

SDM: Malleable Information Graphics

Mei C. Chuah, Steven F. Roth, Joe Mattis, John Kolojchick
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
Tel: +1-412-268-2145
E-mail: {mei+,steven.roth}@cs.cmu.edu

Abstract

Selective Dynamic Manipulation (SDM) is a paradigm for interacting with objects in visualizations. Its methods offer a high degree of selectivity, in choosing object sets, in the selection of interactive techniques and the properties they affect, and in the degree to which a user action affects the visualization. Our goal is to provide a flexible set of techniques and feedback mechanisms that enable users to move objects and transform their appearance to perform a variety of information analysis tasks.

Keywords: Interactive techniques, visualization, direct manipulation.

1 Introduction

SDM is a paradigm for interacting with objects in visualizations in a highly selective and flexible way. SDM is complementary to other interactive visualization techniques, most of which adhere to either a *spatial* or *object* metaphor. These metaphors differ in their: method of control, targets of control, method of maintaining context, control operations, and scope of operation. SDM mostly uses an object metaphor though it uses a mixed metaphor to maintain context and in its implementation of control operations.

As suggested by its name, SDM is an effort to provide users with a high degree of selectivity and object dynamics while manipulating visualizations to analyze and explore diverse information. The goal is to give users freedom to manipulate 2D and 3D objects while providing constraints and feedback to prevent confusion. In support of this, SDM allows flexible control of any object parameter.

For example, in Figure 1, many of the selected objects (green in the color plate) are occluded and it is difficult to analyze them as a set. Using SDM, selected objects can be elevated over the scene, thus eliminating occlusion while maintaining context, as in Figure 2.

Furthermore, SDM operations can be combined to solve problems in different ways. For example, small objects may be elevated, expanded and/or scaled up (Figure 7) so that their height differences can be perceived. Similarly, occluded objects can be emphasized by expanding them, or alternatively by reducing the widths and/or heights of all other objects, as in Figure 4.

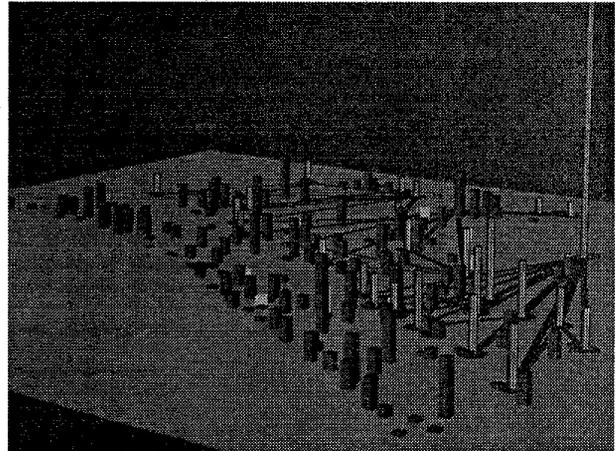


Figure 1: Initial view of occluded objects. (See color plate.)

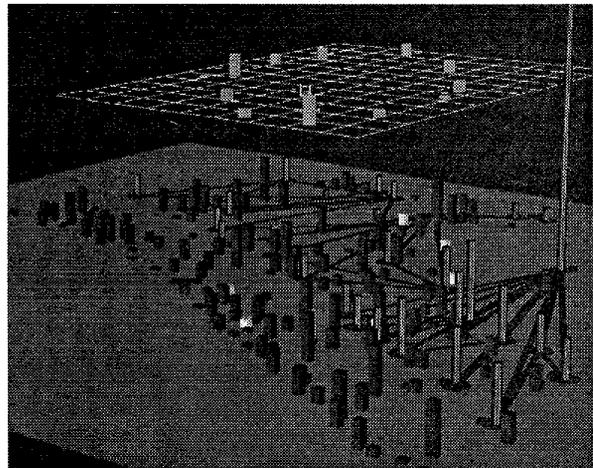


Figure 2: Elevating objects helps occlusion problem. (See color plate.)

While the flexibility offered by free-form manipulation is desirable, unrestricted changes can cause one to lose track of the current state of altered objects and their context. Thus, in addition to *object operations*, we developed a small set of *constraints* and *feedback techniques* to help reduce confusion and maintain context. We first describe these components and then outline a framework

for viewing SDM and other interactive visualization approaches.

The data domain for our examples is from a supply distribution network for a relief effort in a large-scale crisis. In Figure 1, supply centers are represented by cylinders (red in the color plate), main routes among them by dark lines on the floor plane, and shelters where supplies are needed by rectangular bars. The heights of cylinders and bars indicate the quantities of material available at supply centers and needed by shelters. The current SDM implementation was developed in GL and C on a Silicon Graphics Indigo workstation.

2 SDM operations

There are three different sets of operations in SDM: shift, scale, and surface-appearance (e.g. color, brightness) operations. These can be applied in combination and in different sequences but are described individually next.

Shift operations. These techniques alter values of one or more quantitative object parameters by a constant. Shift operations can be used to focus on selected objects while maintaining scene context, and to view occluded objects, by shifting selected objects horizontally or vertically, as in Figures 2 and 3.

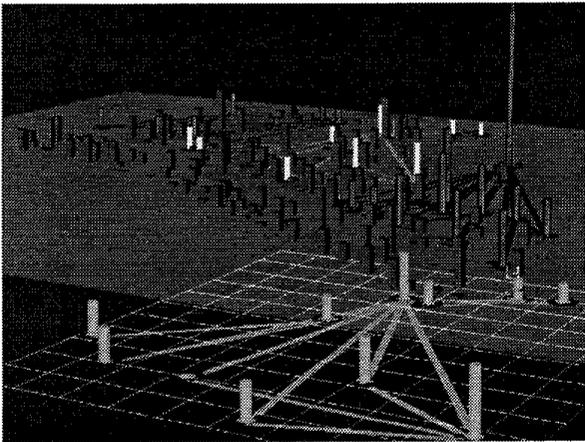


Figure 3: Focus objects are pulled to the front. (See color plate.)

Alternatively, the widths of selected objects may be expanded by a constant amount so that they are larger and more noticeable; similarly, unselected objects can be reduced and made thin, as in Figure 4. It is important to note that shift operations do not maintain relative size or height ratios among objects. Thus, changing object parameters that are tied to particular data attribute values would mean that we can no longer compare the relative heights or sizes of the altered objects, although simple differences can still be observed. In our dataset, the widths of objects are not used to show any data attribute, so no information is lost.

Shift operations are also useful for dividing up subsets of objects in the visualization. In Figure 3, a subset of the



Figure 4: Non-selected objects are reduced to be very thin. (See color plate.)

network is shifted forward to examine it separately. It is also possible to create multiple subsets and move each to another region either above or adjacent to the rest. Spatial shift operations maintain relative positions, and thus can be manipulated separately to view patterns within them.

Shift operations can also be used to classify sets of objects by changing their size or shape so that they are distinct from others. For example, in Figure 5, the selected bars are differentiated first by their color and then further distinguished by elongating their shape. Users can define and change their own groupings or classifications throughout the data analysis process.



Figure 5: Selected bars are differentiated by interactively changing their shape during data analysis. (See color plate.)

Scale operations. These operations include all techniques that scale one or more parameters of an object set. Unlike shift operations, scale operations maintain size and height ratios among selected objects. Thus when the size or heights of objects in the visualization encode a data attribute, scale operations can be used instead of shift operations to expand or contract the objects. This is so that the relative size or heights of objects correspond to the relative values of the data-attribute that they encode.

Scale operations can also be used to put different object sets on different size or height scales. It is often the case that the sizes and heights of objects are diverse. In these cases, the overall scale will dwarf certain objects and the height and size relationships among them. Although logarithmic scales are sometimes used effectively to solve this problem, these obscure simple linear relations and judgments and may be non-intuitive to many users.

In order to maintain size or height ratios among objects in the selected set, and at the same time magnify the size or height differences, SDM enables users to view the set at a scale that is different from other objects. For example, to peruse all shelters with low demand and see whether there are groups that are close together, a user first selects all shelters with low demand using a slider and then scales their heights up. Figures 6 and 7 show the effects of these operations. The ratio axes on the left show that the selected set is scaled seven times larger than the rest.



Figure 6: Objects in selected set are too small to make any height comparisons.



Figure 7: The same dataset as Figure 6 with light gray bars scaled differently from the rest of the environment.

Scale operations can be used to compare objects more easily. Comparisons of object heights and sizes is sometimes hard to make when 3D graphical objects are at

different depths in a scene or when 2D objects are not aligned (e.g. placed on a map). In SDM, users sketch a line on the x,y plane and drag all selected objects toward it (Figure 8). This is achieved by linking the positions of objects with a formula that calculates their perpendicular distance to the reference line. Users scale these distances interactively to animate object movement toward or away from the line. Multiple sets can also be moved to the same line, e.g. to compare variability of green and blue bars, as in Figure 9. Other parameters can be linked and manipulated together using the same formula mechanism.

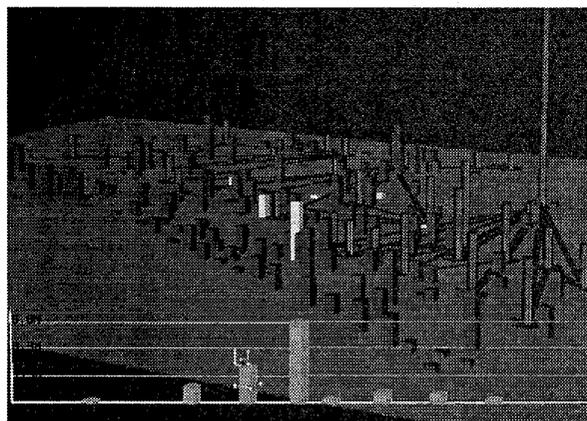


Figure 8: Selected objects moved to a reference line in order to make height comparisons.



Figure 9: Dragging object sets to a reference line enables height comparisons, which were difficult among the previously non-aligned objects. (See color plate.)

Surface-appearance operations. These are operations that change objects' non-geometric properties, such as color, texture, intensity, and opacity. These are used to focus attention on subsets (e.g. by differentiating their color or making them blink), to solve occlusion problems (e.g. by controlling translucency), and to classify objects.

In this section we have presented a set of operations that are indicative of a style of interaction and which is meant to augment current interactive visualization approaches. This set of techniques is by no means complete. We envision that as users work with these operations, they will start to combine them in new ways to solve their tasks.

3 SDM constraints and feedback

SDM is an attempt to create a physics of objects (as in [3]) that provides malleable representations and flexible control. However, free-form movement and control without any constraints can lead to confusion and misinterpretation of the data. Thus we have developed a small set of constraints together with a suite of feedback techniques to help reduce confusion while still maintaining a high degree of flexibility.

One important constraint in SDM is that all objects in a selected set are subject to the same forces applied to any single object in the set. This is useful because it helps base SDM techniques on several real-world analogies. SDM shift operations are analogous to putting all selected objects on a platform and then moving the platform. SDM scaling (e.g. height scaling) operations are analogous to pulling a malleable object or a set of malleable objects from the top or side, with the same force. These maintain relative positions, sizes, and heights of all selected objects. The ability to change multiple objects by manipulating one is a common feature of draw programs as well.

Context persistence is another important goal in SDM and is maintained either inherently or through feedback techniques. Mechanisms exist for restoring context when needed. We've developed the following feedback techniques:

Interactive animation. Interactive animation provides temporal continuity so that users can perceive, track and undo the effects of an operation. Animation enables users to track changes to the scene without having to cognitively reassimilate relationships between the pre-action and post-action scenes[11]. This shift of cognitive load to the perceptual system allows the user to concentrate more on the results of an action rather than the process of the action. In addition, it also provides additional feedback as to which objects have changed and how they have changed relative to their previous states. User tests in [7] showed that insufficient temporal continuity caused poor user performance.

Shell metaphor. When objects are moved or distorted, users still need to view their original states in order to maintain context. To achieve this, each data object is represented by two graphical representations: the 'body' and 'shell' graphical objects. Object shells are displayed in white and are left behind in the original position when the object bodies are displaced. The bodies can be moved through other objects and their appearance can be modified. Shells are static and rigid.

Interactive and dynamic operations. SDM operations are interactive and dynamic, thus objects may be scaled or shifted away from or back towards their positions to get spatial context. For example, in Figure 2, objects are elevated above their original positions. In order to get spatial context users may move the objects up and down, between their elevated positions and their home points. Similarly, in Figure 3, objects may be moved back and forth between their positions at the front and their home positions. In Figure 8, objects have been moved to the reference line for height comparisons; not only have they been displaced from their home positions, but their relative positions to each other is lost. Users may regain both these properties by sliding the objects back and forth between the reference line and their home points. Similar dynamic control is provided for size, height, and other parameters.

Home mechanism. This mechanism automatically returns objects to their original state (position, shape, size, height and color) using continuous animation. This is used to quickly reset the visualization after a task is complete. We are also developing "homing" interfaces for restoring parameters individually and for undoing one or more operations. Of course, it is possible to directly restore objects to their "shells" using shift and scale operations.

Object bumpers. It is sometimes difficult to slide objects back and forth between their displaced positions and their home points (and sizes). Moving objects and changing their sizes precisely back to their home points is difficult because it is easy to overshoot. Thus we are exploring the use of object bumpers. With object bumpers, the initial object move will indicate the direction of freedom and from then on objects can only be moved back as far as their home positions. An alternative method is to use snap-dragging.

Painting object bodies and shells. One way users can view the link between bodies and their shells is by moving the bodies back and forth into their shells. In addition, users may choose colors to paint bodies and see the corresponding shells change to the same color. Painting is also used to define selections.

Ratio axes. Ratio axes display the scale of the environment and the scale of the current selected set (Figures 6 and 7). By comparing the lengths of the axes, one can tell the difference in scales between the selected set and the environment. For example, from the ratio axes in Figure 7, we can tell that the selected set is scaled to be about seven times taller than the rest of the environment.

Object set grid. A white grid may be drawn beneath all selected objects (Figures 2 and 3), so that the approximate positions and spread of its selected elements can be perceived. Thus, the objects that will change in response to a user action are clearly identified.

Information window and text. Users obtain additional information associated with graphical objects in a text window. This is one way to obtain precise data values. Alternatively, users may attach text strings to any graphical object or set of graphical objects in order to show the values of an associated data attribute.

4 Interactive metaphors

A lot of research has been directed towards developing interactive metaphors for visualizing and interactively manipulating information to perform analysis tasks. In order to convey some of the common themes and differences, we view these techniques as following two metaphors: a *spatial* metaphor and/or an *object* metaphor.

Techniques that mostly use a spatial metaphor are spatial distortion techniques such as the fisheye-views [4], bifocal displays [9], stretching [14], and hyperbolic geometry [8]. These techniques provide the ability to display more information than can normally fit on a computer screen by distorting parts of the information space that are not currently under consideration. Techniques such as dynamic queries[1], and brushing/painting[13] use an object metaphor. In these techniques, users control the visibility and appearance of objects based on values of data they convey.

An object/spatial distinction is too simple to convey all the differences in approaches. Many approaches, including SDM, use elements of both metaphors. Therefore, we will enumerate some of the underlying features:

Method of control. This refers to how users control changes in the visualization. In a spatial metaphor, users control their views of objects by operating on a lens or by defining point(s) of focus, as in [12,16]. Object metaphors give users control of the objects themselves instead of an explicit or implicit external object (such as a lens).

Targets of control. This refers to the type of target entities that a user can control. Spatial metaphors allow users to control visualization areas. Areas may be selected by scrolling, choosing focus points, or marking an enclosed polygonal space. In the object metaphor, the targets of control are individual or sets of graphical objects in the visualization. Objects may be selected by direct manipulation or through constraints on their underlying data (e.g., using dynamic query sliders).

Controlling areas instead of objects works well on tasks that are very strongly based on object spatial position but less well when objects are examined based on non-spatial properties and do not occur in rectangular, contiguous regions. In contrast, controlling objects directly enables users to perform spatially based tasks by collecting all the graphical elements in the desired space (either through their underlying geographical data or by selecting them from the visualization) and forming a data set. Operating on that set is effectively the same as operating on the area that contains the set. A disadvantage of

controlling graphical objects, however, is that users cannot operate on parts of objects. For example, enlarging a portion of a road would mean that the whole road has to be enlarged and not just part of it. For this reason, we have combined zoom and pan techniques (which are spatial metaphors of control) with object techniques within SDM.

Method of maintaining context. While it is often necessary for users to focus on subsets of information, it is important to track the relation between the changing focus and the full set. In spatial metaphors context is achieved by distorting sections that are not in focus, so that even though the focus area has been expanded, the entire information space may still fit into a limited 2D screen[4,8,14]. Alternatively, the sections that are not in focus may be left undistorted, but either partially occluded [16] or temporarily displaced [3] by the expanded focus region.

Object metaphors do not alter any spatial properties, instead they allow users to focus on data points and maintain context by changing object color or appearance. Context objects are either temporarily invisible[1] or uncolored [13]. Users view context by making the invisible points reappear or by viewing uncolored points. [10] also uses the object metaphor to maintain context but it does this by making the focus region partly transparent.

Control operations. Spatial metaphors allow users to change the spatial properties of the target(s) under control. [4,10] allow users to expand the selected space. [14] allows users to both expand and contract the selected space. In the object metaphor, users change the surface-appearance of the target(s). In [13], users may change the color of graphical objects in the visualization and in [1] users may change the visibility of objects.

Scope of operation. Scope of operation refers to the objects affected by a user action. In the spatial metaphor, the scope of operation is unlimited. A user action affects the selected target(s) and may affect the rest of the visualization as well. Examples are fisheye lenses[4] and stretching[14]. In the object metaphor, the scope of operation is limited to the selected set. Examples include painting[13] and dynamic query sliders[1]. One disadvantage of unlimited scope is that local actions affect objects in the entire visualization and create unstable views. [7] reported that subjects were disoriented by this behavior when using fisheye lenses. A disadvantage of limited scope though is that it is more difficult to maintain scene context when there are large amounts of data.

Table 1 shows our initial attempt to describe some existing approaches along these dimensions. From Table 1, we can see that fisheye lenses[4], bifocal lenses[9], and PAD++[3] are purely based on the spatial metaphor. Painting[13] and dynamic query sliders[1], on the other hand, are purely based on the object metaphor. Most of the other techniques however have mixed metaphors. Several systems use both object and spatial metaphors for the same function. For example, [16] allows the use of both

	Method of control	Targets of control	Context Method	Control operations	Scope of operation
Fisheye[4],Bifocal-lens[9],PAD++[3]	Spatial	Spatial	Spatial	Spatial	Spatial
Stretching[14]	Object	Spatial	Spatial	Spatial	Spatial
Powers of 10,000 [10]	Spatial	Spatial	Object	Spatial	Object
Magic Lenses[16]	Spatial	Both	Both	Both	Object
Painting[2,13],Dynamic Query[1]	Object	Object	Object	Object	Object
SDM	Object	Object	Both	Both	Object

Table 1: Classification of a set of interactive systems.

metaphors in its control operations and to maintain scene context. In fact, many of the schemes used in the two different metaphors are complementary and could be used together. The appropriate interactive metaphor to use depends on the data visualization, the data, the task and to some extent user expertise and preferences.

5 SDM approach

In this section, we will describe SDM in terms of the five properties we outlined in the previous section.

Method of control (Object metaphor). Users control visualizations using 3D handles that may be attached to objects. Attaching a handle to an object in a selected set, and pulling or pushing its parts causes some or all objects in the selected set to change. When handles are pulled or pushed, the objects contract, expand or move continuously. Use of interactive animation in this way helps users understand and track the changes that occur to an object set. The handles in Figure 10 were designed to look similar for different object types so that users need not recall the functions of many visually different interfaces.

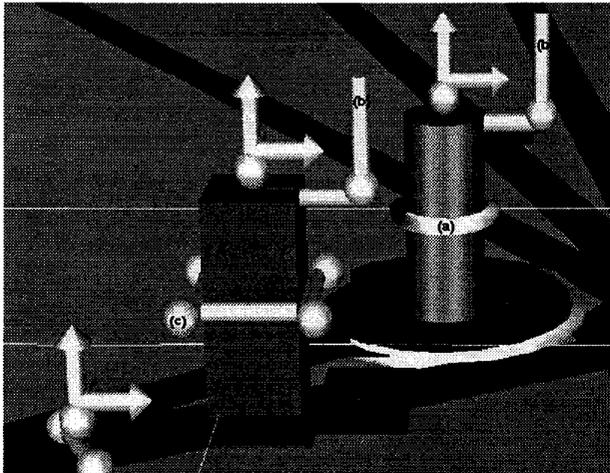


Figure 10: SDM handles.

Our use of handles was motivated by related work on 3D widgets [6,15] and by 2D scaling handles in popular draw programs. Handles provide control for selecting which

object parameters to change. For example, in Figure 10, part (a) controls the radius of the cylindrical object, part (b) controls the height of the object, and part (c) controls the width of the bar. Each arrow handle indicates a single direction of movement for the object. Parameters can also be combined and controlled through one handle.

Targets of control (Object metaphor). In SDM, users control objects and not spaces. Selection of graphical objects can be accomplished by clicking on them, by using sliders to define data ranges for selected objects, or by using a combination of the two. The slider interface in SDM allows users to combine data constraints both conjunctively and disjunctively. In the future, we plan to generalize the selection facility so that there can be multiple linked displays. Selecting elements in one display would cause all related graphical elements in other displays to be selected as well [2].

After object sets are declared, they may be saved. Sets that are created will appear in a scroll menu that is used to select them for further manipulation. Users may switch between multiple self-defined sets easily. Note also that the object sets need not be made up of homogeneous elements (e.g. a set may contain supply centers, shelters and routes).

Methods of maintaining context (Mixed metaphor). Context is maintained in all SDM operations previously mentioned. Context persistence is either inherent in the techniques (e.g. painting, expanding objects in-place) or maintained through the feedback techniques (axes, body-shell pairs).

Control operations (Mixed metaphor). As was previously mentioned, there are three primary methods of control: shift techniques, scale techniques, and surface-appearance techniques. In general, SDM allows users to transform object positions and change object appearance. For example, objects may be elevated (Figure 2) or moved on the floor plane (Figure 3). Objects may be made thinner (Figure 4) or taller (Figure 7).

Scope of operation (Object metaphor). SDM has a limited scope of operation. User manipulation only affects objects in the selected set. Changing the size and position of some objects can cause them to collide with or occlude other objects, but the latter are not distorted.

In summary, SDM is primarily based on the object metaphor. However, like [16] it uses techniques from both metaphors in its control operations and to maintain context. In addition to being strongly object-oriented, our approach is to provide a flexible and general object manipulation paradigm. Even though some of the individual techniques in SDM have been tried before, they have only been tried in isolation. We are trying to create a unifying paradigm and a suite of techniques that can be combined flexibly to solve different data analysis tasks.

Many interactive techniques only allow users to distort the information space in one limited way. Fisheye lenses[4] and stretching[14] allow users to increase the area of the focus region by reducing the area of surrounding regions. Painting[13] allows users to focus on the selected set by changing its color.

An exception to this are magic lenses[16], which allow different distortions depending on the lens that is being used. Like SDM, that research defines a unifying paradigm, derives a tool set from it, and provides composition of multiple operations. SDM operations are designed more to enable users to directly manipulate object appearance as they would in a draw program. Lenses provide metaphors for viewing regions containing objects in different perspectives and can be general purpose (e.g. to distinguish occluded objects) or application-specific (e.g. to view semantically related properties of objects). Lenses do not address parameter manipulation per se. Our implementation of zoom, pan, and related operations for moving through visualizations suggests that SDM and lenses can be combined in a highly complementary way.

6 Conclusion and future work

We have presented a suite of techniques based on the SDM paradigm. These techniques enable users to solve a variety of tasks including:

- viewing selected objects in detail while keeping them in context with the rest of the environment.
- viewing occluded objects by elevating them or by reducing the size and heights of surrounding objects.
- viewing different object sets in different scales.
- interactively classifying different object sets.
- comparing the size and heights of objects even though they may be positioned far from each other.

Many current interactive techniques tend to examine and present solutions to isolated problems. The goal of SDM, however, is to provide users with enough tools and flexibility so that they can solve a wide spectrum of data analysis tasks. We believe that the unifying framework behind the techniques will make the system easy to learn, and the flexibility in using the methods will stimulate users to come up with multiple, novel ways to solve tasks.

Although we have selected examples using rectangular objects and cylinders, the same techniques can be used for other graphical objects as well. We are also integrating current SDM operations with other aggregation and other

interactive data manipulation interface techniques[5]. We are also exploring the creation of multiple linked views.

Acknowledgments

Funding for this project was provided by the Advanced Research Projects Agency (DAAA1593K0005) and the Army Research Laboratory.

References

1. Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic Queries for Information Exploration: An Implementation and Evaluation. *Proceedings CHI'92 Human Factors in Computing Systems*, ACM, May 1992, p.619-626.
2. Becker, R.A., Eick S.G., and Wilks, A.R. Visualizing Network Data. *IEEE Transactions on Visualization and Graphics*, March 1995, p.16-28.
3. Bederson, B.B. and Hollan, J.D. PAD++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *UIST '94*, November 1994, p17-27.
4. Furnas, G.W. Generalized fisheye views. *Proceedings CHI'91 Human Factors in Computing Systems*, ACM, April 1991, p.16-23.
5. Goldstein, J., Roth, S.F., Kolojejchick, J., and Mattis, J. A Framework for Knowledge-based Interactive Data Exploration. *Journal of Visual Languages and Computing*, No. 5, 1994, p.339-363.
6. Herndon, K.P. and Meyer, T. 3D Widgets for Exploratory Scientific Visualization. *UIST '94*, November 1994, p.69-70.
7. Hollands, J.G., Carey, T.T., Matthews, M.L., and McCann, C.A. Presenting a Graphical Network: A comparison of Performance Using Fisheye and Scrolling Views. *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Elsevier Science B.V., Amsterdam, 1989, p.313-320.
8. Lamping, J., Rao, R., and Pirolli, P. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings CHI'95 Human Factors in Computing Systems*, ACM, May 1995.
9. Leung, Y.K., and Apperley, M.D. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction*, Vol. 1, No. 2, June 1994, p.126-160.
10. Lieberman, H. Powers of Ten Thousand: Navigating in Large Information Spaces. *UIST '94*, November 1994, p.15-17.
11. Robertson, G.G., Mackinlay, J.D., and Card, S.K. Cone Trees: Animated 3D visualizations of Hierarchical Information. *Proceedings CHI'91 Human Factors in Computing Systems*, ACM, April 1991, p.173-179.
12. Robertson, G.G., and Mackinlay J.D. The Document Lens. *UIST '93*, November 1993, p.101-108.
13. McDonald, J.A. Painting Multiple Views of Complex Objects. *ECOOP/OOPSLA '90 Proceedings*, October 1990, p.245-257.
14. Sarkar, M., Snibbe, S.S. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. *UIST '93*, November 1993, p.81-91.
15. Stevens, M.P., Zeleznik, R.C., and Hughes, J.F. An Architecture for an Extensible 3D Interface Toolkit. *UIST '94*, November 1994, p.59-67.
16. Stone, M.C., Fishkin, K., and Bier, E.A. The Movable Filter as a User Interface Tool. *Proceedings CHI'94 Human Factors in Computing Systems*, ACM, April 1994, p.306-312.