Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090334

Multidimensional Hierarchical Browser, Keyword Search, and Automatic Management of Photos within Smartphones

Suan Lee, Sunhwa Jo, Ji-Seop Won, Jinho Kim and Yang-Sae Moon*

Department of Computer Science, Kangwon National University, Korea

Received: 4 Aug. 2014, Revised: 5 Nov. 2014, Accepted: 6 Nov. 2014
Published online: 1 May 2015

Abstract: Recently new mobile devices such as cellular phones, smartphones, and digital cameras are popularly used to take photos. By virtue of these convenient instruments, we can take many photos easily, but we suffer from the difficulty of managing and searching photos due to their large volume. This paper develops a mobile application software, called *Photo Cube*, which automatically extracts various metadata for photos (e.g., date/time, address, place name, weather, personal event, etc.) by taking advantage of sensors and programming functions embedded in mobile smartphones like Android phones or iPhones. To avoid heavy network traffic and high processing overhead, it clusters photos into a set of clusters hierarchically by GPSs and it extracts the metadata for each centroid photo of clusters automatically. Then it constructs and stores the hierarchies of clusters based on the date/time, and address within the extracted metadata as well as the other metadata into photo database tables in the flash memory of smartphones. Furthermore, the system builds a multidimensional cube view for the photo database, which is popularly used in OLAP(On-Line Analytical Processing) applications and it facilitates the top-down browsing of photos over several dimensions such as date/time, address, etc. In addition to the hierarchical browsing, it provides users with keyword search function in order to find photos over every metadata of the photo database in a user-friendly manner. With these convenient features of the Photo Cube, therefore, users will be able to manage and search a large number of photos easily, without inputting any additional information but with clicking simply the shutter in a camera.

Keywords: photo metadata, photo annotation, clustered databases, multidimensional data cube, OLAP, hierarchical clustering, keyword search, multidimensional hierarchical browsing, mobile application, smartphones

1 Introduction

Since the recent advance of digital cameras, cellular phones, and mobile smartphones, it is very easy to take pictures thus it becomes popular for a user to get thousands or even more photos. As the number of photos increases, it becomes more difficult to manage and search them. While many photos have been accumulated, users have difficulties in using them for the future and eventually discard them. Therefore, we need a mechanism which facilitates classifying, storing, managing and searching photos easily taken by personal mobile devices.

The most popular way for users to manage photos is to give an appropriate name to every single photo file or to add description (or annotation) about it for future memory [1]. But it is very tedious or almost impossible for users to give such information to a large volume of photos taken in a short period. In order to lessen users' burden of inputting additional information, previous researches investigated several methods managing and searching photos conveniently. Most of these methods utilized low-level content information of photos such as colors, textures, and shape. Additionally GPS(Global Positioning System) was used to group/cluster photos geographically by some other approaches (e.g., Google's Picasa, Yahoo's Flickr, Apple's iPhone App., etc.). These methods utilized low-level information only thus they could not convey any real-world semantic information. Therefore, they can be used in limited purposes and/or they need additional tools like map. Zooming in and out the map with users' fingertips is often inconvenient rather than clicking country or city names.

In order to take advantage of semantic information for photos, social tagging and automatic annotation of images appearing in WWW (World-Wide Web) from their image

* Corresponding author e-mail: jhkim@kangwon.ac.kr

tag information [3,4,5,6,7,8,9,10,11] have been also proposed. These approaches have disadvantages requiring human interaction or prior information given. Recently, there have been several researches which extract automatically some semantic information from photos by using GPS [2] and other smartphone sensors [3]. These approaches are closely related to our approach but they only focused on limited information such as date/time and address and they did not investigate the functions searching and browsing photos based on semantic information.

Hence, recent smartphones maintain photo metadata EXIF(Exchangeable Image File Format) [12] and they are also equipped with several sensors such as GPS, Orientation, Illumination as well as networking functions to access external web sites. By taking advantage of these facilities, this paper develops a mobile application software, called ***Photo Cube***, which automatically gathers various semantic metadata for photos (e.g., date/time, address, weather, personal event, place such as building or district name, person name, etc.) and searches them from the metadata conveniently within mobile smartphones like Android phones or iPhones. In order to reduce the processing overhead gathering photos' metadata, this paper clusters incrementally photos into a set of clusters in real time and it extracts the metadata for each centroid of clusters automatically, not for all photos. Because of the clustering approach, we can make it feasible to provide the photo search feastures based on semantic information within smartphones. The following Fig. 1 shows an overall architecture of the Photo Cube system.

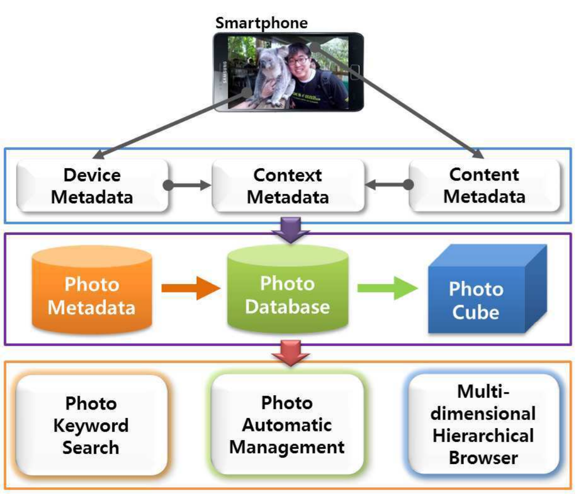


Fig. 1: Photo Cube Architecture

As shown in Fig. 1, the Photo Cube system extracts the device-related metadata (e.g., date/time or GPS) and then derives the context metadata (e.g., address, weather, personal event, etc.). Because it is time-consuming to get every context metadata, the system uses two strategies: (1) partitioning photos into a hierarchy of clusters over address, or date/time and getting the corresponding context metadata for each cluster's centroid photo; and (2) producing some of semantic information by a background process later on. The resulting information is stored into the Photo Database.

In order to incorporate flexible and convenient features for searching photos, the Photo Cube system builds a multidimensional cube view over the Photo Database which can facilitate searches over multiple dimensions (i.e., attributes) and top-down browsing for photos through OLAP roll-up and drill-down operations. In addition to the hierarchical browsing, the system provides users with keyword search feature to search photos in a user-friendly manner over every attributes of the Photo Database. With these facilities, users will be able to manage and browse conveniently many photos taken by mobile smartphones. The Photo Cube was implemented on smartphones using Google's Android and it was evaluated the performance by extensive experiments. The evaluation result shows that the system is feasible and effective for managing photos in smartphones.

This paper mainly focuses on presenting the overall system overview of the proposed Photo Cube system and it is an extended version of our previous work [13,14]. This paper is organized as follows. Chapter 2 describes the types of photo metadata. Chapter 3 explains how to extract semantic information from low-level metadata and the database structures storing it. Chapter 4 describes the multidimensional photo cube concept designed from the photo database and the searching features such as hierarchical browsing and keyword search. In Chapter 5, the performance evaluation for the Photo Cube system is discussed. We introduce the related works in Chapter 6 and we finally conclude the paper in Chapter 7.

2 Photo Metadata

Photos are related to various metadata such as shooting time, shooting place, and appearing people. These metadata are classified into three categories: Content metadata, Device metadata, and Context metadata. First, Content metadata represents the information contained in the photo itself, and it includes color, shape, and feature. Second, Device metadata represents the information obtained from the device such as cameras and GPS sensors. Finally, Context metadata represents the high-level, real-world semantic information derived from Device metadata and Content metadata. Fig. 2 shows three metadata categories collected in our Photo Cube system and their interrelationships. In this section, we explain how we can automatically extract these three categories of metadata in detail.

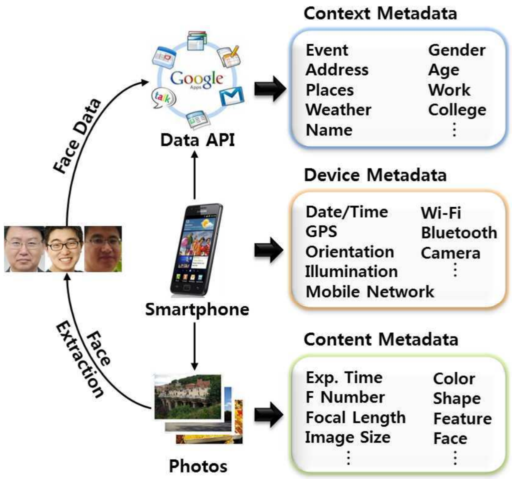


Fig. 2: Metadata Categories and Their Interrelationships

2.1 Device Metadata

Device metadata comes from device-embedded modules or sensors, and its representatives are date/time and GPS data. These date/time and GPS data are important since they represent temporal and spatial information, respectively, and they are also used as source data for deriving Context metadata. Date/time can be indirectly obtained from EXIF data [12] of a photo or can be directly obtained from the device clock when taking a photo. In this paper, we use the latter direct method since different devices might use different EXIF's date/time formats and the device clock is very accurate. Date/time metadata has a hierarchy of $year \geq month \geq day \geq hour$.

We extract GPS metadata from the GPS sensor module embedded in most smartphones with the form of {longitude, latitude, altitude}. GPS data is very important since it represents the absolute position of the photo taken. GPS data can be indirectly obtained from EXIF or can be directly obtained from the GPS sensor module. Like date/time data, we use the latter direct method since GPS data might not be stored in EXIF by users' settings. Most smartphones provide S-GPS(Simultaneous GPS) [15] locations directly computed from satellite signals, and they are more accurate than A-GPS(Assisted GPS) [16] locations computed by mobile communication networks.

2.2 Content metadata

Content metadata is the data contained in a photo itself such as colors, shapes, features, and faces. As shown in Fig. 3, in this paper, we use OpenCV library [17] to extract and maintain four types of content information, i.e., colors, shapes, features, and faces. This Content metadata is used to handling duplicated photos, clustering similar photos, and matching similar faces.

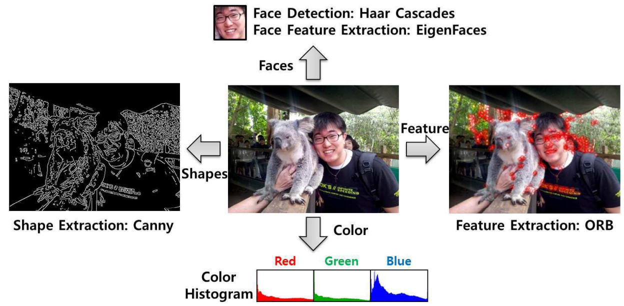


Fig. 3: An Example of Content Extraction using OpenCV

We now explain in detail how Content metadata from photos is extracted. First, a color histogram of [red, green, blue] from each photo can be easily extracted using the function calcHist [18] of OpenCV. Second, boundaries of objects as shapes of a photo is extracted, here we use the Canny edge detector function [19] for this. Third, Oriented FAST and Rotated BRIEF(ORB) [20], which is a recent feature extraction algorithm, is used to derive features from a photo. Fourth, regarding face information, Haar Cascades [21] for face detection and PCA and EigenFaces [22] for face feature extraction are used.

2.3 Context metadata

Both Content and Device metadata described above can be used to derive real-world semantic context data through Google API's. Fig. 4 shows how these Context metadata can be converted by Google Data APIs and XML Parsing modules. More detail description for the mapping and conversion is in the below.

1) **Address metadata**: From GPS data of a photo, its corresponding mailing address is mapped by using Google Geocoding API [23]. We maintain this mailing address for the photo as a real-world semantic metadata. Address metadata has a hierarchy of $country \geq area \geq locality \geq feature$.

2) **Place metadata**: In addition to mailing addresses, place names like building name, park name, or district name are a good source for handling photos. We extract place names from the Google map for the actual spots through Google Places API [24].

3) **Weather metadata**: There are several websites providing the weather information by both GPS data and date/time. We use Google Weather API [25] which provides the weather information including weather status, temperature, humidity, wind velocity, etc. in XML format. We extract these weather elements through XML

parsing. An example URL is http://www.google.co.kr/ig/api?weather=seoul which is to get the weather information for Seoul, Korea.
4) **Event metadata**: Users keep their own personal events or schedules in a calendar application/system. We extract these events from the Calendar App, an Android built-in application, and use them for tagging photos. The Calendar App is automatically synchronized with user's Google Calendar. User's events in Google Calendar can be also accessed through Google Calendar API [26].
5) **Person metadata**: We find person names appeared within a photo by comparing the face feature vector of each face to the faces registered in the Contacts App. The Contacts App is an Android's native application maintaining friends' information such as name, phone number, profile image, etc. The names for the matched persons are added as the Context metadata for the photo.

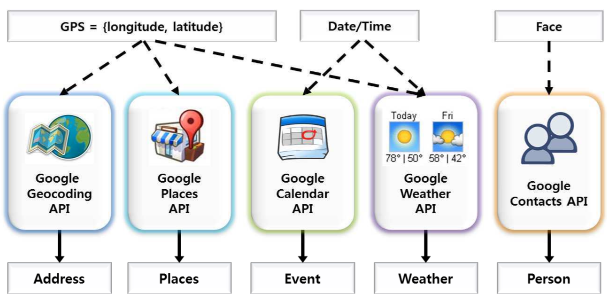


Fig. 4: Mapping/Converting Device/Content metadata to Context metadata

3 Photo Database

In this section, we design a photo database which efficiently stores photo metadata explained in Section 2 and supports the fast search and easy maintenance of photos. Because photo metadata have a lot of duplication, we normalize them for the photo database schema like Fig. 5. As shown in the schema, Date/time, Address, and Weather tables have 1:N relationships to the Photo table. The Photo table has 1:N relationship to the Person table because one picture contains several persons, and the Person table has 1:N relationship to the Face table because each person has several face data. The Event and Place tables have M:N relationships to the Photo table which are represented as two 1:M relationships to intermediate tables EventPhoto table and PhotoPlace, respectively.

In order to design tables for the photo database efficiently, we classify photo database attributes into three categories as follows:

–**Hierarchical type**: address and date/time.
–**Text type**: event, place, weather, and person.
–**Content type**: color, shape, feature, and face.

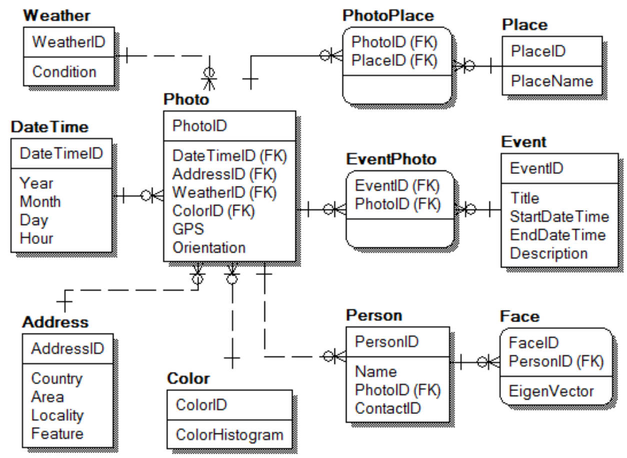


Fig. 5: Photo Database Schema

3.1 Address-based hierarchical clustered index tree

For a given photo, we can derive its mailing address from its GPS data using Google API. Using Google API, however, incurs the Internet connection to the Google site, and getting mailing addresses for all photos may cause a severe communication overhead. Thus, we need an efficient method of assigning mailing addresses to photos. We note that clustering similar photos is an efficient way for the group-based search, and this approach is currently widely used in browsing similar photos. Based on these observations, we devise the clustering-based address assignment, which works as follows.

First, we cluster photos based on their GPS data. For this, we perform the location-based hierarchical clustering using the M-tree [27], which is known to be efficient to hierarchical clustering. We get a mailing address for the centroid of each cluster and label the address to the cluster. Second, we build another hierarchical cluster tree, CH-tree(Concept Hierarchical Tree), by merging the clusters having the same address or adjacent addresses. We can use this CH-tree for searching photo clusters hierarchically according to the address taxonomy.

Fig. 6 shows an example of location-based hierarchical clustering. As shown in the figure, a number of photos are grouped into five clusters of which centroids are *n1, n2, n3, n5, n6*; five clusters are grouped into two higher level clusters (i.e., n4 and n7); and three clusters are grouped again into a highest level cluster. As a result, we eventually get a three-level cluster hierarchy in the

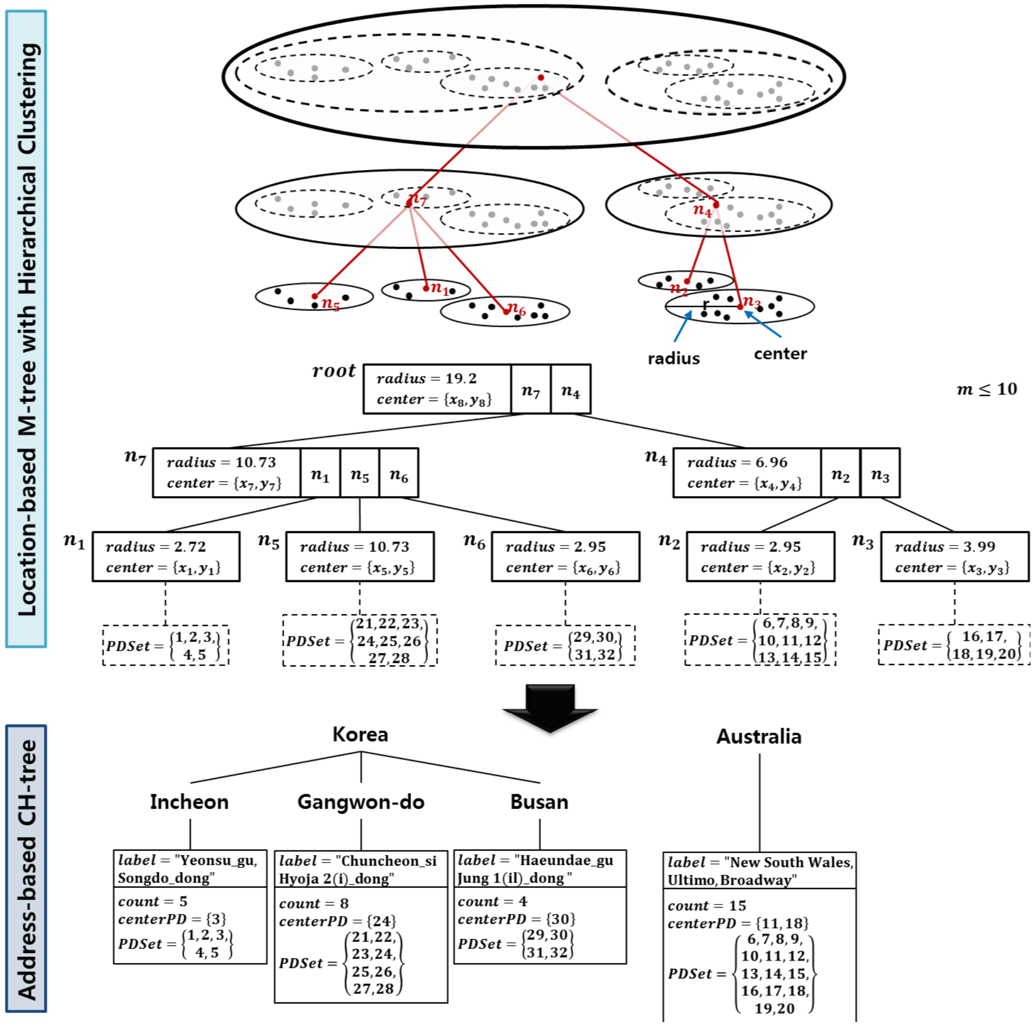


Fig. 6: An Example of Location-based M-tree and its Corresponding Address-based CH-tree

example of Fig. 6. We now build an address-based CH-tree from the clustering result. Fig. 6 shows an example. The upper part of Fig. 6 shows a location-based M-tree which is directly derived from the clustering result of Fig. 6. By assigning mailing addresses to the centroids of clusters, we can also obtain an address-based CH-tree as shown in the lower part of Fig. 6. Here, the size of a cluster (i.e., m) is set to less than or equal to 10, and each leaf node of an M-tree, which represents a cluster, maintains a centroid (center photo) ID, its radius, and its $PDSet = \{PD_1, PD_2, \ldots, PD_m\}$ containing photo IDs. (We set the cluster size as 10, because we believe that it is a reasonable number of clusters/photos to display on the limited screen size of smartphones. To get better clustering performance, we need to find an optimal cluster size but it is out of the scope of this paper.)

Now let us discuss how we build an M-tree and its corresponding CH-tree in more detail. We generate an M-tree by clustering photos based on their raw GPS data, and this incurs two problems: (1) some photos in the same or similar places can be divided into different clusters, and (2) exploring photos is difficult due to the lack of semantic labels of clusters. Thus, in this paper we first identify a mailing address for a centroid of each cluster (i.e., an M-tree node), and we next merge two or more clusters having the same address. We then assign address labels to clusters by the hierarchical concept, i.e., by comparing mailing addresses of hierarchical clusters. We call this hierarchical tree, derived from an M-tree using the address taxonomy, the Address-based CH-tree. We use the hierarchical structure of the CH-tree as an index tree to explore photo clusters based on the address hierarchy. In the traditional applications such as data warehouse and OLAP, the addresses follow a fixed address hierarchy of $country \geq state \geq city \geq streetname\&number$, but the proposed CH-tree can provide a dynamic address hierarchy that is dynamically and adaptively changed by the location density of photos and their clusters. Thus, we can conclude that the proposed CH-tree enables us to explore photos much faster and more flexible.

3.2 Date/Time-based hierarchical clustered index tree

Another important criterion of classifying and exploring photos is date/time. A usual pattern of taking photos is not periodic, but event-oriented, that is, a user takes a series of photos at a specific time period of an event, rather than takes photos periodically, and s/he takes another series of photos at the next event. Likewise, we usually take a series of photos at the unit of our life events. Based on this observation, in the proposed system, we group photos by exploiting the date/time-based hierarchical clustering and explore them based on the date/time hierarchy. For this purpose, in this paper, we propose a date/time-based M-tree, an extended version of M-tree by considering date/time characteristics, which enables the efficient hierarchical clustering for exploring photos using date/time attributes. The date/time-based M-tree has the same structure with the location-based M-tree. The different point is that, in the date/time-based M-tree, each cluster contains not only a centroid of date/time but also min and max values for the cluster size. Fig. 7 shows an example that we first construct a date/time-based M-tree from photos and their clusters, and we then convert it to a date/time-based CH-tree. As shown in Fig. 7, five clusters, *n1*, *n2*, *n3*, *n5*, and *n6*, are formed first, and a date/time-based M-tree is constructed from these clusters. We then assign date/time labels to clusters of the M-tree and construct a date/time-based CH-tree by merging nodes (i.e., clusters) having the same label. A label of a CH-tree node has a form of time interval. For example, if a cluster contains some photos taken at March 4 and other photos taken on March 4 to April 16, then its label becomes March 4 $\sim$ April 16 (or March $\sim$ April).

3.3 Color histrogram-based clustered table

Content derived from a photo itself is low-level metadata such as colors, shapes, features, and faces. This metadata can also be good criteria of classifying and exploring photos. For example, if we cluster a large number of photos by similar colors in advance, we can easily identify similar photos by the arrangement of color-based clusters. Color has three "channels" of red, green, and blue, and each pixel is a vector of these values, i.e., $I = \{I_{red}, I_{green}, I_{blue}\}$, where I $\in \{0,\ldots,255\}^3$. Fig. 8 shows an example of clusters grouped by three colors, red, green, and blue. In this example, we use RGB color histograms for clustering photos and obtain five clusters, C_1, C_2, C_3, C_4, and C_5. After then, we can use this clustering result to search photos having similar color features. In this paper, we also use the M-tree-based hierarchical clustering for color histograms, but we store lowest level clusters (i.e., leaf node clusters) only into the Color table. (Note that we build a CH-tree by merging similar clusters in date/time or location-based clustering.) This is because the concept of hierarchical search is not clear in case of color clusters. Moreover, this is because we use colors as a second criterion for browsing the intermediate result searched by date/time, locations, and/or persons. Table 1 shows an example of Color table that stores the clustering result of Fig. 8.

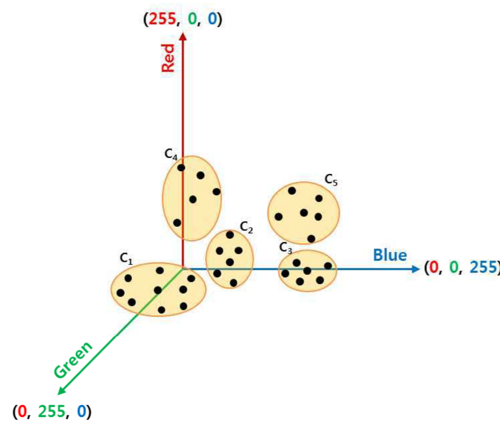


Fig. 8: An Example of Color Histogram-based Clustering

Table 1: An Example of Color Table

ColorID	ColorHistogram $\langle$Red, Green, Blue$\rangle$
C1	$\langle$14, 89, 8$\rangle$
C2	$\langle$31, 45, 79$\rangle$
C3	$\langle$8, 21, 133$\rangle$
C4	$\langle$91, 19, 11$\rangle$
C5	$\langle$86, 15, 162$\rangle$

3.4 Text-type table

Besides date/time and location, photo metadata has many useful data such as weather status, place names, personal events, and person names. These data can be represented as text strings, which is very simple compared with hierarchical structures for date/time and location data as we explained above. Tables 2 to 4 show such simple examples representing events, place names, weather status, respectively.

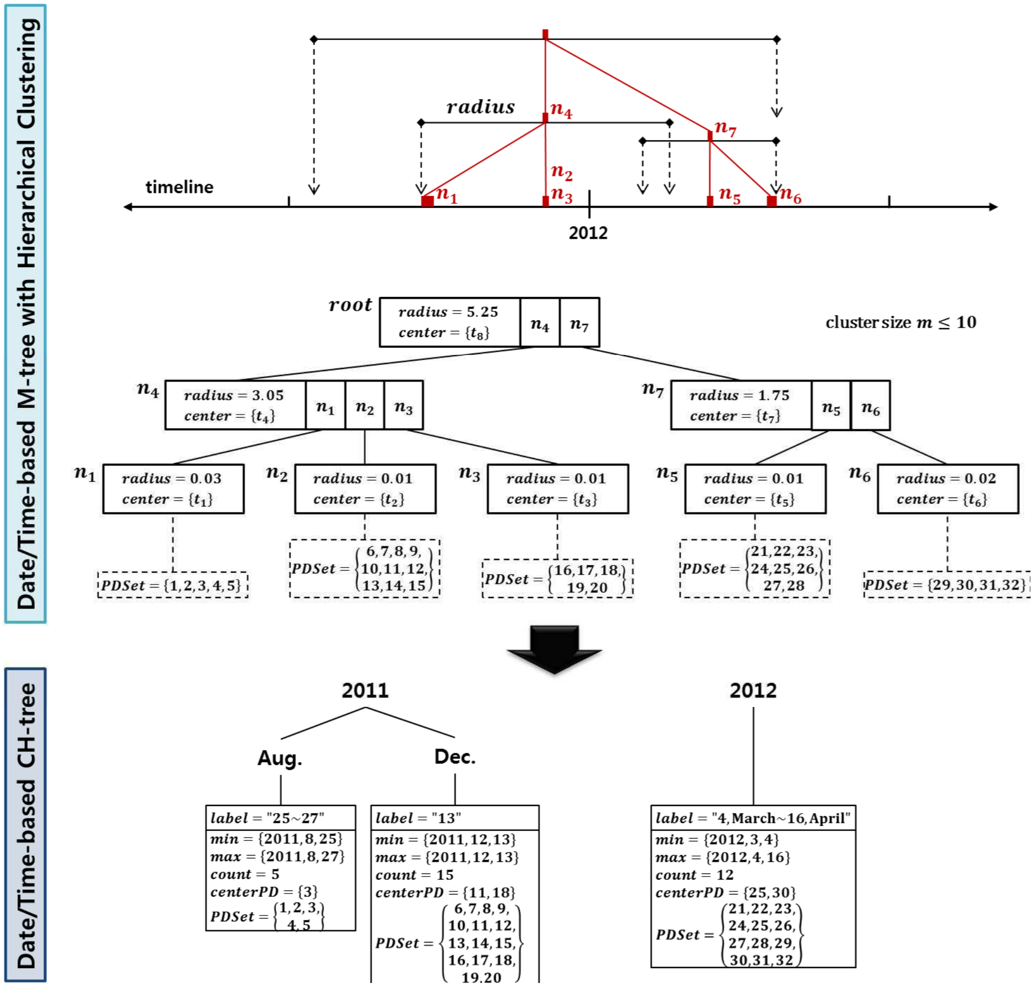


Fig. 7: An Example of Date/Time-basd M-tree and its Corresponding Date/Time-based CH-tree

To search photos using these various types of text data, string matching, in particular, keyword search is very useful. In our design, we use SQLite [28], a lightweight DBMS usually embedded in iOS and Android phones, since it provides a variable length of column for text strings. We build a text index for the text type column, and using the index we provide the efficient keyword search function. Thus, in this paper, we store weather status, place names, personal events, and person names into text type columns of a table and build the corresponding text indexes. These text type columns and indexes make the keyword search fast and efficient.

4 Photo Cube and Photo Browsing

Data Cube [29] is an essential components of OLAP, which are widely used in a variety of business applications for decision making process. In general, data cube has a multi-dimensional structure which maintains aggregate values of a measure for all possible combinations of group-by attributes. Using the data cube we can efficiently find aggregate values of various dimensions and efficiently support roll-up and drill-down operations for the dimensions having hierarchical structures. To provide more convenient browsing and various explorations of photos, in this paper, we organize a data cube for photos, called a photo cube, and propose new functions of browsing those photos in the viewpoint of OLAP operations.

4.1 Data cube and its OLAP operations

In Section 3, we explained the database structure storing various metadata for photos. From this photo database, we can construct a multi-dimensional data cube. For the photo cube, we use date/time and address, which are created in

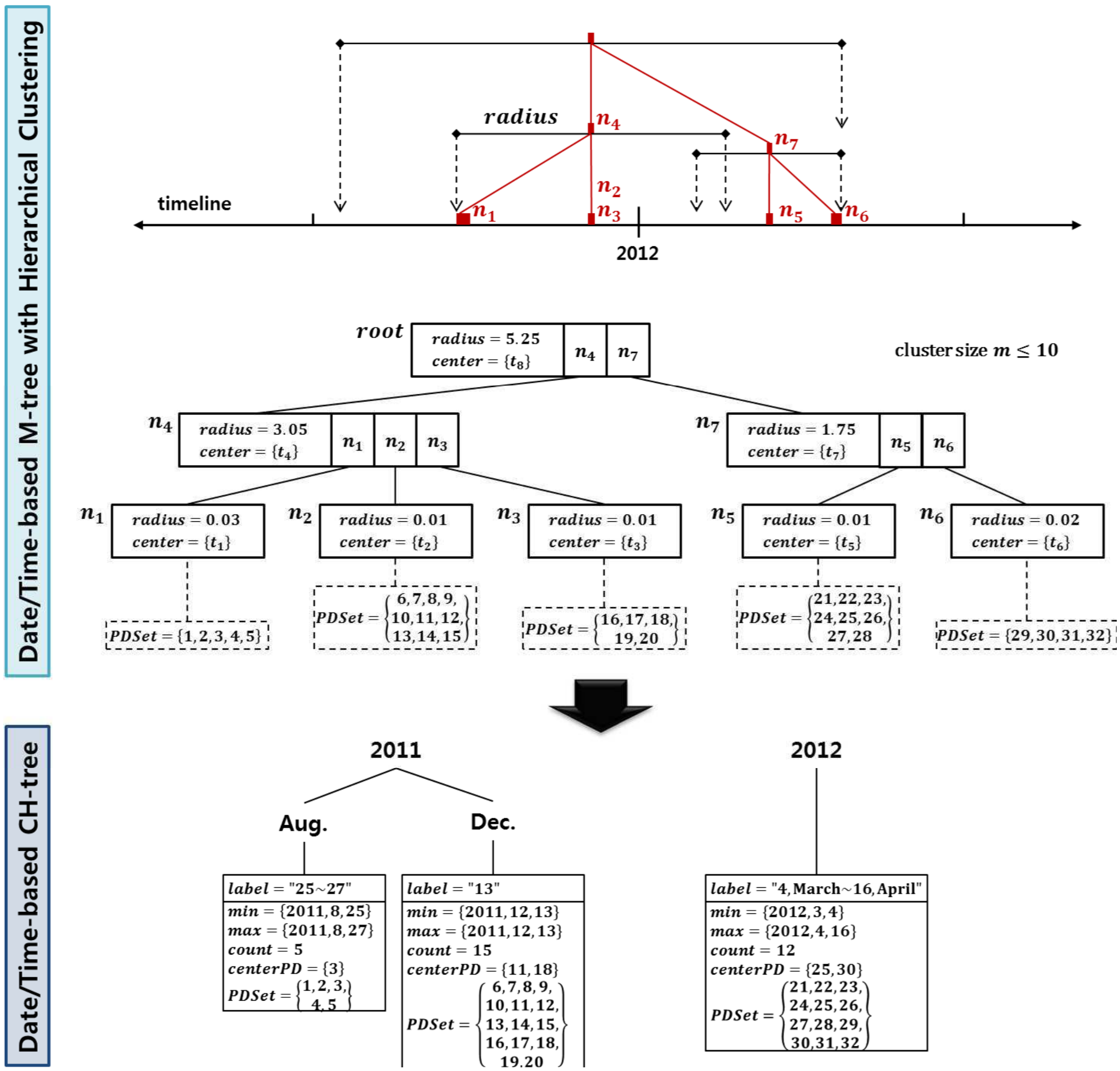


Fig. 9: Screen Shots of Multidimensional Hierarchical Explorer

the photo database of Section 3, as cube dimensions. As the measure of cells, we use the number of photos, the set of photo IDs $(PD_1, PD_2, \ldots, PD_m)$, and the information of content-based photo clusters (i.e., centroids and sizes of clusters).

We provide various functions for efficient explorations over photos based on a multidimensional cube view. A user can efficiently search photos in a various way of combining dimensions to the photo cube, for example, by date/time, by address, or by both date/time and addresses. Because date/time and address have hierarchical structures, furthermore, we can use drill-down or roll-up operations through the hierarchies to browse photos in the detailed levels or higher levels. For example, let us assume that a user first browses a group of photos by years, but s/he may want to focus on a specific year (e.g., 2012). Then, s/he can browse photos of that year through drill-down operations, or vice versa through roll-up operations.

In the proposed photo cube, we use the following basic OLAP operators to explore photos efficiently.

–**Drill-down**: stepwise refinement for address or date/time, which have hierarchical structures, from higher (summarized) levels to lower (detailed) levels (e.g., year → month → day → hour)

Table 2: An Example of Event Table

EID	Title	StartDate	EndDate	Description
1	”EDB2011”	2011-08-25	2011-08-27	”The Third International Conference on Emerging Databases”
2	”CSN2011”	2011-12-12	2011-12-14	”The First International Workshop on Cloud and Social Networking”
3	”My birthday party”	2012-03-04	2012-03-04	
4	”DASFAA2012”	2012-04-15	2012-04-18	”The 17th International Conference on Database Systems for Advanced Applications”
...	...	...	...	...

Table 3: An Example of Place Table

PlaceID	Place
1	”Songdo Park Hotel”
...	...
8	”UTS”
...	...
15	”Kangwon National University”
...	...
22	”Novotel Ambassador Busan Hotel”
...	...

Table 4: An Example of Weather Table

WeatherID	Condition
1	”cloudy”
2	”sunny”
3	”rainy”
4	”windy”

– **Roll-up**: stepwise abstraction for address or date/time with hierarchies from lower levels to higher levels (e.g., hour $\rightarrow$ day $\rightarrow$ month $\rightarrow$ year)

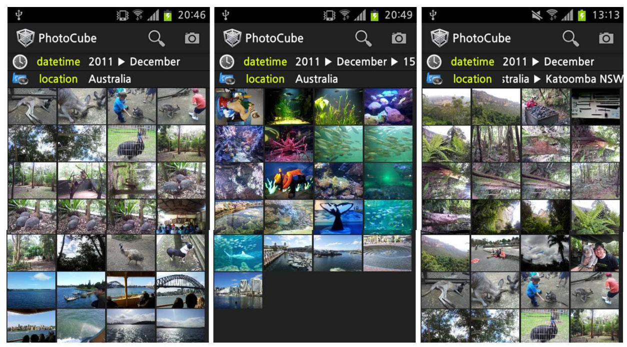
– **Slice**: selection of a subset of photos for one or more members of a dimension having specific values (e.g., year = 2011)

– **Dice**: slice operations for two dimensions (e.g., date/time=2012, address = ”Australia”)

– **Drill-Through**: exploration of photos by accessing all the detailed data of a selected cube cell.

4.2 Multidimensional hierarchical browser

The proposed system provides a novel scheme of browsing photos through OLAP operations. For the hierarchical dimensions, it also provides hierarchical browsing of roll-up and drill-down operations. Clustering photos first and browsing them based on those clusters are already widely used in many existing browsing applications. Thus, the proposed system basically includes those existing browsing techniques and additionally supports a user-friendly OLAP-style browsing over photos.



(a) (b) (c)

Fig. 10: Screen Shots of Multidimensional Hierarchical Explorer

Multidimensional hierarchical browser explores photos multi-dimensionally by exploiting photo OLAP operators over the photo cube. It uses roll-up/drill-down operators for exploring photos in detailed or abstraction levels and slice/dice operators for exploring a subset of photos. Fig. 10(a) shows an example of dicing result for two dimensions, *date/time*=$\{$*2011*, *Dec.*$\}$ and *address*=$\{$*Australia*$\}$, and Fig. 10(b) shows its consecutive result obtained by drilling-down by *date/time*=$\{$*2011*, *Dec.*, *15th*$\}$. Fig. 10(c) shows the result for the drill-down of Fig. 10(a) by *address*=$\{$*Australia*, *Katoomba NSW*$\}$.

4.3 Photo keyword search

Besides multi-dimensional hierarchical browsing, we try to fully utilize various attributes of the photo database explained in Section 3 for the fruitful exploration. For this purpose, we provide the keyword search function for naïve users to easily query the photo database. The keyword search is a representative method for retrieving the information from text documents. Likewise, traditional keyword search has targeted on text documents, but several recent research efforts [30, 31, 32, 33] have also focused on the keyword search over the database tables. Furthermore, several latest DBMSs support full-text search functions over database tables [32, 33]. Based on these efforts, the proposed system provides the keyword search function for the photo database, which we call it the photo keyword search. For example, if a user inputs ”beach”, our photo keyword search retrieves all photos having the keyword ”beach” in addresses, places, or events.

In this paper, we implement the keyword search function over photos by using SQLite DBMS [28] which runs on mobile devices. SQLite builds a keyword index for its virtual table with a segmented B-tree structure and it permits a keyword search operator, called MATCH, taking advantage of the keyword index. Based on the SQLite features, the proposed system maintains the Photo database as virtual tables. When a user inputs keyword queries into the system, it generates SQL statements for looking up the virtual tables according to the given keywords. After running the SQL statements, the system gets the photos matched to the keywords and it displays the resulting photo images on a user's smartphone screens.

4.3.1 Context Virtual Table

Fig. 11 shows the photo context table and the virtual table for photo keyword search. The context table has Photo ID as its primary key and stores the attributes for the shooting time, the place names around the shooting spot, the relevant personal events, and the weather at the shooting time. It maintains the mailing address consisting of Country, Area, Locality, Feature as well. The virtual table is stored with several internal tables and an index supporting efficiently full-text search over the table by using SQLite.

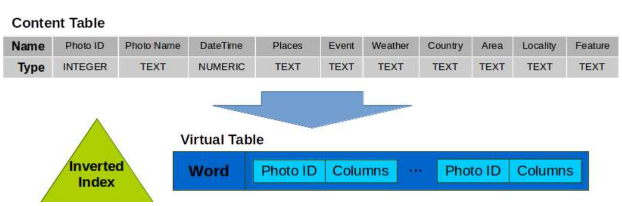
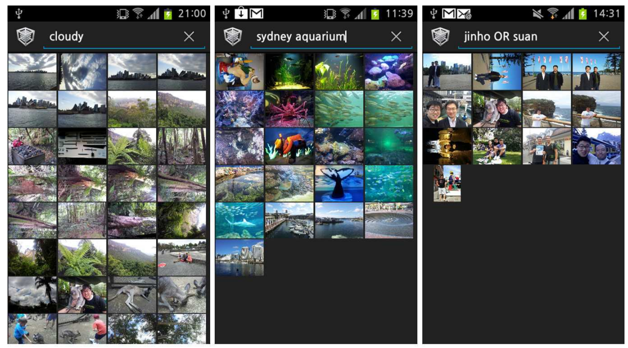


Fig. 11: Content Virtual Table and Inverted Index

4.3.2 Mapping Keyword to SQL

In order to look up the photos matching with users' keywords (or boolean expressions of keywords), it searches the Content table. For doing this, the system translates the users' keywords into SQL queries for the Content table. Note that the queries have to use MATCH operator of SQLite rather than the string comparison operator LIKE of other DBMSs so that it makes the best use of full-text index. When a user enters the keyword ”Sydney” to search the photos shot in Sydney, Australia, for example, the system translates the keyword to SQL query ”SELECT PhotoID FROM Content MATCH 'Sydney”' and then it executes the SQL query in SQLite.



(a) (b) (c)

Fig. 12: Screen Shots of Photo Keyword Search

Fig. 12 shows some example screenshots which describe the keyword search feature of the proposed system. Fig. 12(a) shows an example of the keyword search result for the keyword ”Cloudy”, which returns the photos taken during cloudy days. Fig. 12(b) shows the result for two keywords of ”Sydney aquarium” and Fig. 12(c) the result for the keywords of person's names (i.e., Jinho or Suan). With this Photo keyword search feature, users will be able to search easily lots of photos within their smartphones, just like searching web pages in the Internet.

5 Experimental Evaluation

We evaluate the performance of the Photo Cube system with a Samsung Galaxy S smartphone (made by Samsung Electronics Co., Ltd., Korea) [34], using Android 4.0.3 as its operating system [35]. We extract the content of photos using OpenCV 2.3.1 library. We took 1,000 photos whose resolution is 2560px $\times$ 1920px and we measured (1) metadata extraction and storage time, and (2) clustering and indexing time of photo and its accuracy.

5.1 Photo metadata extraction and storage

The Photo Cube system gathers several kinds of metadata: GPS, date/time, address, weather, place, event,

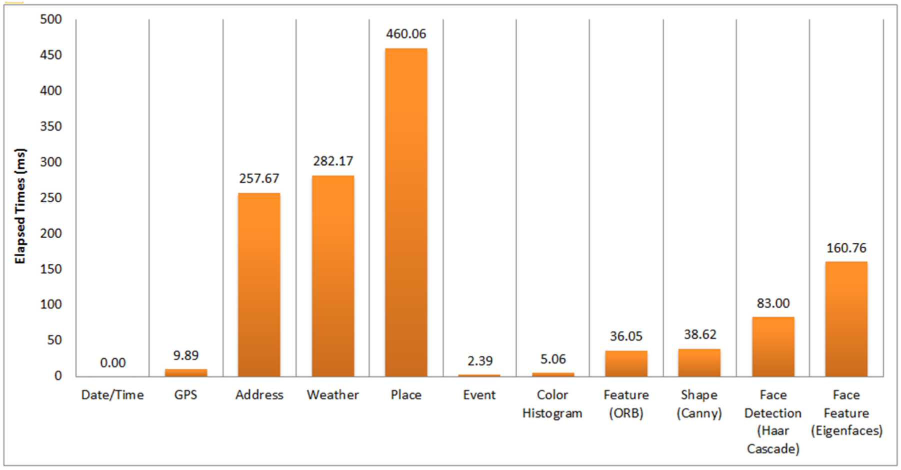


Fig. 13: Photo Metadata Extraction and Parsing Time

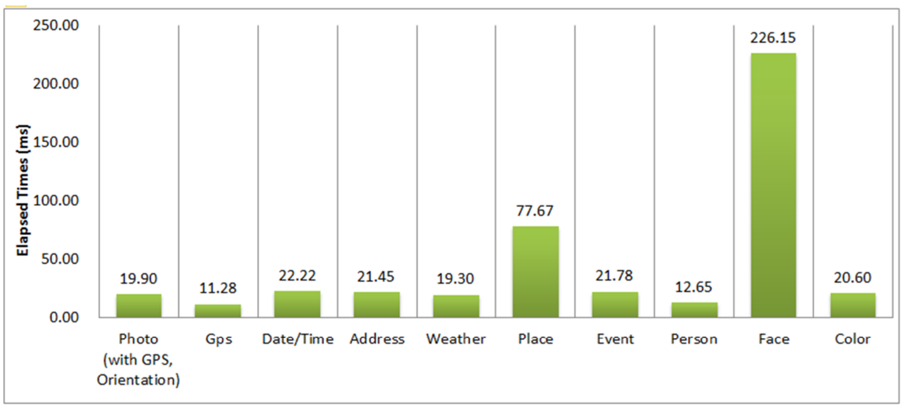


Fig. 14: Photo Database Storage Time

face, color, shape, and feature. Fig. 13 shows the execution time to get those metadata for a photo. The elapsed time is approximately 1.3 second and it seems quite long. It is because the Photo Cube has to access remote sites through the Internet to get the metadata. But the mailing address, weather, place, event, color, shape, feature, and face metadata can be extracted in offline after taking a set of photos. The essential part of the metadata which has to be extracted in real time is the date/time and GPS metadata, and it takes only about 10ms.

Fig. 14 shows the time for storing both a photo itself (which is not a part of the Photo Cube) and its metadata. When storing date/time and GPS of the metadata, it takes 38.22ms, while storing all metadata takes 182ms. If a user needs fast response time, s/he may choose to store some part of metadata in offline. As shown in the result, place, face, and face training need more storage time than other tables. It is because they insert multiple tuples or a bigger tuple. In other words, the storage time is highly dependent on the number of tuples and the size of tuples rather than the number of columns.

5.2 Photo clustering and index tree

For experiments of clustering and index tree of photos, we selected top 10 tourist destinations in the United States and we randomly extracted 500 GPS data within/around these destinations. Fig. 15 shows the clustering time by varying the number of photos from 50 to 500. As shown in the figure, the labeling time takes longer than the times for constructing M-tree and CH-tree, because address labels are obtained via the Internet. However, our approach labeling only centroid photos is faster than the naïve approach labeling all photos. The labeling time for each photo takes approximately 240ms.

Fig. 16 shows the accuracy of the clustering result for each level of the cluster hierarchy. Because all photos in a cluster share the same label as its centroid photo, the accuracy is not perfect but the processing time for labeling reduces 11 times rather than labeling all the photos. In the figure, Level 0, Level 1, and Level 2 represents Country, Area, and Locality units in Address-based CH-tree (and Year, Month, and Day units in Date/Time-based CH-tree), respectively. Meanwhile, Fig. 17 shows the average execution time for the color histogram-based clustering, where the number of photos varies from 200 to 1,000.

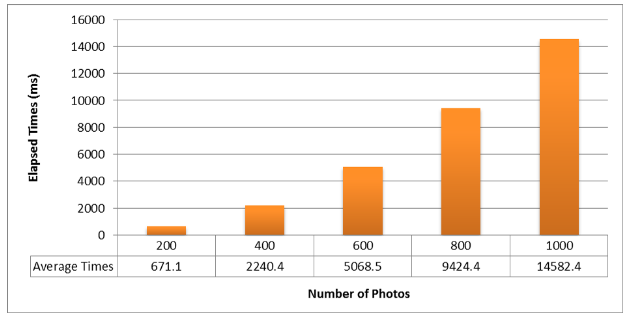


Fig. 17: CH-tree Accuracy

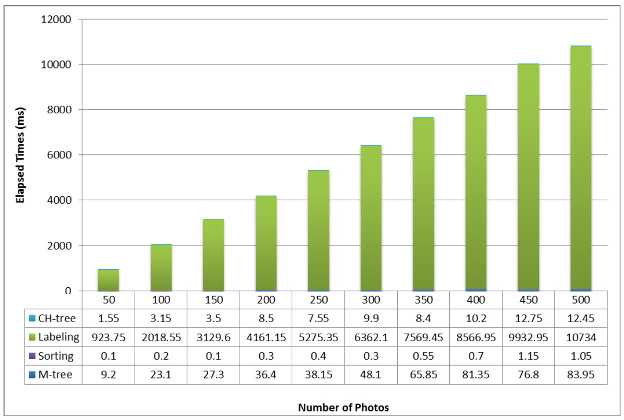


Fig. 15: M-tree and CH-tree Generation Time

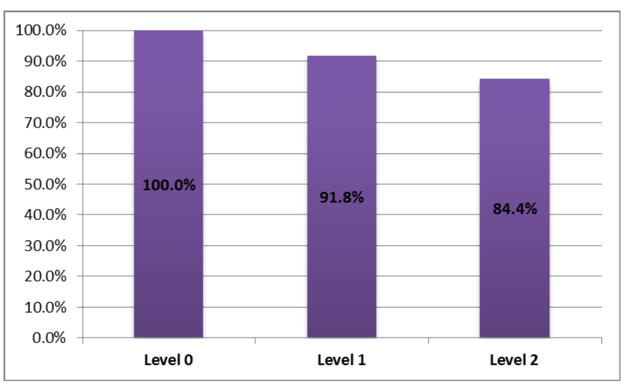


Fig. 16: CH-tree Accuracy

5.3 Photo keyword search

The Photo Cube system applied the SQLite's keyword search technique which uses virtual tables and inverted indices. In order to prove the feasibility and effectiveness of the proposed feature, we measured the execution time of photo keyword search. The Fig. 18 shows a comparison of the execution time for both using a virtual table and using an ordinary table to store the Content metadata. In the experiment, we evaluated several different SQL queries for keyword search which used LIKE, MATCH, AND, OR, or NOT operators. In Addition, we also showed the average execution time for the number of keywords varying 1 to 5. As shown in the result, the virtual table is about 3 times faster than the ordinary table for keyword search.

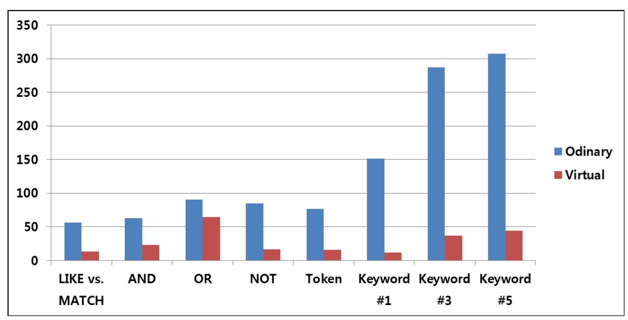


Fig. 18: Photo Keyword Search Time

5.4 Photo Apps and Services Comparative Analysis

Table 5 shows the comparison of the functions of photo Apps and services. we evaluated them on the criteria of EXIT, folder, address, place, time, event, face, color, hierarchical browsing, and keyword search features as shown in Table 5. Looking at the comparison table, the other photo Apps and services do not offer as many features as PhotoCube can do. Furthermore, PhotoCube provides these features on smartphones instead of using the abundant resource of servers. All these features could be feasible within smartphones by employing a hierarchical clustering approach and extracting photo metadata only for the centroid of clusters.

Table 5: A Comparison Table of Photo Apps and Services

Name	PhotoCube	Tidy	Picasa	Flickr	Google
EXIT	O	X	O	O	O
Folder	O	O	O	O	X
Address	O	X	X	O	X
Place	O	O	X	X	X
Time	O	O	O	O	O
Event	O	X	X	X	X
Face	O	X	X	X	O
Color	O	X	X	X	O
Hierarchy	O	X	X	X	X
Search	O	X	X	X	O

6 Related Work

There have been many research efforts on image retrieval. Datta et al. [1] provides a comprehensive survey of content-based image retrieval (CBIR) techniques. These CBIR techniques mainly utilize the low-level information of images such as color, shape, and texture characteristics which each image contains. Jia et al. [2] presents the ReMIX system which supports CBIR features from a large image database. Liu et al. [5] introduces a survey of recent technical achievements in high-level semantic-based image retrieval. Anguera et al. [6] presents a multimodal and mobile image retrieval prototype named NAMI (Multimodal Automatic Mobile Indexing) which allows users to annotate, index and search digital photos on their phones. Photo annotation is a technology for effective photo management using annotation. Chi et al. [7] develops several innovative interaction techniques for semi-automatic photo annotation which provides the following new features: (1) cluster annotation, (2) contextual re-ranking, and (3) ad-hoc annotation. Choi et al. [8] proposes a novel face annotation method that systematically combines contextual information of photos with traditional face recognition technologies. Cooray and O'Connor [9] investigates approaches to enhance person annotation in personal photo management applications. Yang et al. [10] proposes a contextual image retrieval model based on the language modeling approach. These annotation researches are closely relevant to our approach. But most of them utilizes only limited photo annotation for certain purposes.

M. Naaman et al. [3] proposed browsing places and event hierarchy for organizing digital photos automatically. They are annotated hierarchy with geographic names using the time and place. These approaches for photo browsing are similar to our system in the sense of providing user-friendly hierarchical browsing and using clustering techniques. While most of them focus only on date/time and location, our system provides users with searching for various dimensions and also with keyword search feature over the whole photo database. Qin et al. [4] extracted people, activity, and context of a photo in smartphones and used the information for automatic image tagging. K. Kim et al. [36] developed a photo browser with content-sensitive overlapping layout using lightness spatial clustering on smartphone environments. A. Gomi et al. [37] developed a photograph browser using an analysis based on location, time and person. C. Zhu et al. [38] studied multi-modality clustering for (1) the content for effective image management and search, (2) context information, and (3) prediction about the images of interest.

X. Mingjie et al. [39] studied to build a data cube and OLAP from the remote sensing images. X. Jin et al. [40] presented Visual Cube for multi-dimensional image analysis using clustering. These methods tried to use an OLAP style interface for managing photos and they handled the statistical information of images and semantics associated with many visual features. But they assume that the metadata of photos already exist thus they did not care about various inherent metadata of photos. They did not take mobile smartphone properties into account either.

7 Conclusion

Due to the recent advance of smartphones, it becomes possible for users to take massive photos and keep them inside the mobile devices. The large volume of photos makes users difficult to manage and search them thus we need tools assisting users in handling photos in smartphones easily. Several approaches have been investigated in recent systems like Picasa, Flickr, and iPhone but they only used low-level content-based information (e.g., GPS, colors or shapes) and/or utilized limited information for semantic photo information. Because recent smartphones are equipped with various sensors such as GPS, orientation, and illumination as well as networking and programming functions, we can gather diverse external semantic information about each photo from remote sites right after the photo is taken. In this paper, we developed a smartphone mobile application, called ***Photo Cube***, which automatically extracts various semantic annotation information for photos through Google Data API's and searches them easily and efficiently by taking advantage of these semantic information.

The main contributions of the Photo Cube system are as follows. First, it automatically generates abundant semantic context metadata of a photo from both basic device metadata (i.e., date/time and GPS) and image content itself (e.g., colors and shapes). Mailing address, place names (e.g., building or district name), person names appearing in the photo, personal event names (e.g., birthday party, VLDB2012 conference, etc.), and weather are the examples of the semantic metadata acquired automatically by the system. Second, in order to reduce the processing time acquiring semantic metadata, it groups photos into a set of clusters by a distant-based hierarchical clustering algorithm based on M-tree and it extracts the date/time and the address for each cluster's centroid. Third, it combines several clusters which their centroids have the same date/time or address into one. Then it automatically labels all the clusters by unique semantic information (i.e., date/time or address). The semantic labels form hierarchical structures of their components (e.g., year, month, and day), called address-based CH-tree and date/time-based CH-tree. Fourth, it creates a multidimensional cube for photo databases which is popularly used in OLAP applications to browse the summary of business data. Through the paradigm of the multidimensional cube, it allows users to search photos over various dimensions and to browse them in top-down manner along with dimension hierarchies. Interestingly, the cube's dimensions of CH-trees are constructed through clustering thus their dimension hierarchies are determined dynamically rather than fixed in prior. Finally, it also incorporates a keyword search feature over photo databases. Users can search very easily by one or more keywords for all the attributes of text type such as place, person, personal event, and weather as well as date/time and address.

We implemented the Photo Cube system on smartphones using Google's Android and we evaluated the performance by extensive experiments. From the experimental results, we found that the system gathers semantic information within a reasonable time and provides various searching features efficiently. Therefore, it can be feasible and effective for users to manage and to browse photos in their own smartphones.

For the further works, we are going to integrate more content-based retrieval features such as image shapes and features in addition to colors. With the features, we will be able to extract additional information from the photos which don't include any person but landscapes and other objects. We are also going to extend the Photo Cube system to upload (and synchronize) photos taken by smartphones into a server. Users can enjoy the same features also in world-wide web sites and they can share their photos with many others.

Acknowledgement

This research work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2011-0011824) and supported by the Research Grant from Kangwon National University.

References

[1] R. Datta, D. Joshi, J. Li, James, and Z. Wang, "Image Retrieval Ideas, Influences, and Trends of the New Age," *ACM Computing Surveys*, vol. 40, no. 2, pp.51-60, Apr. 2008.

[2] Z. Jia, L. Amselag, and P. Gros, "Content-based Image Retrieval from a Large Image Database," *Journal Pattern Recognition*, vol. 41, no. 5, pp.1479-1495, May 2008.

[3] M. Naaman, Y. J. Song, A. Paepcke, and H. Garcia-Molina, "Automatic Organization for Digital Photographs with Geographic Coordinates," In *Proc. of the 4th ACM/IEEE-CS Joint Conf. on Digital Libraries*, Tucson, AZ, pp. 53-62, Jun. 2004.

[4] C. Qin, X. Bao, R. R. Choudhury, and S. Nelakuditi, "TagSense: a Smartphone-based Approach to Automatic Image Tagging," In *Proc. the 9th Int'l Conf. on Mobile Systems, Applications, and Services*, Washington, DC, pp.1-14, Jun. 2011.

[5] Y. Liu, D. Zhang, G. Lu, and W-Y. Ma, "A Survey of Content-based Image Retrieval with High-level Semantics," *Journal Pattern Recognition*, vol. 40, Issue 1, pp.262-282, Jan. 2007.

[6] X. Anguera, J. Xu, and N. Oliver, "Multimodal Photo Annotation and Retrieval on a Mobile Phone," In *Proc. the 1st ACM Int'l Conf. on Multimedia Information Retrieval*, Vancouver, Canada, pp.188-194, Oct. 2008.

[7] J. Chi, F. Wen, R. Xiao, Y. Tian, and X. Tang, "EasyAlbum: an Interactive Photo Annotation System Based on Face Clustering and Re-ranking," In Proc. of the Int'l Conf. on Human Factors in Computing Systems, San Jose, CA, pp.367-376, Apr. 2007.

[8] J. Y. Choi, S. Yang, Y. M. Ro, and K. N. Plataniotis, "Face Annotation for Personal Photos using Context-assisted Face Recognition," In Proc. the 1st ACM Int'l Conf. on Multimedia Information Retrieval, Vancouver, Canada, pp.44-51, Oct. 2008.

[9] S. H. Cooray and N. E. O'Connor, "Enhancing Person Annotation for Personal Photo Management Applications," In Proc. the 20th Int'l Workshop on Database and Expert Systems Application, Linz, Austria, pp.251-255, Aug. 2009.

[10] L. Yang, B. Geng, A. Hanjalic, and X-S. Hua, "Contextual Image Retrieval Model," In Proc. the ACM Int'l Conf. on Image and Video Retrieval, Xi'an, China, pp.406-413, Jul. 2010.

[11] X. Zhang, Z. Huang, H. T. Shen, Y. Yang, and Z. Li, "Automatic Tagging by Exploring Tag Information Capability and Correlation," World Wide Web, vol.15, Issue 3, pp.233-256, May 2012.

[12] X Exchangeable Image File Format, http://en.wikipedia.org/wiki/\Exchangeable_image_file_format.

[13] J. Kim, S. Lee, J. Won, and Y. S. Moon, "Photo Cube: An Automatic Management and Search for Photos Using Mobile Smartphones," In Proc. of the IEEE 9th Conf. on Dependable, Automatic and Secure Computing, Sydney, Australia, pp.1228-1234, Dec. 2011.

[14] S. Lee, J. Kim, J. Won, N. Kim, J. Kang, and S. Jo, "An Efficient Keyword Search over Photo Images within Mobile

Smartphones," In Proc. of the 3rd Int'l Conf. on Cloud and Green Computing, Karlsruhe, German, pp.558-560, Sept. 2013.
[15] S-GPS, http://en.wikipedia.org/wiki/S-GPS.
[16] A-GPS, http://en.wikipedia.org/wiki/Assisted_GPS.
[17] OpenCV, http://opencv.org.
[18] OpenCV calcHist, http://docs.opencv.org/modules/imgproc/doc/histograms.html.
[19] Canny edge detector, http://en.wikipedia.org/wiki/Canny_edge_detector.
[20] ORB, http://www.willowgarage.com/sites/default/files/orb_final.pdf.
[21] Haar Cascade, http://opencv.willowgarage.com/wiki/FaceDetection.
[22] EigenFaces, http://opencv.willowgarage.com/wiki/FaceRecognition.
[23] Google Geocoding API, https://developers.google.com/maps/\documentation/geocoding/.
[24] Google Places API, https://developers.google.com/maps/\documentation/places/.
[25] Google Weather API, http://www.google.com/ig/api?weather=seoul
[26] Google Calendar API, https://developers.google.com/\google-apps/calendar/.
[27] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," In Proc. of the 23rd Int'l Conf. on Very Large Data Bases, Athens, Greece, pp. 426-435, Aug. 1997.
[28] SQLite, http://www.sqlite.org.
[29] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub-Totals," Data Mining Knowledge Discovery, vol. 1, Issue 1, Mar. 1997.
[30] B. Aditya, G. Bhalotia, S. Chakrabarti, A. Hulgeri, C. Nakhe, P. Parag, and S. Sudarshan, "BANKS: Browsing and Keyword Searching in Relational Databases," In Proc. of the 28th Int'l Conf. on Very Large Data Bases, Hong Kong, China, pp. 1083-1086, Aug. 2002.
[31] Z. Yu, X. Yu, and Y. Liu, "Cascading Top-k Keyword Search over Relational Databases," In Proc. of the ACM 14th Int'l Workshop on Data Warehousing and OLAP, Glasgow, United Kingdom, pp. 95-100, Oct. 2011.
[32] F. Liu, C. Yu, W. Meng, and A. Chowdhury, "Effective Keyword Search in Relational Databases," In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, Chicago, IL, pp. 563-574, Jun. 2006.
[33] S. Bergamaschi, E. Domnori, F. Guerra, R. T. Lado, and Y. Velegrakis, "Keyword Search over Relational Databases: a Metadata Approach," In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, Athens, Greece, IL, pp. 565-576, Jun. 2011.
[34] Samsung Galaxy S, http://www.samsung.com/global/microsite/\galaxys/index_2.html.
[35] Android, http://www.android.com.
[36] K. Kim, S. Kim, and H.-G. Cho, "A Compact Photo Browser for Smartphone Imaging System with Content-Sensitive Overlapping Layout," In Proc. of the 6th Int'l Conf. on Ubiquitous Information Management and Communication, Kuala Lumpur, Malaysia, no. 25, Feb. 2012.
[37] A. Gomi and T. Itoh, "A Personal Photograph Browser for Life Log Analysis Based on Location, Time, and Person," In Proc. of the ACM Symposium on Applied Computing, TaiChung, Taiwan, pp. 1245-1251, Mar. 2011.
[38] C. Zhu, K. Li, Q. Lv, L. Shang, and R. P. Dick, "iScope: Personalized Multi-Modality Image Search for Mobile Devices," In Proc. of the 7th Int'l Conf. on Mobile Systems, Applications, and Services, Krakow, Poland, pp. 277-290, Jun. 2009.
[39] X. Mingjie, "Experiments on Remote Sensing Image Cube and its OLAP," In IEEE Int'l Geoscience and Remote Sensing Symposium, vol. 7, pp. 4398-4401, Sep. 2004.
[40] X. Jin, J. Han, L. Cao, J. Luo, B. Ding, and C. X. Lin, "Visual Cube and On-Line Analytical Processing of Images," In Proc. the 19th ACM Int'l Conf. on Information and Knowledge Management, pp.849-858, Oct. 2010.S

Suan Lee received the B.S. and M.S. degrees in Computer Science from Kangwon National University, Korea in 2008 and 2010, respectively. Now he is a Ph. D candidate in Computer Science at Kangwon National University and he works as a research engineer for Altibase, Korea. His research interests include in-memory database system, real-time data analysis, complex event processing, OLAP & data warehouse, and data mining.

Sunhwa Jo received the B.S. and M.S. degrees in Computer Science from Kangwon National University in 2000 and 2005, respectively. At present, she is a Ph. D student at Kangwon National University, Korea. Her research interests include OLAP & data warehouse, data mining, and in-memory database system.

Jiseop Won received the B.S. degree in Computer Science from Kangwon National University in 2013. At present, he is a graduate student at Kangwon National University, Korea. His research interests include distributed and parallel computing, query processing, and mobile platform.

Jinho Kim He received B.S.(1982) in Electrical Engineering from Kyungpook National University and M.S.(1985) and Ph.D.(1990) degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) respectively. From 1990, he joined at the Department of Computer Science, Kangwon National University. Now he is a full professor at Kangwon National University. He was a visiting scholar at the University of Michgian in 1995 to 1996, at the Drexel Univeristy in 2003 to 2004, and at Georgia Tech in 2009 to 2010, respectively. His research interests include data warehousing and OLAP, data mining, main-memory database systems, new database technology for modern hardware, and analytics for web and social media. He is a member of the IEEE and a member of the ACM.

Yang-Sae Moon He received B.S.(1991), M.S.(1993), and Ph.D.(2001) degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST). From 1993 to 1997, he was a research engineer in Hyundai Syscomm, Inc., where he participated in developing 2G and 3G mobile communication systems. From 2002 to 2005, he was a technical director in Infravalley, Inc., where he participated in planning, designing, and developing CDMA and W-CDMA mobile network services and systems. He is currently a full professor at Kangwon National University. He was a visiting scholar at Purdue University in 2008 to 2009. His research interests include data mining, knowledge discovery, storage systems, access methods, multimedia information retrieval, mobile/wireless communication systems, and network communication systems. He is a member of the IEEE, a member of the ACM, and a member of the IEICE.