Simplifying the Management of Large Photo Collections

Andreas Girgensohn, John Adcock, Matthew Cooper, Jonathan Foote & Lynn Wilcox

FX Palo Alto Laboratory, 3400 Hillview Avenue, Palo Alto, CA 94304, USA

{andreasg, adcock, cooper, foote, wilcox}@fxpal.com

Abstract: With digital still cameras, users can easily collect thousands of photos. Our goal is to make organizing and browsing photos simple and quick, while retaining scalability to large collections. To that end, we created a photo management application concentrating on areas that improve the overall experience without neglecting the mundane components of such an application. Our application automatically divides photos into meaningful events such as birthdays or trips. Several user interaction mechanisms enhance the user experience when organizing photos. Our application combines a light table for showing thumbnails of the entire photo collection with a tree view that supports navigating, sorting, and filtering photos by categories such as dates, events, people, and locations. A calendar view visualizes photos over time and allows for the quick assignment of dates to scanned photos. We fine-tuned our application by using it with large personal photo collections provided by several users.

Keywords: Digital photo collections, automatic event detection, user-centered design.

1 Introduction

The rapidly increasing use of digital cameras is causing a corresponding increase in the number and size of personal digital photo collections. These collections routinely contain thousands of photos and require effective interfaces that facilitate browsing, manipulation, and sharing. Studies of user needs have found that users exploit various aspects of digital photography in usage scenarios such as sharing photos with friends and retrieving photos of important events such as birthdays and weddings (Kuchinsky et al. 1999; Graham et al. 2002; Schiano et al. 2002; Shen et al. 2002). Before sharing, people typically classify photos into those they wish to share versus those they intend to leave "in the shoebox." Thus, there is a demand for powerful tools to help users organize, classify, and browse their collections.

There are already many commercial and research applications supporting the organization of digital photos (ACD Systems; Apple Computer; Bederson 2001; Canon; Kang and Shneiderman 2000; Kuchinsky et al. 1999). While our application shares features with some of them, our goal has been to make organizing and browsing photos simple and quick, while retaining scalability to large collections. To that end, we have concentrated on areas that improve the overall experience without neglecting the mundane components of a digital photo organization application. We facilitate organizing and viewing large photo collections by automatically dividing photos into meaningful episodes or events, such as a birthday party or a trip. Though the events are automatically detected initially, the resulting boundaries can be manually adjusted if desired.

Our application presents photos in a vertically scrollable light table which shows thumbnails of the user's entire photo collection, with markers indicating the start of each event. A tree view for displaying events and other attributes (people, places, etc.) can be used to scroll the light table to a selected event, to sort photos by different categories, or to filter the set of visible photos to show only a particular category.

We fine-tuned our application by testing it with photo collections from six different users each with 500 to 1500 photos. In two of the collections, the photographers organized the photos in directories representing events. We used that information as ground truth for evaluating our event detection algorithm. One collection contained photos taken by several photographers at the same event made us aware of issues with mixing scanned photos and photos from cameras set to different time zones. To test scalability, we used our application with the approximately 4300 photos from all six collections simultaneously.

In the next section, we review several existing applications and algorithms for organizing digital photos and compare them to our approach. We then describe the automatic techniques and user interface mechanisms in our application. Afterwards, we discuss experiences encountered during the development and use of the application. We conclude with a discussion of future directions.

2 Related Work

There are numerous research and commercial applications for viewing and managing collections of digital photos. While our application shares features with some of them, our goal has been to make organizing and browsing photos simple and quick, while retaining scalability to large collections. To that end, we have concentrated on areas that improve the overall experience without neglecting the mundane components of a digital photo organization application.

2.1 Light Table

Several applications show photos in a grid-based light table (ACD Systems; Canon), either by reducing the size of portrait photos or by adding extra margins. PhotoMesa (Bederson 2001) shows all photos using cached thumbnails of different sizes for improved performance with a zoomable interface. iPhoto (Apple Computer) makes changing the image size in the light table very simple by placing a control underneath it to provide users with a choice between overview and detail.

We optimized the light table in our application for fast scrolling by caching and pre-fetching image thumbnails, and we offer three different image sizes to support the quick viewing of the entire photo collection. As a consequence, our application handles collections of thousands of images smoothly. In contrast to grid-based light tables, we pack photos into rows. To pack photos more tightly than iPhoto, we let adjacent rows overlap if they contain portrait photos in different areas such that the photos do not overlap.

Unlike other applications (e.g., ACD Systems), the photos in our light table can be re-ordered by the various categories (people, place, event) such that the specific sub-categories are visually separated and start with a marker. If a photo belongs to multiple sub-categories (e.g., a photo of several people), the photo appears in each of the sections corresponding to an applicable sub-category.

2.2 Automatic Event Detection

Many applications use a hierarchical file system or set of photo collections as their main means for grouping photos (ACD Systems; Apple Computer; Canon). In some cases, only the photos from a single directory can be viewed simultaneously (ACD Systems). Our application is designed to process a user's entire collection. We have emphasized time-ordered organization of the collection by events, such as birthdays or vacations, that are meaningful to the photographer. Other researchers have also used photos' timestamps as input to automatic organizing algorithms. The algorithms in (Graham et el. 2002; Platt et al. 2002) operate using an adaptive local threshold applied to the inter-photo time interval. Researchers at Kodak have developed an event segmentation algorithm based on clustering time differences using *K*-means (Loui and Savakis 2000).

Our automatic event detection algorithm is based on local self-similarity, as detailed in Section 3.2. In (Cooper et al. 2003), the algorithm was compared to Graham's and Platt's algorithms for automatic organization of two collections. The photographer's directory organization served as the ground truth for evaluation. While the other two algorithms had perfect recall (1.0) in finding event boundaries, they also detected many additional boundaries (precision 0.37). In contrast, our algorithm balanced precision (0.78) and recall (0.91) much better. Our automatic event detector also estimates event boundaries at multiple time scales. The interface allows the user to interactively explore the automatically detected event boundaries at the resolution of their choice.

2.3 Filtering and Categorization

Another common feature among photo organizers is metadata-based filtering to facilitate browsing and retrieval. PhotoFinder (Kang and Shneiderman 2000) supports form-based boolean queries. The FotoFile system (Kuchinsky et al. 1999) offers a tree view to display hierarchies of different categories such as events, locations, people, or dates. For our application, we chose to offer simple filtering based on categories shown in a tree view. Unlike FotoFile, the nodes in our tree view contain small representative images for the corresponding set of photos.

Our system is designed so that these categories can be assigned automatically if the supporting data is available, e.g., GPS data or reliable face recognition (Kuchinsky et al. 1999). The tree view can also be used for a bulk assignment of categories simply by dragging a selection of photos on top of a tree node, in contrast to the drag-and-drop labeling in PhotoFinder (Shneiderman and Kang 2000) in which labels are attached to specific regions within a photo.

We also provide keyboard shortcuts for quickly adding subjective ratings to photos. With a single key stroke, a user can rate a photo and advance to the next. No other photo application has this feature.

Like MyPhotos (Sun et al. 2002) and Photoshop Album (Adobe), our application provides a calendar view with photo thumbnails to support the browsing of photos. Unlike other applications, we optimize the use of space in sparse calendar views by magnifying the days of the calendar where images are available and by shrinking the areas where they are not. Also, instead of displaying a single thumbnail on each date, our calendar can indicate visually how many images fall onto a given day using a water-level icon. We also support changing photo dates via drag-and-drop onto the calendar.

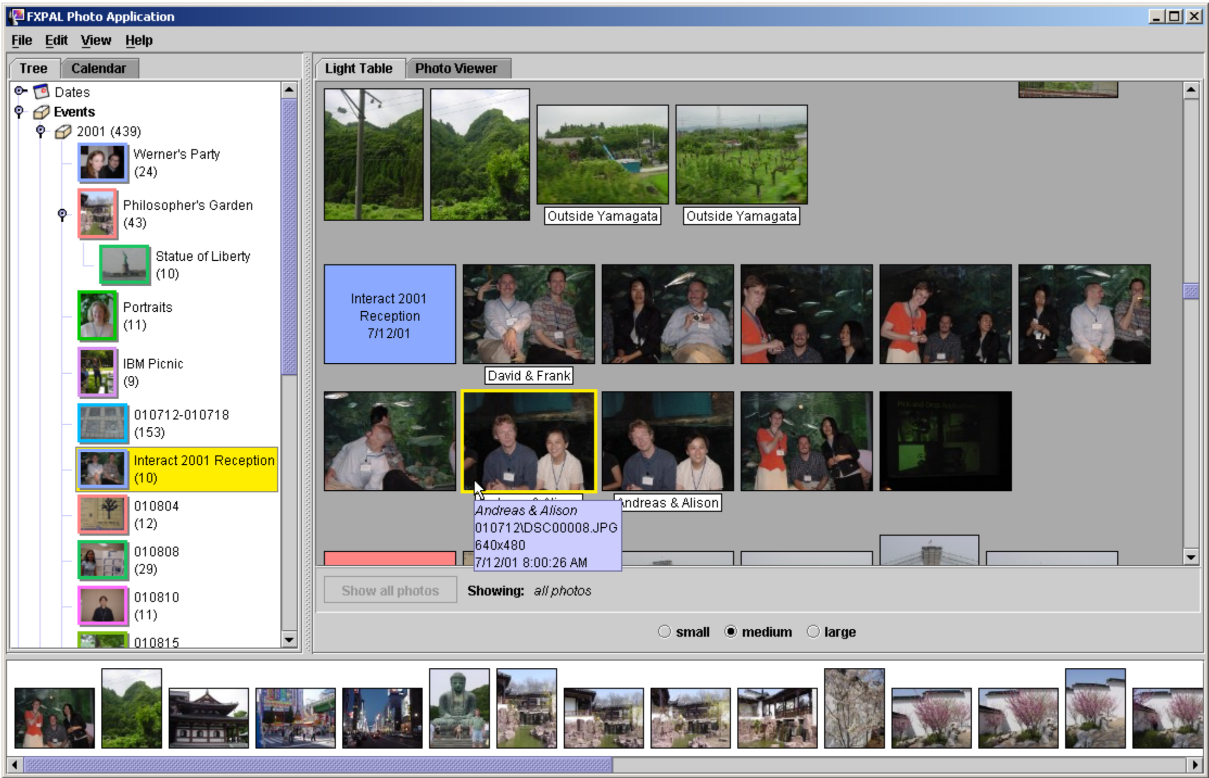

Figure 1: Event tree, light table, and photo tray views in the photo application.

3 Photo Application

Our application's interface presents three panes, as shown in Figure 1. The left pane allows users to view and scroll the set of events using a tree view. This pane also shows sets of people and place categories, as well as any other user-defined labels. By selecting the corresponding tab, this pane also shows a calendar-based visualization of the collection's events and photos (see Figure 10). Both tree and calendar view can be used for navigating and filtering photos.

The right pane is a vertically scrollable light table which, by default, shows thumbnails of the user's entire photo collection in time order with markers indicating the start of each event. Using tabs, this pane alternately provides a larger view of the currently selected photo. The categories in the left pane can be used to automatically scroll the light table to a selected event, or to filter the collection for photos with a particular attribute (people, place, etc.). Attributes can be added to a group of photos with a simple drag-and-drop operation.

The bottom pane presents a photo tray that can be used to collect and arrange photos for viewing or for exporting, e.g., as a Web page. The photo tray can be hidden to provide more space for the other panes.

The light table is optimized to efficiently show a large number of photos. Our application offers the ability to quickly scroll through the whole photo collection to get an overview. Image thumbnails are cached on disk in the three supported sizes and loaded on demand when scrolling. All time-consuming operations such as loading photos at startup are performed in the background so that users can immediately interact with their photos rather than having to wait for the application to complete its processing.

The basic organizational building block in our application is the event, such as a wedding, vacation, or family gathering. When acquiring new photos, our application automatically groups photos into events based on the photos' time stamps. Users can both adjust the granularity of the automatic event detection as well as create new events manually and assign photos to them.

3.1 Use Scenario

We now present a scenario to highlight several of the important features of our system. Imagine that a user of our application went on a trip to Tokyo and took pictures of friends. After returning from the trip, she

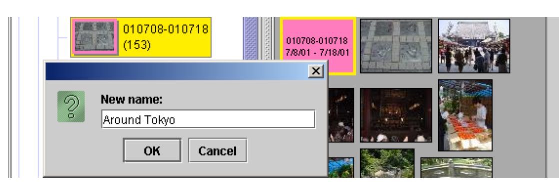

Figure 2: Naming an automatically detected event.

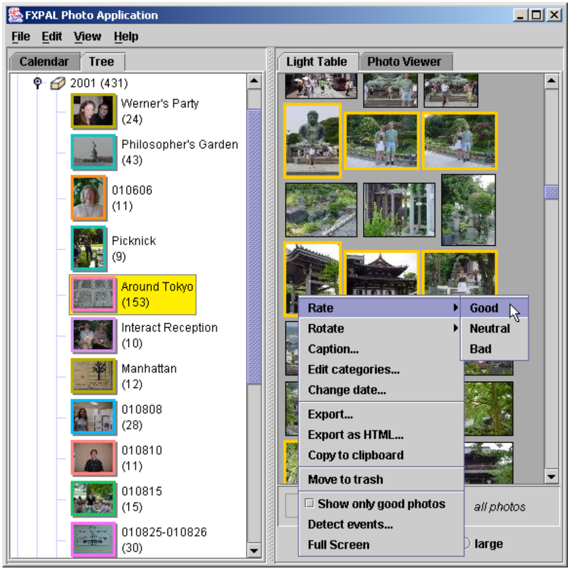

Figure 3: Rating photos.

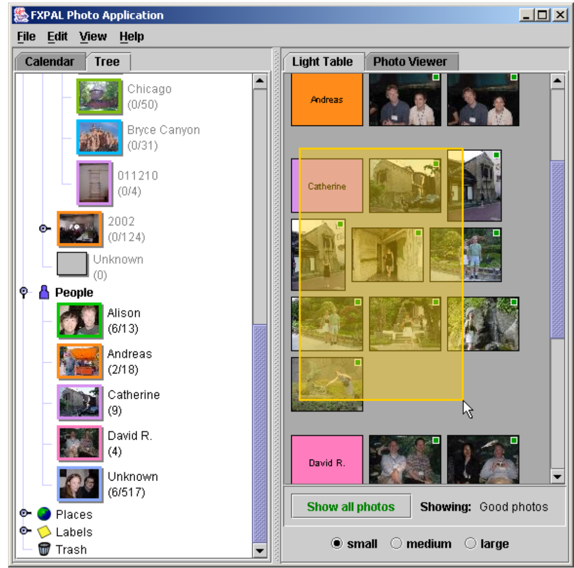

Figure 5: Selecting photos with wire frame.

imports the photos from the camera into our application along with photos from a company picnic that are also still on the camera. Once the photos are imported, the automatic event detector finds two events, one for the company picnic and one for the Tokyo trip. Figure 2 shows the user giving the detected events meaningful names.

Now, the user wants to find some nice photos to share with her friends. First, she selects several good photos in the light table with the mouse and rates them from the context menu (see Figure 3). She could also have rated them sequentially with the keyboard by pressing the "+" and "-" keys to rate a photo good or bad and advance to the next photo.

After restricting the light table to show only the good photos (as indicated below the light table in Figure 4), our user assigns the photos to categories representing the people depicted in the photos. In the tree view, she creates a new person category "Catherine" and drags several selected photos from the light table to the new category node to assign them to that category (see Figure 4).

Finally, our user creates a new node in the tree view under *Labels*, "Tokyo for Catherine," to group the photos to be shared. She sorts the light table by person by clicking in the *People* tree, and scrolls to the photos of Catherine (see Figure 5). She selects those photos with a wire frame and drags them to the label (see Figure 6). Now, those photos can easily be found by choosing the "Tokyo for Catherine" node in the tree. Those photos can be exported to a folder and shared, or copied to the clipboard and pasted into an email message as attachments.

3.2 Automatic Event Detection

In recent reports, researchers have found that organizing photos by time significantly improves users' performance in retrieval tasks (Graham et al. 2002; Gargi et al. 2003). Intuitively, we often describe photos by the events they document or the times they were taken. While events are difficult to define quantitatively or consistently, photographs from the same event are typically taken in close proximity in time. Example events include a birthday party or a trip to Disneyland.

Figure 4: Drag-and-drop to assign person.

Figure 6: Assigning a label by dragging photos.

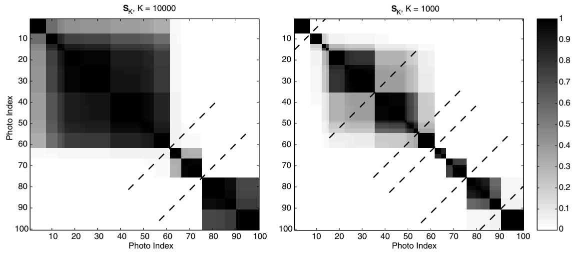

Figure 7: Similarity matrices for different intervals.

As well, visually dissimilar photos can often belong to the same event. For example, pictures taken at a wedding might contain outdoor and indoor shots from both a church and a separate reception hall. Thus, content-based image similarity is often less useful for photo clustering or event detection than metadata. Unlike film, digital photographs typically include metadata, such as the time and date, in a standard image header such as Exif (JEIDA 1998).

We adapt a similarity-based media segmentation algorithm (Cooper and Foote 2001) to hierarchically cluster photographs with similar (i.e., proximal) timestamps. This approach makes no assumptions about the distribution of the timestamps. The first step is to sort the photos into time order. Assume the i^{th} photo in time order has timestamp t_i. To automatically determine temporal structure in the photo collection, we construct a family of similarity matrices S_K whose elements are calculated as follows.

$$S_K(i, j) = e^{\frac{-|t_i - t_j|}{K}} \tag{1}$$

By construction, the rows and columns are indexed by photo, in time order. Thus, the photo index also runs along the main diagonal of each similarity matrix. Figure 7 shows two similarity matrices computed from 100 digital photographs taken over three months' time. Dark blocks of high similarity along the main diagonal indicate sequential clusters of similar photographs. Corners between the dark squares along the main diagonal indicate boundaries between two groups of photos.

In Figure 7, event boundaries appear as checkerboards along the main diagonal. To cluster the collection into groups of similar photos, we calculate a photo-indexed novelty score following (Cooper and Foote 2001). Peaks in the novelty score correspond to the boundaries between adjacent groups of photos that each exhibit high within-group temporal similarity, and low between-group similarity, as measured by Equation 1. The dashed lines in Figure 7 indicate the location of the automatically detected cluster boundaries along the main diagonal of each matrix.

We locate peaks in the novelty score at each scale, performing the analysis from coarse scale to fine (decreasing *K*). To build a hierarchical set of event boundaries, we include boundaries detected at coarse scales in the boundary lists for all finer scales. This procedure results in a list of event boundaries at multiple resolutions. Ultimately, we present users with the boundaries from a single resolution. To determine the "goodness" of the boundaries for each time scale, we calculate a confidence measure from the average within-class similarity and the between-class dissimilarity of the event clustering (Cooper et al. 2003).

Clustering the photos at multiple scales enables flexible user interfaces that allow users to organize their photo collections at different time scales (indexed by the parameter *K*). We give users the option to override our confidence measure for picking the right granularity and to have the system detect events at a coarser or finer resolution. Such a feature is important in situations were users might disagree on the appropriate event boundaries for a given set of photos. For example, a week-long trip through Europe might be seen as a single event by one user whereas another user might consider the visit to each city to be a separate event. Because our approach is hierarchical, we could also show nested events in the tree view in such situations.

When photos are imported into the application, e.g., from a digital camera, events for them are automatically detected. To be able to add the imported photos to existing events, events for all photos are detected but they are only assigned to the newly imported photos. Existing events are reused and new events with generated names are added for photos that do not belong into any of the existing events.

3.3 Light Table

The light table shows the whole collection of photos in a vertically scrollable window (see the right pane in Figure 1). In contrast to grid-based light tables, we pack photos into rows. We let adjacent rows overlap if they contain portrait photos in different areas so that the photos do not overlap.

We chose to offer three different image sizes for the light table to provide users with a choice between overview and detail. This allows us to precompute high-quality, anti-aliased thumbnails in the different sizes for increased performance. We also preload thumbnails centered around the current view to avoid a delay when scrolling the light table a short distance. We keep the smallest thumbnail size permanently loaded to provide a low-resolution proxy until the correct size thumbnail is loaded. As a consequence, our application handles collections of thousands of images very smoothly.

Depending on the criterion the light table is sorted by, colored markers delimit the start of each category (such as days or events) in the light table (see Figure 5 for a light table sorted by person). In cases where a photo belongs to multiple categories

Figure 8: Filtering photos by person.

(e.g., a photo of several people), photos are shown several times in the light table, appearing in the section for each relevant marker. It is also possible to filter the photos shown in the light table such that only photos belonging to a particular category are shown (see Figure 8). The "Show all photos" button below the light table can be used to quickly see all photos.

Users can select multiple photos in the light table and collectively rotate them, export them to a Web page or a file folder, attach captions, or assign them to categories. Categories can be assigned either through a detailed dialog (see Figure 9) or by dragging the selected photos to a category node in the tree view (see Figure 6).

The photo viewer can be displayed in place of the light table to view a single picture at a time at medium (640x480) resolution. While in the photo viewer, operations can be performed on the current photo as if it were selected in the light table. In fact, the photo viewer and the light table are synchronized such that the photo shown in the viewer is selected in the light table. We also offer a full-screen slide-show view that provides transitions between photos and automatic advancement to the next photo.

3.4 Rating Photos

People routinely separate printed photos into classes before sharing them with others, putting the best shots in the family album and rest in the shoebox. With digital photos, this kind of classification is even more critical because there are more ways to share and more photos (people shoot more with digital cameras than their analog counterparts).

Our application helps people rapidly perform this task by providing a single-key rating interface. Initially all photos are rated "neutral". By typing the

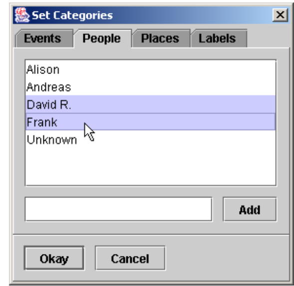

Figure 9: Adding a person to a photo.

"+" key the selected photo is rated "good" and the selection moves to the next photo. Similarly the photo can be rated bad by typing the "-" key. This sequential quick rating method can be used either in the light table or in the photo viewer mode. If photos are rated in the photo viewer, there is a brief pause after each rating to allow the user to see the applied rating icon before moving on to the next photo. A selected group of photos can also be rated good, bad, or neutral through the context menu (see Figure 3). Photos labeled as "good" ("bad") are distinguished in the light table with a small green (red) box. Photos labeled "bad" are also faded out to make it easier to ignore them. It is also possible to set the light table to show only "good" photos.

3.5 Tree View

The tree view (left pane in Figure 1) gives the user a way to visualize the photo categories and navigate the light table in terms of those categories. The tree view has a sub-tree for each of several different categories: *Dates*, *Events*, *People*, *Places*, and *Labels*. The first two of these categories, *Dates* and *Events*, are distinguished from the others in that a photo may only have a single date, and may only belong to a single event, while a photo may belong to any number of *People*, *Places*, or *Labels*.

The currently selected sub-tree determines how the photos are sorted in the light table. Clicking on a tree node scrolls to and selects the first photo with the selected value or tag. Subsequent clicks on the same tree node advance the selected photo sequentially through the photos with that tag. The lowest exposed node along each path shows a thumbnail of the first image found under that node. The color of the thumbnail border matches the color of the corresponding tag label in the light table. Intermediate nodes along an expanded path do not show thumbnails to save display space (see "2001" in Figure 1). Each node displays how many photos belong to that category. If only some photos are shown in the light table, each node also displays how many of its photos are currently shown and is grayed out if none of its photos are shown. Nodes (other than date nodes) can be renamed and new nodes can be created through context menu items.

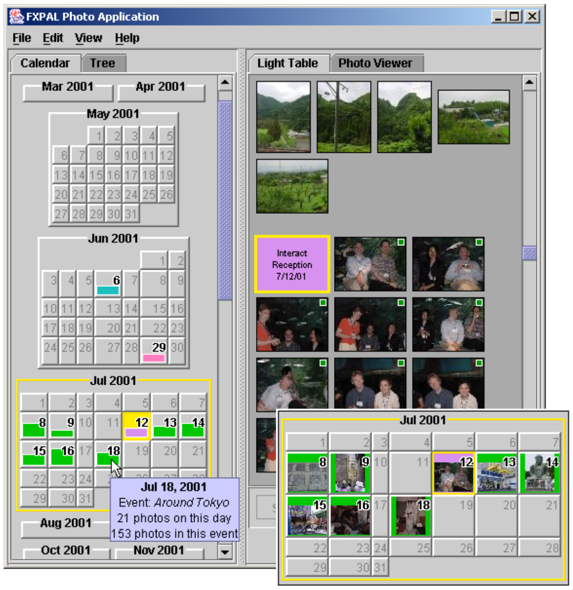

Figure 10: Calendar view with water-level icons. The thumbnail mode is shown in the inset.

All leaf nodes in the tree are drag-and-drop targets. A selection of photos dragged from the light table to a leaf node (non-leaf nodes expand automatically when dwelled over) have the corresponding tag applied. Each sub-tree contains a node labeled "Unknown". Dropping a selection of photos onto this node erases all applied tags in that category.

The user can elect to show only photos that are below a given node through the node's context menu (see Figure 8). By doing so, the light table will show only photos with a particular tag, or from a particular date. While filtering on a category value, the tree and light table still function with the other category types. For example, the user can show only photos from 1999, but then view those photos sorted by *Places* or *People*.

3.6 Calendar View

The calendar interface provides a way to visualize and manipulate the photo collection through the familiar context of a wall-calendar. The calendar represents the empty space between events in a way that is lacking from the tree-based view. For each calendar date, the thumbnail of a representative image for that date is displayed on the calendar. Alternatively, the calendar days can be filled with color to a horizontal level reflecting the number of images for that date, similar to a "water level" (see Figure 10). We also experimented with varying the color intensity and luminance to indicate of the number of images available on that date. A novel technique to maximize the use of space in sparse calendar views is to expand the days of the calendar where there are images and shrink the areas where there are not.

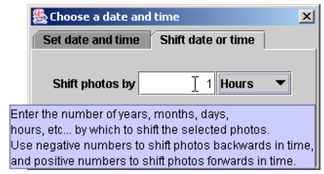

Figure 11: Shifting to a different time zone.

Just like a date node in the tree view, selecting a day in the calendar scrolls the light table to the first photo from that date. Subsequent mouse clicks advance the selection through the photos for that date. When the user wants to assign photos to an event by hand, the calendar provides a natural interface for selecting time-contiguous groups of photos. By selecting a group of days in the calendar, the photos on those days are selected, and can then be assigned to an event or other label.

The calendar view may also be used to apply dates to undated or incorrectly dated images. Most developing labs provide scanning services that typically generate undated images. A user may want to merge dated images from digital cameras and undated images from film cameras or from digital images whose time and date information has been removed (e.g., by manipulating it with an image editor). Photo dates may be set by dragging a selected image or images onto a date in the calendar. For dealing with cameras set to a different time zone or with an incorrect clock, we also offer a dialog to shift the time for the selected photos by a specified number of days, hours, minutes, or seconds (see Figure 11).

4 Experiences

We received personal photo collections from six different users each with 500 to 1500 photos. At various stages of our application development, we met with some of those users to discuss how they would like to see their photos in our application. Several users liked the fact that they finally could see their entire photo collection in the light table rather than having to open one folder at a time. The automatic event detection was also seen as a very useful feature.

One collection in particular challenged our views of a *personal* photo collection. It contained about 1500 photos taken at a wedding weekend with seven different digital cameras. Some of the photos were taken with analog cameras and scanned in. The clocks of the digital cameras were set to four different time zones. To see the different wedding activities in order, the times for the photos had to be adjusted. Our feature for shifting time zones (see Figure 11) turned out to be very useful for the digital cameras. Because a granularity of one day was insufficient to separate the wedding activities, we could not effectively use the calendar view to adjust the times for the scanned-in photos. Instead, activities in those photos had to be identified visually so that small groups of photos could be assigned the correct time through the date change dialog. A more powerful method for adjusting photo dates on a finer resolution is called for.

In observing people's use of the photo application, we noticed interesting behaviors. For example, one user noticed several photos in sequence that were taken from the same camera angle a few seconds apart. He was amused by rapidly switching through those photos in the photo viewer such that the movement between the photos appeared to be animated. Automatically created presentations of this sort could be a worthwhile added feature for our application.

The photo application is currently being prepared for a larger use study from which we expect to gain many additional insights.

5 Conclusions

In this paper, we described a photo management application aimed at facilitating the management of large photo collections by implementing automatic organization techniques, powerful but concise interface methods, and optimized data handling. Our automatic event detector produced results that closely matched the way in which users grouped their photos in folders. Our user interface provides several means for browsing, navigating, sorting, and filtering photos that scale up well to collections containing thousands of photos.

Initial user response has been very positive and indicates that our application addresses some of the perceived deficiencies of existing applications. Future work will continue to refine and augment the application in response to user feedback and also introduce novel and compelling methods for creating media objects for sharing and presenting photos such as slide shows set to music.

References

Adobe. Photoshop Album. http://www.adobe.com/products/photoshopalbum/overview.html

ACD Systems. ACDSee. http://www.acdsystems.com/English/Products/ACDSee/

Apple Computer. iPhoto. http://www.apple.com/iphoto/

Bederson, B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and BubbleMaps. Proc. ACM User Interface Software and Technology (UIST), pp. 71-80.

Canon. ZoomBrowser EX. http://www.powershot.com/powershot2/software/ps_pc_software.html

Cooper, M. and Foote, J. (2001). Scene Boundary Detection Via Video Self-Similarity Analysis. Proc. IEEE Intl. Conf. on Image Processing, pp. 378-381.

Cooper, M., Foote, J., and Girgensohn, A. (2003). Automatically Organizing Digital Photographs Using Time and Content. Proc. IEEE Intl. Conf. on Image Processing, to appear.

Gargi, U., Deng, Y., and Tretter, D.R. (2003). Managing and Searching Personal Photo Collections. Proc. SPIE Storage and Retrieval for Media Databases, pp. 13-21.

Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T. (2002). Time as the Essence for Photo Browsing Through Personal Digital Libraries. Proc. Joint Conf. on Digital Libraries, pp. 326-335.

Japan Electronic Industry Development Association (1998). Digital Still Camera Image File Format Standard. JEIDA-49-1998. http://it.jeita.or.jp/jhistory/document/standard/exif_eng/jeida49eng.htm

Kang, H. and Shneiderman, B. (2000). Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder. Proc. IEEE Intl. Conf. on Multimedia and Expo, pp. 1539-1542.

Kuchinsky, A., Pering, C., Creech, M.L., Freeze, D., Serra, B., and Gwizdka, J. (1999). FotoFile: A Consumer Multimedia Organization and Retrieval System. Proc. of the CHI 99 Conf. on Human Factors in Computing Systems, pp. 496-503.

Loui, A. and Savakis, A. (2000). Automatic Image Event Segmentation and Quality Screening for Albuming Applications. Proc. IEEE Intl. Conf. on Multimedia and Expo, pp. 1125-1128.

Platt, J., Czerwinski, M., and Field, B. (2002). PhotoTOC: Automatic Clustering for Browsing Personal Photographs. Microsoft Research Technical Report MSR-TR-2002-17.

Schiano, D.J., Chen, C.P., and Isaacs, E. (2002). How Teens Take, View, Share and Store Photos. CSCW Interactive Poster.

Shen, C., Lesh, N.B., Vernier, F., Forlines, C., and Frost, J. (2002). Sharing and Building Digital Group Histories. Proc. ACM Computer-Supported Cooperative Work (CSCW), pp. 324-333.

Shneiderman, B. and Kang, H. (2000). Direct Annotation: A Drag and Drop Strategy for Labeling Photos. Proc. IEEE Intl. Conf. on Information Visualization, pp. 88-95.

Sun, Y., Zhang, H., Zhang, L., Li, M. (2002). MyPhotos — A System for Home Photo Management and Processing. ACM Multimedia Demonstration.