Interaction Tools to Support Knowledge Discovery: A Case Study Using Data Explorer and Tcl/Tk

Ray Masters[1]
Center for Academic Computing and Department of Architecture
The Pennsylvania State University

Rob Edsall[2]
Department of Geography
The Pennsylvania State University

Abstract

In this paper we present an approach for creating a software environment and user interface that are appropriate for the process of constructing knowledge from large spatiotemporal data sets. Central to the theoretical framework of Knowledge Discovery in Databases (KDD) is the process of iterative user interaction and the ability of the human content expert to work closely with the computer to uncover and recognize interesting and meaningful patterns in data. Cognitive research indicates that the design of visual representations and interaction techniques can influence a user's conceptualization of the phenomenon represented. As such, a productive human-computer dialogue, key to effective knowledge discovery, often relies on high-bandwidth, discipline-specific, even user-specific, interactors. These interactors are not just jazzy radio buttons or sliders, but mechanisms that help one think and create, prompting mental models that can be exploited in the next iteration of the knowledge creation process. Because the patterns, processes, and relationships of the data are generally unknown in large data sets, interactors need to be more flexible than in the instance where the data and their relationships are well understood.

The KDD, human-in-the-loop approach relies on the enhancement of cognitive processes by linking creative problem solving with data representation and display. Our approach was to create multiple representations and interactors in order to facilitate flexible and creative thinking; through the multiple representations, the visual display presents the data in a variety of contexts and conceptual models. In this study we created a hybrid visualization environment utilizing Data Explorer as the core tool and Tcl/Tk for the development of customized interactors, each linked with the others to form a highly dynamic and interactive user interface. The content expert searches for clusters and multivariate similarities by interacting with multiple data contexts: 1. Geoviews, 2. 3D scatterplots, and 3. Parallel coordinate plots. Tcl/Tk offers the ability to prototype rapidly and implement highly interactive interfaces that are application-specific, discipline-specific, and that can be tailored to the content expert's individual preferences. The content expert interacts within these contexts and controls the: 1. assignment of data, 2. Brushing of data (determining where particular data occur in geographic and/or temporal space, and whether they have similar signatures in the parallel-coordinate plot), 3. focus of the data display (an interesting numerical range, perhaps), 4. colormap manipulation, 5. 3-D viewpoint manipulation, and 6. sequencing of a dynamic variable (usually temporal, but can easily be any attribute dimension in the data).

In this paper we illustrate the potential of our approach by applying this prototype to the problem of finding features in spatiotemporal climatic data sets. We touch on the relative ease of constructing complex power interactors with Tcl/Tk, discuss how others can utilize this approach to enhance transactions in the user interface of Data Explorer, and suggest how these tools can be applied in general.

1. Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) describes an iterative process in which a human domain expert guides and refines the analysis of large data objects. In each iteration a data mining subprocess identifies patterns, features and clusters found in the data, and then displays those discoveries in a manner appropriate to the human domain expert. That domain expert then attempts to suggest and resolve the relationships and meanings in those discoveries, and thereby direct the next iteration of data mining. Simply put, KDD is "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad, et al., 1996, p.6). KDD is a "human-in-the-loop" process grounded in the concept that random, undirected data mining is unlikely to produce meaningful knowledge.

[1] 215-C Computer Building, University Park, PA 16802, ray-masters@psu.edu
[2] 302 Walker Building, University Park, PA 16802, edsall@geog.psu.edu

2. Display and Meaning

Cartographers and graphic designers have long known that the design of visual representations of information significantly influences the understanding and mental processing of the information displayed. The design of symbols and their attributes (color, shape, size, position, etc.) has been formalized according to their perceptual characteristics. Certain attributes of a graphical display are known to prompt specific insights because they invoke learned or embedded knowledge *schemata* (like "larger = more") (Mark 1989; MacEachren 1995; MacEachren 1998). Much of the knowledge schemata prompted by symbols are the result of our interaction with the environment. For example, we come across a tall tree and assume it is of greater magnitude (age) than a shorter one, which leads us to understand that a larger symbol on a display implies an object or value greater magnitude than a smaller symbol (Lloyd 1997).

Certain symbolization techniques in static cartography are generally thought to be appropriate for the representation of certain types of data. For example, the representation of normalized, quantitative data aggregated to geographic enumeration units, like data on population density or tax rates, is best represented using a choropleth map with a sequential color scheme (Brewer 1994; MacEachren 1995). Representation techniques in cartography are, to a certain extent, application-specific: the display evokes certain cognitive responses, given the user's learned and embedded notions of the symbols or the phenomenon itself. The effectiveness of some representations depends on the expertise of the percipient. A certain orientation of uncrossing curved lines over a map might indicate, to a climatologist, the presence of a significant atmospheric phenomenon (and its associated climate effects), yet to a beginning earth science student, the same representation may prompt no conceptual model of the atmosphere at all.

In a sophisticated, interactive visualization environment, the perception and corresponding understanding of symbols take on added complexity. The display of information can be manipulated by the user in ways that can alter the understanding of the phenomenon represented. It follows, then, that the design of the interaction methods provided to the user should prompt certain embedded schemata, which in turn have an impact on the understanding of the data in the visualization.

The ability to interact with a representation expands the potential for insight by allowing users themselves to become designers of the representation. Static graphic displays are criticized because they represent a model of reality belonging to only one individual (the artist or cartographer) (MacEachren and Monmonier 1992; Wood 1992). That model is then imposed, through the use of symbolization techniques, on subsequent users of the graphic. Far from being objective realizations of reality, maps and other graphics tend to have biases, intentional or unintentional, which can influence the amount or type of information that is learned from the display. Interactive displays, because their symbolization methods can be adjusted on the fly, can be seen as tools which overcome (to some extent) those inherent biases, as much of the process of generating the display (classification, color scheme, level of detail, etc.) can be adjusted by the user to fit his or her own analysis needs.

Visualization environments often provide, thanks to their interactivity, multiple perspectives on information. As software like geographic information systems and Internet map servers becoming ubiquitous, users of geographic information will no longer be satisfied with one particular view of geographic reality. The myriad of interaction techniques in such software permit displays to be created and re-created continuously; in just one hour's use of ArcView GIS, an introductory GIS student can see literally dozens of different perspectives on the same information. Visualization environments must take advantage of this epistemological shift in how humans learn from graphics – from the "one optimal representation" paradigm to that of "multiple views leading to a solution" – and provide users with a wide variety of representation opportunities, through the use of innovative and creative ways of displaying and interacting with information (Swayne, Cook et al., 1997; Schmid and Hinterberger, 1994; MacDougall, 1992)

However, just like symbols on a static display can constrain or otherwise influence the insight generated by a graphic image, interactive displays in a visualization environment can also constrain insight because of the finite types of manipulations afforded the user by the environment. If, for example, an animation of a thunderstorm were interactive only to the extent that the user could stop, start, and rewind the animation, important features of the storm might go unnoticed. Those features might be noticed, however, if, the user were allowed to adjust the speed of the animation, or perhaps allowed to reorder the frames or to inspect only every third frame (DiBiase, MacEachren et al., 1992; Kraak and Klomp 1996). The visual symbolization (colors, glyphs, camera angle, etc.) of the display does not change, but the variety of methods for interacting with the display can vary, and it is likely that such variation can lead to different insight about the data.

As such, we believe that it is very important to provide, in the user's toolbox, not only a wide variety of ways to symbolize and otherwise visually represent information, but also a diverse set of tools to adjust the display and its symbolization. This was a significant theoretical consideration when specifying the innovative representation and interaction techniques in a prototype visualization system designed for the exploration of geographic information.

3. Software Environment

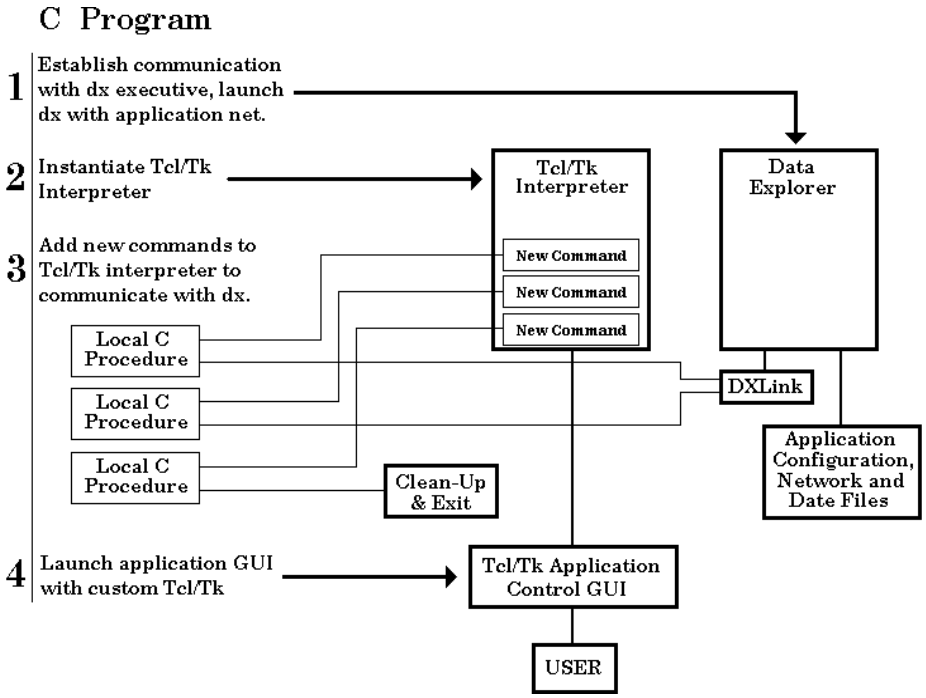
Data Explorer[3] is a general-purpose data visualization software package that runs on most major Unix and Windows 95 – Windows NT platforms. Data Explorer (DX) employs a dataflow client-server execution model that allows one to author visualization applications by selecting modules of appropriate functionality from a large library, and then to describe the flow of data through those modules. The selection of modules and dataflow can be specified in a scripting language, or, more commonly, through a visual program editor (VPE) that provides a point - click - connect graphic interface for application authoring.

The functionality of the provided module library can be extended through a macro-program facility that allows new modules to be created by grouping specific combinations of library modules. Alternately, new modules can be written following well-defined guidelines, and added to local DX libraries. Additionally, one can author applications that link directly to either individual modules or entire applications in DX. These applications often do not use the native DX user interface and instead create a custom environment that provides a look and feel appropriate to a particular discipline or application. In this project, a facility available in DX called DXLink has been used in conjunction with a custom C program and Tcl/Tk[4] to create a unique, appropriate, graphic user interface for controlling a visualization application authored in DX.

Tcl (Tool Command Language) is a simple, yet powerful, platform independent (runs on Unix and Windows) scripting language that is easily imbedded into other applications. Tk is a window system toolkit that adds the functionality of creating and manipulating very sophisticated graphical user interfaces. Tcl/Tk scripts can run standalone, be linked with C programs, or extended over the Web. Tcl/Tk provides a simple, pragmatic, yet elegant, way of defining a graphic user interface that can control complex processes in a visually intuitive fashion.

The total visualization application exists in three parts: 1.) a DX program that performs the appropriate analysis and display; 2.) a Tcl/Tk script that presents and manages the graphic user interface; and, 3.) a C program that defines the execution context and linkages between Tcl/Tk and DX.

A feature of Tcl/Tk is the availability of a custom command interpreter from within a C program, and an important element of this is the ability to extend the functionality of Tcl by adding custom commands that are defined as C procedures. The DXLink facility of DX provides an interface between a C program and the Data Explorer User Interface and the Data Explorer Executive.



Functions are available to load and exit DX programs, set and retrieve named variables, control execution, handle errors, process configuration files and scripting commands, control window visibility, and define messaging between the C program and DX. Creating custom commands that Tcl/Tk interactors can access and use to control DX is accomplished by writing C procedures that invoke DXLink facilities and then adding those procedures to the repertoire of commands recognized by the Tcl/Tk interpreter. If only

[3] www.opendx.org
[4] www.scriptics.com

one possible DX application can be executed, it can be started in the C program and the control of it passed on to the Tcl/Tk interpreter. If more than one DX application is possible, specific individual invocations can be defined and added to the Tcl/Tk interpreter for user selection. Alternatively, a command can be created that uses a character string supplied by the user interface to determine which DX application to invoke.

A simple C procedure to read a Tcl variable,, send it to DX as a character string, and re-execute the DX application with that new value is:

```
int dxproc(ClientData clientData,
Tcl_Interp *interp,
        int argc, char *argv[]) {
        char *sealev;
        char slvl[3];
        sealev =
Tcl_GetVar(interp,"sealev",0);
        slvl[0]=*sealev;
        ++sealev;
        slvl[1]=*sealev;
        slvl[2]=0;
        DXLSetValue(conn,
"contour_line_value", slvl);
        DXLExecuteOnce(conn);
        return TCL_OK;}
```

That procedure is defined to Tcl/Tk as the command, "dx" by this invocation in the C program:

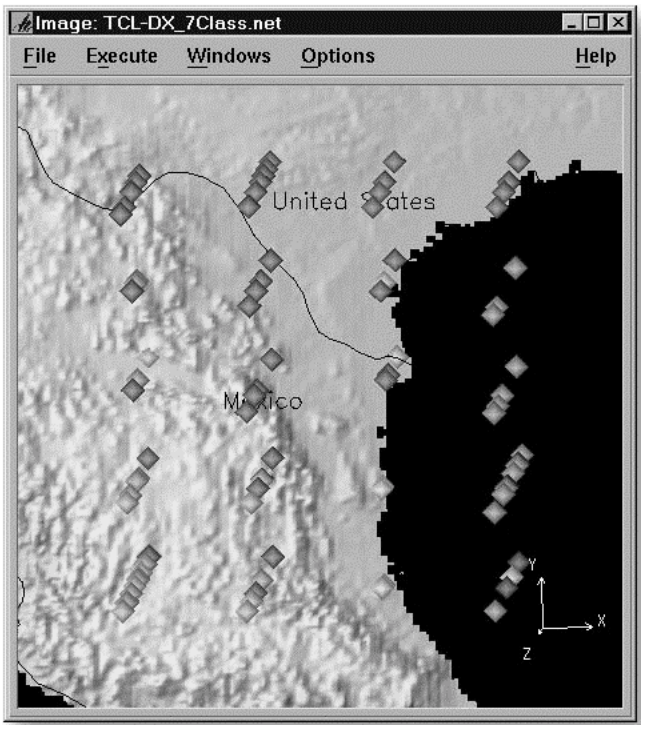
```
Tcl_CreateCommand(interp, "dx", dxproc,
(ClientData) NULL,
(Tcl_CmdDeleteProc *) NULL);
```

Thus, when the Tcl/Tk interpretor executes the following Tcl statement (actually, a push-button with the text, “dx”, appearing on it):

```
button $w.buttons.code -text dx -
command "dx"
```

it will pass control back to the dxproc procedure which will read the current value of the variable, "sealev" (which the user has somehow defined through the graphic user interface), pass it along to DX in the appropriate form, request a single cycle of execution, and return control to the Tcl interpretter.

Every user interaction to DX can be managed in this fashion, from simple, context-based scalar input to sophisticated launches of other application programs that recalculate massive streams of data that ultimately arrive back at DX for additional inspection and interaction. Indeed, visualization applications that are authored within a Tcl/Tk context can easily cascade through whatever analysis and display software a user feels is appropriate to the current insight they glean from the visualization process. For example, a C program can define, launch and manage a DX application. Depending on the resulting DX analysis, additional C programs, additional Tcl/Tk scripts, and additional applications (say, ArcInfo or Tecplot or even locally-written programs in any language) can be launched either serially or in parallel with the current application, on the local host or on remote resources, hidden or visible to the user. In this manner, the most appropriate visualization software environment for an application can be presented, even when that context is not known ahead of time and/or



continually changes throughout the visualization process.

Figure 2. A Geoview

4. Smart Interactors

Three dynamically linked representational structures were implemented as the initial prototype for display and interaction: geoviews, three-dimensional scatterplots and parallel coordinate plots. Each structure is controlled through multiple interaction forms that allow the domain expert to manipulate parameters determining the data-to-display representation.

Geoviews (Figure 2) are familiar three-dimensional map views in which geographic space is represented as a base map and the third dimension is used to display some data attribute with geographical associations.

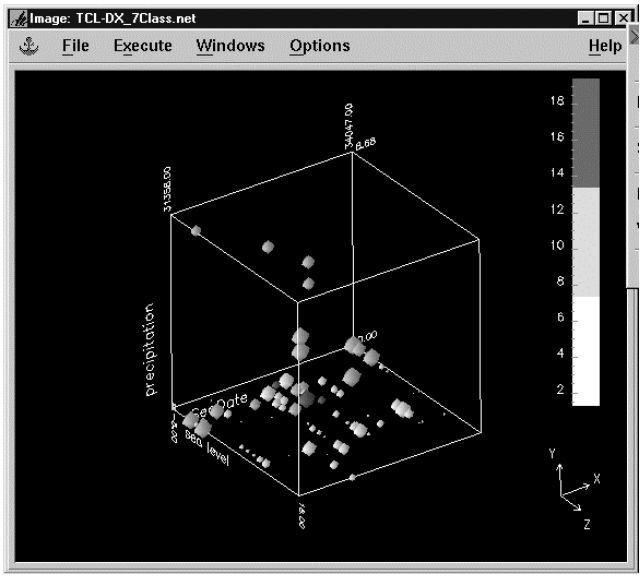


Figure 3. A Three-Dimensional Scatterplot

Three-dimensional scatterplots (Figure 3) display the relationships of any three elements of a data vector by assigning each element to a dimension of a cube. The numeric range of each element being visualized is normalized and assigned to a side of this cube. In assigning spatial characteristics to non-spatial data we capitalize on the metaphor of clustered=similar, dispersed=different as we view this display.

Parallel coordinate plots (Figure 4) utilize several vertical axes, one for each variable in the data vector. All data ranges are normalized to the same vertical scale. Line segments connect all the data values for a particular observation, one vertical axis to the next. This produces a unique shape or "signature" for each observation. Parallel coordinate plots (PCP's) are particularly productive in uncovering relationships among many data variables (Inselberg, 1997). In this implementation the domain expert has control of the assignment of data variables to the vertical axis as necessary to improve the display and interpretation of the signatures.

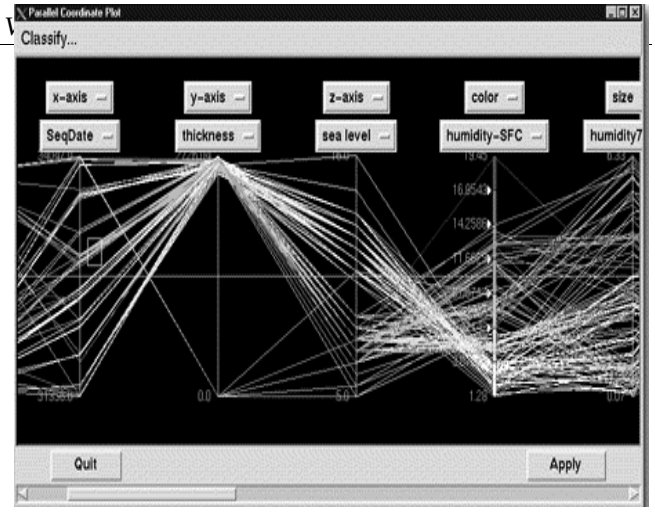


Figure 4. A Parallel Coordinate Plot

Each of these representational structures can be independently manipulated by the domain expert, and the linking of all these structures (Figure 5) allows changes to one structure to be reflected on the other two if the change is appropriate to the individual representational structures.

5. Interaction Modes

These representational structures present a variety of interaction opportunities for the domain expert: assignment, brushing, focusing, colormap manipulation, viewpoint manipulation, and sequencing.

Assignment allows the domain expert to compare the distributions of two or more variables in the data record by placing them in close proximity to one another. This functionality is available in the assignment of variables to axes in the parallel coordinate plot, and in the assignment of variables in the three-dimensional scatterplot. By working with various assignments in these two structures, the domain expert can search for clusters in different combinations of variables.

Brushing is a technique that allows the domain expert to highlight any set of entities that appear interesting in any one of the representational structures and observe the effects in the other linked structures. For example, highlighting a cluster in a three-dimensional scatterplot and observing the changes in the geoview could help one determine if there are corresponding relationships in spatial or temporal space.

Focusing allows the domain expert to limit the inclusion of data in the linked structures to a specific range of values for any particular variable. As with brushing, it is most useful with two or more linked structures.

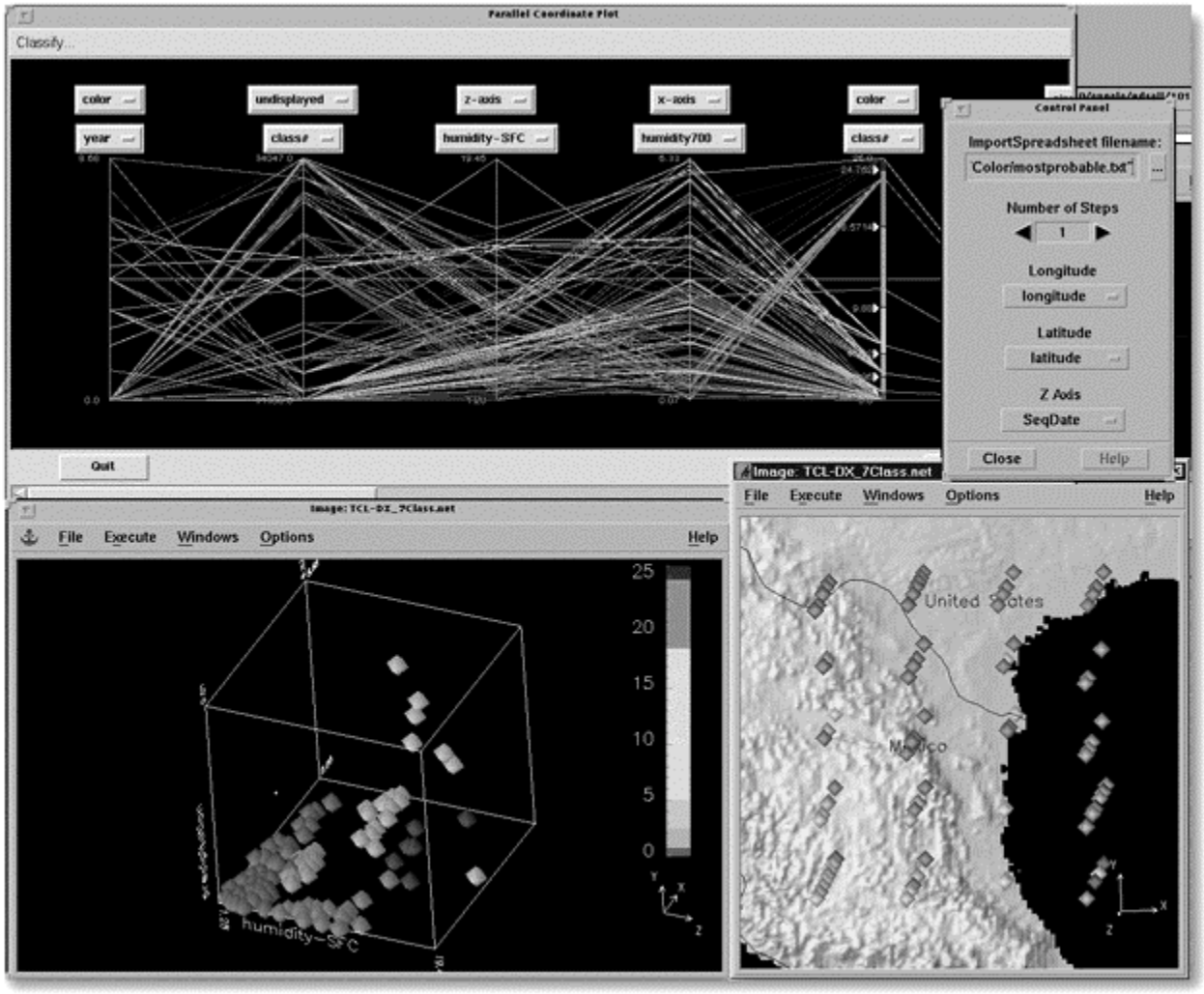


Figure 5. All Representational Contexts

Colormap manipulation allows the domain expert to assign specific colors to individual observations or ranges of values for any particular variable in an observation and observe the effects in all of the representational structures.

Viewpoint manipulation allows the domain expert to dynamically explore the orientation of three-dimensional displays to provide depth cues that help in the perception of relationships in the data.

Sequencing provides an added dimension to all the representations. Trends over time are easily distinguished, as well as non-temporal iterations through any data variable by creating and then animating through uniform data bins.

6. Application Notes

These smart interactors were applied to an analysis of gridded climate data covering northern Mexico and the southern United States. The data were continuous in both space and time. Environmental scientists, particularly climatologists, were the domain experts. The initial data mining operations were performed using Autoclass, a public domain software package that provides unsupervised classification based on Bayesian statistics. The classes that were identified in this mining operation were subsequently visualized and interpreted effectively in the linked representational contexts.

7. Summary

The transition from data to knowledge often fails to occur due to discrepancies in the human-computer collaboration. While advances in computing have produced sophisticated visualization tools that are well within the grasp of today's researchers, work that addresses the human side of the collaboration has lagged. Real progress in these tools can only occur when issues of human cognition, analysis, and creativity are addressed in software interfaces. Interactors need to be more than simple selectors or vehicles to specify numerical or character data; interactors should pique the creative problem-solving process and create opportunities for real insight and understanding to emmerge.

Acknowledgments

The authors wish to acknowledge the support and supervision provided by Professor Alan M. MacEachren and the members of the GeoVISTA[5] Center at Penn State.

References

Brewer, C. (1994). Color Use Guidelines for Mapping and Visualization. *Visualization in Modern Cartography*. A. M. MacEachren and D. R. F. Taylor. Oxford, Pergamon**:** 123-148.

DiBiase, D., A. M. MacEachren, J. Krygier and C. Reeves (1992). "Animation and the Role of Map Design in Scientific Visualization." *CAGIS* **19** (4): 201-214, 265-266.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., 1996, From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining* (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, Menlo Park, CA: AAAI Press / The MIT Press), pp. 1-34.

Inselberg, A., 1997, Multidimensional detective, IEEE Visualization '97, October 19-24, Phoenix, AZ, IEEE Computer Society Technical Committee on Computer Graphics. pp 100-107.

Kraak, M. J. and A. Klomp (1996). "A Classification of Cartographic Animations."

Lloyd, R. (1997). *Spatial Cognition*. Dordrecht, Kluwer.

MacDougall, E. B. (1992). "Exploratory Analysis, Dynamic Statistical Visualization

and Geographic Information Systems." *Cartography and Geographic Information Systems* **19** (4): 237-246.

MacEachren, A. M. (1995). *How Maps Work*. New York, Guilford.

MacEachren, A. M. (1998). Design and Evaluation of a Computerized Dynamic Mapping System Interface. Washington, DC, National Center for Health Statistics.

MacEachren, A. M. and M. Monmonier (1992). "Introduction." *Cartography and Geographic Information Systems* **19** (4):pp 197-200.

MacEachren, A. M., M. Wachowicz, R. Edsall, D. Haug and R. Masters (1999). "Constructing Knowledge from Multivariate Spatiotemporal Data: Integrating Geographic Visualization (GVis) with Knowledge Discovery in Databases (KDD)." *International Journal of Geographic Information Science* **13** (4):pp 311-334.

Mark, D. (1989). Cognitive Image-Schemata For Geographic Information: Relations To User Views And GIS Interfaces. *Proceedings of the GIS/LIS '89*, Orlando, Florida, AAG.pp 551-560

Schmid, C. and H. Hinterberger (1994). Comparative multivariate visualization across conceptually different graphic displays. *Proceedings of the 7th International Conference on Statistical and Scientific Database Management*, Charlottesville, VA, IEEE Computer Society Press.42-51

Swayne, D., D. Cook and A. Buja (1997). "XGobi: Interactive Dynamic Data Visualization in the X Window System." *Journal of Computational and Graphical Statistics* **7** (1):pp 113-130.

Wood, D. (1992). *The Power of Maps*. New York, Guilford

[5] www.geovista.psu.edu