Pocket PhotoMesa: A Zoomable Image Browser for PDAs

Amir Khella
Human-Computer Interaction Lab
Institute for Advanced Computer Studies, Computer Science Department
University of Maryland,
College Park, MD 20742
+1 301 405-2764
akhella@cs.umd.edu

Benjamin B. Bederson
Human-Computer Interaction Lab
Institute for Advanced Computer Studies, Computer Science Department
University of Maryland,
College Park, MD 20742
+1 301 405-2764
bederson@cs.umd.edu

1. ABSTRACT[1]

Small devices such as Palm and Pocket PC have gained wide popularity with the advance and affordability of mobile technologies. Image browsers are among popular software applications on small devices. The limitations introduced by these devices such as screen resolution, processing power and storage impose a challenge for multimedia applications designed for larger displays to adapt to small screens. For an image browser, layout of images and navigation between them are critical factors of the users' experience.

Motivated by these challenges, we developed Pocket PhotoMesa: an image browser for the pocket pc that employs quantum strip Treemaps for laying out images and Zoomable User Interfaces for navigation. In this paper, we discuss the development of Pocket PhotoMesa and we describe a usability study comparing the performance and users' experience using Pocket PhotoMesa and ACDSee image browser (a current commercial offering) for the **P**ocket PC.

1.1 Keywords

Image browsers, Information visualization, mobile devices, mobile multimedia, Treemaps, Zoomable User Interfaces (ZUIs), Animation, Graphics, Pocket PC.

MUM 2004, October 27-29, 2004 College Park, Maryland, USA. Copyright 2004 ACM 1-58113- 981-0 /04/10... $5.00

2. INTRODUCTION

The past decade witnessed a major advance in the development of mobile technologies that provided ubiquity and affordability of small devices, fitting every day's needs and everyone's pocket. Starting in the mid nineties, several companies introduced monochrome portable displays for scheduling and address books. Few years later, Pocket PCs were introduced with color screens, more processing power, and larger storage. However, many limitations still exist for application development on mobile devices: screen resolution and size, limited processing power and stylus interaction are among the toughest challenges for mobile application developers.

Popular applications on PDAs include Personal Information Managers (PIMs), file explorers, board games and image browsers. Due to limited screen size and resolution, image browsing applications use scroll bars to cycle through image thumbnails and locate images of interest. Since scroll bars require finely tuned pointing skills on small devices, we designed and implemented an image browser that eliminates the need of scroll bars. We use Treemaps [3] to layout image thumbnails on a single screen and Zoomable User Interfaces (ZUI) [1] to navigate through images. In this paper, we show our design and the results of a usability study that investigates users' performance and satisfaction with our interface compared to a traditional image browsing interface.

We believe that laying out images on a single screen in groups corresponding to physical folders will help users to visually identify themes of each group of images and be able to locate images faster than using traditional interfaces that display only folder names and We also believe that animated zooming will enable users to maintain the navigation context and improve the users' experience.

3. RELATED WORK

3.1 Zoomable User Interfaces (ZUIs)

Zoomable User Interfaces present users with a single view of large information space populated with graphical objects (images and image groups in our case). The interface allows users to navigate the object hierarchy using smooth animated zooming through different levels of the details. Initially, a ZUI renders the information space in a single screen allowing users to get an overview of the information domain allowing them to identify themes and patterns in the big picture.

ZUIs were introduced more than thirty years ago in Sketchpad interface [4] which implemented an interactive object oriented 2D graphics system that enabled zooming and rotation of rendered objects. Almost a decade later, several systems started implementing interactive zooming: Spatial Data Management System (SDMS) [5] implemented two levels of semantic zooming, Pad and Pad++ [1] were developed as toolkits for building Zoomable User Interfaces. Zooming has also been a component of several other interfaces and toolkits developed later. Two of the major zooming toolkits available are Jazz [7], and its successor Piccolo [8], a toolkit for interactive structured graphics available in Java and C# on the desktop and on Pocket PC. These toolkits have been used in several domains such as slide show presentations [2], navigating ontology information [9], image browsing [10] as well as several other applications.

ZUIs have shown a statistically significant interaction improvement over several image browsing interfaces, but have not yet been shown to outperform traditional thumbnail grid interfaces [11]. The same study concluded that the number of images displayed within the browser is an important factor for users' performance and error rate. However, we believe that the potential for ZUIs is more promising on small devices when screen space is at an even greater premium.

A basic characteristic of ZUIs is that objects can be rendered quite small when you zoom out. Since image thumbnails can be difficult to understand when they are small, it is crucial that the thumbnails focus on the relevant part of the image. One approach to this is via automatic image cropping to generate thumbnails focusing on the salient parts of the image [6]. We think this is an important approach that should be considered in photo browsers, but is not one that we have pursued in Pocket PhotoMesa.

3.2 Treemaps

Treemaps are space-filling visualizations for large hierarchical datasets where the display area is divided into several rectangles whose areas correspond to some attribute of the dataset. Among the algorithms used for the Treemap layout are slice and dice [3], clustered [12], squarified [13] and strip Treemaps [16]. Treemaps are used in several visualizations such as SmartMoney's market map [14], image browsing [15] and many other domains. In image browsing, Treemaps suffered from the problem of aspect ratio: thumbnail sizes varied from one group to another. Quantum treemaps and quantum strip treemaps [16] solved the aspect ratio problem by using fixed size elements (quantum) across different areas.

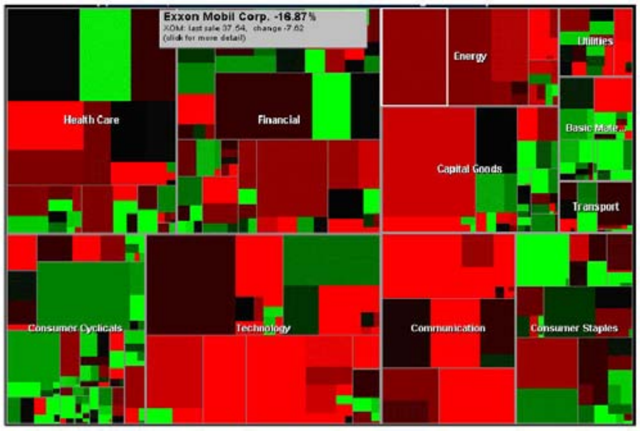


Figure 1: SmartMoney (www.smartmoney.com): An example of the use of treemaps in practice.

3.3 Desktop PhotoMesa

Desktop PhotoMesa is a zoomable image browser for the desktop PC(www.photomesa.com). The browser enables users to view hierarchies of directories in a Treemap layout and browse through them in an animated navigation mechanism. PhotoMesa uses quantum strip Treemaps to arrange images over a single screen. It also uses semantic zooming to display different thumbnail resolution at each level of detail. PhotoMesa is efficient for browsing large set of images, which would otherwise span several screen pages and requires the use of scrollbars in traditional image browsers.

Figure 2 shows 132 images loaded into PhotoMesa from 8 groups into a single screen. When users click inside a group, the view is smoothly zoomed into this directory. When the user is zoomed in to a group, a viewfinder, which can grow and shrink in size using the mouse wheel, allows users to specify a subset of the group images to zoom into or double click on a single image to zoom into a full resolution version of the image. At any time, users can press the right mouse button to zoom out to the previous zoom level. Users can also use arrow keys to navigate between images and groups. When the cursor is left to dwell over a thumbnail for a short period, a 200 pixel wide preview of this image is displayed in a tool tip overlay fashion (shown in the figure). The preview is removed as soon as the mouse moves.

Desktop PhotoMesa is implemented in JAVA using the Jazz toolkit [7]. Jazz is a polylithic structured graphics toolkit developed at the Human Computer Interaction Lab to support building interfaces in Java. The implementation of Jazz empowers programmers to create interfaces in a scene graph object oriented fashion. The scene graph is a hierarchy of nodes that represent relationships between objects. Functionality is built up by composing a number of simple objects. Complexity is therefore tackled by small reusable nodes that build up the scene graph.

Figure 2: Desktop PhotoMesa (www.photomesa.com)

4. POCKET PHOTOMESA

We faced several challenges during the design and the implementation phase to port PhotoMesa from the desktop to the Pocket PC. Limited interaction on the Pocket PC and small screen space imposed many restrictions on the interface design. In the following subsections, we present these challenges and the proposed solutions.

While the progress in mobile technologies has been promising, mobile processors are still not powerful enough to cope with the increasing demands of multimedia applications. Current PDAs are powered by 60-600 MHz processor with 16 to 64 Megabytes of memory. The processing power and memory limitations were critical factors in developing efficient algorithms and interaction techniques tailored to fit these particular devices.

Another limitation is the screen size and resolution. Pocket PCs have a screen resolution of 240x320 with 8 to 16 bit color depth. It is obvious that with this limited screen size, there is a maximum number of images that can be displayed on one screen. Moreover, with the current color depth, it is even harder to identify smaller thumbnails. A final limitation was imposed by the stylus input mechanism A stylus has three modes of operation: up, down, and tap-and-hold, as opposed to the mouse having six modes: mouse over, left button up, left button down, right button up, right button down, and wheel scroll. It is obvious that the lack of these extra modes using stylus interaction introduce major design limitations, especially that all these interactions modes are used in the desktop version of PhotoMesa.

Figure 3: Three views of Pocket PhotoMesa at different zoom levels (www.photomesa.com).

4.1 Pocket PhotoMesa Interface Overview

Pocket PhotoMesa tackles the problem of limited screen space by using quantum strip Treemaps to efficiently layout images on the screen, minimize the amount of white space and render fixed size thumbnails for all the images.

Using the stylus, users can perform the following functionalities:

- Zoom into a specific group by tapping into a white space inside the group or the group's name.
- Zoom into an image by tapping on the image's thumbnail.
- When zoomed into an image, tapping on the image renders the full resolution version and enables users to pan around by dragging the stylus or go back to the fit-screen image mode by tapping on the image again.
- Users can also select an area of the image to zoom into by tapping and dragging to draw the zoom viewport.
- Tapping on a white space within a current level brings the user up one level.
- Hardware keys and toolbar icons can be configured to provide more functionality, such as reordering the group to fill the screen and predefined zooming levels.

4.2 Pocket PhotoMesa Implementation Overview

While the original PhotoMesa was developed using Jazz toolkit, Pocket PhotoMesa was implemented from scratch without the use of any existing API. Several factors influenced this decision: First, there is no structured graphics toolkit available for building applications on the Pocket PC. In addition, Jazz was not suitable since the Java runtime environment is neither stable nor fast enough on the Pocket PC. Now, Piccolo.NET is available for Pocket PC (www.cs.umd.edu/hcil/piccolo), and so at this point, we probably would have chosen to develop Pocket PhotoMesa using Piccolo.NET.

In our case, it was essential to develop a tailored implementation to optimize the application for maximum performance. The bottleneck was rendering smoothly animated zooming transitions. At each step of the zooming, many arithmetic operations were involved to interpolate the position of visible images from the initial to the final position. These calculations were taking more time than rendering the visible portion of the canvas, which caused jagged animations. A successful solution for this problem was to pre-compute all the intermediate positions of each image and store them in a temporary structure. This way, at each zooming step, only the rendering overhead is considered which was fast enough to give a smooth transition. Other optimization techniques were used at several parts of the implementation to improve the performance of the interface.

The application was implemented in Microsoft Embedded Visual Studio 3.0 using MFC and Pocket PC SDK and consists of approximately 10,000 lines of code.

5. USABILITY STUDY

An experiment was conducted to compare Pocket PhotoMesa to Pocket ACDSee (www.acdsee.com). The split interface of

ACDSee shown below is a traditional image browsing interface based on choosing a group of images (folder) to browse, and displays the images in fixed thumbnail size, scrollable interface. It does not use any animated transitions.

Figure 4: Pocket ACDSee (www.acdsee.com)

5.1 The Experiment

We designed an experiment to answer the following questions:

- Is it better to fit all images in one screen or in multiple screens with scrollbars?
- What is the best number of images displayed on a single screen so that thumbnails are still identifiable?
- Do Treemaps provide a better layout than a traditional layout providing equal size rectangles to every group?

Our hypothesis was that the use of an image browser that lays out all the images efficiently in one screen would enable users to quickly locate images of interest by visually identifying the themes in each image group and remembering the location of a previously visited image. Moreover, we thought that the use of animation in zooming will improve users' satisfaction while not having a significant effect on the time to locate images.

Our independent variable is the application used. For this variable, we have three treatments:

- Pocket PhotoMesa with animated zooming
- Pocket PhotoMesa with single step zooming
- Pocket ACDSee

Our dependent variables are:

- Objective: The time required to locate a specific image.
- Subjective: User satisfaction.

The experiment was run within subjects, where each used both Pocket PhotoMesa and Pocket ACDSee for the tasks (order of use was randomized between participants to insure balance). Each interface was used with a different set of images to cancel learning effects. To insure that users use visual identification of images in Pocket PhotoMesa, group labels were disabled. The entire experiment took approximately 30 minutes per participant. The experiment had 15 participants who are all computer science students, of which two are females. All participants were already familiar with pen based interaction and most of them own a Palm or Pocket PC.

After a training period with each interface, users were given a couple of minutes to navigate the interface and remember the location of the images. Users were then asked to locate 5 different images. These five images were given to the users via a written textual description, printed color version, and shown on the Pocket PC screen for 2 seconds. Target images were carefully chosen for each task to ensure a balance between tasks. Tasks ranged in difficulty from locating one of many existing images from description to locating a visually ambiguous image displayed for 2 seconds. By visually ambiguous we mean that the thumbnail of the image is visually similar to some other thumbnails.

We chose sets of 75 images categorized into 6 groups from the Corbis image library (www.corbis.com). The number of images was carefully chosen after running two pilot experiments with 100 images and the users' feedback was clear that identifying thumbnails at this small size was too difficult. A third pilot experiment was run with 75 images and users were able to visually identify most thumbnails on the screen.

5.2 Results

5.2.1 Quantitative results

The compilation of background surveys showed that the participants had mixed backgrounds. All the users had some background using pen based devices (most of them use Palms), mostly for appointments and contacts. None of the participants used it for image browsing.

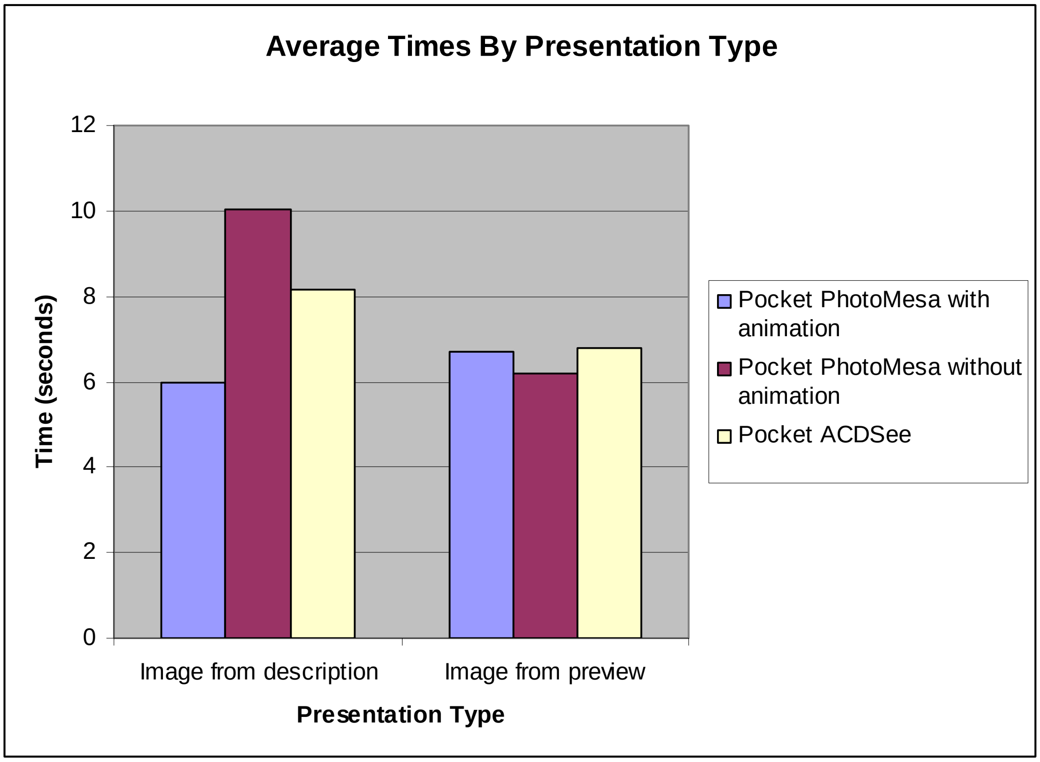
An analysis of the results shows that we got the best results using Pocket PhotoMesa without animated zooming when the image has been previously seen. Pocket PhotoMesa with animated zooming gives the best average time for locating images from a written textual description.

The total average time for locating an image on Pocket PhotoMesa with animation was 6.2 seconds, the time for the same interface without animation was 7.4 seconds and finally the average time using Pocket ACDSee was 6.4 seconds.

The previous analysis shows that there was no significant difference in time between Pocket PhotoMesa with animation and Pocket ACDSee, but interestingly, users’ performance slowed down by over 1 second when animation was removed from Pocket PhotoMesa.

5.2.2 Subjective satisfaction results

On a scale of 1-9, Pocket PhotoMesa scored an average ease of use of 7.5 and interface enjoyment of 6.6 while ACDSee scored 6.2 and 4.8 respectively.



All users found that non-animated zooming was helpful for them to perform the given task (non-animated zooming satisfaction of 4.5/5) and screen layout was useful in arranging the images into equal sized thumbs wasting the least amount of screen real estate possible (layout satisfaction of 4.25/5). When it came to the use of animated zooming, users had mixed opinions: While most of them agreed that the animation increased slightly the time it takes to find the image, they mentioned that it helped them to maintain the context of navigation.

Participants also agreed that the biggest advantage of having all the image thumbnails laid out in one screen is to visually identify themes from colors and contrasts of thumbnails. During the experiment, we hid the group labels in Pocket PhotoMesa to ensure that the navigation would be based on visual grouping of themes and fast identification of thumbnails by targeting a candidate group theme that contains the image of interest. Users easily remembered the constant location of thumbnails provided by the Treemap layout and tend to zoom out to the original view and navigate to another image.

Users also agreed that when the image of interest does not have distinct color theme or contrast features, the task of identifying the image thumbnails visually became difficult at the level of detail showing all images. They also stated that the folder names in ACDSee helped them identify the themes but it was tricky during one of the tasks (e.g. to identify the image of a kangaroo, users using ACDSee went to inspect the folder named "animals" while the original image was in folder named "Australia". To perform the same task on Pocket PhotoMesa, users were relying on their mental model of a kangaroo (light brown vertical shape, long tail and small head, usually exists in the desert) to visually identify thumbnails having mostly brown colors.

Three users said that the thumbnail sizes were large enough to identify average colors and patterns but were too small to identify shapes if the objects don't occupy a good percentage of the image.

5.2.3 Observations

Our major observation for the Pocket PhotoMesa interface is that almost all the users tried to take advantage of the screen layout to reduce interaction steps whenever possible: When they were given a task, they usually start by scanning the interface looking for a specific color or intensity. They were also limiting their visual search to one or more groups that have promising color themes for the target image. If users did not find the target image in the first 5-10 seconds, they start navigating the groups using the stylus. Most users were able to identify most of the images without the need to navigate the interface.

We also observed that in Pocket PhotoMesa, it was easier to locate images from memory than to locate it by description. For example, browsing for a picture of a zebra, users were not thinking about locating an image of an animal, instead they were looking for an image having a pattern of white and black vertical stripes. When they were given a task to locate a chessboard, it took them more time since they board was slightly oriented and had an unusual color theme.

While the interaction (zooming in and out using stylus on different areas of the screen) was not intuitive, users learned to use it quickly and had no problems performing the given tasks. The rate of errors due to wrong interaction was negligible.

6. CONCLUSION

While the quantitative results did not show a significant time improvement in locating images using a zoomable user interface with Treemap layout, user satisfaction showed that the interface was easy and fun to use. We believe that the main use of the interface is exploring and navigating rather than locating images. A search option that finds images by name or by average dominant color will improve the time locating specific images. Since all the images fit in a single screen, thumbnail size depends on the number of images. We believe that having more than 75 images on the pocket pc screen will increase the time to locate a specific image since some users already had difficulties working with the current thumbnail size. We have found that Pocket PCs are not usually used to hold a large number of images because of their storage limitation. Typical Pocket PC users hold pictures of their families, some memorable moments, or some portfolio work, hence 75 images is a good upper bound for such typical usage.

7. FUTURE WORK

Integration of more navigation controls and search features are on the top of our list for the next version of Pocket PhotoMesa.

8. REFERENCES

[1] Bederson, B., Meyer, J., (1998), Implementing a Zooming User Interface: Experience Building Pad++, Software: Practice and Experience, 28 (10), pp. 1101-1135.

[2] Good, L., and Bederson, B.B. (2002) Zoomable User Interfaces as a Medium for Slide Show Presentations, Information Visualization, pp. 35-49.

[3] Johnson, B. and Shneiderman, B., Treemaps: A Space Filling approach to the visualization of hierarchical information structures, in Proc. IEEE Visualization '91, pp. 284-291, San Diego, IEEE Computer Society Press.

[4] Bederson, B.B. et al.;. Pad++: A zoomable graphical sketchpad for exploring alternate Interface physics. J. Vis. Lang. Comput.,7:3, Mar. 1996.

[5] Herot, C.F., R. Carling, M. Friedell, and D. Kramlich. (1980). A Prototype Spatial Data Management System. Computer Graphics, 14(3), 63-70.

[6] Suh, B., Ling, H., Bederson, B. B., & Jacobs, D. W. (2003). Automatic Thumbnail Cropping and Its Effectiveness. UIST 2003, ACM Symposium on User Interface Software and Technology, CHI Letters, 5(2), pp. 95-104.

[7] Bederson, B. B., Meyer, J., and Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters, 2(2), pp. 171-180.

[8] Bederson, B. B., Grosjean, J., & Meyer, J. (2004). Toolkit Design for Interactive Structured Graphics. IEEE Transactions on Software Engineering, 30(8).

[9] Suh, B., & Bederson, B. B. (2002). OZONE: A Zoomable Interface for Navigating Ontology Information. In Proceedings of Advanced Visual Interfaces (AVI 2002) ACM Press, pp. 139-143.

[10] Bederson, B. B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters, 3(2), pp. 71-80.

[11] Combs, T. T. A., and Bederson, B. B. (1999). Does Zooming Improve Image Browsing? In Proceedings of Digital Library (DL 99) New York: ACM, pp. 130-137.

[12] Wattenberg, M. (1999). Visualizing the Stock Market. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 99) ACM Press, pp. 188-189.

[13] Bruls, M., Huizing, K., and van Wijk, J. J. (2000). Squarified Treemaps. In Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000) IEEE Press, pp. 33-42.

[14] Wattenberg, M. Map of the Market (1998). http://www.smartmoney.com/marketmap.

[15] Bederson, B. B. (2001). Quantum Treemaps and Bubblemaps for a Zoomable Image Browser. In Proceedings of User Interface and Software Technology (UIST 2001) ACM Press, (in press).

[16] Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002). Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. ACM Transactions on Graphics, 21(4), pp. 833-854.