Pocket PhotoMesa: A Zooming Image Browser for the Pocket PC

Amir Khella
akhella@cs.umd.edu

Ben Bederson
bederson@cs.umd.edu

Abstract

Small devices such as Palm and Pocket PC have gained much popularity with the advent of mobile technology and price affordability. Image browsers are among popular software applications on PDAs. The limitations introduced by these devices such as screen resolution, processing power and interaction make it harder for multimedia applications designed for larger displays to be as much usable on small screens. For an image browser, layout of images and navigation between them become critical factors for users' experience. These challenges motivated the development of Pocket PhotoMesa, a pocket pc image browser 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 perform a usability study comparing the performance and users' experience using Pocket PhotoMesa and ACDSee image browsers for the pocket pc.

Keywords: Image browsers, Information visualization, PDA, Treemaps, Zoomable User Interfaces (ZUIs), Animation, Graphics

1 Introduction

The past decade has witnessed a major advance in the development of mobile technologies that made it easier and more a? ordable for high processing power devices to become smaller and more e? ective ?tting everyday's needs and everyone's pocket. Starting in mid 90's, several companies introduced monochrome portable displays for scheduling and address books. A couple of years later, Pocket PCs were introduced with more processing power, color screens and larger storage. However, many limitations still exist for application development for PDAs: limited screen resolution, limited processing power and stylus interaction make the real challenge for PDA developers. Popular applications on PDAs are schedulers, ?le explorers, board games and image browsers. Image browsers are few of the application that exploit the challenges of limited screen size and resolution. Popular image browsers use scroll bars to cycle through image thumbnails and locate images of interest to inspect in full resolution. Our interest was to design and implement an image browser that eliminates the need of scroll bars by using Treemaps [3] for laying out images' thumbnails on a single screen and Zoomable User Interfaces (ZUI) [1] to navigate through images. We then perform a usability study to test users' performance and satisfaction with our interface as compared to traditional image browsing interfaces. Our major hypothesis is that laying out images on a single screen will help users to visually identify themes of each group of images (folder) and be able to locate images faster than using an interface allowing users to identify groups by folder names and select folders of interest to inspect images inside it. We also believe that using ZUIs will improve users' experience of image browsing.

2 Related work

2.1 Zoomable User Interfaces (ZUIs)

A Zoomable User Interface presents users with a zoomable view of large information space. Zoomable surface is populated with graphical objects (images and image groups in our case). The interface manages rendering and interaction with di? erent components and objects, allowing users to navigate the object hierarchy by zooming through di? erent levels of the hierarchy. Ideally, a ZUI renders information space into a single screen allowing users to get an overview about the information domain, identifying themes and patterns for later interaction.

ZUIs were introduced more than thirty years ago in Sketchpad interface [4] which implemented a visionary interactive Object Oriented 2D graphics system. This system allowed zooming and rotation of rendered objects. Almost a decade later, several systems started implementing zooming as part of their interaction techniques. Spatial Data Management

System (SDMS) [5] which implemented two levels of semantic zooming. Pad and Pad++ [6](Pad's sophisticated successor) were developed later on as toolkits for supporting Zoomable User Interfaces. Zooming has also been a component in several other interfaces and toolkits developed later. Two of the major zooming toolkits available are Jazz [7], an extensible zoomable user interface graphics toolkit in java, and Piccolo [8] a toolkit for interactive structured graphics. These toolkits were used in several domains such as slide show presentations [2], navigating ontology information [9] , image browsing [10] and several other applications.

ZUIs have shown a statistically signi?cant interaction improvement over several image browsing interfaces, while not outperforming traditional thumbnail grids interfaces [11]. The same study has concluded that the number of images displayed within the browser is an important factor for users' performance and error rate.

2.2 Treemaps

Treemaps are space?lling visualization of large hierarchical datasets. Conceptually, the display area is divided into several rectangles whose areas correspond to an attribute of the dataset. A key element Treemaps is the layout algorithm, such as slice and dice [3], clustered [12], squari?ed[13] and strip Treemaps[16]. Treemaps were deployed for several visualizations such as SmartMoney's market map [14], image browsing[15] and several other domains. As image browsing came into consideration, Treemaps introduced the problem of aspect ratio. With di?erent thumbnails sizes with varying aspect ratios in each group will introduce visual deformations, which will make it harder to identify and locate images of interest. Quantum Treemaps and Quantum Strip Treemaps [16] solve the aspect ratio problem and ?xed size elements (quantum) within the map.

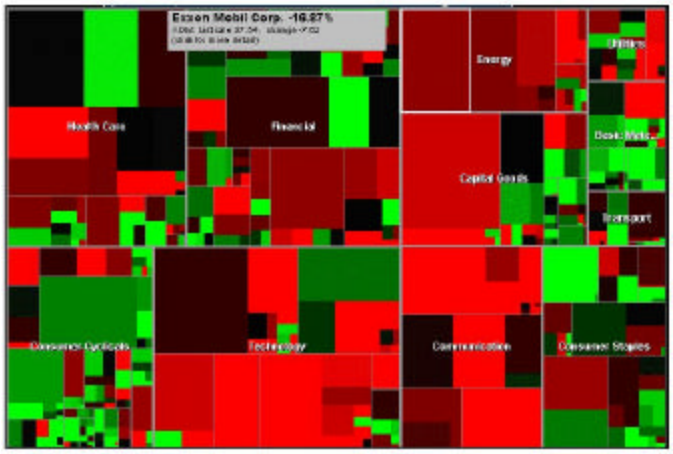

Figure 1: Smartmoney: An example of the use of treemaps in practice

2.3 Desktop PhotoMesa

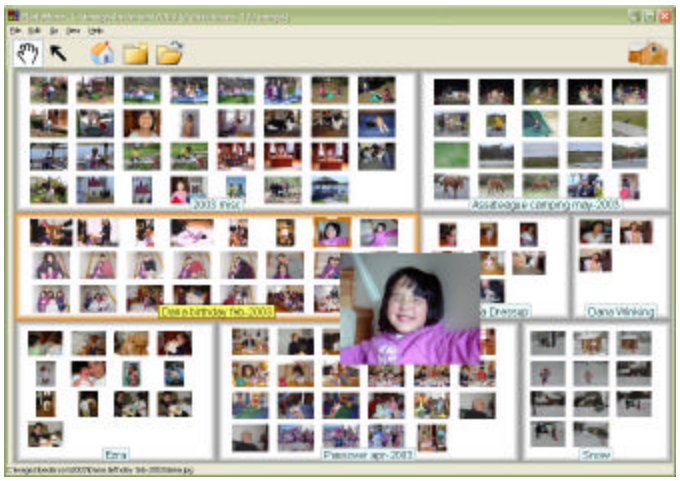

Figure 2: Desktop PhotoMesa

Desktop PhotoMesa is a zoomable image browser developed in JAVA for desktops. 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 save di? erent sets of thumbnails for each zooming level. PhotoMesa is suitable most for browsing large set of images, which would span several screen pages and requires the use of scrollbars in traditional image browsers. 4.1.1 PhotoMesa interface overview The ?gure shows 132 images loaded from 8 groups into a single screen. When the user moves the mouse over a group, the group is highlighted and the group's name is shown in full if it was already clipped. When the user clicks inside a group, the view is smoothly zoomed in to this directory. When the user is zoomed in to a group, a highlight red region (which can grow and shrink in size using the mouse wheel), allows the user 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 ?gure). The preview is removed as soon as the mouse moves.

2.4 Desktop PhotoMesa implementation

Desktop PhotoMesa is implemented in JAVA using Jazz toolkit. Jazz is a polylithic structured graphics toolkit developed at the Human Computer Interaction Lab at the University Of Maryland to support building interfaces in Java. The implementation of Jazz allows programmers to create interfaces in a scene graph object oriented fashion. A Jazz scene graph contains three basic types of objects: nodes, visual components, and cameras. The scene graph is a hierarchy of nodes that represent relationships between objects. Jazz is a simple toolkit, o? ering functionality by composing a number of simple objects. Complexity is therefore tackled by small reusable nodes that build up the scene graph.

3 Pocket PhotoMesa:

Figure 3: Pocket PhotoMesa showing three levels of zoom

Several challenges were faced in both the design and the implementation phase. The limitations of the interaction on the pocket pc as well as the limited screen space imposed many restrictions on the interface design. In the following subsections, we present these challenges and our proposed solutions. 5.1 Hardware limitations While the progress in mobile technologies is promising, PDA processors are still not powerful enough to cope with most multimedia applications. Typical PDAs now contain a 66MHz400MHz processor along with 16 to 64 Megabytes of memory. The processing power and memory limitations was a critical factor to develop e? cient algorithms and interaction techniques tailored to ?t 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 limitation on the 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 ?nal limitation was imposed by the interaction technique: the stylus. A stylus has three modes of operation: up, down, and tapandhold, as opposed to the mouse having the following modes: mouse over, left button up, left button down, right button up, right button down , and wheel scroll . It is obvious that these extra modes being unavailable in the stylus interaction, introduced a major design limitations especially that almost every one of them are used in the desktop version of PhotoMesa.

3.1 Pocket PhotoMesa interface overview

The limitations imposed by the processing power and memory are discussed in the next subsection. Pocket PhotoMesa tackles the problem of limited screen space the same way the original PhotoMesa tackles this same problem: by using the Quantum Strip Treemaps, it is possible to layout images e? ciently into one screen, minimize the amount of lost space and have ?xed size thumbnails for all the images. Using the stylus, users can perform the following functionalities:

1. Zoom into a speci?c group by tapping into a white space inside the group or the group's name(if shown).
2. Zoom into an image by tapping on the image's thumbnail.
3. When zoomed into an image, tapping on the image renders a full resolution version of the image and allows the user to pan around in the image by dragging the stylus or go back to the ?tscreen image mode by tapping on the image.
4. The user can also select an area of the image to zoom into by tapping and dragging, therefore drawing a rectangular selection for the area of interest. This area is rendered once the user releases the stylus o? the screen.
5. At anytime, tapping on a white space within a current level brings the user up one level.
6. some hardware keys and toolbar icons are con?gured to provide more functionalities, such as reordering the group to ?ll the screen and stepwise zooming.

3.2 Pocket PhotoMesa implementation overview

While the original PhotoMesa was developed using Jazz toolkit, Pocket PhotoMesa was implemented from scratch as a standalone application. Several factors in?uenced the decision of choosing this implementation: First, there was no structured graphics toolkit available for building applications on the pocket pc. Jazz was not suitable since the Java runtime environment is neither stable nor fast enough on the pocket pc. The second and most important factor is speed. It has been proved [8] that while structured graphics toolkits provide an easy way to implement interfaces, it imposes some runtime overhead through generic inheritance and functional composition. In our case, it was essential to develop a tailored implementation to ?t the code customization and optimization needs at each step. The bottleneck for the application was rendering smoothly animated zooming transition. At each step of the zooming, many multiplications and divisions are involved to interpolate the position of visible images from the initial to the ?nal position. These calculations were taking as much time as rendering the visible portion of the screen,which caused jagged animations. A

successful solution for this problem was to precompute all the intermediate positions of each image and store them in 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 grossed more than 10,000 lines of code.

4 Usability study

An experiment was conducted to compare Pocket PhotoMesa to Pocket ACDSee. The interface for ACDSee, shown below, is a traditional image browsing interface based on choosing a group of images (folder) to show, and displays the images in ?xed thumbnail size, scrollable interface.

Figure 4: Pocket ACDSee interface

The factors involved in the experiment were:

1. Fitting all images in one screen versus multiple screens with scrollbar
2. Thumbnail size and resolution
3. The ability to view multiple groups at the same time (all subfolders within a folder)
4. Maximum number of images that can be viewed on a single screen at a time.
5. Use of Treemap algorithm for laying out groups of images on the screen versus subdividing the screen evenly between groups.
6. Single click versus double click.

4.1 Hypothesis

Our hypothesis was that the use of an image browser that lays out all the images e? ciently in one screen will 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 as opposed to browsing labeled groups of images, one group at a time. Moreover, we assume that the use of animation in zooming will improve users' satisfaction while not improving much the time to locate images.

4.2 Independent variables

The interface used to navigate the set of images and perform the required tasks Three treatments:

1. Pocket PhotoMesa with animation
2. Pocket PhotoMesa without animation
3. Pocket ACDSee

4.3 Dependent variables

1. Objective: The time required to locate a speci?c image.
2. Subjective: User satisfaction.

4.4 The experiment

The experiment was run between subjects, where each subject used both Pocket PhotoMesa and Pocket ACDSee for the tasks (order of use was randomized between subjects to insure balance). Each interface was used with a di? erent set of images to cancel learning e? ects. To insure that users use visual identi?cation of images in Pocket PhotoMesa, group labels were not disabled. Estimated time per subject is 30 minutes. Subjects were 15 computer science students, of which two are females. All subjects were already familiar with pen based interaction and most of them own a Palm or Pocket PC.

4.5 Tasks

Each user was asked to locate 5 images for each treatment. These ?ve images were given to the users as description, printed color version, or shown on the Pocket PC screen for 2 seconds. Target images were carefully chosen for each task to ensure the balance between tasks. Tasks ranged in dif?culty from locating one of many 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 not distinct from the other thumbnails. We have chosen sets of 75 images categorized into 6 groups from Corbis image library. The number of images was carefully chosen after running two pilot experiments with 100 images and the users' feedback was obvious about the di? culty of identifying thumbnails at this small size. A third pilot experiment was run with 75 images and users were able to visually identify most thumbnails on the screen.

4.6 Study results

4.6.1 Subjects background

The compilation of background surveys showed that the subjects had mixed backgrounds. Users did not agree about a speci?c browser they use. All the users have background using pen based devices (most of them use Palms), mostly for appointments and contacts. None of the subjects used it for image browsing.

4.6.2 Quantitative results

An analysis of the ?gures 5 and 6 (last page) shows that we get the best results using the Pocket PhotoMesa without animated zooming when the image has been previously seen. The same application with animated zooming gives the best average time for locating images from description. The total average time for locating an image on Pocket PhotoMesa with animation was 6.17 seconds, the time for the same interface without animation was 7.436 seconds and ?nally the average time using Pocket ACDSee was 6.39 seconds. The previous analysis shows that there was no major di? erence in time between interface 1 and 3 while there was more than 1 second improvement over interface 2.

4.6.3 Subjective satisfaction results

Out of a scale of 9, Pocket PhotoMesa score an average ease of use of 7.5 and interface enjoyment of 6.63 while ACDSee scored 6.25 and 4.75 for the same factors respectively. All users found that nonanimated zooming was helpful for them to perform the given task (nonanimated zooming scored 4.5/5) and screen layout was also useful in arranging the images into equal sized thumbs wasting the least amount of screen real estate possible (layout scored 4.25/5). When

it came to the use of animatedzooming, users had mixed opinions about it, but most of them agreed that for the task of ?nding an image, the animation increased slightly the time it takes to ?nd the image but they also agreed that it helped them to maintain the context of navigation.(animated zooming scored 3.63/5). Subjects stated that the biggest advantage of having all the images thumbnails laid out in one screen is to visually identify themes from colors and contrasts. During the experiment, we hid the group labels in the Pocket PhotoMesa interface to ensure that the navigation will be based on visual grouping of themes and fast visual identi?cation of thumbnails by visually targeting a candidate theme that contains the image of interest. Another factor that helped them locate images faster on Pocket PhotoMesa was the fact that the image location is always ?xed on the screen. After the tutorial for each interface, they were given a couple of minutes to navigate the interface and they tended to remember the location of the images if they saw them already during the experiment. Users also agreed that when the image of interest does not have distinguishing color or contrast features, the task of identifying the image thumbnails visually became di? cult at the zooming level showing all the images at once. However, users also agreed that when the image had some distinguishing features, it was easy to locate just by looking at the screen and visually identity the thumbnail of the target image. Users also stated explicitly 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". Doing the same task on the Pocket PhotoMesa interface, users were using their mental model of a kangaroo (light brown vertical shape, long tail and small head, usually exists in the desert) to visually identify images 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 or objects if they don't occupy a major area of the image.

4.6.4 Observations

Our major observation for the Pocket PhotoMesa interface is that almost all the users tried to take advantage of the screen layout to eliminate the need of using the stylus whenever possible. When users were given a task, they usually start by scanning the interface looking for the image of interest by looking for a speci?c color or intensity. They were also limiting their search to one or more group that has a candidate color theme for the target image. If users did not ?nd the target image in the ?rst 510 seconds, they start navigating the candidate groups using the stylus. Most of the 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 another image (on screen or printed) than to locate image by description. For example, looking 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. The same reason is behind the observation that it was hard for them to locate an image of chessboard from description since the board had di? erent color theme than what they were looking for. While the interaction (zooming in and out using stylus on di? erent areas of the screen) was not standard, users learned to use it quickly and had no problems using it for the given tasks. The rate of errors due to wrong interaction was negligible.

5 Conclusion

While the quantitative results did not show a signi?cant time improvement in locating images using a zoomable user interface with treemap layout, user satisfaction showed that the interface was easy to interact with and fun to use. We believe that the main use of the interface is exploring and navigating rather than locating images. With an added search option that locates images by name or by average dominant color will improve the time locating speci?c images. While the experiment was performed using 75 images, we believe that having more images on the pocket pc screen will increase the time to locate a speci?c image since some users already had di? culties working with the current thumbnail size (since all the images ?t in a single screen, thumbnail size depends on the number of images). Our claim is 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 and 75 images is a good upper bound for such typical usage.

6 Future work

Integration of more navigation controls and search features is one of our main focuses for improving the current version of Pocket PhotoMesa. We also plan to incorporate better algorithm for creating the thumbnails to enhance their quality.

7 References

[1] Bederson, B., Meyer, J., Software: Practice and Experience, 1998 (in press), Implementing a Zooming User Interface: Experience Building Pad++.

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

[3] Johnson, B. and Shneiderman, B., Treemaps: A SpaceFilling approach to the visualization of hierarchical information structures, in Proc. IEEE Visualization '91, pp. 284291, 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), 6370.

[6] Bederson, B. B., and Meyer, J. (1998). Implementing a Zooming User Interface: Experience Building Pad++. Software: Practice and Experience, 28(10), pp. 11011135.

[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. 171180.

[8] Bederson, B. B., Grosjean, J., and Meyer, J. Toolkit Design for Interactive Structured Graphics HCIL200301 , CSTR4432 , UMIACSTR200303.

[9] Suh, B., and Bederson, B. B. (2002) OZONE: A Zoomable Interface for Navigating Ontology Information. Proceedings of International Conference on Advanced Visual Interfaces (AVI 2002), Trento, Italy, (in press).

[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. 7180.

[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. 130137.

[12] Wattenberg, M. (1999). Visualizing the Stock Market. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 99) ACM Press, pp. 188189.

[13] Bruls, M., Huizing, K., and van Wijk, J. J. (2000). Squari?ed Treemaps. In Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000) IEEE Press, pp. 3342.

[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] Ordered and Quantum Treemaps: Making E? ective Use of 2D Space to Display Hierarchies , ACM Transactions on Graphics (TOG), 21, (4), October 2002, 833854.

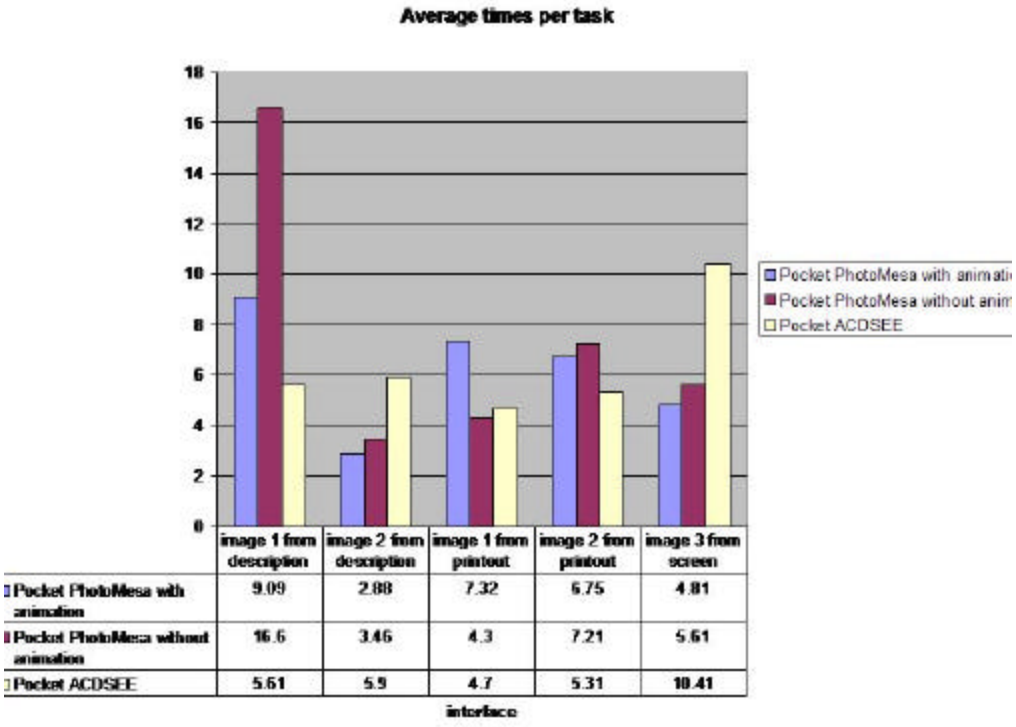

Figure 5: Average time in seconds for tasks performed per interface

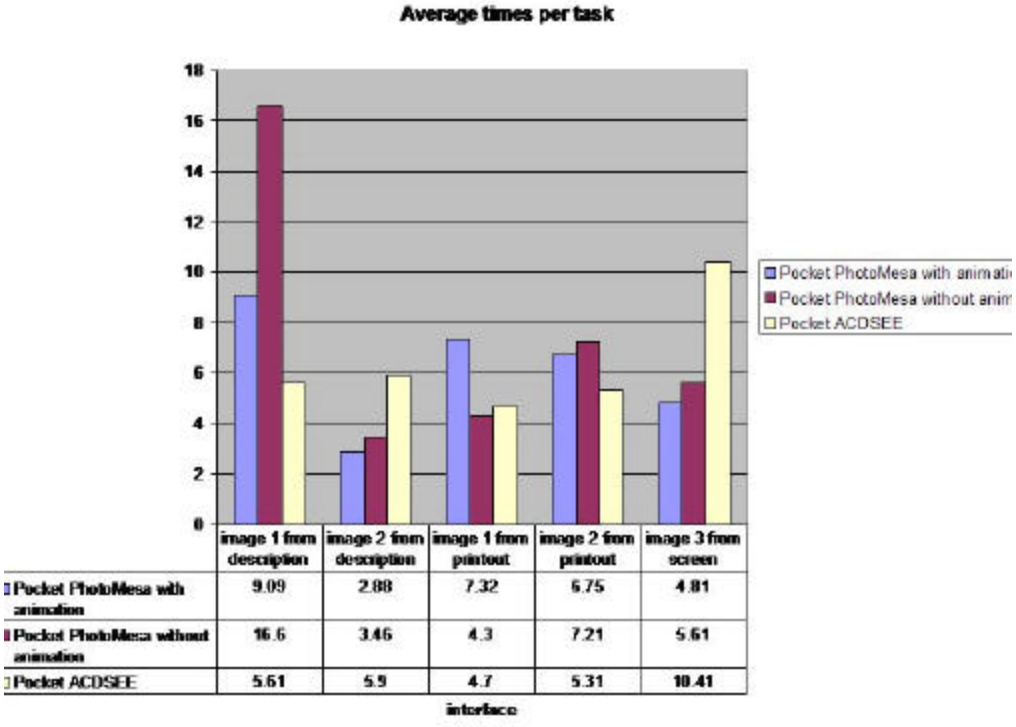

Figure 6: Average time in seconds for tasks performed per interface