An Evaluation of Techniques for Browsing Photograph Collections on Small Displays

Dynal Patel[1], Gary Marsden[1], Steve Jones[2], Matt Jones[2]

[1]Dept of Computer Science, University of Cape Town, Private Bag Rondebosch 7701, Cape Town, South Africa
`{dpatel,gaz}@cs.uct.ac.za`
[2]Dept of Computer Science, University of Waikato, Private Bag 3105, Hamilton, New Zealand,
`{stevej,mattj}@cs.waikato.ac.nz`

Abstract. In this paper we evaluate techniques for browsing photographs on small displays. We present two new interaction techniques that replace conventional scrolling and zooming controls. Via a single user action, scrolling and zooming are inter-dependently controlled with *AutoZoom* and independently controlled with *GestureZoom*. Both techniques were evaluated in a large-scale, 72-subject usability experiment alongside a conventional thumbnail grid image browser. Performance with the new techniques was at least as good as that with the standard thumbnail grid, even though none of the subjects had prior experience with such systems. In a number of cases – such as finding small groups of photos or when seeking for images containing small details – the new techniques were significantly faster than the conventional approach. In addition, *AutoZoom* and *GestureZoom* supported significantly more accurate identification of subsets of photographs. Subjects also reported lower levels of physical and cognitive effort and frustration with the new techniques in comparison to the thumbnail grid browser.

1 Introduction

The nature of photography has changed dramatically. It was once the business or pastime of a small number of individuals—experts in both the technology for capturing images and the chemistry of processing them. However, since the introduction of the Kodak Brownie a little over 100 years ago, personal photography has become increasingly affordable and pervasive. Indeed, photographic technology is now incorporated into a range of devices such as personal digital assistants (PDAs) and mobile telephones enabling photographs to be taken more quickly and cheaply than ever before. Although such devices have ever-increasing capacities to store images, their use presents users with a challenge, as the screens on which those images are browsed and viewed have become smaller.

A question that arises then, is how may a user be supported in browsing a set of photographs on such a device with limited display space? In this paper, we present two new scroll and zoom photo browsing interfaces that simplify navigation controls. Each of these interfaces utilizes two control mechanisms: one that behaves in a similar manner to a scrollbar to support scrolling and provide spatial orientation, and another that combines control over both scrolling and zooming. In the *AutoZoom* interface, this second mechanism utilizes the Speed Dependent Automatic Zooming (SDAZ) technique [8], in which scroll speed and zoom level are inter-dependent. In the *GestureZoom* interface, scrolling and zooming are controlled independently. In both interfaces distinct zooming, panning and scrolling actions are replaced with a mechanism through which control over scroll direction, scroll speed, and magnification level of the user's information space are integrated into a single action.

We carried out an experimental evaluation of the two interfaces, and compared their performance to a conventional vertically-scrolled row-column thumbnail method, as is used in applications such as Apple Computer's iPhoto. Both objective and subjective quantitative measures reflect positively on the new designs.

2 Background

2.1 The Current State of Photo Browsing

To explore the sorts of features used in a digital photo organizer, Rodden and Wood studied participants' use of the "Shoebox system" [13]. This system offered advanced features such as audio and text annotation for playback, and content-based image searching. However, users took little advantage of them, emphasizing the utility of two core facilities found in many commercial photo browsers: chronological arrangement and browsable thumbnails. There are three possible reasons for these user preferences: chronological information access is natural for users as shown in the context of email [15] and personal information spaces [10]; users shy away from the computationally expensive content-based image searches, choosing to exploit the human visual system to rapidly scan and process a grid of thumbnails; and, finally, these schemes do not require user effort, like manual annotation, in organizing or pre-processing of images.

Recently researchers have proposed ways of improving on the two core facilities offered by standard commercial browsers by proposing more efficient image layout algorithms and exploiting metadata automatically added to photographs by digital cameras. Photomesa [3] is an example of a browser that uses novel layout mechanisms (quantum treemaps and bubble maps) that allows users to see as many photos as possible and maintain context. It allows users to group photographs by date, filename and directory. A PocketPC version of the system [9] has been produced but the usability evaluation did not show any improvements over the conventional approach.

PhotoTOC [11] is a browsing user interface that uses an overview and detail design. The detail view is a list of thumbnails laid out in a grid, ordered by time. The overview pane is automatically generated by an image-clustering algorithm, which clusters on the creation time and the color of the photographs. However the evaluation shows that PhotoTOC was no better, and was sometimes out performed by, Light Box (a row-column thumbnail browser which simply showed all the pictures in a flat, scrollable list, ordered by creation time).

The Calendar Browser [6] also exploits the automatically annotated timing data to structure collections of photographs into meaningful summaries. Results from a user study show that summarized collections can lead to significant improvements in the time taken to search for an individual photograph.

While the advanced clustering techniques of the Calendar Browser and PhotoTOC browser may open up interesting ways for users to access their photograph collections, given the known preference for simple, chronological, thumbnail scrolling schemes, we were motivated to improving these within small screen contexts.

2.2 Improving Standard Scrolling Schemes

A number of researchers have been interested in improving standard scrolling schemes. Igarashi and Hinckley [8] have identified two major limitations with using traditional scrollbars. Firstly, when browsing a document, users have to shift their focus between the document and the scrollbar. They suggest that this may increase the operational time and may cause a significant attentional overhead. Secondly, they observed that in large documents, small scrollbar movements can cause a large movement of the document. This rapid rate of change can be too great for users to perceive, resulting in visual blurring and consequent user disorientation.

To counter this visual blurring, Igarashi and Hinckley proposed Speed Dependent Automatic Zooming (SDAZ). This navigation technique also alleviates other problems with conventional scrolling (e.g. attentional overhead). SDAZ unifies rate based scrolling and zooming by automatically adjusting the zoom level during scrolling to reduce the effect of rapid visual flow when a document is scrolled quickly at its normal scale. However their preliminary evaluation of SDAZ for document, map browsing and image browsing on a desktop computer produced disappointing results, with similar or worse performance than traditional methods.

Cockburn and Savage [4] carried out a substantial evaluation of their own implementations of the SDAZ document and map viewing application. Their systems used sophisticated graphical processing techniques to provide more responsive, smoother scroll and zoom animations. Their results are much more promising and show SDAZ in a new light. In their evaluation, Cockburn et al found that participants were 22% faster when using SDAZ than when using a common commercial document viewer. In map browsing, the performance benefits increased to 43%. Furthermore, workload assessments, preferences and the participant's comments all amplified the efficiency and effectiveness of the automatic zooming approach.

Both prior studies of SDAZ focused on its use on standard desktop displays, where a larger percentage of the information space is visible than is the case on small screen devices. The Palm Zire 71, for example, provides roughly 5% of the display area of a standard 15-inch laptop computer screen. The implication is that navigation may require increased user interaction for panning, zooming and scrolling when conventional navigation mechanisms are used. The experiment that we report on in the following section determines the extent to which our variations on SDAZ can ameliorate these problems for browsing photographic collections on small displays.

3 Photo Browsing on Small Displays

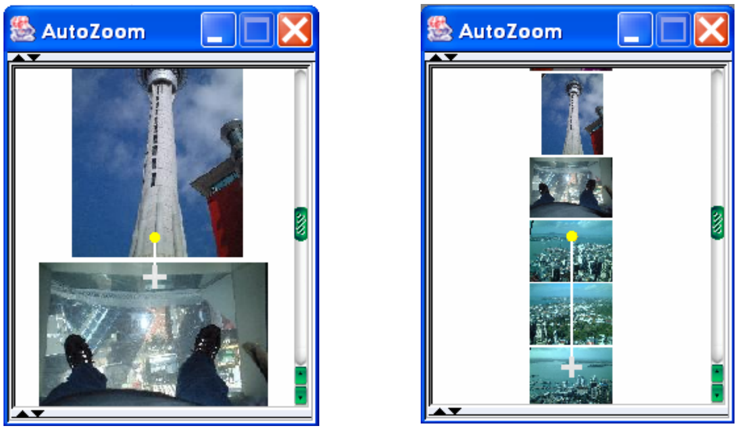
We developed two scroll and zoom based photo-browsing interfaces: *AutoZoom* and *GestureZoom*. In both interfaces, photographs are presented in a vertical list that is a single image wide, with a chronological ordering placing the most recent images at the top of the viewport. This organization is consistent with findings by Rodden and Wood [13], that users were satisfied with a simple chronological and folder/event based arrangement of their digital photographs, leading to more frequent browsing and reducing the effort of finding particular images. Additionally, the use of a vertical list provides methodological consistency with Igarashi and Hinckley, and Cockburn and Savage. However, we are aware that the choice of a vertical or horizontal list is language dependent (Dong et al [5]), and have designed both interfaces to allow users to configure scrolling direction

For the *AutoZoom* interface, the SDAZ variant is operated by vertical dragging actions with the pointing device. These actions control the rate at which images scroll through the viewport, the image size (zoom level) and the scroll direction. The vertical centre of the viewport acts as the threshold for direction change—dragging above the centre moves the images downwards and vice versa. Image size is inversely proportional to the distance of the pointer from the vertical centre, and changes dynamically as the pointer moves either away from or towards the centre (see Figure 1). Images are not reduced beyond a minimum (user specified) size threshold. Once this threshold is reached, an acceleration function maps further increases in drag distance proportionally to scroll speed.

The perceived effect to the user, then, is that as they increase their scrolling speed, the photo images get smaller and smaller, zooming out to get an overview, reducing the effects of visual blur. When the user completes an action by releasing the pointing device, the images are smoothly animated back to their normal size at the current location in the list.

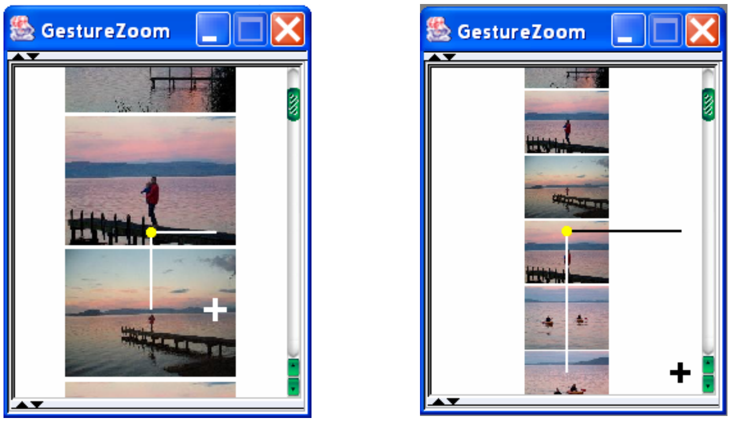
For the *GestureZoom* interface, vertical drag operations control scroll speed and direction as with the *AutoZoom* interface, but do not control image size (zoom level). Zoom level is controlled by horizontal movement of the pointing device away from the horizontal centre of the viewport to the right-hand or left-hand side of the display. Image size is inversely proportional to the horizontal drag distance.

Figure 2 (a) shows a pointer position – indicated by the cross – leading to a moderate scroll speed with small image reduction: the user is dragging below and slightly to the right of the viewport centre. In 2 (b), the user has dragged the pointer to the right-hand corner of the display, producing the maximum scrolling speed and the minimum image size. Returning the pointer to the centre of the viewport returns the images to the full size. As with *AutoZoom*, when the user releases the pointer (e.g. removing the stylus from the screen), the images smoothly animate back to their normal size.



(a) (b)

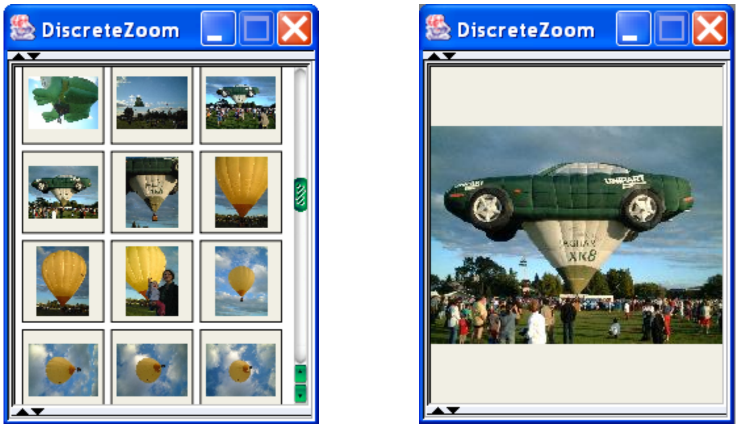
Fig. 1. *AutoZoom* interface: as cursor is dragged away from centre, scroll speed/image size change. (a) moderate speed, images slightly reduced; (b) faster speed and smaller size (Cross added for clarity)



(a) (b)

Fig. 2. *GestureZoom* interface: (a) moderate scroll speed and small image reduction; (b) maximum speed and minimum size. (Cross added for clarity)

The scrollbar has the same appearance and behaviour in the two interfaces—as the user begins to drag the slider the image thumbnails are immediately reduced to their minimum size and normal scrolling follows. At the end of a scrolling operation the images are expanded to their normal size. Hence, the scrollbar can be used for quickly gaining an overview of the image set, allowing users to find an approximate location in the set of photographs. Our approaches extend the original SDAZ implementations in a number of ways ([4],[8]). For instance, our algorithms have been developed to allow support a range of small screen sizes and input devices (see Section 7); they present a simple control feedback (the vertical line); and, the navigation direction can be set to either vertical or horizontal to support language differences.



(a) (b)

Fig. 3. *DiscreteZoom* browser: (a) the thumbnail view; and, (b) the enlarged view

A further browser called the *DiscreteZoom* browser (see Figure 3) was implemented for the purposes of comparative evaluation. It is a thumbnail browser that presents photographs in row and column scrollable list ordered by creation time. Users can click/tap on the desired photo to view an enlarged version. The selected photo is animated to fill the screen. Similarly users can click/tap on the enlarged photo to return to the thumbnail view. This browser reflects the features found in popular commercial browsers such as Apple iPhoto or ACDSee Picture Viewer [1],[2].

4 Evaluation

4.1 Hypotheses

The objective of the experiment was to compare user performance and subjective preferences with each of the three photo navigation techniques. Our hypotheses were as follows:

1. both *AutoZoom* and *GestureZoom* support faster navigation to target photographs than *DiscreteZoom*;
2. both *AutoZoom* and *GestureZoom* support more accurate identification of target photographs than *DiscreteZoom*;
3. subjective task load levels are lower for both *AutoZoom* and *GestureZoom* than *DiscreteZoom*.

4.2 Subjects

Seventy-two subjects (38 male and 34 female) took part in the experiment. Sixty-one subjects were students (either postgraduate or undergraduate), 6 were lecturers and 5 were software developers. 45 of the subjects had previously used photo management software, but only 5 on a small screen device. None of the subjects had used SDAZ interfaces. 70 participants described themselves as casual photographers (i.e. occasionally take photographs). Two participants described themselves as professional photographers (e.g. take photos for magazines or weddings).

4.3 Method

A repeated measure factorial design was employed. Subjects were randomly allocated to one of three groups, each containing 24 subjects. Each group used only one of the three interface designs to complete photo navigation tasks.

The independent variables were as follows:

- Interface. Between-subjects variable with three levels: *AutoZoom*, *GestureZoom* and *DiscreteZoom*;
- Task type. Tasks-types were based on those identified by as key to photo-browsing [13]. The type was within-subjects variable with three levels: *Event* (subjects searched for a set of photos relating to a particular well-defined event, e.g., "locate the Motor Rally"); *Single* (subjects searched for an individual photo containing a specified *Feature,* e.g., "Find this image of the Sky Tower"); and, *Property* (subjects searched for a set of photos taken at different events, but all sharing a property, such as all the photos containing an specific object, e.g., "Count all the photos that contain an hot-air ballon");
- Navigation distance. For *Event* and *Single* task types only. Within-subjects variable with two levels: short and long. Short distances were no more than half the length of the photograph list, and long distances were always more than half the length.

Events could be small (3 or fewer photos), or large (more than 3 photos – Figures 1 & 2, then, contain large events). A photograph *Feature* could also be small or large: a small feature was one that was 1/8th or less of the total image size (e.g. a small child in a forest scene), while a large feature was one taking up more than 1/8th of the image (e.g. a skyscraper).

Each subject completed a total of 27 experimental tasks, using one of the interfaces. For the *Event* task type they completed 3 tasks for each of the 4 navigation distance/event size combinations. For the *Single* task type they completed 3 tasks for each of the 4 navigation distance/feature size combinations. For the *Property* task type they completed 3 tasks (requiring the user to find 16, 30 and 120 images respectively).

Presentation order of the tasks was counterbalanced to minimize learning effects.

4.4 Experimental Measures

For each task the software automatically recorded a range of events including: time to complete task, distinct scrollbar operations and distinct zoom operations.

For *Property* tasks there was a target number of photos (*A*); in completing the task, a user found a number of images (*C*). Accuracy was then calculated as:

$$Accuracy = 100\left(1 - \frac{|A - C|}{A}\right)$$

Finally, we collected subjective responses about the workload required to complete tasks, as measured by the NASA task load index [7]. Responses were on a scale of 1 to 5, with lower values reflecting lower task loads. In all cases, the statistical data was subjected to significance testing using the analysis of variance method (ANOVA).

4.5 Procedure

On arrival, subjects were asked to read a summary of the experiment and provide consent to continue if they were in agreement. They then completed a profile questionnaire and were given 15 minutes to familiarize themselves with the set of photographs to be used in the experiment. At the end of this time, they were required to read instructions that provided a detailed description of each task type and also explained the operation of their assigned interface.

The operation of the interface was then demonstrated, and the subjects were given 10 minutes to explore the operation of the software for themselves. Following this they were given a set of training tasks of the same form as the experimental tasks. Subjects were encouraged to ask questions throughout the training period. Once the training tasks were completed subjects could take a short break, before commencing the experimental tasks. One aim of the training session was to allow users to familiarize themselves with the image set so that any learning effects during the experimental tasks would be reduced.

Subjects controlled progress of the experimental session via an on-screen dialog that allowed them to initiate a task, displayed task instructions, and allowed them to indicate completion of a task. At the start of every task, the viewport was reset to the show the beginning of the image list.

Event tasks were described textually. An event was found by selecting any one of the photographs within the event. For *Single* tasks, subjects were shown the target photograph and its corresponding caption. For both *Event* and *Single* tasks, users were prompted by the system to retry if their selection was incorrect; they were able to attempt the task as many times as they needed.

For *Property* tasks, subjects were required to count the number of photographs that shared a common property. They were given a field into which to enter the number. On completion of all the tasks subjects were requested to fill-out a questionnaire that captured their subjective views of the software and workload estimates via a NASA Task Workload Index.

4.6 Materials

The experiment was carried out on a desktop computer with a 1.7GHz processor, 1GB of RAM, and running Microsoft Windows XP. The viewport size for all three interfaces was set to 240x340 pixels to simulate the display of the HP h5550 Pocket PC. Users used a mouse as a stylus surrogate.

A single set of 300 of photographs was used in the experiment, providing a consistent set of stimuli across all tasks, subjects and conditions. The photographs were typical tourist type images – beach and mountain scenes; individuals and groups in sightseeing locations; and significant events, such as holiday periods – gathered over a 6 month visit to New Zealand by one of the authors.

5 RESULTS

5.1 Locating *Events*

AutoZoom and *GestureZoom* interfaces were significantly faster than the *DiscreteZoom* interface when searching for small events (F(2,69) = 5.0597, p=0.00890), with means of 26.0 seconds, 29.4 seconds and 45.5 seconds, respectively. Over all *Event* tasks, though, interface type had no significant effect on task completion time (F(2,69)=1.2848, p=0.28323).

Regardless of interface, subjects took significantly *longer* to locate events which were a short distance away (F(1,69) = 8.9667, p=0.00381), with short and long distance means of 33.25 and 23.46, respectively. At long navigation distances, large events were found significantly faster than the small events (F(1,69)=6.5946, p=0.01240), with mean search times of 14.04 seconds and 32.88 seconds. For short navigation distances, event size had no significant interaction with the time to locate an event, with search means of 34.38 seconds (large events) and 32.12 seconds (small events).

5.2 Locating *Single* photographs

AutoZoom was significantly faster at finding single photographs than *DiscreteZoom* at long navigation distances (F(1,46) = 9.5749, p=0.00335) with means of 28.90 seconds and 44.06 seconds, respectively. Both *Autozoom* and *GestureZoom* were significantly faster than *DiscreteZoom* when searching for images with small features (F(2,69) = 3.1596,p = 0.04865) with means of 39.15 seconds, 34.52 seconds and 48.69 seconds, respectively. Over all *Single* tasks, though, interface type had no significant effect on task completion times (F(2,69)=0.79012, p=0.45785).

Regardless of interface, subjects took significantly less time to locate single images that were a short distance away (F(1,69)=11.330, p=0.00125), with short and long distance means of 26.85 seconds and 34.98 seconds respectively. Also, images with smaller features took significantly longer to detect than those with larger ones (F(1, 69)=61.446, p=.00000), with small and large means of 40.79 seconds and 21.04 seconds respectively.

5.3 Locating photographs with a *Property*

AutoZoom and *GestureZoom* were significantly more accurate than *DiscreteZoom* (F(2,69)=14.614,p=0.0001), with mean accuracy rates of 92.38%, 89.98% and 76.15%, respectively. Over all *Property* tasks, interface type had no significant effect on task completion time (F(2,69)=1.5150, p=0.22704).

5.4 Subjective Preference

There was a significant difference between the mean task load ratings for the three interfaces (F(2,69) = 6.0275, p=0.00387): the mean rating for *DiscreteZoom* was 3.01; for *Autozoom* it was 2.31; and, for *GestureZoom*, 2.53.

Looking at the individual factors measured by the task load index, subjects found both new interfaces significantly less frustrating than the *DiscreteZoom* interface F(2,69) = 7.9593, p= 0.00078). Furthermore the mental workload (F(1,46) = 8.4033, p = 0.00572) and effort (F(1,46) = 7.9310, 0.00713) were significantly lower for the for the *AutoZoom* interface than the *DiscreteZoom* interface.

6 Discussion

Considering the results in the light of the three hypotheses noted in Section 4.1.

1. **Both *AutoZoom* and *GestureZoom* support faster navigation to target photographs than *DiscreteZoom*.** The results indicate the new techniques performed as well and in some cases better than *DiscreteZoom*. More specifically, both new interfaces were significantly faster when finding *Single* photos containing small-sized features as well as detecting *Events* consisting of a small number of photos. *AutoZoom* was also significantly faster than the *DiscreteZoom* interface at locating *Single* images at long navigation distances.
2. **Both *AutoZoom* and *GestureZoom* support more accurate identification of target photographs than *DiscreteZoom*.** The new techniques were significantly more accurate when finding a set of photographs that fit a given description.
3. **Subjective task load levels will be lower for both *AutoZoom* and *GestureZoom* than *DiscreteZoom***. The results of the task load calculations show that subjects perceived the new systems to be significantly less onerous than the *DiscreteZoom* browser.

It is worth remembering that none of the subjects had previous experience of SDAZ-type interfaces while all would be familiar with the conventional scrolling approach of *DiscreteZoom*. It is encouraging, then, to see such consistently good performance with the new schemes after minimal training. During task completion, the average amount of time spent operating the zoom/scroll control with the new interfaces was 22.5 seconds; this is nearly four times the duration spent using the scrollbar (5.9s). We are satisfied, then, that the benefits provided by the new interfaces come from the integration of scrolling and zooming.

Small features in an image, small groups of photographs and individual, target photos are more easily overlooked with *DiscreteZoom*, as they scroll past at thumbnail size; the explicit zoom-in/zoom-out operations needed to check individual image contents also contributes to the slower performances. Such problems with grid-based thumbnail browsing have been recognized by others who suggest, for example, processing the images to present only the salient details [14]. *AutoZoom's* better performance at finding *Single* images at long navigation distances suggests that these sorts of technique may be of greater benefit for very large sets of image.

7 Future Work

This experiment was simulated on a desktop computer as at the time the software was written, PDAs such as the HP Pocket PC did not have sufficient processing power and memory to run such applications. The apparatus has allowed us to gain very useful insights into the relative benefits of browsing schemes. We have now ported the code to a mobile environment, achieving responsive, smooth animation.

While the approaches have been implemented to accommodate a device using a pointer (e.g. a stylus), they can be extended for use with other interaction devices. For example, *AutoZoom* could be used with physical dial-type wheels as seen on the iPod or the *smartPad* proposed by Rekimoto for use in mobile phones [12], providing one-handed interaction. Meanwhile, joystick-type mechanisms may permit the use of *GestureZoom* schemes.

8 Conclusions

Our work provides evidence that small screen photo browsing may be improved with interaction schemes that integrate scrolling and zooming. As camera enabled mobile devices become more common, and picture taking and sharing more prevalent, it will become increasingly important to manage photograph collections using a small screen and input devices such as a stylus. We believe that the work presented here forms a good foundation for future generations of this software.

9 Acknowledgements

This work was supported by grants from the University of Waikato, the University of Cape Town and from Microsoft Corporation. Stephen Brewster and Richard Schroder provided helpful inspiration with respect to *GestureZoom.* We are also grateful for the comments of the anonymous reviewers.

9 References

1. ACDSee (2001). http://www.acdsystems.com/english/products/acdsee/acdsee-node.htm
2. Apple iPhoto. (2002). http://www.apple.com/iphoto
3. Bederson, B., (2001). Photomesa: A zoombable image browser using quantum treemaps and bubblemaps, in *Proceedings UIST 2001*, 71-80, ACM Press.
4. Cockburn, A. & Savage, J. (2003). Comparing Speed-Dependent Automatic Zooming with Tradional Scroll, Pan, and Zoom Methods, *People and Computers XVII: British Computer Society Conference on Human Computer Interaction. Bath, England. 2003*, 87-102.
5. Dong, Jianming & Salvendy, G. (1999). Designing menus for the Chinese Population: Horizontal or Vertical? *Behaviour and Information Technology* Vol. 18, 467-471.
6. Graham, A., Garcia-Molina, H., Paepcke, A & Winograd, T. (2002). Time as essence for photo browsing through personal digital libraries. In *Proc. ACM/IEEE JCDL (2002)*, 326–335. ACM Press.
7. Hart, S. & Staveland, L. (1988). Development of NASA-TLX (task load index): Results of empirical and theoretical research, in P. Hancock & N. Meshkati, *eds*, *Human Mental Workload*, 139–183, Elsevier Science.
8. Igarashi, T. & Hinckley, K. (2000). Speed-dependent automatic zooming for browsing large documents, *Proc. UIST 2000*, 139-148, ACM Press.
9. Khella, A. & Bederson, B. (2003). A Zooming Image Browser for the Pocket PC. University of Maryland, Technical Report: www.cs.umd.edu/~akhella/pocketphotomesa.pdf
10. Krishnan, A. (2003*). Interactive Time-Centered Workspace Visualization*. MCMS Thesis Department of Computer Science, University of Waikato, Hamilton, New Zealand. Call no. TK7882.I6K75.
11. Platt, J C., Czerwinski, M. & Field, B. (2002). PhotoTOC: Automatic Clustering for Browsing Personal Photographs, *Microsoft Research Technical Report MSR-TR-2002*.
12. Rekimoto, J., Oba, H. & Ishizawa, T. (2003). SmartPad: a finger-sensing keypad for mobile interaction, *Proc. of Human Factors in Computing Systems, CHI 2003*, 850–851. ACM Press.
13. Rodden, K & Wood, K. (2003). How do people manage their digital photographs? *In Proc. of Human Factors in Computing Systems, CHI 2003*, 409-416, ACM Press.
14. Wang, M-Y, Xie, X, Ma, W-Y & Zhang, H-J. (2003). MobiPicture - Browsing Pictures on Mobile Devices, *Proceedings 11th ACM International Conference on Multimedia*, 106-107, Berkeley, CA, USA, Nov. 2003.
15. Whittaker, S. & Sidner, C. (1996). Email overload: exploring personal information management of email, in *Proceedings of Human Factors in Computing Systems, CHI 1996*, 276–283. ACM Press.