Behaviour & Information Technology
Vol. 30, No. 6, November–December 2011, 853–866

Taylor & Francis
Taylor & Francis Group

The promise of zoomable user interfaces

Benjamin B. Bederson*

Human-Computer Interaction Lab, Computer Science Department, University of Maryland, 3171 A.V. Williams Building, College Park, MD 20742, USA

(*Received 1 September 2010; final version received 21 April 2011*)

Zoomable user interfaces (ZUIs) have received a significant amount of attention in the 18 years since they were introduced. They have enjoyed some success, and elements of ZUIs are widely used in computers today, although the grand vision of a zoomable desktop has not materialised. This paper describes the premise and promise of ZUIs along with their challenges. It describes design guidelines, and offers a cautionary tale about research and innovation.

Keywords: zoomable user interfaces; zooming user interfaces; multi-scale; information visualisation

1. Introduction

The essential problem that zoomable user interfaces (ZUIs) aim to solve is a fundamental one – that there is more information than fits on the screen. The common solutions to this problem are, roughly, scrolling, linking and searching, along with denser representations (i.e. information visualisation). Zooming, like fisheye displays (Furnas 1986), is an instance of the latter – a kind of information visualisation that aims to take advantage of human spatial perception and memory. ZUIs place documents in 2D space at any size, enabling (and requiring) animated spatial navigation to move among documents.

1.1. What is a ZUI?

Before we go further, let us think a little about what it means to be a ZUI. Many applications include some kind of visual scaling functionality. Modern file browsers let users control the size of icons. Web browsers, word processors, image editors, and in fact, almost all full-featured document editors and browsers let the user control the magnification of the document. Many let the user zoom far enough out to see small thumbnails of all the pages of even long documents on modest size screens. However, that kind of simple scaling is outside the scope of this paper.

This paper defines ZUIs to be those systems that support the multi-scale and spatial organisation of and magnification-based navigation among multiple documents or visual objects (examples in Figures 1 and 2). Admittedly there is a grey area where it might not always be clear whether a particular application is a ZUI according to this definition. For example, a word processor or document viewer that lets you zoom out and see thumbnails of pages laid out in a grid minimally meets that definition. However, the pages are really elements of a whole, and not movable in space individually. On the other hand, a viewer that let you zoom out, see and move arbitrary numbers of separate documents, even that were one page each, would count as a full ZUI according to this definition.

According to Cockburn *et al.*'s (2008) survey of approaches to fitting information on the screen, ZUIs display information that is *temporally* separated. The essence of this approach is that the user moves through space and builds up a spatial model of the information in their head. This is distinguished from *spatial* separation (found in overview + detail interfaces such as those found in maps) and *focus* + *context* or 'fisheye' distortion such as that found in Apple's OS X Dock (2010) and with the tabular approaches of TableLens (Rao and Card 1994) and DateLens (Bederson *et al.* 2004a).

1.2. Why ZUIs excite people

Based on my own experience and analysis of the literature, I have identified three key characteristics that have attracted people's attention over the years.

*Email: www.cs.umd.edu/~bederson; bederson@cs.umd.edu

ISSN 0144-929X print/ISSN 1362-3001 online

http://dx.doi.org/10.1080/0144929X.2011.586724
http://www.tandfonline.com

The promise of ZUIs comes largely from the following general expectations.

1.2.1. *ZUIs are engaging*

The animation is visually attention grabbing. It takes advantage of human visual perception abilities. People can process 'visual flow' (Ware 2004) preconsciously, so it feels easy to build a mental map of the information.

1.2.2. *ZUIs are visually rich*

There are more degrees of freedom to visually structure objects, and thus they offer the potential for great creative expression. This was identified by Perlin and Fox in their original paper on Pad with their example of a branching tree story (Perlin and Fox 1993).

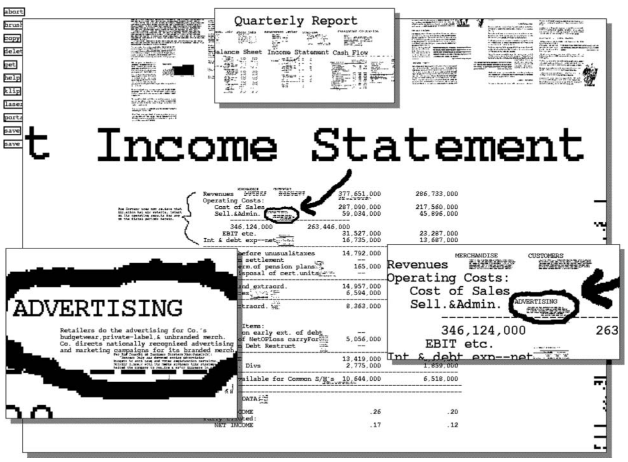

Figure 1. Early ZUI: Pad shows content at different sizes with portals that show a remote region of the data surface (Perlin and Fox 1993, Figure 1). © 1993 Association for Computing Machinery, Inc. Reprinted by permission.

1.2.3. *ZUIs offer the lure of simplicity*

The fact that you find information by looking for it in a place implies a promise of simplicity that will solve our organisational and information retrieval problems. The overview one sees when zoomed out seems like a panacea – since one finally knows where everything is. This idea was captured in the conclusion of Perlin and Fox's original paper: 'As compared to standard current window models, this system makes it easier for the user to exploit visual memory of places to organise informationally large workspaces' (Perlin and Fox 1993).

Yet, even with these qualities, the grand vision of a zoomable desktop has never been broadly achieved. For example, several commercial efforts have disappeared (e.g. Cincro Zanvas, GeoPhoenix Zoominator and Innovative iBrowser). Variations such as Task Gallery (Robertson *et al.* 2000), which used linear zooming in a 3D environment among others also, have not been broadly used. There has, however, been a resurgence of commercial interest in ZUIs recently, but as we will see they have significantly scaled back expectations as to where zooming can be useful.

2. ZUIs' premise and promise

ZUIs have interested researchers since they were introduced by Perlin and Fox (1993). There have been a number of widely cited papers focused in several areas:

Figure 2. Recent ZUI: Zumobi's ZoomCanvas zooms in from the entire canvas on start-up. Dragging left/right pans, and tapping on a region zooms in for interaction with the detailed content (Zumobi 2010).

- ***Systems*** [e.g. Pad (Perlin and Fox 1993), Pad++ (Bederson and Hollan 1994), Jazz (Bederson *et al*. 2000) and Piccolo (Bederson *et al*. 2004b)].
- ***Applications*** [e.g. PadPrints (Hightower *et al*. 1998), PhotoMesa (Bederson 2001), CounterPoint (Good and Bederson 2002), KidPad (Druin *et al*. 1997), AppLens& LaunchTile (Karlson *et al*. 2005), Microsoft Seadragon (2010), Fly (Lichtschlag *et al*. 2009), Microsoft pptPlex (2010), Microsoft Canvas for OneNote (2010), Microsoft Pivot (2010) and Prezi (2010)].
- ***Theoretical constructs*** [e.g. Space-Scale Diagrams (Furnas and Bederson 1995) and Desert Fog (Jul and Furnas 1998)].
- ***Studies*** (e.g. Combs and Bederson 1999, Boltman *et al*. 2002, Hornbæk *et al*. 2002, Klein and Bederson 2005 and Plumlee and Ware 2006) to support and understand them.

Many of these have had a fair amount of accolades and positive user response, yet it is fair to say that none of them have been great commercial successes (defined either monetarily or by large numbers of users).

In fact, I would argue that ZUIs have never reached the level of broad use envisioned by their original creators. There has been a huge amount of effort relating to ZUIs with dozens of published papers and a commensurate amount of technology developed. Zooming has been successful in that some kind of zooming is commonly used in a wide range of interfaces (e.g. Google Maps, Microsoft Word, Adobe Photoshop and Apple iPhone). But a richer kind of zooming that takes over the desktop and becomes the primary interface to one's data never materialised.

As one of those early creators, and as someone who had worked on many aspects of ZUIs since nearly their beginning, I think it is worth reflecting on what that original excitement was about, where ZUIs have been successful, where they have not been and why. For example, a number of commercial efforts that pursued a deeper kind of zooming had only modest success (e.g. an effort by Sony to include zooming in early VAIO computers (Kunkel 1999, pp. 160–165), and Hillcrest Labs' HoME product that has not had significant distribution (Hillcrest Labs 2010).

In reflecting on this body of work, it may be possible to increase our understanding of why it is sometimes harder to bring innovations to broad use than initially thought. And as researchers, why we should perhaps be a bit more tempered in our enthusiasm – while, of course, not stifling innovation before it has the opportunity to flower.

2.1. *The early history*

All ZUIs were built with the goal of addressing the fundamental issue that this paper started with – that people need to interact with more information than fits the screen. This is such a fundamental issue in computing and interface design that it seemed exciting and appropriate to explore a new approach to address the problem. The challenge was that the problem was vaguely defined – without users or tasks – and the solution was technically very difficult. This led to several years of building general solutions rather than specific ones. Let us start by looking at the early history.

'Pad' was the first system that explored this space (Perlin and Fox 1993) (Figure 1). Pad ran on a Sun SPARCstation with black and white graphics, displayed one bit per pixel bitmaps, and used non-animated 'jump' zooming (each mouse click would redisplay the view magnified or reduced by a factor of two). Pad offered navigation, authoring, semantic zooming and portals. The term 'semantic zooming' (coined by David Fox) refers to how objects can have different visual representations at different sizes. Portals were rectangular objects on the surface that acted like cameras that showed other parts of the surface. Portals were designed to solve the limitation that spatial layout implied. It enabled objects that were physically far apart to be used near each other.

Back in 1992, I was finishing my PhD at New York University (NYU), watching Perlin's work closely, and when I joined Bellcore after my graduation, I began building the first of what turned into a series of successors to Pad. While working with Jim Hollan, I started with Pa3D, a ZUI with richer vector and bitmap colour graphics and smoothly animated zooming – running on much more expensive SGI computers. Notably, Pa3D was in a 3D environment where every polygon could be a zoomable surface. There are no papers about Pa3D, but a video showing it is available online (Bederson 1993).

Pad and Pa3D assumed that information would be placed at different scales in this gigantic information space, and users would access the information by panning and zooming through the space using portals to connect otherwise distance objects. In other words, we aimed to build a zoomable desktop. We made many demonstrations of how people might organise their information in this space, but notably, despite our conviction that zooming was an incredible solution, we had only limited examples of how people would perform day to day tasks.

Being confident that zooming was a good idea, but just not sure what for, I started building Pad++ also with Jim Hollan (Bederson and Hollan 1994a, Bederson *et al*. 1996). Pad++ was a successor to

Pa3D, which incorporated a well-defined application programmer's interface (API), so others could build zoomable applications, and hopefully reveal the power that zooming offered.

Over the years, teams I led (with Jon Meyer, Jesse Grosjean and Aaron Clamage) extended Pad++ and then built ZUI platforms Jazz (Bederson *et al.* 2000) and Piccolo (Bederson *et al.* 2004b, Piccolo2D 2010). We supported other platforms and languages, and even supported mobile devices with PocketPiccolo.NET. There was also a significant effort to support smooth zooming, high-quality animated graphics and overall high performance on what at the time were fairly meagre computing systems (Bederson and Meyer 1998). Many other researchers and a few companies built extensions and applications on these platforms (especially Pad++) as did we. In more recent years, commercially available platforms such as Java, .NET and Flash made it easier to build zoomable applications, and client/server solutions have also been built (Microsoft Seadragon, Zoomorama, Google Maps).

One significant effort was by Sony starting in 1998. They started an internal effort to produce ZUIs for their VAIO PC computers that were new at the time (Kunkel 1999, pp. 160–165). Franklin Servan-Schreiber who led that effort coined the term 'ZUI',[1] and is the person who led the recently defunct effort at Zoomorama (2010). However, Sony never shipped any ZUI products related to that effort.

Looking at the core original ZUI ideas (zooming, semantic zooming and portals), it is interesting to see which ones eventually had broad usage. Basic zooming has been used widely in everything from maps to document viewers. Semantic zooming has also had some broader use – even outside the realm of ZUIs. The 'ribbon' in some Microsoft Office products (such as Word) uses semantic zooming to display more information when more space is available. It dynamically changes as you resize the ribbon to show more buttons, and to show more imagery as the panel gets larger. Similarly, several on-screen music players show more controls and information when the player is enlarged. It is notable that not a single one of these applications uses portals. Many applications (such as Microsoft Word, Google Maps and Adobe Photoshop) offer multiple views or a small fixed navigation overview. However, none of these applications that focus on zooming as a core organisational and navigation technique use portals in a way similar to how they were envisioned.

2.2. *ZUI characteristics*

There are a few essential characteristics of ZUIs that affect their usefulness and their usability.

Different ZUIs have had various levels of *layout flexibility*. By this, I refer to how much creative control the end user has over the layout of information in the space. On one extreme, users can put anything at any size and place. This yields complete artistic freedom, but can be difficult to author – and has the potential for creating a visual mess that users can get lost in. On the other extreme, the environment might be constrained, allowing information only in specific places according to a grid or layout algorithm. A related characteristic is whether specific layouts are *multi-level*, by which I mean that objects in the space appear at significantly different sizes requiring a user to navigate to different zoom levels in order to see all the objects.

Different ZUIs have also had various *navigation mechanisms* – which are ways for users to move through the space. Again there is a trade-off between flexibility and usability. Some interfaces allow users to fly through the space going absolutely anywhere – including deep into the spaces between objects [resulting in some researchers labelling this phenomena *Desert Fog* (Jul and Furnas 1998)]. Very few other applications let a user navigate beyond the actual content. Almost every document browser and editor limits navigation to the available content (with the notable exception of Microsoft Excel's scroll bar arrows – Apple Numbers and Google Spreadsheet, on the other hand, do limit navigation). On the other hand, some interfaces allow you only to click on objects to zoom into them and click on a zoom out button to zoom out – making it impossible to get lost, but also giving less control over exactly where you look.

One approach to managing the complexity of navigation is to zoom automatically when needed using a technique called speed-dependent automatic zooming (SDAZ).

2.2.1. *Speed-dependent automatic zooming*

One of the reasons that navigation is such a big issue is that there are too many degrees of freedom to easily control. In a traditional computer interface, one button is typically used for selection or action, and a scroll bar or specialised key is used for navigation. ZUI users, however, must pan along two axes instead of just one, and they must be able to zoom in and out. For most applications, this must be done with the same hardware used by non-zoomable interfaces.

Some researchers addressed this by reducing the number of navigational degrees of freedom that users needed to control in the first place (Igarashi and Hinckley 2000, Cockburn *et al.* 2005, Sun and Guimbretière 2005). One of the reasons for zooming out is simply to make it faster to pan a large distance [as explained with space-scale diagrams (Furnas and Bederson 1995) and elaborated on by Van Wijk and Nuij (2004)]. This observation led to SDAZ where the application would automatically zoom out when the user panned quickly, and then zoom back in when the user slowed down.

Several studies showed the effectiveness of this approach (Cockburn *et al*. 2005, Sun and Guimbretière 2005), and yet not a single research or commercial ZUI application (outside those that are explicitly exploring SDAZ) uses SDAZ as even an optional navigation mechanism. Why is this? There is not a clear answer, but it is important to observe that while SDAZ simplifies one thing (degrees of freedom of control), it complicates another (device control and perception).

SDAZ requires the user to engage in a real-time perceive-think-act loop (Card *et al*. 1986). This real-time interactive manipulation was typical of the 'flying' in many of the earliest ZUI systems – and that caused significant usability problems. People who play video games like the interaction challenge of real-time interaction. People doing information processing tasks typically do not. They are more likely to want to spend their cognitive resources on the underlying task, not on analysing the interface – even if it results in them being slightly slower.

2.2.2. Mouse wheels

While SDAZ is a somewhat complicated way to control zooming, some clear winners for controlling zooming have emerged in the marketplace, and they are the mouse wheel and multi-touch 'pinching'. The mouse wheel has become fairly widespread and is used for numerous linear navigation actions – from scrolling to selection to zooming. It offers simple control over the extra degree of freedom, and while it is somewhat slow (requiring movement for each further bit of zooming), its simplicity makes it trivial to learn. The fact that zooming navigation often requires only a small scale change makes for a perfect mapping to the mouse wheel.

2.2.3. Touch screens and multi-touch interaction

The use of multi-touch 'pinch' gestures on touch screens is another control mechanism with growing popularity. Ishii and Ullmer originally envisioned this approach by indirectly using physical objects to scale and rotate images on a screen in 1997 (Ishii and Ullmer 1997). Then, in 2002, Rekimoto developed a direct solution by using multiple fingers on a capacitive touch screen to scale and rotate images (Rekimoto 2002). Finally, this approach was commercialised and is now broadly used in Apple's iPhone and other mobile phones, tables as well as Microsoft Surface. This kind of direct multi-touch interaction is easy to learn and operate. More generally, the rapidly growing use of high-end touch screen-based smartphones is increasing the need for zooming (with people frequently reading documents designed for larger screens on small screens), and pinching seems to be the de facto standard for controlling this.

2.3. Applications

To see something of the range of uses of ZUIs, and how their characteristics have changed over time, Table 1 shows a selection of zoomable applications. Only 'true' ZUIs are shown (those supporting more than one document or object). It captures a range of what people have been using zooming for, and makes apparent the range of approaches that people have taken with regard to layout flexibility and navigation. It is also clear that the essential problem of getting lost in Desert Fog has not been consistently avoided. Furthermore, it is clear that there is no consistency in the mechanisms that are used to navigate through space.

Of broader import, there is no obvious focus on users, tasks or problem domains. This is characteristic of both the potential and the challenge of ZUIs. Zooming is a completely general strategy, and so one should not be surprised to see such a lack of focus any more than one might see scroll bars and search used in an incredibly diverse set of scenarios. That said, there does seem to be a concentration of ZUIs applied to domains with highly visual documents – which, of course, makes sense given the visual nature of ZUIs.

2.4. Studies

Some of these and other ZUI applications have been studied for usability or effectiveness. However, it is the nature of those application-centric studies that it is difficult to tease out the relative benefit or cost of the zooming characteristics of those applications.

In fact, there are relatively few studies that look at the value of animated zooming more generally. But a few are particularly useful, so let us look at them.

2.4.1. Overviews

The first study by Hornbæk *et al*. (2002) looked at multi-scale information structure as well as the use of small overview windows that show a zoomed out view of the information. Surprisingly, it found that while participants like the overview window, its use resulted in lower performance. The study used geographical maps as the data set. Navigation worked by dragging the mouse to pan, holding down the left button to zoom in and holding down the right button to zoom

Table 1. Some ZUI systems and applications.

Year	Name	Description	Layout flexibility	Multi-level	Navigation mechanism for zooming
1993	Pad (Perlin and Fox 1993)	First ZUI platform, included several conceptualisations of usage scenarios	Unconstrained	Yes	Left click to zoom in. Right click to zoom out
1994	Pad + + (Bederson and Hollan 1994)	Second ZUI platform, included API for others to write ZUI apps	Unconstrained	Yes	Left click and hold to zoom in. Right click and hold to zoom out
1997	KidPad (Druin *et al.* 1997)	Uses 'local tools' (Bederson *et al.* 1996) to enable children to create stories	Unconstrained. Drawings, images, text positioned manually	Yes	Hyperlinks for path. Zoom in/out modes. Click to fly in/out
1998	PadPrints (Hightower *et al.* 1998)	Creates a visual hierarchical map of web pages visited	Dynamically generated tree-based node-link diagram	No	Right click-and-hold flies in/out
2001	CounterPoint (Good and Bederson 2002)	A plug-in to PowerPoint that enables presentation authors to lay out slides in zoomable space	Contains PowerPoint slides. Small number of layout algorithms with manual override	Yes	Click on slide for next. Right click-and-hold flies in/out
2001	PhotoMesa (Bederson 2001)	A personal photo browser, desktop and mobile	Contains photos in a dynamically generated group of grids using a treemap algorithm	No	Left click zooms in. Right click zooms out
2002	Seadragon (2010)	Client/server high perf.ZUI for images over a network (web and mobile)	Contains photos laid out by a small number of fixed layout algorithms	No	Click to zoom in. On-screen button zooms out
2002	Spacetree (Grosjean *et al.* 2002)	Hierarchy exploration tool	Dynamically generated tree-based node-link diagram	No	Right click-and-hold flies in/out
2006	Dynapad (Bauer 2006)	Supports organisation of personal collections	Dynamic, very flexible	No	Multi-touch
2007	Apple iPhone (2010)	Application launcher for Apple's mobile phone	Contains icons laid out in a fixed grid	No	Tap to zoom in. Physical button to zoom out
2008	Microsoft pptPlex (2010)	A plug-in to PowerPoint that enables presentation authors to lay out slides in zoomable space	Contains PowerPoint slides. Small number of layout algorithms with manual override	No, but with manual override	Click on slide for next. Click to zoom in. Esc to zoom out. Scroll wheel flies in/out
2008	Zoomorama (2010)	Client/server to enable high-performance zooming of images over a network	Dynamically generated grid containing images	No	Pop-up on-screen buttons for zoom in/out

(*continued*)

Table 1. (*continued*)

Year	Name	Description	Layout flexibility	Multi-level	Navigation mechanism for zooming
2008	Prezi (2010)	Website for creating and making free-form presentations	Unconstrained. Drawings, images, movies, text positioned manually	Yes	Tab for next. Scroll wheel, on-screen buttons and mouse/keyboard combo to fly in/out
2009	Zumobi (2010)* ZoomCanvas	Launcher for a small number of related content areas on high-end mobile devices	Fixed hand-tuned layout turned to content	No	Tap to zoom in, on-screen button to zoom out
2009	Microsoft Canvas for OneNote (2010)	Plug-in for Microsoft OneNote shows visual overview of all a user's notes	Dynamically generated grid of grids containing images of notes from Microsoft OneNote	No, but with manual override	Click on note to zoom in. Esc, right click or bgnd click to zoom out
2009	Schematic (2010)	Example of custom website with fixed layout of fixed content	Fixed hand-tuned layout tuned to content	No	Click on slide to zoom in. Click on background to zoom out
2009	Microsoft Pivot (2010)	Faceted search tool that uses zooming to navigate through search results	Fixed grid or grid within barchart layouts	No	Multiple mechanisms including clicking and on-screen buttons

Note: *Note that I, the author, am a cofounder of Zumobi, Inc.

out. The study varied whether or not users were shown overviews, and whether the map information was single or multi-level. That is, single level maps displayed all textual information in a single font size, so when the user zoomed out, the text could not be read. Multi-level maps displayed textual information at different sizes. Text referring to larger areas on the map was displayed at much larger fonts, so that when users zoomed out, they could read the text covering larger geographical areas.

Before this study, it was believed that overviews were a good idea and that they improved subjective satisfaction (North and Shneiderman 2000) and efficiency (Beard and Walker 1990, Bauer 2006). However, this study found a discrepancy between preference and performance. Study participants preferred the overview condition, but actually performed better in the no-overview condition, especially when using the multi-level map.

It turns out that switching between overview and detail windows was correlated with higher task completion time, suggesting that integration of overview and detail windows requires mental and motor effort.

This result is notable because a fundamental concern about ZUIs is that they require time to use (since the animations take time to occur). Worse, there is the potential that ZUIs tax human short-term memory (STM) because users must integrate in their heads the spatial layout of the information. On the other hand, there is the hope that the animated transitions are understood preconsciously by the human visual system, and thus may not in fact place a significant load on STM.

This study gives some evidence that the cost of understanding an animated zoomable map of information is in fact less than the use of one with an overview that is one of the primary alternative interface designs. Of course it does not say anything about the benefit or cost of a spatially organised information space compared to a non-spatially organised space. It also does not say anything about the benefit or cost of animation.

2.4.2. Animation

A second study looked at simple animated navigation transitions and found that these animations significantly increased performance – even including the time the animations took. These kinds of animations have long been expected to provide benefit (Card *et al.* 1991), but evidence backing up this understanding had been lacking. Klein and Bederson (2005) looked at a very simple but highly controlled use of animation when scrolling while reading documents. Understanding animation precisely is important because the potential benefit of animation is undermined by the fact that animation by its nature takes time. So, even if the animation provides some benefit, there is a cost as well, and it is not obvious whether the benefit outweighs the cost.

Participants in this study read documents out loud and when they got to the end of a page, they scrolled by pressing the down arrow key. The animation speed varied between 0 (i.e. unanimated), 100, 300 and 500 ms. The document type also varied between unformatted text, formatted text and abstract symbols (where participants counted symbols instead of reading). The reason for varying the document type was to understand how visual landmarks interact with animation.

The essential result was that animation did in fact provide a significant benefit, even considering the time spent on the animation. Reading errors were reduced by 54% with 500 ms animations. Reading task time was reduced by 3% and counting task time was reduced by 24% for 300 ms animations. The formatted documents had a higher overall performance (and lower improvement), implying that visual landmarks appear to be a good idea, which is consistent with the Hornbæk study's observation about multi-level maps.

The lesson from this study is that when designed correctly, the benefits of simple animated navigational transitions can outweigh their costs. This supports the design decision to make zooming transitions use short animations. This study is not definitive however since it is possible that the benefit that was seen while scrolling would not occur while zooming. However, there is no obvious reason that the same benefits would not occur, so this kind of animated transition continues to seem valuable.

2.4.3. Presentations

Two other studies looked at how people learned from zoomable spaces (Boltman *et al.* 2002, Good and Bederson 2002). The outcomes here were more mixed, where the zoomable presentation did not produce better learning or memory of the presentation – although participants did remember the structure of the presentation better. The first study looked at how children responded to a story that was presented with or without animated zooming. The children with the animated condition elaborated more during a discussion, but did not otherwise recall the story better. The second looked at how college students responded to a slide presentation – again with or without animated zooming. The presentations were structured visually in a 2D hierarchical format with six sections arranged in a circle where each section contained slides arranged in a circle. These students also did not recall the presentation better, but the ones who saw the animated zooming condition did recall the *structure* of the presentation better.

Together, these studies imply that zooming alone is probably not enough to help people remember content better. However, users of ZUIs may be more engaged, and they may remember the spatial structure of the content better – even if they do not remember the actual content itself any better. However, the value of remembering structure depends on the task. If the structure has value in and of itself, then this is a positive value. If, on the other hand, the visual structure was added as a means to make the presentation more visually interesting and had no intrinsic value then such improved retention would provide no benefit.

2.4.4. Zooming vs. multiple windows

Two related papers (Plumlee and Ware 2002, Plumlee and Ware 2006) look more generally at modelling and then studying how people perform visual comparisons of multi-scale data using zoomable and multi-window interfaces. Informed by an understanding of the extreme limitations of human visual working memory, they modelled how people can search using the two interfaces. The model showed that comparison tasks that exceeded the capacity of visual working memory (which can hold roughly three visual objects) would be better served by adding windows. Multiple windows decrease visual working memory load because different views can be directly compared. In contrast, ZUIs require the user to hold everything outside of the current view in their memory – thus making them inappropriate for complex comparison tasks. The studies that Plumlee and Ware performed demonstrated this essential characteristic and the trade-off between ZUIs and multiple windows closely matched their models. They also showed that users of ZUIs had significantly higher error rates than users of multiple windows. The practical impact of these studies depends on the frequency and difficulty of performing zooming compared to creating and managing windows. And it also applies only to visual comparison tasks.

So again, the lessons are interesting and important to specific usage cases, but there remains no study telling us 'whether zooming is good'. The real answer is that it depends.

3. Discussion

A trend in ZUI applications that appears to be fairly clear over the years is that over time, these applications have tended to get simpler and easier to use while at the same time offering less creative control to end users. This is an important trend because researchers often have a tendency to create tools that are rich and powerful. After all, it is easier to conceptualise an innovation as offering 'more' rather than offering 'less'.

But the marketplace is clear – success often requires simplicity first and power second. Of course, marrying the two where the power is under the hood is often the best solution – but also very difficult to design. The success of so many modern applications (e.g. search, twitter, photo sharing, iPhone App Store) are examples of this in that they can be used with almost no training in tiny little bursts.

I used to think that only public kiosks had to have these simplistic characteristics – but now it seems that even applications that people use for hours a week and become true experts in follow the same rules. If true, this implies that we as researchers have to strongly consider not only what new things technologies can help people do but also consider simplicity and speed as fundamental goals.

3.1. Why ZUIs are challenging

As summarised in Table 2, the potential benefits of ZUIs are sometimes mirages. ZUIs are generally

Table 2. Promise and reality of ZUIs.

Promise	Reality
Engaging	*If designed well*. The animated nature of ZUIs is sometimes referred to as 'eye candy', implying that people cannot help but pay attention to the movement on the screen. However, this very movement is a double-edged sword because if the user does not pay attention and misses the animation, then the connection between the different views is lost and the user is likely worse off than without a ZUI.
Visually rich	*Can also make space more complex to navigate and understand*. ZUIs can have visual objects at any size and at any place – thus offering the possibility of placing significant content (scaled down) even in a tiny crack between two other objects. This kind of structure is very difficult to perceive (since users cannot see the tiny content in the first place) and conceptualise (since at any given view, the vastly different sized objects can not be seen simultaneously).
Simplicity	*Does not scale to very large data sets*. One of the attractions of ZUIs is that they appear conceptually simple ('just zoom in for a closer look'). But in practice, navigating large information spaces in ZUIs is often difficult to execute and requires significant short-term memory to navigate.

engaging (although they make some people feel physically sick) and visually rich. But the promise of simplicity falls short.

While human visual perception does make it easy to see where one is navigating, the reality is that it places a heavy load on short-term memory to remember where in space you just were and where things are. And the requirement of human memory to know how space is organised means that ZUIs do not scale up very well. ZUIs are often motivated by the physical world and how people like laying papers out on their desk. But no one wants *all* of their papers on their desk. It is much more common to have only a relatively small number of papers that are actually being worked with.

The visual overviews that ZUIs offer for free by zooming out may seem like a solution to the load on human memory, but in practice it does not because visual overviews of any complexity require significant visual search in order to find anything. If there are just a small number of objects then the visual search task is not hard – but of course, for a small number of objects, you do not need a ZUI to solve your organisational problems.

In addition, the visual richness of ZUIs is a double-edged sword. It requires skill to design a complex space with documents of arbitrary size, aspect ratio and colour that people can comprehend and scan. Also, people are not as good at scanning 2D designs as 1D layouts – unless the layout is highly structured. But highly structured 2D layouts do not work well for visual objects of arbitrary aspect ratios. Designers are obligated to leave a lot of unused space, scale down the large objects so they are unreadable, or crop the large objects – thus losing much of their distinguishability.

Finally, it is crucial to consider the efficiency of ZUIs and their alternatives. It is the nature of zoomable structures to organise information hierarchically. Zooming in to an object shows you more information about that object. Of course, hierarchical structures are also commonly exposed with a simple tree widget as is commonly found in most file system browsers. The file system browsers, exemplified by Windows Explorer, are not only very efficient but also more easily let the user compare different areas in the hierarchy by simply opening and collapsing the desired nodes. So, in many cases, the alternative to a ZUI is a simple tree widget that is simple, compact, easy to learn and fast to use. Thus in addition to the broader challenges of ZUIs, we should always consider the alternatives that are also powerful.

3.2. *Design guidelines*

So how should ZUIs be designed to best overcome these obstacles? Let us start with the big issues.

3.2.1. *Support the right tasks*

ZUIs, like most visualisations, have strong potential for supporting users in understanding the big picture, identifying trends, patterns and outliers. But they typically are not good at helping users simply get the best answer. So, be sure that some sort of overview-based design is called for in the first place.

3.2.2. *Be cautious about using zooming as the primary interface*

Zooming navigation can be difficult, and is uncommon. Unless the data set is fundamentally spatial (e.g. maps), users probably do not want zooming as their primary interaction with the data. Instead, think about how people search and create data sets – which may then be accessed partially by zooming; i.e., create hybrid systems that combine mechanisms such as faceted search and zooming [e.g. PhotoMesa (Bederson 2001), the Hard Rock Memorabilia (2010) or Microsoft Pivot (2010)].

3.2.3. *Only use ZUIs when a small visual representation of the data is available*

Zooming works best when the objects can be recognised when they are zoomed out. So, certain domains are better suited to ZUIs. Photos are good, purely textual documents are bad and audio recordings are terrible (unless there are associated images). Recent research shows how a small visual snippet may be automatically created from web pages (Teevan *et al.* 2009).

3.2.4. *Don't limit yourself to fixed data sets*

People rarely want to interact long with a fixed set of data. But it is often technically difficult to dynamically create and modify large zoomable spaces. Be sure that the user interface does not get blocked or slowed down when new data is added to a zoomable space.

And then there are several more specific suggestions:

3.2.5. *Maintain aspect ratio*

It is tempting to use semantic zooming to significantly change the visual representation of objects at different sizes. For example, a visual chart may be condensed to a single value or a text document may be replaced by its title when small. However, the small object must have the same aspect ratio – or at least fit in the same box as the full object. Otherwise, nearby objects can end up overlapping when the user zooms out.

3.2.6. Use meaningful layout

The problem of finding a specific object among a large number of them can be alleviated if the objects are laid out in a semantically meaningful way. If the user can predict where an object will be based on attributes that they are likely to be familiar with, then they can trim the search space. Traditional paper phones books with their alphabetic ordering are a good example of this approach.

3.2.7. Use consistent layout

Ideally, once an object is placed in space, it should not move. That way, the user can build up an internal map over time of where objects are. Unfortunately, this ideal is often in direct conflict with the previous one of meaningful layout. If objects are to be laid out meaningfully then the layout must change over time as the data changes and as the organisational mechanism changes. PhotoMesa (Bederson 2001) is an example of a ZUI that went for a meaningful layout but had no consistency at all.

3.2.8. Use scannable layout

Objects must be positioned in such a manner as to enable the user to visually scan the space to look for an object without missing or repeating objects. Simple 1D lists of objects are easy to scan because the person just runs down the list and the only state that must be kept is the current item and the direction of scanning. In 2D, however, for any layout except for a grid, there is the potential for a much heavier load on short-term memory.

3.2.9. Use breadth over depth

It is tempting to place objects in space at many different sizes, so more information can be discovered by zooming in further – and in fact, many of the earlier ZUI applications did this. However, when objects are placed at many different levels, the user would not know that the small objects exist (unless the space is carefully designed to indicate them), and they can get lost. Further, even if the user knows they are there, it is typically burdensome to zoom in and out repeatedly. SpaceTree (Grosjean *et al*. 2002) is an example of an application that was designed explicitly with breadth in mind. Even with hundreds of thousands of objects, all of them are at the same size.

3.2.10. Use small data sets

The problems that come with a large number of objects in a zoomable space can be mitigated by not building applications around large data sets. ZUIs appear to work better for users when they have no more than about 100 screens worth of information. However, ZUIs with 1000 or more screens worth of information become problematic because of the issues raised in the previous section (why ZUIs are challenging). This is a fundamental problem since ZUIs are often designed to work for user generated data – which may not easily be limited. So, application designers may choose to partition the space into manageable subsets.

3.2.11. Use simple and consistent navigation interaction

People must be able to discover how to navigate the space with no training, and it must be difficult or impossible to get lost in Desert Fog. Unfortunately, there is significant inconsistency among applications. They vary between single and double clicks to zoom in, how much the screen is zoomed per action, whether zooms scale by a fixed percentage or whether the target object fills the screen. And mechanisms are all over the map to zoom out – from right click, left click on the background, fixed on-screen buttons and pop-up buttons. And as with zoom in, the actual amount that is zoomed out is inconsistent across applications as well.

4. Conclusion

Understanding ZUIs is difficult. They have clearly been a success by traditional research measures (significant visibility, publications and citations). And elements of ZUIs have been broadly deployed in commonly used applications. But the original broad vision of zooming as a basic information organisational principle [such as Raskin's vision of hospital management (Raskin 2000)] has not happened. Of course, lack of commercial success does not mean an idea is not a good one (maybe the business reasons did not line up, or maybe the right tools got built, but with the wrong marketing strategy, the timing may have been wrong or any number of other reasons might have stalled a successful business). But still, the lack of broad success coupled with the fundamental issues raised in this paper appears significant.

I definitely would not rule out further success of ZUIs, and I think that the successes that have occurred, such as Google Maps, Microsoft Word and Apple iPhone, are substantial. However, their applicability is much narrower than original envisioned. And the design and technical challenges of building effective ZUIs make it that much less likely that they will be broadly deployed. At the same time, there are clear cases where zooming is valuable – such as with documents that are larger than screens (i.e. photos, web pages and maps). And the fact that they are visually engaging cannot be ignored.

Furthermore, as described in this paper, there is a range of potential benefits. For some users and tasks, those benefits will outweigh the costs. The problem, though, is that the precise balance is not well understood. Furthermore, the full range of zoomable designs is surely not fully explored. This leads to the following key areas for further research.

4.1. *Definitive understanding of benefit*

What, specifically, are the benefits of zooming transitions, and under what circumstances are those benefits found? Do different users have different performance capabilities and preferences when using ZUIs?

4.2. *Standardised navigation*

ZUIs need standardised navigation mechanisms. So many approaches have been used that they need to be catalogued so that it becomes easier to see what are good candidates for standards. And any companies that make more than one ZUI application must ensure that they at least have a company-wide standard.

4.3. *Design space exploration*

Despite all the effort over the years envisioning interaction, design and application domains for ZUIs, it is a very large space design. There are very likely still new approaches and applications to be discovered. I expect that creative design effort will continue to extend the richness and reach of ZUI applications.

4.4. *Reflections*

In retrospect, I believe that we as researchers sometimes let ourselves get too caught up in the excitement of something new and the coolness of something so engaging. When we could not figure out what the killer app in a zoomable environment was, perhaps we should not have been so fast to think that others would figure it out if only we solved the technological problems. Perhaps one or two platforms were a good idea, but were five necessary?

The idea that we should only develop technology when we know what it is for is dangerous because it is the nature of basic research that you do not always know what a new technology will enable. Transistors and lasers come to mind as technologies that the inventors had no idea what they were good for.

But I believe that in human–computer interaction (HCI), which is user facing by its nature, it should be much more apparent what the value to users is when developing technology. An example of HCI effort that saw more success only when researchers focused on user needs more than underlying technology is software engineering (Myers and Ko 2009). We are not developing technology for its own sake, we are developing technology to address user needs, and if we do not start by addressing those needs, we are putting our effort at risk. This is a point that seems too obvious to make except that many of us (especially computer scientists) sometimes miss it.

Acknowledgments

I have worked with many people on ZUIs over the past 18 years since Ken Perlin's first paper on Pad. I can't thank them all, but want to acknowledge the people I worked most closely with on ZUIs, especially in the early days. This includes, of course, Ken Perlin who had the original conceptualisation of ZUIs immediately followed by close collaborations on Pad + + with me, Jim Hollan, Jon Meyer and Allison Druin. I also appreciate the enthusiasm and commercial efforts of Franklin Servan-Schreiber who I worked with when he was at Sony. I worked closely with students Lance Good and Bongwon Suh and staff Jesse Grosjean and Aaron Clamage. Then my later work with Mary Czerwinski and George Robertson at Microsoft and John SanGiovanni, first at Microsoft and then at Zumobi, helped me to understand more closely what it really takes to understand an interface, polish a product and make it work for a large audience. And finally, my colleagues at the HCIL (especially Ben Shneiderman and Catherine Plaisant) have been instrumental in being rigorous and ruthless in helping me think about the real costs and benefits of the technologies we build.

Note

1. People commonly say 'zoomable user interfaces' or 'zooming user interfaces'. I prefer the former since it focuses on users being in control. However, there is neither a substantive difference between these terms, nor predominant usage of one over the other.

References

Apple iPhone [online]. Available from: www.apple.com/iphone [Accessed 7 January 2010].

Apple OS X Dock [online]. Available from: www.apple.com/macosx/what-is-macosx/dock-and-finder.html [Accessed 7 January 2010].

Bauer, D.S., 2006. *The cognitive ecology of dynapad, a multiscale workspace for managing personal digital collections*. Thesis (PhD). UMI Order Number: AAI3222983. University of California at San Diego.

Beard, D.B. and Walker, J.Q., 1990. Navigational techniques to improve the display of large two-dimensional spaces. *Behavior and Information Technology*, 9 (6), 451–466.

Bederson, B.B., 1993. *PA3D: early zoomable user interface (ZUI)* [online]. Available from: www.youtube.com/watch?v=5JzaEUJ7IbE [Accessed 7 January 2010].

Bederson, B.B., 2001. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. *In*: *Proceedings of the ACM Symposium on User interface Software and Technology*. UIST 2001. New York, NY: ACM, 71–80. DOI: http://doi.acm.org/10.1145/502348.502359.

Bederson, B.B., *et al*., 2004a. DateLens: a fisheye calendar interface for PDAs. *ACM Transactions on Computer–Human Interaction*, 11 (1), 90–119. DOI: http://doi.acm.org/10.1145/972648.972652

Bederson, B.B., Grosjean, J., and Meyer, J., 2004b. Toolkit design for interactive structured graphics. *IEEE Transactions on Software Engineering*, 30 (8), 535–546. DOI: http://dx.doi.org/10.1109/TSE.2004.44

Bederson, B.B. and Hollan, J.D., 1994. Pad++ : a zooming graphical interface for exploring alternate interface physics. *In*: *Proceedings of the 7th Annual ACM Symposium on User interface Software and Technology UIST 1994*. New York, NY: ACM, 17–26. DOI: http://doi.acm.org/10.1145/192426.192435.

Bederson, B.B., *et al*., 1996. Local tools: an alternative to tool palettes. *In*: *Proceedings of the 9th Annual ACM Symposium on User interface Software and Technology*. UIST 1996. New York, NY: ACM, 169–170. DOI: http://doi.acm.org/10.1145/237091.237116.

Bederson, B.B., *et al*., 1996. Pad++ : a zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing*, 7, 3–31.

Bederson, B.B. and Meyer, J., 1998. Implementing a zooming user interface: experience building Pad++. *Software: Practice and Experience*, 28 (10), 1101–1135.

Bederson, B.B., Meyer, J., and Good, L., 2000. Jazz: an extensible zoomable user interface graphics toolkit in Java. *In*: *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology*. UIST 2000. New York, NY: ACM, 171–180. DOI: http://doi.acm.org/10.1145/354401.354754.

Boltman, A., *et al*., 2002. The nature of children's storytelling with and without technology. *In*: *Proceedings of American Educational Research Association (AERA) Conference,* New Orleans. Washington, DC: AREA.

Card, S.K., Moran, T.P., and Newell, A., 1986. *The psychology of human–computer interaction*. Mahwah, NJ: Lawrence Earlbaum Associates.

Card, S.K., Robertson, G.G., and Mackinlay, J.D., 1991. The information visualizer, an information workspace. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 1991. New York, NY: ACM, 181–186. DOI: http://doi.acm.org/10.1145/108844.108874.

Cockburn, A., Karlson, A., and Bederson, B.B., 2008. A review of overview + detail, zooming, and focus + context interfaces. *ACM Computer Surveys*, 41 (1), 1–31. DOI: http://doi.acm.org/10.1145/1456650.1456652

Cockburn, A., Savage, J., and Wallace, A., 2005. Tuning and testing scrolling interfaces that automatically zoom. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 2005. New York, NY: ACM, 71–80. DOI: http://doi.acm.org/10.1145/1054972.1054983.

Combs, T.T. and Bederson, B.B., 1999. Does zooming improve image browsing? *In*: *Proceedings of the Fourth ACM Conference on Digital Libraries*. DL 1999. New York, NY: ACM, 130–137. DOI: http://doi.acm.org/10.1145/313238.313286.

Druin, A., *et al*., 1997. KidPad: a design collaboration between children, technologists, and educators. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 1997. New York, NY: ACM, 463–470. DOI: http://doi.acm.org/10.1145/258549.258866.

Furnas, G.W., 1986. Generalized fisheye views. *ACM SIGCHI Bulletin*, 17 (4), 16–23. DOI: http://doi.acm.org/10.1145/22339.22342

Furnas, G.W. and Bederson, B.B., 1995. Space-scale diagrams: understanding multiscale interfaces. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 1995. New York, NY: ACM Press/Addison-Wesley, 234–241. DOI: http://doi.acm.org/10.1145/223904.223934.

Good, L. and Bederson, B.B., 2002. Zoomable user interfaces as a medium for slide show presentations. *Information Visualization*, 1 (1), 35–49. DOI: http://dx.doi.org/10.1057/palgrave/ivs/9500004

Grosjean, J., Plaisant, C., and Bederson, B.B., 2002. SpaceTree: design evolution of a node link tree browser. *In*: *Proceedings of Information Visualization Symposium (InfoVis 2002)*. New York: IEEE, 57–64.

Hard Rock Memorabilia [online]. Available from: http://memorabilia.hardrock.com [Accessed 7 January 2010].

Hightower, R.R., *et al*., 1998. PadPrints: graphical multiscale Web histories. *In*: *Proceedings of the 11th Annual ACM Symposium on User interface Software and Technology UIST 1998*. New York, NY: ACM, 121–122. DOI: http://doi.acm.org/10.1145/288392.288582.

Hillcrest Labs [online]. Available from: www.hillcrestlabs.com [Accessed 7 January 2010].

Hornbæk, K., Bederson, B.B., and Plaisant, C., 2002. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions Computer–Human Interaction*, 9 (4), 362–389. DOI: http://doi.acm.org/10.1145/586081.586086

Igarashi, T. and Hinckley, K., 2000. Speed-dependent automatic zooming for browsing large documents. *In*: *Proceedings of the ACM Symposium on User interface Software and Technology*. UIST 2000. New York, NY: ACM, 139–148. DOI: http://doi.acm.org/10.1145/354401.354435.

Ishii, H. and Ullmer, B., 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 1997. New York: ACM, 234–241. DOI: http://doi.acm.org/10.1145/258549.258715.

Jul, S. and Furnas, G.W., 1998. Critical zones in desert fog: aids to multiscale navigation. *In*: *Proceedings of the 11th Annual ACM Symposium on User interface Software and Technology*. *UIST 1998*. New York, NY: ACM, 97–106. DOI: http://doi.acm.org/10.1145/288392.288578.

Karlson, A.K., Bederson, B.B., and SanGiovanni, J., 2005. AppLens and launchTile: two designs for one-handed thumb use on small devices. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 2005. New York, NY: ACM, 201–210. DOI: http://doi.acm.org/10.1145/1054972.1055001.

Klein, C. and Bederson, B.B., 2005. Benefits of animated scrolling. *In*: *Extended Abstracts on Human Factors in Computing Systms. CHI 2005*. New York: ACM, 1965–1968. DOI: http://doi.acm.org/10.1145/1056808.1057068

Kunkel, P., 1999. *Digital dreams: the work of the Sony design center*. New York: Universe Publishing.

Lichtschlag, L., Karrer, T., and Borchers, J., 2009. Fly: a tool to author planar presentations. *In*: *Proceedings of the 27th International Conference on Human Factors in Computing Systems*. CHI 2009. New York, NY: ACM, 547–556. DOI: http://doi.acm.org/10.1145/1518701.1518786.

Microsoft Canvas for OneNote [online]. Available from: www.officelabs.com/projects/canvasforonenote/ [Accessed 7 January 2010].

Microsoft Pivot [online]. Available from: http://research.microsoft.com/en-us/downloads/dd4a479f-92d6-496f-867d-666c87fbaada/default [Accessed 7 January 2010].

Microsoft pptPlex [online]. Available from: www.officelabs.com/projects/pptPlex/ [Accessed 7 January 2010].

Microsoft Seadragon [online]. Available from: www.seadragon.com [Accessed 7 January 2010].

Myers, B. and Ko, A.J., 2009. The past, present and future of programming in HCI. *In*: *Human–Computer Interaction Consortium (HCIC 2009), Winter Park, CO*. February. Available from: http://www.cs.cmu.edu/~natprog/papers/MyersKoHCIC09.pdf.

North, C. and Shneiderman, B., 2000. Snap-together visualization: evaluating coordination usage and construction. *International Journal of Human Computer Studies*, 53 (5), 715–739.

Perlin, K. and Fox, D., 1993. Pad: an alternative approach to the computer interface. *In*: *Proceedings of the 20th Annual Conference on Computer Graphics and interactive Techniques*. SIGGRAPH 1993. New York, NY: ACM, 57–64. DOI: http://doi.acm.org/10.1145/166117.166125.

Piccolo2D [online]. Available from: www.piccolo2d.org [Accessed 7 January 2010].

Plumlee, M. and Ware, C., 2002. Zooming, multiple windows, and visual working memory. *In*: M. De Marsico, S. Levialdi, and E. Panizzi, eds. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '02)*. New York, NY: ACM, 59–68. DOI: http://doi.acm.org/10.1145/1556262.1556270.

Plumlee, M.D. and Ware, C., 2006. Zooming versus multiple window interfaces: cognitive costs of visual comparisons. *ACM Transactions on Computer–Human Interaction*, 13 (2), 179–209. DOI: http://doi.acm.org/10.1145/1165734.1165736

Prezi [online]. Available from: www.prezi.com [Accessed 7 January 2010].

Rao, R. and Card, S.K., 1994. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 1994. New York, NY: ACM, 318–322. DOI: http://doi.acm.org/10.1145/191666.191776.

Raskin, J., 2000. *The humane interface*. Upper Saddle River, NJ: Addison-Wesley Professional.

Rekimoto, J., 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 2002. New York: ACM, 113–120. DOI: http://doi.acm.org/10.1145/503376.503397.

Robertson, G., *et al*., 2000. The task gallery: a 3D window manager. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI 2000. New York: ACM, 494–501. DOI: http://doi.acm.org/10.1145/332040.332482.

Schematic [online]. Available from: www.schematic.com [Accessed 7 January 2010].

Sun, L. and Guimbretière, F., 2005. Flipper: a new method of digital document navigation. *In*: *Extended Abstracts on Human Factors in Computing Systems*. CHI 2005. New York, NY: ACM, 2001–2004. DOI: http://doi.acm.org/10.1145/1056808.1057077

Teevan, J., *et al*., 2009. Visual snippets: summarizing web pages for search and revisitation. *In*: *International Conference on Human Factors in Computing Systems*. CHI 2009. New York, NY: ACM, 2023–2032. DOI: http://doi.acm.org/10.1145/1518701.1519008.

van Wijk, J. and Nuij, W., 2004. A model for smooth viewing and navigation of large 2D information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 10 (4), 447–458.

Ware, C. 2004. *Information visualization*. 2nd ed. Waltham, MA: Morgan Kaufmann.

Zoomorama [online]. Available from: www.zoomorama.com [Accessed 7 January 2010].

Zumobi [online]. Available from: www.zumobi.com [Accessed 7 January 2010].