Working Paper Series
ISSN 1170-487X

DEEPDOCUMENT: USE OF A MULTI-LAYERED DISPLAY TO PROVIDE CONTEXT AWARENESS IN TEXT EDITING

Masood Masoodian, Sam McKoy, Bill Rogers and David Ware

Working Paper: 05/2004
May 2004

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

DeepDocument: Use of a Multi-Layered Display to Provide Context Awareness in Text Editing

Masood Masoodian, Sam McKoy, Bill Rogers, David Ware

Department of Computer Science
The University of Waikato
Hamilton, New Zealand
{masood, coms0108}@cs.waikato.ac.nz

Abstract

The most commonly used view in word processing software shows only the paragraphs of text immediately adjacent to the cursor position. Generally this is appropriate, for example when composing a single paragraph. However, when reviewing or working on the layout of a document it is necessary to establish awareness of current text in the context of the document as a whole. This can be done by scrolling or zooming, but when doing so, focus is easily lost and hard to regain. Furthermore, in a collaborative editing/review setting it is not only necessary for each user to understand their own context, but also to have an awareness of the contexts of the other participants. Although systems have been developed that provide awareness in collaborative settings, they usually rely on multiple windows, which use valuable screen real-estate.

We have developed a system called DeepDocument using a two-layered LCD display in which both focussed and document-wide views are presented simultaneously. The overview is shown on the rear display and the focussed view on the front, maintaining full screen size for each. The physical separation of the layers takes advantage of human depth perception capabilities to allow users to perceive the views independently without having to redirect their gaze. DeepDocument has been written as an extension to Microsoft Word™. It also includes awareness features to track focus positions for both single and multiple users.

Keywords: Multi-layered display, context awareness, collaborative awareness, CSCW, text editing, word processing, Deep Video™, Microsoft Word™.

1. Introduction

Computers are frequently used to display and edit large documents. Limitations in screen size and resolution restrict users' options in working with such documents. They must choose between close-up views of parts of a document, in which it is possible to read detailed information or perform precise editing operations, and zoomed-out overviews in which the document can be managed as a whole, but then the resolution is too coarse for information access. Users are, therefore, faced with either the cognitive load of remembering context whist working on detail views; or of having to navigate between detail and overview viewing modes, with the associated difficulties of orientation and focus finding that view changing implies.

The challenge for interface designers is to develop systems in which users can see enough detail to comfortably work with large documents, whilst supporting them in retaining an awareness of the context in which that detail lies. This is an important issue which has been identified by various researchers who have proposed numerous solutions to the problem. A few of the more novel approaches are briefly reviewed in the next section, where we also discuss some of their limitations. We have developed a solution which we believe has merits for a range of word processing scenarios. In this paper we discuss the design of our prototype system called DeepDocument (DD), its extension to a CSCW context, and our plans for extending and evaluating this system.

2. Related Work

The users of software applications which provide a single view of their document at any given time, are generally able to navigate between detailed (focused) view and overview representations of the document. This limitation of single often causes various difficulties. For instance, navigation may require several command actions, and therefore, distracting attention from the substantive document editing task. This is especially problematic in moving to a detail view because exact image positioning may be important for convenient editing and viewing. Users must reorient themselves after each view change. Reorientation is sometimes made easier by animating fixed viewing transitions, or allowing smooth interactive control of zooming and panning (for example Maya [1]). However, there is a trade-off here. For example, issuing the zooming commands distracts the user from the document task, which means that what is gained in retaining orientation may be lost in distraction. The users are required to remember the overall context when in a detail view, but when editing a document that may change in size, they can remain unaware of global consequences of editing actions until they explicitly 'go out and look'.

The most obvious solution to this problem is to provide the possibility of multiple windows (views) on the document. Small windows can share the workspace on an equal basis, although this is rarely satisfactory because available screen area is usually only just adequate for a single detail view. Nevertheless, this does address each of the issues raised above. The user does not have to navigate once suitable views have been established. However, reorientation when the user switches attention from one window to another is still an issue (but at least the views are not changing). Also, while the context is always available, a new problem arises in that the users must somehow maintain their understanding of the mapping between views.

The window size problem can be avoided by using a virtual desktop (e.g. Linux KDE) which allows quick switching between alternative screen sized views. This solution minimizes the navigation overhead, but reintroduces the disadvantages of seeing only one view at a time. Multiple monitors allow a user to keep multiple screen sized windows simultaneously in view at the expense of desk space. This is becoming a popular option, particularly as the cost of flat panel displays comes down, and research has shown improved productivity in a range of applications [9, 10] when multiple displays are used. Cox et al note however [5] that "physical separation of overview from the viewport sometimes causes people to neglect the overview".

An alternative approach is to distort a single display. A range of techniques have been developed starting with Apperley et al's bifocal display [11]. The goal of this system was to present objects in a large information space in such a way as to allow manipulation of some parts (in focus), in the context of the entire space. The system proposes a central area of detail blending into border areas in which the remainder of the document is compressed, using a fisheye lens effect. This system requires only a single display and provides the user with a clear context for the detail area. The mapping between detail and overview is intuitive and immediate. In particular the navigation direction to any non-focus item is obvious. The disadvantage is the distortion of, and the significant fraction of screen area that must be devoted to, overview areas. Depending on the distortion geometry there can also be high processing costs and consequent difficulty in providing smooth manipulation. Bi-focal displays in the pure form are at their best in locating discrete objects in a two dimensional space. They are less satisfactory in providing a clear view of the finished appearance of a large document. Screen space tradeoffs are not dissimilar to those of multiple views. To obtain a good overview, screen area must be taken from the detail area. A simplification of the pure bifocal display used as a window manager is described in [6]. In this system a central fixed focus rectangle occupies a large fraction of the display surface. Windows that are dragged outside this space shrink to thumbnail size and can be 'parked' for later re-activation, allowing a user to establish a space of window representations. Logical extensions of this which use less space for the off focus objects are the City Lights system [13], in which the off focus items are represented simply as narrow bands in the borders of windows and Halo [2] in which off focus items again feature as annotations near the edges of an image. All of the bifocal systems present significantly distorted views of the non focus area. The latter forms are again more suitable for finding discrete objects than for visualizing a large document.

An interesting recent development is Baudisch et al's Focus+Context system [3, 4]. The idea here is to present the entirety of a large document on a large screen using a data projector, and to have a part of that screen operate at a high resolution. The user faces the high resolution area – a normal LCD display set into, and coplanar to, the large screen. When looking straight ahead they see the detail area in the centre of their field of view, with the remainder of the document extending out around to the periphery of their field of view. In situations where the large display is practicable this looks to be an ideal solution for many problems.

In multiple window systems mapping improvements can be obtained when software explicitly supports the detail/overview usage pattern. Miniature overviews (*radar* overviews) have been used in a number of applications, including games, to show location of a detail viewport in a larger document. Gutwin et al describe some radar views for a CSCW context in [8]. This mapping can be active, allowing the user to pan the detail area by manipulation of the overview. The obvious limitation of such a system is the level of accuracy that is possible with a small overview window.

Cox et al [5] report on a system that presented a radar style overview at full screen size as a 'transparent' overlay superimposed over the detail work area. Despite the "unusual way of working" this imposed, the authors report that users were able to comprehend and successfully use this system, switching focus between the overview and the detail layers. However, they also report that users sometimes forgot that they were working over two layers and tried to associate objects on separate layers. This system has many advantages. There is no need for navigation action to change view - all that is required is a change of attention. There must still be re-orientation, but hopefully to a familiar view. Both detail and overview have full screen windows providing maximum resolution. A new feature is the possibility of the user retaining a peripheral awareness of the overview whilst working on detail. For example they may notice unexpected changes in overall document structure whilst working on detail. The availability of new multi-layer displays, such as the one described below, has allowed us to devise a novel overlay transparency system based on similar ideas.

3. Deep Video™

The Deep Video™ [7] two-level display takes advantage of the transparency of LCD displays. Two LCD panels are mounted a few centimeters apart. Behind the back panel is a white surface uniformly illuminated with white light, as in a normal LCD display. If the front panel is programmed to 'white' it becomes transparent and anything displayed on the back panel is visible. Any black or coloured information placed on the front display appears floating in front of information on the back. For the viewer, head movement leads to the expected parallax effects.

A dual monitor video card is used to drive the two surfaces, so it is operable from a reasonably standard modern computer. The effect is a genuinely "3D" display, at least to the extent of two distinct layers of information. A typical use is to put data such as might occur in a "heads-up" display overlaid on some primary image. The physical separation of layers makes attending to one layer much easier than is the case with simulated layers on a conventional screen. When data on the front layer obscures data on the back, a small head movement often allows the rear data to be seen. The two layer display fits into the same desk space as a single layer display (it is a few centimeters thicker than a conventional LCD panel display).

4. System Design

We adopt a similar approach to that of Cox et al [5], in that we use overlaid full screen document views. We present the document overview on the rear panel of a Deep Video™ display, with detail view and controls on the front. Physical separation of the two panels considerably reduces the display ambiguity that occurs with layering on a single screen. The particular document editing problem we have addressed in our system is that of word processing applied to documents up to about twenty pages in length.

The goal of our research is to improve the contextual information provided to the user(s) of a word processor. In contrast to a paper document, an on-screen document tends to lack clues to show the overall structure of a document. For example, with a book, rough location is always obvious, and the thickness of the pages before and after the current point of focus can be felt. In our project we have chosen to concentrate on conference paper sized documents – up to twenty pages. If necessary such a document on paper can be spread out on a desk surface and viewed in its entirety.

There are several processes in document development. Depending on personal preference a writer may type some headings and come back to fill in the text later, maybe working backwards and forwards through the document. Alternatively they may start at the beginning and type until they reach the end. Identifiable processes include: planning; entering text; reviewing; correcting, revising, reorganizing. The first two processes are well supported by existing word processors. A screen will comfortably show two or three paragraphs and allow editing of text. Reading is well supported, so long as the reader works from beginning to end. However, jumping about is usually cumbersome in contrast to the experience of using paper. Frequently, one needs to refer back to some earlier point when reviewing a document. Although it is easy to scroll or search back, a user can easily lose their focus of attention.

There is no simple equivalent to keeping ones finger on a focus point and flicking to another page.

Furthermore, when revising and reorganizing a document, such as a conference paper, the author will often need to check various aspects of the overall layout. For instance, are figures placed close to the parts of the text that refer to them? Do references have the correct numbers? Are sections in a sensible order? Are the pages full, or has some image floated over a page boundary leaving a gap? This kind of process requires working back and forth between detail view and overview.

On printed paper various zooming options are available. The user can spread all the pages of a (reasonably small) document on a table and look at the overall pattern. It is also possible to look at a page as a whole, or to focus in on text with little more than a change of attention. On a computer, zooming is possible, but again cumbersome. When in zoomed in mode, the text is a in comfortable size to read, a few paragraphs are visible, and maybe the top or bottom of a page is also visible. However, images usually occupy too much space to allow clear appreciation of their textual context. Also, although it is possible to see an entire page with a large high definition monitor, or one that is built in portrait format, most monitors are generally smaller and do not provide a full page view in which reading text is comfortable. With smaller monitors it is possible to zoom out further and see several pages at once, but this requires a sequence of mouse actions.

To summarise, processes that are well supported by conventional word processing software are detail view reading and writing. Processes that are more poorly supported are those that involve changing view scale or position, relating current text to the document as a whole. Generally these are tasks in which detail must be viewed in its context. A conventional word processor provides very limited context information – just a scroll bar to show where the current screen of information lies in the whole document and in a small document, how large a fraction it comprises of the whole. Status information may also show current position and document size.

Document review, such as when writing a research paper, sometimes involves collaboration. For two people, remotely located, to be able to work together on a review, there is a need for both synchronised displays with shared pointers and also mutual location information when browsing the document independently.

Our system is based on the hypothesis: that it would be helpful to provide two views of a document: one being the zoomed-in detail view in which a few paragraphs of text can be easily read and edited; and the other being a zoomed-out view showing the entire document. With the possible exception of very large headings text will not be readable in the zoomed-out view. However the general layout will be clear; paragraphs, sections and images in particular will stand out.

In our system, the use of two layers leverages the human ability to focus at one depth, ignoring distractions on the other level. Any (rare) situation in which it is difficult to read front text because of distraction from the rear can be resolved by the parallax shift that results from slight head movement. Information at the back is peripherally available and attention can be directed to the back layer by simply altering the focus of the eye. No explicit command is required.

In order to be able to do user trials with our experimental system, particularly to allow tests on multi-page layout tasks we needed to support trial documents with images and an interesting variety of heading styles, fonts etc. For this reason we chose to develop our system around Microsoft Word™. The disadvantage of this choice is that we didn't have access to source code, so some implementation options were unavailable or awkward. For example the mapping between screen location and text was difficult and we couldn't layout multiple page displays in the way we would have preferred. The advantages of using MS Word are however considerable. It meant that user trials using a (largely) familiar user interface on documents already generated by users themselves will be possible.

4. DeepDocument implementation

The following screen shots show how our user interface is built. We make use of the fact the Microsoft Word™ can display multiple windows on the same document. The basis of the back panel display is a window in zoomed-out mode, showing our entire document. Figure 1.

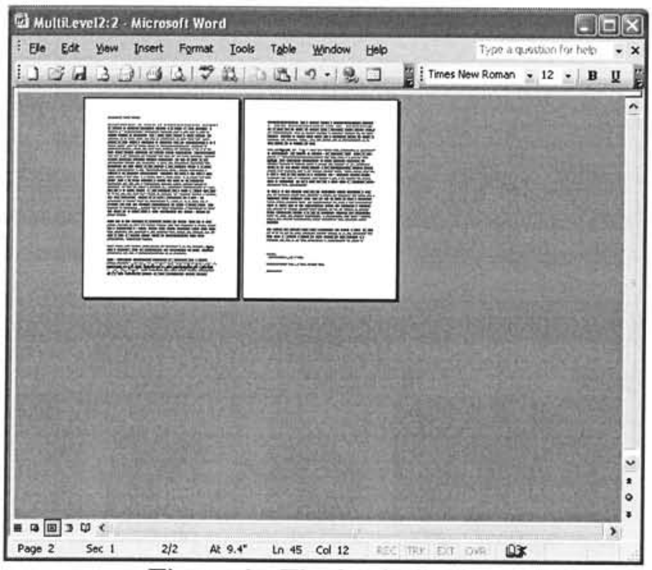

Figure 1. The back display

The front screen is a second window on the document, zoomed to a comfortable reading/editing level. Our system controls the zoom level of the back window. Layout of the front window can be altered by the user in the same way that they would normally operate Word, however a zoom setting of "Text Width" in which the text of the document fills the screen is most satisfactory, because this gives a white background to any area not covered by text or image, through which the back image can be most easily seen. A typical front window display is shown in Figure 2.

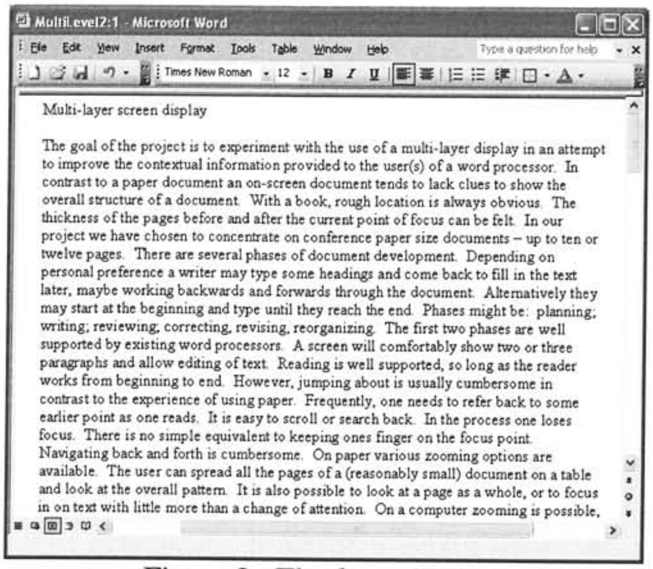

Figure 2: The front display.

If we put a window like Figure 2 on the front display of the DeepVideo unit and a window like Figure 1 at the back, the result is something like that shown in Figure 3.

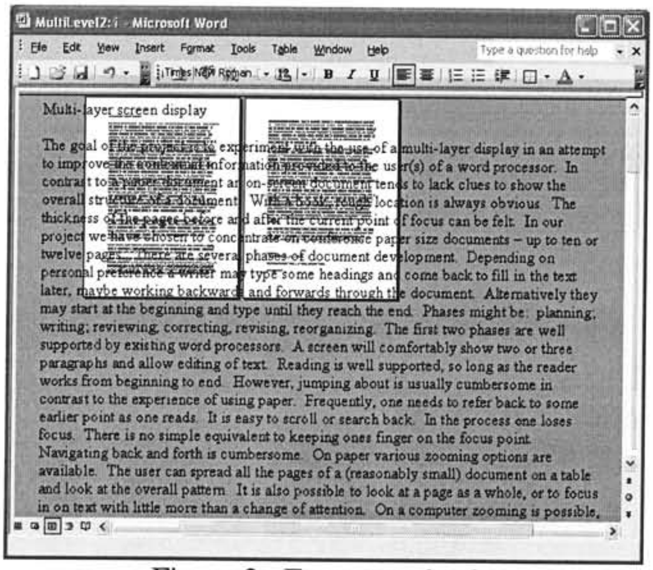

Figure 3: Front over back.

This has much of the style we want, but some significant problems. Firstly the back display has higher contrast than the front. It is easier to read front text on the real two level display unit than it is on the flat overlaid image of Figure 3, but still it was clear to us that a lower contrast in the rear image would be more consistent with it being the view that did not usually need to be examined in fine detail. In addition the physical requirements of the two level display must be taken into account. Clarity of the front display depends on a good level of light being let through the back display. If the back display is plain white all over, then the front display functions just like a single layer LCD screen. If the back display is black, then visibility is very poor on the front screen – just like a normal LCD display with its rear illumination turned off. To get the best effect in our system we require not only low contrast at the back, but also high brightness.

Secondly, both displays have menus, toolbars and scrollbars. These are simply confusing when overlaid. Our intention was to have the rear layer under automatic software. Controls and status information were not needed. Even rear scroll bars should be unnecessary as the intention is to display the whole document. Implementation options were to remove controls on the back display area, allowing more space for the zoomed-out view, or to leave them in place and cover them somehow. We adopted the latter option because it seemed best to retain the same division of control/data screen areas on both displays. It would be confusing to have back screen data underlying front screen controls. To ‘cover’ the rear controls our system places opaque white windows over them. (These are displayed on the back panel as normal overlay windows.) This solution can be implemented with minimal interference with MS Word. A disadvantage is that our solution assumes a particular configuration of Word tool bars. In principle it might be possible to interrogate Word for toolbar settings and decide on the appropriate geometry. We have not done this. A problem with any project of this kind is to decide on just how far the ‘parallel engineering’ will go. Our guideline has been to do enough to get a testable system.

Lowering the contrast and increasing the brightness of the back display was also possible by appropriate window manipulation. It was achieved by adding a new rear backdrop ‘window layer’ on the rear display panel. The new window is opaque and white. The zoomed-out Word window is alpha blended over it in a partially translucent form. The alpha blending level is selected to give a good degree of contrast reduction. The white rear window leads to a high level of brightness. Brightness level can be easily adjusted if necessary by altering the colour of the backdrop window. The general appearance is as shown in Figure 4. The overall structure of the entire display is shown in Figure 5.

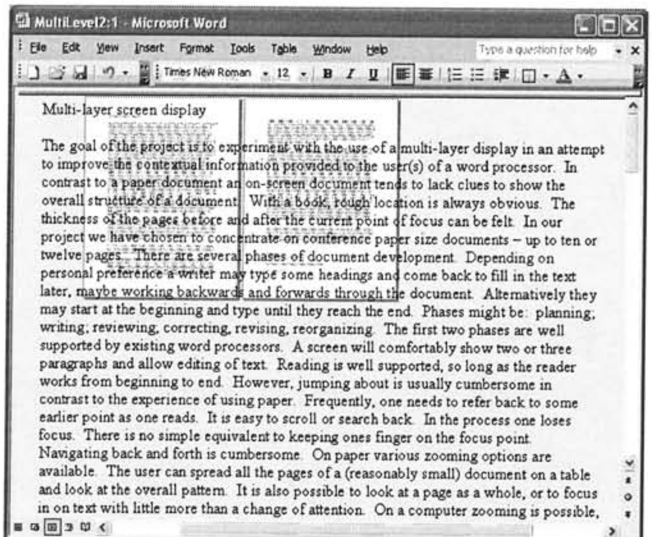

Figure 4. DeepDocument display appearance.

Once the display was constructed we were able to address awareness issues. The first need was to relate the front and back displays. The system we have used is to colour the area on the back display that is currently displayed on the front. This is implemented by placing one or two translucent windows over the appropriate area. One window suffices if the active area is wholly within a single page. Where the bottom of one page and the top of another is visible, two windows are placed over the back image to show the view area. This is not so visually satisfactory, but does accurately represent the display state. The position awareness window(s) dynamically follow activity on the front display.

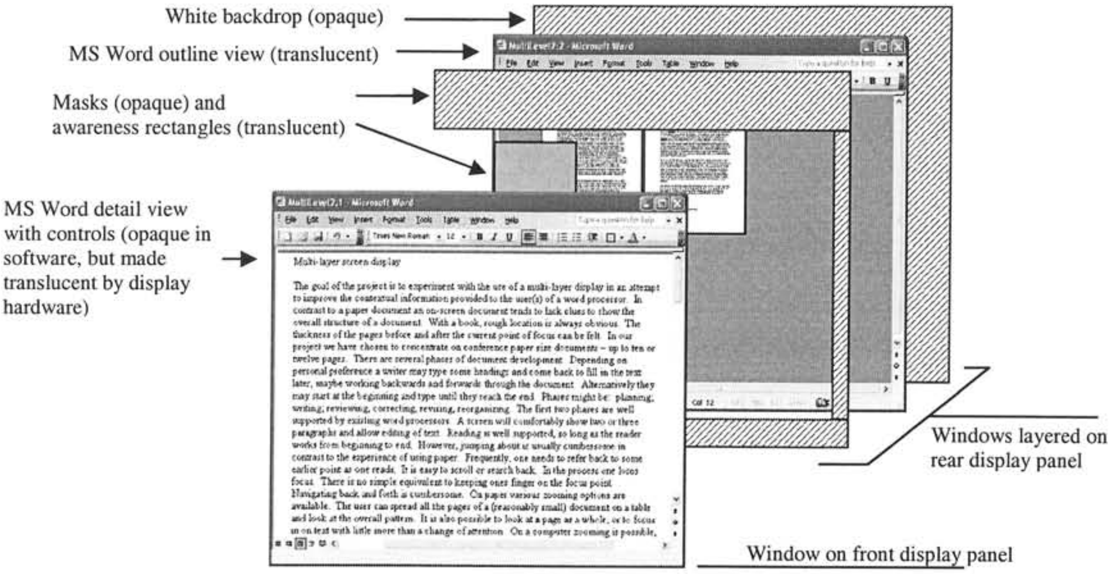

Figure 5. Display implementation

Implementation of the automatically arranged rear display meant that program control of MS Word was necessary. We took advantage of the fact that MS Word's internal object model is made available to external programs via a COM (Common Object Model) [12] interface. A session with our system must be started by running DeepDocument, selecting a document file to edit and pressing 'GO'. The program starts MS Word, loads the document, arranges the two views, and adds the additional windows required to complete the display configuration. DD maintains a control window throughout a session (usually kept minimized) to allow parameter adjustment, particularly the zoom level of and page arrangement on the overview display. However the intention is that the user should see the system as a normal instance of MS Word running on the front display screen and use it as such, taking advantage of the overview on the rear display as they wish.

In early testing it became clear that the users would expect to be able to navigate using the rear window. It was important to us not to change the normal operation of Word, which meant that no user interface control was available, except 'hot key's. We compromised and 'stole' the 'click scroll wheel' operation from MS Word. Clicking the scroll wheel is interpreted in DD as 'shift focus to centre on the area of the rear window currently below the mouse cursor'. This is not a very precise operation as the mouse cursor is displayed on the front screen, and parallax error may occur, depending on viewing position. In practice the error does not seem to matter.

MS Word itself maintains synchronization between the two screens. Updates made at the front automatically generate updates of the rear screen. DD monitors activity on the front screen for two purposes. One is to keep the location windows in the correct position on the back window. The second is to support remote collaboration. That is more complex and is described in the following section. Word can send notifications to the control program when certain events occur. However, it does not notify on ordinary text entry. In order to keep the position information current during typing therefore DD must poll Word frequently to check status (currently several times a second).

6. Collaboration

From the outset it was our intention to support collaborative document review and editing. The coloured rectangles that are used to show a single user where their front screen data lies on the rear screen are applicable for showing the focus area of remote partners. Different colours can be used to distinguish each person's area of interest. Overlaps result in overlapped translucent windows, and therefore show as more darkly shaded areas. Because the rectangles are windows it is also straightforward to place additional information on each if colour alone is not sufficient (as described for example in [8]).

DD supports collaborative editing. The control program of each participant observes updates (text insertion, formatting change, etc) and broadcasts them to collaborating partners. The collaboration architecture used is a server/client one. The clients report updates to the server which sends to all other collaborators. A server/client architecture normally has the advantage of being able to order updates. If each client sends updates to the server and only performs the action when the update request is received back, then all clients are guaranteed to receive the same sequence of updates, in the same order – thus keeping all clients documents the same, even if race conditions lead to some of the changes being not exactly as the users had intended. In our system this uniformity is not achieved. The reason is that a DD client only finds out about updates after they have been actioned. The result is that occasionally race conditions can lead to divergence of client documents. Experimentally this has only been observed when tests were especially designed to cause it. We assume that social conventions will usually prevent users from doing competitive updates in the same area of the document. Certainly, document review sessions are very difficult if more than one person is making changes at any given time. If any uncertainty arises the DD system has a 'resynchronize' command to restart clients on a common source document.

The DD server takes responsibility for holding the starting source document and also keeps a list of all editing operations performed. If a new client joins an active session, it is given a copy of the document and then passed all edits on the list, so as to bring it into synchronization with existing clients.

Supporting collaborative editing was the most difficult part of the implementation. The COM interface to Microsoft Word™ was never intended to support this kind of activity. Our implementation polls the undo buffer to identify completed editing actions. It is capable of processing many of the possible editing actions, but not all. A complete implementation would be a very large parallel engineering exercise. In its current state of implementation DD is capable of tracking text and image changes, and most common formatting changes. Some formatting changes are not mapped – for example, setting a document into a new pattern of columns. The resynchronize command can be used after such (rare) cases. Note also that it is only collaborative mapping that has this limitation. In the single user situation there are no such restrictions.

Location colouring allows users an overview of each other's locations. In addition, when two users share (partially or wholly) the same focus area, DD provides tele-pointers – remote copies of mouse cursors, tagged with the name of the user – allowing participants to communicate by gesture.

Finally DD provides a shared audio channel. This works most satisfactorily with headset earphones and microphone. At present we use a wired headset, but are investigating the possibility of using Bluetooth equipment.

To support comparative experimentation the DeepDocument software will operate on a single screen, using transparent overlays, as well as on the DeepVideo equipment

7. Conclusions and plans for further work

Although we have not evaluated our system, and planning to do so in the near future, our preliminary experiments with the system supports our expectation that the two-layer display would have significant advantages over a single-layer display with transparent overlays, as it is significantly easer to focus on each of the separate layers in a two-layer display. The parallax movement that results from small head movements, and the physical focus differences are very helpful. The rear layer is useful in a number of situations. Users do seem to use their 'geographical location' sense to develop a picture of document layout, and thus the 'lost image' problem is addressed. Even if an image doesn't go exactly where they want it, at least they can see where it went. The consequences of local reformatting changes are clearly apparent. This is especially clear when adding a few words leads to an image being pushed over a page boundary.

Our system is still under development and we plan to include additional features. The first of these is to provide a means of marking a focus position (leaving the area visible on the overlay screen) so that it can be easily returned to after moving around the document. Another feature to include would be a means of annotating documents. We have observed that collaborative editing becomes unsatisfactory when large changes are being made to a document. The situation is, not surprisingly, like that in programming code review. Participants in a session quickly loose their understanding of the document. It would be preferable to mark up required changes and have one person perform them offline. The third feature to add is a way of making some text on the overview display large enough to read. As the system stands, only large text such as the document title is visible. In contrast, images are usually clear, and provide the best navigation "landmarks". If some words, for instance the section headings, could be enlarged in the display, then those words would also provide landmarks.

References

1 Alias Systems division of Silicon Graphics Limited, Maya, *http://www.alias.com*

2 Patrick Baudisch , Ruth Rosenholtz, Halo: a technique for visualizing off-screen objects, *Proceedings of the conference on Human factors in computing systems*, p.481-488, April 05-10, 2003, Ft. Lauderdale, Florida, USA.

3 Patrick Baudisch , Nathaniel Good , Paul Stewart, Focus plus context screens: combining display technology with visualization techniques, *Proceedings of the 14th annual ACM symposium on User interface software and technology*, November 11-14, 2001, Orlando, Florida, USA.

4 Patrick Baudisch , Nathaniel Good , Victoria Bellotti , Pamela Schraedley, Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming, *Proceedings of the SIGCHI conference on Human factors in computing*

systems: Changing our world, changing ourselves, p. 259-266, April 20-25, 2002, Minneapolis, Minnesota, USA.

5 Donald A. Cox , Jasdeep S. Chugh , Carl Gutwin , Saul Greenberg, The usability of transparent overview layers, *CHI 98 conference summary on Human factors in computing systems*, p.301-302, April 18-23, 1998, Los Angeles, California, USA.

6 Mary Czerwinski, Human-Computer Interaction for Large Display Surfaces, *IEEE Symposium on Human Centric Computing Languages and Environments*, 2003, Auckland, New Zealand.

7 Deep Video Imaging, http://www.deepvideo.com

8 Carl Gutwin , Saul Greenberg , Mark Roseman, Workspace awareness support with radar views, *Conference companion on Human factors in computing systems: common ground*, p.210-211, April 13-18, 1996, Vancouver, British Columbia, Canada.

9 Rachna Luthra, Kerry Lorimer, Productivity and Multi-Screen Displays, *ATI Technologies and the University of Utah Report* summarized in *http://www.necmitsubishi.com/newsNew*, 2003

10 Darrell Norton, Seeing Double: An Unbiased View on the Benefits of Multiple Monitors, *http://dotnetjunkies.com/WebLog/darrell.norton/posts/3432.aspxm*, 2003

11 Spence, R. and Apperley, M.D. (1982). Data Base Navigation: An office environment for the professional. Reprinted in Card, S.K., Mackinlay, J.D. and Shneiderman, B. (eds.), 1999, *Readings in Information Visualization*. Morgan Kaufman, San Francisco, CA, USA.

12 Sara Williams and Charlie Kindel, The Component Object Model: A Technical Overview, *Microsoft Corporation White Paper*, October 1994

13 Polle T. Zellweger , Jock D. Mackinlay , Lance Good , Mark Stefik , Patrick Baudisch, City lights: contextual views in minimal space, *CHI '03 extended abstracts on Human factors in computer systems*, p. 838-839, April 05-10, 2003, Ft. Lauderdale, Florida, USA.