3

Coordinated Multiple Views for Exploratory GeoVisualization

Jonathan C. Roberts
Computing Laboratory, University of Kent

3.1 Introduction

Over recent years researchers have developed many different visualization forms that display geographical information, from contour plots and choropleth maps to scatter plots of statistical information. Each of these diverse forms allows the user to see their data through different viewpoints. In fact, it is often the case that, when the user sees the information through different views and in different ways, they get a deeper understanding of the information.

Geographical databases often hold a diverse range of different data and are thus complex to understand. For example, spatial data may be held that explains a particular geographical landscape, the land usage or details of buildings found on that land; non-spatial elements may also be stored which detail land ownership, the salary of the land owner and other statistical information about the land owner. The users’ goal is to gain knowledge of that information and make sense of a large volume of potentially diverse data with multiple components and different data types.

Thus, to comprehensively understand the information contained within any complex geographical database, the user would need to select some information to display, present it in different forms, manipulate the results, compare objects and artefacts between views, roll back to previous scenarios or previous sessions, understand trends by seeing the data holistically as well as specifically, and finally take measurements of objects and areas in the display. Consequently, developers have created specific exploration environments that utilize coordinated multiple view techniques (CMV), where each of the views are linked together such that any user manipulation in one view is automatically coordinated to that
of any other and the environment. These exploratory visualization (EV) environments are highly interactive systems and rely on the premise that 'insight is formed through interaction'.

If we, as developers and researchers, are to provide the best possible environment for the users, we need to take stock of what functionality is being provided by current tools and examine what we are doing well and what we are perhaps not doing so well. This chapter is a systematic analysis of exploratory geovisualization using CMV techniques. We consider the strengths and weaknesses of the area and make a comprehensive review of CMV for geovisualization.

**3.1.1 Exploration**

The goal of exploration is to search, locate and find out something new. A user starting out on the investigation process may not know anything about the data, let alone the questions to ask. Thus the system should help the user to not only display visual results of the information but also browse and locate pertinent information. Visual exploration, in particular, enables the user to visually investigate the data. In short, the visual exploration enables the user to try out some parameters, instantly view the results of the parameter change, manipulate the data through selection and highlight operations and relate that information to other sources and visualizations. The user may directly select some interesting elements using a bounding box tool, see the selected list of results in both a scatter plot visualization and a textual list of results, and perhaps edit the items on the selection list by adding some more or deleting some in the list. Thus exploration is part of a larger discovery process.

Various researchers have described this discovery process by a variety of models. In geovisualization specifically DiBiase (1990), focusing on the role of visualization in support of earth science research, summarizes this discovery process as four stages. First, exploration reveals the questions and achieves familiarity with the data through testing, experimenting, acquiring the right skills and learning about the underlying model. Second, the user needs to confirm the relationships that exist in the data through comparison operations, relating the information to other explorations and disproving other hypotheses. Third, the results are synthesized; this is achieved by identifying the pertinent features and summarizing the content. Fourth, and finally, the results are presented and they are taught and demonstrated at professional conferences and in scholarly publications.

DiBiase’s model emphasizes the uncertainty and goal-seeking nature of the discovery process; it is a conceptual model that encourages the user to personally explore the information, thus to gain a better understanding of the information before presentation to a wider audience. Obviously, these traits are highly important in exploratory geographical visualization and developers need to decide how these features map to individual tools. However, the model de-emphasizes the processes of data preparation and simplifies how the results are gathered and presented to the user.

Sense-making models, on the other hand, are more goal-oriented models and are often discussed in the context of intelligence analysis. These models emphasize the whole process from data preparation to hypothesis presentation (Thomas and Cook, 2005). Specifically, sense-making according to Russell et al. (1993) is the process of searching for a representation
and encoding data in that representation to answer task-specific questions. One of the most comprehensive sense-making models is by Pirolli and Card (2005), Figure 3.1. In their schema model the analyst searches and filters the data for relevant information, which provides a demonstration set that is stored for future reference (known as a shoebox). Specific and relevant information or inferences may then be extracted and stored to provide evidence files. Then the data is structured, organized and represented in a schematic way. This is both to highlight interesting facts and to confirm relationships within the data; it matches with the first and second stages of DiBiase’s model. This organizational part of the process may be achieved through the representation of the data by an informal diagram or a complex visualization. Then the case is built, and finally presented to the client. This is a process-oriented model, where a developer can clearly map the stages into processes of an exploratory visualization tool.

Exploration itself is a process that tends to generate many possibilities and widens the search space. Techniques of generating multiple views, selecting and highlighting elements, zooming and displaying additional detail all help the user to explore. However, the user still needs to draw conclusions and present that information to colleagues. Thus, for exploration tools to be successful, developers need to provide methods to both widen and narrow the solution space, support the user through an exploration and support them in their presentation of that learned information.

Therefore, exploratory environments require a wide range of features: (1) tools that perform effective data preparation; (2) informative visualization techniques that display the information comprehensively and clearly and that display the information in different forms; (3) interaction techniques that allow the user to manipulate the information; and (4) extensible and easy-to-use tools and toolkits. These four parts are extremely important
for exploratory visualization. They match well with the models of both DiBiase (1990) and Pirolli and Card (2005); hence, in the next few sections we discuss issues of where we are with exploratory geovisualization under these four headings.

3.2 Data preparation

In our information age data is being produced at a phenomenal rate. In some projects data is being produced at such a rate that is only possible to store it, and with current technologies and techniques it is often impossible to analyse it fully. An ever-increasing amount of data is at high resolution, with companies and agencies creating and processing high-quality digital land use and socio-economic data with spatial attributes. With the ubiquity of phones and global positioning devices, more spatial and time-stamped data is being stored.

Additionally, such geospatial databases are collected by different researchers, created by an assortment of sensors, saved in different formats, held physically at various locations around the world and incorporate many variables and types of data. Hence, the user is faced with several challenges when they wish to integrate the geographical data from multiple databases, across different domains and potentially use the data for a purpose that was not originally envisaged. Each of these challenges impinges upon Exploratory Visualization.

3.2.1 Too much data!

The total quantity of data is often one of the most crucial challenges for a developer to consider. Size certainly impinges upon the processing time. It takes longer to process huge amounts of data, especially because users wish to mine similarities and structure within the information, hence parts of the data need to be correlated with other elements. Friesen and Tarman (2000) write: ‘even at tomorrow’s gigabit-plus networking bandwidths, sharing these enormous data sets with distributed sites becomes impractical, and we’ll need high performance visualization resources to have any hope of near-real-time interaction or observation of this data’. Solutions to processing this huge quantity of data include: parallel and distributed computing or remote computing solutions (Harwick, Coddington and James, 2003), and more recently interest has turned toward service- and grid-based architectures (Aktas et al., 2005; Brook et al., 2007). Spatial datasets certainly bring specific challenges that high-performance computing methodologies can address, but in reality few researchers have looked at solutions to these challenges. Reducing the size of the data obviously would make the exploration more interactive.

Methods such as filtering generate smaller demonstration sets (shoebox datasets) that would be processed more readily, take up less screen space and allow the user to focus on relevant data. Spatial filtering may be achieved through cropping, sampling, averaging (binning), partitioning, clustering and aggregation (Tang and Shneiderman, 2001). These techniques can be achieved through pre-processing the data or closely linking the filtering with the visualization (dynamically visualizing the results). Dynamic queries (Ahlberg and Shneiderman, 1994) permit the user to directly interact with the visualization by adjusting
sliders, buttons and menu items to filter the data and instantly view the result in the display. Furthermore, other pre-processing may be achieved through knowledge discovery (KD) and data mining (DM) methods to find patterns in large spatial datasets (MacEachren et al., 1999). In addition, the plain quantity of data often exacerbates the problem. Not only does it slow the interactive process, but saving and loading the data takes time and effort. Hence caching techniques would also speed up the interaction.

Storing and caching the processed (shoebox) data are important not only for interactivity, but they can also be utilized by the investigator as a temporary copy or by other investigators to confirm a hypothesis. Thus, there is a need for both curation of geographical data (especially shoebox information) and storage of information on the provenance of the data and the session. For instance Thomson et al. (2005) discuss issues of lineage and provenance in geovisualization and Koua and Kraak (2004a–c) discuss some ideas on tools to support ‘knowledge construction throughout the exploration process’, but much work is still to be achieved.

While filtering methods pre-process or dynamically select a smaller demonstration set, other processing methods aim to group, aggregate and assign structure to the data. Patterns that are assigned through KD and DM techniques can be visualized. Many of the techniques are non-trivial as it is often not possible to provide brute-force algorithms to find similarities within the data (Miller and Han, 2001). Parameters of these algorithms can be tightly coupled to the interface, such that the results are consequently updated (MacEachren et al., 1999; Koua and Kraak, 2004a–c). For example, by adapting values to (say) the \( k \)-means algorithm, the user can explore different partitioning configurations (Torun and Duzgun, 2006). Thus, it is possible to combine knowledge discovery, data mining and visualization techniques. In fact, various researchers have called for the closer integration of data mining and visual exploration techniques (Kreuseler and Schumann, 2002; Kocherlakota and Healey, 2005). Classification hierarchies can be used to explore the data (Kemp and Tan, 2005); these can be displayed as graphs and other structured organizations (Rodgers, 2004). Classifications are important, and if, in particular, they are visualized, they allow the user to understand the underlying structure of the data and furthermore can be used for filtering and processing. Diverse categorizations provide different ontologies (Arpinar et al., 2006). These classifications can then be visualized separately and thus provide multiple views on the same information; see Section 3.3. Not only are there challenges with processing the quantity of data, but there are also challenges in displaying it all at once. Venkataraman et al. (2006) write: ‘LIDAR data can easily exceed 100 Mpixels making it impossible to have detail as well as context on a single desktop display’. Some researchers, particularly from the database community, have looked at pixel-based visualizations (Keim, 2000) and some work has been done in spatial visualization using pixel displays (Keim, et al., 2004), but there is still a limited number of pixels on a screen and hence a limit to the amount of information that can be displayed at once.

The crucial factor is not just the size but the quantity of pixels it can display. High-quality large-format cinema-style video projectors are still very expensive and have huge running costs, typically because of the short life-span of the bulbs. Hence researchers have developed tiled screens. Such multiple display devices have been successfully used for geospatial exploration (Venkataraman et al., 2006). However, these displays have their problems. They need special device drivers to control all the multiple displays, the screens need to be housed in a customized rig, there may be problems with the viewing angles, and there will always be a gap between each screen due to the screen’s housing.
3.2.2 Complex and diverse datasets

Not only is it useful to view the information through different forms, and gain better understanding through looking and manipulating the data, but value can be gained by algorithmically fusing information from assorted databases. For instance, a second database may provide information that was missing from the first, errors or discrepancies of the data may be highlighted through the amalgamation of data, and information may be inferred through the integration of the two datasets.

Furthermore, large and complex geographical datasets often contain missing or erroneous information. Missing data obviously affects how the user explores the information. For missing data, the developer needs to work out what to do with it: whether to ignore it by pre-filtering it out, make an assumption and substitute the information, or classify it as missing and treat it specifically. If it is going to be included then the data-processing algorithms may need to be re-written to specifically handle these cases. For instance, what happens when three values are averaged when one value is null? Likewise, erroneous data can be deleted or flagged as being potentially wrong, but again these concepts need to be fully integrated into the system, from data-processing and visualization to interaction and manipulation. If users are to explore the complete dataset, then it is imperative that the developer integrates methods that deal with this information. It is far too easy to develop tools that merely ignore this type of data. Often missing or erroneous information can help the user infer some fact or it can be used to support a hypothesis. Seminal work in this area is Unwin et al. (1996) in their MANET (Missings Are Now Equally Treated) system. However, current systems still do little to integrate this uncertain data and much work needs to be done here to fully integrate and allow the user to appropriately manipulate missing information.

3.2.3 Data processing challenges

The sheer size, complexity and diverse nature of geographical datasets definitely have consequences for exploratory analysis. Certainly researchers have developed some clever and useful algorithms to address many of these issues, but the main limitation is that they are not commonplace in most of the general purpose geovisualization tools. For example, techniques such as parallel algorithms or the use of remote high-performance computers have their benefits, but although much work has been done looking at such techniques individually and their application to geographical visualization techniques, they are not commonplace and not included natively in many exploratory geovisualization systems. Data abstraction techniques also have the desired effect of speeding up the processing of large datasets, and again a reasonable amount of research has been achieved. However, although visual filtering methods are in widespread use, there has been little work on tightly integrating data mining techniques with exploratory visualization techniques.

Furthermore, additional research on the curation of geographical data is required. That is, both the storage of temporary datasets from an exploration and also details of the operation history (the commands used in the exploration) should be saved. This would enable provenance of the exploration and permit additional researchers to reproduce and confirm any results.
The use of dense pixel visualization for the representation of abstract data is an area of research that certainly has potential. It permits large amounts of data to be displayed and creates a holistic view of the information. However, such techniques are currently underused in geovisualization.

Finally, there are certainly challenges with incorporating non-existent or erroneous data into a geovisualization system, although there are clear benefits to incorporating such data. Some work has been achieved in visualizing uncertain information, but researchers who initially store the data need to save and appropriately mark up the information, and system developers need to integrate uncertainty concepts throughout the whole system, such that uncertainty is a principal component of the system.

### 3.3 Informative visualizations

The canonical form for geographical display is certainly the map. However, there are many different types of visualization that are used in geovisualizations. In multiview exploratory systems these additional (non-map-based) realizations are often used alongside the map display; all user controls and operations may be associated with any or all of the views to provide a powerful exploratory environment. In this section we discuss various terminologies and implications of multiple views; we review a variety of display forms that are used in geovisualization and consider representation and re-representation methodologies. The forms detailed within this section are not meant to form a comprehensive review, but are intended to give the reader an understanding of the breadth of techniques that are currently available. Indeed, this book and others, such as Exploring GeoVisualization (Dykes, MacEachren and Kraak, 2005), provide many more examples of geovisualization forms.

#### 3.3.1 Multiple views

The term multiple-views is all-encompassing; it includes any system which allows direct visual comparison of multiple windows, including visualizations from different display parameters. It usually implies that the visualizations are viewed in a desktop windows environment. However, researchers have also used the term to describe various tile-based displays. There are three further implications. The first is that the operations are coordinated and that any operation is linked with any other view, hence the more specific term of multiple coordinated views may be used. The second is that the user can instantiate any number of views that they require. The third is that the views are encapsulated in their own window. By having these multiple representations encapsulated in separate windows, the user can easily compare two (or more) representations side-by-side. In fact, there are many examples of developers utilizing dual-views (Convertino et al., 2003) to allow the user to compare the information side-by-side.

The term multiform on the other hand is specific (Roberts, 2000). It describes how the information can be displayed using different types of representation. For example, one view may show the information as a map, while another window shows the information in a table. In this example, the user can select regions on the map and display the statistical information in the associated table. The environment may also allow the user to investigate...
another location on the map, and thus the numerical data can be displayed in another table, in a further window. There are other terms in the literature that have specific meaning. For instance, alternative representations has been used to explain multiple forms (Koua and Kraak, 2004a–c), but the phrase is also used to promote multiple opinions (Roberts and Ryan, 1997). Users may have a different understanding of the data and hence provide various interpretations of the phenomenon; for example, historic information may be sparse, there may be disagreements between eye-witness reports or interpretations of the data may vary. In archaeology, experts may disagree and their suppositions may change over time as they become influenced by other people’s opinions. Alternative representations are also useful in education, in that the learner better understands a process if it is explained or demonstrated in multiple ways.

Multiple views are also used for control. Many of the views utilize direct manipulation, where the user directly controls the information in the window. Consequently, a user may find it easier to manipulate the data in one type of representation in comparison to another; for instance, the user is given two views, one being a three-dimensional network view showing the position of objects found from an archaeological excavation and the other an alphabetical list of the objects found. If the user wished to see where all the pottery vessels were located, then they could select ’pottery vessels’ from the textual list to highlight all those in the three-dimensional view. However, if they wanted to see what information is located near a specific location then drawing a bounding box on the plot would be easier. This is also a demonstration of a master and slave relationship, where one view directly controls another. Another use of multiple views is to create focus and context, where one view depicts a summary of all the information with the other showing detail. In addition, both focus and context information can be displayed in one unified view using distortion techniques.

In a windows environment it is possible to gain information on demand through popup information; these popup views provide specific information dependent on the context in which it was requested (Geisler, 2000). In virtual reality a similar style of master and slave relationship exists. The concept is named view on a bat or worlds in miniature (Stoakley, Conway and Pausch, 1995). In this case the user has a virtual bat that moves around the virtual world with the user. Different pieces of information can be mapped onto the bat such as a geographical map of the area or some extra information that is relevant to the context of the user at that point in time. Similar concepts occur in other domains. For instance augmented table-top environments use markers to instantiate additional information (Hedley et al., 2002).

Finally, the last type is small-multiples (Tuft, 1990). By displaying the data in many different small representations, the user can perceive trends in the data. It is through an overall perception of the texture that the user perceives trends in the information.

### 3.3.2 Multiple forms

There are many types of geographical visualizations. Inspired by the categorization of Lohse et al. (1994), the various geovisualizations can be divided into seven categories: Maps/Cartograms, Networks, Charts/Graphs, Tables, Symbols, Diagrams and Pictures.

Maps communicate spatial meaning and there is a direct association between the physical space and the represented space. There are clearly many different types of maps that are used in geovisualization. Geographical maps aim to represent proportions and the geography of our world and can be two- or three-dimensional. For example, elevation information, such
3.3 INFORMATIVE VISUALIZATIONS

as from LIDAR data, can be mapped to colour or height. Two-dimensional maps are often used as the main reference visualization, with other statistical information being layered on top. Furthermore, different types of data, such as agriculture, transportation, boundaries or population density, can be represented as different layers of the map. Other forms of maps include choropleths, where the land areas are shaded with values proportional to statistical measures of that land, while cartograms distort the map dependent on another statistical measure, e.g. the time it takes to travel from one point to another.

Networks describe relational and associational information, e.g. that a connects to b and then c. Networks include trees, hierarchies, routing diagrams and graph visualizations. A well-known hand-crafted network is that of Harry Beck’s London underground map; it is really a network presentation because it represents relational information of the underground rail lines. In fact, aspects of Beck’s map have inspired network layout algorithms (Stott and Rodgers, 2004). Network visualizations have been used in geovisualization to represent various types of associated data and have been realized through different layout strategies. Rodgers (2004) provides a good overview of graph techniques for geovisualization. Depictions range from node-edge graphs and treemaps to trails.

Charts display statistical or mathematical information. In this category we include line graphs, histograms, circular histograms, pie charts, surface plots, scatter plots and parallel coordinate plots (Edsall, 2003). Each of these visualizations addresses a specific need; line graphs and histograms visualize continuous data on a two-dimensional plot, surface plots display continuous data in three dimensions, bar charts and pie charts display quantitative information and parallel coordinates display multidimensional data, while scatter plots allow users to see clusters.

Tabular and matrix layouts are popular for displaying statistical quantities and numerical information contained within geographical databases. They are familiar forms and as such can be extremely useful. Spreadsheets enable large amounts of numbers to be shown and manipulated; with table lens views (which utilizes distortion techniques) even more information can be displayed (Rao and Card, 1994), while reorderable matrix techniques map the values into the size attributes of symbols to allow trends and similarities to be viewed (Siirtola and Mäkinen, 2005).

Symbols may be used in two ways. Either they are used to identify individual aspects of the information, such that objects or buildings can be located on a map, or they are used to notify trends. In the latter case, the trend is understood through perceiving the texture that is formed from viewing multiple symbols close together. For instance, Nocke et al. (2005) plotted multiple icons onto a map to visualize land usage.

Diagrams realize some process, concept or phenomenon; most are hand-crafted to display a particular phenomenon or result. They are often used in teaching to clearly explain a process. There are many examples in the literature, demonstrating phenomena like how volcanoes function, the migration of various animals or the campaigns of Napoleon’s army.

Pictures are often associated with geographical datasets. Aerial photographs or site photographs can be easily associated with ground or land-use data. With the ubiquity of GPS it is easy to take a picture with position information. Applications like Google Maps (maps.google.com; maps.google.co.uk, accessed May 2007) montage multiple aerial photographs to make a detailed view of the Earth, while projects such as ‘MyLifeBits’ from Microsoft Research utilize cameras that automatically take pictures at regular instances throughout the day and correlate them with positional information from GPS (Bell, Gemmell and Lueder, 2007).
3.3.3 Challenges to create informative visualizations

The geographical and cartographic communities have many years of experience. Developers know how to generate effective and clear map representations. Furthermore, the geovisualization community has borrowed from this long heritage and have created many dynamic and interactive exploratory visualization systems. Systems such as cdv (Dykes, 1997), Mondrian (Theus, 2002) and CommonGIS (Andrienko and Andrienko, 1999) include many different forms, from scatter plots, bar charts and line graphs to parallel coordinate plots. However, there is too much choice. Gahegan (1999) described one of the major challenges with exploratory geovisualization as being the ‘vast range of potential approaches and mappings’, and the problem still exists today.

A user does not necessarily know which representation is most effective; consequently they may miss out on some important information because of the representation methodology they used. Thus, there is a need to examine which representation form is most effective for a given task, and methods are required to aid the user in their exploration.

3.4 Interaction and manipulation

Both the developer and the user need to make choices, not only how to display the data, but how to interact with the information. There are usually lots of parameters that can be changed to alter the visual appearance of the display. In fact, just the simple act of altering the colourmap can have a significant impact on what is emphasized and what is hidden in the display.

From a developer’s standpoint there are some important steps required to generate a representation. There are different models to follow, but the principles remain the same. First, a demonstration dataset is created from the original data. This is achieved through filtering, aggregating and processing the original data. Second, the developer must choose what visual form to use and then decide how to map the data into it. Often the data is first mapped into an intermediate form, which is then rendered and displayed. The intermediate form is a data structure that holds summary or pre-calculated information. For instance, if the visualization is a three-dimensional surface, then triangles, coordinates, colour and other geometric information may be stored. Other data may also be stored, such as cached values to speed up rendering; associated arrays may be calculated to control coordinated views, and a list of the user’s selected elements. Third, the user then manipulates the parameters to change how the information is displayed.

One of the initial choices of a developer and user is to decide where the result is displayed, that is, what alters when the parameters are changed. Parameter changes can be made in four areas: first, to control the appearance of the demonstration data; second, to control how the data is mapped to the display; third, to navigate the information such as zooming, clipping or changing the level of abstraction of the information being displayed; and fourth, controlling the environment and other meta-information associated with the system (such as determining where the views are placed on the screen, loading or saving files or managing bookmarks or controlling other exploration management facilities). Finally, interaction can also be coordinated and thus affect multiple visualizations simultaneously.
3.4 INTERACTION AND MANIPULATION

Replace
On a parameter change the old visualization is replaced by the new.

Replicate
On a parameter change the new visualization is placed in a separate window.

Overlay
On a parameter change the new visualization is merged with the old.

Figure 3.2 On a parameter change the information in the view can be replaced by the new information (replace), displayed in a new window (replicated) or merged with the existing representation (overlay)

3.4.1 Where is the information displayed?

The simple answer to ‘where is the result displayed?’ is that the information contained within the current view is replaced completely by the new information, i.e. when the user changes a parameter a single display is updated. This is often achieved interactively, such that, when the user changes the position of a value slider, the display instantly updates. Such dynamic query interfaces provide pre-attentive visualization of the information. However there are two further models of re-representation: replicate and overlay (Roberts, 2004), see Figure 3.2. Replication occurs when, on a parameter change, a new window appears with the new information. This is useful when it is imperative to compare information from one parameterization to another; but if unrestricted the screen can get cluttered with many windows and then it is difficult to understand what view represents which parameterization. Overlaying the information is the final strategy. There are several ways to overlay information or join information together from different parameterizations. The information could be stacked on top in another layer, or the new information could be merged together to provide a difference view (a difference view explicitly demonstrates the difference between the two parameterizations). Some systems are better at allowing the user to change where the output goes. Module visualization environments (MVE), such as AVS™ (www.avs.com/) and IRIS Explorer™ (www.nag.co.uk/), allow the developer to adapt the flow of information by changing the configuration of the modules. The data flows through different modules (filter, map, render, etc.), concluding in a display module. When the user changes a parameter in a module, the new data flows through the system and the display is updated. However, to create a new view (a replication), the user needs to copy and connect multiple modules in order to perform the operation. The data flow is split and the modules provide a fan-out flow.

Overlaying the information, especially generating difference views, is extremely useful for the user. It shows unambiguously the changes between two parameterizations or between two
datasets. However, it is difficult to design intuitive and useful difference modules. Layering is used in geographical visualization; for instance, different land uses can be layered on top of each other and these layers can be separated in space or time such that the user can explore information on individual layers. However, it can be difficult to perceive the information comprehensively on each layer due to the occlusion of one layer to another. It is possible to imagine other strategies to merge the data, such as incorporating different information through various graphical lenses (Tominski et al., 2006), but most general geovisualization systems do not support such operations.

3.4.2 Interactive filtering

Interactive filtering is an important exploratory technique. The user changes a parameter, or the range of values, to reduce the quantity of information that is being displayed. What once was a dense, overpopulated display can clearly display trends and highlight important outliers. Methods range from constraining ranges (Williamson and Shneiderman, 1992) to changing the bin width of a histogram (Theus, 2002). One important aspect of this type of interaction is that the query immediately updates the display.

**Dynamic queries** originated from work done by Ahlberg, Williamson and Shneiderman (1992). There are many examples where dynamic queries have been used to display geographical information. HomeFinder (Williamson and Shneiderman, 1992) demonstrates how geographical information, in this case houses for sale, can be explored through dynamic queries; IVEE (Ahlberg and Wistrand, 1995), which was later developed into SpotFire™, demonstrates how heavy metals in Sweden can be explored using dynamic queries; Dang, North and Shneiderman (2001) show how choropleth maps can be used with dynamic queries; Burigat and Chittaro (2005) present interactive queries for geographic data on mobile devices. Range sliders can be directly integrated with the visualization; for example, researchers have placed range sliders directly upon parallel coordinate plots (Shimabukuro et al., 2004). They have also been incorporated into many general geovisualization tools (e.g. Andrienko and Andrienko, 2003; Brodbeck and Girardin, 2003; Feldt et al., 2005).

3.4.3 Interactively adapting mapping parameters

Colour is important in geographical visualization. In fact, changing how colour is mapped to the display can considerably alter how the information is perceived. Not only can a colour map affect whether a colour-blind person can perceive the information, but elements may be hidden or emphasized through different colour mappings; colour can permit the user to perceive depth and can be used to delimit areas and borders. It is well known, but infrequently applied, that certain colour maps, such as the rainbow colour map, may mislead the user (Rogowitz and Treinish, 1998). Hence choosing the correct colour map with an appropriate transfer function is important (Brewer et al., 1997; Rogowitz and Kalvin, 2001). Colour itself is one type of **visual variable**, thus it is often possible to exchange colour with another type (such as grey scale, size or texture). However, it is prudent to exchange the visual variable with one with similar traits, such as exchanging one visual variable that displays continuous data with another that is also perceived continuously (Bertin, 1981).
3.4 INTERACTION AND MANIPULATION

Direct manipulation techniques allow the user to point to and pick interesting elements to select, delete or adapt them. Various researchers have used different names to express the same technique, including highlighting (Robinson, 2006), brushing (Carr et al., 1986; Becker and Cleveland, 1987) and painting multiple views (Buja et al., 1991). However, the techniques remain the same: a user visually selects one or more elements, the elements are stored or processed, and the selected elements are displayed to the user through highlighting.

Commonly the selection operation is coordinated to all windows to allow the user to see how elements in one projection are displayed in others. Linked brushing is important, as the user can brush in one view in one projection and see the results of that operation in other dimensions in other views. Linked brushing can also be used to discover outliers between multiform views (Lawrence et al., 2006).

There are different tools to directly select the required elements. Commonly a freehand lasso is used to delimit the elements (Wills, 1996), but the brush can be a point, line or area. For example, Stuetzle (1987) demonstrates point selection and Becker, Cleveland and Weil (1988) describe a line-brush, while there are many examples of bounding boxes (Becker, Cleveland and Wilks, 1987; Buja et al., 1991; Swayne et al., 2003; Ward, 1994, Piringer et al., 2004; Weaver, 2006).

Often highlight is the only operation that developers implement. However, even 20 years ago, Becker and Cleveland (1987) specified four brush operations: highlight, shadow highlight, delete and label. The highlight operation changes the appearance of the selection list in all windows; shadow highlight changes the appearance of the selected elements in the view and removes them from all other linked views; delete removes the elements from the view; while label displays additional information about the selected elements, such as a text label. More recently, researchers have extended these ideas, e.g. multiple brushes or compound brushing. Multiple brushes allow users to control many brushes, with each new brush in a different colour. Compound brushing combines the results from multiple brush operations. Chen (2004) describes a flexible compound brushing system that permits selections to be combined using various logical operations (e.g. AND, OR, XOR), the operations being controlled through a graph layout, while Wright and Roberts (2005) present a direct methodology named click and brush.

Finally, the selected elements need to be visually highlighted. Commonly this is through a colour change, but other methods include linked lines, depth of field, transparency, contour lines or shadows surrounding the elements (Robinson, 2006).

3.4.4 Navigation

Navigation and viewpoint manipulation can come in various forms. With three-dimensional geovirtual environments the user can walk, fly, clip to remove information, and change the projection transformation (such as from parallel projection to perspective projection). However, users often get lost in their navigation, for instance map users lose orientation information (Gahegan, 1999). Thus, techniques that control and constrain the user’s navigation are often useful (Buchholz, Bohnet and Dollner, 2005). Linked master–slave navigation can also allow the user to navigate efficiently in dual view systems (Plumlee and Ware, 2003), see Section 3.3.1. In two-dimensional maps the user can zoom or pan (Harrower and Sheesley, 2005) and scroll, and often the zooming operations are associated with a semantic change (Cecconi and Galanda, 2002). Often multiple views are linked together such that one is the focus and the other provides detail (Convertino et al., 2003).
3.4.5 Interacting with the environment

Users need effective ways to interact with the environment. Often, because of the small screen size, a user needs to move, iconize or delete unwanted windows. However, rather than permitting windows to overlap, some developers have constrained how the windows are displayed. For instance, GeoWizard (Feldt et al., 2005) collages the windows together and allows the user to enlarge or reduce the size of a window by changing the position of the dynamic splitters. In human–computer interaction researchers have proposed various other interfaces, from zoomable user interfaces (Bederson and Meyer, 1998) to elastic windows (Kandogan and Shneiderman, 1997), but these have not been integrated into many exploratory geovisualization systems.

Owing to the fact that exploration is often unguided, a user should be able to interact with the session history and roll back to a previous result. Few geovisualization tools include the rich functionality required to comprehensively manage the session history. Recent research has been achieved in this area, including: vistrails (Callahan et al., 2006) and provenance for visual exploration systems (Streefkerk and Groth, 2006). However, techniques that save the session history, allow users to extend different exploration paths and integrate explorations from different users are missing from most geovisualization systems.

Finally, in a large exploration session it is often difficult to remember which views are linked together and which parameters were used to create a particular view. Techniques to visualize this subsidiary information are named meta-visualization. Little work has been done in this area; however MacEachren et al. (2001) describe a watcher window that presents summary information of which windows are active, who is in control of which windows in a collaborative environment, and which windows are shared, while Weaver (2004) describes a technique that explicitly visualizes which views are linked together through arrows that are annotated on top of the visualizations.

3.4.6 Coordination

The area of coordination is rich and varied and is used in many disciplines (Olson et al., 2001). In fact, any forms of interaction can potentially be linked together: data processing, filtering, mapping, navigation and interacting with the environment (Boukhelifa, Roberts and Rodgers, 2003). The two most common are linked brushing (Buja et al., 1991) and navigational slaving (North and Schneideman, 2000). However, there are many aspects of coordination. For instance, the users’ needs may change as they continue on their exploration, thus they may require some views to be initially coordinated and then need to uncoordinate them to follow a new trail.

Roberts (2005) details seven rudiments of coordination based on program variables:

1. coordination entities describes what is coordinated, such as data, parameters, process, function or event;

2. entities are typed, such that elements of the same type can be connected – some form of translation may be required to connect entities with different types;
3.4 INTERACTION AND MANIPULATION

3. each entity has temporal and chronology information, such that some coordinations persist while others are temporary for a task;

4. each coordination is defined in a scope – for example it may be that the entities are coupled for a specific task and then uncoupled when the task has finished;

5. the coordination has a certain granularity – for example, some entities may be coordinated point-to-point while for others it may be possible to chain multiple entities together;

6. the coordination of entities may be initiated by the user or automatically by the system; and finally,

7. there are different methods to update the information when entities are coordinated – for instance, cold linking permits elements to be coupled once, warm linking permits users to decide when the views are updated, and hot linking is when the information is dynamically updated (Unwin, 2001).

There are three principle coordination architectures: (1) constraint-based programming (Mcdonald et al., 1990); (2) a data centric approach taken from the database community, where change to one component of a relational database is tightly coupled to other components (North and Schneiderman, 2000); and (3) the module view controller pattern, where one component observes the model and updates the view when changes are made (Pattison and Phillips, 2001; Boukhelifa et al., 2003). Tools such as Waltz (Roberts and Waltz, 1998) and Improvise (Weaver, 2004) implement many of the ideas of the rudiments. However, few CMV developers incorporate comprehensive coordination capabilities in their systems.

3.4.7 Challenges for interaction and manipulation

Most exploratory systems allow the visualization to be adapted and viewed again from a new parameterization, whether this information replaces the old or is displayed in a new window. However, there are still many challenges. First, not many systems allow the user to overlay the information. One overlay strategy is to generate the difference between the two parameterizations (Suvanaphen and Roberts, 2004). This can be extremely useful as the visualization would clearly show what has changed. However, few systems supply this functionality. Second, it is not clear as to how many views are actually useful. For instance, a user may easily get lost in a explosion of overlapping windows, hence there is certainly a trade-off to using replacement, replication and overlay. Consequently, more research is required to evaluate how many views are useful and to provide guidelines to when each strategy should be used. Third, with large explorations the user can easily forget which view represents what parameterization. Thus methods to support the user in their exploration are required.

Both interactive filtering and interactively adapting mapping parameters are important exploratory techniques and it is clear that much research has been achieved. However, many of these techniques are not implemented in modern exploratory tools. Developers seem to
rely on a small subset commands, such as picking and highlighting. Consequently, users are missing out on rich interaction techniques.

Navigation is still tricky. Users get lost in their exploration, they forget where they are and they may start to misinterpret information. Researchers need to create interfaces that clearly demonstrate where a user is, what orientation the data is and what scale it is displayed at. Furthermore, it is important to generate usable and intuitive exploration environments. If the user does not know how to control the environment, or if they get overwhelmed by too many windows, then they will not be effective in their exploration process. Thus, developers need to strike a balance between comprehensive function-rich environments on the one hand, and simple to use interfaces on the other. Developers should continue to research into methods that aid the user in their exploration and store session information.

Finally, coordination is obviously a major part in CMV systems. However, coordination tends to be achieved automatically and often performed silently by a system (such that users do not necessarily know what is connected together). Hence, meta-visualization techniques should be developed to enable the user to see how the system is setup.

3.5 Tools and toolkits

Geographical and spatial data has been long used to demonstrate research ideas in coordination. Felger and Schroeder (1992) in the visualization input pipeline (VIP) describe linked cursors of three-dimensional maps. LinkWinds (Jacobson, Berkin and Orton, 1994) demonstrates coordination using precipitation and ozone depletion; Visage (Roth et al., 1996) demonstrates coordinated manipulation of maps; the tight coupling interface of DEVise (Livny et al., 1997) has been used to demonstrate various examples including looking at product purchases by location; Spotfire (Ahlberg, 1996) uses map visualizations along with other statistical forms.

There are a few different ways to develop CMV systems. Tools such as CommonGIS (Andrienko and Andrienko, 1999), GeoVISTA studio (Takatuska and Gahegan, 2002) and Improvise (Weaver, 2004) contain comprehensive coordination capabilities. However, if developers wish to develop from scratch, then languages such as Java provide a convenient medium for development. For instance, tools like Mondrian (Theus, 2002) have been developed in Java.

In addition, toolkits such as the InfoVis toolkit (Fekete, 2004), Prefuse (http://prefuse.org/) and Piccolo (Bederson, Grosjean and Meyer, 2004) provide developers with the functionality to create complex visualization environments in Java. However, it still takes a lot of careful planning design and time to create effective geovisualization tools, and often developers wish to integrate algorithms and tools from different researchers.

One way to develop prototype tools is to use Flash or scalable vector graphics (SVG), such as those used by Steiner, MacEachren and Guo (2002) and Marsh, Dykes and Attikakou (2006), respectively. These solutions have the added advantage that they are instantly web accessible and available by many remote users. Alternatively, techniques such as the Information Visualization CyberInfrastructure (IVC) software framework, which extends original work on the Information Visualization Repository (Börner and Zhou, 2001), provide a uniform method to interact with a multitude of algorithms by providing a programming interface to algorithms and a user interface to end-users using the Eclipse Rich Client.
3.6 CONCLUSIONS

Many geovisualization tool developers seem to ignore or forget the richness of the published research and implement only basic coordination, interaction and navigation effects. Certainly developers and researchers have at their disposal a large heritage of geovisualization techniques. Researchers have proposed many geovisualization forms, a multitude of manipulation strategies and detailed aspects of coordination, brushing and navigation. However, concepts such as handling missing data, visualizing uncertain values, coordination rudiments and interactive manipulation techniques need to be included in the systems and included from the beginning of development. From experience, it is difficult to twist a system to perform techniques that were not originally planned.

Exploration itself is still difficult. It is difficult to know where to go, where users have been and how to control the interface effectively. One challenge is that either systems use a replacement strategy and the user fails to understand the history trail through the provided linear history tree or the system provides a replication strategy and the user gets lost in an explosion of windows. Thus, researchers should perform more research into how to manage the user’s exploration better.

Finally, the web certainly provides a convenient way to disseminate geovisualizations. Consequently, the utilization of Flash and SVG seems promising and should be encouraged. However, there is also a need to allow multiple users to collaborate in these visualization sessions. Although some work has been done in collaboration, it remains a niche technique and the functionality to appropriately manage and merge ideas from multiple participants is still naive.

3.5.1 Challenges for geovisualization tool developers

There are still many challenges faced by tool developers. First, it takes a lot of time and effort to develop and maintain a tool, but toolkits such as InfoVis toolkit (Fekete, 2004), Prefuse (http://prefuse.org/) and Piccolo (Bederson, Grosjean and Meyer, 2004) certainly aid the developer. Utilizing Flash or SVG to quickly develop and disseminate prototypes is certainly a promising strategy. These also permit the applications to be disseminated on the web, which allows many remote users to operate them and try out the techniques. However, it is still difficult to get the remote users to collaborate in an exploration. Current collaboration techniques are not well integrated with modern CMV systems, and techniques to support efficient and complex collaboration explorations are still naïve.

Finally, interoperability and extensibility still plague developers. Systems such as Börner and Zhou’s (2001) IVC certainly provide one method to integrate multiple algorithms and techniques. However, interoperability and extensibility concepts should be included in system development from the start of the project.

3.6 Conclusions

Platform (RCP). Finally, recently there has been interest in component and bean technologies. For example, ILOG’s Jviews (www.ilog.com/products/jviews/) uses Java, while GeoWizard (Feldt et al., 2005) uses Microsoft’s .NET to provide geographical visualization components.
References


REFERENCES


CH 3  COORDINATED MULTIPLE VIEWS FOR EXPLORATORY GEOVISUALIZATION


REFERENCES


CH 3 COORDINATED MULTIPLE VIEWS FOR EXPLORATORY GEOVISUALIZATION


REFERENCES


48

CH 3 COORDINATED MULTIPLE VIEWS FOR EXPLORATORY GEOVISUALIZATION

