Browsing and Sorting Digital Pictures using Automatic Image Classification and Quality Analysis.

Otmar Hilliges, Peter Kunath, Alexey Pryakhin, Andreas Butz, and Hans-Peter Kriegel

Institute for Informatics,
Ludwig-Maximilians-Universität,
Munich, Germany
otmar.hilliges@ifi.lmu.de, kunath@dbs.ifi.lmu.de,
prykhin@dbs.ifi.lmu.de, butz@ifi.lmu.de,
kriegel@dbs.ifi.lmu.de

Abstract. In this paper we describe a new interface for browsing and sorting of digital pictures. Our approach is two-fold. First we present a new method to automatically identify similar images and rate them based on sharpness and exposure quality of the images. Second we present a zoomable user interface based on the details-on-demand paradigm enabling users to browse large collections of digital images and select only the best images for further processing or sharing.

Key words: Photoware, digital photography, image analysis, similarity measurement, informed browsing, zoomable user interfaces, content based image retrieval.

1 Introduction

In recent years analog photography has practically been replaced by digital cameras and pictures, which led to an ever increasing amount of images taken in both professional and private contexts. In response to this, a variety of software for browsing, organizing and searching of digital pictures has been created as commercial products, in research [1, 10, 17, 20] and for online services (e.g., Flickr.com, Zoomr.com, Photobucket.com).

With the rise of digital photography the costs of film and paper no longer apply and the storage and duplication costs have become negligible. Hence, not only the pure number of photos that are being taken has changed but also are people taking more pictures of similar or identical motives such as series of a scenery or person from just slightly different perspectives [9].

In consequence these changes in consumer behavior require more flexibility from digital photo software than support for pure browsing or finding a specific image. In this paper we present a software that supports basic browsing of image libraries namely the grouping of images into collections and the inspections thereof. In addition, the presented approach does specifically support users in selecting good (or bad) pictures from a series of similar pictures by the means of automatic image quality analysis.

1.1 Browsing, Organizing and Sorting Photos

An extensive body of HCI literature deals with the activities users engage with when dealing with image collections (digital or physical) [4, 6, 11]. For digital photos the whole life cycle – from taking the pictures, through downloading, selecting, and often sharing the photos as an ultimate goal – has been researched extensively.

All studies confirm that users share a strong preference for browsing through their collections as opposed to explicit searching. This might be due to the difficulty of accurately describing content as a search query versus the ease of recognizing an image once we see it. But even more important might be the fact that the goal for a search is, at best, unclear (e.g., "find a good winter landscape picture") even if the task (e.g., "create a X-mas album") is not.

Two strategies to support the browsing task can be identified. First, maximization of screen real-estate and fast access to detailed information through zooming interfaces [1, 8] is a common strategy. Second, search tools and engines help users to find pictures in a more goal-oriented way. Since images are mostly perceived semantically (i.e., the content shown), effective searching relies on textual annotation or so-called tagging of pictures with meta-data [10, 13, 20, 22]. However, users are reluctant to make widespread use of annotation techniques [19]. Hence, textual annotation of image collections is mostly found in the public and shared context (i.e., web communities or commercial image databases). In some commercial products (e.g., Adobe Photoshop), a content-based image retrieval (CBIR) mechanism is available, but its results are hard to understand for humans who apply semantic measurements for the similarity of images [18].

In addition to the browsing and searching activities users often and repeatedly sort, file and select their images. These activities sometimes serve archiving purposes so that only the best pictures are kept and are additionally organized in a systematic fashion. Users also sort and select subsets of images for short term purposes such as sharing and storytelling. For example, selecting just a small number of vacation pictures to present them at a dinner party with friends and family.

Current photoware does not account for this wider flexibility in users' behavior. Especially the sorting and selecting activities are seldom explicitly supported. Hence the common approach to assess the qualities of new photo software is to construct a browsing or searching task and then measuring the retrieval times [8, 17]. However, the time users spend with selecting and sorting is significant especially because these activities occur repeatedly (e.g., at capture time, before and after downloading, upon revising the collection). This suggests that supporting these processes may be central for photoware.

We think that automatic image analysis can help supporting users in the sorting and selecting tasks especially when these technologies are carefully instrumented to support the users' semantical understanding of images instead of stubbornly collecting as much data as possible to be used in a search-by-similarity approach – an attempt whose results might in the end be hard to understand for users.

2 Combining CBIR and Zoomable Interfaces

In our work we present a new approach to browsing and selecting of images based on a combination of CBIR and the zooming interface paradigm. The presented solution provides two mechanisms to help users in gaining overview of their collection in a first step. Furthermore the tool specifically supports selecting of images to decide which images "to keep" and which "to delete" in a second step.

In previous work similarity based approaches often pursued a search-by-similarity approach, for example returning similar images in response to specifying a certain image as query item. The problem with this approach is, that one has to find the search query item in the first place. Current photo collections easily extend the amount of several thousand images. Hence, without special treatment it is easy to get lost and as a consequence frustrated in this process.

We propose to utilize a pre-clustering algorithm to help users in narrowing down the search space so that users are supported in a more focused way of browsing. This makes it possible to deal with only a limited set of image groups (of similar content) instead of several thousand individual images. Ultimately this approach eases the process of finding pictures without explicit support for query based searching.

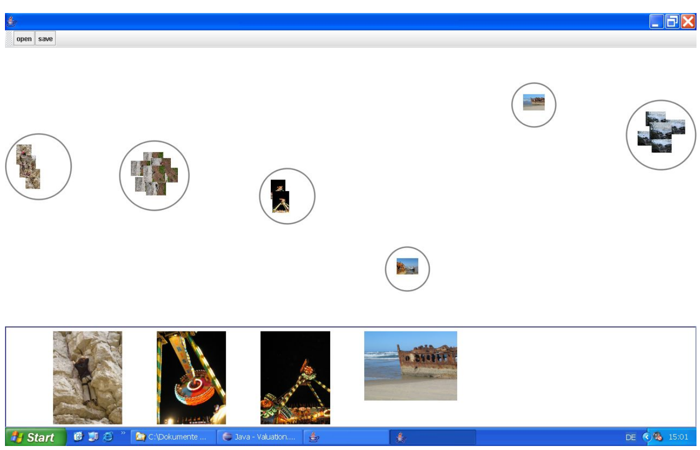


Fig. 1. Similar pictures are grouped into clusters. A temporary tray holds selected pictures from different clusters.

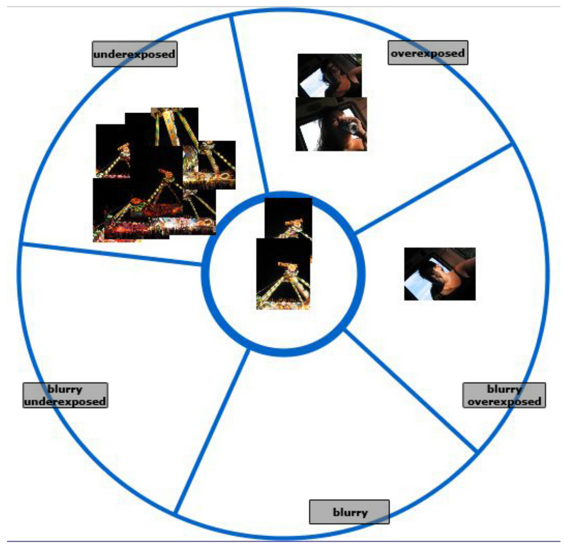


Fig. 2. Quality-based presentation of a cluster. The best pictures are in the center. Out of focus or too dark/bright pictures are grouped around the centroid.

In addition to browsing we wanted to support the selection of "good" and "bad" pictures. After grouping similar pictures together our software does an automated quality labeling on the members of each cluster. The criteria for the quality assessment are exposure and sharpness of images. Again, this step is meant to support users in isolating unwanted images or otherwise identifying wanted images while still maintaining an overview of all images in the respective cluster to facilitate the selecting process.

2.1 Selection Support through Semantic Zooming

In order to present a space-efficient view onto image collections we opted for an zoomable user interface which allows salient transitions between overview, filtered and finally detailed views of the collection and individual images respectively.

Upon startup the system is in the overview mode where pictures are matched according to a set of low level features. While this is not a real semantic analysis, it reliably finds groups of pictures of the same situation, which very often have similar content (See Figure 1). A few representatives are selected for each cluster (shown as thumbnails). The number of thumbnails in this view gives an approximation of the ratio of "good" pictures in the group versus the "bad" pictures. A cluster with many representatives has many pictures in the best quality group. The overall size of the cluster is depicted by the groups diameter - so spatially larg clusters contain many pictures.

Through fully zooming into one cluster users begin the selection of images. In this stage of the process clusters are broken down into six quality regions.

Fig. 3. Detail view of individual pictures in order to identify the best available picture.

The best rated pictures are shown in the center region while the five other regions serve as containers for the combinations of "blurry" and "under-" or "overexposed" images (See Figure 2).

Finally, individual pictures can be inspected and selected for further use, such as printing, sharing or manipulation also bad images could be deleted. On this last level images are ordered by the time of capture. We opted for this ordering to ensure that images taken of the same motive from slightly different angles appear next to each other, hence facilitating triaging of images (See Figure 3).

Users can zoom through these semantically motivated layers in a continuous way. The interface provides a good overview at the first levels by hiding unnecessary details. Whenever users need or want to inspect particular pictures they can retrieve these by simply zooming into the cluster or quality group respectively. At the lowest level, single pictures can also be zoomed and panned.

3 Image Analysis

In this section, we describe our approach to analyze a given collection of images. The analysis is based on a set of low-level features which are extracted from the images. In the first step, we identify series of images automatically by applying a clustering algorithm. The second step operates on each single series and matches the images contained in this series to different quality categories.

3.1 Extracting Meaningful Features

In order to describe the content of a given set of images, color and texture features are commonly used. Thus, for all pictures in a given collection, we calculate several low-level features which are needed later for grouping picture series and organizing each group by quality. The extracted features are color histograms, textural features, and roughness.

For the color histograms, we use the YUV color space which is defined by one luminance (Y) and two chrominance components (U and V). Each pixel in an image is converted from the original RGB color space to the YUV color space.

Similar to the Corel image features [14], we partition the U and V chrominance components into 6 sections each, resulting in a 36 dimensional histogram.

Although the HSV color space models the human perception more closely than the YUV color space, and is therefore more commonly used, we have shown in our experiments (cf. Section 4) that the YUV color space is most effective for our purposes.

The textural features are generated from 32 gray-scale conversions of the images. We compute the Haralick textural feature number 11 using the co-occurrence matrix [7], where N is the number of gray levels in the co-occurrence matrix $C = p(i,j), 1 \leq i,j \leq N$:

$$f_{11} = -\sum_{i=0}^{N-1} p_{x-y}(i) \cdot log(p_{x-y}(i)) \text{ , where } p_{x-y}(k) = \sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j), |i-j| = k$$

Finally, we also compute the first 4 roughness moments of the images [2]. The roughness basically measures some small-scale variations of a gray-scale image which correspond to local properties of a surface profile.

3.2 Identifying Series of Images

Our next goal is to detect image series. Pictures which belong to the same series have a very similar content, but it is possible that the quality of the pictures differs. So it seems reasonable to use UV histograms as the basis for this task. We ignore the luminance component (Y) because we are only interested in similar colors at this stage, but not in the brightness of the pictures.

In general, the detection of image series is an unsupervised task because there is usually no general valid training set for all kinds of pictures. Moreover, the number of image series in an image collection is usually unknown. As a consequence of these two observations, the method for image series detection should be unsupervised and has to determine the number of groups automatically.

We propose to apply an clustering algorithm for the image series detection. In order to distinguish series of images and to determine the number of image series automatically, we employ a clustering algorithm using X-Means [15]. X-Means is a variant of K-Means [12] which performs model selection. It incorporates various algorithmic enhancements over K-Means and uses statistically-based criteria which helps to compute a better fitting clustering model.

3.3 Labeling Images by Quality

The quality of a picture is a rather subjective impression and can be described by so called high-level features such as "underexposed", "blurry", "overexposed". We propose to use classifiers in order to derive high-level features from low-level features.

Support vector machines (SVM) [3] have received much attention for offering superior performance in various applications. Basic SVMs use the idea of linear separation of two classes in feature space and distinguish between two classes by calculating the maximum margin hyperplane between the training examples

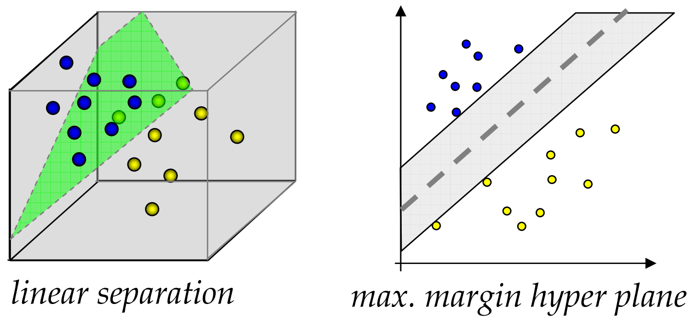


Fig. 4. Basic idea of a Support Vector Machine (SVM).

of both given classes as illustrated in Figure 4. Several approaches have been proposed in order to distinguish more than two classes by using a set of SVMs.

A common method for adapting a two-class SVM to support N different classes is to train N single SVMs. Each SVM distinguishes objects of one class versus objects of the remaining classes, this is also known as the “one-versus-rest” approach [21]. Another commonly used technique is to calculate a single SVM for each pair of classes. This results in $N * (N-1)/2$ binary classifiers. Finally, the classification results have to be combined by an AND-operation. This approach is also called “one-versus-one” [16]. The author of [5] proposes to improve the latter approach by calculating so-called confidence vectors. A confidence vector consists of N entries which correspond to the N classes. The entries are computed by collecting voting scores from each SVM. Thus, $N * (N-1)/2$ votings are summarized in one vector. The resulting class corresponds to the position of the maximum value in the confidence vector.

A SVM-based classifier maps low-level features, such as texture and roughness to group labels, which correspond to semantic groups such as "blurry" or "underexposed". We propose to apply an “one-versus-one” approach which is enhanced by confidence vectors because the “one-versus-rest” method tends to overfit, as shown in [16]. Users can either use an already trained classifier which comes with the installation archive of our tool, or provide training data to define their own quality classes.

4 Discussion

We have implemented a prototype, which can classify several hundred pictures within a few seconds and allows browsing them in real time. We evaluated our prototype using 3 different datasets (See Table 1).

In a first experiment, we turned our attention to finding a suitable feature representation for the automatic detection of image series. For each dataset, we investigated 3 different color models HSL, HSV and YUV. As discussed in Section 3, the luminance was ignored (i.e., we used only two of the three color

Table 1. Summary of the test datasets.

Dataset	content	# pictures	# series
DS1	animals	287	26
DS2	flowers & landscapes	328	35
DS3	flowers & people	233	18

dimensions for the histogram generation). Figure 5 depicts the quality of the clustering result for our datasets, which reflects the percentage of correctly clustered instances. We observed that the YUV feature achieves the best quality of the clustering-based image series detection for our datasets. Therefore the YUV feature was implemented in our prototype.

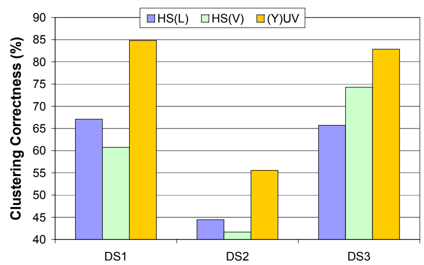
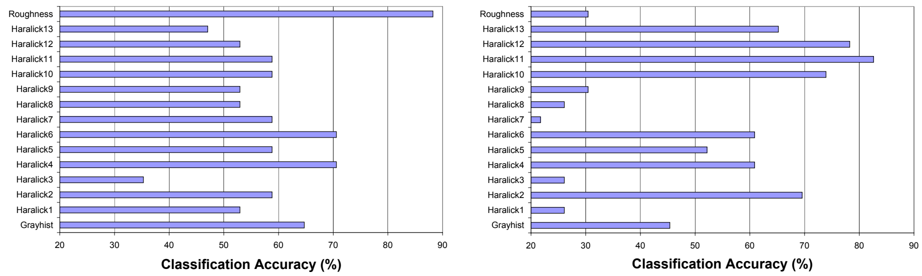


Fig. 5. Quality of clustering-based image series detection.

In a second experiment, various features were tested in order to find representations for the high-level feature mapping. We compared the suitability of different features which measure local structures of an image. Since the Haralick texture features and the roughness feature are based on a grayscale version of an image, we also included grayscale histograms in our evaluation. Figure 6 illustrates the results of our experiments. We observed that roughness performs well when distinguishing the classes 'underexposed/normal/overexposed'. For labeling the pictures according to 'sharp/blurry', the Haralick feature 11 seems to be the best choice.

To sum up, the performance of our prototype is encouraging and the classification according to high-level features matches human perception surprisingly well.

To this end we have not formally evaluated our prototype in a user study. The results from experience sessions with few users (who brought their own pictures with them) are encouraging. The things they liked most were the support for selecting images. One user said "this tool makes it easier to get rid of bad pictures and keep those I want". Also the possibility to quickly compare a series of similar images was appreciated. Others were surprised how good the similarity analysis worked.



(a) Accuracy w.r.t. underexposed/normal/overexposed.

(b) Accuracy w.r.t. sharp/blurry.

Fig. 6. Accuracy of high-level feature mapping (dataset DS1).

However, there were also things that our test candidates did not like. Foremost the lack of alternative sorting options. While most users found the grouping by similarity helped on narrowing down the search space some pointed out that a chronological ordering would make more sense in some situations. In future versions we plan to add support for different clustering criteria – basic ones – such as time or file properties as well as more complicated ones like identifying similar objects or even faces in the pictures.

We also plan to extend the scalability of the applied image analysis mechanism as well as the interface techniques to support more realistic amounts of data (i.e., several thousand instead of several hundred). Finally we plan to run extended user tests to further assess the quality of the similarity and quality measurements as well as the usability of interface.

References

1. B.B. Bederson. 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.
2. G. Chinga, O. Gregersen, and B. Dougherty. Paper Surface Characterisation by Laser Profilometry and Image Analysis. *MICROSCOPY AND ANALYSIS*, 96:21–24, 2003.
3. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
4. A. Crabtree, T. Rodden, and J. Mariani. Collaborating around collections: informing the continued development of photoware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 396–405, New York, NY, USA, 2004. ACM Press.
5. J. Friedman. Another approach to polychotomous classification. Technical report, Statistics Department, Stanford University, 1996.
6. 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.

7. R. M. Haralick, I. Dinstein, and K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:610–621, 1973.
8. D.F. Huynh, S.M. Drucker, P. Baudisch, and C. Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In *CHI '05: extended abstracts on Human factors in computing systems*, pages 1937–1940, New York, NY, USA, 2005. ACM Press.
9. A. Jaimes, Shih-Fu Chang, and A.C. Loui. Detection of non-identical duplicate consumer photographs. *Information, Communications and Signal Processing*, 1:16–20, 2003.
10. H. Kang and B. Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *IEEE International Conference on Multimedia and Expo (III)*, pages 1539–1542, 2000.
11. D. Kirk, A. Sellen, C. Rother, and K. Wood. Understanding photowork. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 761–770, New York, NY, USA, 2006. ACM Press.
12. J. McQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics*, volume 1, pages 281–297, 1967.
13. M. Naaman, S. Harada, Q.Y. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 196–203, New York, NY, USA, 2004. ACM Press.
14. M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, S. Mehrotra, and T.S. Huang. Supporting ranked boolean similarity queries in MARS. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):905–925, 1998.
15. D. Pelleg and A.W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
16. J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.
17. J.C. Platt, M. Czerwinski, and B.A. Field. Phototoc: Automatic clustering for browsing personal photographs, 2002.
18. K. Rodden, W. Basalaj, D. Sinclair, and K.R. Wood. Does organisation by similarity assist image browsing? In *CHI*, pages 190–197, 2001.
19. K. Rodden and K.R. 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, New York, NY, USA, 2003. ACM Press.
20. B. Shneiderman and H. Kang. Direct annotation: A drag-and-drop strategy for labeling photos. In *Fourth International Conference on Information Visualisation (IV'00)* , page 88, 2000.
21. V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.
22. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2004.