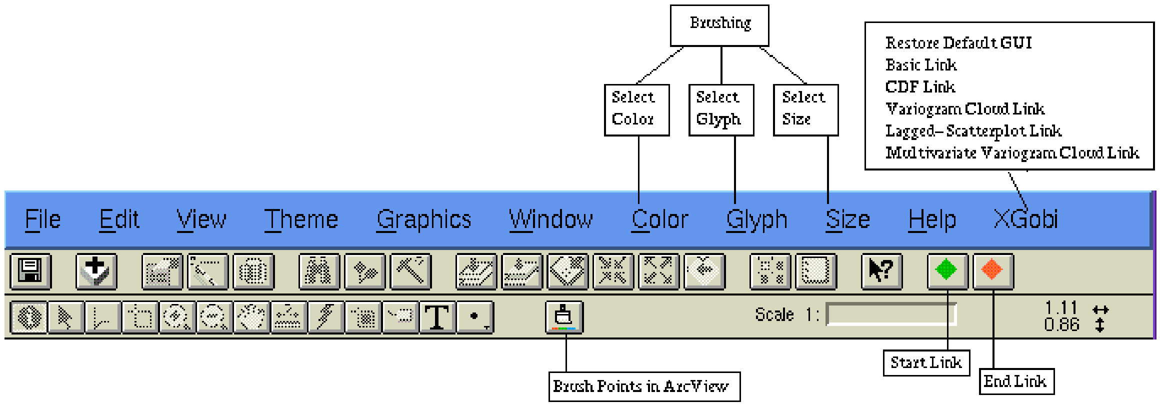

Figure 4: The *View* Document GUI in ArcView 3.0, used for all features of the link except the SCDF link. The following new features of the GUI have been added for the ArcView/XGobi link but are not part of the standard ArcView GUI. The red and green diamond buttons allow the user to start and stop XGobi. The paint brush button switches user control to brushing mode. The "XGobi" pulldown menu allows the user to select one of the features of the link. The "Color", "Glyph", and "Size" pulldown menus allow the user to select the plotting symbol.

Features on the GUI added to facilitate the ArcView/XGobi link are:

- the "XGobi" pulldown menu: select one of the features of the link;
- the green and red diamond buttons: start and stop XGobi;
- "Color", "Glyph", and "Size" pulldown menus: set the current brushing characteristics, i. e., color, glyph, and size of the symbol used for plotting the locations; and
- the paint brush button: brush locations in the ArcView map view.

The following Avenue scripts, listed in alphabetical order, are required in ArcView to support the basic (or multivariate) link:

CreateErrorDictionary: Builds the table of error messages that is internally used in ArcView based on the file `AV2XGErrTable` (described in Section 5.2.1).

RGBColorNameDictionary: Creates a dictionary containing ASCII color names and RGB color values that are related to the colors used in XGobi.

SetCurrentColor: Selects and sets a new color in the "Color" pulldown menu.

SetCurrentGlyph: Selects and sets a new glyph in the "Glyph" pulldown menu.

SetCurrentSize: Selects and sets a new size in the "Size" pulldown menu.

XG.Finish: Sends a "finish" request to XGobi, terminates the RPC Server within ArcView, and resets the buttons in the ArcView GUI.

XG.ReadXGobiColors: Attempts to read the `$HOME/app-defaults/XGobi` file (if available) and tries to get the current XGobi colors from it.

XG.RPC.Brush.Some: Called from within XGobi when points are brushed in the XGobi window and an update of the ArcView map view is required. The data that is passed is of the type "list of index:size_color_glyph" where "index" is the internal index number of the point and "size_color_glyph" is the color/glyph/size information encoded according to internal XGobi specifications and, therefore, first has to be decoded in ArcView.

XG.RPC.XGobi_Is_Up: Halts ArcView while ArcView is waiting for XGobi to start up. It does not do anything other than delaying execution for a while.

XG.Select: Allows the user to select locations in the current ArcView map view by drawing a boundary based on polygons. The locations inside the boundary are brushed using the current selections in the "Color", "Glyph", and "Size" pulldown menus. Also, remote procedure 16 (described in Section 5.2.1) in XGobi is called, with the color/glyph/size information of all selected locations, encoded according to internal XGobi specifications.

XG.Start: Initiates the ArcView to XGobi link. It activates the RPC mechanism in ArcView, starts XGobi as a background process, redraws the selected view as graphics, and initializes the new buttons and functionality in the ArcView GUI. In this script, all global Avenue variables that are later used in other scripts are defined. This script also reads environment variables and executes the scripts **RGBColorNameDictionary** and **CreateErrorDictionary**. Internally, this script calls remote procedures 11, 01, 02, 12, 13, and 14 (described in Section 5.2.1) in XGobi. Each link type has its own startup script.

Similar Avenue scripts are used to support the four other features of the link.

Since the introduction of the "Dialog Designer" extension by ESRI, we have designed custom dialogs for each of the links. In the variogram–cloud link, we have introduced a slider to predict the number of points in the variogram–cloud so that the user can interactively select the cutoff distance and control the amount of data passed to XGobi.

5.4 Security and Concurrency

Security concerns the ability of an unauthorized user to "break" in and access or even destroy the ArcView data base. The ArcView RPC server supports only one function (function number "1": "script execution – executes the given script and returns a string representation of the last object referenced or produced during the execution of the script" (Environmental Systems Research Institute, Inc., 1996)). It takes a string as the input where the string must contain an Avenue script. This script can be a single Avenue statement or a very long and complex segment of Avenue code. This could be a potential security problem in the case that a request by a non–authenticated user that contains a valid Avenue script of the form `system.execute("XXX")`, where `XXX` may be any Unix command, is granted.

RPCs and existing security issues have been described in the technical literature, e. g., Corbin (1991). Three levels of authentication exist. Null authentication means that any RPC request will be granted. Unix authentication causes the transmission of additional fields (such as a time stamp, the name of the local host, and the client's effective user and group IDs) with every RPC request. Data Encryption Standard (DES) authentication promotes secure exchange of data in a standard fashion since it encrypts/decrypts data through public and private keys associated with the effective user ID of the calling process. For Unix and DES authentication, only valid requests are granted.

ArcView 2.1 and 3.0 provide Unix authentication as the highest authentication mechanism. In our link between ArcView (2.1 and 3.0) and XGobi we make use of Unix authentication and process only those requests where local host and user ID match between ArcView and XGobi (which is started from within ArcView — thus IDs will match for valid requests). Unfortunately, there is no verifier for Unix authentication. Thus, the previously mentioned credentials can still be faked. Therefore, we suggest that ESRI provides an RPC mechanism for DES authentication in addition to the current Unix authentication for future releases of ArcView.

ArcView 2.0 only supported null authentication. There was no protection against the manipulation or deletion of entire ArcView 2.0 data bases or any of the analyst's files by an external user connected to the running ArcView 2.0 RPC server, hence a total lack of data security.

Concurrency means "What happens if at least two XGobi processes provide different update information to ArcView at the same time?" It has been solved by a built–in feature of XGobi. Only one XGobi process, the one invoked from within ArcView, provides the features of an RPC server and RPC client. Any other XGobi processes, e. g., cloned ones, are "regular" XGobi processes that do not initiate or respond to RPCs. XGobi processes (the one invoked from within ArcView and the cloned ones) communicate to each other through the production and consumption of *XEvents*. Examples of XEvents are the pressing or releasing of a mouse button, crossing a window boundary with the mouse, or events created by any of the XWindow client programs (see Nye, 1992, p. 38, for more details). When points are brushed or subsetted in XGobi, an appropriate XEvent is generated. XGobi sequentially processes the XEvents that have been generated by other XGobi processes. Information will get back to ArcView when the XGobi invoked from within ArcView processes an XEvent related to brushing or subsetting. Due to this one–to–one link and the serialization of the XEvents, it was not necessary to consider additional steps to solve the concurrency issue.

6 Discussion

One of the current limitations of the ArcView/XGobi link is the number of points that can be handled. No severe problems arise for the multivariate and SCDF links since, in both cases, the link reasonably supports as many points as can be managed in real time within XGobi itself. However, severe problems arise for the variogram–cloud link, spatially lagged scatterplot link, and multivariate variogram–cloud link if the selected cutoff distance results in hundreds of thousands of points in XGobi. While XGobi supports up to about 500,000 points (Swayne et al., 1998), the RPC mechanism does not. The communication and encoding/decoding of data into strings that have to be passed between ArcView and XGobi requires considerable computational time. With large amounts of data, it is possible that linked brushing between ArcView and XGobi works only partially or even fails completely when multiple XGobi processes (the one invoked from within ArcView and the cloned ones) are involved. For example, when the encoding/decoding of data into strings takes a larger amount of time for the XGobi process that has been invoked from within ArcView, this XGobi process might eventually lose some of the XEvents related to brushing or subsetting information created by cloned XGobi processes since the XEvent buffer may eventually be erased after a while. Problems can also occur when a timeout on the client side makes the client believe that the server does not work properly any more although the server may be just slow processing a particular (brushing) request. There is no immediate solution for these problems due to the restrictions imposed by the ArcView 3.0 RPC mechanism. It has been indicated by ESRI that future versions of ArcView will be able to process C code. This would allow a much faster RPC mechanism where compressed data is passed between ArcView and XGobi. Both programs might also be allowed to use shared memory.

Currently, the ArcView/XGobi link does not allow simultaneous invocations of XGobi with different features. Unfortunately, we cannot look at an SCDF plot and a variogram–cloud plot at the same time. To allow such an option in our software environment, we plan to implement a new type of linked brushing, called "hierarchical" linked brushing. This linked brushing is required for a future environment where ArcView communicates with several XGobis, each of them displaying a different feature of the link, e. g., one XGobi and a cloned child for the attribute data, one XGobi for the spatial cumulative distribution functions, and one XGobi for the variogram–cloud plot. Extensions to our ArcView Avenue code would be required.

The emphasis of the link is on exploratory spatial data analysis. ArcView is mostly used to store the data and display additional geographic features that relate to the statistical data of interest. Currently, the geographic brushing in ArcView exists only in what Monmonier (1989) defines as its simplest form: "use a mouse to highlight specific areas on the map". However, another extension on the ArcView side is planned to facilitate more complex types of geographic brushing, e. g., brushing statements of the form "brush all spatial locations that are at most 10 km away from the next city boundary and have no major road within a distance of 1 km". This could be easily done now with a query dialog.

An extension of the ArcView/XGobi link towards the statistical computing environment XploRe (Härdle, Klinke & Turlach, 1995) has been described in Symanzik, Kötter, Schmelzer, Klinke, Cook & Swayne (1998b), Symanzik, Klinke, Schmelzer, Cook & Lewin (1997a), and Symanzik, Cook, Klinke & Lewin (1998a). It should be noted that only few modifications were required on the ArcView/XGobi side to support this extension while XploRe had to undergo considerable changes since it did not support any RPC facilities prior to its addition to the ArcView/XGobi link. Recently, an experimental link among the Virtual Reality GIS ViRGIS (Pajarola, 1998), XGobi, and XploRe has been implemented (Symanzik, Pajarola & Widmayer, 1998c) that makes use of the same RPC technology initially developed for the ArcView/XGobi link. In the future, other packages with IPC facilities such as S (Becker, Chambers & Wilks, 1988; Chambers, 1998) with its connection class object (Chambers, 1997; Chambers, 1998) might be linked as well. We expect only little difficulties and modifications of the currently linked three programs when additional programs will be linked. But any additional software package most likely requires several changes to conform to the standard that has been set. However, we do not expect any major technical problem on either side when including additional software into the ArcView/XGobi/XploRe/ViRGIS environment.

Acknowledgements

Research related to this article was supported by an EPA grant under cooperative agreement # CR822919–01–0. The article has not been subjected to the review of the EPA and thus does not necessarily reflect the view of the agency and no official endorsement should be inferred. Symanzik's research was also partially supported by a German "DAAD–Doktorandenstipendium aus Mitteln des zweiten Hochschulsonderprogramms" and by a National Science Foundation Group Infrastructure Grant DMS–9631351. Nicholas Lewin–Koh was supported from a grant by the John Deere Foundation. We want to thank Noel Cressie for the helpful discussion of the theory related to the software. Thanks are also due to the anonymous reviewers for their comments that helped in providing an improved presentation of our work.

Appendix

The main web address for the software is:

`http://www.public.iastate.edu/~arcview-xgobi/homepage.html`

Additional web pages with information are:

http://www.math.usu.edu/~symanzik/papers/2000_jcgs/functdesc.ps
http://www.math.usu.edu/~symanzik/papers/2000_jcgs/AV2XGErrTable

References

Alexandrov, V. and Dongarra, J., editors (1998). *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Springer, Berlin. Lecture Notes in Computer Science; Vol. 1497.

Anselin, L. and Bao, S. (1996). Exploratory Spatial Data Analysis Linking SpaceStat and ArcView. Technical Report 9618, West Virginia University, Morgantown, WV.

Anselin, L. and Bao, S. (1997). Exploratory Spatial Data Analysis Linking SpaceStat and ArcView. In Fischer, M. M. and Getis, A., editors, *Recent Developments in Spatial Analysis*, pages 35–59. Springer, Berlin.

Anselin, L., Dodson, R. F., and Hudak, S. (1993). Linking GIS and Spatial Data Analysis in Practice. *Geographical Systems*, 1(1):3–23.

Asimov, D. (1985). The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143.

Bao, S. (1997). *User's Reference for the S+Grassland Link*. Mathsoft, Inc., Seattle, WA.

Bao, S. and Anselin, L. (1997). Linking Spatial Statistics with GIS: Operational Issues in the SpaceStat–ArcView Link and the S+Grassland Link. In *1997 Proceedings of the Section on Statistical Graphics*, pages 61–66, Alexandria, VA. American Statistical Association.

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

Bode, A., Dongarra, J., Ludwig, T., and Sunderam, V., editors (1996). *Parallel Virtual Machine — EuroPVM'96*. Springer, Berlin. Lecture Notes in Computer Science; Vol. 1156.

Bradley, R. and Haslett, J. (1992). Interactive Graphics for the Exploratory Analysis of Spatial Data — The Interactive Variogram Cloud. In Dowd, P. A. and Royer, J. J., editors, *2nd CODATA Conference on Geomathematics and Geostatistics, Sci. de la Terre, Ser. Inf.*, volume 31, pages 373–386, Nancy.

Brunsdon, C. and Charlton, M. (1996). Developing an Exploratory Spatial Analysis System in XLisp–Stat. In Parker, D., editor, *Innovations in GIS 3*, pages 135–145. Taylor & Francis, London, U.K.

Buja, A. and Asimov, D. (1986). Grand Tour Methods: An Outline. *Computing Science and Statistics*, 17:63–67.

Buja, A., Cook, D., and Swayne, D. F. (1996). Interactive High–Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99.

Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. (1987). Scatterplot Matrix Techniques for Large N. *Journal of the American Statistical Association*, 82(398):424–436.

Carr, D. B., Olsen, A. R., and White, D. (1992). Hexagon Mosaic Maps for Displays of Univariate and Bivariate Geographical Data. *Cartography and Geographic Information Systems*, 19(4):228–236, 271.

Chambers, J. M. (1997). Evolution of the S Language. *Computing Science and Statistics*, 28:331–337.

Chambers, J. M. (1998). *Programming with Data: A Guide to the S Language*. Springer, New York, NY.

Chauvet, P. (1982). The Variogram Cloud. In Johnson, T. B. and Barnes, R. J., editors, *17th International Symposium on the Application of Computers and Operations Research in the Mineral Industry*, pages 757–764, New York, NY. Society of Mining Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc. (Colorado School of Mines).

Cook, D., Cressie, N., Majure, J., and Symanzik, J. (1994). Some Dynamic Graphics for Spatial Data (with Multiple Attributes) in a GIS. In Dutter, R. and Grossmann, W., editors, *COMPSTAT 1994: Proceedings in Computational Statistics*, pages 105–119, Heidelberg. Physica–Verlag.

Cook, D., Majure, J. J., Symanzik, J., and Cressie, N. (1996). Dynamic Graphics in a GIS: Exploring and Analyzing Multivariate Spatial Data Using Linked Software. *Computational Statistics: Special Issue on Computeraided Analysis of Spatial Data*, 11(4):467–480.

Cook, D., Symanzik, J., Majure, J. J., and Cressie, N. (1997). Dynamic Graphics in a GIS: More Examples Using Linked Software. *Computers and Geosciences: Special Issue on Exploratory Cartographic Visualization*, 23(4):371–385. Paper, CD, and `http://www.elsevier.nl/locate/cgvis`.

Corbin, J. R. (1991). *The Art of Distributed Applications: Programming Techniques for Remote Procedure Calls*. Springer, New York, Berlin, Heidelberg.

Cressie, N. (1984). Towards Resistant Geostatistics. In Verly, G., David, M., Journel, A. G., and Marechal, A., editors, *Geostatistics for Natural Resources Characterization, Part 1*, pages 21–44, Dordrecht. Reidel.

Cressie, N. A. C. (1993). *Statistics for Spatial Data (Revised Edition)*. Wiley, New York, NY.

DiBiase, D., Reeves, C., MacEachren, A. M., von Wyss, M., Krygier, J. B., Sloan, J. L., and Detweiler, M. C. (1994). Multivariate Display of Geographic Data: Applications in Earth System Science. In MacEachren, A. M. and Taylor, D. R. F., editors, *Visualization in Modern Cartography*, pages 287–312. Pergamon (Elsevier), Oxford, U.K.

Dykes, J. (1996). Dynamic Maps for Spatial Science: A Unified Approach to Cartographic Visualization. In Parker, D., editor, *Innovations in GIS 3*, pages 177–187. Taylor & Francis, London, U.K.

Eddon, G. and Eddon, H. (1998). *Inside Distributed COM*. Microsoft Press, Redmond, WA.

Environmental Systems Research Institute, Inc. (1996). *ArcView 3.0 On–Line Help: RPCServer Page*. Redlands, CA.

Goodchild, M. F., Haining, R. P., and Wise, S. (1992). Integrating GIS and Spatial Data Analysis: Problems and Possibilities. *International Journal of Geographical Information Systems*, 6(5):407–423.

Günther, O. (1998). *Environmental Information Systems*. Springer, Berlin, Heidelberg.

Günther, O., Müller, R., Schmidt, P., Bhargava, H. K., and Krishnan, R. (1997). MMM: A Web–based System for Sharing Statistical Computing Modules,. *IEEE Internet Computing*, 1(3):59–68. Paper and `http://computer.org/internet/ic1997/w3toc.htm`.

Haining, R., Ma, J., and Wise, S. (1996). Design of a Software System for Interactive Spatial Statistical Analysis Linked to a GIS. *Computational Statistics: Special Issue on Computeraided Analysis of Spatial Data*, 11(4):449–466.

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

Haslett, J. and Bradley, R. (1991). Dynamic Graphics: Linked Points, Lines and Regions with Applications to Spatial Data Modelling. *Computing Science and Statistics*, 23:425–429.

Haslett, J., Bradley, R., Craig, P., Unwin, A., and Wills, G. (1991). Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies. *The American Statistician*, 45(3):234–242.

Klein, R. and Moreira, R. I. (1994). Exploratory Analysis of Agricultural Images via Dynamic Graphics. Technical Report 9/94, Laboratório Nacional de Computação Científica, Rio de Janeiro, Brazil.

Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992). Visualizing the Embedding of Objects in Euclidean Space. *Computing Science and Statistics*, 24:208–217.

MacDougall, E. B. (1992). Exploratory Analysis, Dynamic Statistical Visualization, and Geographic Information Systems. *Cartography and Geographic Information Systems*, 19(4):237–246.

Majure, J. J., Cook, D., Cressie, N., Kaiser, M., Lahiri, S., and Symanzik, J. (1995). Spatial CDF Estimation and Visualization with Applications to Forest Health Monitoring. ASA Statistical Graphics Video Lending Library (contact: dfs@research.att.com).

Majure, J. J., Cook, D., Cressie, N., Kaiser, M., Lahiri, S., and Symanzik, J. (1996a). Spatial CDF Estimation and Visualization with Applications to Forest Health Monitoring. *Computing Science and Statistics*, 27:93–101.

Majure, J. J., Cook, D., Symanzik, J., and Megretskaia, I. (1996b). An Interactive Environment for the Graphical Analysis of Spatial Data. ASA Statistical Graphics Video Lending Library (contact: dfs@research.att.com).

Majure, J. J. and Cressie, N. (1998). Dynamic Graphics for Exploring Spatial Dependence in Multivariate Spatial Data. *Geographical Systems*, 4(2):131–158.

Majure, J. J., Cressie, N., Cook, D., and Symanzik, J. (1996c). GIS, Spatial Statistical Graphics, and Forest Health. In *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, NM, January 21–26, 1996*, Santa Barbara, CA. National Center for Geographic Information and Analysis. CD and `http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/main.html`.

MathSoft (1996). *S+GISLink*. MathSoft, Inc., Seattle, WA.

McDonald, J. A. and Willis, S. (1987). Use of the Grand Tour in Remote Sensing. ASA Statistical Graphics Video Lending Library (contact: dfs@research.att.com).

Monmonier, M. (1988). Geographical Representations in Statistical Graphics: A Conceptual Framework. In *1988 Proceedings of the Section on Statistical Graphics*, pages 1–10, Alexandria, VA. American Statistical Association.

Monmonier, M. (1989). Geographic Brushing: Enhancing Exploratory Analysis of the Scatterplot Matrix. *Geographical Analysis*, 21(1):81–84.

Nye, A. (1992). *Xlib Programming Manual (Third Edition, for Version 11 of the X Window System)*. O'Reilly & Associates, Inc., Sebastopol, CA.

Openshaw, S. and Perrée, T. (1996). User–Centred Intelligent Spatial Analysis of Point Data. In Parker, D., editor, *Innovations in GIS 3*, pages 119–134. Taylor & Francis, London, U.K.

Otte, R., Patrick, P., and Roy, M. (1996). *Understanding CORBA: The Common Object Request Broker Architecture*. Prentice Hall, Upper Saddle River, NJ.

Pajarola, R. B. (1998). *Access to Large Scale Terrain and Image Databases in Geoinformation Systems*. PhD thesis, Diss. ETH No. 12729 Department of Computer Science, ETH Zürich.

Pannatier, Y. (1996). *VARIOWIN: Software for Spatial Data Analysis in 2D*. Springer, New York, Berlin, Heidelberg.

Rossi, R. E., Mulla, D. J., Journel, A. G., and Franz, E. H. (1992). Geostatistical Tools for Modeling and Interpreting Ecological Spatial Dependence. *Ecological Monographs*, 62:277–314.

Scott, L. M. (1994). Identification of a GIS Attribute Error Using Exploratory Data Analysis. *The Professional Geographer*, 46(3):378–386.

Stevens, W. R. (1990). *UNIX Network Programming.* Prentice–Hall, Englewood Cliffs, NJ.

Swayne, D. F., Cook, D., and Buja, A. (1998). XGobi: Interactive Dynamic Graphics in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):113–130.

Symanzik, J., Cook, D., Klinke, S., and Lewin, N. (1998a). Exploration of Satellite Images in the Dynamically Linked ArcView/XGobi/XploRe Environment. In Bodt, B. A., editor, *Proceedings of the Third Annual U.S. Army Conference on Applied Statistics, 22–24 October 1997*, pages 23–33, Aberdeen Proving Ground, MD. Army Research Laboratory ARL–SR–74.

Symanzik, J., Klinke, S., Schmelzer, S., Cook, D., and Lewin, N. (1997a). The ArcView/XGobi/XploRe Environment: Technical Details and Applications for Spatial Data Analysis. In *1997 Proceedings of the Section on Statistical Graphics*, pages 73–78, Alexandria, VA. American Statistical Association.

Symanzik, J., Kötter, T., Schmelzer, S., Klinke, S., Cook, D., and Swayne, D. (1998b). Spatial Data Analysis in the Dynamically Linked ArcView/XGobi/XploRe Environment. *Computing Science and Statistics*, 29(1):561–569.

Symanzik, J., Majure, J., Cook, D., and Cressie, N. (1994). Dynamic Graphics in a GIS: A Link between ARC/INFO and XGobi. *Computing Science and Statistics*, 26:431–435.

Symanzik, J., Majure, J. J., and Cook, D. (1995). Dynamic Graphics in a GIS: Analyzing and Exploring Multivariate Spatial Data. ASA Statistical Graphics Video Lending Library (contact: dfs@research.att.com).

Symanzik, J., Majure, J. J., and Cook, D. (1996). Dynamic Graphics in a GIS: A Bidirectional Link between ArcView 2.0 and XGobi. *Computing Science and Statistics*, 27:299–303.

Symanzik, J., Majure, J. J., and Cook, D. (1997b). Dynamic Graphics in a GIS: A Bidirectional Link between ArcView 2.1 and XGobi — An Update. *Computing Science and Statistics*, 29(2):35–40.

Symanzik, J., Majure, J. J., and Cook, D. (1997c). The Linked ArcView 2.1 and XGobi Environment — GIS, Dynamic Statistical Graphics, and Spatial Data. In Shekhar, S. and Bergougnoux, P., editors, *Proceedings of the Fourth ACM Workshop on Advances in Geographic Information Systems, Rockville, Maryland, November 15–16, 1996*, pages 147–154, New York, NY. ACM.

Symanzik, J., Megretskaia, I., Majure, J. J., and Cook, D. (1997d). Implementation Issues of Variogram Cloud Plots and Spatially Lagged Scatterplots in the Linked ArcView 2.1 and XGobi Environment. *Computing Science and Statistics*, 28:369–374.

Symanzik, J., Pajarola, R., and Widmayer, P. (1998c). XGobi and XploRe Meet ViRGIS. In *1998 Proceedings of the Section on Statistical Graphics*, pages 50–55, Alexandria, VA. American Statistical Association.

Unwin, A., Wills, G., and Haslett, J. (1990). REGARD — Graphical Analysis of Regional Data. In *1990 Proceedings of the Section on Statistical Graphics*, pages 36–41, Alexandria, VA. American Statistical Association.

Williams, I., Limp, W. F., and Briuer, F. L. (1990). Using Geographic Information Systems and Exploratory Data Analysis for Archaeological Site Classification and Analysis. In Allen, K. M. S., Green, S. W., and Zubrow, E. B. W., editors, *Interpreting Space: GIS and Archaeology*, pages 239–273. Taylor & Francis, London, U.K.

Zhang, Z. and Griffith, D. A. (1997). Developing User–Friendly Spatial Statistical Analysis Modules for GIS: An Example using ArcView. *Computers, Environment and Urban Systems*, 21(1):5–29.