Chapter 8

Exploratory Visualization with Multiple Linked Views

Jonathan C. Roberts, Computing Laboratory, University of Kent, Canterbury,
    Kent CT2 7NF, UK

Abstract

Exploratory visualization enables the user to test scenarios and investigate possibilities. Through an exploration, the user may change various parameter values of a visualization system that in turn alters the appearance of the visual result. For example, the changes made may update what information is being displayed, the quantity or resolution of the information, the type of the display (say) from scatter plot to line-graph. Furthermore, the user may generate additional windows that contain the visual result of the new parameters so they can compare different ideas side-by-side (these multiple views may persist such that the user can compare previous incarnations). Commonly these windows are linked together to allow further investigation and discovery, such as selection by brushing or combined navigation. There are many challenges, such as linking multiple views with different data, initializing the different views, indicating to the user how the different views are linked. This chapter provides a review of current multiple linked-view tools, methodologies and models, discusses related challenges and ideas, and provides some rudiments for coordination within a geovisualization context. The types and uses of coordination for exploratory visualization are varied and diverse, these ideas are underused in geovisualization and exploratory visualization in general. Thus, further research needs to occur to develop specific geovisualization reference models and extensible systems that incorporate the rich variety of possible coordination exploration ideas.

8.1   Introduction

This chapter advocates the use of many lightweight views that are linked together. They are lightweight in that they are: (i) easy to generate by the user, where the user does not spend unnecessary time and effort to explicitly link the new view to existing ones; and (ii) do not take many computer resources (e.g., memory, computation). Such multiple linked

45   views (MLVs) enable the user to quickly view a scenario, compare it with previous
46   realizations, examine properties such as dependencies and sizes, put this view to one side
47   and try out another scenario. There are many good principles that can be learned from
48   examining how other systems achieve this MLV exploration. In geovisualization, the
49   explorer often generates many spatial or abstract representations. With such exploratory
50   environments, the user is able (even encouraged) to take a hands-on approach to gain a
51   deeper understanding of the underlying information. They may examine multiple
52   different graphical realizations that reveal different aspects of the data. These principles
53   are applicable to the geovisualization domain (indeed, many MLVs use spatial
54   information databases to demonstrate the techniques). This chapter highlights current
55   trends in MLVs. In order to provide an overview of different multiple-view exploration
56   strategies, we start by placing the MLVs in context, then discuss exploration strategies
57   and expand upon appropriate methods to enable interactive and effective investigation
58   and management techniques that oversee and encourage the user to explore.

## 8.2   Current Themes in Exploratory (Multiple View) Visualization

62   When carrying out research, analysts often proceed by using an experimental cycle where
63   the experiment is set up perhaps with some default parameters, the results are noted
64   down, then the parameters are adapted and the results are compared with previous
65   versions. Each new investigation enhances the analyst's knowledge and understanding.
66   When starting the investigative process we may not know anything about the database
67   let alone what questions to ask. DiBiase (1990) focusing on the role of visualization in
68   support of earth science research, summarizes the research process as "a sequence of
69   4 stages: exploration of data to reveal pertinent questions, confirmation of apparent
70   relationships in the data in light of a formal hypothesis, synthesis or generalization of
71   findings, and presentation of the research at professional conferences and in scholarly
72   publications". Gahegan (Chapter 4) offers a perspective for "the entire process of
73   GIScience". The need for exploration techniques grows as the data become larger and
74   more complex. In such cases, the important aspects of the data are smaller, in comparison
75   with the whole, and specific details are more likely to be hidden in a swamp of elements.
76   Thus, in general, exploration techniques allow us to sift through volumes of data to find
77   relationships, investigate various quantities and understand dependencies.

78   One method to achieve this exploration, which has been the trend in the recent
79   years, is by "dynamic queries" (Shneiderman, 1994). These are highly interactive
80   systems that enable the visualizations to be manipulated, dissected and interrogated. The
81   user dynamically interacts with the visualization by adjusting sliders, buttons, and menu
82   items that filter and enhance the data and instantly update the display. By doing so the
83   "user formulates a problem concurrently with solving it" (Spence, 2001). For instance,
84   what was once a dark dense black region on a scatter plot can be immediately changed
85   into a colourful and meaningful realization (see Chapter 6). Systems that use this
86   technique include HomeFinder (Williamson and Shneiderman, 1992) and FilmFinder
87   (Ahlberg and Shneiderman, 1994) both now regarded as seminal work on dynamic
88   queries. Ahlberg and Wistrand (1995a,b) developed these techniques into the Information

89   Visualization and exploration environment system (IVEE). In one example, they depict
90   an environmental database of heavy metals in Sweden; IVEE was then developed into the
91   commercial Spotfire system (Ahlberg, 1996). Another early example is the "density dail"
92   (Ferreira and Wiggins, 1990), where visual results were chosen dependent on the dial
93   position. More recently, Steiner et al. (2001) provide an exploratory tool for the Web and
94   the Descartes system (Andrienko and Andrienko, 1999a–f) both provide dynamic
95   queries; these systems include map-based views linked to other views.
96   As an alternative to adapting sliders and buttons (as used in dynamic queries),
97   the user may directly manipulate the results; such direct manipulation may be
98   implemented using brushing techniques (Ward, 1994) or methods that select to highlight
99   or filter the information directly. Much of the original work was done on scatter plot
100   matrices (Becker and Cleveland, 1987; Carr et al., 1987). Brushing is used in many
101   multiple-view systems from multi-variate matrix plots, coplot matrices (Brunsdon, 2001)
102   to other geographic exploratory analysis (Monmonier, 1989). One map based
103   visualization toolkit that utilizes multiple views and brushing is cdv (Dykes, 1997a,b).
104   cdv displays the data by methods including choropleth maps, point symbol maps, scatter
105   plot and histogram plots. Statistical and geographic views are linked together, allowing
106   elements to be selected and simultaneously highlighted in each. MANET (Unwin et al.,
107   1996), developed from the earlier tools SPIDER and REGARD, provides direct
108   manipulation facilities such as drag-and-drop and selection and control of elements in the
109   display, for example.
110   Moreover, other direct manipulation techniques allow the inclusion of
111   manipulators and widgets; for example the SDM system (Chuah and Roth, 1995)
112   provides the user with handles mounted on visual objects to control the parameters
113   directly. Often the widgets are applied to the objects when they are needed and provide
114   additional functionality. The widgets may be multi-functional, where different
115   adornments provide specialized manipulation. Figure 8.1 shows a jack manipulator
116   where the outer cubes allow rotation; both the horizontal plane and vertical tubes allow
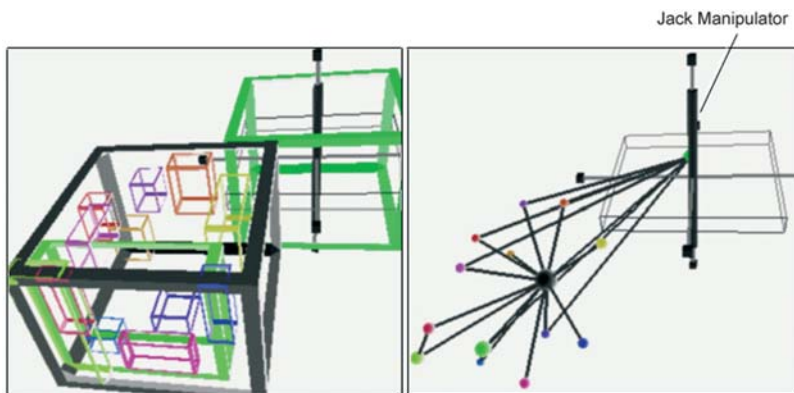
Figure 8.1. Diagram taken from the Waltz visualization system (Roberts, 1998a,b), showing the use of the Inventor Jack manipulators.

133 constrained planar translation. This manipulator is provided by Open Inventor libraries
134 and integrated in the Waltz multiple-view visualization system (Roberts, 1998a,b). In the
135 figure, the manipulator has been attached to an object that has been moved along the XZ
136 plane (using the large horizontal rectangle). Other manipulators exist; for example,
137 selection in Mondrian (Theus, 2002a,b) may be operated through the use of rectangle
138 areas. In this tool, the user may modify the regions by selecting handles on the rectangles,
139 multiple selection areas can be used at once, and the selected items are highlighted in
140 related windows.

8.3   Strategies of Exploration

In any interactive visualization, the decision needs to be made as to where the
information goes, that is, when the parameters are changed does the new visualization
replace the old, get overlaid, or is it displayed alongside and in separate windows?
Roberts et al. (2000) names these strategies replacement, overlay and replication,
respectively. This is depicted in Figure 8.2. This fits in well with the design guidelines of
Baldonado et al. (2000), who describe the rule of "space/time resource optimization",
where the designer must make a decision whether to present the multiple views side-by-
side or sequentially.

8.3.1   Replacement

The replacement strategy is the most common and has some key advantages, that is,
the user knows implicitly where the information is updated and what information has
changed. However, there are some major challenges with this strategy. First, there are
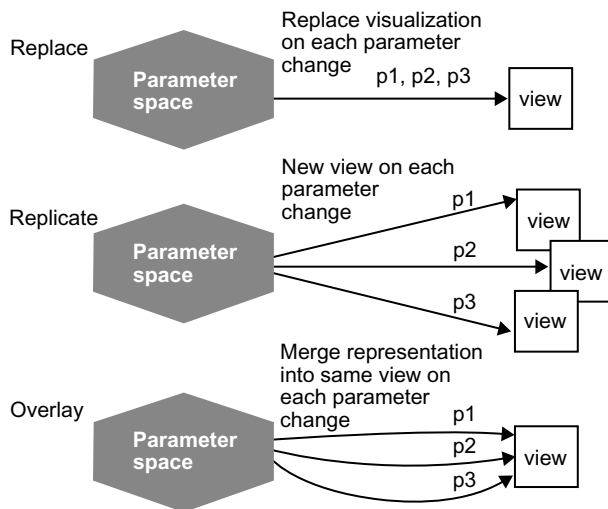

Figure 8.2. There are three strategies of exploratory visualization that determine where the information is placed: replacement, replication and overlay.

problems by using such an ephemeral exploration environment. Information about previous experimentations is usually lost, the user cannot compare different graphical realizations side-by-side, and there is often little guidance as to the sensitivity of different parameters (i.e., whether a small change of a parameter will make a small change in the image, or in fact it makes a large amendment to the visualization). Second, there is a risk of losing navigation context. For example, when a user zooms into a subpart of the display the context of how the zoomed area fits in with the whole is lost.

Some visualization systems overcome the transient nature of the display by storing past visualization commands (as data or variable values) in a database, such as Grasparc (Brodlie et al., 1993) and Tioga (Stonebraker et al., 1993). In the case of Grasparc, or HyperScribe (Wright, 1996) as implemented as a module in IRIS Explorer, the user can "roll back" to a predefined state and re-visualize the data with the "old" parameters. As in the case of HyperScribe these states are usually stored in a "history tree" where data arising from the experiment process is modelled in a tree structure and the user can alter parameters and roll back to previous versions (Figure 8.3).

As the user explores, it can become unclear how the filtered, extracted and specialized information fits in with the whole. Methods such as animation and distortion help to keep this context. For example, animation is used in ConeTree (Card, 1996); in this instance, a selected node is brought to the foreground by animating the 3D tree (for a explanation and figure, see Schroeder, this volume, Chapter 24). The animation occurs long enough for the observer to see a continuation and short enough so that the user still observes the visual momentum. Moreover, there is a current trend towards generating detail-in-context views also known as Context + Focus displays (Lamping et al., 1995). Many implementations are non-linear magnification systems using methods such as those described by Keahey and Robertson (1997). They appear with a linear (and traditional) mapping in the centre or focus of the screen and squashed or distorted mapping outside the focus area. For example, Snyder (1987) generated various magnifying glass projections of the earth. Other people who use distortion to provide a clear field-of-view to an interesting object in three dimensions include Sheelagh (1997).
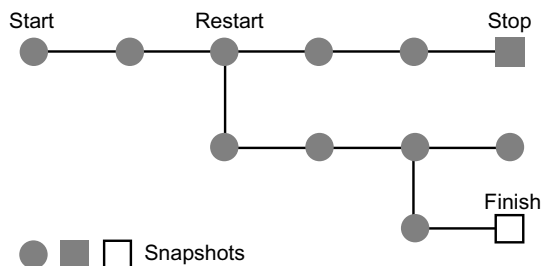
Figure 8.3. Diagram showing the history tree where data arising from the experiment process are modelled in a tree structure and the user can alter parameters and roll back to previous versions.

8.3.2   Replication

Another way of working is to use a replication strategy for information exploration. In this strategy, various parts of the information, parameters or views are copied or duplicated and aspects are displayed in multiple ways and in different windows. Replication refers to the action of the experimenter who wishes to repeat an experiment or procedure more than once. Replication may be used to provide methodical or random repetition of the experiment to confirm or reduce the error of the results (by perhaps averaging the different findings) or to confirm the outcomes. Far too often a user relies upon one display, presenting data by their "favourite" visualization algorithm. However, they may be missing out on the richness of the underlying information. Hence by duplicating and replicating the displays and slightly adapting the parameters for the next incarnation the user is able to observe and compare the result of different scenarios and experiment with the detail of their data.

    Replication can be divided into two subcategories of usage: (i) the procedure – where the results that are generated by the change of parameters are displayed in separate windows; (ii) the course of action – where the same data may be presented by different mappings. These different forms of the same information are known as multiforms (Roberts et al., 2000).

    It is useful to display the results of a parameter change in a new window: the user can clearly observe and compare side-by-side the differences and similarities of the results. For example, the user may wish to explore different isosurfaces depicting alternative concentrations of some phenomena. If, as the user changes the threshold value a new window appears displaying the new isosurface then the user can easily observe (and compare) the varying concentrations from the current and previous explorations. As we shall see in Section 8.4.4, such a dynamic replication could provide a multitude of views. Such a view-explosion could confuse, rather than support the user in their exploration tasks.

    Not only can different parameterizations be displayed in multiple windows, but also the same information may be displayed in multiple forms. By doing so the user may be able to see information that was previously obscured, or the different form may abstract the information to provide a clearer and simpler representation, or the different views may represent alternative interpretations on the same information (such as those given by different experts). Indeed, the alternative view may help to illuminate the first. Yagel et al. (1995) advocate the use of "…visualization environments that provide the scientist with a toolbox of renderers, each capable of rendering the same dataset by employing different rendering schemes". Consequently, the user may gain a deeper understanding of their data. For example, our eyes use binocular vision to present two slightly different observations of the same scene, which provides us with a rich depiction of the information. Certainly, we miss out when we look at one picture of something, such as a still photograph of an historic building, and we gain a better understanding of the size, colours, textures and details when we browse through many photographic pictures, fly through a virtual 3D model and view it from multiple viewpoints, and read written explanations from an interactive guidebook. Likewise, it is often beneficial to the data explorer to see the information from different perspectives and in different forms.

265 There are many advantages in using replication, for example, the separate views
266 hold a history of the exploration, allow comparisons between images, and the multiforms
267 may emphasize different aspects of the information. Replication should be encouraged
268 (Roberts, 1998a,b). However, not many current systems inherently support many views
269 and the module visualization environments, which can display the data in many
270 representations, leave all the effort of duplication to the user. Indeed, such a replication
271 strategy is possible in the module building visualization environments, such as AVS,
272 IRIS Explorer and IBM Data Explorer (Williams et al., 1992). However, exploring the
273 information in such a way with these tools requires copying and reconnection of multiple
274 modules, and thus the replication strategy is not necessarily encouraged or easy to operate
275 in these module-building environments. It is not a lightweight operation. The system
276 itself should have the functionality to support multiple views, created with little effort
277 from the user, managed appropriately by the system and automatically coupled to other
278 views. Moreover, further understanding may be gained through linking and coupling of
279 information. For example, selections that are made in one view can be reflected in other
280 views, other operations such as zooming and rotation operations can be cordially applied
281 to any associated view – hence the phrase MLVs.
282

283 ### 8.3.3 Overlay

284
285 A third method of generating the visualization result is to overlay the visualization
286 method in the same display. Overlays allow different visualizations to share the same
287 coordinate space. Such a fan-in method allows different representations of the same
288 information in the same display to be layered together. The advantage of this is that it is
289 easy to understand each view in the context of the other, and the information may be
290 readily compared. Different representation methods may be mixed together in the same
291 view. For example, one view may include 2D pseudo-colour slices, surface
292 representations, legends and useful annotations. However, when too much information
293 is presented in the one view, or layered over a previous version, it may be difficult to
294 select and navigate through or understand specific information. This may be because the
295 presentation is too crowded and complex or that parts of the visualization are occluded.
296 Indeed occlusion may be a problem in 2D visualizations as the objects may lay
297 directly over each other. This may cause a misunderstanding of how many elements are in
298 fact at a particular coordinate. Solutions such as the use of transparency or randomly
299 jittering the points may help to clarify the depictions. Additionally, aggregation followed
300 by different mapping techniques may be useful, as demonstrated by the sunflower plot of
301 Dupont and Plummmer (2003). Obviously, the usefulness and appropriateness of the
302 overlay method depends on the graphical visualization technique and the visualization
303 tasks being used.
304 Related work includes the excellent Toolglass and Magic lenses (Bier et al., 1993)
305 widgets that allow the user to see through and focus on details of the display. Geospace
306 (Lokuge and Ishizaki, 1995) usefully employs translucency between the layers, and Kosara
307 et al. (2002) uses a semantic depth of field to blur layers to keep the context. Döllner (Chapter
308 16) uses texture-mapping methods to implement a lens effect that draws upon transparent

309 layers. Moreover, Gahegan (1998) provides an example of an integrated display to achieve
310 more complete integration of geovisualization views. Ongoing work on MANET (Unwin
311 et al., 1996) is focussed on methods to overlay different plots on the same view.
312   The challenge here is to develop effective overlays that enable the user to keep
313 the context information, understand the depth of knowledge and not become
314 overwhelmed by a complex visual representation. Specific challenges include how to
315 effectively operate the overlaid views – does the interaction go through a view or is only
316 the top view active? How is the user made aware that the views may differ in their data?
317 How are the data linked to the data, and can it be coupled?

8.4   Multiple Linked Views

321 Linking and relating the information in one view to that of other views assists the user in the
322 exploration process and may provide additional insight into the underlying information.
323 Certainly, "multiple views should be coordinated" (Carr, 1999). As the information is
324 explored and placed in separate windows, it is important that the relationships between the
325 views and the context of how one view relates to another are maintained. Indeed,
326 Shneiderman and North (2000) in their user experiments discover that MLVs are beneficial
327 and state that "the overview and detail-view coordination improved user performance by
328 30–80% depending on task". Such additional "overview" realizations provide context
329 information that enhances the understanding of the associated view.
330   Many different forms of information may be linked and coordinated. For
331 instance, manipulation operations (such as rotation, translation, zoom, etc.) may be
332 concurrently applied to separate views so as when one view is manipulated the other
333 views respond appropriately to the same manipulation operations; the spatial position of a
334 pointer or probe may be linked between multiple views; filter, query and selection
335 operations may be simultaneously applied. Moreover, these operations need only affect
336 the same information but, more interestingly, to collections of different information.
337 Coordination and abstract views provide a powerful exploratory visualization tool
338 (Roberts, 1998a), for example, in a 3D visualization, a navigation or selection operation
339 may be inhibited by occlusion, but the operation may be easier using an abstract view.
340 Fuhrmann and MacEachren (1999) describe the use of an abstract view to guide
341 navigation in a 3D geospatial representation, ideas that are further developed by
342 Fuhrmann and MacEachren (2001). Thus, a linked abstract view may be used to better
343 control and investigate the information in the coupled view.
344   Accordingly, there are different reasons for coordination. North and
345 Shneiderman (1997) state there are two different reasons for using coupled views, either
346 for selection or for navigation. Although Pattison and Phillips (2001) disagree by saying
347 that there are additional forms of coordination other than selection and navigation, for
348 example, "coordinating the data in preparation for the visualization such as sorting,
349 averaging or clustering". Likewise, Roberts (1999) believes in a broader use of
350 coordination, exemplified by the layered model (Roberts, 1999; Boukhelifa et al., 2003)
351 where the user may link any aspect of the dataflow and exploration process.

353   Selection allows the user to highlight one or many items either as a choice of
354 items for a filtering operation or as an exploration in its own right; this is often done by
355 direct manipulation where the user directly draws or wands the mouse over the
356 visualization itself (Cleveland and McGill, 1988; Ward, 1994). Becker and Cleveland
357 (1987) describe this as a brushing operation. Examples, of systems that implement the
358 brushing technique include XmdvTool (Ward, 1994), IVEE (Ahlberg and Wistrand,
359 1995a,b) and Spotfire tools (Ahlberg, 1996).
360   Joint navigation provides methods to quickly view related information in multiple
361 different windows, thus providing rapid exploration by saving the user from performing the
362 same or similar operations multiple times. Objects, such as pointers, annotations or meta-
363 information, may be coupled. For instance, the developers of the visualization input
364 pipeline (VIP) (Felger and Schröder, 1992) describe an example that displays several views
365 of the data with the cursors linked together; movement of one pointer causes the others to
366 move correspondingly. Other forms of navigation include data probing, as implemented
367 within both LinkWinds (Jacobson et al., 1994) and KBVision (Amerinex, 1992), and
368 changing the viewport information, as accomplished in SciAn (Pepke and Lyons, 1993)
369 and Visage (Roth et al., 1996), which provide coordinated manipulation of 3D views.

370
371 ## 8.4.1   Linking architectures

372 The study of coordination is interdisciplinary and there is much to learn from other
373 disciplines. Taking the simplistic view of coordination being "sharing things" then we
374 may learn from areas such as sharing hardware devices in a computer system or
375 managing, delegating roles in a human organization or collaborative support, for
376 example, see Brodlie et al. (this volume, Chapter 21). For an in depth interdisciplinary
377 view of coordination, see Olson et al. (2001).
378   In this particular chapter, we focus on four models: Snap (North, 2002),
379 presentation graphics (McDonald et al., 1990) and the View Coordination Architecture
380 (Pattison and Phillips, 2001) and a Layered Model for Coordination (Boukhelifa et al.,
381 2003). Andrieko et al. (this volume, Chapter 5) provide an in depth discussion of software
382 issues in geovisualization.
383   The Snap conceptual model (North, 2002) takes a data-centric approach to
384 coordination. It uses concepts from database design to provide the required interaction.
385 Relational database components are tightly coupled such that an interaction with one
386 component results in changes to other components. The Snap architecture is designed to
387 construct arbitrary coordinations without the need for programming. However, Snap's
388 user interactions are currently limited to "select" and "load", whereas exploratory
389 visualization permits rich and varied interactions such as representation-oriented
390 coordinations in addition to data-centric coordinations.
391   McDonald et al. (1990) describe a constraint system based on the presentation-
392 graphics programming model (Figure 8.4). In this system, lenses map the subjects
393 (objects) in the database into their visual presentations counterparts, a user interacts with
394 the presentation and the subjects get updated through the input-translator, and finally,
395 a constraint system updates corresponding properties and updates any other related
396 graphical presentations.
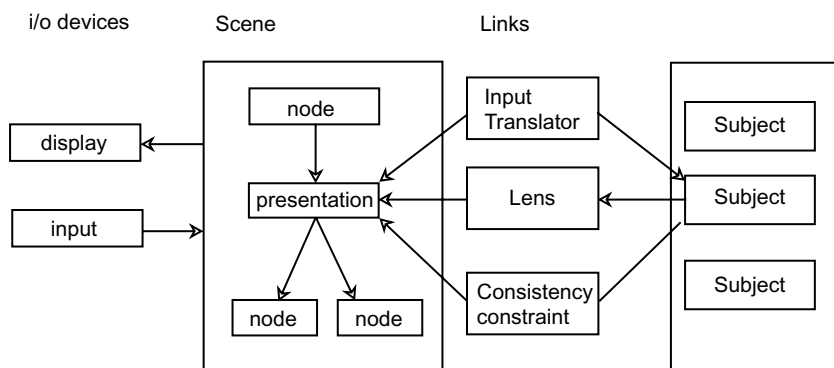
Figure 8.4. Presentation graphics programming model (McDonald et al., 1990).

Pattison and Phillips (2001) developed an architecture based on the model view controller (MVC) design pattern that originated in the Smalltalk architecture (Figure 8.5a). This pattern describes three objects: the model, view and controller, where the model holds the state of the process and publishes notifications to the views when its state changes, the view(s) reflect the state of the data model, and the controller updates the model with requests from external events. The MVC architecture inherently supports multiple views, and Pattison and Phillips (2001) have adapted the model for Information Visualization (Figure 8.5b). Where the presentation component observes the model for changes and updates its display as necessary, the model component observes both the specification and data model components for change modifications to the specification component are propagated up. This architecture fits in with the dataflow paradigm (Haber and McNabb, 1990).

Rather than concentrating on the implementation architecture, our work has focussed on a layered approach that is based on the dataflow model (Roberts, 1999; Boukhelifa et al., 2003) and incorporates more layers than that of Pattison and Phillips (2001). In this approach, the coordination may occur between any parameter at any level of the visualization flow (Figure 8.6). Therefore, the user can link a broad range or aspects between several windows, for instance, the view projection transformations can be shared (to co-rotate several 3D objects included in separate windows) or characteristics of the objects can be simultaneously changed (such as their appearance, colour, texture or position, etc.), or window-operations can be coordinated (such as moving, deleting or iconizing windows).

8.4.2   The role of MLVs in the exploration process

The exploration process may be described as a history-tree, indeed, even if the views are a result of a set of random thoughts, each view still relates in some way (however tenuous) to former investigations. Often the newest explorations are close to the former; this is the case especially if the user makes minor amendments to a copy of the previous view. Consequently, it is sensible to consider clusters or groups of closely related views. This can occur as "render groups" (Yagel et al., 1995) where different renderers are used to

441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
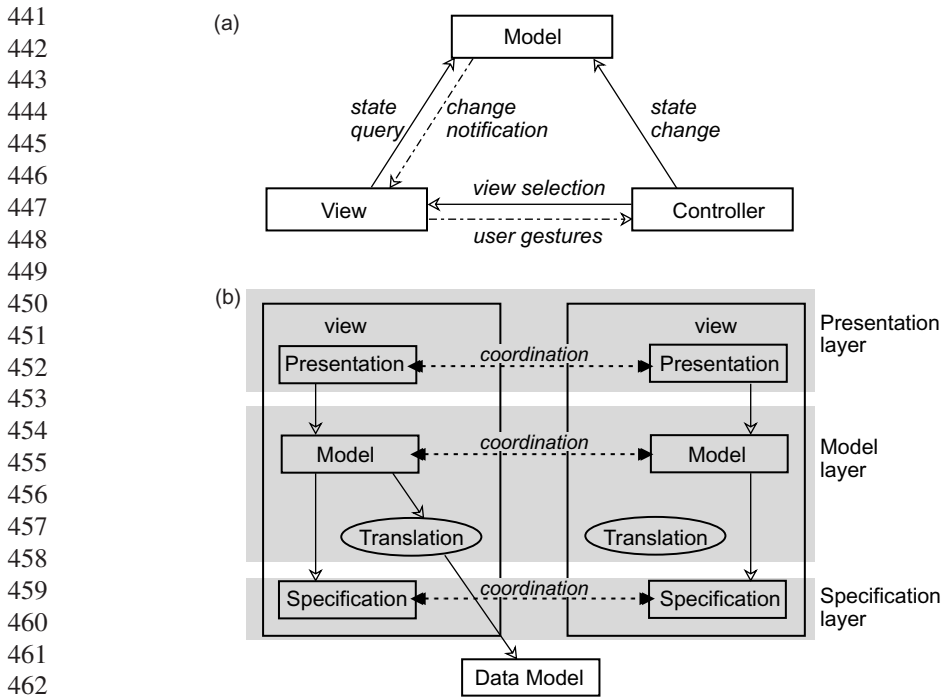
Figure 8.5. (a) Left, depicts the traditional MVC pattern. The views reflect the current state of the model; the information held in the model is updated via the controller. (b) Right, shows the coordination model by Pattison and Phillips (2001) based on an MVC pattern, where the presentation component observes the model for changes and updates its display as necessary, the model component observes both the specification and data model components for change and changes to the specification component are propagated up.

display the same data filtering (at an equivalent the level to the "Data Model" in Figure 8.5). Information within each render group may be straightforwardly related to each other such that default coordinations may be readily defined (Roberts, 1999).

Generating multiple views from any part of exploration process may be useful; here the user keeps older versions of their investigations such that they can compare previous incarnations. They provide a context of the whole exploration process. However, linking outside render groups is challenging as some operations may not be generally applicable such as highlighting elements between two disparate data models when each contains a set of disparate non-intersecting elements. It is both possible and often beneficial to coordinate outside the render groups, for instance, multiple 3D worlds may be simultaneously rotated even if they contain dissimilar realizations. There is an advantage in grouping the multiple views together as Kandogan and Shneiderman (1997) discover through their evaluations: the user better understands the relationships in the views, and can more easily find and drill down to the important aspects of the display.
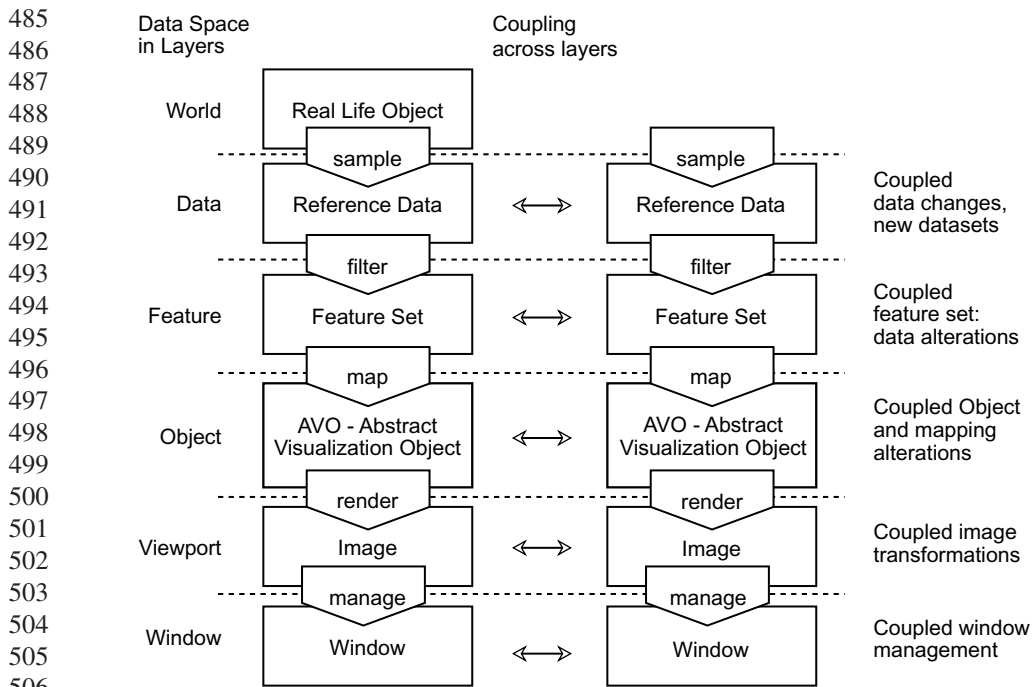
Figure 8.6. The diagram shows a layer model, where many different forms of information may be linked and coordinated. For instance, manipulation operations (such as rotation, translation, zoom, etc.) may be concurrently applied to multiple views so as one view is manipulated the other views respond appropriately to the same manipulation operations, the spatial position of a pointer or probe may be linked between multiple views.

8.5   Linking and Coordination Concepts

All the aforementioned ideas allow many windows to be created and linked with other views, but, rather than arbitrarily creating and linking views there is usually structure in an investigation. Certainly, when developing a coupled visualization system there are many questions to consider about the coupling. What is being coupled? What are their types? What gets changed? How does the information change? It may be that some links do not make sense and in fact may confuse the user, especially in visualization applied to exploration. Therefore, there are many challenges and much research still to be done. We distil these ideas into some rudiments of coordination.

8.5.1   The rudiments of coordination

In essence, the linking of information between views may be described as "information sharing" For example, if two objects in separate windows were projected using the same shared transformation matrix then any change to that matrix would update both views

simultaneously. Accordingly, coordination may be thought of as in terms of program variables. Thus, using this analogy the links have the following elements:

- *Coordination* entities details what is being coordinated. For example, it could be aspects of the data, record, parameters, process, event, function, aspects of the window or even time.
- The *type* expresses the method by which the views are linked. Coordinating parameter values such as coupling binary threshold operations or selecting ranges may be implemented by sharing primitive types (float, integer, etc.) while other operations may use more complex data structures. Some form of translation (or casting) may be required to coordinate entities with different types. In addition to this translation function, it is often useful to allow more intricate functions, such as to allow entities to be related via an offset (or by some other relation). In virtual reality it may be useful to provide two 3D views with one being at ground level and the other tethered above; the tethered view could provide an overview and thus move correspondingly with the ground view, for example, see Döllner (this volume, Chapter 16). The types may also determine the directionality of the links whether unidirectional or bidirectional.
- *Chronology* details temporal aspects such as the persistence or lifetime of the coupling, that is, how long the coupling exists? For example, it may be that objects in the scene are coupled for a specific task and then uncoupled when the task is over. Incidentally, like program variables, persistence and scope are inherently related. Moreover, the coordination may be synchronous, asynchronous, reactive, and proactive. For example, it may be useful to join the rotation of two views, one from a fast and the other a slow renderer, such that the slower render gets updated at a lesser rate; additionally, the user may make and review a change, then decide whether to commit or cancel this operation. McDonald et al. (1990) describes these capabilities as markup and commit/cancel.
- *Scope* controls the "area" of the correlation, whether two specific views, many realizations, or all realizations are coupled within an exploration. For example, the render group scenario is equivalent to a local variable and the global variable would be equivalent to coupling every view in the exploratory session.
- *Granularity* expresses how many entities may be connected together. For example, how many entities are coordinated, how many views are connected in one coordination operation.
- *Initialisation* indicates who creates a correlation, whether the user or the system. For example, in spreadsheet system it is possible to name particular views for specific operations, or by using a render group method it is possible to automatically correlate aspects of the views. There is a similar issue regarding the creation of the views themselves. Some visualization systems automatically create the visualizations from a database of knowledge (metadata information) and user requirements. The Vista tool (Senay and Ignatius, 1994), for example, creates appropriate visualizations by asking the user to list the variables in order of preference.

573   •   *Updating* describes how and when the information within the views and child
574   modules are updated and refreshed, such as lazy update, or greedy update or user
575   initiated. This is similar to the cold/warm/hot-linking concepts mentioned by
576   Unwin (2001). Cold linking allows an adjacent view to be coupled once and
577   ignores any changes to the former view (similar to copying values rather than
578   copying a formulae in a spreadsheet), warm linking allows the user to decide
579   when to update, hot linking provides automatic and dynamic updating of the
580   linked views. Moreover, the interface should reflect the current state, for
581   example by shading out the out-of-date views. However, it may be that views
582   depend on other views and if the user is relying on the data-history it may be
583   prudent to allow the user to force the update when required.
584
585   Currently, some general-purpose visualization systems do provide some of
586 these rudiments, for instance, IRIS Explorer allows parameters to be coordinated
587 through unidirectional events and more intricate functions may be formed using the
588 p-func editor; however, IRIS Explorer does not provide bidirectional links and
589 disallows simultaneously connecting the reverse linkage to inhibit circular event
590 explosions taking place. In geovisualization, a good example of linking is that of the
591 bi-directional link between ArcView and xGobi (Symanzik et al., 2000).
592 Coordination is used in other geovisualization systems; the GeoVISTA studio for
593 example (MacEachren et al., 2001) incorporate some coordination features. Many
594 systems provide an overview map to manage the manipulation of the whole (Steiner
595 et al., 2001; Andrienko and Andrienko, 1999a–f). Additionally panoraMap (Dykes,
596 2000) allows panoramic photographs (georeferenced with GPS positions) and other
597 information to be dynamically linked with an interactive map, other information such
598 as key-points visited and qualitative and quantitative information collected on site are
599 also shown by icons and symbols on the map.
600   It is clear that there are many issues still unanswered regarding each of these
601 rudiments, for example, are there specific rudiments for geovisualization? Or in general:
602 does it make sense to coordinate different types together? And if so: what translators are
603 required? How does the user recognize the scope of the coordination or indeed
604 understand the persistence or recognize whether something is out-of-date? Moreover,
605 many systems do not provide the full rich set of linking strategies that are possible.
606
607 ## 8.6   Management of Views and Linkages
608
609 In addition to the linking concepts there are some subsidiary issues to consider, such as
610 managing the views and linkages, placement of the views and temporal aspects.
611
612 ### 8.6.1   Managing the MLVs
613 The essence of lightweight MLVs is that they are easy and quick to generate, but by
614 supporting such a strategy the user may generate many views (that will create a view
615 explosion) where many of the representations are only slightly different to the previous
616 This creates two main problems. First, these many representations may easily clutter

the screen-space (there is a limited "real-estate" in any screen technology), and thus their needs to be either some form of restraint to guard the user from generating too many windows or management strategies to appropriately and automatically place each window (the latter is detailed in §8.4.4). Second, the user may also be confused as to "which image relates to which data-instance". The systems in the literature provide different solutions.

One solution is to inhibit the number of views: Baldonado (2000) provides a useful set of guidelines for using multiple views, and include the rule of parsimony – use multiple views sparingly. Another solution is to trade space by time. Spence (2001) discusses this solution and provides the idea of rapid serial visual presentation (RSVP); this allows the user to rifle through a set of objects analogous to flicking through the pages of a book in order to acquire some understanding of its content. This space/time trade off may be described as an overlay methodology. Finally, a good policy would be to use the three strategies (replacement, overlay and replication) together, allowing the user to replace certain instances and replicate when they need to achieve side-by-side comparisons.

It is important that the user should clearly understand the relationship of how each view relates to each data model. Many systems display the history tree (on a work-pane or canvas) allowing the user to rollback to previous versions (Brodlie et al., 1993; Wexelblat and Maes, 1999). Then the problem becomes how to relate the views with the canvas. This can be achieved using various methods. In the Waltz system (Roberts, 1998a,b), each window is labelled, relating it to its respective module on the work-pane. This is a hierarchical numbering scheme, like the sections of a book, and is used to name each view. The names are then displayed on the history tree. The spiral calendar (Mackinlay et al., 1994) provides a graphical solution by using lines to relate one window to another.

There is still much work to be done in developing effective view management strategies for MLVs; whether managing the placement of the views, controlling a possible view explosion, or relating the view information to that of the exploration hierarchy.

8.6.2   View placement strategies

The placement of the many windows can have a significant impact on the usability of the system: it is an important human computer interaction issue. Overlapping windows can cause the user to spend more time arranging the windows rather than doing the task (Kandogan and Shneiderman, 1997), whereas the screen may not be large enough to display each required view simultaneously. There are different placement strategies described as follows.

First, the user is given the responsibility to position, iconize and scale the windows. As it is often difficult to select and find occluded windows, the system provides a repository or toolbar to hold a list of the displayed windows. This may take the form of a list of the named views, collection of icons, or thumbnail representation of the current views.

Second, the system holds the responsibility for placing the views on the screen. These "intelligent" interfaces tile (or tabulate) the windows such that they appear

661 adjacently without overlap. Elastic views (Kandogan and Shneiderman, 1997) provide a
662 good example; in this methodology, the windows are hierarchically placed on the screen
663 and dynamically scaled to fill the available space. Alternatively, spreadsheet styles are
664 becoming popular (Chi et al., 1998; Jankun-Kelly and Ma, 2001) where the views are
665 positioned in a tabular formation. Furthermore, the strategy may depend on some aspect
666 of the data exploration or some other metric. For example, windows could be scaled
667 smaller if less important, implemented by a zoomable interface such as Pad++ (Perlin
668 and Fox, 1993), or presented in a scatter plot form where the placement of each is
669 dependent on two variables, or hierarchically as in the Flip zoom technique (Holmquist
670 and Ahlberg, 1997).
671   Many of the current multiple view visualization systems hand the responsibility
672 to the user, however, there is much benefit in structuring the position of the views relative
673 to each other. Thus, strategies for positioning the views appropriately should be
674 researched. Many questions remain including: are the requirements of an MLV
675 visualization system very much different to that of a traditional windowing system?

8.6.3   Chronology, animation and timing in MLV

678 Many datasets are time dependent; their visualization in an MLV environment may be
679 treated in different ways. The simple case is to generate an animation of the data. In the
680 above terminology, each frame would replace the previous. Alternatively, each
681 individual frame (or a sample of frames) may be displayed in a separate view (or
682 stacked and overlaid in a single view). Coupling multiple-view animations would involve
683 synchronizing the two streams. This may be at a fine granularity (e.g., tightly
684 synchronizing each individual frame) or coarse granularity (e.g., synchronizing on
685 specified key-frames).
686   Additionally, it may be that there are objects animated or moving in the scene
687 (such as people, planes or boats). It may be useful to couple one view to the moving
688 object and provide another view of the whole environment. The linked view may be
689 tethered such that it looks down on the object being moved (separated by an appropriate
690 distance). For example, the GeoZui3D of Plumlee and Ware (2003) provide different
691 "frame of reference coupling" methods that describe how the new view moves in relation
692 to the animated objects.

8.7   Current Objectives and Challenges

696 Recent research has focussed on providing principles for multiple views (Baldonado
697 et al., 2000) and examining linking methods such as Roberts' taxonomy of coordination
698 (Roberts, 1999; Boukhelifa et al., 2003) and North's Snap-together system (North and
699 Shneiderman, 2000a) that allows unforeseen combinations of coordinated visualizations.
700 This research is opening the way for more expressive investigation environments that
701 support the user in their task rather than distracting the user from their task.
702   Currently many multi-view systems only really support a few views where the
703 system determines what and how the information is linked. Thus, further research should
704 focus on developing systems that utilize many lightweight views that are truly quick to

generate and automatically linked with other information and implicit to operate. Indeed, the system could be designed that would suggest or automatically generate other views that the user had not thought of using. The user may find these non-traditional views unfamiliar, but this unfamiliarity itself may provide a better understanding.

There are many issues surrounding MLVs that are lightweight (some have been highlighted in this chapter). To develop an appropriate MLV system that utilizes these aforementioned concepts, it may be that the system needs to automatically generate the visualizations on behalf of the user, such as in the Vista system (Senay and Ignatius, 1994) or at least make it as easy as possible to generate further representations (Roberts, 1998a,b). Furthermore, if the system provides a diverse and functional-rich interface then the user may be overwhelmed by the nature of the system. Overall, a balance needs to be found both to generate the right amount of views for the task (whether they are by replacement, replication or overlay), and to provide an expressive linking mechanism that also restrains the user from performing incomprehensible and unprofitable coupling operations.

In addition, more empirical research needs to take place on the different designs to evaluate what is useful. Kandogan and Shneiderman (1997) have evaluated the effectiveness of certain multiple view systems and North and Shneiderman (2000b) have looked at coordinated views. However, more studies are needed. It is well understood that the effectiveness of a particular system or design is highly dependent on the visualization or investigative task and the domain; to this end Baldonado et al. (2000) offers some guidelines, but it still remains unclear when the user should replace, replicate or overlay the information to gain the best understanding.

The geovisualization domain poses many challenges (MacEachren and Kraak, 1992). Indeed, highly interactive systems have already been developed such as Descartes (Andrienko and Andrienko, 1999a–f), GeoVIBE (Guoray Cai, 2001) and cdv (Dykes, 1997a, b). However, further research is required to put in place the tools and techniques that will allow appropriate multiple-view exploratory geovisualization systems to be easily developed.

We propose the following strands of research:

(1) Specific geovisualization reference models and toolkits need to be developed that incorporate lightweight MLVs and include the rudiments of coordination.
(2) The tools need to support dynamic queries and complex coordination operations enabling highly interactive context + focus navigation.
(3) The developed systems need to be easily extensible that will allow the data from the ever increasing and diverse range of data to be suitably visualized.
(4) Methods need to be developed that integrate a wide range of different presentation methods, thus, allowing the user to view the information from different perspectives and try out different scenarios.

Acknowledgements

References

Ahlberg, C., (1996) "Spotfire: an information exploration environment", *SIGMOD Record*, 24(4), 25-29.

Ahlberg, C., and Shneiderman, B., (1994) "Visual information seeking: tight coupling of dynamic query filters with starfield displays", In: Adelson, B., Dumais, S., and Olson, J. S., (eds.), *CHI'94 Human Factors in Computing Systems,* Boston, MA, 24–28 April, ACM, pp. 313-317.

Ahlberg, C., and Wistrand, E., (1995a) "IVEE: an environment for automatic creation of dynamic queries applications", *Human Factors in Computing Systems. Proceedings of CHI'95*, ACM.

Ahlberg, C., and Wistrand, E., (1995b) "IVEE: an information visualization and exploration environment", *Proceedings International Symposium on Information Visualization,* Atlanta, GA, pp. 66-73.

A.I. Amerinex Inc., (1992) General Support Tools for Image Understanding. Online: http://www.aai.com/AAI/KBV/KBV.html (23/10/03).

Andrienko, and Andrienko, (1999a) Seasonal Migration of White Storks, 1998–1999. Online: http://www.ais.fhg.de/and/java/birds/ (23/10/03).

Andrienko, G., and Andrienko, N., (1999b) "GIS visualization support to the C4.5 classification algorithm of KDD", *Proceedings of the 19th International Cartographic Conference*, pp. 747-755.

Andrienko, G., and Andrienko, N., (1999c) "Knowledge-based visualization to support spatial data mining", *Proceedings of Intelligent Data Analysis*. Berlin: Springer, pp. 149-160.

Andrienko, G., and Andrienko, N., (1999d) "Making a GIS intelligent: CommonGIS project view", *AGILE'99 Conference,* Rome, April 15–17, pp. 19-24.

Andrienko, G., and Andrienko, N., (1999e). Seasonal Migration of White Storks, 1998–1999. Online: http://www.ais.fhg.de/and/java/birds/ (03/10/02).

Andrienko, G. L., and Andrienko, N. V., (1999f) "Interactive maps for visual data exploration", *International Journal Geographic Information Science*, 13(4), 355-374.

Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A., (2000) "Guidelines for using multiple views in information visualization", *Proceedings of AVI '2000*, pp. 110-119.

Becker, R. A., and Cleveland, W. S., (1987) "Brushing scatterplots", *Technometrics*, 29(2), 127-142.

Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T., (1993) "Toolglass and magic lenses: the see-through interface", *Proceedings SIGGRAPH'93,* Anaheim, CA, pp. 73-80.

Boukhelifa, N., Roberts, J. C., and Rodgers, P. J., (2003) "A coordination model for exploratory multi-view visualization", *Proceedings of Coordinated and Multiple Views in Exploratory Visualization (CMV2003),* July 2003, IEEE.

Brodlie, K., Poon, A., H., W., L., B., G., B., A., and G., (1993) "GRASPARC – a problem solving environment integrating computation and visualization", *Proceedings Visualization'93*, IEEE Computer Society Press, pp. 102-109.

Brunsdon, C., (2001) "The comap: exploring spatial pattern via conditional distributions", *Computers, Environment and Urban Systems*, 25, 53-68.

Card, S. K., (1996) "Visualizing retrieved information: a survey", *IEEE Computer Graphics and Applications*, 3(16), 63-67.

Carr, D., (1999) "Guidelines for designing information visualization applications", *Proceedings of ECUE'99,* Stockholm, Sweden, pp. 1-3.

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.

Chi, E. H. et al., (1998) "Principles for information visualization spreadsheets", *IEEE Computer Graphics and Applications*, 18(4), 30-38.

Chuah, M. C., and Roth, S. F., (1995) "SDM: selective dynamic manipulation of visualizations", *Proceedings UIST'95*.

Cleveland, W. S., and McGill, M. E., (1988) *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth.

DiBiase, D., (1990) "Visualization in the earth sciences, earth and mineral sciences", *Bulletin of the College of Earth and Mineral Sciences, The Pennsylvania State University*, 59(2), 13-18.

Dupont, W. D., and Plummer, W. D. J., (2003) "Density distribution sunflower plots", *Journal of Statistical Software*, 8(3).

Dykes, J., (1997a) "cdv: a flexible approach to ESDA with free software connection", *Proceedings of the British Cartographic Society 34th Annual Symposium*, pp. 100-107.

Dykes, J. A., (1997b) "Exploring spatial data representation with dynamic graphics", *Computers & Geosciences*, 23(4), 345-370, Online: http://www.mimas.ac.uk/argus/ICA/J.Dykes/.

Dykes, J. A., (2000) "An approach to virtual environments for fieldwork using linked geo-referenced panoramic imagery", *Computers, Environment and Urban Systems*, 24(2), 127-152, Online: http://www.geog.le.ac.uk/jad7/CE&US/ (6/21/1999).

Felger, W., and Schröder, F., (1992) "The visualization in-put pipeline – enabling semantic interaction in scientific visualization", *Proceedings of Eurographics 92*, Eurographics Association, pp. 39-151.

Ferreira, J. J., and Wiggins, L. L., (1990) "The density dial: a visualization tool for thematic mapping", *Geo Info Systems*, 10, 69-71.

Fuhrmann, S., and MacEachren, A. M., (1999) "Navigating desktop GeoVirtual environments", *IEEE Information Visualization 99, Late Breaking Hot Topics Proceedings,* San Francisco, CA, Oct. 23–28, pp. 11-14. Online: http://www.geovista.psu.edu/publications/amm/ammIV99.pdf.

Fuhrmann, S., and MacEachren, A. M., (2001) "Navigation in desktop geovirtual environments: usability assessment", *20th International Cartographic Conference – Mapping the 21st Century,* Beijing, pp. 2444-2453.

Gahegan, M., (1998) "Scatterplots and scenes: visualization techniques for exploratory spatial analysis", *Computers, Environment and Urban Systems*, 22(1), 43-56.

837 Guoray Cai, G., (2001) "GeoVIBE: a visual interface to geographic digital library",
838   *JCDL Workshop on Visual Interfaces*, ACM/IEEE.
839 Haber, R. B., and McNabb, D. A., (1990) "Visualization idioms: a conceptual model for
840   scientific visualization systems", In: Nielson, G., Shriver, B., and Rosenblum, L., (eds.),
841   *Visualization in Scientific Computing*, IEEE Computer Society Press, pp. 74-93 .
842 Holmquist, L. E., and Ahlberg, C., (1997) "Flip zooming: a practical focus + context
843   approach to visualizing large information sets", *Procedings HCI International'97.*
844   Amsterdam: Elsevier, pp. 763-766.
845 Jacobson, A. S., Berkin, A. L., and Orton, M. N., (1994) "LinkWinds: interactive
846   scientific data analysis and visualization", *Communications of the ACM*, 37(4),
847   43-52.
848 Jankun-Kelly, T. J., and Ma, K.-L., (2001) "Visualization exploration and encapsulation
849   via a spreadsheet-like interface", *IEEE Transactions on Visualization and Computer
850   Graphics*, 7(3), 275-287.
851 Kandogan, E., and Shneiderman, B., (1997) "Elastic windows: evaluation of multi-
852   window operations", *ACM Proceedings CHI'97: Human Factors in Computing
853   Systems,* Atlanta, GA, ACM, pp. 250-257.
854 Keahey, T. A., and Robertson, E. L., (1997) "Nonlinear magnification fields",
855   *Proceedings of the IEEE Symposium on Information Visualization: IEEE
856   Visualization,* Oct.1997, pp. 51-58.
857 Kosara, R., Miksch, S., and Hauser, H., (2002) "Focus + context taken literally", *IEEE
858   Computer Graphics & Applications*, 22(1), 22-29, (Jan/Feb 2002).
859 Lamping, J., Rao, R., and Pirolli, P., (1995) "A focus + context technique based on
860   hyperbolic geometry for visualizing large hierarchies", *Proceedings of CHI'95 –
861   Human Factors in Computing Systems,* Denver, CO, May 1995. New York: ACM
862   Press, pp. 401-408.
863 Lokuge, I., and Ishizaki, S., (1995) "Geospace: an interactive visualization system for
864   exploring complex information spaces", *CHI'95 Proceedings*, ACM, pp. 409-414.
865 MacEachren, A. M., Hardisty, F., Gahegan, M., Wheeler, M., Dai, X., Guo, D., and
866   Takatsuka, M., (2001) "Supporting visual integration and analysis of geospatially-
867   referenced statistics through web-deployable, cross-platform tools", *Proceedings,
868   dg.o.2001, National Conference for Digital Government Research,* Los Angeles,
869   CA, May 21–23, pp. 17-24.
870 Mackinlay, J. D., Robertson, G. G., and DeLine, R., (1994) "Developing calendar
871   visualizers for the information visualizer", *Procedings ACM UIST,* Nov., 109-119.
872 McDonald, J. A., Stuetzle, W., and Buja, A., (1990) "Painting multiple views of complex
873   objects", *ECOOP/OOPSLA'90 Proceedings*, ACM Press, pp. 245-257.
874 Monmonier, M., (1989) "Geographic brushing: enhancing exploratory analysis of the
875   scatterplot matrix", *Geographical Analysis*, 21(1), 81-84.
876 North, C. and Shneiderman, B., (1997) A taxonomy of multiple window coordinations,
877   *Tecnical Report #CS-TR-3854*. University of Maryland, Computer Science Dept.
878 North, C., and Shneiderman, B., (2000a) "Snap-together visualization: a user interface
879   for coordinating visualizations via relational schemata", *Proceedings of Advanced
880   Visual Interfaces 2000,* Italy, pp. 128-135.

North, C., and Shneiderman, B., (2000b) "Snap-together visualization: can users construct and operate coordinated visualizations?", *International Journal of Human–Computer Studies*, 53(5), 715-739.

Olson, G. M., Malone, T. W., and Smith, J. B., (2001) *Coordination Theory and Collaboration Technology*, Lawrence Erlbaum Associates, Inc.

Pattison, T., and Phillips, M., (2001) "View coordination architecture for information visualisation", *Australian Symposium on Information Visualisation,* Sydney, Australia, ACS.

Pepke, E. and Lyons, J., (1993) *SciAn: User's Manual*. Online: http://www.scri.fsu.edu/~lyons/scian (01/10/02).

Perlin, K., and Fox, D., (1993) "Pad: an alternative approach to the computer interface", In: Kajiya, J. T., (ed.), *Proceedings of the 20th Annual ACM Conference on Computer Graphics (SIGGRAPH'93),* Anaheim, CA, Aug. 2–6. NY: ACM Press, pp. 57-64.

Plumlee, M., and Ware, C., (2003) "Integrating multiple 3D views through frame-of-reference interaction", *Proceedings of Coordinated and Multiple Views in Exploratory Visualization (CMV2003),* July 2003, IEEE.

Roberts, J. C., (1998a) "On encouraging multiple views for visualization", *Information Visualization IV'98*, IEEE Computer Society, pp. 8-14.

Roberts, J. C., (1998b) "Waltz – an exploratory visualization tool for volume data, using multiform abstract displays", *Visual Data Exploration and Analysis V, Proceedings of SPIE,* Jan. 1998, p. 11.

Roberts, J. C., (1999) "On encouraging coupled views for visualization exploration", *Visual Data Exploration and Analysis VI, Proceedings of SPIE,* Jan. 1999, SPIE, pp. 14-24.

Roberts, J. C., Knight, R., Gibbins, M., and Patel, N., (2000) "Multiple window visualization on the web using VRML and the EAI", In: Hollands, R., (ed.), *Proceedings of the Seventh UK VR-SIG Conference,* Sept. 2000, pp. 149-157.

Roth, S. F. et al., (1996) "Visage: a user interface environment for exploring information", *Proceedings of Information Visualization*, 3-12.

Senay, H., and Ignatius, E., (1994) "A knowledge-based system for visualization design", *IEEE Computer Graphics and Applications*, 6(14), 36-47.

Sheelagh, M., (1997) "Extending distortion viewing from 2D to 3D", *IEEE Computer Graphics and Applications*, 4(17), 42-51.

Shneiderman, B., (1994) "Dynamic queries for visual information seeking", *IEEE Software*, 11(6), 70-77.

Snyder, J. P., (1987) "Magnifying glass azimuthal map projection", *The American Cartographer*, 14(1), 61-68.

Spence, R., (2001) *Information Visualization*. Harlow: Addison Wesley/ACM Press Books, p. 206.

Steiner, E. B., MacEachren, A. M., and Guo, D., (2001) "Developing and assessing light-weight data-driven exploratory geovisualization tools for the web", *20th International Cartographic Conference,* Beijing, China, Aug. 6–10.

925   Stonebraker, M., Chen, J., Nathan, N., Paxon, C., Su, A., and Wu, J., (1993) "Tioga: a
926   database-oriented visualization tool", *Proceedings Visualization'93*, IEEE Compu-
927   ter Society Press, pp. 86-93.
928   Symanzik, J., Cook, D., Lewin-Koh, N., Majure, J. J., and Megretskaia, I., (2000)
929   "Linking ArcView™ and XGobi: insight behind the front end", *Journal of
930   Computational and Graphical Statistics*, 9(3), 470-490.
931   Theus, M., (2002a) "Geographical information systems", In: Kloesgen, W., and Zytkow,
932   J., (eds.), *Handbook of Data Mining and Knowledge Discovery*, Oxford University
933   Press.
934   Theus, M., (2002b) "Interactive data visualization using Mondrian", *Journal of
935   Statistical Software*, 7(11).
936   Unwin, A., (2001) "R objects, two interfaces! (R objects to interfaces?)", *Proceedings of
937   the 2nd International Workshop on Distributed Statistical Computing*.
938   Unwin, A. R., Hawkins, G., Hofmann, H., and Siegl, B., (1996) "Interactive graphics for
939   data sets with missing values – MANET", *Journal of Computational and Graphical
940   Statistics*, 5(2), 113-122.
941   Ward, M., (1994) "XmdvTool: integrating multiple methods for visualizing multivariate
942   data", *Proceedings Visualization'94*, IEEE Computer Society Press, pp. 326-333.
943   Wexelblat, A., and Maes, P., (1999) "Footprints: history-rich tools for information
944   foraging", *Proceedings of CHI 99,* Pittsburgh, PA, ACM, pp. 270-277.
945   Williams, C., Rasure, J., and Hansen, C., (1992) "The state of the art of visual languages
946   for visualization", *Proceedings Visualization'92*, IEEE Computer Society Press,
947   pp. 202-209.
948   Williamson, C., and Shneiderman, B., (1992) "The dynamic home finder: evaluating
949   dynamic queries in a real-estate information exploration system", *Proceedings of the
950   ACM SIGIR'92 Conference*, ACM, pp. 338-346.
951   Wright, H., (1996) The HyperScribe Data Management Facility, *NAG Technical Report
952   TR1~96*, NAG.
953   Yagel, R., Ebert, D. S., Scott, J. N., and Kurzion, Y., (1995) "Grouping volume renderers
954   for enhanced visualization in computational fluid dynamics", *IEEE Transactions on
955   Visualization and Computer Graphics*, 2(1), 117-132.