User-Friendly Personal Photo Browsing for Mobile Devices

Sang-Kyun Kim, Jae-Won Lee, Ryong Lee, Eui-Hyeon Hwang, and Min Gyo Chung

In this paper, a user-friendly mobile photo album system and albuming functions to support it are introduced. Stand-alone implementation in a mobile device is considered. The main idea of user-friendly photo browsing for albuming functions is to enable users to organize and browse their photos along semantically meaningful axes of events, personal identities, and categories. Experimental results demonstrate that the proposed method would be sufficiently useful and efficient for browsing personal photos in mobile environment.

Keywords: Photo classification, database management, MPEG-7, photo search, photo browsing, photo indexing, ontology, semantic indexing, photo categorization, face detection, face search.

Manuscript received Aug. 29, 2007; revised Feb. 4, 2008.

Sang-Kyun Kim was supported by 2007 Research Fund of Myongji University, Rep. of Korea, and Min Gyo Chung was supported by a research grant from Seoul Women's University (2007), Rep. of Korea.

Sang-Kyun Kim (phone: + 82 31 330 6443, email: goldmunt@gmail.com) is with the Computer Engineering Department, Myongji University, Yongin, Rep. of Korea.

Jae-Won Lee (email: jwonlee@samsung.com), Ryong Lee (email: ryong.lee@samsung.com), and Eui-Hyeon Hwang (email: ehhwang@samsung.com) are with the CIL Lab., Samsung Advanced Institute of Technology, Yongin, Rep. of Korea.

Min Gyo Chung (email: mchung@swu.ac.kr) is with the Department of Computer Science, Seoul Women's University, Seoul, Rep. of Korea.

I. Introduction

Recently, mobile phones equipped with digital cameras are becoming popular as a convenient way to take photos and video clips anytime and anywhere and to instantly review them at the moment of acquisition. At home, digital devices are gradually replacing traditional film cameras; consequently, the volume of digital photos is continually increasing. The digitization of photos and videos makes it easy for people to share their memories, experiences, and events with their friends and family.

In reality, people do not frequently move their photos or video clips from their digital cameras to personal storage on a PC or portable multimedia player (PMP), as moving pictures often involves extra work, such as selecting, sorting, and annotating. Unless people have enough time and passion to do this extra work or are willing to take care of the photos after each event, they typically leave the photos in the memory space of their digital cameras. The situation is similar for mobile phones equipped with digital cameras. There should be an efficient and effective means to automatically index and browse photos taken or stored in a mobile device.

The main problem with the traditional digital photo album is that users are often forced to perform this type of manual work. People often feel that it is a nuisance and that it is difficult to browse their photos in some meaningful order when they move their photos to a digital photo album on a PC or browse the photos stored on mobile devices. Manual cataloging is time-consuming, tedious, error-prone, and inconsistent. Thus, some automatic or semi-automatic methods are strongly required to manage a great number of photos and provide easier ways to browse groups of photos that are semantically linked together.

According to recent findings [1]-[3], people would be satisfied by automatic sorting tools that sort in chronological (time of acquisition) order by displaying thumbnails. Moreover, studies have found that people want to browse their photos by events so as to organize the photos into meaningful groups.

However, defining the concept of a meaningful group is one of the most difficult problems to resolve, as people often have different preferences or tendencies regarding how to classify their photos. Furthermore, related concepts generally contain a high level of semantics, so it is difficult to automate albuming.

The solution is related to traditional content-based image analysis. In recent decades, many related studies have been conducted in this area. Even for the specific domain of photo albuming, acquisition-time-based indexing [4]-[8] and event indexing [5]-[8] have been studied. For a user interface of a photo album, the thumbnail view is reasonably effective for browsing for users to see an entire series of photos in a short time. In particular, in a related study [5], [6], a combination of a time stamp and the color information of a photo have been used to automatically cluster photos according to specific events. However, as previously mentioned, high-level semantic events tend to exhibit only a slight relationship with low-level visual features, such as color, texture, or shape; thus, visually dissimilar photos could be labeled as belonging to the same event. It is difficult to capture event concepts from low-level features without interaction or feedback from the user.

In this paper, a user-friendly mobile photo album system is proposed and several albuming functions are described to support it. Although a user-centered mobile application for multimedia has been proposed recently in [13], the present proposal contains more advanced functionalities, providing user-friendly interactions. The main idea of user-friendly photo browsing for albuming functions is to enable users to organize and browse their photos or video clips along semantically meaningful axes of events, personal identities, and categories in a mobile device. Events are automatically indexed into meaningful groups by combining the situation groups extracted from low-level information with ontology analysis based on sparse annotations by the user. In addition, the functions must be light enough to be successfully processed in a mobile computing environment.

The background concept for photo albuming is explained in section II. The main functions are explained in detail in section III. The experimental results, actual user interface implementation, and indexing schemes along with the photo database management for efficient queries are explained in section IV. The proposed method envisions how the semantic concepts are practically organized.

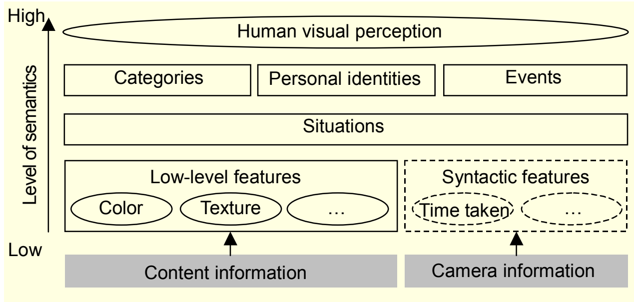


Fig. 1. Hierarchy of semantic albuming concepts.

II. Personal Photo Albuming

1. Concept of User-Friendly Digital Photo Album [20]

This section outlines a digital home photo album system. It starts by considering how users want to organize or browse their photos in a human perceptional way.

A. Scope and Goal

People would like to reorganize their original albums into a "personal album." They can reorganize their albums based on specific person(s), events, or by categories of pictures. Person groups can include "me," "my family," or even "my best friends." Event groups may be groups such as "Tom's high school graduation ceremony." Someone may be interested in collecting photos of special categories, such as "buildings," "animals," or "landscapes."

Ideally, automatic albuming should cover the whole range of user semantics. For this, all of the semantic concepts can be defined *a priori* and they can be modeled to provide desirable results for any user. However, it is impossible to make a complete concept space due to the uncertainty and the boundlessness of user semantics. Instead, the photo album would provide albuming functions based on several key semantics.

B. Semantic Photo Albuming Concepts

In this paper, three basic albuming functions for mobile applications are proposed: event clustering, categorization, and person identification. The main idea is to group the photos automatically or semi-automatically along semantically meaningful axes of events, categories, and personal identities with minimum manual effort on the part of the user.

Figure 1 shows a hierarchy of semantic albuming concepts. The target concepts (events, categories, and personal identities) contain the top level of semantics. They are close to human visual perception in that people often manually annotate a group of photos. Basically, content-based low-level features are combined into syntactic features in order to obtain the target concepts. The syntactic features consist of camera information including the acquisition time, GPS, and camera setting, which is obtained from exchangeable image file format (Exif) headers of the photos [14]. They play an important role in bridging the low-level semantics to the higher-level semantics.

In this paper, an intermediate level of semantics known as the *situation* is also proposed. With the situation grouping, people can see photos not only in time-order but also in groups with similar image background properties. Moreover, the situation group is related to location, specifically, the geographical information in which the photos were taken. The situation can be a fundamental cluster unit based on human visual perception. The situation includes a lower level of semantics compared to the event. The situation cluster helps users to review their photos quickly in terms of time and location. This facilitates joining a series of situation clusters into an event.

III. Algorithms

Digital photo management describing user-friendly functions is explained in next three subsections.

1. Event-Based Photo Clustering

A. Situation-Based Clustering

A solution that includes a means to roughly group pictures that are taken sequentially is proposed. Pictures are often grouped according to similar background scenery. The similar background scenery serves as a clue for the rough grouping of the pictures. With this rough grouping, people can see their pictures not only in time-order but also in groups with similar background properties.

In this paper, a set of photos with similar location, time, and view direction is referred to as belonging to a situation. In general, photos associated with the same situation are known to have close relation in time. Time differences between photos from the same situation are relatively small compared with time differences between photos from different situations. Hence, the acquisition time stamp information of a photo plays a significant role in situation-based photo clustering. The time stamp can be obtained from the metadata of the acquisition time and date from a standard header such as Exif [14].

Fig. 2. Example of photos with several situation changes.

Figure 2 shows a series of photos with several situation changes. The 15 photos in the figure were taken on the same day but at different places. They are shown in an ascending time order. There are five situation change boundaries; thus, six situation groups can be created. The situation change boundaries were manually determined by human perception. As shown in Fig. 2, photos associated with the same situation group appear to have a similar background. However, they appear to have slightly different foregrounds due to the different characters, camera angles, camera zooming distances, and camera flash settings. Normally, the background region of a photo is larger than the foreground region. Thus, in terms of human perception, a change of the background is regarded as more significant than a change of the foreground.

The goal of situation clustering is to detect situation change boundaries in a sequential photo series. In order to obtain the situation change boundaries of photos, the input photos to be clustered are initially arranged according to their acquisition times in ascending order.

The situation change boundary is examined using a sliding window that contains several neighboring photos. The sliding window moves from the oldest photo to the most recent photo in an ascending time order. For example, if the number of photos is N, the size of the sliding window is 3 and the current photo is denoted as the i-th photo; the three photos, which are the $(i–1)$th, i-th, and $(i+1)$th photos, are considered to detect whether the current i-th photo has changed from the previous $(i–1)$th photo. Each time the window slides, multiple low-level features are extracted from the three photos. The multiple low-level features are from MPEG-7 [12] visual descriptors: one color descriptor (scalable color) and one texture descriptor (edge histogram) [21]. The default parameters from MPEG-7 XM software are used. Next, some of the multiple features are selected and weighted. The photos in the sliding window have the same feature set and the weight of each feature is automatically determined by analyzing the photo characteristics, such as degree of colorfulness and so on [15]. The situation boundaries are determined by combining the time-taken differences between photos and low-level feature differences.

B. Semantic Event-Based Indexing Using User Annotation

When people want to search for some photos from a photo collection, they commonly prefer to browse their photos by high-level semantic event groups, such as "a friend's wedding in 2005." While there have been many previous studies of photo event clustering, most of them only use the acquisition

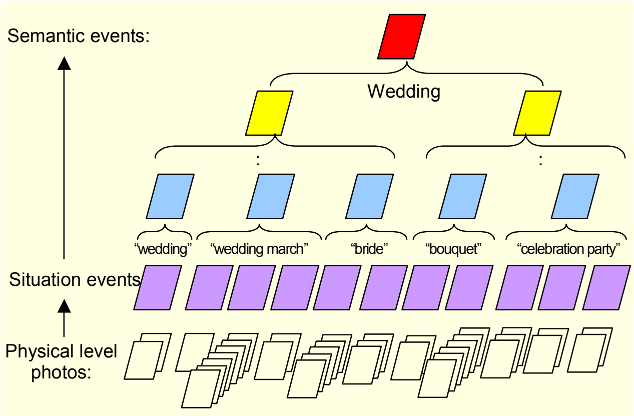


Fig. 3. Illustration of the semantic event clustering and indexing process.

date or acquisition time information and visual features [16].

Moreover, the short message service (SMS) in a mobile phone is popular these days. For semantic event clustering, it is assumed that users seldom annotate photos by using an easy annotation interface (such as phone buttons), which is similarly used for the SMS. Annotation for each situation is not required. If there happens to be some short annotation with a photo, this information can be combined with the situation clusters to produce a better semantic clustering of events.

In this section, a method that clusters given photos into semantic events is proposed. Here, it is assumed that users sometimes annotate some photos using an easy annotation tool. The proposed indexing method then exploits ontology to analyze sparse annotations by the user.

Figure 3 illustrates the process of semantic event clustering and indexing. The proposed method consists of two phases.

① Concept word extraction: certain types of event-related concept words are extracted from user annotations in a free text style. Keywords are then obtained, such as "wedding," "bride," and "commencement."

② Event clustering and indexing: the clusters semantically close to each other are merged. For instance, a "bride" is a woman participant in her own "wedding" event. Therefore, the initial cluster with the "bride" concept is merged into the cluster with the "wedding" concept.

C. Concept Word Extraction

The major objective of the concept word extraction phase is the extraction of pre-defined event-related concepts from photo annotations. Specifically, concepts related to the date, location, people, and object are extracted. As shown in Fig. 3, the concept word extraction phase is performed on the annotations of the clusters from a situation-based clustering described in the previous section. This phase is performed in the following four steps.

Step 1. Morphological analysis is performed, identifying morphemes in annotations and tagging parts of speech to them.

Step 2. Pattern analysis is performed, detecting event-related concepts by applying finite state grammar to the results of the morphological analysis.

Step 3. Stopwords are eliminated, filtering out words that are useless for clustering. Articles, prepositions, and conjunctions are included in this class.

Step 4. Concept analysis is performed, determining the concept of the words that remain after an ontology search and extraction of event-related concepts.

For instance, the annotated text for a photo is "family travel to Jeju Island." In this case, the concept words "Jeju Island: location," "family: people," and "travel: event" are extracted.

D. Event Clustering and Indexing

In order to merge the clusters that are semantically close to each other, semantic knowledge-based analyses of users' sparse annotations are utilized. Concepts related to given social events are extracted from a large corpus using collocation statistics, which were added to a WordNet-style ontology [17]. The following are examples of the related concepts.

- Wedding: bride, bridegroom, marriage, after-wedding celebration, bouquet.
- Commencement: address, speech, graduate, university, diplomas, cap, gown.
- Birthday party: gift, card, cake, candlelight, music, dinner.

The proposed event indexing algorithm is shown in Fig. 4. The initial clusters consist of the situation segments generated in the previous section. In addition to the time and content features, a location feature is used to cluster the event groups at a finer level. Each situation-based segment becomes an initial cluster by itself. The clusters may include event-related concepts extracted in phase 1. Two semantically related clusters are then combined into a new cluster, and the concept lists of the two clusters are also merged. When two clusters are

```
while (there is change) {
  for (i = 0; i < # of cluster; i++) {
    for (j = i+1; j < # of cluster; j++) {
      if NegativeSemanticRelation (i, j) then break;
      else if PositiveSemanticRelation (i, j) then
        Merge (i, j);
      else NoDecision;}
  }
}
```

Fig. 4. Semantic event clustering algorithm.

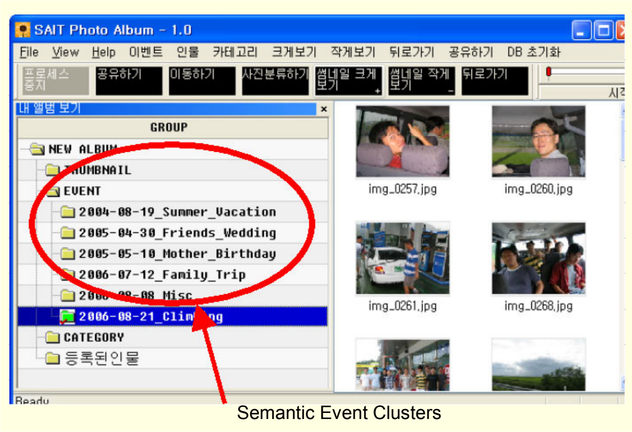


Fig. 5. Simulated example result of the semantic event indexing system.

combined into a cluster, they must be adjacent to each other and must have some semantic relationship to the event ontology. The combining process is continued until there is no semantic relationship between adjacent clusters at each level.

Figure 5 shows a simulated example of the result of semantic event clustering on a personal photo set from a PC. The same algorithm is implemented in the mobile environment. The proposed systematic method provides a more sophisticated way to give descriptive names to clusters, allowing event-based clustering functionality to enable users to browse and retrieve semantic events.

2. Category-Based Photo Classification

There are three steps in the automatic photo classification by category: local region feature extraction, local concept determination, and local concept concatenation for a dominant category determination.

The local region should be considered in the global concept generation, as a photo is composed of many different concepts (for example, a photo including a building, a mountain, and a lake). As current mobile phones do not have sufficient computing power, a photo is divided into three sub-parts (the upper-half, lower-half, and center) to acquire the local features. The scalable color descriptor (256 bins) and the edge histogram descriptor (80 bins) extracted from the situation grouping are used and are composed of 336 feature dimensions (672 bytes per image). Therefore, a photo has 3×336 feature vectors. As the feature vectors are neither orthogonal nor Cartesian, they can be optimized further. Moreover, each component of the features has its own characteristics.

For the determination of local concepts, the Adaboost algorithm is used. Two methods are considered for the classification. The Adaboost algorithm could weigh feature components that have different meanings differently. The algorithm can select important (non-orthogonal) components that have different meanings. Using the simplest discrete Adaboost algorithm, 200 features could be taken out of 336. Better performance can be achieved when additional features are used, but the performance is rarely increased when more than 200 feature dimensions are used.

Five categories, landscape, night-scene, architecture, miscellaneous, and portrait, are employed for the automatic classification. As a portrait can be determined by a face detector, a training set was provided for the remaining three categories (landscape, night-scene, architecture). The miscellaneous category contains photos that do not fit into any of the above categories.

Finally, the local concept concatenation part merges the results from three sub-regions. If the category results from all three regions are identical, the category is fixed as it is. If two category results are identical, slightly more weight is given to the two scores in order to reduce the effect of one dominant region. If all three regions return different category results, each score is compared with a threshold and the choice is then made. For a mobile device, only one dominant category result is assigned for each photo.

3. Face-Based Photo Search and Retrieval

Humans represent the most important component in ordinary photos. Therefore, the detection and recognition of this component can generate many useful applications. In order to recognize faces, a series of processes is required, which includes face detection, eye detection, face normalization, facial feature extraction, and similarity matching. Face detection is a step to find the locations of faces from an input photo. Eye detection locates the position of the eyes in the detected face regions. Face normalization normalizes a face image in terms of size and illumination. Facial feature extraction extracts face-specific information from the normalized face image. Similarity matching is a step that calculates the distance between two compared facial features.

A cascaded classifier trained by a gentle Adaboost algorithm is used to detect faces and eyes [9], [10]. A Harr-like simple feature is used for the face and eye detection step. For eye detection, eye candidates are found separately and the algorithm attempts to make a pair out of each combination of eye candidates.

The MPEG-7 advanced face recognition descriptor (AFRD) is used on the facial features for recognition [11]. The AFRD is based on facial image decomposition and on the projection of each face component by linear discriminant analysis (LDA). By combining this component LDA with holistic LDA, it is transformed into another LDA.

IV. Experimental Results and Photo DB Management

1. Experimental Results

A smart phone, the Samsung SCH-M4300, was used for the experiment and UI implementation. This device has a 520 MHz Intel Bulverde CPU and 57.03 MB of main memory. It has 60.0 MB of internal flash memory for database storage and uses a 2 GB SD card for content storage. The OS was Windows CE, and CE DB API was embedded. The display was a 2.8” 65K color TFT LCD with 320×240 resolution.

The model training (Adaboost) for the categories and faces/eyes was performed on the PC (off-line). The training data was then moved to the mobile device and used for classification. Except for the training process, the matching processes were performed solely on the mobile device. The extraction of features and classification were performed simultaneously when photos were saved in the mobile DB.

For the evaluation measure of photo clustering, recall, precision, and F_1 value were used.

$$recall = \frac{Detected}{Detected + Missed},$$

$$precision = \frac{Detected}{Detected + FalseAlarm},$$

$$F_1 = \frac{2 \times recall \times precision}{recall + precision}.$$

Photo clustering according to specific events was tested using 2,300 photos from an MPEG-7 photo database with a ground truth of 460 event groups. These photos were ordinary enough to be typical of any personal photo album. All of the photos were resized to a resolution of 400×300 in order to extract the visual features described in the previous sections.

The performance of the automatic situation clustering for the 460 ground truth situation groups of 2,300 photos showed the F_1 value of approximately 86%. Figure 6(a) shows the situation groups after situation clustering with representative photos from each event. Users can easily and quickly browse each photo event by looking at the representative photos and can find what they are originally looking for. Each event can then be browsed in detail, as demonstrated in Fig. 6(b). Using the text annotations of the user, each event can be automatically indexed according to the scheme explained in section III.1.B. With this implementation, users can easily search for relevant photos (events), such as “coast scenery,” “at the beach,” and “Pacific Ocean” by entering keywords, such as “sea.”

For the automatic photo category classification, 885 architecture images, 285 night scene images, and 495 terrain images selected from an MPEG-7 photo database were used. The F_1 value was 86.55% for the architecture photos, 95.09% for the night scenes, and 89.00% for landscapes. Figure 7 demonstrates browsing by categories containing the portrait category.

Fig. 6. Browsing by event groups.

Fig. 7. Browsing by category: (a) landscape, (b) architecture, (c) night scenes, and (d) portrait.

The face detector introduced in this paper only detects frontal faces (roll, pitch, yaw ≤ ±15 degrees). The face detector can

Fig. 8. Face detection and query by face.

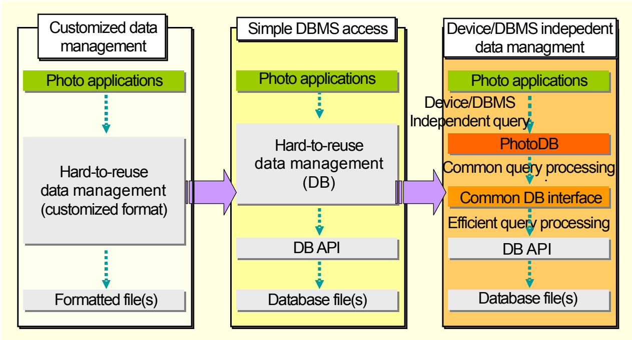


Fig. 9. Comparison of photo data management models.

detect faces with eyes that are more than 24 pixels apart. The detector detects faces regardless to their expression. The frontal face detection rate was approximately 99.9% with a personally corrected photo database containing 9,458 photos, while the eye detection rate was approximately 94.3% [10]. The face recognition rate was measured using the average normalization modified retrieval rate (ANMRR) [12], in which 0 indicates that all the ground truths are retrieved for the top ranks, and 1 signifies that all the ground truths are missed for the top m ranks. By using MPEG-7 AFRD, it was possible to obtain ANMRR 0.359 [11]. These results are better than those obtained by using the method described in an earlier study [11]. The left picture in Fig. 8(a) shows the result of the face detection using bounding boxes around detected faces. Figure 8(b) shows the retrieval results after giving the right side of a girl's face as an input query.

In the SCH-M4300 test device, the loading time for each photo was approximately 0.12 s. Event clustering took 0.61 ms for 367 input photos and 3.2 s for writing into the DB. As the DB access was performed on a flash memory, it required significant time. The automatic category classification took 0.614 s per photo and 0.6 s for the DB writing and thumbnail generation. For a query by face, it required nearly 0.6 s per query for approximately 800 faces.

2. Photo DB Management

Handling photos and metadata DBs in a mobile environment is another important issue. The main points of database management are efficiency, accessibility, reusability, and extensibility. To help with the increased number of functions and manage the visual knowledge data of photo applications, the data management model shown in the right side of Fig. 9 is proposed. The conventional model that is typically used does not use a database system as shown on the left side of Fig. 9. It only manages photos and metadata by means of its own data files. Thus, it suffers directly from storage management, memory/file synchronization, and very slow reaction time issues.

Compared to these direct management schemes, the adoption of database systems, as in the center section in Fig. 9, can help photo application systems and developers address such problems. Traditional database systems can handle dynamic data inserts, deletes, and searches without consideration of secondary physical storage management issues [18]. However, they remain unable to handle some query functions required in a visual knowledge database, such as a similarity search based on high-dimensional indexing [19], a continuous data query for sensor-based context-aware information, and a common data transfer protocol for the sharing and exchanging of visual knowledge.

For these additional requirements on a traditional DBMS, the proposed model splits the data management functions into a logical device/DBMS independent interface part and a physical common DB interface part. The first part can handle high-level application queries that are independent of devices and DBMS types. It defines application-specific query process functions, appropriate data schema, and network-transparent data access with a strong transaction policy. With the help of this flexibility, application developers can easily embed the systems into other devices or DBMS platforms. On the other hand, the second physical interface deals with common queries issued by the first part. This part consists of functions to process standard queries, such as DB schema designs, data inserts, deletes, and searches (with conditions). In contrast to the flexibility of the first part, this second part focuses on the efficiency of query processing.

In our implementation, the Windows CE database and its API were adopted as the fundamental DB platform. This DB platform is, in fact, too primitive and restricted to support the required functions for the photo album. Most mobile DBs do not support SQL-like high-level query languages. For example, sometimes there is a need to search/insert/delete photos by query combinations of "category" and "date" (for example, search/insert/delete "night scene photos between 2005 and 2006") but there is no transaction method for the CE DB API to deal with such a data consistency directly. High-level query commands such as "insertImage," "searchImagebyCategory," and "countImage" are defined in the level of the PhotoDB layer. Through the common DB interface layer, the high-level commands are translated and optimized to the low-level query commands discerned by the DB API layer. The DB API layer contains low-level DB controls and functions, such as "insert/delete/read," "Search API," and "Data Aggregator."

V. Conclusion

Using underlying metadata associated with digital content to represent semantically meaningful information from the perspective of a user is essential to providing an intuitive experience when managing media on mobile devices. A photo management application and a video browsing application built on top of a metadata-based database management service were presented. The underlying content management architecture and software modules were developed to interoperate with MPEG-7 visual descriptors and Exif metadata. The experimental results prove its usability in terms of performance as well as its easy and simple accessibility in terms of user interaction. This research will be built upon and extended to allow for more seamless exchanges of multimedia and its associated metadata (for example, in multimedia application formats) within a heterogeneous network of devices.

References

[1] M. Balabanovic, L.L. Chu, and G.J. Wolff, "Story Telling with Digital Photographs," *Proc. of ACM CHI*, 2000, pp. 564-571.

[2] B. Shneiderman and H. Kang, "Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder," *Proc. of IEEE ICME*, vol. 3, 2000, pp. 1539-1542.

[3] Y. Yagawa, N. Iwai, K. Yanagi, and K. Kojima, "The Digital Album: A Personal File-tainment System," *Proc. of IEEE ICMCS*, 1996, pp. 433-439.

[4] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd, "Time as Essence for Photo Browsing Through Personal Digital Libraries," *Proc. of ACM Digital Libraries*, 2002, pp. 326-335.

[5] A.C. Loui and A. Savakis, "Automated Event Clustering and Quality Screening of Consumer Pictures for Digital Albuming," *IEEE Trans. of Multimedia*, vol. 5, no. 3, 2003, pp. 390-402.

[6] J.H. Lim, Q. Tian, and P. Mulhem, "Home Photo Content Modeling for Personalized Event-Based Retrieval," *IEEE Trans. of Multimedia*, vol. 10, no. 4, 2003, pp. 24-37.

[7] J.C. Platt, M. Czerwinski, and B.A. Field, "PhotoTOC: Automatic Clustering for Browsing Personal Photographs," *Proc. of 4th IEEE Pacific Rim Conf.*, vol. 1, 2003, pp. 6-10.

[8] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox, "Temporal Event Clustering for Digital Photo Collections," *Proc. of ACM Multimedia*, 2003, pp. 364-373.

[9] J.B. Kim, Y.H. Sung, and S.C. Kee, "A Fast and Robust Face Detection Based on Module Switching Network," *Proc. of IEEE FGR*, 2004, pp. 409-414.

[10] J.B. Kim, S.C. Kee, and J.Y. Kim, "Fast Detection of Multi-View Face and Eye Based on Cascaded Classifier," *Proc. of IAPR Conf. on MVA*, May 16-18, 2005.

[11] T.K. Kim, H.W. Kim, W.J. Hwang, and J. Kittler, "Component-Based LDA Face Description for Image Retrieval and MPEG-7 Standardization," *Image and Vision Computing*, vol. 23, no.7, 2005, pp. 631-642.

[12] B.S. Manjunath et al., *Introduction to MPEG-7*, John Wiley and Sons, Ltd., 2002.

[13] B. Gandhi, A. Martinez, and F. Bently, "Intelligent Multimedia Content Management on Mobile Devices," *2004 IEEE ICME*, 2005, pp. 1703-1706.

[14] "Exchangeable Image File Format for Digital Still Cameras: EXIF Version 2.2 (JEITA CP-3451)," Standard of Japan Electronics and Information Technology Industries Association, 2002.

[15] S. Yang, S.K. Kim, K.S. Seo, and Y.M. Ro, "Automated Situation Clustering of Home Photos for Digital Albuming," *Electronic Imaging*, vol. 5682, 2005, pp. 212-223.

[16] A. Stent and A. Loui, "Using Event Segmentation to Improve Indexing of Consumer Photographs," *Proc. SIGIR*, 2001, pp. 59-65.

[17] G. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, 1995, pp. 39-41.

[18] H. Garcia-Molina, J.D. Ullman, and J.D. Widom, *Database Systems: The Complete Book*, Prentice Hall, 2002.

[19] C. Bohm, S. Berchtold, and D.A. Keim, "Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases," *Proc. ACM Computing Surveys (CSUR)*, vol. 33, 2001, pp. 322-373

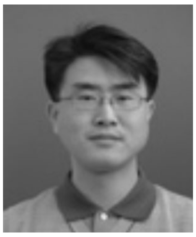
[20] S.J. Yang, K.S. Seo, Y.M. Ro, S.K. Kim, Ji.Y. Kim, and Y.S. Seo, "User-centric Digital Home Photo Album," *Proc. the 9th Int'l Symp. Consumer Electronics*, (IEEE Cat. No. 05TH8790), 2005, pp. 226-229.

[21] C.S. Won, D.K. Park, and S.J. Park, "Efficient Use of MPEG-7 Edge Histogram Descriptor," *ETRI Journal*, vol. 24, no. 1, Feb. 2002, pp. 23-30.

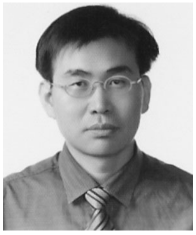
Sang-Kyun Kim received the BS, MS, and PhD degrees in computer science from University of Iowa in 1991, 1994, and 1997, respectively. In 1997, he joined the Samsung Advanced Institute of Technology as a researcher. He was a senior research staff member as well as a project leader on the Image and Video Content Search Team of the Computing Technology Lab until 2007. He is an assistant professor in the Department of Computer Engineering of Myongji University. His research interests include digital content (image, video, and music) analysis and management, fast image search and indexing, MPEG-7, and multi-modal analysis. He serves as a co-chair of the MPEG-7 Visual Core Experiment Group and a project editor of the MPEG-7 Visual Group.

Jae-Won Lee received the MS and PhD degrees in computer science from Korea Advanced Institute of Science & Technology (KAIST), Daejon, Korea, in 1993 and 1999, respectively. In 1996, he joined the Samsung Advanced Institute of Technology as a research staff member. He is a senior research staff member with the Computing Technology Lab. His research interests include digital content analysis and management, information retrieval, information extraction, and natural language processing.

Ryong Lee received the BS degree in telecommunications and information engineering from the University of Hankuk Aviation (Korea) in 1998, and the MS and PhD degrees in social informatics from the University of Kyoto (Japan) in 2001 and 2003, respectively. In 2003, he joined the Samsung Advanced Institute of Technology, where he is currently a senior research staff member of the Image and Video Content Search Team of the Computing Technology Lab. His research interests include multimedia databases, Web information retrieval, data mining and geographic information systems.



Eui-Hyeon Hwang received the BS and MS degrees in electrical engineering from Korea University in 1995 and 1997, respectively. In 1997, he joined the Samsung Advanced Institute of Technology as a researcher. He is a senior research staff member of the Image and Video Content Search Team of the Computing Technology Lab. His research interests include image processing and analysis, embedded system using DSP, fast signal processing, and system architecture.



Min Gyo Chung received the BS degree in computer engineering from Seoul National University in 1985, the MS degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1987, and the PhD in computer science from the University of Iowa in 1996. He worked with Korea Telecom until 2000 and Vivcom Inc. until 2002. He is now an assistant professor of Seoul Women's University. His current research interests include computer vision, pattern recognition, video/image compression, biometrics, and digital watermarking.