Usability Issues of Multiple-Layer Display Technology

Carey Bishop

University of Canterbury

Abstract

The recent development of the Multiple-Layer Display (or MLD) is unusual because unlike most inventions, the MLD was brought about by innovation, rather than as a solution to an existing problem. As such, little is known about its influence on usability and Human Computer Interaction (HCI). This paper is an investigation into the usability effects of the PureDepth Multiple-Layer Display, particularly in relation to map reading tasks. It also describes scenarios where the multiple-layer display can be put to best use.

This experiment has shown that the participants could easily determine the layer that an image was being displayed on. This suggests that the layers can be used in place of colour coding, and other techniques, as a way of adding semantic information to an image. Participants also had no trouble working with information that had a direct relationship between layers of the display. A Focus+Context interface for the MLD was also reviewed, but user opinion suggests that this implementation needs further refinement.

Contents

Acknowledgements

Many thanks to Dr. Tim Bell for his supervision and guidance throughout the year. Thanks also to the Bishop Julius Hall of Residence and HITLabNZ for their hospitality, and to the students of the University of Canterbury who volunteered their time. Finally, thanks to PureDepth [7] for supplying the multiple-layer display.

1 Introduction

Multiple-layer displays are a relatively new technology and, as such, little is known about usability and user performance with these devices. While there are already several reasons for purchasing the device, such as the ability to show multiple views on the one display, and to illustrate depth between layers, the developers of this device, PureDepth[1], would find it easier to market these devices if there was proven research to support their work. William Wong [9] has already done some research into the usability of multiple-layer displays. The aim of my experiment is expand on his research to provide some formal evaluation results, and to investigate scenarios that make good use of the display's capabilities.

A Multiple-Layer Display (or MLD) is built in a similar fashion to normal, single-layer displays. Rather than using a single LCD panel, two LCD panels are placed on top of each other, so that the user can see through the front layer to the layer behind. A small gap between the layers, typically about eight millimetres, provides the ability to render depth on the display. The fact that the layers are fixed, directly in front of each other, allows for related content to be displayed on separate layers to keep it visually separate, while still showing the relationship between the layers. In a mapping application, for instance, roads can be rendered on the front layer while railways, waterways and other features are rendered on the rear layer of the MLD. This allows users to see the relationship between the position of the roads and the rest of the landscape, while allowing them to distinguish between the layers through the use of parallax. Mapping tasks were chosen for this experiment, as maps are inherently layered documents and can easily be split into multiple logical layers.

[1]'PureDepth' and 'MLD' are registered trademarks of PureDepth, Inc. Ltd.

2 Background

The research has been broken down into two sections, those papers that are directly related to multiple-layer displays, and research from other areas of the field of Human Computer Interaction (HCI).

2.1 Multiple-Layer Display (MLD) Research

Wael Aboelsaadat and Ravin Balakrishnan [1] have devised and evaluated the results of Stroop tests between single-layer and multiple-layer displays. Their goal was to determine how the display performed in terms of visual interference. With the single-layer display, a colour name was displayed as a word in the middle of the screen, with either a rectangle behind the word or a line underneath the word, rendered in a different colour. In multiple-layer mode, the word was displayed on one layer and the rectangle or line was displayed on the other layer of the display. They found that, in almost all cases, users performed worse in the dual-layer display mode, than they did in single-layer mode. The only situation where the dual-layer display outperformed the single-layer display was when the saturation of the coloured rectangle was reduced so that the rectangle became almost transparent. In this case, it was easier for users for ignore the rectangle and focus only on the word

William Wong [9] has researched the display, and identified a number of properties of the display that may prove to increase (or decrease) usability. Among his positive conclusions were the ability to dynamically link content between the two layers, being able to have both the context and the focus in view, being able to spatially interact with objects on the screen, and improving information extraction. He also noted that multiple-layer displays could reduce legibility and distract the user from the task at hand. This paper is the basis for evaluation to prove these claims.

Masood Masoodian, Sam McKoy, Bill Rogers and David Ware [8] have created an application that uses the multiple-layer display to create a focus and context view of a word processing document. Their aim was to enhance the speed at which users could navigate the document, because they would always have the full document in view. The front layer of the display contained a standard editing window, with the text shown full size on a white background. This white background made the rear layer visible, because on the front layer of the MLD, white appears to be transparent. On the rear layer they displayed a multiple page preview of the document, and alpha blended it with white to reduce its opacity. They did not have the opportunity to perform a formal evaluation of this technology, but user opinion was that this system was easier to use than the standard setup of the word processing application.

2.2 Related Research

Patrick Baudisch and Carl Gutwin [2] have identified a technique called "multiblending" that has been proven to be better than alpha-blending when over-

Figure 1: An example of multi-blending on a single layer display to reduce the interference between tool palettes and an image, as it is being edited. [2]

lapping windows are displayed. Their technique uses contextual information to ensure that the outlines of the important objects, such as buttons, are always visible. Less important information areas, such as blank areas, are left transparent to show as much of the underlying layer as possible. This technique is illustrated in Figure 1.

Patrick Baudisch and Ruth Rosenholtz [3] have created a way to improve navigation to objects not shown in the current viewport. The layered model that they use may prove a useful application of this multiple-layer display. A circle (or 'halo') is drawn around the off-screen object with a radius such that the edge of the circle is displayed in the current viewport. As the viewport moves further away from the object, the radius of the circle increases to keep the halo in view. The increased radius results in a wider arc on the viewport. This arc provides a visual clue to the user as to both the direction in which the object is located and also the approximate distance. Their research concluded that the technique had been proven to be faster for users to navigate than when they had no visual aid.

Jean-Daniel Fekete and Michel Beaudouin-Lafon [6] have detailed how multiple layers can be used to provide advanced features within graphical editors. They also came up with a framework design that would make it easy for developers to add this functionality to their existing applications. By placing different objects on different layers, users can edit each object individually while still having each object in the same merged image. This is similar to the vector-based, layer support now available in programs like Corel Paint Shop Pro, where users can manipulate a number of dynamic vector objects on each layer, while still maintaining a view of the overall image.

Figure 2: The PureDepth MLD 3000 [5]

3 The Multiple-Layer Display

The PureDepth MLD 3000 that was used in this experiment, was a two-layer display with a gap of about eight millimetres between the panels. The panels ran at a resolution of 1280 by 1024 pixels, and had separate VGA connections to a dual-head graphics card. To the computer, the layers appear to be two separate, single layer LCD displays.

3.1 Developing Software for an MLD

Development for this project was done using the Microsoft Windows operating system, although almost any operating system would have been suitable. There is no need to install any special drivers to use the PureDepth MLD, because most modern operating systems have native support for multiple monitors. Microsoft Windows assumes by default that the two displays are side by side, so it maps the two layers of 1280x1024 into a single virtual desktop with a resolution of 2560x1024. Moving the mouse pointer to the far left and far right makes the pointer move between the two layers of the display. The PureDepth MLD comes with a utility that allows the user to move the cursor between the layers by clicking the middle mouse button, which makes moving between the layers easier and more intuitive.

Because the virtual location of the layers (in terms of pixel coordinates) can differ based on the way the operating system is configured, it is best to use the available APIs[2] to determine where the layers are located. Under Windows, this can be achieved with the `EnumDisplayMonitors` function. It is important to note that if the graphics card is configured for "Horizontal Span" mode, this function will only report one monitor. For the purpose of consistency, the software used here assumed that the primary monitor (the first monitor reported) was the front layer of the display. Unfortunately with the current hardware, there appears to be no way to distinguish the two layers through software, if the MLD happens to be connected the other way around.

To use standard desktop applications on the display, it was best to set the desktop background to white, to ensure that the colour would not interfere with the other windows. White areas appear transparent on the front layer so that the back layer can be viewed clearly, and likewise, white areas on the rear

[2]Application Programming Interfaces

layer allow all the light from the backlight to shine through and illuminate the front layer. One of the original test utilities rendered a white rectangle, of the same size as the main window, on the opposite layer to the main window but in the same position. This ensured that the main window was always visible independent of which layer it was being displayed on, and without being affected by other applications and images shown on the same display. When using an MLD, it is best to ensure that the graphics card driver is configured to prevent windows spanning multiple layers and to ensure that the taskbar only appears on the front layer. This seems to provide the most the most comfortable way to use the display in a standard desktop environment.

It is possible to configure the display to emulate a single layer display, by simply setting the rear layer to white, and rendering the content only on the front layer of the display. This is useful to provide support for user interfaces where the multiple-layer ability might hinder the view of the content.

3.2 Colour Model

It is interesting to note that the colour model of the PureDepth MLD is both additive and multiplicative. Traditionally, illuminated displays use the additive RGB colour model to make up a single colour by varying proportions of the core colours Red, Green and Blue (RGB). This is the basic colour model for computer monitors, televisions and video cameras, but the intriguing concept with the MLD is that while each layer independently uses the additive model, the colours between layers are effectively multiplied. Table 1 shows examples of multiplicative blending, while Figure 3 illustrates the differences between additive and multiplicative blending.

Consider a standalone LCD backlight producing white light. LCD panels appear transparent when energised, so imagine a single, energised LCD panel in front of the backlight. This device still appears to be white. Now, if the input to the LCD panel is reduced to half, the device will appear to be grey, halfway between black and white. With multiple-layer displays, this limits the colours that can be displayed on any foreground layers. In this scenario, the front layers have only half the original light output to work with, so even white (transparent) will appear grey. Likewise, any dark colours shown on the front layer may obscure information displayed on the rear layers.

Multiplicative colour blending is generally used by software developers when developing graphics rendering applications[4], as a way to blend lighting and texture layers to create a unique looking surface from generic images. This allows graphics applications and games to create a realistic looking environment from a small number of images. Figure 4 demonstrates how a simple brick tile image can be transformed to look like a non-uniform texture, through multiplicative blending.

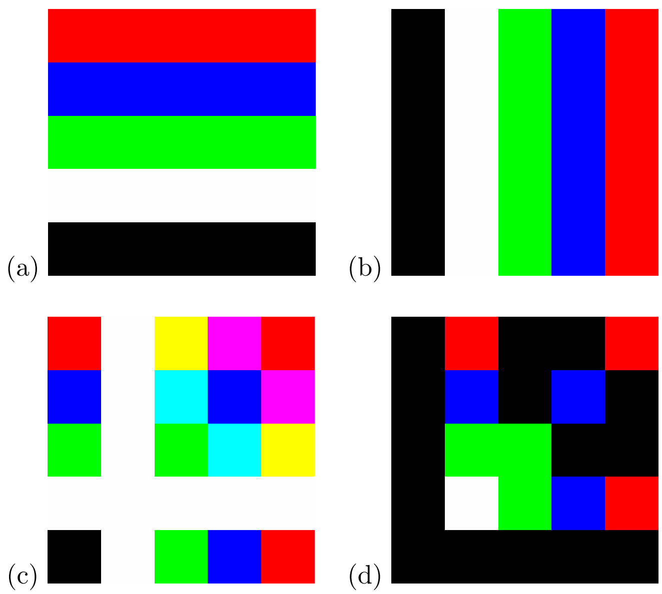
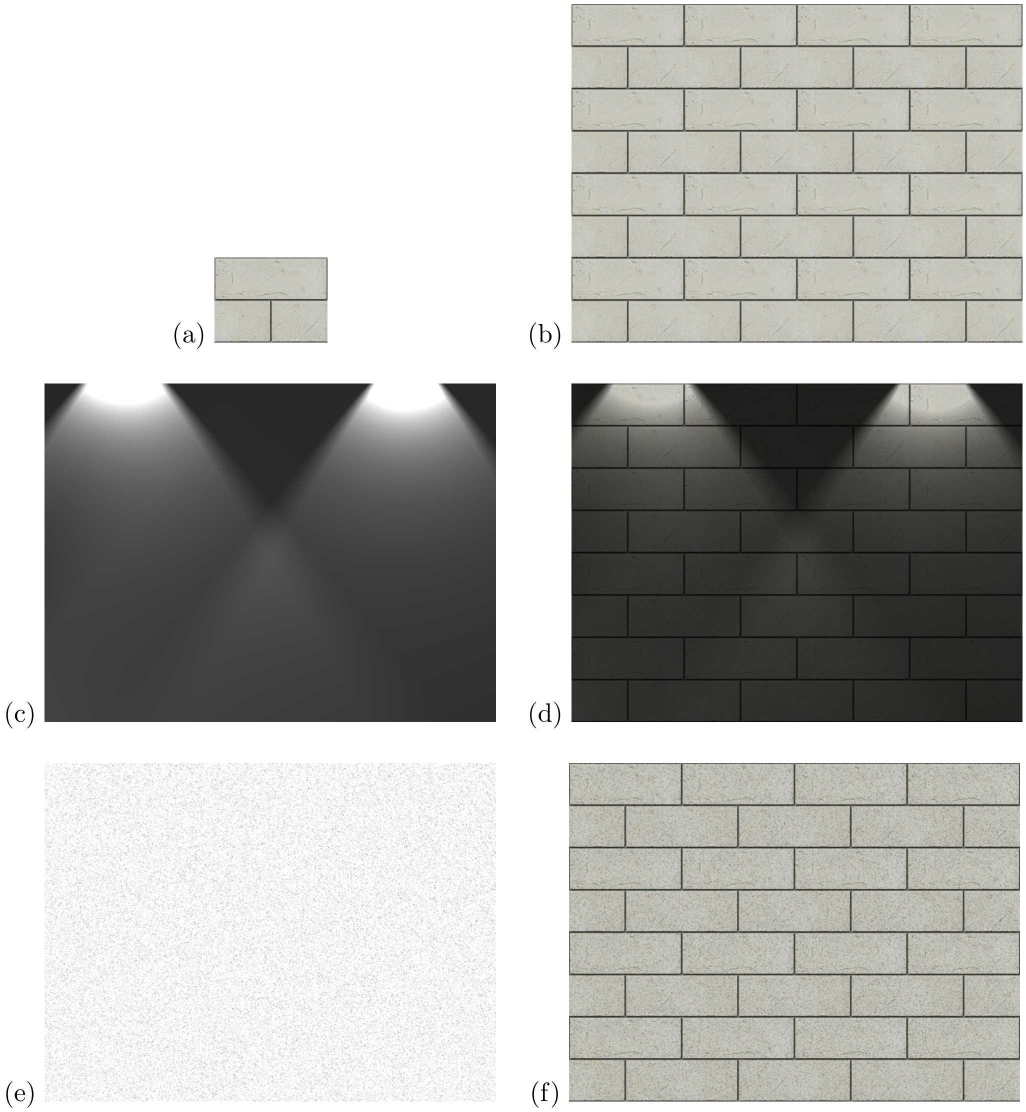


Figure 3: Additive versus multiplicative blending.
(a) a small test image containing red, blue, green, white and black areas.
(b) the same image rotated 90 degrees clockwise.
(c) the effect of combining (a) and (b) through additive blending.
(d) the effect of combining (a) and (b) through multiplicative blending.
Note how multiplicative blending always produces a colour that is similar or darker than the originals.



(a) (b) (c) (d) (e) (f)

Figure 4: Using multiplicative blending to create unique textures.
(a) shows a small texture image representing brick work.
(b) shows the same image tiled four across and four down. Notice how the image looks too uniform.
(c) an image representing the effect of two street lights.
(d) the effect of multiplicatively blending (b) and (c) to create a unique texture.
(e) randomly generated gaussian noise.
(f) the effect of multiplicatively blending (b) and (e) to create a unique texture.

			ANY		THE SAME
WHITE			COLOUR		COLOUR
Red	1.00	$\times$	x	$=$	x
Green	1.00	$\times$	y	$=$	y
Blue	1.00	$\times$	z	$=$	z
			ANY		
BLACK			COLOUR		BLACK
Red	0.00	$\times$	x	$=$	0.00
Green	0.00	$\times$	y	$=$	0.00
Blue	0.00	$\times$	z	$=$	0.00
RED			GREEN		BLACK
Red	1.00	$\times$	0.00	$=$	0.00
Green	0.00	$\times$	1.00	$=$	0.00
Blue	0.00	$\times$	0.00	$=$	0.00

Table 1: Examples of multiplicative blending

4 Experiment

The custom software used for the experiment was written and built in Visual Basic.net using Microsoft Visual Studio 2003. The computer used for the experiment was an AMD Athlon XP 2000+ with 512MB RAM and an NVIDIA GeForce4 4600 Ti graphics card.

4.1 Tasks

The MLD was used in two different modes: single-layer and multiple-layer. In the single-layer mode, the rear layer of the display was left white, and the front layer of the display was used for the experiment. In multiple-layer mode, both layers of the display were used simultaneously.

Participants were exposed to three task types in order, with multiple iterations of each task. The first task involved target selection, the second task looked at dealing with related layers of information, and the third task looked at using the display as a Focus+Context exercise.

4.1.1 Task One: Target Selection

The first task was designed to compare reaction times for selecting highlighted objects, across the different display types. There were three conditions used in this task: coloured objects in single-layer mode, coloured objects in multiple-layer mode and black objects in multiple-layer mode. For this task, a grid of airplane icons (120x120 pixels each) where displayed on the screen. In the coloured modes, 20 target icons were rendered in blue and the rest were left black. In the black mode, all the icons were rendered in black, but the target icons were rendered on one layer of the display and the rest of the icons were rendered on the other layer of the display (see Figure 5). Participants were asked to select all 20 targets as quickly as possible by single clicking each one with the mouse, while the time taken to select all the targets was recorded.

4.1.2 Task Two: MLD Map

The second task was designed to determine how participants performed when dealing with two layers of related information. For this task, a pipe network was rendered on top of a map in multiple-layer mode. The pipe network was rendered on the front layer and a map of Christchurch was displayed on the rear layer. As participants dragged the front layer around, the rear layer moved with it.

Participants were given the name of a Christchurch street, and were asked to locate the pipes on that road, then to click on the marked targets on that pipeline to 'mend' a break in the pipe. The pipeline was rendered as a single pixel wide, blue line, with red, perpendicular lines indicating the breaks in the pipe. Any clicks that did not occur on the targets on the named street, were counted as errors. To reduce the time taken to locate the target street, the streets used for this task were all chosen to be located near the University of Canterbury, an area the participants would be familiar with.

Example images for this exercise are shown in Figure 6.

4.1.3 Task Three: Focus+Context

The third task, referred to as "Focus+Context", was set up in a similar way to the second task. A fixed map of entire city of Christchurch was displayed on the rear layer, while a pannable map of the city, shown at 100% zoom, was displayed on the front layer. A rectangle was drawn on the rear layer to indicate the current viewport being shown on the front layer. In this way, the rear layer showed context of the overall location within the city, while the front layer could display the fine details of the roads and the pipe network. The brightness of the map on the rear layer was increased, making it appear washed out, to improve the readibility of the front layer.

Participants could navigate by dragging the map on the front layer or by dragging the rectangle on the rear layer. Moving the mouse pointer between layers of the display was accomplished through the standard PureDepth convention of clicking the middle mouse button (or scroll wheel). The zoom level of both layers was fixed, but the each layer could be individually toggled by pressing a key on the keyboard.

Example images for this exercise are shown in Figure 7.

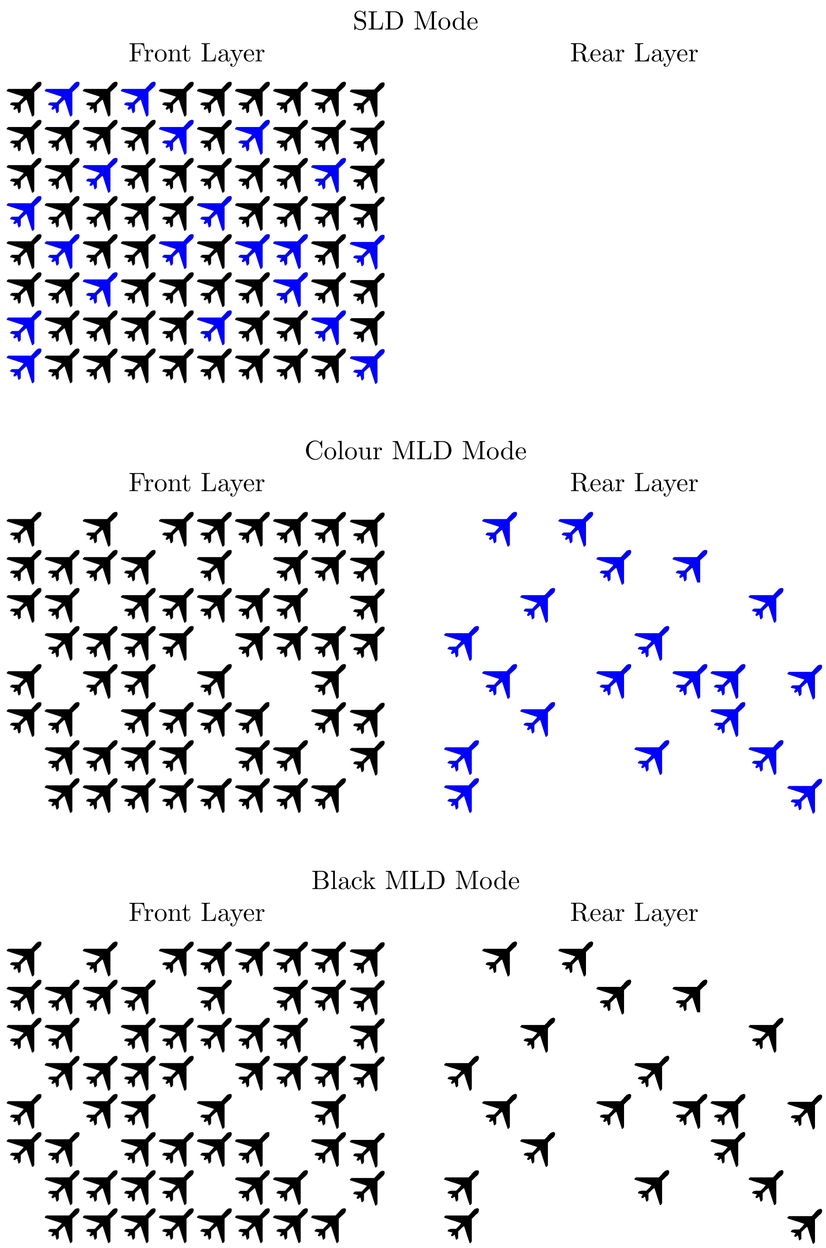


Figure 5: Example images used for Task One, on each layer of the MLD, in each mode.

Front Layer Rear Layer

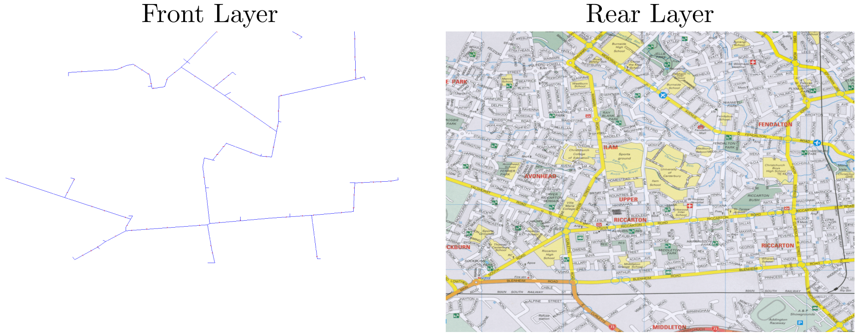


Figure 6: Example images used for Task Two (MLD Map), on each layer of the MLD.

Front Layer Rear Layer

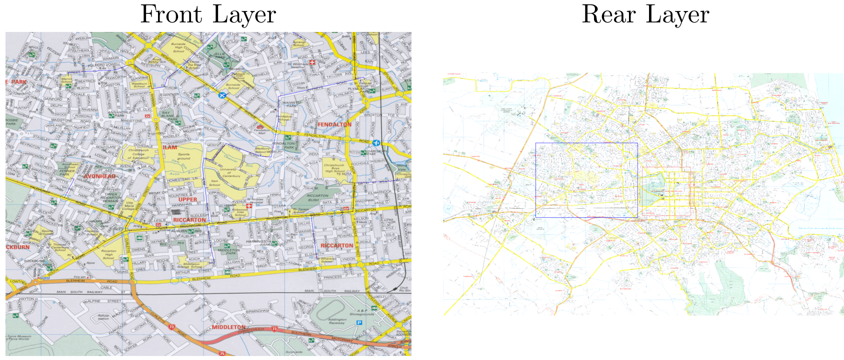


Figure 7: Example images used for Task Three (Focus+Context) on each layer of the MLD.

4.2 Method

The participant was informed about the format of the experiment and that the experiment would take 15-20 minutes to complete, with each task taking between twenty seconds and one minute to complete. The participant was briefed before each task as to what would be required of them and they were given the opportunity to ask questions before beginning. The tasks were all run in fullscreen mode (1280x1024 pixels), and audio feedback was given during the tasks to give both positive and negative feedback. If the participant had trouble completing the task, they were allowed to move onto the next task.

Before each type of task was examined, the participant was given the opportunity to familiarise themselves with the controls and the view provided by the display, by completing warm-up rounds. The partipicants were observed throughout the experiment. At the end of the experiment, the participant was asked to complete a short questionnaire to find out their opinions.

4.3 Results

A total of 16 University of Canterbury students, 11 male and 5 female, participated in the experiment over two days. Data collected during the warm-up trials was not used for analysis.

In the first task, participants performed slightly faster in the multiple layer mode than they did in the single layer mode, but the results are not statistically significant (ANOVA: $F = 3.596$, $P = 0.076$) and are most likely due to a learning effect. The average time for the single-layer task was 18.1 seconds ($SD = 3.94$), while in multiple-layer mode users averaged 16.8 seconds ($SD = 5.88$) for the coloured icons and 17.1 seconds ($SD = 1.41$) for the black icons.

The second task was looking at whether participants were able to relate information shown on different layers of the display. While it was intended to compare completion times for this task, there was a significant difference of local knowledge between the participants, and as such the completion time was not an accurate reflection of the performance of the display. This task formed the basis for showing participants the abilities of the display, so its results are reflected in the questionaire.

In the third task "Focus+Context", a total of 32 trials were completed, excluding the warm ups. Having given the participants the ability to toggle the layers on and off as wanted, the context layer was kept visible in 18 of them, hidden after the map was focussed in 7 trials, and hidden immediately after the task started in 7 trials. None of the participants turned off the focus layer for any significant period of time. Participants used only the focus layer to navigate in 17 of the trials, and both layers to navigate in 15 trials.

In the questionaire, participants were asked to provide values between one and five for a series of six statements. The results were analysed using the Kruskal-Wallis H-test ($df = 5$, $\chi^2 = 8.78$, $p = 0.118$). The results for each statement are shown in Table 2.

Statement	Min.	Lower Quartile	Median	Upper Quartile	Max.
I like the concept of a Multiple-Layer Display (1 = Disliked, 5 = Liked)	1.00	3.00	5.00	5.00	5.00
I found the MLD easy to use (1 = Hard, 5 = Easy)	2.00	2.50	4.00	5.00	5.00
I found the MLD easy to understand (1 = Hard, 5 = Easy)	2.00	3.00	4.00	5.00	5.00
I liked the Focus+Context user interface (1 = Disliked, 5 = Liked)	1.00	2.50	3.50	4.00	5.00
I found the Focus+Context UI easy to use (1 = Hard, 5 = Easy)	1.00	2.00	3.00	4.00	5.00
I found the Focus+Context UI easy to understand (1 = Hard, 5 = Easy)	1.00	3.00	4.00	5.00	5.00

Table 2: Results of the questionaire

5 Discussion

This experiment suggests that multiple-layer displays offer little or no performance benefit over single-layer displays for target selection tasks. However, the experiment also suggests that the multiple-layer display does not hinder the performance of users completing these tasks, and that users have little trouble distinguishing which layer an item has been rendered on. As such, the layers of the display can be used to provide semantic information in their own right.

The Focus+Context interface seems like promising idea, in terms of user opinion, but user performance was poor. Participants were often confused when they dragged the left side of the image over towards the right, and the viewport on the other layer moved left. This seems counter-intuitive at first, but it does make logical sense and there appears to be no straightforward way to resolve this issue.

Once they had completed the experiment, each participant was asked to fill out a short questionaire, and to comment on both the multiple-layer display and the experiment itself. Almost all the participants said that they thought the display was 'cool' and that 'it looked fancy', but few people were able to explain why they thought this. Those that did commented on both the element of depth that the display could render, and the ability to display a large amount of information in a small area. One participant commented that the display was easy to learn, while another commented that they felt it was confusing for a first time user. In terms of the software, two people commented that the reason why they didn't like the Focus+Context interface implementation was because it was confusing and too cluttered.

Points that the partipicants didn't like about the MLD, centered around having too much information on the display at a time, such as during the Focus+Context exercise. This could be resolved through further research and development. However, a large number of participants also commented that the display was sometimes unclear or out of focus. This is probably due to the effect that the front layer of liquid crystal has on the light travelling through from the rear layer. When the front layer is transparent, it can make the edges of objects on the rear layer appear to be blurred. This suggests that the rear layer is not particularly suited to displaying fine, detailed information such as text. One participant also commented that they didn't like the narrow viewing angle, although this might be beneficial in some applications.

Industries and applications that participants thought the display would be most useful for included games, graphic design, map reading, data mining, civil/mechanical/electrical engineering, security and infrastructure management.

5.1 Future Work

There are a number of thoughts that have arisen during this research that would form the basis for further investigation, including emulating the properties of this display on a single-layer monitor, improving applications that already deal with layers of information, and developing new applications.

Many of the features of this display could be implemented in a similar fashion on a single-layer display, so it would be useful to compare usability between an MLD and an emulation on a single layer. The blending exhibited by the MLD is multiplicative, which is easily implemented in software and with the performance of modern computers and their graphics cards, this would still allow for real-time interaction with widgets shown on the display. Likewise the depth effect can be implemented through perspective, and the parallax effect can be replaced with the ability to drag virtual layers around, on top of each other.

Users seem to find that information that is directly related between both layers is the easiest to understand on a multiple layer display. This suggests that interfaces such as those described by Fekete and Beaudouin-Lafon [6] are most suitable for use on an MLD. It would be interesting to take a graphical editor that works in layers (such as Adobe Photoshop or Corel Paint Shop Pro) and render the layers of the image on the layers of the MLD, rather than in a single window. This would involve careful use of blending and masking techniques to reduce the issues created by multiplicative blending, but it would enable users to work on multiple layers of an image simultaneously.

One of the interesting ideas that arose, was to use the MLD to ensure privacy of documents. Because the display has a narrow viewing angle, this could be used to best effect by rendering confidential documents in white text on a black background. If this is shown on both layers of the display simultaneously, then only a person sitting directly in front of the display would be able to clearly read the text. Users viewing the display from any other angle would only see a black screen or a series of unintelligible white blobs. This technique would be particularly suited to people such as payroll controllers and managers, who don't want office visitors to see what is displayed on their screen.

There is also the potential to build a software utility and/or API that combines much of the shared code used in MLD software. Most users seem to appreciate having the ability to toggle layers on and off as required, and this could be used to allow the MLD to work as a single-layer display for those desktop applications that don't require the MLD functionality. It would also be useful if the software could determine the order in which the panels were physically connected to the computer, and then configure the graphics card driver accordingly so that applications can determine which layer is which. This would eliminate the need to reconfigure the graphics card when switching between applications produced by different vendors.

6 Conclusion

While this research has not shown any performance improvement between tasks completed on a single-layer display compared with tasks completed on a multiple-layer display, the multiple-layer display does not appear to perform any worse either. These results seem to be inline with some of the conclusions made by William Wong [9]. He suggested that it was possible to show relationships between content on different layers. This has been demonstrated in the MLD Map and Focus+Context tasks. He also suggested that the layers made it possible to have both the context and the focus in view, again demonstrated by the Focus+Context task, although this feature in its current implementation is of limited use. Overall, these findings seem to support William Wong's research.

However, there is still a significant amount of work to be done to establish whether the multiple-layer display has a place on the desktop. Even if, through further research, a conclusion is made that the multiple-layer display is not particularly suited to desktop applications, it would continue to find many uses in the fields of advertising and entertainment, due to its ability to impress almost anyone that experiences it. The PureDepth Multiple-Layer Display will always be a truely innovative design.

References

[1] Wael Aboelsaadat and Ravin Balakrishnan. An Empirical Comparison of Transparency on One and Two Layer Displays. *The 18th British HCI Group Annual Conference 2004*, 11(4):61–63, 2004.

[2] Patrick Baudisch and Carl Gutwin. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. *CHI '04: Proceedings of the 2004 conference on Human factors in computing systems*, pages 367–374, 2004.

[3] Patrick Baudisch and Ruth Rosenholtz. Halo: A Technique for Visualizing Off-Screen Locations. *Conference for Human Factors in Computing Systems (CHI) 2003*, pages 481–488, 2003.

[4] Eric Chadwick. An Artist's Real-Time 3D Glossary - Texture Blending Examples. `http://www.brilliantdigital.com/developers/3dglossary/examples/blending%.html`.

[5] Deep Video Imaging. PureDepth MLD 3000 Datasheet. `http://www.puredepth.com/partners/mld3000.pdf`.

[6] Jean-Daniel Fekete and Michel Beaudouin-Lafon. Using the multi-layer model for building interactive graphical applications. *Symposium on User Interface Software and Technology*, pages 109–118, 1996.

[7] Deep Video Imaging. Puredepth website. `www.puredepth.com`.

[8] Masood Masoodian, Sam McKoy, Bill Rogers, and David Ware. DeepDocument: use of a multi-layered display to provide context awareness in text editing. *AVI '04: Proceedings of the working conference on Advanced Visual Interfaces*, pages 235–239, 2004.

[9] William Wong. Technology affordances of the ActualDepth™Multi-Layered Display. *TBG Grant Ref. Num. DVIL 0201: Human Computer Interaction of the ActualDepth™3D Display Technology*, 2003.

A Source Code

The software developed for this project is available in source code as Visual Basic.net solution, or as a compiled Microsoft Windows executable, by emailing the author, Carey Bishop:
`cjb133@student.canterbury.ac.nz`.