Appendix B: Review of Coordinated-Visualization Systems

Coordinated visualization systems have become an important and diverse topic. Many such systems have been built. Most of these systems are data flexible (defined in Chapter 2). That is, typically they can be used to visualize different data sets, but are usually fixed in terms of the visualizations and coordinations in their user interface. This Appendix reviews many of these systems from the field. As in the rest of this dissertation, the focus is on coordinations for information exploration.

A simple taxonomy is used to lay out the space of these systems [NS97], loosely based on the conceptual model of visualization coordination described in Chapter 3. Visualizations have two basic classes of actions:

- **Select**: Users can select and highlight data items in the visualization to express interest in them, or possibly to initiate other forms of manipulation on them.

- **Navigate**: Users can navigate the visualization to focus on data items or to display other data items (e.g. scroll, pan, zoom, slice, rotate, ascend/descend tree, follow link, open file, etc.). For the purposes of this taxonomy, navigate also includes the *load* action to load other data into a visualization as a form of navigation through the larger data context.
Coordinating a pair of visualizations tightly couples one of these actions in the one visualization to another action in the other visualization. The taxonomy classifies coordinations by the three possible combinations of actions (Figure B.1):

1. Select ↔ select
2. Navigate ↔ navigate
3. Select ↔ navigate (which is equivalent to navigate ↔ select due to bi-directionality)

![Figure B.1: A taxonomy of coordinations](image)

### B.1 Select ↔ Select

This coordination tightly couples selecting items in one visualization to selecting items in another visualization, to help users correlate equivalent or related items. When users select (highlight, paint, brush) an item (or set of items) in one visualization, the system immediately highlights the equivalent item (or set), representing the same underlying data elements, in the other visualization.

Many exploratory data analysis systems use this coordination to visualize high-dimensional data point sets with multiple coordinated plots. Common examples are Datadesk [Vel88], SAS Insight, JMP, EDV [EW95], Spotfire [AW95], XGobi [BCS96],
XmdvTool [WA95]. Invention of this brushing-and-linking concept is generally credited to Prim-9 [FFT74] or Newton [New78]. [Mon89] introduced brushing to GIS by brushing between plots and geographic choropleth maps. XmdvTool provides the capability to brush regions in attribute space as well as individual data items. For example, in Figure B.2 an n-dimensional region is selected in both the plot matrix and parallel-coordinates graph.

![Figure B.2: XmdvTool](image)

For examples with other types of data, the Navigational View Builder [MFH95] (Figure B.3) brushes nodes in hierarchical information, linking Treemaps (emphasizing numerical and categorical attributes), ConeTrees (emphasizing structure), and outliners (emphasizing node names). With Lilac [Bro91], a two-window document editor, selecting text in the WYSIWYG page window also selects the corresponding text in the source text window (similar to HTML code).
An interesting variation is the Attribute Explorer [STD95], which uses additive encoding of multiple brushes (Figure B.4). It displays multi-dimensional data in a series of 1-dimensional histograms, and users can select a range in each histogram. Then, data points are color coded by the number of attribute selections they are contained in. Points that satisfy more selections are lighter, fewer selections are darker.
Visage VQE [DRK97] extends brushing to multiple relations. Visualizations containing joins of relations can be brushed if they share a common relation anywhere in their join paths. An early prototype of LinkKit [Nor98] demonstrates brushing across many-to-many joins for exploring authors, publications, and other references (Figure B.5).

Figure B.5: LinkKit prototype in Elastic Windows
B.2 Navigate ↔ Navigate

This coordination tightly couples navigation in one visualization to simultaneous navigation in another visualization. This maintains synchronization of visualizations while navigating (e.g. scrolling, panning, zooming, slicing, traversing, etc.) through correlated information spaces (e.g. Figure B.6)

Figure B.6: Synchronized scrolling

Synchronized scrolling tightly couples the scroll bars of two visualizations. WordPerfect displays a document’s formatting codes in a separate frame adjacent to the main text that with synchronized scrolling. This approach avoids losing the relationship between representations and saves users from tedious repetition of scrolling actions in each frame. With Logos Bible Software, users can simultaneously scroll through different Bible translations, commentaries, and study guides, which all share a common ordered hierarchical structure of book, chapter, and verse. SeeDiff [BE96] synchronizes scrolling through two version of a source code file for analyzing changes (Figure B.7).
DEVise [LRB97] generalizes this synchronized navigation strategy to 2D, allowing users to synchronously pan and zoom multiple 2D plots with common X and Y axes. The Neighborhood Viewer [CSP97] (Figure B.8) extends this to 3D slicing by synchronously panning correlated cross-section, CT, and MRI images through the human body. Chi et al. [CBR97] (Figure B.9) extends synchronized navigation to general 3D. It arranges many small 3D visualizations in a spreadsheet grid and synchronizes their rotation, zooming, etc.

Figure B.7: SeeDiff
Figure B.8: Neighborhood Viewer

Figure B.9: Spreadsheet Visualization
B.3 Select ↔ Navigate

This coordination tightly couples selecting items in one visualization to navigating in another visualization, and vice versa (i.e. navigate to select). Users can select items from overviews to navigate to corresponding detailed information in separate visualizations. Likewise, navigating the detailed visualization indicates the corresponding selection in the contextual overview (Figure B.10).

Figure B.10: Overview and detail

Overviews provide a global map of information, and detail visualizations provide detailed information about a small portion. Coordinating the visualizations indicates the location of and provides a mechanism for navigating the detail from within the context of the overview. This is advantageous over detail-only browsers since overviews indicate what information is available, provide context for details, guide browsing, promote exploration, and help avoid getting lost. This strategy contrasts with distortion-oriented techniques [LA94], which attempt to show details within the context
of the overview in a single visualization by distorting the view. An important metric is
the zoom factor between the overview selection and detail. Larger zoom factors allow
for more information. While zoom factors for distortion techniques are typically
limited to 5 or less, coordinated visualizations can reach zoom factors of 20 for attribute
spaces [PCH92] and 1000 for data aggregation strategies. Also, several of these
 coordinations can be chained together using intermediate visualizations [PCS95] to
multiply zoom factors.

With the Navigational View Builder [MFH95] (Figure B.3), and other web site
visualization tools, users can select any node in a visualization of a large site to display
that web page in a separate browser window. This strategy has become commonplace
in user interface design. It is used in many standard tools such as Microsoft Word and
Windows Explorer. It is also used with frames on web pages. Simultaneous menus
[HKV00] enables users to select from multiple overviews to display results in a single
detail visualization based on all the selections (Figure B.11).

Figure B.11: Simultaneous Menus
A variant of this approach shows details of selections in a new popup window instead of a given static window, as in the FilmFinder [AS94] (Figure B.12). Selecting a dot on a scatter plot displays that record’s fields, including pictures. However, this requires additional clicks to dismiss the popup each time or move it aside.

The select-to-navigate coordination can be used to drill down through layers of a database, with separate visualizations for each layer. CASCADE [SMH96] (Figure B.13) provides four layers of coordinated visualizations for zooming through 4 different levels of scale within a large document database: the Docuverse level (collection of up to 5000 documents), Webview (up to 500 documents), Landmarks (within a single document), and Preview (individual item in a document, such as a hyperlink).
For attribute spaces, dragging or resizing a field-of-view indicator (selection) in the overview is tightly coupled to pan or zoom (navigation) the detail visualization, and vice versa. Scroll bars, albeit poor overviews of their associated main window, are a simple 1D example. The Information Mural [JS95] (Figure B.14), SeeSoft [BE96] (Figure B.15), ValueBars [Chi92], and others [Eic94] provide highly reduced images of large documents or software code, using color coding and anti-aliasing algorithms, for navigating 1D document windows with fields-of-view.

The “cursor” link in DEVise [LRB97] links a 2D field-of-view in an overview plot to the panning control of the axes in a detail plot. Similar 2D approaches are used in Pad++ portals [BH94] and in PDQ Trees [KPS97] (Figure B.16) for hierarchies laid out on a 2D surface. Plaisant et al. [PCS95] developed a formal notation for specifying this coordination for browsing large 2D images that is replicated in many digital imaging packages such as Adobe Photoshop.
We have developed a method for displaying and navigating large information spaces using multiple view techniques. This work is described in this subsection, the implementation of the message space visualization is discussed here, and the specific techniques and the global view functionality are presented. The implementation demonstrates one method of the new approach to visual mechanisms. We call our views of large information spaces "information murals," and describe them in Section 2. In Section 4 we discuss several application areas where the information murals are useful, and compare our notions with related work in these areas.

Figure B.14: Information Mural

Figure B.15: SeeSoft
For a 3D volumetric image space, with the Visible Human Explorer [NSP96] users can rapidly navigate each orthogonal 2D cross-section visualization through the human body by dragging the corresponding cut lines in the other visualizations, and receive continuous feedback of contents (Figure B.17).
An extension to this approach is to use one visualization to keep a history of navigation in other visualizations. With select-to-navigate coordination, users can revisit previous states. PadPrints [HRH98] (Figure B.18) and the Graphical History Browser [AS95] both maintain iconic node-link diagrams of visited web pages for a web browser. Utting and Yankelovich [UY89] review several such approaches for
hypertext navigation. They extend their Intermedia system to include a map of destinations that can be reached from the current page as well, hence providing a selectable visualization of both history and potential future.

Figure B.18: PadPrints

B.4 Summary

Many coordinated-visualization interfaces have been developed, and have proven to be very useful and effective. Yet, these are only a small number in comparison to the myriad different combinations of visualizations and coordinations that are needed for so many unique users, data, and tasks. Clearly, these many examples serve to point out the need for Snap-Together Visualization.
References


