CounterPoint: Creating Jazzy Interactive Presentations

Lance Good, Ben Bederson
Human Computer Interaction Lab
University of Maryland
College Park, MD 20742 USA
+1 301 405 7445
Email: {goodness,bederson}@cs.umd.edu

ABSTRACT

In this paper, we introduce Counterpoint, a zooming presentation tool. CounterPoint supports the construction of slide show style presentations with content arrangement in a 2.5D space. As with other Zoomable User Interfaces, CounterPoint provides animated navigations as transitions through this space. Our tool also supports both automated sequential paths and interactive navigation through the presentation. Multiple paths may also be defined within a single presentation space. This paper describes the functionality of our tool, some implementation details, and potential benefits of CounterPoint over more traditional slide show tools.

Keywords

Interactive presentations, Zoomable User Interfaces (ZUIs), spatial hypertext, dynamic paths, slide shows.

INTRODUCTION

The model for hypertext on the World Wide Web has been one of user isolation. An author creates a web page and many users view that web page in relative isolation from the author.

A more recent variation on this traditional model is spatial hypertext. Spatial hypertext is ideal in situations where the line between authors and users is blurred [12]. Here documents may not have well defined nodes and links and may only represent the current state of the authors understanding of a collection of information. In most existing spatial hypertext systems a primary goal is to foster emergent structure in document creation, avoiding the restrictions imposed by strictly defined organizations and links.

We have created CounterPoint, a tool combining elements from both traditional and spatial hypertext, for authoring interactive slide show presentations. In slide show presentations, the users are both the author, who needs to operate the slide show equipment, and the audience, who needs to understand the presentation content.

One of the features of hypertext that has made it a highly successful media for the World Wide Web is its fundamentally dynamic nature. The node-link format of hypertext allows users to dynamically tailor a reading/browsing session to suit their current interests. In this regard, current linear software presentation tools, such as Microsoft PowerPoint [15], are fundamentally limiting. While these tools greatly simplify the creation of linear slide shows, they do not facilitate presentation-time modifications to this linear path based on audience feedback, time constraints, or other factors.

Clearly traditional hypertext ideas, such as underlined links, could be used to solve this issue for presentations. However, these ideas have their own drawbacks, primarily audience disorientation (see [6] for example). In contrast, CounterPoint provides animated transitions in a spatial hypertext environment to support both audience orientation and interactive presentations.

A similar drawback to current presentation software is the coupling of presentation content and presentation ordering. This coupling often forces authors to create different presentations for different audiences on a given topic, despite only minute differences in content between presentations. CounterPoint, again borrowing from hypertext concepts, supports multiple scripted paths through the same presentation space. This allows an author to create a single presentation space for a given topic with multiple paths, each specifically tailored for a particular audience.

Our presentation tool was built in Jazz [3], a Java toolkit for Zoomable User Interfaces (ZUIs). Jazz offers animated transitions within a large two-dimensional surface where information is displayed at different scales. Consequently, CounterPoint allows for the organization of presentation content in freeform two and a half dimensional spatial arrangements.

Because it uses a spatial layout of presentation material, CounterPoint can potentially exploit humans' natural spatial abilities during a presentation. Likewise, the use of animated transitions between spatial locations during the presentation in CounterPoint may further lend itself to spatial perception.

First, we will discuss related work in the areas of hypertext and presentation tools. We will then describe CounterPoint and its implementation. We will also list some potential benefits of CounterPoint over previous slide show presentation tools. Finally, we will conclude and suggest areas for future work.

PREVIOUS WORK

This work came out of many years of ongoing research into ZUIs and the actual use of ZUIs for presentations. As such, it builds primarily on the results and experiences gained from both the Jazz [3] and Pad++ [2] systems.

Our system is perhaps most similar in spirit to VIKI, a spatial hypertext tool for supporting emergent structure during authoring [13]. One particularly relevant application of VIKI was its use in gathering and organizing content for educational presentation on the web [18]. Here preexisting web content and annotations were combined to create directed paths through collections of related information.

CounterPoint is similar to VIKI in that it involves spatially structuring information. However, because they are displayed to the audience, the structures created in CounterPoint are an end in themselves rather than a representation of the author's current understanding. Moreover, paths in CounterPoint are animated transitions through the author's explicitly defined spatial layouts whereas in VIKI the spatial layout defined the path itself [18]. Further, authoring in CounterPoint differs from authoring in VIKI because it involves the complexities of manipulating objects at multiple scales.

Directed paths through hypertext documents, similar to those available in CounterPoint, have also been explored in other settings. Some of the earliest work in hypertext paths was Zellweger's Scripted Documents [23,24]. Scripted Documents allowed the author to define timed traversals through a collection of documents with specifiable actions performed at each stop in the traversal. The "Audio-visual presentation" application of scripts described in [23] closely resembles our use of scripted paths in CounterPoint.

Trigg's Guided Tours and Tabletops also defined a hypertext path authoring and navigation tool [22]. This system provided tools for creating a collection of "tabletops," each of which contained a spatial arrangement of hypertext documents. An author could then define arbitrary paths through these tabletops with any number of available branches at each point in the path.

CounterPoint resembles Trigg's system in several aspects. CounterPoint paths are similar to Trigg's in that they combine both scripted and dynamic components. However, the dynamic changes available on a scripted path in CounterPoint do not have to be specified when the path is created. Second, CounterPoint and Guided Tours both allow for navigation through collections of spatially arranged objects. However, CounterPoint presents data in a single continuous space whereas Guided Tours supports sets of disjoint spatial arrangements. Finally, CounterPoint also implements standard hypertext style visited colorings similar to those available in Guided Tours.

Using hypertext as a slide show presentation tool is also itself not a new idea. One such example of the use of hypertext in this context can be found in [14]. While this use of traditional hypertext can facilitate better reuse and interconnection of related material, it can also suffer from the traditional hypertext problem of audience disorientation. CounterPoint tries to alleviate this disorientation through the use of animated transitions in a two and half dimensional virtual space.

Another tool suggested for interactive slide show presentations is Hyper Mochi Sheet [21]. Hyper Mochi Sheet employs a multi-focus distortion-oriented view to display a hypertext network. During a presentation, the system automatically resizes nodes in the network based on the current user focus. While the multi-focus views allow it to show both focus and context, its non-deterministic nature makes it less desirable for the slide show setting where layouts and object sizes are often parameters of primary concern.

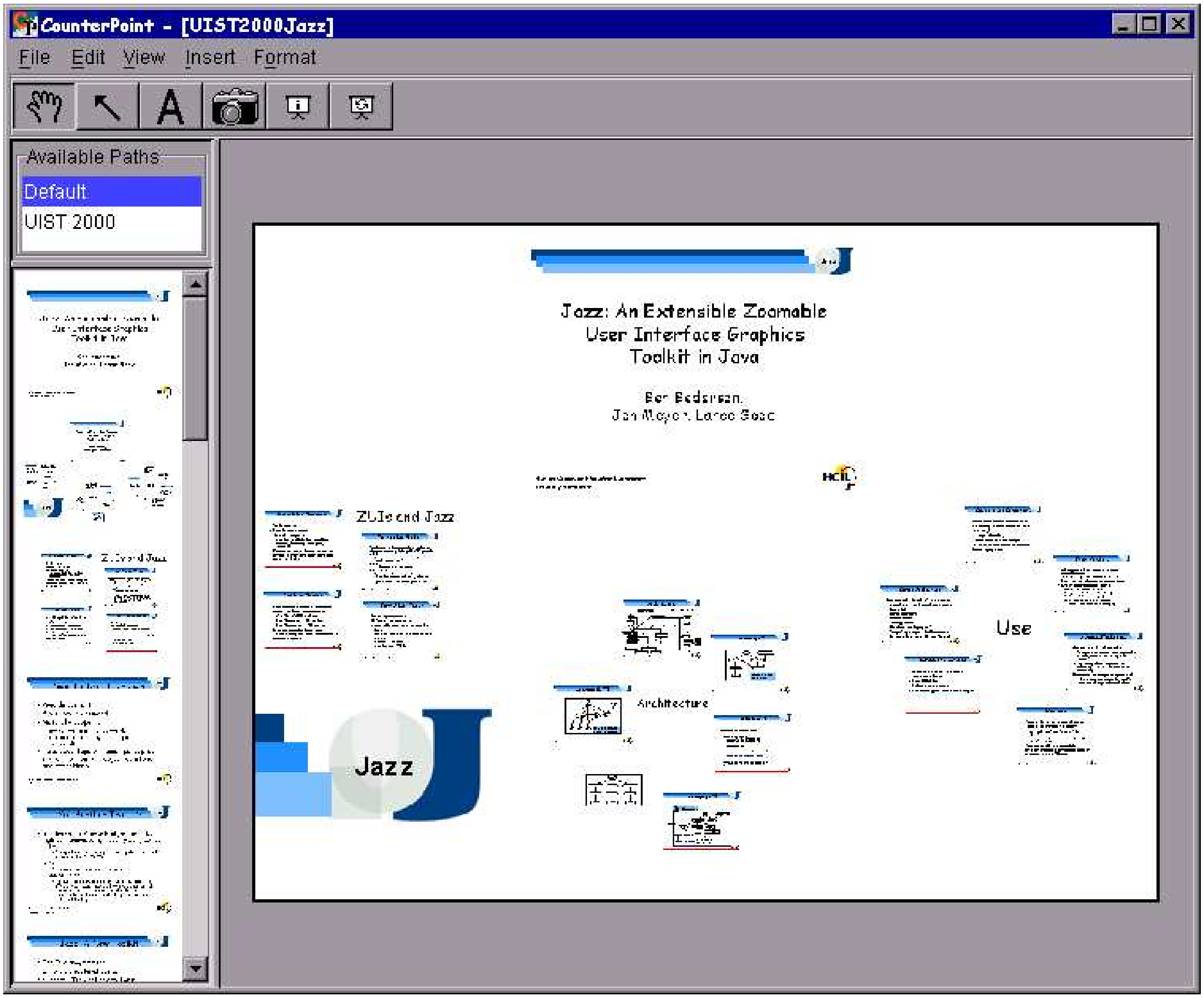
TOOL DESCRIPTION

CounterPoint is built on top of Jazz, a toolkit for building ZUIs. ZUIs are a technique for displaying information on an infinitely large two-dimensional plane. ZUIs allow users to change their view of this plane through panning and zooming to access more information than can typically be displayed on a single screen.

A fundamental characteristic of these types of zooming and panning operations in ZUIs is that they are animated. These types of animations give a sense of physical movement by mimicking such physical acts as sliding a paper on a table (panning), looking at a paper more closely for detail (zooming in), or holding a paper at a distance for more context (zooming out) [1].

In building CounterPoint, we also wanted to take advantage of existing presentation tools. Although there are currently a handful of commercial slide show presentation tools available, the tool that clearly dominates the market is Microsoft PowerPoint [15]. Therefore, to have the greatest potential impact on presentation authors, we chose to create CounterPoint as a plug-in to PowerPoint. This connection to PowerPoint not only allows for compatibility with existing PowerPoint documents, but also reduces the functionality needed in CounterPoint.

Consequently, the model we have envisioned for using CounterPoint begins in PowerPoint. An author begins by creating slides in PowerPoint in much the same manner as if the slides were actually to be used in PowerPoint. The author can use almost any of the available PowerPoint tools for creating presentation content. One of the primary sets of PowerPoint features that is currently unsupported in



Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java

Ben Bederson,
Jon Meyer, Lance Good

Figure 1 – A screen shot of CounterPoint in spatial arrangement mode. The panel on the right is used to modify the positions and magnifications of pre-authored slides. The panel on the left is used to edit scripted paths through the presentation.

CounterPoint is slide transitions. CounterPoint's animated navigation transitions are meant to replace any of the between slide transitions in PowerPoint. Still, there are some transitions within slides, such as incrementally revealing slide content, that we intend to support in future versions of CounterPoint.

We made an early decision not to try to replicate the functionality of PowerPoint in CounterPoint and to allow manipulations only at the slide level. While we feel that this was the best short-term solution, our long-term ideal for CounterPoint is to migrate the functionality of PowerPoint into CounterPoint (or vice-versa) for a finer granularity of control. In the mean time, we have added a single piece of this functionality that we found to be generally useful, namely simple text labels. Without this functionality, the author could only create a text label in CounterPoint by adding a new PowerPoint slide with the necessary text.

Once the slides have been created in PowerPoint, pressing a custom toolbar button starts CounterPoint and transmits the slide contents from PowerPoint to CounterPoint. After the slides have been transmitted, the author begins working in CounterPoint to create spatial arrangements for the slides and author paths through the presentation space.

The typical first step in creating a presentation in CounterPoint is to arrange the slides in the two dimensional space. Currently, we use simple tools for manipulating objects in this space similar to those found in PowerPoint, drawing programs, and previous zoomable demo programs (eg. Paddraw and Jazz HiNote).

More powerful tools for editing and arranging objects in this space are on our list for future research. One of our most important observations towards this end is that presentations are fundamentally hierarchical. For example, a presentation might have a title with four main points, each of which has 2 or 3 sub-points, etc. We expect that this same hierarchical nature can be found even at the slide level. Accordingly, we hope in the future to provide a hierarchical slide editor by which an author can specify a layout for a particular slide. The slide can then use layout to arrange any sub-slides in the hierarchy.

The next step in the authoring process is to create paths through the documents. Because paths through this space are not connected to the actual content, it is a simple matter to have multiple paths through the same space.

Paths are composed of two types of components. The first, more obvious type is the actual imported PowerPoint slide, which is inserted on the path to center the slide at full screen size. These PowerPoint slides can also occur multiple times in a single path. The second type of path component is a view onto a particular region of the space. These views are the more interesting path component as they allow the author to include views containing multiple slides and the structure of the presentation. Views are useful for showing an overview of the entire presentation or focused overviews of particular subsections of the presentation.

This current mechanism used to create these types of views is similar to taking a picture or creating a screen snapshot. First, the author navigates to the particular region of space to be added to the path. The author then presses the camera toolbar button (see Figure 1) and a new component, represented by a thumbnail image of the view, is added to the path.

While a one-dimensional representation of the current path is available in standard editing mode, CounterPoint also provides a two-dimensional path editor that mimics the functionality of PowerPoint's slide sorter. We believe that this will allow for the transfer of pre-existing PowerPoint skills since the concepts of path editing and slide sorting are so similar.

Some indication of the current path is also available while spatially arranging slides. When the mouse is positioned over a slide on the editing canvas, the system displays arrows indicating the locations reachable from the slide in the current path. While this feedback is not intended as a primary path-editing interface, it does give some subtle coupling between the two tasks.

Perhaps the most interesting and novel interactions occur in CounterPoint's presentation mode. The default behavior of sequentially stepping through one of the author's predefined paths is still available. This default behavior is achieved with the standard PowerPoint controls of left mouse button, the space bar, or right arrow key on the keyboard.

However, CounterPoint offers two modifications to this standard interaction. First, the presenter can press the up arrow key to get an overview of the entire space. In a future version of CounterPoint where hierarchical organization has been implemented, pressing the up arrow will zoom out a single level in the hierarchy.

A second interaction allows a presenter to dynamically navigate to a particular location in the presentation. First, the author must navigate to an overview where the target location is visible. Currently, this is typically achieved by zooming out to an overview of the entire presentation. Right clicking on the target slide then animates the view to that location.

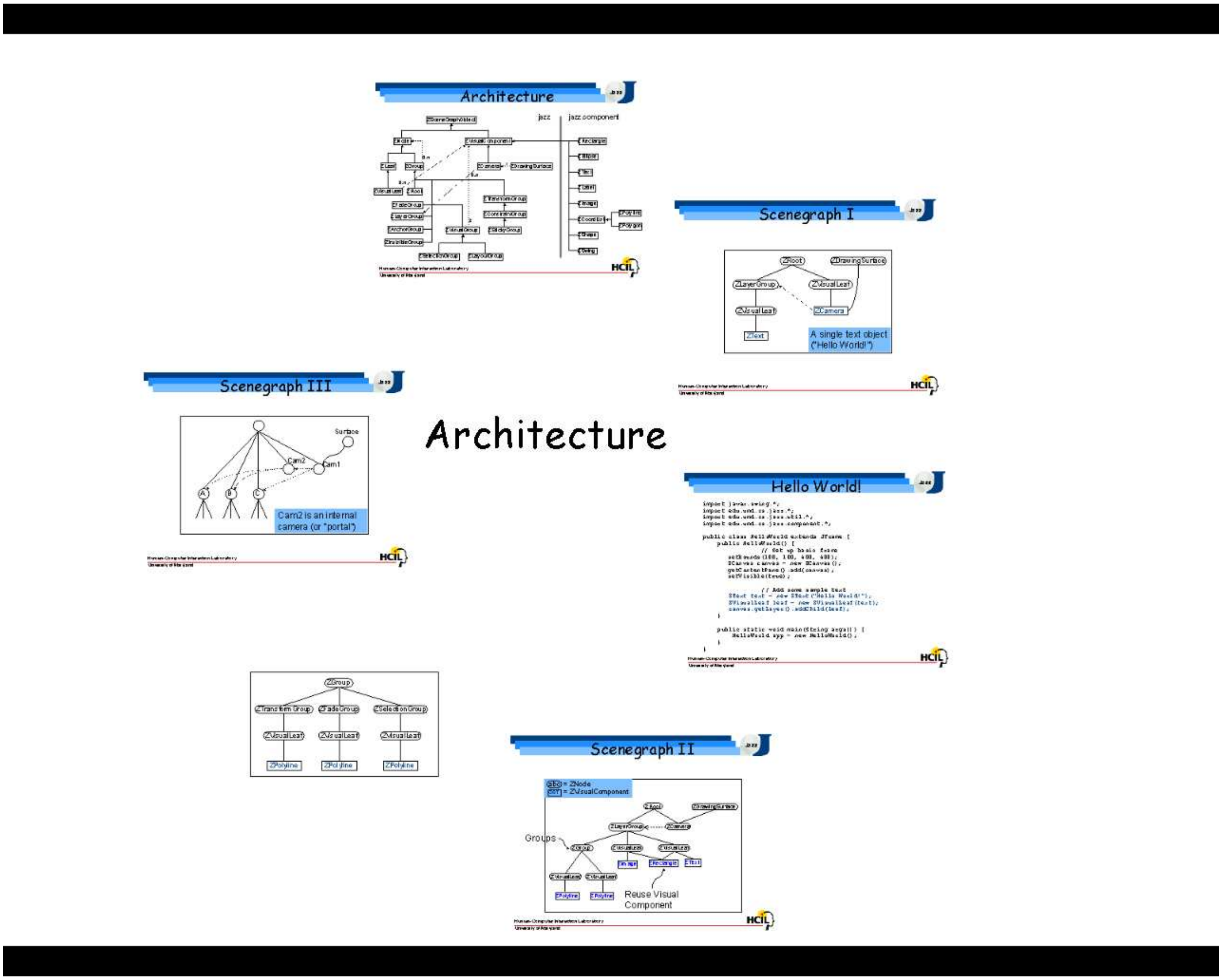
In cases where a presenter alters the presentation path by dynamically navigating to a slide, the system attempts to pick an appropriate point in the path from which to resume. In cases where the target slide appears in multiple places on the path, CounterPoint picks the path entry closest to the point at which the presenter deviated from the path. If the slide does not appear at all in the current path, the system does not try to infer a new path entry but rather resumes from the point at which the presenter deviated from the scripted path.

One other traditional hypertext element that we have added to CounterPoint to improve usability is visited colorings. CounterPoint provides modifiable slide border colorings to indicate which slides have been visited during a presentation. We have found these colorings to be useful both for the presenter and the audience for providing feedback as to which slides the presenter has visited and to give a sense of the overall progress of the presentation.

The CounterPoint portion of the presentation data, such as slide border colorings, slide spatial positions, and path orderings, are currently stored in a custom XML file residing in the same directory as the PowerPoint file. Because the format is XML, the file can be manually edited in a text editor in cases where the CounterPoint data has become out of sync with the PowerPoint presentation or for finer grain control over slide positions.

IMPLEMENTATION DETAILS

As previously mentioned, CounterPoint is implemented as a plug-in to PowerPoint. CounterPoint uses Visual Basic's COM hooks into PowerPoint to add a toolbar button and manipulate slide content. Because the majority of CounterPoint is built on top of Jazz in Java, one of the Visual Basic application's primary



```
import javax.swing.*;
import edu.umd.cs.jazz.*;
import edu.umd.cs.jazz.util.*;
import edu.umd.cs.jazz.component.*;

public class HelloWorld extends JFrame {
    public HelloWorld() {
                // Set up basic frame
        setBounds(100, 100, 400, 400);
        ZCanvas canvas = new ZCanvas();
        getContentPane().add(canvas);
        setVisible(true);

                // Add some sample text
        ZText text = new ZText("Hello World!");
        ZVisualLeaf leaf = new ZVisualLeaf(text);
        canvas.getLayer().addChild(leaf);
    }

    public static void main(String args[]) {
        HelloWorld app = new HelloWorld();
    }
}
```

Figure 2 – A screen shot of CounterPoint in presentation mode. Here, the presenter can alter pre-scripted traversals using various presentation-time interactions. Colored borders indicate slides already visited during the presentation.

responsibilities is to start a Java application when its toolbar button has been pressed. Its other major responsibility is to start a TCP/IP client by which it will communicate with this Java application.

Similarly, the first responsibility of the Java application is to create a TCP/IP server to communicate with the Visual Basic component. Once a connection has been established, the PowerPoint slide contents are transmitted to CounterPoint. For both efficiency and convenience reasons, the slide contents are not transmitted via the TCP/IP connection but are passed instead via the Windows clipboard.

This transfer of PowerPoint slide contents is possible because PowerPoint uses the Windows metafile format (ie. files with a list of drawing commands) for posting to the clipboard, rather than something similar to their proprietary file format. However, this metafile format also has positive performance implications for our application since Windows provides native support for metafile rendering.

Consequently, a third component of our application is implemented in Windows native code for managing and rendering Windows metafiles. Our Java code uses the Java Native Interface (JNI) to communicate with the native code and to switch between native and Java rendering as appropriate.

BENEFITS OF COUNTERPOINT

Aside from hypertext-style interactivity, CounterPoint presentations offer a number of potential advantages over traditional software slide show style presentations. Below we list some of these potential advantages.

Meaningful Spatial Structure

Research suggests that, in certain situations, the memory for data and the spatial location of that data are correlated [6]. For presentations, this implies that more meaningful spatial layouts may increase the retention of the underlying presentation content.

As a result, one of the CounterPoint's potential advantages over previous presentation tools is the ability to spatially organize data in two dimensions at different magnifications. This spatial layout may provide the audience with an additional attribute or memory pathway with which to recall the presentation content.

A related advantage of CounterPoint is that the structure or logical organization of the presentation can be incorporated into the spatial layout of the data. Then, because CounterPoint slide transitions animate through the space, this structure is itself revealed to the audience during the normal course of the presentation.

Revealing the structure of a presentation in this manner exhibits a design principle similar to what Norman calls "visibility"[11]. Likewise, Thuring et. al. suggest that presenting a hypertext document's structure to the audience is a necessary component "for reducing the mental effort of comprehension"[20].

This visual communication of structure has the potential to allow the audience to better understand the high-level concepts of a presentation and properly fit them into their own mental frameworks. In this way, the audience may be better able to incorporate the new knowledge with existing knowledge.

Sense of Semantic Distance

When moving between topics in a presentation in current slide show presentation software, the presenter must bear the burden of orienting the audience to the context change. In fact, two adjacent slides may contain no semantic relationship though positioned in close proximity in presentation sequence.

A different solution to which CounterPoint's spatial layout lends itself is indicating the semantic difference between two slides by their separation in the virtual presentation space. Transitions between these two slides will consequently portray this virtual separation through the distance traveled in the CounterPoint transition animations.

A similar example of this concept from hypermedia is the "warp coefficient" suggested by Kaplan and Moulthrop [10]. Here a number is associated with each link on a hypermedia page to indicate the semantic difference between the content of the current page and the link's destination page.

Relative Location Cues

A major problem in hypermedia that also seems to plague slide show presentations is disorientation.

In physical space, we find our way, in the absence of maps, using relative location cues. That is, we know where we are in the larger world based on local landmarks or other objects in our local surroundings [6][20].

One possible implementation of these relative location cues in the presentation setting is to include such cues on every presentation slide. For instance, each slide could contain a thumbnail representation of surrounding slides. This approach has several drawbacks, the most significant of which is reducing the amount of screen real estate available for actual data. As a result, this is not the approach used by CounterPoint.

Instead, the solution that CounterPoint adopts is to modify the appearance of the slide transition. Transitions in current slide show presentations provide neither information about their overall position in the presentation nor indicate their position relative to neighboring slides (other than, possibly the previous slide). Alternatively, because CounterPoint transitions give a sense of physical motion, they potentially offer the same type of relative location cues available in the physical world. Unfortunately, this solution also has trade offs, including increasing transition times, causing distraction, and potentially ignoring the preferences of certain users .

Improved Overview Support

Spatial, hierarchical overviews of hypermedia networks have been demonstrated to improve recall of overview titles when compared to both hypermedia with linear overviews and hypermedia without overviews [17]. This suggests that displaying a more overt and meaningful spatial overview during a presentation can increase the memorability and possibly the comprehensibility, of high-level presentation concepts.

Overviews are intrinsic in the nature of ZUIs. One of the previously mentioned capabilities of ZUIs is the ability to zoom out to get more context. As a result, it is always possible in CounterPoint to zoom out so that all presentation data, or certain localized subsets of that data, are in view. Whether these overviews convey meaningful information, of course, depends on the structure of the presentation. Nevertheless, this overview visualization capability exists at arbitrary magnifications in the presentation without any additional effort or input from the presenter.

Here again, we had the option of making the overview persistent, that is, visible on all slides at all times. However, we again chose not to do this because of the screen real estate it would sacrifice.

Natural Sense of Presentation Progress

A related deficiency of current presentation software tools is that they provide no notion of presentation progress for the audience. In earlier physical presentation media, this issue may have been addressed by the physical size of the stack of remaining overhead transparencies or the remaining slides in the slide projector carousel. Unfortunately, current presentation software tools provide little or no built-in support for conveying this important information to the audience.

In constrast, if the various pieces of CounterPoint's spatial metaphor function properly, such as overviews and relative location cues, a sense of presentation progress may naturally follow. However, CounterPoint also provides a more explicit indicator of progress by visually altering visited slides. We have found the combination of these implicit and explicit progress indicators generally effective at conveying progress.

Inherently Hierarchical

One of the fundamental structures used in the presentation setting is the hierarchy. Hierarchies are a natural format for organizing data as they allow topics to be recursively subdivided into increasingly smaller units of information. Many current presentation tools have recognized the importance of hierarchies in presentations and made them the primary structure for data layout. Nonetheless, these hierarchical capabilities are usually applied to organizing information within slides, while the organization of the slides themselves usually remains linear.

Moreover, though hierarchical language often metaphorically draws on the language of spatial objects, trees for example, the depiction of these hierarchies often approaches linearity. These linear representations can be observed in many outline editors such as those found in many presentation tools.

ZUIs facilitate a more spatial portrayal of hierarchies. Instead of depicting hierarchy levels through indentation, as is frequently done, ZUIs can present hierarchies in a format that more closely approximates a 2D represenation of a tree. Alternatively, ZUIs allow for visually distinguishing hierarchy levels by placing them at varying levels of scale or magnification. This change in magnification can naturally vary with the level of the hierarchy.

Eliminating Jumping During Slide Transitions

Incoherent transitions have been mentioned as a source of increased cognitive load in using hypertexts [17]. Consequently, we believe the greatest area for potential improvement in current presentation software is the slide transition. While most current presentation software implementations provide various forms of animated transition, these animations generally provide no useful information about the underlying data. More importantly, the most commonly employed transition is undoubtedly the most basic, namely one slide instantaneously replacing another.

A better slide transition would provide some insight into the relationship of the source slide to the destination slide. More precisely, a slide transition should prevent the audience from becoming disoriented by expressly relating the source and destination to their surrounding context. The default slide transition in current presentation tools requires audience members to reorient themselves before each slide and relate the current slide contents to higher-level concepts.

As already mentioned, CounterPoint implements slide transitions as animated viewpoint navigations through the presentation space. As such, these animations are able to display the changing context as the system transitions from one point in the 2D space to another.

Although the actual benefits of viewpoint animation still require further investigation, initial research indicates that these animations are beneficial for learning spatial organizations and data relations [1]. This study further suggests that viewpoint animations allow for a more constant understanding of object positions and relationships than viewpoint transitions without animation.

Research also indicates that animation may improve long-term understanding of presented material. This improvement was most profoundly observed in those with low spatial abilities [9].

Moreover, user responses indicate a subjective preference for animated systems over non-animated systems [7]. As user preference is a recognized quantitative measure of software usability [19], these preferences constitute a valid potential improvement to current presentation tools.

One of the biggest risks associated with animations is the time consumed by presenting extra intermediate frames during a transition. However, research also indicates that the extra time spent on animation does not result in longer task completion times [1], which relates directly to comprehension time.

Mixed Cognitive Encoding

The most frequent use of a presentation tool occurs in combination with a presenter's oral discourse. Hence, the audience receives, usually simultaneously, visual input from the presentation tool and verbal input from the presenter.

An ideal presentation tool would be considerate of this multi-modal input and exercise, whenever possible, different cognitive resources than the coincident verbal input. This strategy could not only reduce the cognitive demand on the audience but also improve audience comprehension of both forms of data.

Psychology hypotheses suggest that spatial and verbal information are encoded separately in memory (see research summarized in [16]). A presentation may therefore exercise a larger portion of the memory resources of the audience if it combines both verbal and spatial forms of data.

Robinson et al, performed research into this phenomenon by comparing graphic organizers and concept maps with linear lists and outlines [16]. Graphic organizers and concept maps are simply graphical layouts of text, for example, tables and flowcharts. They suggest that the information in the graphic organizers and content maps is encoded more spatially than the information in linear lists and outlines.

The spatial organization of data in CounterPoint, though unconstrained, lends itself to structures similar to graphic organizers and content maps. As a result, CounterPoint allows for spatial memory encoding of the presentation data. Combining this approach with the verbal encoding of the oral discourse may potentially reduce the audience verbal load and increase the retention of the presented data.

Creative Control

Because it supports the arbitrary arrangement of existing presentation slides, CounterPoint offers an additional degree of creative freedom over current presentation tools. Additionally, unlike other novel user interface approaches such as [21], CounterPoint offers deterministic control over presentation layouts and transitions. This type of direct control ensures predictability, which authors are likely to expect for presentations.

CONCLUSION

In this paper, we describe CounterPoint, a tool for novel slide show presentations. CounterPoint presentations combine scripted paths and traditional hypertext-style interactions with two and half dimensional spatial arrangements. One of the characteristic features of ZUIs that distinguishes CounterPoint from previous presentation tools is its use of animated spatial transitions.

We have also suggested several potential advantages of CounterPoint presentations over more traditional slide show presentations. However, future empirical studies are needed to verify these advantages.

FUTURE WORK

Our future work will mainly focus on better support for creating spatial arrangements. As previously mentioned, one of the main components of this work will be to create tools for more easily authoring hierarchical slide layouts. We also intend to design a set of zoomable and projector-friendly layout templates. Lastly, we hope to create improved tools for awareness and navigation in 2.5D spaces.

A somewhat different area we intend to explore is designing techniques for authoring semantic zooming transitions. This will empower authors to create presentations that change their appearance based on the presentation's current magnification.

We expect that our future work in these areas will likely draw on the lessons learned from such systems as Alice [4], DataSplash [5], MUSE [8], and VIKI [13].

ACKNOWLEDGMENTS

We would like to thank Allison Druin and Maria Jump for their early work on CounterPoint.

REFERENCES

1. Bederson, B., Bolman, A. Does Animation Help Users Build Mental Maps of Spatial Information. *Proceedings of the Information Visualization Symposium (InfoVis 99)* New York: IEEE, 1999.
2. Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, pp. 3-31.
3. Bederson, B., Meyer, J., Good, L. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *Proceedings of UIST 2000* (San Diego CA, Nov 2000), pp. 171-180, http://www.cs.umd.edu/hcil/jazz.
4. Conway, M., et. al. Alice: Lessons Learned from Building a 3D System for Novices. *Proceedings of CHI 2000* (The Hague Amsterdam, Apr 2000), pp. 486-493.
5. DataSplash: http://datasplash.cs.berkeley.edu
6. Dillon, A., McKnight, C. Richardson, J., Space - The Final Chapter or Why Physical Representations are not Semantic Intentions. In C. McKnight, A. Dillon, and J. Richardson, Eds. *Hypertext - A Psychological Perspective*. Ellis Horwood, New York: IEEE, 1999.
7. Donskoy, M., & Kaptelinin, V. Window Navigation With and Without Animation: A Comparison of Scroll Bars, Zoom, and Fisheye View. *In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 279-280.
8. Furnas, G., Zhang, X. MuSE: A Multiscale Editor. *Proceedings of UIST '98* (San Francisco CA, 1998), pp. 107-116.
9. Hays, T. Spatial Abilities and the Effects of Compuer Animation on Short-Term and Long-Term Comprehension, Journal of Educational Computing Research, 14(2), pp. 139-155, 1996.
10. Kaplan, N., Moulthrop, S. Where No Mind Has Gone Before: Ontological Design for Virtual Spaces. In *Proceedings of the European Conference on Hypermedia Technologies* (Sep 18-23, Edinburgh, Scotland). ACM, New York, pp. 206-216, 1994.
11. Norman, D. *The Psychology of Everyday Things*. Basic Books, New York, 1988.

12. Marshall, C., Shipman, F. Spatial Hypertext: Designing for Change. *Commun. ACM 38*, 8, (Aug 1995), pp. 88-97.

13. Marshall, C., Shipman, F., Coombs, J. VIKI: Spatial Hypertext Supporting Emergent Structure. *Proceedings of ECHT '94* (Edinburgh Scotland, Sep 1994), pp. 13-23.

14. Moore, T. Active Use of Hypertext to Aid Learning and Classroom Instruction. *SIGCSE Bulletin*, 27(4), pp. 741-750, 1999.

15. PowerPoint. http://www.microsoft.com/office/powerpoint

16. Robinson, D., Robinson, S., Katayama, A. When Words are Represented in Memory Like Pictures: Evidence for Spatial Encoding of Study Materials. *Contemporary Educational Psychology*, 24, pp. 38-54, 1999.

17. Rouet, J., Levonen, J., Dillon, A., Spiro, R. *Hypertext and Cognition.* Lawrence Erlbaum Associates, New Jersey, 1996.

18. Shipman, F., Furuta, R., Marshall, C. Generating Web-based Presentations in Spatial Hypertext. *Proceedings of IUI '97* (Orlando FL, 1997), pp. 71-78.

19. Shneiderman, B., *Designing the User Interface: Strategies for Effective Human Computer Interaction*. Addison-Wesley, Massachusetts, 1998.

20. Thüring, M., Hannemann, J., Haake. Hypermedia and Cognition: Designing for Comprehension. *Commun. ACM*, 38(8), pp. 57-66, Aug 1995.

21. Toyoda, M., Shibayama, E. Hyper Mochi Sheet: A Predictive Focusing Interface for Navigating and Editing Nested Networks through a Multi-focus Distortion-Oriented View. *Proceedings of CHI '99* (Pittsburgh PA, May 1999), pp. 504-511.

22. Trigg, R. Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment. *ACM Transactions on Office Information Systems*, 6, 4 (October, 1988), pp. 398-414.

23. Zellweger, P. Active paths through multimedia documents. In J.C. van Vliet (editor), *Document Manipulation and Typography*, Proc. Int'l Conference on Electronic Publishing, Document Manipulation and Typography, Nice (France), April 20-22, 1988. Cambridge University Press, 1988, 19-34.

24. Zellweger, P. Scripted Documents: A Hypermedia Path Mechanism. *Proceedings of Hypertext '89* (Pittsburgh PA, Nov 1989), pp. 1-14.