CHAPTER 10

Interaction Concepts

Interaction within the data and information visualization context is a mechanism for modifying what the users see and how they see it. Many classes of interaction techniques exist [418], including:

- **navigation**—user controls for altering the position of the camera and for scaling the view (what gets mapped to the screen) such as panning, rotating, and zooming.
- **selection**—user controls for identifying an object, a collection of objects, or regions of interest to be the subject of some operation, such as highlighting, deleting, and modifying.
- **filtering**—user controls for reducing the size of the data being mapped to the screen, either by eliminating records, dimensions, or both.
- **reconfiguring**—user controls for changing the way data is mapped to graphical entities or attributes, such as reordering the data or layouts, thereby providing a different way of viewing a data subset.
- **encoding**—user controls for changing the graphical attributes, such as point size or line color, to potentially reveal different features.
- **connecting**—user controls for linking different views or objects to show related items.
- **abstracting/elaborating**—user controls for modifying the level of detail.
- **hybrid**—user controls combining several of the above in one technique, for example, increasing the screen space assigned to one or more focus areas to enable users to see details, while showing the other areas of data in a smaller space, in a way that preserves context.
10. Interaction Concepts

A variety of techniques and tools for performing interactions within data and information visualization systems have been proposed to date. While some of these tools appear quite unrelated, they actually share a number of features and serve a common purpose. As the field of data and information visualization evolves, it is beneficial to try to identify unifying themes and frameworks to help solidify our understanding of the basic building blocks of the field.

In this chapter, we describe such a framework for interaction techniques, identifying distinct classes and shared concepts that will help facilitate discussions and focus future research. We begin by identifying classes of interactive operations and describing them in terms of operators and the operand (the space upon which the operator is applied). Each is described in detail, with references to relevant techniques in the literature. We then define an architecture that combines the different interaction spaces into a single pipeline, along with the interface tools needed by the user to control the process. We conclude with some ideas for future research in the development and assessment of this framework. This chapter draws heavily from a paper written by one of the authors [379] and presented at the 2004 Eurographics Symposium on Visualization.

10.1 Interaction Operators

In this section we describe in more detail a wide range of interaction operations commonly found in data and information visualization. This list is not exhaustive, but it covers many typical interaction tools. Readers interested in more extensive lists of visualization interactions are directed to Keim’s classification [204] and Chi’s taxonomy [61]. One important note is that interaction operators often can fall into many of the suggested interaction classes, and that almost all operators can be made interactive or automatic when in a visualization, or actually a noninteractive part of some visualization. For example, zooming is available in almost all visualizations; however, it can be thought of as generating a new visualization, especially if different data (such as more roads in a map) need to be displayed.

10.1.1 Navigation Operators

Navigation (also sometimes referred to as exploration) is used to search for a subset of data to be viewed, the orientation of this view, and the level of detail (LOD). The subset in question may be one that is recognized by some
visual pattern or one on which further or more detailed exploration is desired. In a typical three-dimensional space, this can be specified using a camera location, a viewing direction, the shape and size of the viewing frustum, and an LOD indicator. In multiresolution visualizations, LOD changes can correspond to drilling down or rolling up hierarchical representations of the data.

Navigation operators can work in absolute or relative coordinates within their particular spaces. Incremental navigation may have different granularities, depending on whether the user wants a small or significant change. Navigation can be user-driven or automatic; a good example of automated exploration is the grand tour [15], where multidimensional data is explored by flying along a path that smoothly covers many or all possible orientations of the data space, as projected onto two dimensions (Figure 10.1). The user can control the step size between views, with the trade-off being smoothness versus the number of projections that need to be inspected. Another automated form of exploration is projection pursuit [167], where projections are computationally analyzed and the subset of views that exceed a user's threshold for "interestingness" is displayed.

### 10.1.2 Selection Operators

In selection, the user isolates a subset of the display components, which will then be subjected to some other operation, such as highlighting, deleting, masking, or moving to the center of focus. Many variations on selection have been developed to date [395], and decisions need to be made on what the results should be for a sequence of selections. For example, should the new
selection replace the previous selection or supplement the previous selection? The granularity of selection is also an issue. Clicking on an entity in the display might result in selection of the smallest addressable component (e.g., a vertex or edge) or might target a broader region around the specified location (e.g., a surface, region of the screen, or object).

Selection can be articulated in many different ways. The user may click on entities, paint over a selection of entities (e.g., holding the mouse button down while moving over the entities of interest), or otherwise isolate the entities via techniques such as bounding boxes and lassos. Finally, selections may be performed in an indirect manner, where the system selects elements that match a user's input set of constraints. An example would be the selection of nodes in a graph that have a user-specifed distance from a selected node.

### 10.1.3 Filtering Operators

Filtering, as the name implies, reduces the volume of data to be visualized by setting constraints specifying the data to be preserved or removed. A good example of such a filter is the *dynamic query specification* described by Shneiderman et al. [318]. One- or two-handled sliders are used to specify a range of interest, and the visualization is immediately updated to reflect the changes made by the user. Range queries are just one form of filtering, however. One might also select items from a set or list to preserve or hide, such as the column hiding operation in Excel. Figure 10.2 shows the effects of filtering on a visualization to simplify the view and ease interpretation.

![Figure 10.2](image-url)

*Figure 10.2.* Filtering rows and columns of the grades data set using XmdvTool.
The distinction between filtering and selection followed by deletion or masking is a subtle, but important point. Filtering, in general, is most often done in an indirect manner, e.g., the filter specification is not performed on the data visualization itself, but via a separate interface or dialog box. In fact, filtering is often done prior to viewing the data, to avoid overloading the data display. Selection is most often done in a direct manner, by indicating objects on the visualization via mouse motions, for example. The operation performed on the selected subset can result in a view that is indistinguishable from a filtering operation.

10.1.4 Reconfiguring Operators

Reconfiguring the data within a particular visualization can often be used to expose features or cope with complexity or scale. By reorganizing the data, say by filtering some dimensions and reordering those that remain, different views are provided to the user. For example, a powerful tool with table-based visualizations is to sort the rows or columns of the data to highlight trends and correlations. Other types of reconfiguration might be to change the dimensions being used to control the $x$- and $y$-coordinates of a plotted marker, or even to derive positions based on transformations of the data. Popular instances of this include the use of principal component analysis (PCA) or multidimensional scaling (MDS) to layout data points so that relationships among all the dimensions are best conveyed in the 2D layout.

10.1.5 Encoding Operators

Any given data set can be used to generate countless different visualizations. Recoding can provide the user a library of possible different types of visualization; features of the data that are difficult or impossible to see with one such mapping might become quite apparent in another. For example, a scatterplot with one axis representing years may have many points overlap, whereas a parallel coordinate display would represent these uniquely. Many visualization tools today support multiple types of visualization, because no single visualization is effective for all tasks that need to be carried out by the user. Each visualization is most suitable for a subset of data types, data set characteristics, and user tasks. While some work has been done to identify or select the best visualization, it is apparent that such guidelines are at best suggestions, and the analyst is most likely to benefit from examining their data using a number of different mappings and views. This is the essence of interactive visualization.
Other forms of encoding operations include those that modify the color map used, the size of graphical entities, and their shape. These can be considered variations within a particular type of visualization, and can be used to emphasize or reveal features of interest. Even limitations of some visualizations can be overcome using variations. The overlapping issue in scatterplots where occluded points are not visible can be overcome by jittering the points or making the size of the points reflect the number of points at that same position. Other attributes of graphical entities that can be controlled include opacity, textures, line or fill style, and dynamic attributes such as fade or flashing rate. Note that these effects can often be mimicked by performing transformations on the data itself, rather than on the graphical entities.

10.1.6 Connection Operators

A frequent use for selection operations is to link the selected data in one view to the corresponding data in other views. While other forms of connection between subwindows of an application exist, such as when opening a new data file, linked selection is probably the most common form of communication between windows found in modern visualization tools. Its popularity stems in large part from the fact that each view of one's data can reveal interesting features, and that by highlighting such a feature in one view, it

Figure 10.3. Example of linked brushing. A cluster is isolated in parallel coordinates, with linked selection in the scatterplot matrix. Selected data is dark, while brush extents are shown as light bands or rectangles.
is possible to build a more complete mental model of the feature by seeing how it appears in other views (see Figure 10.3). This can also help reveal relationships between this feature and others in the data set. For example, when examining univariate spatial data, it is often useful to jump between the spatially referenced view and the dependent variable view, which often does not preserve the spatial attributes.

When the selection data is allowed to be interactively changed, the operator is called brushing, in which case the user is continuously changing the selection in one view, and the corresponding linked data in one or more other views is highlighted. The resulting interactive and dynamic display provides information about the changes in values in the linked displays.

Another strength of linked brushing is in specifying complex constraints on one's selection. Each type of view is optimized for conveying certain types of information, as well as for specifying conditions on particular types and with a particular degree of accuracy. Thus, for example, one might specify a temporal constraint using a visualization containing a timeline, a constraint on a name field using a sorted list view, and a geographic constraint using a map. While each is effective as a tool for accurate and intuitive specification of a part of a query, none could be used for the complete query.

In some situations, the user may want to unlink some visualizations in order to maintain a given view while exploring a different area of the data or different data set. Some systems allow the user to indicate for each window whether it is transmitting information to other views, and from which other windows it will receive input. A user may also want to constrain the type of information being communicated, as well as its direction. Some types of interaction may be local to a particular window, e.g., zooming in and out, while others are meant to be shared, such as reordering dimensions. Also, in some situations, such as with hierarchically related windows, it may make more sense for the information to move from parent to child, but not the other way. Thus, a fairly rich set of connection and communication options may be needed to maximize flexibility.

10.1.7 Abstraction/Elaboration Operators

In dense data and information displays, it is often desirable to focus in on a subset of the data to acquire details (elaboration) while reducing the level of detail (abstraction) on other parts of the data set. One of the most popular techniques of this type is using distortion operators. While some researchers classify distortion as a visualization technique, it is actually a transformation that can be applied to any type of visualization. Like panning
and zooming, distortion is useful for interactive exploration. Many distortion operators (also called functions) have been proposed in the past [232]. These include methods that distort the entire space being analyzed, and others that have more localized effects. The distortion may take place within the original visualization, or may appear in a separate window. Distortions vary in the features that are preserved and the amount of context maintained. For example, text distortion techniques strive for readability within a small region of interest, with the rest of the text positioned to reinforce document structure, but not generally readable. For other types of distortion, it is important that the undistorted and compressed regions continue to convey useful information, while details are provided in the focus area.

Distortion operators may be linear or nonlinear, with $0^{th}$, $1^{st}$, or $2^{nd}$ order continuity (discontinuous operators are also possible). Operators may also operate on structures, rather than on continuous spaces, and thus may be specific to a particular type of operand (see the next section for details). Different operators have different footprints, e.g., the shape and extents of the space affected by the transformation. Common footprint shapes include rectangular and circular, and their analogous hyperboxes and hyperellipses for higher dimensional spaces. Extents are usually specified by a distance function within the space being distorted, and are often multidimensional. These extents can be fixed or variable, user-controlled, or driven by the semantics of the information (e.g., page or paragraph extents for text distortion). Finally, operators generally have a variable degree of magnification, depending on the level of detail desired.

### 10.2 Interaction Operands and Spaces

Parameters of the interaction operators described in the previous section are discussed in more detail later in the chapter. First, however, we present our categorization of the interaction operands, as this will help clarify the role these parameters take in the interaction process, and their semantics within the different spaces.

An interaction operand is the section of space upon which an interactive operator is applied. To determine the result of an interactive operation, one needs to know within what space the interaction is to take place. In other words, when a user clicks on a location or set of locations on the screen, what entities does he or she wish to indicate? Possibilities include the pixel(s), the data value or record mapped to the location, or even the component of the visualization structure (e.g., an axis) at or near that location. We have
identified several distinct classes of interaction spaces. Each is described below, including examples of existing interaction techniques that fall into each class.

10.2.1 Screen Space (Pixels)

Navigation in screen space typically consists of actions such as panning, zooming, and rotation. Note that in each case, no new data is used; the process consists of pixel-level operations such as transformation, sampling, and replication.

Pixel-based selection means that at the end of the operation, each pixel will be classified as either selected or unselected. As previously mentioned, the selection can be performed on individual pixels, rectangles or circles of pixels, or on arbitrarily shaped regions that the user specifies. Selection areas may also be contiguous or noncontiguous.

Distortion in screen space involves transformations on pixels, e.g., \((x', y') = f(x, y)\). In order to avoid occlusion, this function should be order-preserving and at least \(C^0\) continuous [190]. The magnification \(m(x, y)\) at a particular point is simply the derivative of this transformation, and, in fact, it is

![Figure 10.4](image_url)

In screen space techniques, pixel regions are enlarged or reduced to provide selective detail. In this scatterplot matrix display, a center of focus has been selected and magnified using a confocal lens technique. (Image from [379], © 2004 IEEE.)
10. Interaction Concepts

useful to be able to switch between transformations and their associated magnifications when controlling the distortion process [190]. Examples of screen space techniques are the fisheye lens [121] and rubber sheet methods [48, 306], although the latter techniques could also be placed in the object-space category described below. Figure 10.4 is an example of this type of distortion.

10.2.2 Data Value Space (Multivariate Data Values)

Navigating in data value space involves using the data values as a mechanism for view specification. The analogous operations for panning and zooming would be to change the data values being displayed; panning would shift the start of the value range to be shown, while zooming would decrease the size of this range.

Data value space selection is similar to a database query in that the user specifies a range of data values for one or more data dimensions. This can be performed via direct manipulation, as in the data-driven brushing reported in [249] (see Figure 10.5(a)) or via sliders or other query specification mechanisms [318]. Selection may involve a single value, or one or nonlinear ranges of values.

Data value space is perhaps the most obvious space in which to perform filtering. When visualizing extremely large data sets, it is common to first reduce the data to a particular region of data space. For spatial data, this is analogous to clipping the data falling outside of a viewing region; for nonspatial data, this involves eliminating some records, dimensions, or both. For example, sampling might be used to examine a representative subset of a large data repository [96] where the visualization might otherwise be too cluttered to retrieve any useful patterns. Dimensions may also be filtered [411] to allow the user to either examine a subset of dimensions with similar characteristics or select representatives from clusters of related dimensions.

For distortion in data value space, data values \((d_0, d_1, \ldots, d_n)\) may be transformed via a function \(j : (d'_0, d'_1, \ldots, d'_n) = j(d_0, d_1, \ldots, d_n)\) prior to visualization. In fact, each dimension may have its own transformation function \(j_i : d'_i = j_i(d_i)\). In its most general case, the function \(j_i\) could depend on any number of dimensions, although user control of such a function might be problematic. An example of data value-space distortion is the dimensional zooming found in XmdvTool [118], where each dimension of a selected subset of the data is scaled so that the subset fills the display area (see Figure 10.5).
10.2. Interaction Operands and Spaces

Figure 10.5. In data value space distortion, transformations are performed according to the dimensionality of the data. In this example, generated using XmdvTool [118], an N-dimensional hyperbox is selected via painting over a section of an axis and scaled in all dimensions (by different amounts) to fill a unit hypercube, which is then displayed. Animation is used to preserve context. Clusters and anomalies within the selected region are much easier to see in the zoomed version. (Image from [379], © 2004 IEEE.)

10.2.3 Data Structure Space (Components of Data Organization)

Data can be structured in a number of ways, such as lists, tables, grids, hierarchies, and graphs. For each structure, one can develop interaction mechanisms to indicate what portions of the structure will be manipulated, and how this manipulation will be manifested. Navigation in data structure space involves moving the view specification along the structure, as in showing sequential groups of records, or moving down or up a hierarchical structure (as in drill-down and roll-up operations). For example, Figure 10.6 shows the difference between a screen space zoom (involving pixel replication) and a data structure space zoom (involving retrieval of more detailed data). A technique presented by Resnick et al. [289] selects subsets of data to be visualized by specifying a focus, extents, and density in a regular grid structure, where the density can be a function of distance from the focus.

Selection in data structure space generally involves displaying the structure and allowing the user to identify regions of interest within it. This
Figure 10.6. In screen space zooming (left), pixels are replicated to provide selective size, while in data space zooming (right), the data itself can be resampled at the appropriate resolution.

in turn can drive the display of the data corresponding to the selected substructure. For example, structure-based brushing [119] involves controlling the selection of data stored in a cluster hierarchy, with interactions such as highlighting data that fall within a particular branch of the tree. Similarly, InterRing is a radial space-filling hierarchy visualization tool that allows semi-automatic selection of nodes, according to the hierarchical

Figure 10.7. Selection of nodes in a hierarchy via InterRing. Nodes with a red stripe in them have been selected via a user-specified query, rather than one node at a time. (Image from [409], © 2003 IEEE.)
structure [409]. Figure 10.7 shows a dimension hierarchy in InterRing, with a subset of terminal nodes automatically selected via a query on their common parent node.

Filtering is often performed in data structure space to reduce the amount of information on the display. For example, in time-series visualization, it is common to identify a range in the time axis (implied by the data ordering) on which to focus one's attention [377]. Examining neighborhoods in a graph visualization often consists of filtering out nodes and links that are greater than a particular number of links away from a focus point, and many techniques for hierarchy visualization allow users to filter based on the level of the hierarchy.

An example of three-dimensional grid distortion is presented by Carpendale et al. [49]. They apply concepts from screen space distortion to elements with three spatial dimensions. Four classes of distortion are defined: stretch orthogonal, nonlinear orthogonal, nonlinear radial, and step orthogonal. To provide improved visibility to entities within the volume of data, they define a visual access distortion that shifts data to provide a clear line of sight to internal objects.

Distortion of hierarchies is a common practice due to the density of information that can result from broad or deep hierarchies. Several researchers have developed techniques based on radial hierarchy displays, such as Andrews and Heidegger [7], Stasko and Zhang [336], and Yang [409]. Other multiresolution techniques, such as wavelet transforms [404], have been used to visualize details in a focused region of an ordered list of data records.

In each of the cases above, it is the structure holding the data, rather than the data values themselves or the mechanism by which they are visualized, that is the focus of the distortion. Formalization of this procedure is somewhat more complicated than for the other spaces, but we can classify most of these distortions as mapping a vector (D, S), where D is the data and S is the structure holding the data, to (D', S'), where the transformation may modify the data, the structure, or both.

10.2.4 Attribute Space (Components of Graphical Entities)

Navigation in attribute space is similar to that in data value space; panning involves shifting the range of the values of interest, while zooming can be accomplished by either scaling the attributes or enlarging the range of values of interest. As in data value-driven selection, attribute-space selection requires the user to indicate the subrange of a given attribute of interest. For example, given a visual depiction of a color map, a user can select one or more
entries to highlight. Similarly, if data records have attributes such as quality or uncertainty, a visual representation of these attributes, accompanied by suitable interaction techniques, can allow users to filter or emphasize data according to the attributes. Remapping is often done in attribute space, either via selecting different ranges of an attribute to be used in the data to graphic mapping, or by choosing a different attribute to be controlled by the data. For example, in the GlyphMaker system [291], users could select mappings for a given data dimension from a list of possible graphical attributes. Many visualization tools provide an assortment of predefined color scales to be used for the visualization, some perceptually designed, others designed to be compatible with a particular application domain.

Given an attribute $A$ of a graphical entity being used to convey information, we can perform a distortion transformation by applying a function

![Figure 10.8.](image)

Attribute-based distortion modifies one or more attributes of the graphical objects used to depict the data, as shown in this colormap modification, generated using the colormap editor in OpenDX. The color map is distorted to allot a greater portion to values in the middle of the data range.
10.2. Interaction Operands and Spaces

We can assume $A$ can take on values in the range $[a_0 \rightarrow a_1]$, or that $A$ is specified as a vector. For example, distortion of a color map would allocate a wider or narrower range of colors for some subranges than for others, thus enabling fine variations to be more readily perceived (see Figure 10.8). This form of distortion is often used in medical image analysis to identify regions of interest. The size attribute of a data glyph or scatterplot marker, when not used to convey a data dimension, can also be distorted to emphasize or deemphasize selected subsets. Attribute-space techniques can be seen as complementary to data value space methods, since similar effects may be attained through either approach if one or more of the data dimensions is controlling the specified attribute.

10.2.5 Object Space (3D Surfaces)

In these displays, the data is mapped to a geometric object, and this object (or its projection) can undergo interactions and transformations. Navigation in object space often consists of moving around objects and observing the surfaces on which the data is mapped. The system should support global views of the object space as well as close-up views. The latter may be constrained to enable the user to find good views more quickly. Selection

![Figure 10.9.](image)

Object-based techniques distort an object upon which data has been projected. In this example, inspired by the perspective wall [245], a parallel coordinates display is projected onto walls, and perspective is used to make a selected wall more readable, while maintaining context with the rest of the data (from [379]).
10. Interaction Concepts

involves clicking anywhere on the object(s) of interest, or indicating target objects from a list.

A typical example of remapping in object space would be changing the object upon which the data is being mapped, such as switching the mapping of geographical data between a plane and a sphere.

For distortion, examples of this form of interaction are perspective walls [245] (Figure 10.9) and hyperbolic projections [261]. These methods can be envisioned as a variant on screen-based methods, where the object onto which the data is projected encapsulates the distortion function. However, after mapping, the surfaces can undergo additional transformations in 3D, such as rotation, scaling, and perspective distortion. For example, Kreuseler et al. [224] map hierarchies first to a hemisphere, and then adjust the focus by changing the center of projection, resulting in a distortion that enlarges one region while shrinking others. We can represent the process of object-space distortion as a sequence of two functions. The first maps the data (generally parameterized to two dimensions) onto a 3D structure \((x, y, z) = g(a, b))\), and then this structure is transformed and projected to the screen \((i, j) = h(x, y, z))\).

10.2.6 Visualization Structure Space

A visualization consists of a structure that is relatively independent of the values, attributes, and structure of data. For example, the grid within which a scatterplot matrix is drawn and the axes displayed in many types of visualization are each components of the visualization structure, and can be the focus of interactions.

Examples of navigation in visualization structure space might include moving through pages in a spreadsheet-style visualization tool or zooming in on an individual plot in a scatterplot matrix. For selection, typical operations would include choosing components to hide, move, or rearrange. For example, one might select an axis in parallel coordinates and drag it to a new location to discover different relationships among the data dimensions.

A good example of distortion in this space is the table lens technique [284, 347], which allows users to transform rows and/or columns of a spreadsheet to provide multiple levels of detail. See Figure 10.10 for an example of this process, as applied to a scatterplot matrix.
10.3. A Unified Framework

For each interaction operator to be applied to a specified space/operand, several parameters are required. Some of these may be constants for a given system. The parameters are described below.

**Focus.** The location within the space at the center of the area of user interest. There may be multiple simultaneous foci, though for navigation this usually requires multiple display windows.

**Extents.** The range within the space (can be multidimensional) defining the boundaries of the interaction. The metric used for specifying the range is specific to the space; in screen space this would be in pixels, while in structure-space this might be the number of rows in a table or links in a graph.
**Transformation.** The function applied to the entities within the extents, generally a function of distance or offset from the focus. The shape of this transformation might also depend on the type of information being affected. For example, text distortion is more likely to have a flat peak to the transformation function. Another component of the transformation is the degree or scale factor for the transformation, thus allowing varying amounts of the specified action.

**Blender.** How to handle parts of space touched by more than one interaction. For selection, this operation may include performing logical operations on overlapping entities [249]. For distortion, Keahey and Robertson identify several approaches, including weighted average, maximal value, and composition [189]. Each has advantages in terms of smoothness and ease of interpretation.

In Figure 10.11 we show a pipeline depicting the structure of the generalized distortion process (similar figures can be generated for other forms of interaction). At each stage, the user can control any or all of the operator parameters described above. While no system implemented to date supports all of these pipeline components, most visualization systems support one or more of them, allowing users interactive control over one or more of the operator parameters. It should be noted that the order in which the operations are applied may be modified, although the screen space method is most intuitively placed last. The order of operation presented in Figure 10.11 seems to the authors to progress in an intuitive, progressive fashion, but experiments are needed to verify this hypothesis.

![Figure 10.11](image-url) The distortion pipeline. The user interactively controls each stage of the pipeline. Each distortion operation is optional. (Image from [379], © 2004 IEEE.)
10.4 Summary

In this chapter we presented a framework for enveloping the wide assortment of interaction techniques developed to date for data and information visualization. By identifying the type of the operator (navigation, selection, manipulation, distortion, filtering) and the space of the interaction (screen, data value, data structure, attribute, object, or visualization structure), along with the parameters of the interaction operator (focus, extents, transformation, blender), we can define an extensive assortment of interaction operations. We also described a computational architecture to support interactions within the visualization pipeline.

Most visualization systems developed to date support, at most, a small set of interaction techniques. Future work should involve assessing user reactions to an environment containing a wider range of interaction operators. Questions to be addressed include:

- Given training in the use of individual interaction operations, how readily will users acquire expertise in composing interactions in different spaces?
- What combinations of operations will prove to be most effective, and in what situations?
- What are the best ways to provide users with unambiguous controls of the individual operations?

Initial experiments aimed at combining data value space and data structure space selection, navigation, and distortion within XmdvTool [119] have shown clear advantages to including all types of interaction; users are provided with many alternative ways of viewing and exploring their data sets, which can increase the likelihood of discovering features of interest. No problem in predicting the effects of the composition of operations has been discerned in this environment. We hope to expand this work into the other interaction spaces, and to attempt to answer the questions mentioned above, as well as others that arise during our investigations.

10.5 Related Readings

Several surveys and categorizations of interaction types in information visualization have been published, including the ones by Yi et al. [418] and Ward and Yang [379]. Individual types of interaction have also been the
focus of surveys, including Leung and Apperley’s excellent taxonomy of distortion techniques [232] and papers on selection by Wills [395] and Resnick et al. [289]. Most other papers we reference focus on a single interaction operator within a particular visualization technique, though many are readily generalizable to other spaces.

10.6 Exercises

1. Give three examples of distortions in two distinct spaces generating identical or very similar results.

2. Give an example of two specific distortions in different spaces that are commutative, i.e., the results do not depend on the order of application. Give an example where they are not commutative. Are there any general rules you can think of for identifying the conditions under which commutativity would hold or not hold?

3. In most situations, the user should be able to control the degree of distortion being applied. However, the initial amount should be set to some default level. Discuss how one might set defaults for different kinds of distortion. Consider techniques that are driven by characteristics of the data as well as those independent of the data.

4. Related to the question above, discuss strategies to set the default extents for different kinds of distortion.

5. Give examples of distortions with 0th-, 1st-, and 2nd-order continuity. For what reasons might the user choose a particular continuity level?

6. Select a visualization tool with which you are familiar and examine the types of interaction it supports. List the interaction operators and operands, as well as the parameters of the interaction that the user can control.

7. Continuing the previous exercise, identify some interaction operators and operands not present in the tool that you feel would be useful additions to the system. Give an example of how they might be used.

10.7 Projects

Programming projects dealing with interaction are included in the next chapter, which covers details of interaction techniques based on the concepts covered in this chapter.