EUROGRAPHICS 2003 / P. Brunet and D. Fellner
(Guest Editors)

Volume 22 (2003), Number 3

Interaction Spaces in Data and Information Visualization [†]

M. Ward and J. Yang

Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA USA

Abstract

User interaction plays an integral part in the effective visualization of data and information. Typical interaction operations include navigation, selection, and distortion. A problem that can occur when these operations are specified using direct manipulation is determining which object or space is the focus of the interaction. In some operations the user wants to indicate a region of an image, while in others the focus might be the data being projected or the surface upon which the projection is occurring. In this paper we attempt to identify a complete list of spaces within which interactive operations can occur in data and information visualization. These interaction spaces *help disambiguate the focus of interactive operations, and their study can potentially reveal new and powerful methods for supporting the visual exploration process. We define the distinctions between the spaces and provide examples of interactions within each space.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display Algorithms

---

1. Introduction

Interaction within the data and information visualization context is a mechanism for modifying what the users see and how they see it. In its basic form, navigation consists of panning and zooming allow the user to control the camera position and range of the view (what gets mapped to the screen). Selection is also a fundamental operation, enabling the user to indicate an object or region of interest to be the subject of some operation, such as highlighting, deleting, and modifying. Distortion is a common operation in the area of exploratory visualization; screen space for one or more focus areas are increased to enable users to see details, while showing the other areas of data in a smaller space in a way that preserves context.

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 may 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 paper, we propose such a framework for interaction techniques, identifying distinct classes and shared concepts that we hope 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.

2. Interaction Operators

In this section we attempt to categorize the wide range of interaction operations commonly found in data and information visualization. This list is probably not exhaustive, but represents a significant percentage of available tools.

2.1. Navigation Operators

Navigation is used to select the subset of data to be viewed, the orientation of this view, and the level of detail (LOD). In a typical N-dimensional space, this can be specified using a camera location, a viewing direction, the shape and size of

---

[†] This work was partially supported under NSF grant IIS-0119276.

the viewing frustrum, and an LOD indicator. In multiresolution visualizations, LOD changes can correspond to drilling down or rolling up a hierarchical representation 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 navigation is the Grand Tour [Asi85], where multidimensional data can be explored by flying along a path that smoothly covers all possible orientations of the data space as projected onto two dimensions. 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 navigation is Projection Pursuit [Hub85], where projections are computationally analyzed and the subset of views that exceed a user's threshold for "interestingness" are displayed.

2.2. Selection Operators

In selection, the user isolates a subset of the display components that 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 [Wil96], 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 the selection of the smallest addressable component (e.g., a vertex or edge) or target a broader region around the specified location (e.g., a surface, region of the screen, or object).

Selection can also be classified as to whether the user clicks on entities, paints over a selection of entities (e.g., holding the mouse button down while moving over the entities of interest), or otherwise isolating the entities via techniques such as bounding boxes and lassoes. Finally, selection can be performed in a semiautomatic manner, where the system selects elements that match a user's set of constraints. An example would be the selection of nodes in a graph that have a user-specified distance from a selected node.

2.3. 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 a method useful for interactive exploration. Many distortion functions (which we call *operators*) have been proposed in the past. 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 non-linear, 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*, i.e., 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 in nature. 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.

3. Interaction Operands and 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 be indicating? 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.

3.1. 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 arbitrary shaped regions that the user outlines. Selection areas may also be contiguous or non-contiguous.

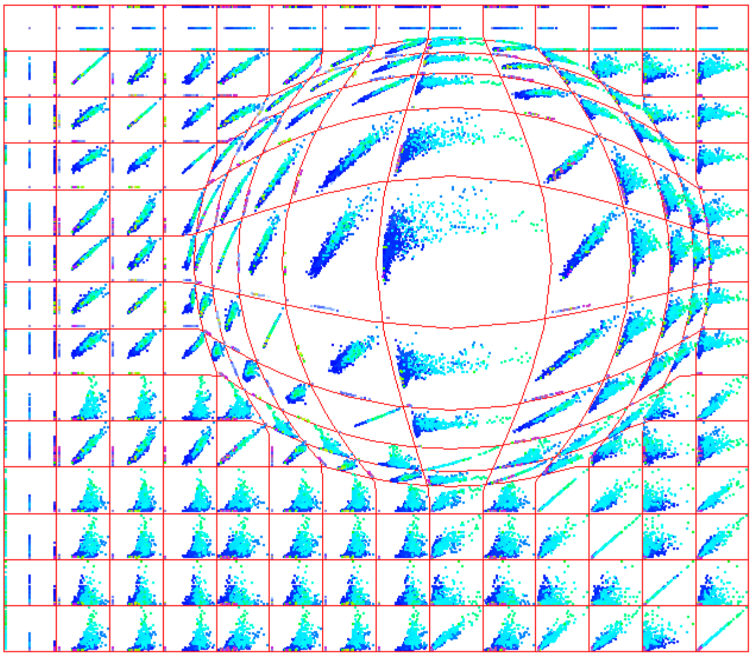

Figure 1: *In screen-space techniques, pixel regions are enlarged or reduced to provide selective detail. In this scatterplot matrix display, a center of focus has been selected and magnified using a confocal lens technique.*

3.2. Data Value-Space (multivariate data values)

Distortion in screen space involves transformations on pixels, i.e., $(x', y') = f(x, y)$. In order to avoid occlusion, this function should be order-preserving and at least C^0 continuous [KR97]. The magnification $m(x, y)$ at a particular point is simply the derivative of this transformation, and, in fact, it is useful to be able to switch between transformations and their associated magnifications when controlling the distortion process [KR97]. Examples of screen-space techniques are the fisheye lens [Fur86] and rubber sheet methods [SSTR93, CF95], although the latter techniques could also be placed in the object-space category described below. Figure 1 is an example of this type of distortion.

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 [MW95] (see Figure 2a) or via sliders or other query specification mechanisms [Shn94]. Selection may involve a single value or one or more ranges of values.

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 [FWR99], where each dimension of a selected subset of the data is scaled so that the subset fills the display area (see Figure 2).

3.3. Data Structure-Space (components of data organization)

Data can be structured in a number of ways, such as lists, tables, grids, and hierarchies. 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 3 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. [RWR98] selects subsets of data to be visualized by specifying a *focus*,

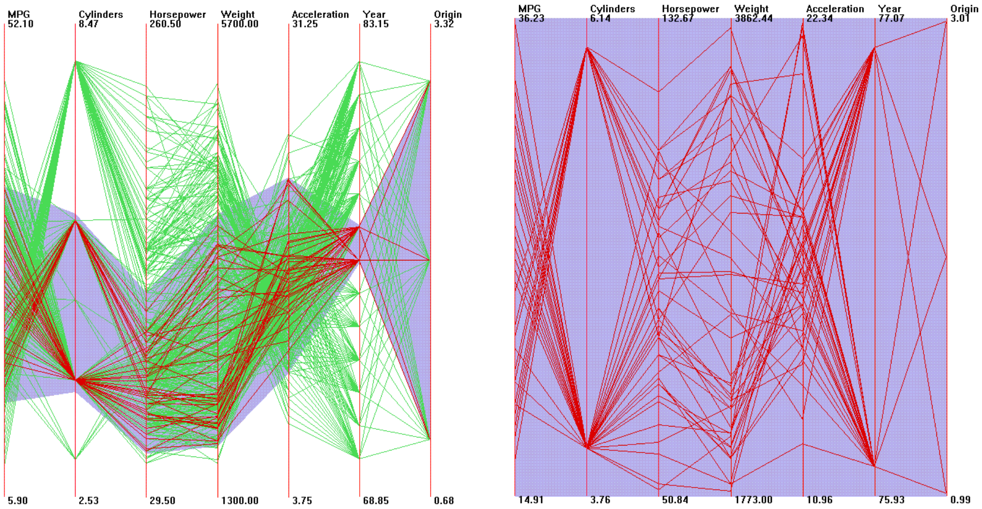

Figure 2: *In data value-space distortion, transformations are performed according to the dimensionality of the data. In this example, generated using XmdvTool [FWR99], an N-dimensional hyperbox is selected via painting over a section of an axis and scaled in all dimensions (by different amounts) to 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.*

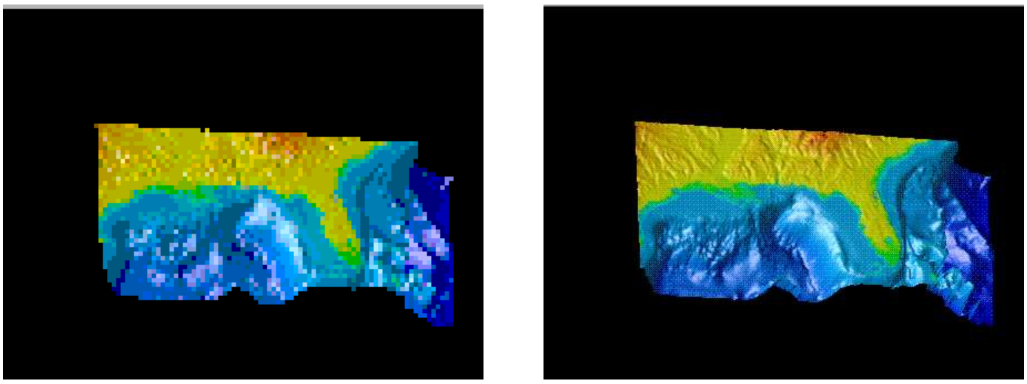

Figure 3: *In screen-space zooming (left), pixels are replicated to provide selective size, while in data space zooming (right), the data itself can be resampled at the appropriate resolution.*

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 in turn can drive the display of the data corresponding to the selected substructure. For example, structure-based brushing [FWR00] involves controlling the selection of data stored in a cluster hierarchy, with interactions such as highlighting data that falls 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 structure [YWR02]. Figure 4 shows a dimension hierarchy in InterRing with a subset of terminal nodes automatically selected via a query on their common parent node.

An example of 3-D grid distortion is presented by Carpendale et. al. [CCF97]. They apply concepts from screen-space distortion to elements with three spatial dimensions. Four classes of distortion are defied: stretch orthogonal, non-linear orthogonal, non-linear 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

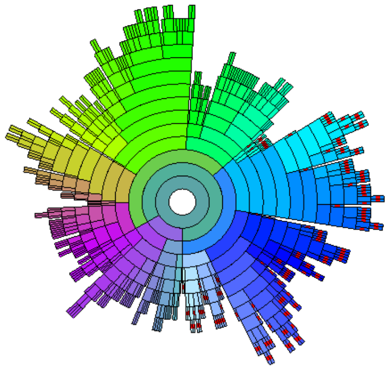

Figure 4: *Selection of nodes in a hierarchy via InterRing [YWR02]. Nodes with a red stripe in them have been selected via a user-specified query rather than one node at a time.*

based on radial hierarchy displays, such as the work of Andrews and Heidegger [AH98], Stasko and Zhang [SZ00], and Yang [YWR02]. Other multiresolution techniques, such as wavelet transforms [WB96], 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.

3.4. 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.

Given an attribute A of a graphical entity being used to convey information, we can perform a transformation by applying a function $k : a' = k(a)$. 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 others, thus enabling fine variations to be more readily perceived (see Figure 5). 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.

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

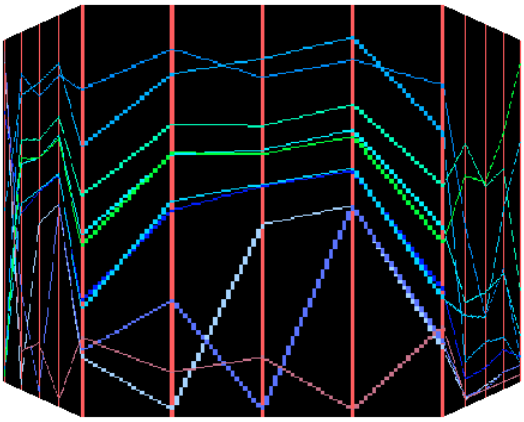

Figure 6: *Object-based techniques distort an object upon which data has been projected. In this example, inspired by the Perspective Wall [MRC91], a parallel coordinates display is projected onto walls, and perspective is used to make a selected wall more readable while maintaining context with the rest of the data.*

For distortion, examples of this form of interaction are perspective walls [MRC91] and hyperbolic projections [Mun97]. 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 3-D, such as rotation, scaling, and perspective distortion. For example, Kreuseler et. al. [KLS00] map hierarchies first to a hemisphere, and then adjust the focus

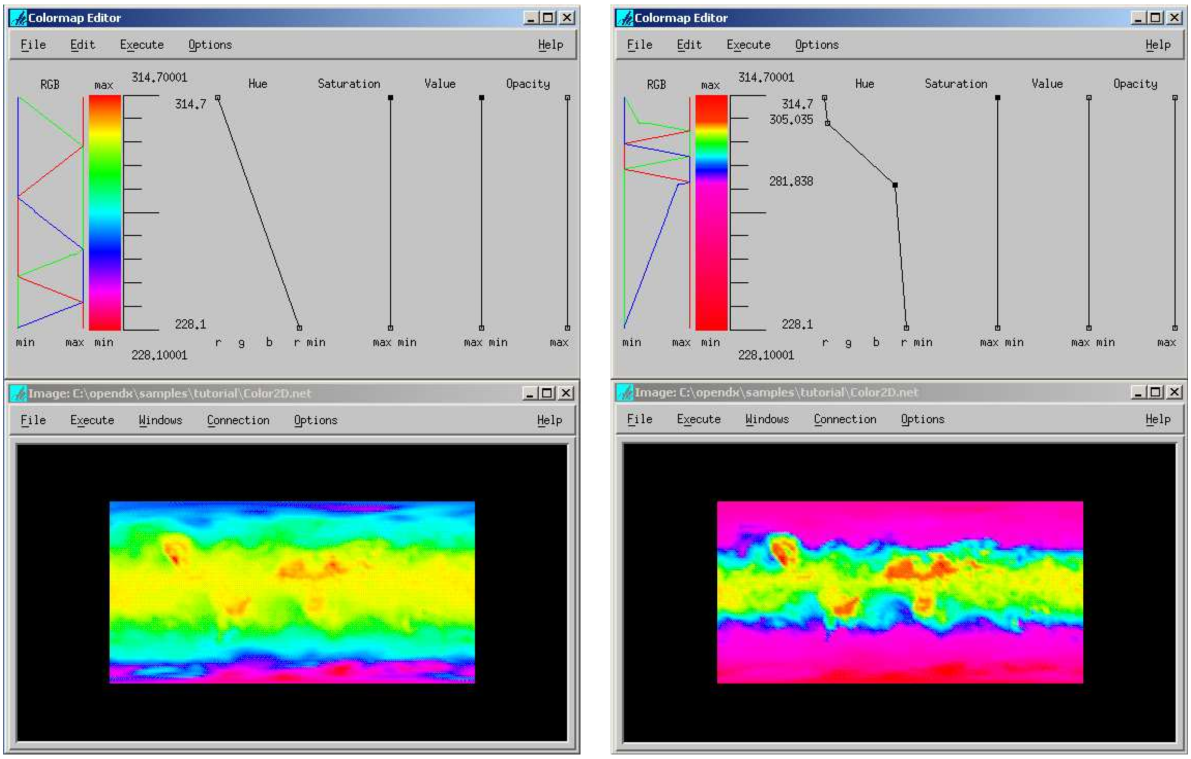

Figure 5: *Attribute-based distortion modifies one or more attributes of the graphical objects used to depict the data, as shown with this colormap modification, generated using the colormap editor in OpenDX. The color map is distorted to allot a greater portion to values in the middle of the data range.*

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 3-D structure ($(x,y,z) = g(a,b)$), and then this structure is transformed and projected to the screen ($(i,j) = h(x,y,z)$) (see Figure 6).

3.6. 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 or the axes displayed in many types of visualizations 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 [RC94, TR97], which allows users to transform rows and/or columns of a spreadsheet to provide multiple levels of detail. See Figure 7 for an example of this process as applied to a scatterplot matrix.

4. 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

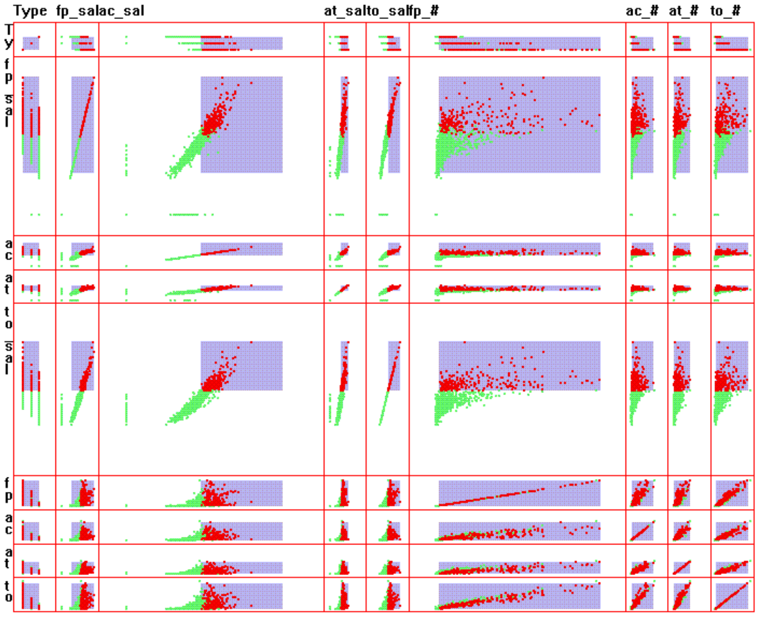

Figure 7: *Structure-based distortion modifies the underlying structural elements of the visualization. This example, inspired by Table Lens [RC94], shows a scatterplot matrix with two grid cells (and their corresponding rows and columns) magnified, with a corresponding shrinkage in other cells.*

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, while for other types of information this constraint might not be desirable. 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 union, intersection, or other logical operations on overlapping entities [MW95]. For distortion, Keahey and Robertson identify several approaches, including weighted average, maximal value, and composition [KR96]. Each has advantages in terms of smoothness and ease of interpretation.

In Figure 8 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 information 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 8 seems to the authors to progress in an intuitive, progressive fashion, but experiments are needed to verify this hypothesis.

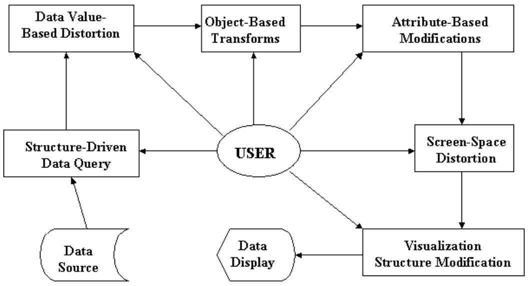

Figure 8: *The Distortion Pipeline. The user interactively controls each stage of the pipeline. Each distortion operation is optional.*

5. Interaction Control

At each stage of the pipeline introduced in the previous section, the user requires mechanisms to control the type, location, and level of each interaction as he or she navigates within both the data space and the visualization. The realization of these controls must be intuitive, unambiguous, at a level of detail and accuracy appropriate for the space being operated upon. In particular, the following lists typical controls and reasonable candidates for their implementation:

Focus Selection: Selection is most readily accomplished via direct manipulation tools, i.e., using a mouse or other selection device to indicate the focus location. In screen and object space, this can be easily accomplished via normal selection operations. In data space, an N-dimensional location might need to be indicated. Depending on the method of display, this could involve multiple selections (e.g., selecting in a scatterplot matrix only enables simultaneous specification of two dimensions). In attribute and structure space, one first needs a graphical depiction of the structure or the range of the attribute, such as a display of a tree or table or a curve showing the range of colors in the color map. Finally, the focus can be specified implicitly, by assuming the focus is the center of the extents of the interaction, which can be specified as outlined below.

Extent Selection: Specifying the extents for an interaction is generally dependent on the type of interaction and the space in which the interaction is being applied, and can be done either via direct manipulation or separate interface tools. It may be specified via a single value (e.g., a radius or maximum number of items) or via a vector of values (e.g., a range for each data dimension or a set of constraints). In many systems the extents are often hardcoded to reduce the effort in performing the operation.

Interaction Type Selection: Given the many types of interaction possible, and the variety of spaces in which they may be applied, a reasonable interface for this task would be a pair of menus: one to select the space and the other to specify the general class of the interaction.

Interaction Level Selection: The degree of interaction is an important control parameter that can be specified by a single value (e.g., the magnitude of scaling that will occur at the focal point). A slider or dial is sufficient for this activity, along with a button to reset the operation to its minimum level. A direct manipulation equivalent would be to associate upward mouse motions with an increased interaction level, perhaps in conjunction with direct manipulation of the extents via horizontal mouse motions.

Blender Type Selection: If more than one interaction can be simultaneously viewed and manipulated, there must be some mechanism for selecting a strategy for mixing regions of space affected by more than one interaction. As with Interaction Type Selection, this is best accomplished via a menu of options. Available options might be dependent on both the space in which the interaction is occurring and the type of interaction being used. As interactions in different spaces are applied at different points in the pipeline, it is unnecessary (or at least, too complicated) to consider methods for controlling the combination of interactions involving two or more spaces.

An important feature that should be present in all operations is the animation of interpolated values of the interaction parameters as they are changed. This has been shown to be extremely effective in many implementations of operators for helping users to both preserve context and obtain a better understanding of the effects of the operation on the data [vWN03]. Rapid changes can lead to confusion and a loss of orientation, especially when interactively exploring large data or information repositories. Related to this, users should have some control over the rate of this animation (the number of frames or steps in the interpolation).

6. Conclusions

In this paper 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, distortion) 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, magnitude, blender), we can define an extensive assortment of interaction operations. We also described a computational architecture to support interactions within the visualization pipeline and suggested interface tools for enabling the user to control each of the components.

Most visualization systems developed to date support, at most, a small set of interaction techniques. Part of our future work will 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?

Our initial experiments at combining data value-space and data structure-space selection, navigation, and distortion within XmdvTool have shown clear advantages to including all types of interaction; the user is provided with many alternative ways of viewing and exploring their data sets, which can increase the likelihood of discovering features of interest. We also found that there is no problem in predicting the effects of the composition of operations. We hope to expand this work into the other interaction spaces and attempt to answer the questions mentioned above, as well as others that arise during our investigations.

Acknowledgments

The authors would like to thank the members of the XmdvTool group (past and present) for their help in developing the ideas that are presented in this paper. We'd also like to thank the NSF and NSA for their support of this research.

References

[AH98] ANDREWS K., HEIDEGGER H.: Information slices: visualizing and exploring large hierarchies using cascading, semicircular discs. *Proc. Information Visualization '98 Late Breaking Hot Topics* (1998), 9–12.

[Asi85] ASIMOV D.: The grand tour: A tool for viewing multidimensional data. *SIAM Journal of Scientific and Statistical Computing 6* (1985).

[CCF97] CARPENDALE M., COWPERTHWAITE D., FRACCHIA F.: Extending distortion viewing from 2d to 3d. *IEEE Computer Graphics and Applications 17* (1997), 42–51.

[CF95] COWPERTHWAITE M. C. D., FRACCHIA F.: 3-dimensional pliable surfaces: for the effective presentation of visual information. *Proc. UIST '95* (1995), 217–226.

[Fur86] FURNAS G.: Generalized fisheye views. *Proc. Computer-Human Interaction '86* (1986), 16–23.

[FWR99] FUA Y., WARD M. O., RUNDENSTEINER E. A.: Hierarchical parallel coordinates for exploration of large datasets. *Proc. Visualization '99* (1999), 43–50.

[FWR00] FUA Y., WARD M. O., RUNDENSTEINER E. A.: Structure-based brushes: a mechanism for navigating hierarchically organized data and information spaces. *IEEE Trans. Visualization and Computer Graphics 6* (2000), 150–159.

[Hub85] HUBER P.: Projection pursuit. *Annals of Statistics 13* (1985), 435–475.

[KLS00] KREUSELER M., LOPEZ N., SCHUMANN H.: A scalable framework for information visualization. *Proc. Information Visualization 2000* (2000), 27–36.

[KR96] KEAHEY T., ROBERTSON E.: Techniques for nonlinear magnification transformations. *Proc. Information Visualization '96* (1996), 38–45.

[KR97] KEAHEY T., ROBERTSON E.: Nonlinear magnification fields. *Proc. Information Visualization '97* (1997), 51–58.

[MRC91] MACKINLAY J., ROBERTSON G., CARD S.: The perspective wall: detail and context smoothly integrated. *Proc. ACM CHI* (1991), 173–179.

[Mun97] MUNZNER T.: H3: laying out large directed graphs in 3d hyperbolic space. *Proc. Information Visualization '97* (1997), 2–10.

[MW95] MARTIN A., WARD M.: High dimensional brushing for interactive exploration of multivariate data. *Proc. Visualization '95* (1995), 271–8.

[RC94] RAO R., CARD S.: The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proc. ACM SIGCHI* (1994), 318–322.

[RWR98] RESNICK R. J., WARD M. O., RUNDENSTEINER E. A.: Fed: a framework for iterative data selection in exploratory visualization. *Proc. Int'l Conf. Scientific and Statistical Database Management* (1998), 180–189.

[Shn94] SHNEIDERMAN B.: Dynamic queries for visual information seeking. *IEEE Software 11* (1994), 70–77.

[SSTR93] SARKAR M., SNIBBE S., TVERSKY O., REISS S.: Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. *Proc. UIST '93* (1993), 81–91.

[SZ00] STASKO J., ZHANG E.: focus + context display and navigation techniques for enhancing radial, space-filling hierarchy visualization. *Proc. Information Visualization 2000* (2000), 57–65.

[TR97] TENEV T., RAO R.: Managing multiple focal levels in table lens. *Proc. Information Visualization '97* (1997), 59–63.

[vWN03] VAN WIJK J. J., NUIJ W.: Smooth and efficient zooming and panning. *Proc. Information Visualization '03* (2003).

[WB96] WONG P., BERGERON R.: Multiresolution multidimensional wavelet brushing. *Proc. Visualization '96* (1996), 141–148.

[Wil96] WILLS G.: Selection: 524,288 ways to say 'this is interesting'. *Proc. Information Visualization '96* (1996), 54–60.

[YWR02] YANG J., WARD M. O., RUNDENSTEINER E. A.: Interring: An interactive tool for visually navigating and manipulating hierarchical structures. *Proc. Information Visualization '02* (2002), 77–84.