

Zoomable User Interfaces – Literature Review

Nikitas Liogkas, Manas Tungare

December 2002

Introduction

Over the past thirty years the WIMP (Windows, Icons, Menus, Pointer) user interface paradigm has made the computer into a tool that allows non-specialists to get a variety of tasks done. In recent years, however, the applications available under this interface paradigm have become larger and more unwieldy, in an effort to accomplish distinct functional goals. As a successor to the traditional desktop interface, the *Zoomable User Interface* (ZUI), sometimes called multi-scale user interface, is a relatively new paradigm. It takes advantage of the user's spatial memory to create a more customized and dynamic working environment. It also facilitates the development of finer-grained applications that automatically interoperate with various types of data objects and other applications. A ZUI uses the metaphor of an infinite two-dimensional plane to represent the user's workspace. The user's view of this workspace can be varied both in position and scale by panning and zooming respectively, and the size of the objects in the workspace can be similarly altered. This model creates sufficient space, so that the user's data objects can all be assigned permanent absolute geographic locations, leading to an advantageous use of human spatial intuition.

The term ZUI was coined by Bederson and Hollan [2]. They proposed to use both panning and zooming to navigate through a large information space via direct manipulation. They claimed that the physics of the zoomable interface allows scaling to larger information spaces in a way that the current metaphor of files, menus and windows cannot match. Their belief was that human-computer interfaces should take advantage of the natural human capabilities of spatial cognition, and that the zoomable interface paradigm is one approach that does so. In this literature review, we provide a summary of the historical background of ZUIs, as well as outline the current state of the art.

1978 – 1986: The idea is born

A number of researchers have developed ways to visually structure interactive information that offer an alternative to the WIMP paradigm. One of the first such systems was the Spatial Data Management System (SDMS) [5], developed in 1977 by the Architecture Machine Group at MIT. The SDMS consisted of a sophisticated (for its era) environment of keyboard-less, interactive and large-scale graphics, where computer images, audio and video are viewed as one. The MIT researchers made the important observation that humans often organize their own collections of information spatially, placing it in convenient or easily remembered locations.

The SDMS took advantage of the user's a priori understanding of space for the purposes of managing a very large database. The user was allowed to "fly" around the information spaces of the database. To facilitate the navigation, the MIT researchers employed the use of a main/focal display and an ancillary/peripheral display, which they dubbed the "world view monitor". Consequently, the information landscape was presented on two screens: the world view monitor provided a panoramic overview showing a highlighted you-are-here rectangle, and the application screen provided a closer detailed view. The user could either pan locally around on the main screen or jump directly to an area by pointing and clicking on the panoramic view. As a result, movement in the information space was performed either via two pressure-sensitive joysticks or via the touch-sensitive screen of the world view monitor. Zooming in particular was controlled by the left-hand joystick. An important contribution of the work done at MIT was the introduction of the notion of displaying different kinds of information about an object depending on the current magnification factor for that object (we will see later that this is called *semantic zooming*). For example, a movie in the SDMS was shown as a still image, but when the user zoomed in, the image was switched to a video recording.

A second important step to the direction of ZUIs was made by George Furnas, who, in 1986, suggested the use of fisheye views for the display of large hierarchical data structures [6]. By *fisheye views* he meant the technique of representing the close neighborhood in great detail, yet only major landmarks further away. His work stemmed from identifying a major problem in traditional interfaces, which is the fact that it is easy to get lost while browsing through a large information space, since traditional views provide little information about the global structure and where the current view fits in. In contrast, he proposed the use of wide-angle lenses that show places nearby in great detail while still showing the whole world. He presented the fundamental motivation for such an approach to be the provision of a balance of local detail and global context.

Furnas further formalized his idea by observing that, many naturally-occurring fisheye views combine the relative distance from an object and its "a priori" importance to decide the level of detail in which that object is presented. As a result, he defined a *degree of interest* (DOI) function for every object in the information space as the difference of the a priori importance of the object and the distance from the current view. He subsequently showed that such an approach can be used for numerous structures, including lists, trees and acyclic directed graphs. He also indicated that, unlike SDMS, this approach could be used with structures that are neither spatial nor do they have graphic output. Finally, he made the interesting observation that some cases ask for multi-focus views, meaning that the user might need to see detail in more than one place at a time in the information space, having in effect a fisheye context around each.

1993 – 1994: The lens metaphor

Furnas' abstract idea of having multiple focus points within a single application was instantiated and even further extended by Bier *et al.* with the introduction of the *see-through interface* [4] in 1993. According to this new interface paradigm, new widgets, dubbed Toolglass widgets, appear on a virtual sheet of transparent glass between the application and the traditional cursor. These widgets may provide a customized view of the application underneath them, using

viewing filters called Magic Lens filters. Each filter comprises a screen region together with an operator performed on objects viewed in the region. The user positions a toolglass sheet over desired objects and then points through the widgets and lenses to perform the desired operation. The main advantages of these new widgets are that they do not require dedicated screen space and they provide both rich context-dependent feedback and the ability to view details and context simultaneously.

Several tools that demonstrate the features of the see-through interface were developed. Most of them were implemented in the graphical editing domain, but this paradigm could be used in a wide variety of application domains. The specific advantages of lenses as a user interface tool beyond their use in the see-through interface were demonstrated in [12]. It was emphasized that the magic lens metaphor can, under specific circumstances, be used uniformly across applications, thus increasing the consistency of the user interface. That is, rather than being tied to a single application window, magic lenses provide an interface to common functionality applicable to several applications.

1993-1996: Introduction of multi-scale interfaces

In 1993, Ken Perlin and David Fox developed Pad [9], taking a significant step towards the ZUI concept. The Pad embodied a single infinite shared desktop, any part of which could be visible at any given instant. A similar concept to the lens metaphor, dubbed *portals*, was introduced. Portals provided similar functionality to magic lenses, but were presented primarily as a metaphor for navigation as opposed to a general-purpose tool for exploring and modifying information. The main motivation behind Pad was the belief that navigation in information spaces is best supported by tapping into the users' natural spatial and geographic ways of thinking. In that respect, Pad, being an infinite two-dimensional information plane that is shared among users, allowed for assigning each object a well-defined region on the surface. Using "magic magnifying glasses" the user could read, write, or create cross-references on this zoomable surface.

A novel concept introduced in Pad was *semantic zooming*, which consists of an alternative to scaling objects purely geometrically. It ensures that far objects not only reduce in size, but also in information content. For example, instead of representing a page of text so small that it is unreadable, it might make more sense to present an abstraction of the text, perhaps just a title that is readable. Another important concept introduced by Pad was that of *portal filters*. Portal filters modify the view of the object underneath them and its appearance through the particular filter, but not the underlying object. For example a portal filter could display all tabular data as bar charts, while leaving the data itself unchanged.

The Pad interface can be compared to hypertext: users can click their way around cyberspace in both worlds. The chief difference is that hypertext does not assign a position to each page; thus, pages are separate from each other, and are linked solely via hyperlinks. There can be no "top-level" view that shows the entire scope of this cyberspace. Also, while navigating around hypertext, users often do not know their exact position. Additional tools such as history lists compensate for this shortcoming, though not in the most natural way.

Shortly after the development of Pad, the need to more formally analyze multi-scale interfaces brought about the concept of *space-scale diagrams* [7]. These diagrams represent both a spatial world and its different magnifications explicitly, thus allowing the direct visualization and analysis of important scale-related issues for user interfaces. This means that these diagrams constitute an analytical tool for accurately describing multi-scale information spaces. They typically consist of two axes with the vertical one representing scale, i.e. the magnification of an object at any given level. In this context, a point in the original space becomes a ray in the diagram. The principal virtue of these diagrams is that they represent scale explicitly. They proved to be very useful when trying to understand multi-scale interfaces geometrically, for guiding the design of programming code, and even as interfaces to authoring systems for multi-scale information.

Extensive use of the space-scale diagrams was performed in order to analyze and develop the first so-called ZUI toolkit, named Pad++ [2], the successor to Pad. The chief motivation was to develop a "substrate for exploration of novel interfaces for information visualization and browsing in complex information-intensive domains". Another main motivation was the capability to display large structures of information on a small display. Pad++ was developed at a time when mobile devices with small but readable displays had just made their way into the market, and the UI challenges these devices presented could be addressed nicely by such a tool.

The creators of Pad++ stressed the fact that metaphor-driven interface designs force the designer to think in terms of non-computer interfaces and then use those ideas for computer interfaces. This totally ignores the vast set of interfaces that have no equivalent in the real world, yet are perfectly suited for computer implementation. They insisted that such exploration of new media has been hampered by the desire to stick to known metaphors. To further prove their point, they noted that users who have used a metaphor-driven interface for a long time tend to forget the original metaphor and begin to think in terms of the interface objects themselves as first-class objects. Since Pad++ was essentially a successor to Pad, it featured the same portals and lenses. There was also great effort expended to make Pad++ efficient in terms of graphics rendering, as well as user-friendliness. Lastly, animation was an important facet of Pad++. Its creators believed that animating motions is responsible to a large degree for consolidating the feeling of a user when working with an infinite canvas.

1999-2002: State of the art in ZUIs

In 1999, Ken Perlin and Jon Meyer combined ZUIs with nested UI widgets, which led to the interesting concept of recursively nested user interfaces [10]. Their major goal was to present large and layered controls to the user as an easy-to-navigate UI surface. They thus managed to bring these two areas together to achieve more than either paradigm individually can. Having nested components by themselves leads to a finite depth to which nesting can occur. When used inside a ZUI, however, these controls can be nested to an arbitrary depth. Not all controls need to be visible at the top-most level, and more controls can be added as the user deems fit. Navigation remains consistent between the previously visible controls and any newly exposed controls; the user does not need to shift focus. Changing the values of controls can be accomplished at any

zoom-level, and the screen is immediately updated to reflect the changes made. The creators of this new paradigm additionally stressed the important role of animation in transitions, in order to assist the human cognitive system in maintaining the relations between states in the application.

Along with the complexity of ZUIs came the need to prevent this very complexity from being a cognitive load on the user. Pook *et al.* suggested multiple ways for the user to maintain context at the same time as viewing zoomed-in data [11]. Their method was different from Furnas' fisheye metaphor, according to which objects closer to the focus area are shown larger than those away from the focus. Instead, they combined a context layer with a history layer. The user can choose to set either of these as transparent or as opaque as necessary, depending on the action that is being performed at the moment. They also proposed the use of a hierarchy tree; however, this can only be used on hierarchical data sets and not on an arbitrary zoomable canvas.

The latest ZUI toolkit was developed at the University of Maryland and is called Jazz [3]. It borrows ideas from earlier toolkits such as Pad and Pad++, and adds the concept of a scene graph to make ZUI application development even easier. The use of a modern programming language like Java, further enhanced the feasibility of developing real-world applications based on the ZUI paradigm. Jazz also opens up the possibility of embedding a ZUI component inside a regular Java application built using the Swing toolkit. Jazz essentially combines the best concepts from both 2D and 3D graphics worlds, in order to effectively create a 2.5D space. Transformation nodes are separate from visual components, so the content and display of each information item can be independently changed.

Various applications were built on the Jazz toolkit. PhotoMesa [1] is an image browser that provides a zoomable view into a directory structure containing images. It uses treemap algorithms to perform automated screen space management. When the user hovers the mouse over a small thumbnail image, a larger thumbnail is displayed in its place, much like what happens in the tooltip metaphor. PhotoMesa clearly illustrates the practical applicability of ZUIs. CounterPoint [8] is a presentation tool designed to enhance an existing tool (Microsoft PowerPoint) with features only possible using a ZUI. Innovative such features include visual slide sorting and ordering, and the ability to follow multiple paths through the slides (and to return to a desired position later). It also allows presentation-time modifications to a slideshow based on time-constraints and audience feedback, a feature that PowerPoint lacks at the time of this writing. Using CounterPoint it is possible to develop only a single version of slides for different audiences or different available time constraints. Its creators argue that it leads to better retention due to spatial correlation and that it follows Don Norman's "visibility principle", thus enhancing its usability.

Zoomable User Interface research has been drawing attention from more and more researchers lately. There have been several implementations of ZUI toolkits developed by individuals for research purposes, as well as some commercial ones that have not been widely publicized. It is our belief that ZUIs can and should be used as a user interface paradigm in cases of applications that can take advantage of the natural human capabilities for spatial cognition.

REFERENCES

1. Bederson, B. B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In *Proceedings of User Interface Software and Technology (UIST 2001)*, ACM Press, pp. 71-80.
2. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. In *Journal of Visual Languages and Computing*, 7, pp. 3-31.
3. Bederson, B. B., Meyer, J., & Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proceedings of User Interface Software and Technology (UIST 2000)*, ACM Press, pp. 171-180.
4. Bier, E., Stone, M., Pier, K., Buxton, W., & DeRose, T. (1993). Toolglass and Magic Lenses: The See-Through Interface. In *Proceedings of Computer Graphics (SIGGRAPH 93)*, ACM Press, pp. 73-80.
5. Donelson, W. (1978). Spatial Management of Information. In *Proceedings of Computer Graphics (SIGGRAPH 78)*, ACM press, pp. 203-209.
6. Furnas, G. W. (1986). Generalized Fisheye Views. In *Proceedings of Human Factors in Computing Systems (CHI 86)*, ACM Press, pp. 16-23.
7. Furnas, G. W. & Bederson, B. B. (1995). Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proceedings of Human Factors in Computing Systems (CHI 95)*, ACM Press, pp. 234-241.
8. Good, L. & Bederson, B. B. (2001). CounterPoint: Creating Jazzy Interactive Presentations. *HCIL Tech Report #2001-03*, University of Maryland, College Park, MD 20742.
9. Perlin, K., & Fox, D. (1993). Pad: An Alternative Approach to the Computer Interface. In *Proceedings of Computer Graphics (SIGGRAPH 93)*, ACM Press, pp. 57-64.
10. Perlin, K., & Meyer, J. (1999). Nested User Interface Components. In *Proceedings of User Interface Software and Technology (UIST 99)*, ACM Press, pp. 11-18.
11. Pook, S., Lecolinet, E., Vaysseix, G., & Barillot, E. (2000). Context and Interaction in Zoomable User Interfaces. In *Proceedings of Advanced Visual Interfaces (AVI 2000)*, ACM Press, pp. 227-231.
12. Stone, M. C., Fishkin, K., & Bier, E. A. (1994). The Movable Filter As a User Interface Tool. In *Proceedings of Human Factors in Computing Systems (CHI 94)*, ACM Press, pp. 306-312.