PhotoTOC: Automatic Clustering for Browsing Personal Photographs

John C. Platt, Mary Czerwinski, Brent A. Field
Microsoft Research
1 Microsoft Way
Redmond, WA 98052
{jplatt,marycz,brentfi}@microsoft.com

February 2002

Technical Report
MSR-TR-2002-17

This paper presents Photo Table Of Contents (PhotoTOC), an interface that helps users find digital photographs in their own collection of hundreds or thousands of photographs. PhotoTOC is a browsing user interface that uses an overview+ detail design. The detail view is a temporally ordered list of all of the user's photographs. The overview of the user's collection is automatically generated by an image clustering algorithm, which clusters on the creation time and the color of the photographs. PhotoTOC was developed by design iteration on an earlier clustering user interface: AutoAlbum. PhotoTOC was tested on users' own photographs against three other browsers: a hierarchical folder browser (with image thumbnails and the user's own folder structure), a flat detail view with no automatically generated overview, and AutoAlbum. Searching for images with PhotoTOC was subjectively rated easier than all of the other browsers and PhotoTOC's task performance was not slower than any other browser. This result shows that an automatic organization of personal photographs is effective: it requires no organization effort by the user and yet facilitates efficient and satisfying search.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

Millions of consumers have recently bought digital cameras, and millions more are expected to buy them over the next several years. Taking photographs with a digital camera is so convenient and low cost that it is easy for a user to generate more than 1000 photographs per year. This flood of photographs presents a user interface challenge: how can a user find photographs in his or her collection?

Previous work in image browsing, search, and management has concentrated on solving the problem of a user interacting with a large, impersonal, possibly annotated image database. Unfortunately, many of the lessons learned from that problem may not carry over to searching through one's own personal photographs. Unlike interacting with impersonal databases, users have very good memories about photographs within their personal collection. These memories are not only visual but emotional as well. Also, users are often reluctant to spend effort annotating their own images: the images will often be stored in a small, shallow hierarchy of folders on a computer. Users can therefore potentially spend large amounts of effort with standard browsing tools searching through disorganized collections for their photographs. These issues have not been studied by previous work, because user studies have not been performed on users' own photographs and folder structures.

We propose that users should interact with their personal photographs through the use of an image browser which automatically organizes the user's images. To this end, we present Photo Table Of Contents (PhotoTOC), a browser for personal digital photographs that uses a clustering algorithm to automatically generate a table of contents of a user's personal photograph collection. The clustering algorithm segments the stream of photographs into events by analyzing both the creation time of the photographs and their color histograms. PhotoTOC then automatically chooses one representative image per cluster to place into a table of contents. This table of contents is presented in an overview+detail [16] user interface.

1.1 Outline of Paper

In section 2, this paper describes AutoAlbum: an overview+detail image browser. Section 3 defines a task that we use to test and refine AutoAlbum. Section 4 describes a pilot study run on AutoAlbum versus other alternative image browsers. Based on lessons learned in that pilot study, we modified both the representative photograph selection algorithm and the user interface design to produce a new system: PhotoTOC, which is described in section 5. We present the algorithmic details of PhotoTOC in section 6. Finally, we describe the main user study in section 7, which compares PhotoTOC to other browsers on searches through users' own photographs and folder structure. In that study, PhotoTOC did not sacrifice performance compared to other browsers. In addition, PhotoTOC was rated by users as the most efficient browser. Thus, PhotoTOC is the first automatically organized media browser that has scored reliably higher in subjective satisfaction than browsing with a user's own folder structure.

1.2 Related Work

There have been several image browsers proposed in the literature. In Similarity Pyramids [2] and the work of Rodden [20], photographs are organized and clustered according to their color. PhotoTOC uses metadata provided by digital cameras to provide a simpler, more intuitive, time-based user interface. In other work [3, 8, 9, 24], the browsing interfaces work by strongly encouraging users to annotate their images. PhotoTOC uses a pure browsing solution to minimize a user's organizational effort. In PicHunter [5], instead of a fixed organization, a dynamic organization is created by having a user select one photograph out of four that is the most similar to the desired photograph. Four new photographs are then shown at each subsequent iteration, which requires many selection iterations to find the desired photograph. In PhotoTOC, we rely on the fact that users can visually scan hundreds of thumbnails with ease. This leads to a substantial reduction in the number of selection iterations. PhotoMesa [1] uses a zoomable user interface (ZUI) to browse personal photographs. The clustering algorithms of PhotoTOC may be combined with a ZUI to produce a very effective browser. Loui, et al. [12, 11, 23] have proposed a combination of time and color clustering in order to separate events in a photograph browser. Note that none of the previous image browsers were studied with users' own photographs, nor were they tested against browsing users' own folder structures.

Automatic table of content generation has previously been proposed for media types other than photographs. Hypertext is an example of a media type that is amenable to automatic table of content generation [13]. Video has been temporally segmented, both for shot detection [26] and for scene detection [22, 25]. This temporal segmentation has been used to create a video table of contents [22]. Like these other works, PhotoTOC uses a temporally ordered set of photographs to generate a table of contents. However, photographs may be harder to segment than video, since they do not have motion cues to help segmentation.

2 AutoAlbum User Interface

The AutoAlbum user interface was inspired by the fact that users organize their physical photographs into physical albums by topic or event. Rodden has found that the most requested feature for photograph organization is the ability to automatically sort photographs into albums [19].

The AutoAlbum interface is shown in Figure 1. The interface consists of two parts: the overview pane on the left and the detail pane on the right. Both panes contain thumbnails of images. The overview pane also contains calendar hints: the thumbnails are grouped and labeled by month and year. Each thumbnail in the overview pane corresponds to a cluster of photographs, while each thumbnail in the detail pane corresponds to an individual photograph. When the user clicks on a thumbnail in the overview pane, all of the photographs in the corresponding cluster are shown in the detail pane. When the user clicks on a thumbnail on the detail pane, a full-sized version of the photograph is shown in a new window.

The AutoAlbum user interface can be used with manual assignments of pho-

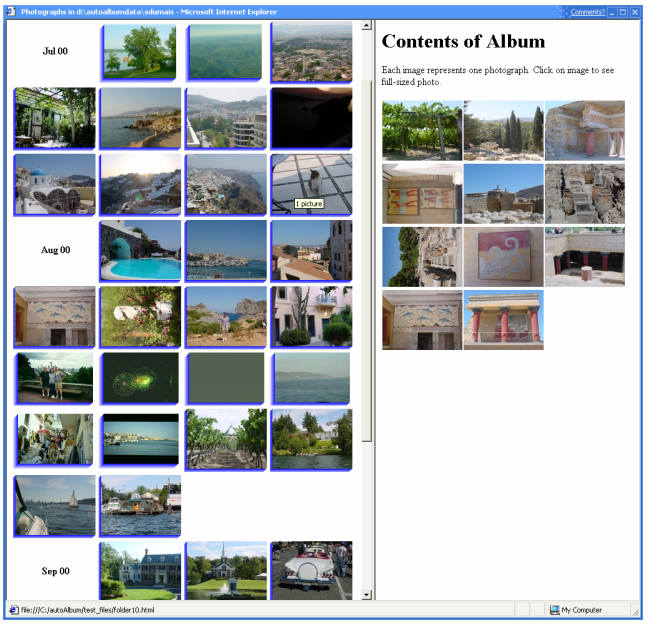

Figure 1: A screen shot of the AutoAlbum user interface

tographs to clusters. However, users will probably not want to assign all of their photographs, due to the length of time it would take. Therefore, AutoAlbum also contains an automatic photograph clustering algorithm that assigns photographs to albums with no user intervention. The clustering algorithm is described in section 6. The clustering algorithm is described later in the paper. If the automatically generated albums do not satisfy the user, it is possible to modify the AutoAlbum user interface to allow the user to re-arrange the albums. We have not yet prototyped a modifiable AutoAlbum UI.

In addition to an automatic clustering algorithm, AutoAlbum chooses one photograph for display that will be mnemonic for an entire cluster, by finding a representative photograph that is similar to each cluster. This representative photograph algorithm is described in section 6.5.

In order to show that AutoAlbum is an effective user interface, we need to define a task to measure user performance and satisfaction. This task is presented in section 3, and a pilot study with this task is presented in section 4.

3 Task Definition

When a user searches or browses for a digital photograph, they have an end goal in mind. For example, it could be showing the photograph to a friend, attaching it to an e-mail, or placing it on a web page. Under many of these scenarios, the user is searching for a particular photograph that has some significance. The user has a mental image

of the desired photograph and searches his or her collection until a photograph that matches the mental image is found.

We could ask a user to repeatedly think of a photograph in his or her collection, and then find it. However, this could introduce uncontrolled variability into task difficulty. For example, the user may select the next photograph based on how easy it is to find that photograph in that particular browser.

In order to achieve better experimental control on task difficulty, we select a randomly chosen photograph from the user's own image set. This photograph is presented to the participant as a target for search. Showing this photograph emulates the mental image that the user has when searching for a desired photograph. In debriefing sessions with our study participants, we were told that this task fairly represents the actual task of trying to find a picture some period of time after storing it in a digital collection of images.

4 Pilot Study

Using the task defined in section 3, we performed a pilot usability study to examine two different versions of AutoAlbum, and to see if the user interface design could be improved. We used four conditions in the pilot study, each corresponding to a different photo browser. The browsers were: (1) a traditional folder browser with thumbnails for each image ("Folders"); (2) a thumbnail browser that simply showed all of the pictures in a flat, scrollable list, ordered by creation time ("LightBox", as shown in Figure 2); and (3 and 4) two different versions of the AutoAlbum UI ("AutoAlbum1" and "AutoAlbum2"). All of the browsers operated on participants' own pictures and folder organization.

The Folders browser was a standard hierarchical folder browser. Each folder was represented with an icon of a folder, four small thumbnails, plus a folder label; rather than with a representative thumbnail without a label, as in the AutoAlbum browsers. The contents of a folder are shown as thumbnails. A user can double-click on a folder to show its contents, and click on an "up" button to go up in the folder hierarchy. A folder tree view was also available.

The LightBox and AutoAlbum browsers provided calendar hints (month and year). The clustering algorithm used in AutoAlbum1 ignored the user's own folder structure, while the clustering algorithm in AutoAlbum2 used the user's own folder structure to assist in clustering. The details of the AutoAlbum1 clustering algorithm are described in section 6.

The Folders browser was included to test AutoAlbum versus the users' own folder organization using a traditional existing tree hierarchy viewer. The LightBox browser was included to test AutoAlbum versus a flat sorted organization, with no clustering. Two different AutoAlbum conditions were included to try and tune the AutoAlbum algorithm.

We gathered six Microsoft Research participants with a broad range of experience with photography in general, but all with experience in managing their digital pictures. All six participants had collected between 146 and 933 photographs.

The task for each browser consisted of 10 search trials (plus two initial practice

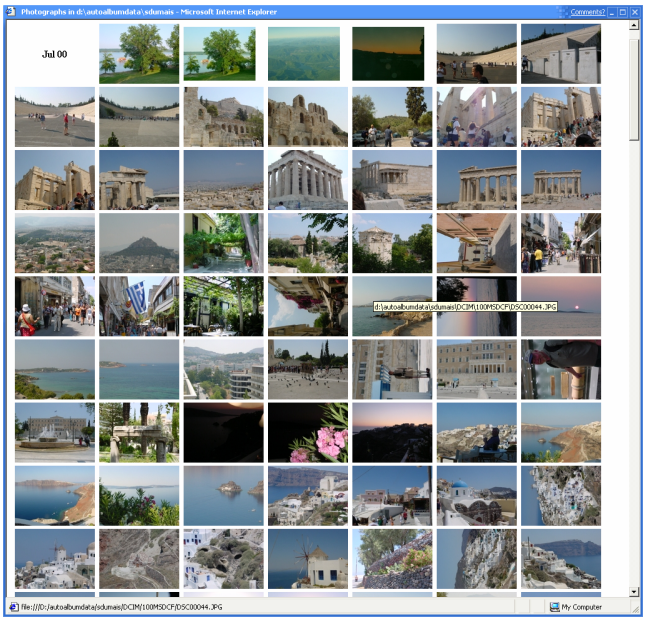

Figure 2: A screen shot of the LightBox user interface

trials). The experimenter ensured that the participant did indeed find the correct photo. The four browsers were presented in random order of usage to each participant. When the participant locates the target image in the browser, he or she is instructed to press a "Finished" button in the experimental program window, which is displayed to the upper left side of the browser's window. This allows us to collect timing information across the various image browsers for comparison purposes.

The user's display was a CRT set to 1280 by 1024 resolution: the browser occupied a 1024 by 1024 window, while the search target occupied a 256 by 256 image in the upper left hand corner of the screen. The maximum dimension of the image thumbnails for all browsers except Folders was 128. In the Folders browser, folder contents were shown as image thumbnails plus a filename for each image. The thumbnail plus filename filled a region of 144 by 96 pixels, which was comparable to the screen area per thumbnail of the other browsers.

All thumbnails are computed only once and then cached, so that system response differences across the browsers were minimized.

4.1 Results of the Pilot Study

For the pilot study, the mean search completion times for the four browser conditions were 36.9s for the Folders browser, 23.1s for the Lightbox browser, 23.4s for AutoAlbum1, and 24.8s for AutoAlbum2. Histograms of the raw search time data indicated that the underlying distribution for completion time was strongly positively skewed.

Logarithmically transformed completion times were distributed approximately normally. Because of this, statistical analyses were conducted on the log of each task's completion time. A Browser X Trial Repeated Measures (RM) ANOVA was performed on the log search time resulting in no significant differences across the four browser styles. This lack of significant difference is not surprising, considering the limited number of participants in the pilot study.

There was a significant linear correlation between the log of the number of pictures in a participant's database and the log of the completion time ($R^2 = 0.135$, $F(1, 238) = 37.10$, $p < 0.001$). The slope of this linear correlation is 0.588 ± 0.106, which shows that browsing is sub-linear with respect to the size of the database. There were no significant differences between browsers in the slope of the linear correlation. A reliable sub-linear relationship between data set size and task completion time would imply that browsing is scalable to image search on larger personal collections.

A subjective rating questionnaire was given to the participants after using each browser. In some responses to the questionnaire, subjects expressed frustration with the difficulty of the AutoAlbum interfaces. By observing user behavior, we noted that users seemed to understand the semantics of each cluster (i.e., they would recognize a tourist attraction in Greece, or photos of a particular party) and they would remember approximately when the search target was created. However, subjects occasionally had a poor memory associating individual photographs with exact events. In the LightBox browser, users would use the scroll bar to quickly move to the approximately correct position in the ordered list, and then perform a brief linear search looking for the exact photograph. The AutoAlbum interface did not allow for the easy linear search, instead it forced users to select numerous overview thumbnails to see the individual photographs associated with each cluster, leading to user frustration.

Based on the pilot study, we gained the design insight that AutoAlbum and LightBox are complementary: AutoAlbum allows users to quickly move to the approximate correct location, and LightBox can be used for short linear searches that find the exact photograph.

5 PhotoTOC: A New, Improved AutoAlbum

After the pilot study, we created a new image browser that combined the best features of AutoAlbum and LightBox. We call this new browser "PhotoTOC," for Photo Table of Contents. PhotoTOC is shown in figure 3. PhotoTOC is still an overview+detail interface. The selection of the overview photographs is still performed via the AutoAlbum1 clustering algorithm, described in the next section. However, the detail interface is now simply the LightBox interface from the pilot study. That is, the detail pane contains an array of all of the user's photographs, sorted by creation date and shown as thumbnails. Clicking on a detail thumbnail still shows the full-sized image. Clicking on an overview thumbnail from the left pane now scrolls the detail pane to show the corresponding cluster within the entire list, with the first thumbnail of the cluster at the top of the pane. The thumbnail that was selected in the overview pane is highlighted in red in the detail pane, to orient the user and give feedback for the overview selection action. The user is free to use the overview pane to "power scroll" the detail pane, or

simply scroll the detail pane and ignore the overview pane. We correctly anticipated that allowing the user the freedom of either using overview selection or detail scrolling would increase user satisfaction.

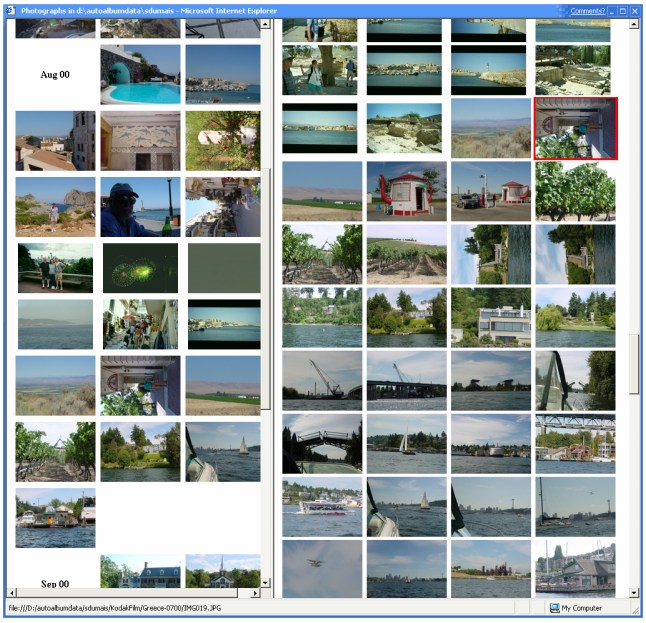

Figure 3: A screen shot of the PhotoTOC user interface

A demo of the PhotoTOC user interface is available on the web at http://research.microsoft.com/∼jplatt/autoAlbum/ex2.html

6 Algorithms: Clustering and Representative Photographs in PhotoTOC

PhotoTOC attempts to identify events in a user's collection. Identifying events from pure image information is very difficult. However, digital cameras and computers provide extra information that allows automatic event identification. Almost all digital cameras time stamp each photograph when the image is created. These time stamps are typically stored in the EXIF metadata format [6].

Unfortunately, EXIF time stamps are sometimes incorrect due to an improperly set camera clock. Also, some users have scanned their photographs, which does not preserve photo creation time. The file creation date, file modification date, or filename can still be used to order the photographs, although extracting events is more difficult when the true creation time is missing. For cases where the creation time is missing or corrupt, PhotoTOC uses the order of the photographs plus the color information in the photographs to identify events. Because digital photographs can be ordered in time

even when the exact creation time is unavailable, adding color information is sufficient to identify events. If two temporally adjacent photographs have similar colors, they are most likely from the same event, while two photographs that have similar color but are temporally far apart are very unlikely to be of the same event or subject.

Thus, there are two different clustering algorithms that can be applied to a user's collection of photographs. One is time-based clustering, where the creation time is used to cluster the photographs. The other is content-based clustering, where the creation time is used only to order the photographs, and color information is then used to cluster. The time-based clustering is preferred when the data is reliable: the content-based clustering is used as a backup algorithm. In PhotoTOC, the two clustering algorithms are combined to ensure a sensible clustering even with missing or corrupted time data.

6.1 Time-Based Clustering

The goal of time-based clustering is to detect noticeable gaps in the creation time. A cluster is then defined as those photographs falling between two noticeable gaps. These gaps are assumed to correspond to a change in event. The time gap detection is adaptive: it compares a gap to a local average of temporally nearby gaps. A gap is then considered a change of event when it is much longer than the local gap average. Time gaps have a very wide dynamic range. In order to handle this dynamic range, the gap detection algorithm operates on logarithmically transformed gap times.

More specifically, time-based clustering first sorts the photographs by creation time. Then, if g_i is the time difference between picture i and picture $i+1$ in the sorted list, g_N is considered a gap between events if it is much longer than a local log gap average:

$$\log(g_N) \geq K + \frac{1}{2d+1} \sum_{i=-d}^{d} \log(g_{N+i}), \tag{1}$$

where K is a suitable threshold (chosen empirically to be $K = \log(17)$), and d is a window size (chosen to be $d = 10$). If $N+i$ refers to a photograph beyond the ends of the collection, the term is ignored, and the denominator $2d+1$ is decremented for every ignored term, to keep the average normalized.

The algorithm that adaptively determines the gap between events is new to this paper: previous versions of time-based clustering [17] used a fixed threshold. Empirically, the adaptive gap algorithm identifies events better than the fixed threshold. The adaptive gap algorithm was used in the pilot study and for both AutoAlbum and PhotoTOC in the main experiment described in the next section.

6.2 Content-Based Clustering

Content-based clustering is based on a probabilistic generative model of photographs. The probabilistic model is a left-right Hidden Markov Model (HMM). The assumption behind the model is that the colors of pixels of an image are drawn independently from a multinomial (a histogram model). This color histogram model is piecewise stationary: a histogram model will generate one or more images (in creation time order) and then the HMM will transition to a new histogram model. Each state in the

HMM thus corresponds to a single event, which has its own color histogram model. More mathematical details can be found in [17]. Unfortunately, the standard Baum-Welch algorithm for fitting an HMM to data gets stuck in poor local minima for this problem and fails to find sensible albums. Therefore, PhotoTOC uses best-first model merging [15], which avoids local minima by stepwise merging of adjacent clusters (see figure 4).

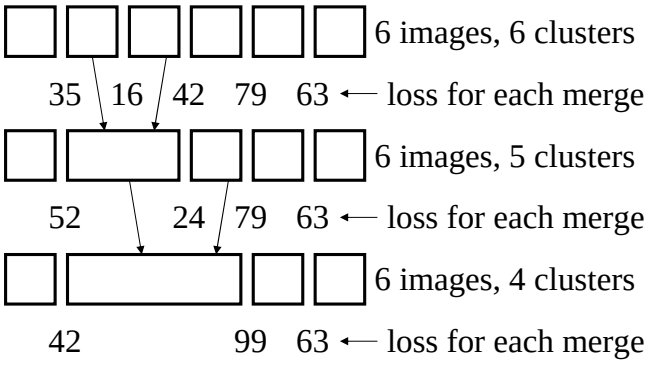

Figure 4: Best-first model merging

As applied to PhotoTOC, best-first model merging starts with every image in its own cluster and having its own color histogram model. The log likelihood of every image to be generated by its own model can be computed (see section 6.3). At every step of best-first model merging, all possible pairs of adjacent clusters are considered for merging. When two clusters are merged, their histogram models are combined by summing the bins of each histogram and then normalizing them. For every possible merge, best-first model merging computes the new log likelihood of all of the pixels in the merged cluster as generated by the merged model. This log likelihood is less than the sum of the log likelihoods of the two original clusters because the merged model is more general and cannot fit the data as well as two individual models. Specifically, if L_X is the log likelihood of all of the pixels assigned to a cluster X given an associated model of X, and if clusters X and Y are being merged to form cluster Z, then the change in log likelihood associated with the possible merge is

$$\Delta L = L_Z - L_X - L_Y. \tag{2}$$

This log likelihood change is the Jensen-Shannon divergence between X and Y [10]. This divergence is measured for all possible adjacent pairwise merges; the merge with the least divergence is chosen and executed. This merging of clusters and models continues until a desired number of clusters is reached. In the PhotoTOC user interface, as tested below, the number of desired clusters is 1/12 the number of photographs (rounded to the nearest integer). This choice of cluster number yields a "zoom factor" of the detail view of approximately 12. The "zoom factor" of 12 was chosen based on a typical "zoom factor" that was generated by a time-based clustering on a small sample of users. In other words, users generate about 12 photographs per distinct event.

6.3 Histogram Details

PhotoTOC uses a 2D color histogram model for each cluster because luminance changes are assumed to come from lighting conditions and therefore do not necessarily reflect changes in event. This model was chosen after empirically testing several different models: it was the model which gave a content-based clustering that most closely matched the time-based clustering on the same data [17].

The colors are defined in the 1976 CIE $u'v'$ color space [7]. The CIE color space was chosen because a given Euclidean distance in $u'v'$ corresponds to roughly constant percept of color difference. Thus, when the $u'v'$ space is divided into bins, each bin will encompass roughly the same amount of perceived color difference. The $u'v'$ space is divided into 256 square bins (16 bins on a side). The probabilistic model is that a bin is chosen with probability proportional to the stored histogram count and the color of a pixel is the color of the center of the chosen bin. There are 16 bins in u' that span from 0.1612 to 0.2883. There are 16 bins in v' that span from 0.4361 to 0.5361. Colors outside of these spans are clipped to the nearest bin. These colors were chosen to cover roughly 90% of the pixels found in typical digital photographs.

To increase the speed of the clustering, only the histogram statistics for each cluster is kept during best-first model merging. The log likelihood of a pixel that lands in histogram bin i is given by

$$\log\left(X_i / \sum_j X_j\right), \tag{3}$$

where X_i is the histogram count in bin i. This log likelihood arises because the pixels are assumed to be generated independently by a multinomial model. Therefore, the log likelihood L_X of all of the pixels in cluster X is simply the sum of the individual log likelihoods:

$$L_X = \sum_i X_i \log\left(X_i / \sum_j X_j\right). \tag{4}$$

In order to not to generate infinitely negative log likelihoods, the histogram bins can never have zero counts. Therefore, before an image is histogrammed, each histogram is initialized with all bins equal to a small value (10/256), which implies that the histogram estimate will be a *maximum a posteriori* estimate, and that the prior probability for this estimate is a uniform prior over all colors. Then, the image is downsampled by 8. The RGB pixel values are converted to $u'v'$ and the histogram counts are computed.

The histogram is further smoothed with a kernel, in order to make the histogram model insensitive to small changes in color: slight changes to a pixel should alter the histogram model only slightly. For every $u'v'$ pixel value in the downsampled image, a bilinear tent-shaped kernel is placed over the pixel value in $u'v'$ space. This kernel is two bin widths wide in each dimension. The histogram bins are then increased by an amount equal to the value of the kernel at the center of each bin. For example, if a pixel value lies directly on a bin center, that bin is increased by 1, and no other bins are changed. If a pixel value lies in the exact center of four bin centers, all four of those bin centers are increased by 0.25. Thus, the histogram counts are bilinearly interpolated [18].

6.4 Combining Time and Content Clustering

PhotoTOC combines the time-based and content-based clustering because the creation times are not always reliable. A signal of unreliable creation time is that time-based clustering yields large clusters. Therefore, PhotoTOC uses this signal to combine the time-based and content-based kinds of clustering.

First, PhotoTOC extracts the creation time from the EXIF tags of the digital image. If this time is unavailable or considered corrupt (i.e., before Jan 1, 1999), then the file creation time is used. The images are sorted, then the time-based clustering algorithm is applied to the images. If any of the time-based clusters are too large (i.e., more than 23 images), then content-based clustering is applied to each large cluster, which produces a number of smaller clusters. All of the resulting clusters are then displayed in the overview and detail panes. This combination method was used in the pilot study and the main experiment.

After the main experiment described in the next section, we implemented a mathematically more elegant solution to the combination of time and content clustering in PhotoTOC. In the new solution, the pixel model jointly generates both the color and the time of the pixel. Each pixel is assigned its own time, in order to balance the effects of the histogram of each image and the time of each image. If the time were only generated on a per-image basis, the time data would be overwhelmed by the color data, since there are many more pixels than images.

In this joint model, the color model is still a histogram, while the time model is a Gaussian. Thus, the likelihood of all the pixels in a cluster X becomes

$$L_X = \sum_i X_i \log \left(X_i / \sum_j X_j \right) - \frac{N_X}{2} \log(2\pi e \sigma^2), \tag{5}$$

where N_X is the number of pixels in the cluster X and σ^2 is the variance of the time Gaussian for that cluster. Clustering is then performed via best-first model merging, as above. When two clusters are merged, the two time Gaussians are replaced with the best single time Gaussian: the sufficient statistics of the two Gaussians are summed together. When the clustering is initialized, each cluster starts with a time mean equal to the creation time of the single image, and a time standard deviation of one hour.

Qualitatively, the more elegant solution yields clusters very similar to the simple combination of time and clustering. It is not clear yet if it actually improves user performance and satisfaction, but we expect it is more robust.

6.5 Choosing a Representative Photograph

For the pilot study, the representative photograph for a cluster was chosen to be the photograph in the middle of the cluster when sorted by creation time [17]. However, the results of the pilot user study showed that, for some clusters, the middle photograph was unrepresentative of the overall cluster. For example, for one subject, one of the middle photographs was an out-of-focus photograph of a ceiling. When a poor photograph is chosen in the overview pane, it will not be mnemonic for the user, and the user will never find the photographs in that cluster.

Therefore, after the pilot study, we updated the algorithm to choose one photograph from a cluster that is the most representative of that cluster. The photograph is chosen by measuring the Kullback-Leibler (KL) divergence [4] between the histogram of every photograph in the cluster and the averaged histogram over all photographs in the cluster. More specifically, let P_{ij} be the normalized histogram count in bin i for picture j. Let A_i be the average histogram count in bin i over all images in the cluster. Then, the picture j is chosen to be representative when it maximizes

$$\sum_i P_{ij} \log \left(P_{ij} / A_i \right) . \tag{6}$$

If images of an event have a number of uniquely colored regions, then the image with the highest number of those regions will tend to get selected by the KL divergence metric. Poor quality images very rarely get selected by the KL divergence criteria: when the new algorithm was applied to the data from the pilot study, all of the overview photographs were of good quality.

7 Main Experiment

The main experiment was designed to test PhotoTOC versus the more standard browsers (Folders and LightBox), and versus its previous version (AutoAlbum). In addition, both AutoAlbum and PhotoTOC were updated to use the more sophisticated representative photo selection algorithm, as described above.

As in the pilot study, the Folders browser was included to test whether automatic organization was the same or better than the user's organization using existing tools. The LightBox browser was included to test whether organizing by clustering (versus merely sorting) improved user performance and satisfaction. The AutoAlbum browser was included to test whether the design iteration improved the browser.

7.1 Experimental Details

7.1.1 Participants.

There were 8 participants (1 female) with an average age of 36.9 years. Two participants were from Microsoft Research, the rest were not Microsoft employees. Each provided a personal set of digital pictures (their libraries ranged from 345 pictures to 1298 pictures, average size was 850 pictures). All users were at least intermediate Windows users as assessed by a background screening questionnaire. Participants had a range of experience with photography: skill levels ranged from casual photographers who simply took photographs of their vacations all the way up to a professional wedding photographer. No participants from the pilot study participated in this second study.

7.1.2 Apparatus

The picture browsers were executed on a high-end Dell Precision P610 computer running a beta version of Windows XP. A NEC MultiSync FE1250 21" monitor was used.

The display resolution, browser window size, search target size, and thumbnail sizes for all browsers were the same as in the pilot study. The participants were observed behind a one-way mirror and through a video split from their display to a monitor in the observation room.

7.1.3 Methods

As in the pilot study, the four browsers were presented to each participant in random order. For each browser, there were two practice trials and ten measured trials, for a total of 320 data points over all subjects. After each browser was used, a satisfaction questionnaire about that browser was presented to the participant.

7.2 Results

7.2.1 Completion Times

Two outliers were identified from the 320 data points. These were the only points more than 5 standard deviations from the mean (in log space), and both appeared to be unrealistically fast responses. These two values were replaced with the mean response rate (in log space) for a given browser and trial.

As in the pilot study, the completion times were transformed logarithmically before statistical analysis. A Browser X Trial RM ANOVA was performed on the log task completion time data. The type of browser contributed significantly to the overall variance ($F(3, 21) = 3.191$, $p < 0.047$), resulting in a reliable main effect. The Light Box condition had a mean completion time of 28.4s, PhotoTOC had 37.3s, AutoAlbum had 45.1s, and the Folders browser had 58.7s, as shown in Figure 5. The respective medians were 17.43, 18.64, 20.93, and 24.66s. Trial number or repetition did not contribute significantly to the overall variance ($F(9, 63) = 1.24$, $p = 0.286$). This coupled with a non significant one-way RM ANOVA for task order ($F(3, 21) = 1.117$, $p = 0.364$) indicated that the there were no significant learning effects.

Pair-wise *post hoc* comparisons across the browser conditions showed that there was no significant difference in task completion time between any two browsers.

As with the pilot study, there was a significant linear correlation between the log of the number of photographs in a participant's database and the log of the task completion time. This accounted for 10.9% of the variance ($R^2 = 0.109$, $F(1, 318) = 38.85$, $p < 0.001$). For the main experiment, the slope of the correlation is 0.883 ± 0.142, which is not significantly different from a linear time relationship between task completion time and size of database. No significant differences in the linear correlation were found between browsers. Unlike the pilot study, the main experiment confirms the linear time relationship in image browsing reported by [3], in this case on somewhat larger data sets. However, it must be noted that, unlike [3], each participant browsed their full personal database of images. Therefore, in the present experiments, participant speed is confounded with image database size. The linear time result should thus be considered preliminary.

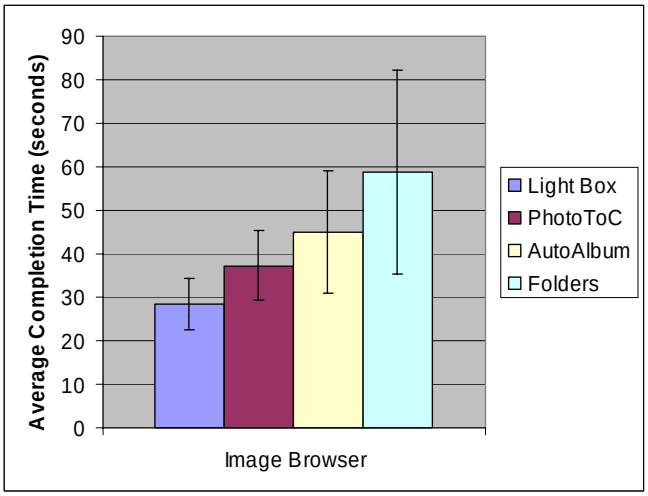

Figure 5: Mean task completion time for all four browsers, with error bars representing ± 1 standard error of the mean

7.2.2 Questionnaire

The satisfaction scores, taken after each condition was completed, are shown in Table 1. These scores showed that PhotoTOC was viewed most favorably on average, followed by Folders, Light Box, and AutoAlbum. A Browser X Question RM ANOVA indicated that variance explained by browser was non-significant ($F(3, 21) = 0.581$, $p = 0.519$). However, there was a significant effect of questionnaire item ($F(6, 42) = 4.77$, $p < .01$), as well as a significant interaction between browser and questionnaire item, ($F(18, 126) = 2.1$, $p < .01$). Planned comparisons for each individual question using the Bonferroni correction for multiple tests revealed several significantly higher ratings for the PhotoTOC browser compared to the other browsers. For example, PhotoTOC scored reliably higher than all other browsers for the questionnaire item, "It is easy to find the photo I am looking for with the image browser." PhotoTOC scored reliably higher than AutoAlbum for the item, "This image browser is efficient." PhotoTOC scored reliably higher than AutoAlbum and LightBox for "If I came back a month from now, I would still be able to find many of these photos with this browser." The Folders browser was rated reliably higher than the AutoAlbum browser on the item, "I was satisfied with how the pictures were organized." No other reliable differences were observed in the questionnaire data.

Overall, the individual questionnaire data indicates that users think that using PhotoTOC to browse photographs is subjectively easier than a folder browser, a detail-only view, and AutoAlbum (the original design). The PhotoTOC versus AutoAlbum results show that good interface design is important for high satisfaction: the two browsers share the same underlying technology, yet have very different satisfaction results.

	Folders	Light-Box	Auto-Album	Photo-ToC
I like this image browser.	2.63	2.88	2.50	3.25
This browser is efficient.	2.63	2.88	2.38	**3.38**
This browser is easy to use.	3.25	3.63	3.50	3.88
This browser feels familiar.	3.88	3.63	3.00	3.00
It is easy to find the photo I am looking for.	2.75	2.75	2.50	**3.75**
A month from now, I would still be able to find these photos.	3.63	3.25	3.25	**4.13**
I was satisfied with how the pictures were organized.	**3.50**	2.75	2.63	2.88
Average	3.18	3.11	2.82	3.46

Table 1: Mean satisfaction scores across participants, using a 5 point Likert Scale with 1 being strongly disagree and 5 being strongly agree (boldface marks significant differences).

8 Discussion and Future Work

Although the completion time data and the overall questionnaire data provide some initial evidence of the superiority of PhotoTOC to some or all of other browsers, only certain individual questionnaire items revealed statistically significant differences between browsers. The lack of a reliable performance advantage is primarily due to the limited number of users in the main study, which limits the statistical power of our comparisons. We can run future image browsing studies with more participants in order to further refine our statistical analyses and conclusions.

Many interesting image search behaviors were identified during the two studies. For example, it was often observed that subjects were quite good at determining the approximate time that a picture was taken. However, sometimes their hypotheses would be wrong (probably based on some cue in the target image itself), and these would lead users down garden path searches that they strongly believed were correct. For example, one participant misrecognized one target as taking place on a different lake, which caused them to look around for the target in the wrong month. When their theories failed, subjects would resort to serial search, effectively scrolling through their entire database either forward or backward. In addition, it was observed that participants would often return to a given category of items multiple times when they held a strong but mistaken belief about the date or event of an image. This multiple return behavior was most noticeable in the Folders browser, where the user descended up and down the folder hierarchy repeatedly. Participants sometimes found this quite frustrating, which confirms research that has shown that searching through hierarchies is problematic, even for fairly shallow hierarchies [14]. It would be interesting to design an image browser that might assist the user by providing alternative interfaces that might break the user out of their incorrect hypotheses. Analogous to [3], it would be interesting to combine content-based image retrieval [21] with image browsing to help users find their photographs even when they are confused about how the photograph fits into its context.

A strong difference in organizational behavior was noted between the professional and high-end consumer photographers and the more casual photographers. Serious photographers, due to their long history of taking pictures and their large databases of images, have built a categorical hierarchy that is well-honed and memorized by the user. Casual photographers had fewer, less well-defined categories. Serious photographers used their folder system very effectively, with minimal incorrect hypotheses, and would most likely reject a software tool that didn't support their rich folder structure. Casual photographers are grateful for any sensible organizational guidance the system provides. Therefore, PhotoTOC can be redesigned to support both categories of users by allowing a switch between a table of contents view and a folder view.

In the main experiment, but not in the pilot study, the search time increased linearly with the size of the collection. However, our study was conducted over a relatively narrow range of picture database sizes, so there may still be a sub-linear browsing time component. In an ideal future study, we would vary the number of pictures per participant to disentangle speed of participant from size of database. A future study could look at completion times across the range of 100 to 10,000 pictures.

After the conclusion of the experiment, the PhotoTOC interface was improved.

First, a visual affordance (a blue drop shadow) was added to each overview thumbnail to indicate that each thumbnail represents more than one photograph. Second, the scrolling behavior upon selecting the overview thumbnail was changed: the detail thumbnail corresponding to the overview thumbnail was scrolled to be in the center of the detail pane. These improvements need to undergo further user study.

9 Conclusions

Users are being overwhelmed by an incoming flood of their own digital photographs. They are starting to demand automatic organization tools: specifically, systems that automatically group photographs into albums or clusters [13]. This paper presents PhotoTOC as an example of an automatic organization tool. PhotoTOC is a system that automatically clusters photographs: it allows a user to browse his or her collection in an overview+detail view.

We compare PhotoTOC to other image browsers by performing user studies on users' own photographs and their own folder organization. These studies allow us to objectively compare traditional folder browsing user interfaces to specialized image browsing user interfaces that utilize various forms of automatic organization.

The main user study showed that automatic organization of images combined with a suitably designed UI is subjectively more satisfying to browse than standard browsing interfaces that can leverage a user's own organization. Automatic organization does not sacrifice browsing performance. This increased user preference was found for collections up to and beyond 1000 photographs. For consumers, higher user preference may be more important than improved search performance. Thus, automatic organization is a practical management technique for personal photographic collections.

The observed differences between AutoAlbum and PhotoTOC in the main study show that simply automatically organizing personal photographs is not enough. AutoAlbum and PhotoTOC were based on the same underlying automatic algorithm, but produced very different satisfaction results due to differing user interface designs. Therefore, it is not adequate to simply design automatic organization algorithms in a vacuum. Any automatic organization algorithm development must be coupled to iterative interface design and user studies in order to be truly useful.

10 Acknowledgments

The authors would like to acknowledge the suggestions and support of our co-workers, including Susan Dumais, Hagai Attias, David Heckerman, Dan Robbins and Ken Hinckley.

References

[1] B. B. Bederson. Quantum treemaps and bubblemaps for a zoomable image browser. In *Proc. User Interface Systems and Technology*, pages 71–80, 2001.

[2] J. Chen, A. Bouman, and J. C. Dalton. Similarity pyramids for browsing and organization of large image databases. In *Proc. SPIE/IS&T Conf. on Human Vision and Electronic Imaging III*, volume 3299, pages 563–575, 1998.

[3] T. T. A. Combs and B. B. Bederson. Does zooming improve image browsing? In *Proc. Digital Library (DL99)*, pages 130–137, 1999.

[4] T. Cover and J. Thomas. *Elements of Information Theory*, chapter 2.3. Wiley-Interscience, 1991.

[5] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. PicHunter: Bayesian relevance feedback for image retrieval. In *Proc. ICPR*, pages 361–369, 1996.

[6] EXIF image format. http://www.pima.net/standards/ it10/PIMA15740/exif.htm.

[7] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, 1990.

[8] H. Kang and B. Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the PhotoFinder. In *Proc. IEEE Intl. Conf. on Multimedia and Expo*, 2000.

[9] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. FotoFile: a consumer multimedia organization and retrieval system. In *Proc. ACM CHI Conf.*, pages 496–503, 1999.

[10] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Info. Theory*, 37(1):145–151, 1991.

[11] A. Loui and A. E. Savakis. Automatic image event segmentation and quality screening for albuming applications. In *ICME 2000*, pages 1125–1128, 2000.

[12] A. Loui and M. Wood. A software system for automatic albuming of consumer pictures. In *Proc. ACM Multimedia*, pages 159–162, 1999.

[13] D. A. Nation, C. Plaisant, G. Marchionini, and A. Komlodi. Visualizing websites using a hierarchical table of contents browser:WebTOC. In *Proc. 3rd Conf. on Human Factors and the Web*, Denver, CO, 1997.

[14] K. L. Norman. *The Psychology of Menu Selection: Designing Cognitive Control and the Human/Computer Interface*. Ablex Publishing, 1991.

[15] S. M. Omohundro. Best-first model merging for dynamic learning and recognition. In *Advances in Neural Information Processing Systems*, volume 4, pages 958–969, 1992.

[16] C. Plaisant, D. A. Carr, and B. Shneiderman. Image browsers: Taxonomy and design guidelines. *IEEE Software*, 12(2):21–32, 1995.

[17] J. C. Platt. AutoAlbum: Clustering digital photographs using probabilistic model merging. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 96–100, 2000.

[18] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*, chapter 3.6. Cambridge University Press, 2nd edition, 1992.

[19] K. Rodden. How do people organise their photographs? In *BCS IRSG 21st Ann. Colloq. on Info. Retrieval Research*, 1999.

[20] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *Proc. ACM CHI 2001*, pages 190–197, 2001.

[21] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, present directions, and open issues. *J. Visual Communications and Image Representation*, 10:39–62, 1999.

[22] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-content for videos. *ACM Multimedia Systems Journal*, 7(5):359–368, 1999.

[23] A. Stent and A. Loui. Using event segmentation to improve indexing of consumer photographs. In *Proc. SIGIR*, pages 59–65, 2001.

[24] L. Wenyin, S. Dumais, Y. Sun, H.-J. Zhang, M. Czerwinski, and B. Field. Semi-automatic image annotation. In *Proc. Interact*, pages 326–333, 2001.

[25] M. Yeung, B.-L. Yeo, and B. Liu. Video browsing using clustering and scene transitions on compressed sequences. In *Proc. Multimedia Computing and Networking*, volume SPIE-2417, pages 399–413, 1995.

[26] H.-J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.