Leveraging Face Recognition Technology to Find and Organize Photos

Andreas Girgensohn, John Adcock, and Lynn Wilcox
FX Palo Alto Laboratory
3400 Hillview Avenue
Palo Alto, CA 94304, USA
{andreasg, adcock, wilcox}@fxpal.com

ABSTRACT

With digital still cameras, users can easily collect thousands of photos. We have created a photo management application with the goal of making photo organization and browsing simple and quick, even for very large collections. A particular concern is the management of photos depicting people. We present a semi-automatic approach designed to facilitate the task of labeling photos with people that opportunistically takes advantage of the strengths of current state-of-the-art technology in face detection and recognition. In particular, an accurate face detector is used to automatically extract faces from photos while the less accurate face recognizer is used not to classify the detected faces, but to sort faces by their similarity to a chosen model. The sorted faces are presented as candidates within a user interface designed for quick and easy face labeling. We present results of a simulation of the usage model that demonstrate the improved ease that is achieved by our method.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—*Evaluation/methodology.* H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Graphical user interfaces (GUI)*. H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Retrieval models.*

General Terms

Algorithms, Design, Human Factors.

Keywords

Digital photo collections, face detection and recognition, user interface design, home users.

1. INTRODUCTION

As digital cameras become increasingly ubiquitous, personal digital photo collections are growing both in number and size. These collections may easily grow by hundreds of photos a year, quickly reaching into the thousands of photos. These large collections require effective interfaces that facilitate browsing, manipulation, and sharing. There are many commercial and research applications that support users in digital photo organization but the bulk of the user's work remains manual [1, 2, 3, 4, 5, 6, 10, 11]. A major challenge is to make photo organization simple and quick while enabling users to enjoy, present, and share their voluminous collections.

Towards this end, we have opportunistically applied automation in the development of our photo organizer [7]. We aim to apply automation where it is robust and to augment the application with user interfaces where it is not robust, thereby leveraging what automatic analysis can provide in as unobtrusive a way as possible. As a consequence, the overall user experience is enhanced without adding unnecessary complexity or requiring excessive user intervention to compensate for imperfect automation technologies.

One area where we have applied this approach is in the management of photos of people. Such photos are prevalent in photo collections and organizing them is therefore a common task. In particular, users frequently want to retrieve photos depicting a particular person or combination of people. Despite our previous efforts to simplify photo labeling with an easy to use drag-and-drop interface, labeling each photo by hand with the people pictured in it remains a tedious task. Meanwhile, current automatic face recognition technology is not sufficiently robust to consistently identify people in widely varying lighting conditions and poses.

In this paper, we describe an extension to our application that can identify photos with faces, extract the faces, and use the faces themselves as a user interface resource to label photos and the faces within them. Traditionally, photo applications have not made extensive use of face recognition because of the insufficient reliability of the technology. Instead of an exclusive reliance on automatic face recognition, we combine user interface techniques with an accurate face detector and a semi-accurate face recognizer. The face detector extracts faces from photos. As the user assigns the extracted faces to a person, the face recognizer forms a model for the person. This model is in turn used by the face recognizer to determine similar faces in the unassigned set of extracted faces. Faces can be sorted by similarity to previously labeled faces to quickly identify likely candidates that the user can then evaluate and collectively label. The face thumbnails act as UI proxies for assigning person labels to photos.

In the next section, we review several approaches for labeling faces in photos and compare them to our approach. We then describe our photo organization application and show how we use face detection and recognition technologies to facilitate the labeling of faces in photos. Then, we examine how well our usage model can be expected to work with a series of simulations. The empirical data shows how close correct faces are to the top of the similarity list and how many faces can be labeled without scrolling the light table window. 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. Some of those applications use the presence or absence of people in photos as a

MIR'04, October 15-16, 2004, New York, New York, USA.

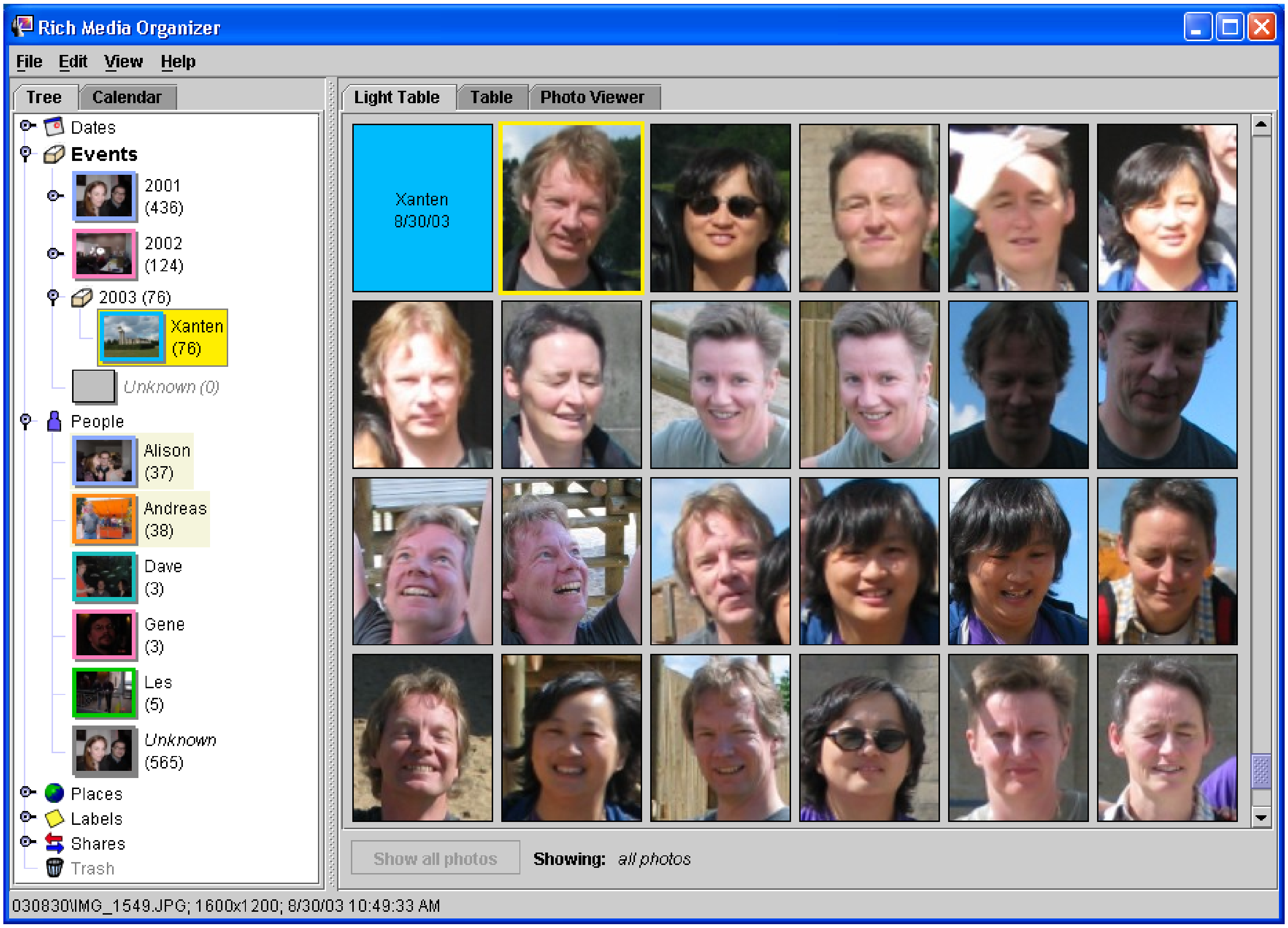


Figure 1: The Photo Application in face view. Category tree on the left, scrollable light table on the right.

sorting criterion. The PhotoFinder system [10] provides stickies with people's names that can be attached to points in photos. Photos can be retrieved with boolean queries that can refer to people's names among many other attributes. However, PhotoFinder relies entirely on the user identifying and manually labeling each person by dragging a label onto the appropriate photos [14]. An earlier version of our photo application [7] uses a similar method but instead of drag-and-dropping labels onto a photo, users drag-and-drop one or more photos onto a labeled node in a tree-view.

The FotoFile system [11], like our system, uses a semi-automatic approach that combines automatic face recognition and user intervention. It automatically assigns a name to an automatically detected face and then asks the user to confirm the assignment. This interaction style requires user confirmation of each automatically labeled face and is not well suited for labeling hundreds of faces.

The MediaBrowser [6] uses automatically detected metadata to help users organize their photos. One of the detected features is the presence of faces that can be used by the user to find or show only photos with faces. The authors describe the ability to recognize specific faces and to automatically attach names as desirable features in future versions.

Lim et al. [12] describe a system for organizing and retrieving photos from a collection of home photos. They detect different visual keywords, among them face and skin, and use those keywords for grouping photos into events and for retrieving them.

Zhang et al. [16] have built a system that automatically determines the most likely identities for each detected face in a photo. Their system presents a list of candidate names for each face and lets the user choose among them. While this is in much the same spirit as our semi-automatic approach, it still requires a user interaction for every face rather than our approach of simultaneously assigning groups of faces. Another paper by the same group to appear later this year presents a system where the user assigns thumbnails of photos to different people. The system automatically detects faces and finds a solution for propagating names from the photos to the faces within them. While this approach allows bulk-assignments similar to ours, it requires the user to recognize people in thumbnails, a sometimes difficult task. This approach also does not support users by sorting or grouping photos by face similarity.

Suh et al. [15] studied the effectiveness of automatically cropping thumbnails in manual image-matching tasks. Participants in a study were asked to find matching images in a collection of thumbnails. The thumbnails were created by different means including scaling down, cropping by saliency map, and cropping by detected faces. Cropping by detected faces was most effective in tasks where participants had to find a photo of a particular person.

3. OUR PHOTO ORGANIZER

Our application [7] shows the entire collection of photos in a vertically scrollable light table (see the right hand pane in Figure 1). A tree widget (left pane in Figure 1) displays the photo categories with a top-level node for each of several different category types.

Tree nodes are drag-and-drop targets for selections of photos. Categories can be assigned by dragging a selection of photos onto a category node in the tree view, thereby applying that category to each photo in the selection. Each sub-tree contains a node labeled "Unknown". Dropping a selection onto this node removes all existing tags of that category type from the selected photos.

The currently active top-level sub-tree determines the sort order of the photos in the light table. Based on the current sort order, colored markers delimit the start of each new category (such as days or events) in the light table. (In Figure 1, the light table is sorted by events.) A photo may have any number of tags or labels (except for dates and events where a photo can only have one label within each of these categories) and in cases where a photo has multiple labels within a category (e.g., a photo labeled with several people), that photo is shown multiple times in the light table, appearing in the section corresponding to each of the relevant tags.

The tree view can also be used to quickly navigate the light table to the photos tagged with a particular value. Left-clicking on a tree node scrolls the light table to the first photo with the selected value or tag and makes that photo the current selection. Additional operations on tree nodes are available by right-clicking on a tree node and selecting an operation from a pop-up menu. For example, the user can elect to display in the light table only the photos that are below a tree node by right-clicking on the node and choosing "Show only these photos" from the pop-up menu and thus filtering the light table.

While filtering the light table using a tree node that represents one category, the tree and light table still function with the other category types. For example, the user can show only photos from 1999 by right-clicking on the "1999" tree node and choosing "Show only these photos". By doing so, the light table will display only the photos assigned to the node's category, i.e., all photos taken in 1999. The shown subset of photos can be sorted and navigated by a different label type, for example, by left-clicking on the tree node representing a particular person to find all photos of that person taken in 1999. The "Show all photos" button below the light table is used to revert back to the unfiltered mode where all photos are displayed.

4. USE OF FACES

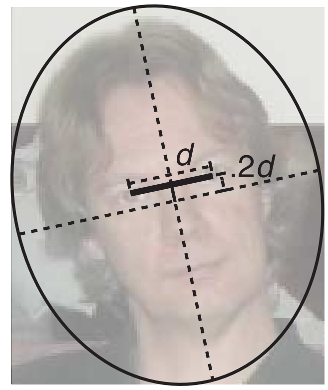
When new photos are imported into the application, a face detector automatically runs in the background and thumbnails are made for the detected faces. The light table can be switched into a view where thumbnails of the detected faces are shown in place of the full-size photos (see Figure 1). The face thumbnails act as proxies for the photos and can be viewed and organized just like photos. Faces can be assigned to a person via a drag-and-drop operation onto a node in the person category as described above. When a face is labeled, the face is added to a face recognition model representing the faces with that label.

Faces in the light table can be sorted by decreasing similarity to a face recognition model to help the user identify unlabeled faces of the same person. In this mode, faces already assigned to another person are not shown and the user can quickly scan and decide which of the unlabeled faces near the top of this sorted list truly belong to the person in question, and manually label them. In the remaining subsections, we will describe how a) faces are detected, b) faces are displayed in the light table, c) faces are labeled, and d) similar unlabeled faces are recognized.

4.1 Detecting Faces

The face detector [9] used in our photo application is highly accurate for photos with faces where both eyes and the nose are visible. The recall rates are between 80% and 90% for a precision up to 100%[1]. However, it is unable to detect faces where only one eye is visible. Unlike other face detectors [13] that return the bounding box of the face, our detector returns instead the actual eye positions such that the orientation of the face can be determined.

(a) tightly cropped and rotated face



(b) moderate crop

Figure 2: Determining the face crop from the eye positions. The crop is chosen to bound the ellipse determined by the eye coordinates.

As new photos are imported into the photo application, the face detector automatically runs in the background. The amount of time it takes to detect faces depends on the range of sizes and rotations for which it is configured to search. With the settings we use, face detection takes roughly one second per photo.

4.2 Displaying Faces

In the face-view mode, the application displays in the light table the cropped face images in place of the full photo thumbnails (see Figure 1). In the first version of our interface, we chose to display the images extracted and used by the face recognizer described below. A normalizing rotation is applied to those faces such that the line between the eyes is horizontal and the image is cropped tightly to the face (see Figure 2a). Our users enjoyed browsing the faces in their photo collections but found the aesthetic effect of the tightly cropped and rotated faces to be less than ideal, referring to them as "mug shots."

To improve the look of the face thumbnails, we designed an alternative presentation that does not rotate the image and uses a more generous cropping, more reminiscent of typical portrait photos. Faces are cropped such that there is a sufficiently large margin around them to avoid a boxed-in look. Figure 2b depicts the alignment of an ellipse[2] to the connection between the eyes that is centered on a point 20% of the eye distance d below the eye connection. The rectangular bounding box of that ellipse determines the face crop (see Figure 2b). If a face is close to the edge of a photo, the bounding box is zoomed in and shifted to avoid including an area outside the photo (see the right-most face in the second row in Figure 1 as an example). We found that this approach produces satisfactory results even in cases where the top of the head is tilted forward or backwards and thus the face is foreshortened. The approach is also somewhat forgiving of errors in the detected eye positions (e.g., when the detected coordinates correspond to the corner of rather than the center of the eye).

[1] Out of 1540 detected faces from 821 photos, there were 8 real false positives (no faces), 10 faces of buddhas, madonnas, carnival masks, and paintings, and 1 face of a sea lion.

[2] Ellipse size is 3.8 x 4.7 times the eye distance d.

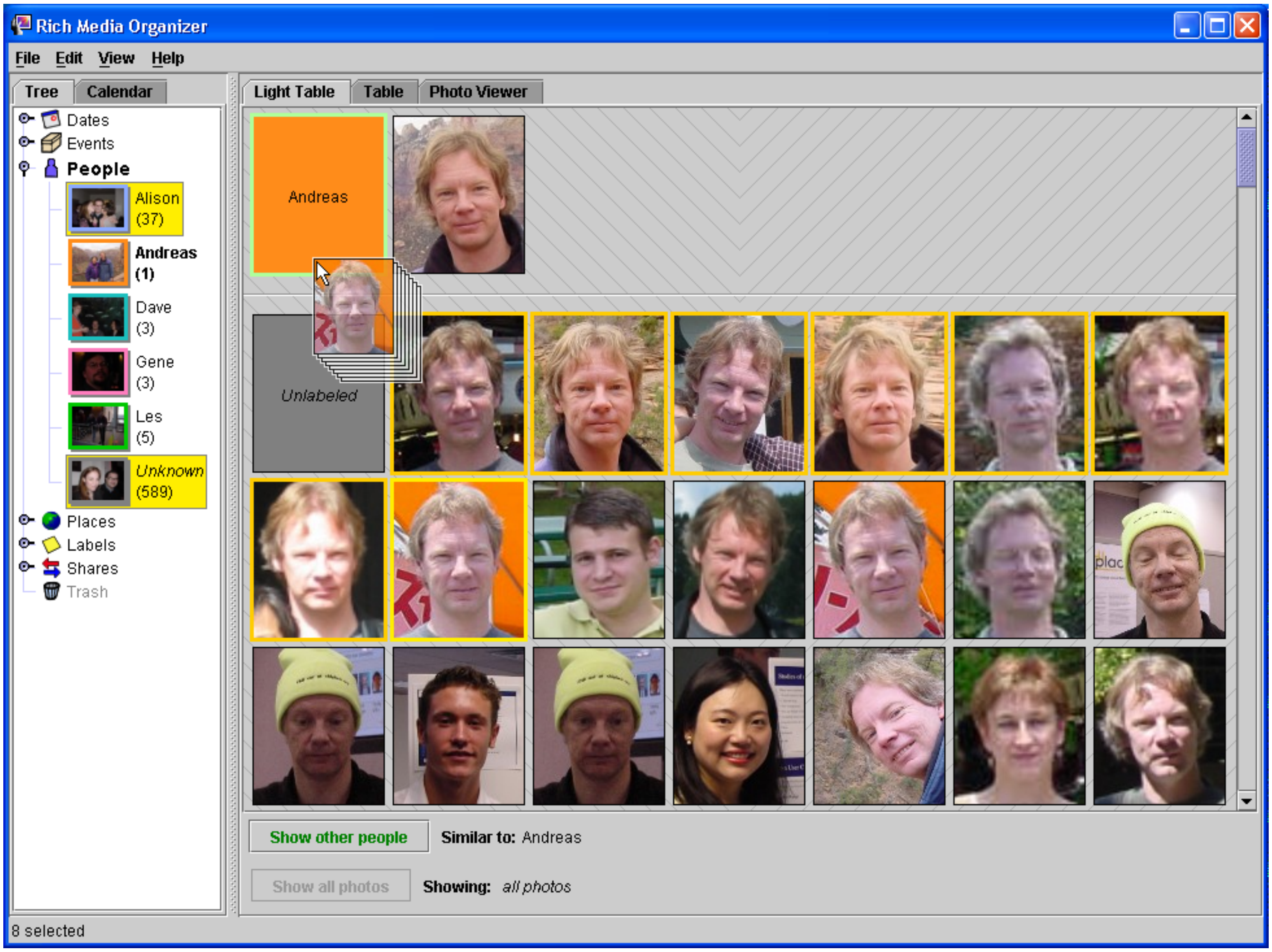


Figure 3: Finding pictures of Andreas. The faces are sorted by their similarity to the Andreas model. The eight highest ranked selected photos (shown as a translucent stack) are being dropped on the Andreas icon to label them.

4.3 Labeling Faces

The primary use of the face view in our application is to assign faces to people such that it is easy to find photos that depict a person or combination of people. The face thumbnails can be manipulated in the interface just like the photos and they act as proxies for the photos, i.e., operations on the faces are applied to the underlying photos. A face can be assigned to a person simply by dragging it onto any of the iconic representations of that person in the tree or the light table (see Figure 3). The act of assigning a face to person category also assigns the entire photo to that person category.

Once photos have been labeled with the people shown in them, they can be sorted or filtered by person so that photos depicting a single person or a combination of people can be found easily.

4.4 Automatic Face Recognition

The face detection library [9] we use also contains a face recognizer. Given a set of faces, a model for those faces can be constructed. The distance between the model and a face can then be measured. Model data for individual faces is computed as part of the face detection when importing new photos and stored on disk for later use. Models of hundreds of individual faces can be combined into a single model in a fraction of a second and a model can be compared against thousands of faces in a second, so that updat-

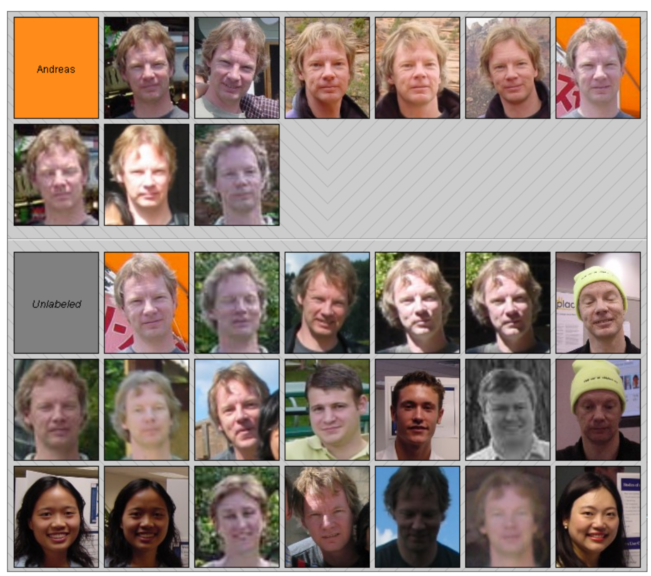


Figure 4: After dragging in Figure 3.

ing models and similarity scores on the fly is practical. Models do not grow in size as more faces are added so that models represent-

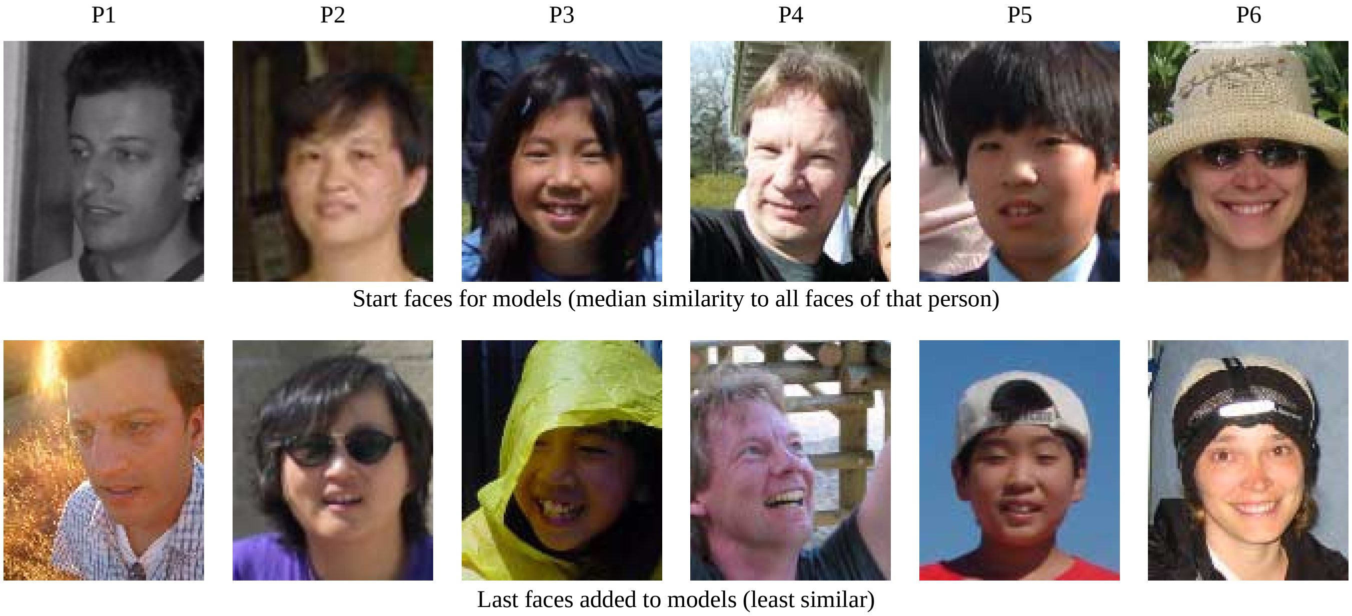


Figure 5: The first and last faces assigned for each candidate person.

ing thousands of faces are practical. However, as people change over the years, one can achieve higher accuracy by only using faces from recent years when trying to recognize new faces. As discussed later, models formed from 10 random faces already approach the maximum recognition rates.

Unfortunately, the face recognition accuracy is much lower than the face detection accuracy. The faces that are ranked closest to a particular model do not necessarily belong to the person used to build the model despite training the model with a large number of faces. This makes completely automatic classification infeasible to perform in an acceptably robust manner.

Presented with this limitation, we opted to assist the user in the labeling task instead of attempting to automatically assign faces to people. When one or more faces are associated with a person, a model representing those faces is created and the application displays unlabeled faces sorted in order of their similarity to that model (see Figure 3). This view allows the user to quickly decide which faces at the top of the similarity-sorted list depict the person of interest. Using a range selection or a wire frame, the user can select several faces and assign them to the appropriate person with a drag-and-drop interaction technique. The ability to select multiple faces allows for faster assignment of a large number of faces than in FotoFile [11] where the automatic face recognition has to be confirmed one face at a time. The model can be updated very quickly to incorporate the newly identified faces and the unlabeled faces are immediately resorted by their similarity to the new, presumably improved, model. Figure 4 shows the new view after the eight top-ranked faces in Figure 3 were assigned to a person. Now the nine top-ranked faces can also be correctly assigned to the same person.

To further simplify this face labeling task, faces that are already associated with a person other than the current person of interest are not shown in the light table. Furthermore, faces from photos where another face is already associated with the person in question are not shown either. While there are situations in which the same person is shown more than once in a photo (e.g., in a mirror), those situations are rare enough to let users deal with them in views where all faces are shown. The user interface presented here allows users to label dozens of faces very quickly.

4.5 Usage Model With Other Face Recognizers

Our approach for assisting the user in the face labeling task can be done with any face recognition library that can sort faces by similarity to a model and that can update the model quickly after more faces have been added. The fact that the face detector that we use can detect eye positions is beneficial for cropping faces but face detectors that only return bounding boxes could be used as well, resulting in a slightly less appealing presentation of cropped faces.

5. SIMULATION RESULTS

Current systems for labeling photos with faces require users to confirm or to select names for every face. The reason is that state-of-the-art face recognizers are not robust enough to automatically label even a subset of the faces correctly. Zhang et al. [16] indicated that a list of 5 most likely names determined by a face recognizer using an Euclidian face distance measure would include the correct name for 75% of the faces. They were able to increase that number to 94% by using a Bayesian model that incorporates the body of the person as well.

While photo organizers are currently constrained by the accuracy of face recognizers, we believe that there are approaches like the one we describe in the this paper that can make the labeling task less tedious and more efficient. Specifically, it is very efficient to assign a group of faces to a person at once rather than to select a name from a list for every face. Our approach is dependent on the combination of an accurate face detector and a less accurate face recognizer to present as many correct faces as possible at the top of the similarity list shown in Figure 3.

In this section, we examine how well our strategy would work by conducting a simulation of the canonical use of our labeling system for a collection containing the faces of several people. We collect statistics on how close the correct faces are to the top of the similarity list, and on how many correct faces can be found without having to scroll the window. Our approach assumes that users will create a category for a new person with one or more faces and then drag similar faces onto that category. As the model is updated with new faces, more faces of the person of interest will percolate to the top of the sorted face display where they can also be quickly

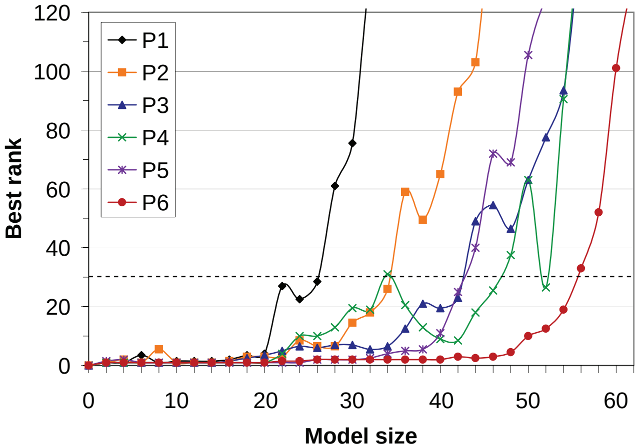


Figure 6: Rank of the best matching face at each step.

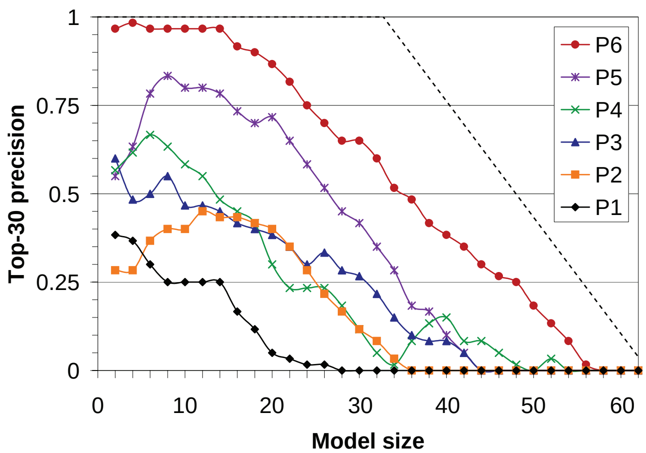


Figure 7: Fraction of correct faces among 30 most similar to the model (precision).

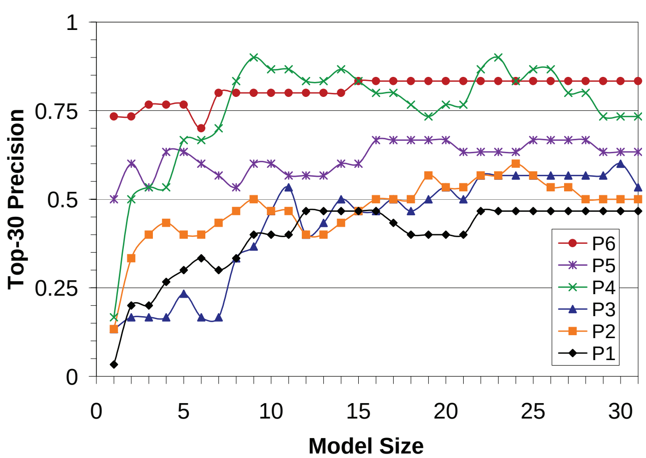


Figure 8: As Figure 7 with model updated from separate training faces.

selected and added to the model. The process continues until all faces are found or the most similar remaining face of the person cannot be seen and the user stops labeling. Thus, our experiment examines what percentage of the faces for a person can be labeled before the remaining faces are too far down the list to be easily found (i.e., not on the first screen of results). For the experiment, we assume that users would be able to see at least 30 faces. This number of faces can easily fit in the application window even on a small screen[3].

For the experiment, we labeled all faces of people in a large photo collection created by merging several personal photo collections. We used 63 faces each from 6 people. Figure 5 shows two rows of faces of those people in different poses. We also included an equal number of faces from other people for a total of 756 faces in 488 photos. The incremental labeling process described above requires a choice of a single face as starting point. Hence, for each of the 6 test people we selected as the seed face for that person the face with the median similarity to all faces of that person. Those faces are shown in the top row of Figure 5. The choice of that face should lead to results somewhere between starting with the best and the worst face.

Starting with the seed face, we iteratively added the most similar face for that person to the model. At each step, we determined the rank of the closest correct face in the list of all faces not contained in the model sorted by their decreasing similarity to the model. With this measure, a perfectly accurate face recognition algorithm would always produce rank 1. That is, the top-ranked face would always be correct.

The bottom row of Figure 5 shows the faces that were added last to the models and thus are the least representative of the depicted people. One can see that unusual face positions, back lighting, and partially covered face can be challenging for the recognizer.

Figure 6 plots the rank of the most similar correct face for each step of the experiment for the six people[4]. The horizontal axis displays the model size, i.e., the number of faces already assigned to the model. The vertical axis displays the best rank in the ordered list of a face belonging to the person represented by the model. While adding more faces to the model improves its accuracy, those faces also become unavailable for selection in the next step. As more faces are assigned to the model, the performance degrades because the "easy" faces are assigned first and the remaining faces depict poses that make it more difficult to recognize the person. As stated above, we assume that users can see 30 faces without scrolling the window. Points below the dashed line in Figure 6 indicate that the top-ranked correct face would be visible to the user.

Figure 7 shows how many faces are from the correct person when only looking at the first 30 faces in the sorted list (information retrieval precision). The horizontal axis is the same as in Figure 6 and the vertical axis displays the fraction of the 30 unassigned faces most similar to the model that picture the correct person. The dashed line indicates the ideal curve where the precision only drops below 1 when fewer than 30 faces remain unassigned (model size > 33 for 63 total faces) and thus the 30 most similar faces cannot all depict the correct person. Some people (e.g., P6) have a very high precision which means that many faces visible in the application window will belong to the target user and can be selected as a group to be assigned in a single step.

The precision in Figure 7 drops because the "easy" faces are assigned to the model and thus removed from the test set of faces. To look at the effect of the model size without removing faces, we also conducted a different experiment. For each person, we split the faces into a training set of 31 faces and a test set of 32 faces.

---

[3] 5 rows of 6 faces with 25% of the window width assigned to the tree view fit into a 900x800 window.

[4] To smooth the curves, we plot the average value for two subsequent model sizes.

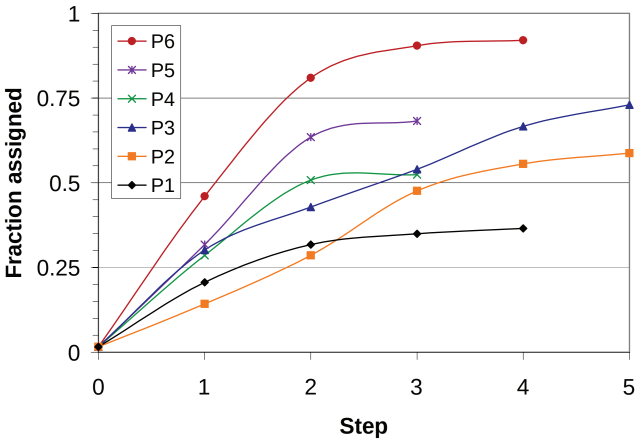


Figure 9: Assigning all correct faces among 30 most similar. All except P1 assign >50% in 4 steps.

The 378 faces of strangers were added to the test set for a total of 570 faces. Figure 8 shows how increasing the model size by adding more faces from the training set affects the precision of the 30 faces most similar to the model. The precision rates for small model sizes are lower than those shown in Figure 7 because the faces from the training set are unavailable for the similarity list. In this experiment, the test set of faces remains unchanged when increasing the model size. One can see that models with 10 faces produce a precision between 40% and 87% among the top-30 faces with only small improvements for larger models.

P3 and P5 are siblings with a strong family resemblance and interfere with each other. Either of their results would improve if the other were excluded. The 378 additional faces from strangers affected mostly the precision plotted in Figure 7 and made it on average 0.1 worse than it would be without the addition. P3 and P5 were affected the most by the addition of other faces, probably because the additional faces contain images of their parents.

To closely simulate the real usage, we also looked at the effects of assigning all correct faces among the 30 most similar ones in every step[5] (see Figure 9). The horizontal axis of the chart displays the step number and the vertical axis displays the fraction of the 63 faces assigned to the model. As before, the initial model consists only of a single face and thus the fraction at step 0 is $^{1}/_{63}$. Each graph ends when the top-30 faces do not contain any correct faces. In many cases, there are only minimal improvements after the first two or three steps after which 30% to 80% of the faces are assigned. With three or four steps, on average 60% of the faces can be assigned. This shows that at least 35% and in some cases more than 90% of the faces of a person can be labeled by just looking at the 30 most similar faces.

The experiment demonstrates that our approach allows users to assign a large fraction of faces with a few bulk drag-and-drop operations. The remaining faces can be assigned quickly as well as long as there are not too many strangers whose faces are not intended to be assigned. Faces already assigned to another person or faces from a photo already assigned to the current person are not shown. Therefore, assigning the “easy” faces for each person and then making a second pass once some faces of every person have been assigned would be an efficient approach for labeling faces. Users might also choose to label faces of other people while they have faces sorted by similarity to a particular person. This can be done by dragging a face onto a person node in the tree view with the result that the face disappears from the category of “Unknown” people. Strangers can be dealt with by creating a single label just for them so that those faces could be assigned to it and thereby removed from the light table.

[5] This assumes non-contiguous selections. If only the largest contiguous selection is used, more steps are needed.

6. CONCLUSIONS

This paper introduced and demonstrated how faces in photos can be a valuable resource to help with the challenge of organizing large photo collections. Face detection and face recognition stand apart from other content-analysis techniques because they are so uniquely relevant to so many of the photos typically found in personal photo collections. While the face detection we employ has an extremely low false alarm rate, it is limited by only being able to find faces in frontal views. We expect as the technology matures, it will be able to more quickly and thoroughly detect faces in all aspects, sizes and angles.

Because current face recognition technology is not accurate enough to automatically assign faces to people without requiring significant user intervention to correct mistakes or resolve ambiguities, we developed a semi-automatic system that automatically detects faces and allows the user to sort them by similarity to a previously labeled group of faces. We designed a usage model based on this capability wherein the user initializes the face recognition model for a person of interest with one or more faces and then uses the sorted face view to mine the unlabeled faces for instances of the person in question. This approach takes maximum advantage of the strong face detection performance and circumvents the relatively poor performance of the face recognition technology with a user interface appropriate for the users’ task. We quantitatively demonstrate that our approach allows users to assign a large fraction of faces with a few drag-and-drop operations.

The different uses of faces for the organization and presentation of photos provide a good view of the possibilities in using faces as a resource when dealing with personal photo collections. They can make photo organization a simple and enjoyable activity. The techniques described in this paper are incorporated in a commercial application that is currently in beta test. We will use the results of the beta test and other user feedback to further improve our photo organizer. Finally, we believe that our usage model can be adapted relatively easily for other face detection libraries that a) can return bounding boxes for the faces and optionally detect eye positions, b) can sort faces by similarity to a model, and c) can update the model quickly after more faces have been added.

7. REFERENCES

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

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

[3] Apple Computer. iPhoto. http://www.apple.com/iphoto/

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

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

[6] Drucker, S.M., Wong, C., Roseway, A., Glenner S., De Mar, S. MediaBrowser: Reclaiming the Shoebox. Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 433-436, 2004.

[7] Girgensohn, A., Adcock, J., Cooper, M., Foote, J., and Wilcox, L. Simplifying the Management of Large Photo

Collections. Human-Computer Interaction INTERACT '03, IOS Press, pp. 196-203, 2003.

[8] Hua, X.-S., Lu, L., and Zhang, H.-J. Photo2Video. Proc. of ACM Multimedia, pp. 592-593, 2003.

[9] Ioffe, S. Red Eye Detection with Machine Learning. IEEE International Conference on Image Processing, vol. II, pp. 871-874, 2003.

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

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

[12] Lim, J.-H., Tian, Q., and Mulhem, P. Home Photo Content Modeling for Personalized Event-Based Retrieval. IEEE Multimedia, 10(4), pp. 28-37, 2003.

[13] Schneiderman, H., and Kanade, T. A Statistical Model for 3D Object Detection Applied to Faces and Cars. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 746-751, 2000.

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

[15] Suh, B., Ling, H., Bederson, B.B., and Jacobs, D.W. Automatic Thumbnail Cropping and its Effectiveness. Proc. ACM User Interface Software and Technology (UIST), pp. 95-104, 2003.

[16] Zhang, L., Chen, L., Li, M., and Zhang, H. Automated Annotation of Human Faces in Family Albums. Proc. of ACM Multimedia, pp. 355-358, 2003.