

IPR No.: IPR2016-00500
Patent No. 7,864,163

EXHIBIT 1008

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL '04, June 7–11, 2004, Tucson, Arizona, USA.

Copyright 2004 ACM 1-58113-832-6/04/0006...\$5.00.

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 photos¹ 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

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

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



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

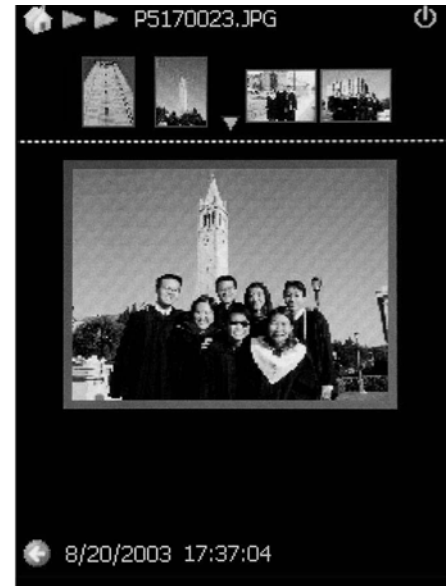


Figure 3. End-game ("Preview") screen for the Baseline browsers

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

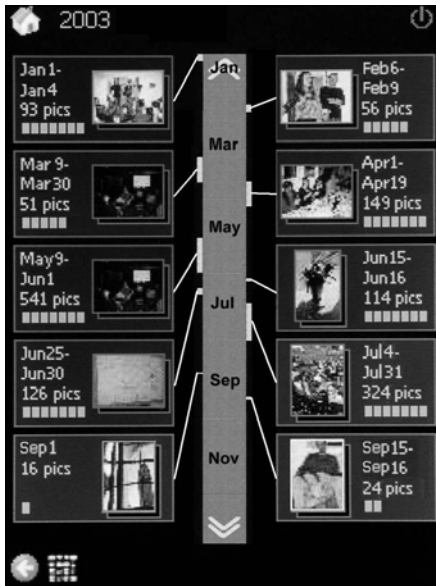


Figure 4. Year-view of the Timeline browser



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

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.