

PhotoSim: Tightly Integrating Image Analysis into a Photo Browsing UI

Chen Yaxi, Andreas Butz

Media Informatics, University of Munich, Germany
yaxi.chen@ifi.lmu.de, andreas.butz@ifi.lmu.de

Abstract. Current photo browsers for personal photo collections mostly use the folder structure or camera-recorded time stamps as the only ordering principle. Some also allow manually provided meta-data (tags) for organizing and retrieving photos, but currently, only professional tools allow a pre-grouping of photos by image similarity. We believe that similarity is indeed a useful criterion both for image retrieval and casual browsing, and that the tight integration of content analysis techniques in media UIs in general can lead to more powerful UIs. In this paper we present a prototype, in which we have tightly integrated image analysis techniques and user feedback into a zoomable user interface for browsing and sorting digital photos. We have discussed our system with domain experts and received encouragement as well as valuable ideas for future research.

Key words: Photo browsing, image analysis, personal photo collection, zoomable UI, clustering, user feedback.

1 Introduction

The digitalization of media influences many areas of our life, but often, the additional possibilities, which come with this change, are hardly used. Digital photography, for example, has brought along a steep increase in the overall number of photos taken, but the archives for digital photo collections mostly follow the same principles as those for analog photography. They use capture time and folder structures as the main ordering principles, and do not provide much advantage other than increased speed of retrieval [1]. In order to keep up with the growing amounts of data, novel paradigms for managing, archiving and retrieving digital photographs have become a major challenge for research, and these novel paradigms create the opportunity of more substantial changes in our way of dealing with digital media. According to studies [2], [4], users browsing their photo collections often do not have a specific search goal, or a technically rather unclear one, such as “look for a better photo of my daughter”. Sometimes they might just browse for pleasure or story telling. At the same time, users are very reluctant to use tedious and time-consuming techniques, such as tagging each photo manually. Most of them prefer their photo collections to be organized automatically and hence cope with the organization by available metadata, such as folder hierarchy or camera-recorded capture time [1], [3]. Therefore, most current photo browsers are based on the latter.

1.1 Related Work

The traditional UI for photo management (e.g., ACDSSee, Picasa, iPhoto or Windows Explorer) uses a two-panel strategy. When the user selects a specific folder in the left panel, all photos in this folder are shown as thumbnails in the right panel, where the user can navigate within the folder by a vertical scrollbar and select thumbnails to see the original photo. More elaborate approaches include treemaps and zoomable UIs. In order to use the available screen space in an optimal way and provide a consistent mental model of the entire collection's organization, it has become a key research issue to maximize the usage of a given area and present as many photos as possible in it. One promising solution is a zoomable user interface, which presents information in one large plane and lets the user smoothly zoom in to see information in more detail, or zoom out to get an overview. Photomesa [6] is one of the popular zoomable photo browsers. It adopts a time-ordering and space-filling presentation strategy. Based on the theory of a quantum treemap [7], each directory is displayed in a different size depending on its content, which provides additional visual clues in the overview. All folders are ordered chronologically, and photos inside each folder are displayed in a non-hierarchical way. PhotoFinder [8], [9] aims at the organization of manually tagged photos. It lets the user organize photos into "collections" which are presented with a representative image and also enables the user to search in a boolean query interface. Other systems, such as TimeQuilt [11], PhotoToc [12] and CalendarBrowser [13], group photos into different events and offer automatic selection of a representative image. Even on large displays, researchers are looking for interactive and space-saving presentations, such as circular [14] and spiral [15] displays.

2 Tightly integrating similarity and user feedback in the UI

Digital media collections allow us to derive additional ordering principles from the actual media content. In the case of photo collections, this means that image analysis can be used to derive low-level image features, which in turn can be used for the organization of images. This idea is used in a simple way in professional tools, such as Aperture [16], where successive pictures can be automatically grouped if they exhibit similar color histograms. This speeds up the initial phase of photo selection by conveniently partitioning the entire set into easily tractable subsets. Our goal is to go beyond this simple preselection mechanism by

- using more elaborate image features in the analysis process,
- applying this to entire collections or bigger subsets, not just to subsequent photos,
- tightly integrating image analysis in the UI for browsing and retrieving images,
- harnessing user feedback to improve over fully automatic techniques.

We are targeting a general audience, who produce high numbers of photos, but also are very reluctant to explicitly manually tag them. We believe that the tight integration of image analysis techniques in media UIs, combined with immediate user feedback, can partially make up for manual tagging and that it will lead to UIs, which op-

erate close to human categorization principles, such as visual similarity. In our previous work [17], a fully automatic organization has been provided by low level image features. Our current attempt to create such a UI enriched by content analysis is a zoomable photo browser, which uses image analysis to further structure photo collections below the level of time, but allows manual overrides of this ordering and infers feedback from the manipulations used for overriding. On a general level, the collection is organized hierarchically according to capture time (see Figure 1), and forms a tree with subsets, where photos can be grouped further according to their similarity. In order to do this, we compute low level image features in a preprocessing step and then cluster the images according to these features.

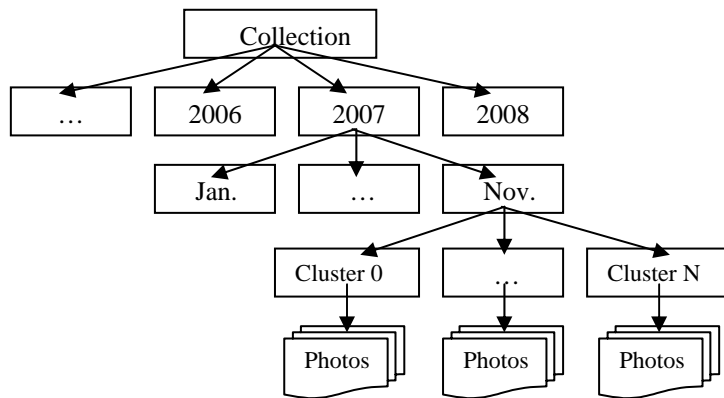


Fig. 1. Hierarchical organization of the photo collection according to capture time.

3 The PhotoSim User Interface

Our photo browsing application was implemented based on the prefuse toolkit for interactive Information Visualization [19]. PhotoSim presents photos clustered by time and content similarity and provides basic interaction techniques, such as drag-and-drop, pan and zoom. User feedback is derived implicitly from mouse operations (see Integration of user feedback).

The UI contains four main panels: a graph view, a tree view, an overview and a control panel, as shown in Figure 2. The three view panels act as coordinated multiple views. In the tree view, when a day, month, year or the entire collection is selected, all the corresponding photos appear in the graph view, creating a potentially very large graph. Photos can be clustered in this way at different levels of the temporal hierarchy. This allows, for example, finding similar photos across different days, months or even years.

The graph view is not limited in size, thereby not limiting the overall number of photos and clusters which can be shown. The user can freely pan and zoom within the graph view, but will always be provided with an overview of the entire graph in the overview pane in the top right corner. In the control panel (bottom right), the user can adjust the threshold used for clustering and explicitly save the current arrangement.

Since the location of each photo relative to the cluster center is determined by its similarity to the centroid photo and neighboring photos, a dent in the cluster shape visually identifies outliers, which are less similar than other photos in the cluster. These outliers can easily be dragged to other clusters according to the user's better judgment of similarity, or to any other ordering principle he/she may think of.

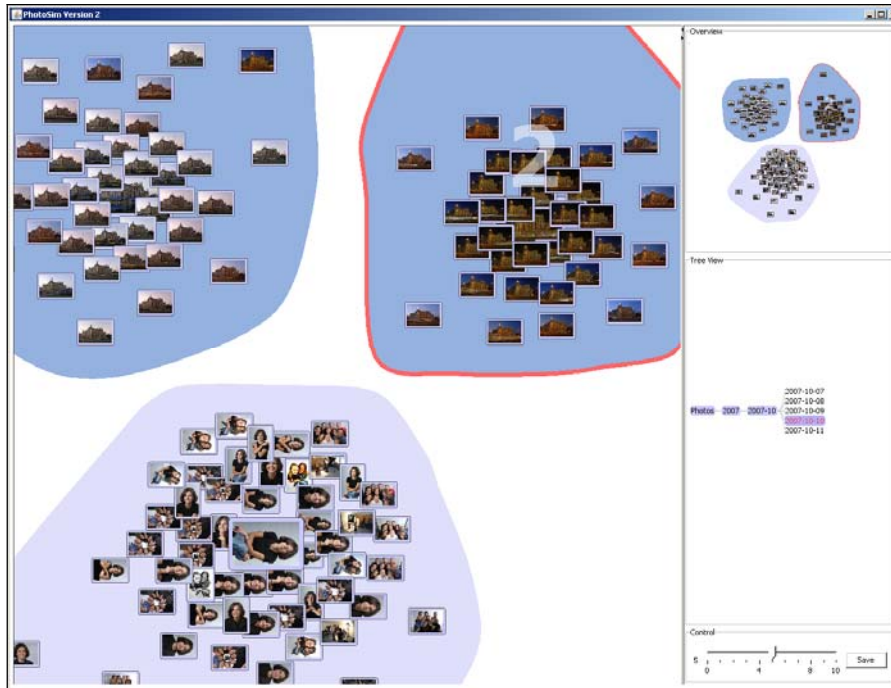


Fig. 2. The user interface of PhotoSim contains a graph view on the left and a column with an overview pane, a tree view, and a control panel on the right.

4 Image Analysis

The image analysis we use, entails several processing steps. First, a set of low level features is extracted from the photos. Based on these features, photos are then grouped into different clusters.

4.1 Low level features used

In order to compute similarity between photos, we use color features, textural features and roughness. Color features are computed in YUV color space, which has proven to yield results which are more consistent with human perception [17]. The U and V components are partitioned into 6 bins, which lead to a color histogram with 36 di-

mensions. Four related statistic features are also extracted: mean, standard deviation, skewness and kurtosis of the color moment in YUV [21], yielding an overall of 48 color-related features. Besides these, we extract Haralick features [20], which provide easily computable textural features based on gray-tone spatial dependencies and finally also roughness statistics derived from the gray level variation. In total, 161 features are extracted. Feature extraction does not work in real time and is done in a pre-processing step. The preprocessing of a collection of 813 photos (2M Pixel each) took about 2 hours in our current implementation.

4.2 Clustering algorithm

After extracting these low level features, the photos are grouped using a standard clustering algorithm. Since we cannot make any well-grounded assumptions about the photo content in the general case, the number of clusters in particular is unknown beforehand. Since this would have to be known for a supervised clustering process, we can only employ an unsupervised clustering algorithm. In our implementation, we use Simple K-Means [22], [23]. Figure 2 shows the clustering result of a photo collection with 3 different motives: portrait, building in daylight and at night. The clustering algorithm correctly groups them into 3 clusters, which is consistent with human perception in this case. Depending on the actual degree of similarity between and within motives, this approach might still create too many or too few clusters in the general case. Although this might also be tweaked (see Future Work), we are convinced that a fully automatic approach will never work perfectly in all cases.

4.3 Integration of user feedback

From the discussion above, it becomes clear that it will never be possible to perfectly match a user's visual understanding by a fully automatic approach. One solution to this problem is to take the human into the loop and combine the image analysis techniques with user feedback. We therefore provide manual overrides of the automatic results. The simplest such override is to just drag a photo from its original cluster into another existing cluster. On top of this, there are two ways to create new clusters:

4.3.1 Create a new cluster manually

First, the user can set the slider for the clustering threshold to zero, which means he/she wants to create cluster manually. Then the user can drag a photo outside its original cluster, and a new cluster will be created which contains this photo. If the user wants to add more photos in this newly created cluster, he/she can then manually drag other photos into this cluster one by one. This functionality can be used for permanent storage of personal favorites, or to create clusters as temporary structures for storytelling. Since all these operations are executed by the user manually, they are regarded as intentional and saved permanently.

4.3.2 Create a new cluster automatically

The user can also automatically create a new cluster of similar images from a manually selected example photo. First, he/she has to adjust the slider in the control panel to select a clustering threshold other than Zero. Then the user can drag the example photo outside the original cluster, as Figure 3(a) shows. A new cluster will be created which contains this example photo and other similar photos in the currently selected time period, as Figure 3(b) shows. If the clustering threshold is set to a high value, this means that the required degree of similarity is relative low, and therefore some other less similar photos will be also included in the new cluster. Since the result of this functionality is somewhat less predictable, the new cluster is not saved automatically in order not to destroy the existing presentation. If the user is satisfied with the result, he/she can click the “Save” button in the control panel to save it permanently.

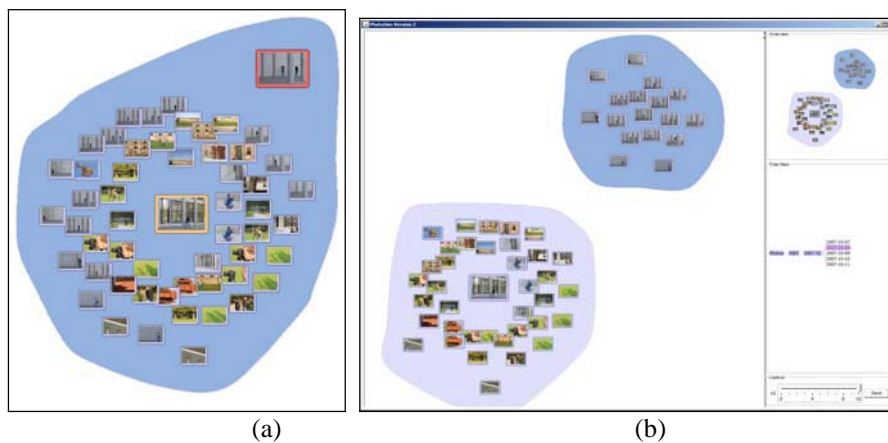


Fig. 3. Creating a new cluster automatically: (a) Dragging the example photo outside the original cluster. (b) The newly created cluster contains the example photo and other similar photos.

5 Preliminary evaluation

We have tested our prototype with five different photo collections. Four of them result from a photography workshop and were taken in the course of three days by four different ambitious amateur photographers. The size of the collections varies between 317 and 812. Another collection was offered by a former professional photographer and contained a selection of 603 photos in total, captured over five years.

Although we have not conducted a formal user study, the discussions with the authors of our example photo collections were quite encouraging. The former professional photographer said that he would like to use such a photo browser even more, if the structure of the collection after clustering and manual modifications could be translated back into a directory structure in the file system. He claimed that this might be a very valuable tool for potentially reorganizing his entire photo collection, not just the selection used in the test.

6 Summary and Future Work

In this paper, we have described an example for the tight integration of content analysis in a media UI and we have shown how this can lead to a more powerful set of manipulations for browsing and sorting of the media content. We have also seen that the tight integration with the UI is important, because it allows the user to manually override the automatic decisions of the system by providing implicit feedback. Although several forms of feedback are provided in our prototype, we still think that the interaction could be improved by allowing the user to organize photos in a more natural way, e.g., by supporting a more flexible and fluent sorting functionality. One parameter, which probably merits even more attention, is the average size and overall number of clusters for any given subset of a collection. The correct value will largely depend on personal taste and the actual content of the collection. Finally, if the similarity between any pair of two media items in the entire visualization could be visually approximated by the distance between them, this might close the gap towards self-organizing media networks as described in [25].

7 Acknowledgments

This research was funded by the Chinese Scholarship Council (CSC) and by the German state of Bavaria. We would like to thank the photographers who provided their collections as test sets and their opinions about our prototype, and we are also very grateful to Dominikus Baur, Michael Sedlmair and Marc Breisinger for their valuable input to this work.

References

1. K. Rodden, K. Wood. How do people manage their digital photographs? In CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems , pages 409–416, Ft. Lauderdale, Florida, USA, 2003. ACM Press.
2. S. Drucker, C. Wong, A. Roseway, S. Glenner, and S. De Mar. Mediabrowser: reclaiming the shoebox. In AVI '04: Proceedings of the working conference on Advanced visual interfaces, pages 433–436, Gallipoli, Italy, 2004. ACM Press.
3. D. Frohlich, A. Kuchinsky, C. Pering, A. Don, and S. Ariss. Requirements for photoware. In CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work, pages 166–175, New York, NY, USA, 2002. ACM Press.
4. D Kirk, A Sellen, C Rother, K Wood. Understanding photowork. In CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 761–770, Montréal, Québec, Canada, 2006. ACM Press.
5. Robert Spence. Information Visualization: Design for Interaction (Second Edition). Prentice Hall, 2006.
6. Bederson, B.B. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In UIST '01: Proceedings of the 14th annual ACM Symposium on User interface Software and Technology, pages 71–80, New York, NY, USA, 2001. ACM Press.

7. Shneiderman, B. Tree visualization with treemaps: a 2-D space-filling approach. *ACM Transactions on Graphics*, 11:92-99, 1992.
8. Kang, H., Shneiderman, B. Visualization methods for personal photo collections: browsing and searching in the photoFinder. In *ICME '00: Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1539-1542, New York, NY, USA, 2000.
9. Shneiderman, B., Kang, H. Direct annotation: a drag-and-drop strategy for labeling photos. In *IV '00: Proceedings of IEEE International Conference on Information Visualization*, pages 88-98, London, England, 2000.
10. C. North, B. Shneiderman, C. Plaisant. User controlled Overviews of an image library: A Case Study of the Visible Human. *Proceedings of the first ACM international conference on Digital libraries*, pages 74-82, Bethesda, Maryland, USA, 1996.
11. D. Huynh, S. Drucker, P. Baudisch, and C. Wong. Time quilt: Scaling up zomable photo browsers for large, unstructured photo collections. In *CHI '05: extended abstracts on Human factors in computing systems*, pages 1937-1940, 2005.
12. J. Platt, M. Czerwinski, and B. Field. Photoc: automatic clustering for browsing personal photographs. *Microsoft Research Technical Report MSR-TR-2002-17*, 2002.
13. A. Graham, H. Garcia-Molina, A. Paepcke, T. Winograd. Time as essence for photo browsing through personal digital libraries. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, Pages 326 – 335, Portland, Oregon, USA, 2002. ACM Press.
14. C. Shen, N. B. Lesh, F. Vernier, C. Forlines, J. Frost. Sharing and building digital group histories. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 324 – 333, New Orleans, Louisiana, USA, 2002. ACM Press.
15. O. Hilliges, D. Baur, A. Butz. Photohelix: browsing, sorting and sharing digital photo collections. In *TABLETOP '07: Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 87-94, Newport, RI, USA, 2007.
16. Aperture, software of photo editing and management: <http://www.apple.com/aperture>.
17. O. Hilliges, P. Kunath, A. Pryakhin, H. Kriegel, and A. Butz. Browsing and sorting digital pictures using automatic image classification and quality analysis. In *Proceedings of HCI International '07*, pages 882-891, Beijing, China, 2007. Springer Berlin / Heidelberg.
18. AF Smeaton. Content vs. context for multimedia semantics: the case of SenseCam image structuring. In *SAMT '06: Proceedings of Semantic Multimedia*, pages 1-10, 2006. Springer Berlin / Heidelberg.
19. Prefuse, interactive information visualization toolkit: <http://www.prefuse.org/>
20. R. M. Haralick, I. Dinstein, and K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 610-621, 1973.
21. M. Stricker and M. Orengo. Similarity of color images. In *SPIE '95: Proceedings of Storage and Retrieval for Image and Video Databases*, pages 381-392, San Jose, CA, 1995.
22. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y., Analysis of a simple k-means clustering algorithm. *16th Annual Symposium on Computational Geometry*, pages 100-109, 2000.
23. Weka, data mining with open source machine learning software, online available at <http://www.cs.waikato.ac.nz/~ml/weka/>
24. J. McQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics*. pp. 281-297, 1967.
25. Andreas Nürnberger and Marcin Detyniecki. Weighted Self-Organizing Maps: Incorporating User Feedback. In: *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003*, pages 176-184, Springer LNCS 2714, 2003