Lost in Memories: Interacting With Photo Collections On PDAs

Susumu Harada, Mor Naaman, Yee Jiun Song, [†]QianYing Wang, Andreas Paepcke
Stanford University
{harada, mor, yeejiun, paepcke}@cs.stanford.edu [†]wangqy@stanford.edu

ABSTRACT

We developed two browsers to support large personal photo collections on PDAs. Our first browser is based on a traditional, folder-based layout that utilizes either the user's manually created organization structure, or a system-generated structure. Our second browser uses a novel interface that is based on a vertical, zoomable timeline. This timeline browser does not require users to organize their photos, but instead, relies solely on system-generated structure. Our system creates a hierarchical structure of the user's photos by applying time-based clustering to identify subsets of photos that are likely to be related. In a user experiment, we compared users' searching and browsing performance across these browsers, using each user's own photo collection. Photo collection sizes varied between 500 and 3000 photographs. Our results show that our timeline browser is at least as effective for searching and browsing tasks as a traditional browser that requires users to manually organize their photos.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces, input devices and strategies, interaction styles.*

General Terms

Algorithms, Management, Measurement, Design, Human Factors.

Keywords

Handheld devices, mobile computing, pen and tactile input, digital photos, photo browser

1. INTRODUCTION

As digital cameras become increasingly prevalent, large personal libraries of digital photographs are becoming more common. Like calendars, photographs are most useful when available at a moment's notice. Personal photographs in particular are often shown spontaneously during the course of a conversation with friends and family. As portable devices with credible processing, storage, connectivity and display capabilities emerge, these devices are becoming potential platforms that enable users to have their entire digital photo collection available to them at all times. Examples of devices that can potentially be used for photo browsing include PDAs, cellular phones, and of course, digital cameras. In this work, we explore the PDA as a platform for a photo browser application, and believe that our results can be extended to other similar platforms with little change. We are not concerned with the problem of transferring photographs to the PDA, but instead focus on the interface issues that arise, assuming that all of a user's photographs are already available on the PDA.

The instant access requirement presents an interface challenge: it is difficult to design a system that allows instantaneous access and retrieval of photographs, particularly on a device that has limited screen space. This problem is exacerbated by users' increasing capacity to create digital photographs, which results in rapidly growing photo collections.

Interacting with photo collections on a small device presents an additional problem. Most approaches to compensating for the lack of screen real estate involve the use of textual metadata, such as what the photos depict. However, casual photographers often lack the time and inclination to create such textual metadata. Image analysis techniques that generate metadata are still immature. In addition, the entry of search terms on PDAs is cumbersome at best, making the interaction with metadata frustrating.

It is also inherently difficult for users to manipulate photo collections on small devices such as PDAs, making it unreasonable to expect users to manually organize their photos on the device itself. This problem can be tackled either by having users pre-organize their photographs on a more generous platform, most probably their desktop or laptop computer, before downloading that information onto their PDAs; an alternative approach uses the metadata that digital cameras embed in digital photos [1] to automatically induce the structure of the photo collection without user intervention. As the size of photo collections increases, the second approach will likely prove to be the more viable solution to this problem. In this paper, we explore one approach towards solving this problem of browsing and searching through large collections of personal photographs on a small screen.

We limit ourselves to personal photo libraries; that is, all images that were taken or collected by the person who then interacts with the images on the PDA. Chronology has been shown to be a very

JCDL'04, June 7–11, 2004, Tucson, Arizona, USA.
Copyright 2004 ACM 1-58113-832-6/04/0006…$5.00.

[1] Most current digital cameras embed metadata in image files using the Exchangeable Image File Format (EXIF).

important factor in users' interaction with their own photo collections [12]. Work in [3][12] confirms that events are users' natural way of thinking about their photos.

In previous work [5], we confirmed that single-photographer photo creation time can be related to events by well-known mathematical methods, which we summarize below. This work uses the time metadata embedded in all digital photographs to automatically organize photo collections into events based on their inherent time structure.

While [5] explored the use of photo time as a foundation for *desktop* browsers that arrange and manage photos automatically, we report now on our experience with the use of time as an organizational principle on PDAs. As is usually the case, even if the underlying algorithms transfer in part to the PDA, user interfaces need to be re-thought from scratch.

Time, of course, is not the only, and reportedly not even the most powerful memory cue. Stronger memory-to-image associations seem to be, in order of power, the factors 'who' and 'where' [15]. In addition to time references, visual cues in the form of representative photos must also have a pervasive screen presence.

On these principles, we construct a series of time-informed interfaces, the first of which we call the Timeline browser (Figure 4). Timeline is an interface designed specifically to take advantage of the automatically generated time-based organization structure. We compared this new interface to a baseline of more traditional, folder-based interfaces (see, for example, [9][11]). To this end, we pitted the Timeline browser with its automatically generated hierarchical structure against a traditional browser under two conditions. For the first condition, we had subjects operate the traditional browser over their own photo collections as they had organized them on their personal computers. For the second condition, we replaced the owner's storage organization with one that was created automatically by our algorithm. This experiment allows us to test the efficacy of both our automatic organization and the time-informed interface. We show that, in most cases, a system that uses automatic photo organization can perform as well as one that requires users to manually organize their photos. In some cases, the former can even outperform the latter.

The rest of the paper is organized as follows: Section 2 describes the photo organization algorithm; Section 3 presents the two photo browsers; and Section 4 describes the goals and design of our experiment. Section 5 presents the results of the experiment, and Section 6 discusses and describes the implications of the results. Section 7 provides an overview of related work. We conclude and present future work in Section 8.

2. CLUSTERING ALGORITHM

The clustering algorithm we use in this paper was originally presented in [5] and was slightly modified for this paper to adapt to the constraints of a PDA. We include a brief summary of the algorithm here for completeness; for details, please refer to [5].

The clustering algorithm is based on the observation that photographs are taken in bursts. For example, one person's photo-taking activities during a three-month period may exhibit the following pattern. She first takes pictures during a one-week trip to California. This is then followed by five days when no pictures are taken. That period of inactivity is followed by a flurry of photos during her daughter's birthday.

The clustering algorithm would partition the resulting collection into two clusters, A and B. Cluster A would contain the trip photos, and cluster B would comprise the birthday shots. Within each cluster, picture taking will usually again be lumpy, this time at a finer time granularity than trip and birthday. The algorithm will therefore recursively decompose A and B into smaller clusters. The California trip might be partitioned into three sub-clusters that include, respectively, images taken on a one day visit to San Francisco, a series of photos shot during a three-day road trip, and photos of the last day's travel home. All this clustering is based purely on the time when images were taken, not on image recognition.

More technically, the initial clusters are created by detecting gaps of more than 24 hours between two consecutive photos. The intuition behind this first pass is that picture-taking events are usually more than 24 hours apart from each other. Then, the photos in each cluster are examined further to detect outliers - time gaps between two consecutive photos that are considerably larger than the average. When such an outlier is found, the cluster is divided further at the detected time gap. This process is applied recursively for each cluster until each cluster holds less than 30 photos and spans less than eight hours. Thirty is, not coincidentally, the number of thumbnails we can fit on one PDA screen.

Of course, this algorithm is based on heuristics and is not perfect. If the birthday event took place immediately upon the return from California, the two will fall into the same high-level cluster. However, in most cases the clustering algorithm will detect these as distinct events when analyzing the cluster during the next step of the algorithm. In fact, often times the algorithm will actually split events more than necessary; it is possible, for example, that the algorithm will split the different days of the California trip into completely distinct events (i.e. resulting in four high-level events: the San Francisco visit, the road trip, the travel home, and the birthday).

3. BROWSERS

As mentioned above, we implemented two browser interfaces: the traditional Baseline browser, and a novel Timeline browser (TL). The Baseline browser was implemented in two variants, which we call Baseline Manual (BM) and Baseline Automatic (BA). We designed the interaction using three views that correspond to three main phases of search. This three-view framework is often used in search and browsing interfaces [2][16]: an opening game during which the user navigates to the general vicinity of the target photo, a middle game where the search is narrowed, and an end game where users examine an individual photo.

3.1 The Baseline Browser

The Baseline browsers are designed in a fashion similar to folder-based photo browsers available today for the PDA [9][11]. BM (Baseline Manual) is based on the user's own organization, while BA (Baseline Automatic) is based on a system-generated organization. We present the BM interface briefly, and then describe the difference in photo organization between BM and BA, and its effect on the interface.

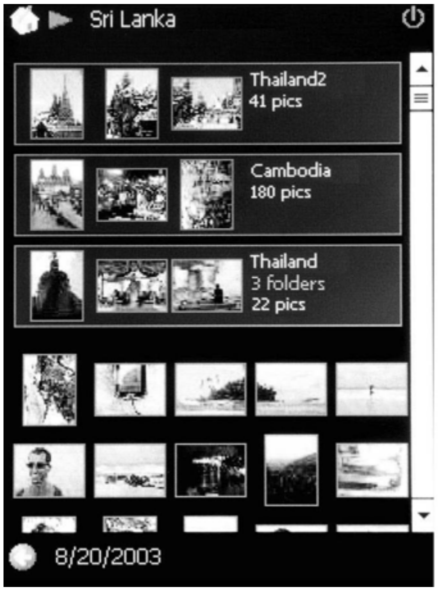

Figure 1. Opening-game screen for the Baseline Manual browser

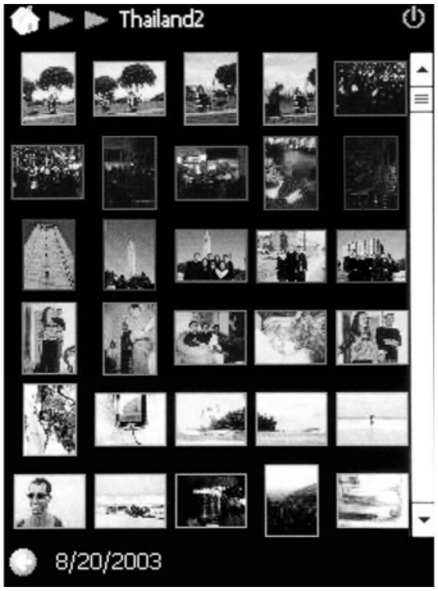

Figure 2. Middle-game screen for the Baseline Manual browser

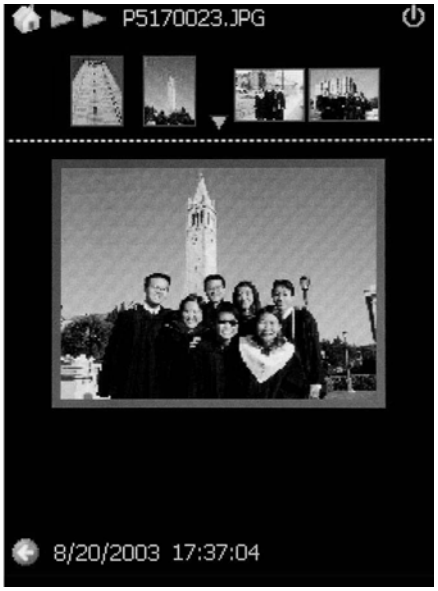

Figure 3. End-game ("Preview") screen for the Baseline browsers

The opening game for BM, shown in Figure 1, lists all folders and photos under the current folder. Instead of showing a folder icon, each folder is represented as a button control (the first three rows in Figure 1), each consisting of three sample photos, the name of the folder, number of subfolders (if any), and the total number of photos contained in this folder. Tapping on a folder will drill down into it, showing its subfolders and photos in the same fashion. Below the rows of the folders, we display a grid of thumbnails of all photos within the current folder.

At all times, the top of the screen displays a series of red arrowheads to indicate how deep the current folder is situated in the hierarchy. The 'back' and 'home' buttons carry the usual meaning of climbing one level up and returning to the top of the hierarchy respectively.

In the BM browser, the middle game is just a special case of the opening game, where there are no more subfolders in the current folder. As shown in Figure 2, only the grid of thumbnails is shown. The user can scroll down the grid if it spans more than one screen.

At any point, tapping on one of the photos in the grid will take the user into the end game (Figure 3). In this screen ('preview screen'), the selected photo is enlarged. Four other photos appear in a row at the top. These are the next and previous two photos, included for context and quicker navigation. Tapping any of the four replaces the enlarged photo with an enlargement of the tapped photo. The context changes accordingly. Tapping on the enlarged photo causes that photo to fill the entire screen. Another tap will return the user to the 'preview screen'.

Baseline Automatic (BA) is a variation on BM that uses the system-generated structure instead of the user's own photo organization. Note that the system-generated structure is pre-computed using the clustering algorithm described in the previous section, rather than generated on the fly. There are a few differences between the BA and BM browsers. First, in BA the photos appear only in the lowest level of the hierarchy, as opposed to BM in which photos may appear at any level. Therefore, we have a distinct opening game where users only choose between subfolders to drill down into. When there are no further subfolders (middle game), users see a grid of photos in the current folder. The second difference between BA and BM is BM's lack of folder names in the opening game. Instead we show the time span of the folder (the times when the first and last photos in the folder were taken).

For an illustrative example, we revisit the running example of the California trip. In the BA opening screen, the user will see two "folders" – one corresponds to the California trip, and the other to the birthday event. Each is identified by the time span, the beginning and end of the event. When the user clicks on the folder corresponding to the California trip, she drills down the hierarchy – now the screen will show three folders corresponding to the three California sub-events. Clicking on one of the sub-events, say the San Francisco visit, may take the user to the middle game with all the photos taken during this San Francisco visit.

3.2 The Timeline Browser

The Timeline browser (TL) presents an innovative photo-browsing interface that relies heavily on the system-generated hierarchical organization described above. As in the BA browser, the system-generated structure is pre-computed, before being downloaded to the PDA. There are three main concepts behind the Timeline design. First, we enhance the role of time in browsing the collection by providing the user with maximal time context. Second, we reduce the number of photos shown on the screen, using the additional screen space to provide enhanced time orientation instead. Third, we do not overload the screen, deliberately preserving as much negative space as we can. Again, we hope this allows the user to focus on the time context and the few photos as clues. At least one study [12] concludes that on desktops, browsers should display as many thumbnails as possible. We depart from this advice in our PDA browser on the basis of page layout principles that warn of clutter. Our sense is that

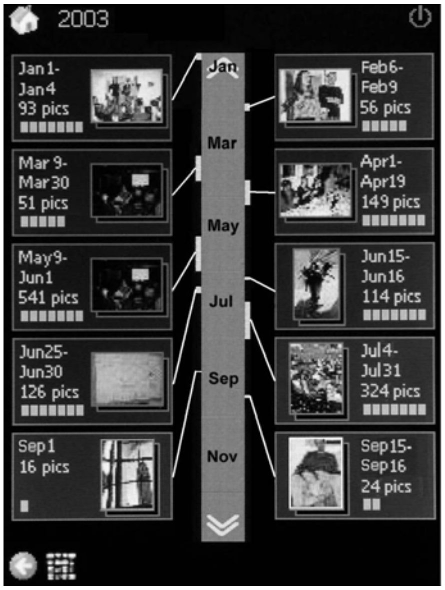

Figure 4. Year-view of the Timeline browser

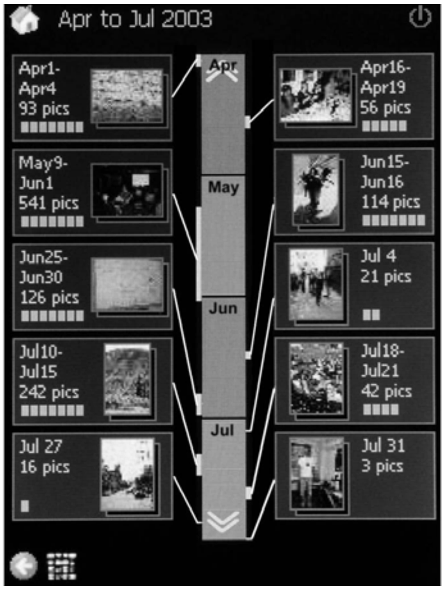

Figure 5. Four-month-view of the Timeline browser

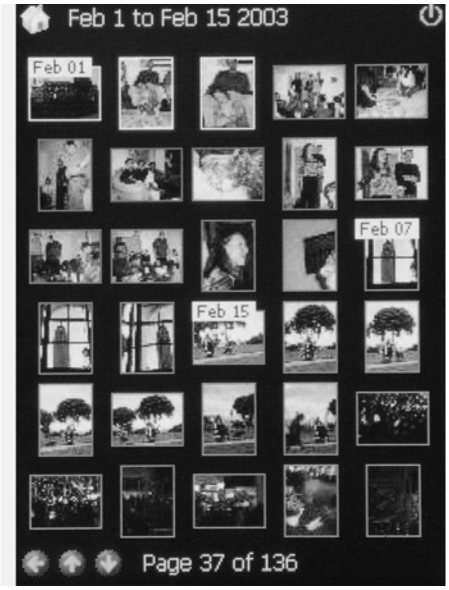

Figure 6. Middle-game ("Thumbnail") screen for the Timeline browser

clutter is at least as detracting on a small screen as, for example, on print media.

The opening game for the Timeline browser is shown in Figure 4. The screen is partitioned into three columns. In the middle is a vertical timeline, and both to the right and left of the timeline is one column of pictures. Each picture represents a system-generated cluster ("album"). The time range that an album covers is shown next to the picture. You also see how many photos are in the album. For a quick overview there is also a row of dots - the more dots, the more pictures are in the album.

The timeline in Figure 4 spans one year. At any screen, we display the time range that is covered by this screen at the top. There are never more than 10 albums shown on the screen. If the clustering algorithm generates more than 10 albums, we merge the closest ones until we have 10 albums. Again, keeping a limited number of clusters is geared towards maximizing the time context, trading it off with narrower representation of albums and photos on the screen.

Back to our running example, now the California and the birthday event may be merged, in the opening game, to one cluster. Say this event is represented on the screen using a photo from California. Hopefully, the time context will be strong enough to remind the user that this is where she has to look for her birthday pictures. The user memory cues are both the calendar data, and the proximity to the California event.

A white line runs from each album to the timeline. The point at which the line touches the timeline corresponds to the time when the photos in that album were taken. If there is a thick vertical line at the point of their intersection, the height of the vertical line indicates the time span during which the photos in the corresponding album were taken. For example, you can easily see the top left album in Figure 5 spans the first few days of April.

Tapping on an album will drill into a finer grained time range: the best range (in integral number of month, weeks or days) this album fits in. We are still in the opening game, only now we see the lower-level albums inside the tapped album and possibly other adjacent albums that fall into this time range, only the former are highlighted while the latter are not. Another way to display a finer grained time range is to drag the stylus on the timeline to select the desired time range. The selected time range will expand to fill the screen, showing in more detail the albums contained in it. For example, dragging the stylus over the April to July period of Figure 4 will bring the browser to a state shown in Figure 5.

We also implemented a way to quickly "peek" into the photos in an album during the opening game. The user can hold the stylus on a photo, and then start circling the stylus around the photo. This allows the user to flip in place through all the photos in that album. Users can circle forwards or backwards, by moving the stylus in a clockwise or counterclockwise direction.

Eventually, the user navigates and taps on a leaf album that contains just photos. Hopefully, she had done so when getting very close to the photo she was searching for. This final tap takes the application into the middle game. In the middle game (Figure 6), the user is presented with a grid of thumbnails. The thumbnails of the tapped album are aligned on the top, and are highlighted. From this point, the user can navigate back and forth (using the arrow icons at the bottom of the screen) through thumbnails of her entire collection, ordered by time. To aid in the navigation, dates are displayed above the thumbnails wherever the date changes. The range of dates represented on the screen appears on the top of the screen; the current position within the entire collection appears on the bottom.

There is also a shortcut that allows the user to switch quickly from the opening game into the middle game. From any level of the hierarchy in the opening game, tapping the thumbnail icon (second button from the left on the bottom row of Figure 4 and Figure 5) will take the user to the middle game, showing thumbnails beginning from the start date of the previous screen.

We decided not to implement the end game for the Timeline browser as we thought it might not be required. Thus, in the TL middle game, tapping a photo will simply cause it to fill the entire screen. Tapping again will return the user to the middle game view of thumbnails.

4. DESIGN OF EXPERIMENT

We wanted to test our photo browsers on large personal collections of digital photos. Since digital cameras have only become popular in the last few years, it was difficult to find subjects who had sufficiently large digital photo collections, and at the same time felt comfortable about giving us access to them. Our clustering algorithm also required that the metadata of the digital photos contain valid timestamp information. This made our subject search even harder, as this requirement disqualified subjects whose collections consisted of scanned images, rather than pictures taken with digital cameras, and subjects who had processed their photos using software which dropped the timestamp metadata.

At the end of the subject search process, we were able to recruit 17 subjects for our experiment, which is possibly the most extensive published research study, in terms of subject pool size, of photo browsers to date. The most important criterion for subject selection was that each subject was required to have a sizeable collection of digital photographs. Subject ages ranged from 19 to 48, with the highest representation in the 20s. Twelve subjects were male, and five were female. One of the photo collections comprised 474 photographs, but all others were significantly larger. Nine exceeded 1,000 photographs, and the largest collection contained 2972 photographs. The average collection size was 1,200 images.

For the experiment, we loaded the subject's photo collection onto a Hewlett-Packard H5500 Pocket PC. The relevant hardware specifications for this PDA are 128 MB of RAM, a 400MHz Intel XScale processor, and a 240x320/64K-16 bit (65K) color display with a viewing area of 57.6mm x 76.8mm (WxH). The operating system was Microsoft Windows Mobile 2003. We used a 512MB SecureDigital memory card for storing all the photos.

We reduced all photos to a resolution that fits the PDA screen, so photos are not difficult to manage and do not require too much unnecessary memory. At 240x320, each photo required about 20 KB of space, small enough to allow our memory card to store more than 25,000 photos. Since the price of memory cards continues to drop and their capacity continues to increase, we are not concerned (as we mentioned above) with issues of storing or transferring the photos onto the PDA, even if the resolution is not reduced.

The thumbnails we generated for the photos were 40 pixels long on one edge and about 30 pixels long on the other edge (depending on the photo's original ratio). The large image in the Baseline browser Preview Screen (Figure 3) was a 190x143 rendition that was not stored separately, but instead generated by scaling down the full-screen image file.

Our experiment followed a within-subject design. We exposed each subject to three experimental conditions: the Baseline browser with the subject's own, manually created photo organization (BM); the Baseline browser with the time cluster based organization that our system generated automatically (BA); and the Timeline browser (TL); also with the system generated organization.

Each subject completed two tasks on each browser. The first was a Search Task. We showed the subjects one of their own photos on a computer monitor and asked them to find that photograph in their collection by navigating on the PDA. We set a three-minute time limit for this task, and asked subjects to work as efficiently as they could.

In order to minimize experimenter bias during the selection of photos for the Search Task, we had a computer randomly select the photos from each subject's collection. The computer presented one random photo after another to one of the experimenters. The experimenter accepted or rejected each photo based on the following criteria: a photo was rejected if (1) the picture was taken at the same event as one that had already been chosen, or (2) the photo did not display any recognizable context, and the subject was therefore not likely to identify it in her collection. All other photos were accepted. The study in [10] followed a similar procedure and reports positive experience with this approach.

The second task was a Browsing Task. We asked the subject to select 'good' pictures for a collage that represented some portion of the subject's life. For example, we asked for pictures that would make a collage of the subject's friends. Other collage assignments were family, trips, special events, and scenery. We asked subjects to select photos from as broad a time span and set of occasions as possible. For this task we imposed a time limit of two minutes.

For each browser we had subjects complete the photo Search Task four times, having each subject find a different photo each time. We asked subjects to perform the Browsing Task once for each browser. The collage target (friends, events, etc.) was different for each browser.

Each time subjects completed both tasks under one of the conditions, they were asked to complete a questionnaire. We asked questions such as the helpfulness of the photo organization, the subject's degree of satisfaction, the amount of frustration, and adequacy of the allowed time. Answers were encoded on a 10-point Likert scale.

Notice that had we simply run subjects through the three conditions, BM, BA, and TL, the subjects would have been exposed to the Baseline user interface twice, while interacting only once with the Timeline interface (recall that BM and BA differ only in the organization of the folders, not in their interface). To avoid disadvantaging the Timeline interface we had subjects perform the Search and Browsing tasks twice on the Timeline interface. In addition to the fairness issue, this double exposure (TL1 and TL2) allowed us to study learning effects on the Timeline interface. The two exposures were never administered in immediate succession. We counterbalanced BM, BA, TL1, and TL2 by rotating the order in which successive subjects were exposed to these conditions.

Throughout the experiment we recorded quantitative data, such as the amount of time subjects spent in each view of the respective browser, the number of view switches they initiated during the tasks, and the number of times the subjects made use of special features, such as the photo flipping facility. We also recorded the completion time for each trial of the Search Task and the number of photos collected within the allotted two minutes in the Browsing Task.

5. RESULTS

In this section, we present the results of our experiments. We first list results that concern the speed and performance of the different tasks. We then present the users' subjective evaluation of the browsers and the tasks. We conclude this section with statistics about the usage of the features of our Timeline browser. Further discussion of the results will be presented in Section 6.

To summarize, the questions the experiment was designed to answer were as follows: (1) With a traditional PDA thumbnail interface, can an automatic, time-based photo collection analysis even approach the effectiveness of an organization that was created manually by the collection's owner? (2) How well can our novel Timeline interface with its underlying automatic organization support users in search and browsing tasks? (3) Do users of the Timeline interface improve their search and browse performance if they use the interface repeatedly?

Unless explicitly stated, all results in the following section are significant at the level of $p < 0.05$. We use the term 'highly significant' for results with $p < 0.01$.

5.1 Speed Performance

There was no statistical difference in search performance, i.e. the mean time to find a photo in the Search Task, between Baseline Manual (BM), Baseline Automatic (BA), and the first Timeline exposure (TL1). However, when subjects operated the Timeline for the second time (TL2), we observed a significant learning effect. The mean search time dropped by 24% when comparing TL1 and TL2. This performance gain lifted the Timeline significantly beyond both BM and BA. Comparing TL2 with BM brought TL2's search performance to a significant 29% improvement over BM.

We also tried to gauge how much the *Baseline* interface profited from learning. For this analysis we separated out the data of nine subjects who had first been exposed to BA and then to BM. This data thus maximally favored the Baseline interface in that these subjects had experienced the (Baseline) interface twice and additionally enjoyed their own manually created organization. We call this special data set BM2.

Comparing BM2 with TL2 we found that the double exposure to Baseline did benefit the Baseline interface. Its search time improved over the general Baseline average. However, the interface could still not outperform the automatically organized Timeline interface. Even when we allow for learning of the Baseline interface as well, TL2 still performs as well as BM2.

In addition, we checked if the situation changed when we separate the success rate of finding photos from the search speed. In other words, we look at the times of completed search tasks only. TL's speed was again equivalent to BM's, and TL's success rate was significantly (11%) higher than BM's.

As for the browsing task, the success rate for finding photos was a significant 14% higher for TL than for BM and BA. The second exposure to the Timeline interface did not significantly improve on this result.

As a measure of navigation performance, we counted the number of backtracking ('Back' or 'Home') operations during the search task. The fewer backtracking moves, the better the subject's orientation. The BA interface required 56% more backtracking than BM, and 62% more than TL. There was no significant difference between BM and TL. We again see improvement from TL1 to TL2. The difference in backtracking between TL2 and BM is significant.

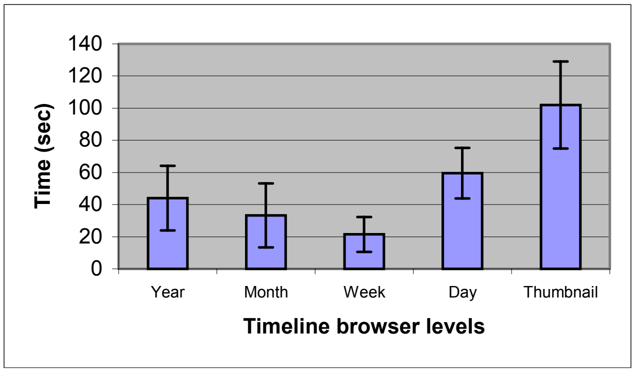

Figure 7: Average time subjects spent in each level of the Timeline browser hierarchy

5.2 Subjective Measures

Subjective measures also revealed familiarity effects from TL1 to TL2. The perceived easiness of the search task was higher for TL2 than TL1 (highly significant). TL2 and BA drew even for 'ease' of use. No difference was found between TL and BM.

We asked the subjects to rate how well they knew where to look for each picture when they started the search. With BM, subjects reported to have known best, at the outset, where the picture was, followed by TL2, TL1 and BA. The difference between BM and BA was significant, and the difference between TL2 and BA was marginally significant (p=0.07).

For the browsing task, most users felt that their own organization (i.e., BM) was more helpful than the system's organization (TL and BA). However, the performance data mentioned above did not reflect this perceived helpfulness. And there was no difference between the conditions in terms of satisfaction or perceived completeness of the browsing task.

Data from our questionnaires further shows that users did not show much preference between the different conditions. None of the photo organizations or interfaces was perceived as being more 'satisfying', 'helpful', or as enabling a more 'organized search process', except for the reported preference for BM with the browsing task.

5.3 Usage of Timeline Browser Features

Figure 7 shows the average total time users spent at each level of the Timeline application during the four Search Tasks. For example, users spent an average aggregate of 44 seconds in the year level (11 seconds per task, 17% of the time). In total, users spent an average of 58% of the time in 'timeline' view (the opening game on its different levels) and the rest (42%) in 'thumbnail' view (the middle game). The numbers were a little different for the Browsing Task, where users spent relatively more time (48%) in the thumbnail view than during the Search Task. The relative times spent in day and year views are similar to the Search Task, while the relative times spent in week and month view are reduced.

The subjects used the drag-zoom feature on average 3.8 times during TL1, and 2.9 times during TL2. Recall that this feature

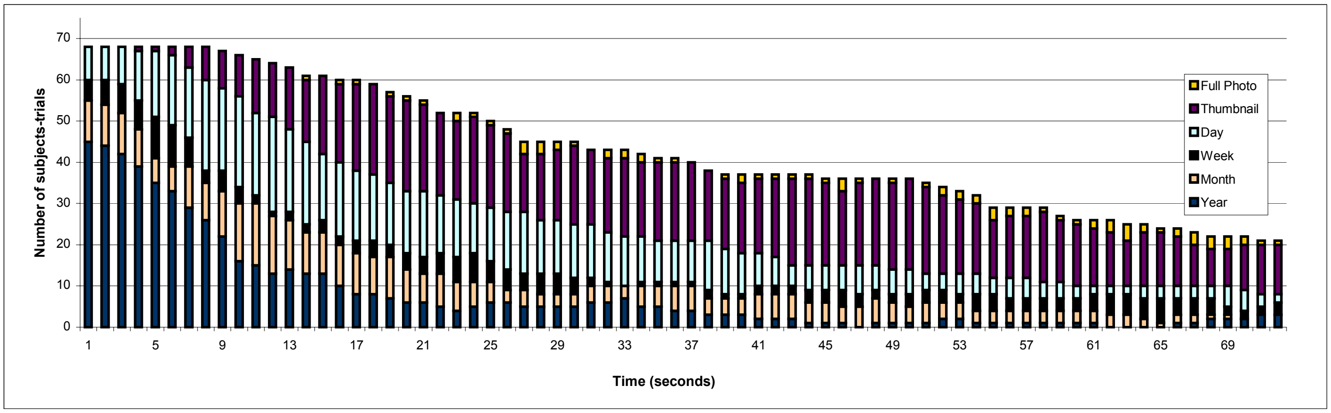

Figure 8. View dwelling time distribution for TL2

enables drilling into finer grained time ranges by dragging the stylus on the timeline to cover the desired range.

The 'flipping' feature, where users can tap the representative photo of the album and circle around it to see other photos from the album, was not extensively used. Only seven of the subjects tried to use this feature.

6. DISCUSSION

Our experimental results were quite encouraging. Not only did the Timeline interface, with its automatic photo organization, do at least as well as the Baseline interface with users' manual organization in all of our tests, it even performed significantly better in several areas. This suggests that the automatically generated photo organization, coupled with an appropriate interface, can compete with manually organized collections underneath a traditional interface. Our results informed our future work along three main dimensions. We learned (i) how manual and automatic photo organization and a time-informed user interface affect search and browsing performance. We learned (ii) that we could count on continued exposure to the Timeline browser to improve user performance even more. Finally, (iii) we gained insight into how much the interfaces encourage zooming behavior.

6.1 Time and Browsing Performance

The most important answers emerged from the task of finding known photographs. As we showed above, the automatic organization, coupled with the new interface of the TL browser performed at least as well as BM, the manually organized Baseline browser. Perhaps more surprisingly, our experiments showed that BM was also equivalent in the search time measure to the automatically organized Baseline, BA. Given the identical interfaces of BM/BA, their search time equivalence suggests that the automatic organization works well.

6.2 Learning Effect

We found a significant learning effect for the Timeline interface. This was not unexpected, as the Timeline interface was a new interface that users were not familiar with. We found that TL2 yielded a significant speed improvement of 18% over TL1, and a marginal improvement in success rate.

This learning effect can be attributed in part to the user's growing familiarity with the automatically generated structure of the TL interface. In other words, while using the interface, users both built better expertise *and* learned how the events are grouped together. This fact may suggest that users feel that the generated structure is meaningful, and are likely to learn and remember it. It should be mentioned that users are already familiar with their manual organization, thus "structure learning" was not expected when interacting with the BM browser.

Nevertheless, and despite the clear performance improvement, subjective data from the questionnaires showed that the subjects did not feel in TL2 that they were better able to predict ahead of the search where they would find a given photo. The "structure learning" hypothesis should therefore be further validated in future work.

6.3 Navigation

The small PDA screen requires frequent screen changes, either through scrolling or through 'drilling' down. Both of the interfaces in question emphasize drilling down. The time subjects spent at each level of depth is therefore crucial for understanding user behavior.

We observe a pattern when looking at the usage log of the Timeline browser. Recall that the 'levels of depth' in that interface corresponds to time granularities, such as 'year' 'month', etc. The distribution of time spent at these various levels shows sharp preferences, which are strongest for the Browsing task. As the results show, by far predominant in either task were extended stays in the 'day' and 'thumbnail' views.

In Figure 8 we illustrate exactly where in the time granularity and full vs. thumbnail views subjects spent time as they progressed through the search task in TL2. The x-axis shows the advancing search time. The y-axis is the number of subjects currently dwelling in any stage of the process of looking for a photo. The maximum value corresponds, naturally, to the number of subject trials (17 x 4 = 68 for the TL2 search task). Subjects drop (overall bar height is shorter) as time progresses and the required photo is found. Every bar is partitioned into six portions, each of which corresponds to one time granularity (for the opening game) or view (see legend). The relative size of each portion reflects the number of subjects that were browsing at that granularity during the respective bar's time range. Notice that at the outset (first click) most subjects stayed in the year view. By the 20th second, the majority of subjects had switched to day or thumbnail view. Half of the photos were found by the 47th second. This pattern suggests either that the year view was very effective (allowing users to focus quickly on a few days), or that the intermediate views need to be improved. Remember that the interface allowed drilling down from any level to any other level, e.g. from year to day level; therefore we believe that indeed, in many cases the month/week views were skipped altogether when users navigated the hierarchy.

Another measurement we studied was the frequency of descending and backtracking through the view hierarchy. This is a good indicator of false trails, which are usually costly in terms of time and user satisfaction. The experimental results show that BA exhibits significantly higher backtracking rates than either BM or TL. The lowest backtracking rate is evident in the TL2 condition, an encouraging result. This is one possible factor that accounts for the superior TL2 time performance.

6.4 Additional Subject Feedback

For a majority of the subjects the thumbnails were too small. We had purposely pushed the thumbnail size down to the very limit of possible acceptability. The rationale was that the subjects would be interacting with their own collections and thus should be able to recognize their photos with minimal level of detail. We will revisit this issue in the future.

There were quite a few suggestions made by the subjects for improving the interface. The specific screen that received the most number of suggestions was the Baseline browser end-game screen ('Preview Screen'). One of the suggestions was to provide more context about the thumbnails on the top row by increasing the number of preceding and succeeding thumbnails, and also to indicate the current index of the center photo (e.g. '12 of 47'). The subjects also expressed the desire to have the ability to scroll to the next screen-full of photos in the Preview Screen, instead of being restricted, as is currently the case, to scrolling at most by two photos.

We also received suggestions to add support for the circular flipping gesture in the Preview Screen so that the user can flip through the medium sized images in that view. There were also requests for a full-screen slideshow feature.

7. RELATED WORK

There are a number of commercial photo browser applications available for the various handheld device platforms. We reviewed six of the most popular commercial photo browsers on a popular PDA software website (http://www.handango.com). None of the browsers support automatic organization of photos, or allow browsing through the photos based on creation time. The features supported by the commercial browsers include thumbnail views, Windows-Explorer-style folder views, and slide show support. Two of the commercial browsers [9][11] also provide an intermediate screen similar to our Baseline browser Preview Screen. Most browsers are geared towards browsing a small collection of pre-selected photos, rather than the user's complete image collection.

On the other hand, quite a few research projects have studied photo browser applications for the desktop [1][6][7][10] or other platforms [13]. In [1], for example, the authors try to maximize screen utilization for displaying images in a hierarchical organization.

Some of these photo browser projects use timestamps of photos as a hint for photo organization [10]. However, most of the browsers rely primarily on the manual annotation/grouping of photos, and use time only as a secondary tool. Some commercial desktop photo browsers, such as Adobe's Photoshop Album, have begun to use time as a organizational tool, but often use time only as a hint for the ordering and density of photos, and do not use automatic tools to group photos based on time.

Additional time-based clustering techniques for photos have been suggested in [4][10][14]. The interface we present in this paper is orthogonal to time clustering techniques, and in fact, can be adapted to take advantage of different clustering methods.

8. CONCLUSIONS AND FUTURE WORK

Building on previous work for automatically organizing digital photographs, we developed a user interface for browsing thousands of photographs on a personal digital assistant, without requiring users to manually organize their photographs. In an extensive user experiment, we showed that the automatically organized system and interface perform *at least* as well as a traditional manually organized system in search and browsing tasks. Of course, our system eliminates the time and effort users typically need to invest, and often do not invest, in manually organizing their photographs. Specifically, we observe that users showed a decreased need for search backtracking and an increase in search success using our new time-based system. We are therefore convinced that our automatic photo organization scheme and timeline interface provide a solid foundation for further development.

Although our interface is designed to be scalable and extensible, we wish to determine how it actually performs with photo collections that comprises upwards of tens of thousands of photos and span many years. We hope to revisit these questions when users' collections grow to a suitable size so that such studies become feasible.

A few questions regarding our interface were not measured in the user experiments. For example, experiments were not designed to evaluate the usefulness of negative screen space versus using a lot of visual information with the risk of clutter. We will investigate the issue further in future work.

We are also interested in exploring ways to enable users to manipulate their photo collections on small devices, rather than just browse through them. For example, PDA users can use their time on the train to enhance the automatically created structure by annotating photos (e.g., with names of people in the photos).

Furthermore, other work in our group has demonstrated that for cameras enhanced with location capabilities such as a GPS, photo shot location metadata can be used to automatically label photographs [8]. This promises to be yet another dimension which can be used in the automatic organization of digital photographs. We are currently working on algorithms to create this organization, and are aiming to design the appropriate interface for the PDA.

Manual photo organization is a chore that many photographers dislike. The experiments described in this paper make us optimistic that significant progress can be made towards helping users with their photo mess.

9. ACKNOWLEDGMENTS

We thank the subjects of our experiment, and Matt Wang and Wendy Tobagus who helped parse our log files.

10. REFERENCES

[1] B. B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80. ACM Press, 2001.

[2] O. Buyukkokten, H. G. Molina, A. Paepcke, and T. Winograd. Power browser: Efficient web browsing for pdas. In *Proceedings of the conference on Human factors in computing systems*, 2000.

[3] D. Frohlich, A. Kuchinsky, C. Pering, A. Don, and S. Ariss. Requirements for photoware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.

[4] U. Gargi. Time-based analysis and event clustering. HP Tech Report Number HPL-2003-165, 2003.

[5] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002. Available at http://dbpubs.stanford.edu/pub/2002-4.

[6] H. Kang and B. Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *Proceedings of IEEE International Conference on Multimedia and Expo (III)*, pages 1539–1542, 2000.

[7] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. Fotofile: a consumer multimedia organization and retrieval system. In *Proceedings of the conference on Human factors in computing systems*, pages 496–503. ACM Press, 1999.

[8] M. Naaman, A. Paepcke, and H. Garcia-Molina. From where to what: Metadata sharing for digital photographs with geographic coordinates. In *Proceedings of the 10th Interational Conference on Cooperative Information Systems*, 2003.

[9] PhotoPals. Photopals photo album. http://www.photopals2002.com/.

[10] J. C. Platt, M. Czerwinski, and B. A. Field. Phototoc: Automatic clustering for browsing personal photographs. Microsoft Research Tech Report Number MSR-TR-2002-17, 2002.

[11] Resco. Resco picture viewer for pocket pc. http://www.resconet.com/picview.asp.

[12] K. Rodden and K. R. Wood. How do people manage their digital photographs? In *Proceedings of the conference on Human factors in computing systems*, pages 409–416. ACM Press, 2003.

[13] C. Shen, N. Lesh, and F. Vernier. Personal digital historian: story sharing around the table. *interactions*, 10(2):15–22, 2003.

[14] A. Stent and A. Loui. Using event segmentation to improve indexing of consumer photographs. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–65. ACM Press, 2001.

[15] W. A. Wagenaar. My memory: A study of autobiographical memory over six years. In *Cognitive Psychology*, pages 225–252, 1986.

[16] K.P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the conference on Human factors in computing systems*, pages 401–408. ACM Press, 2003.