BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LVIII No. 2/2006	73 - 78	Seria Matematică - Informatică - Fizică

Querying and Information Retrieval in Multimedia Databases

Cătălina Negoiță, Monica Vlădoiu

Universitatea Petrol-Gaze din Ploieşti, Bd. Bucureşti 39, Ploieşti, Catedra de Informatică
e-mail: mvladoiu@upg-ploiesti.ro

Abstract

Multimedia database management systems can be seen as storage and retrieval systems, in which large volumes of media objects are created, indexed, modified, searched and retrieved. There are two types of retrievals: navigation through database to locate the desired data and database querying that finds the desired data associatively, attribute-based, or content-based. Query processing in multimedia databases has the following stages: query specification and refining, query processing (compiling, optimizing, and executing), generation of relevancy ranked results, and relevance feedback. In content-based retrieval systems, the collections of multimedia objects are stored as digitized representations. The queries are characterized by fuzziness and ambiguity. Therefore, the user will not get from the beginning what s/he wants. Metrics (precision, recall) and indexation and content-based retrieval techniques can help, but the user is the one who will have the last word in appreciation of a query results.

Key words: *querying multimedia data, query processing, content-based retrieval techniques*

Introduction

Multimedia database management systems can be seen as storage and retrieval systems, in which large volumes of media objects are created, indexed, modified, searched and retrieved. There are two types of retrievals: navigation through database to locate the desired data and database querying that finds the desired data associatively, attribute-based, or content-based. Query processing in multimedia databases has the following stages:

- query specification and refining;
- query processing which consists of compiling, optimizing, and executing;
- generation of relevancy ranked results;
- relevance feedback.

In content-based retrieval systems, the collections of multimedia objects are stored as digitized representations. The queries are characterized by fuzziness and ambiguity. Therefore, the user will not get from the beginning what s/he wants. Metrics (precision, recall) and indexation and content-based retrieval techniques can help, but the user is the one who will have the last word in appreciation of a query results.

Query Processing

In classical database management systems, the search criteria are precise and attribute-based. The search and retrieval cycle is clear and simple: the user addresses a request and gets the results from a table. Even though incertitude can be included in queries, generally the query-addressing manner is direct. In multimedia databases, the most interesting queries are content-based, so they need to be more interactive and iterative than the classical ones. Query processing in multimedia databases has the following stages: query specification and refining, query processing (compiling, optimizing, and executing), generation of relevancy ranked results, and relevance feedback.

Query specification. The query language, which is media data specific, must be extended to manipulate fuzzy search predicates, to carry out proximity searches, nearest neighbor searches, to perform content-based searches etc. The query specification in multimedia database management systems has four aspects that are not relevant in classical systems: visual queries by examples, choice ranges (the user can select a color, a texture etc.), weights and incertitude (e.g. the use of the attribute "similar with", where the argument can be a complex image).

Query processing. Similarly with conventional systems, a query is compiled, optimized, and the execution tree is generated to get the results. The optimization is much more complicated if we consider weights, fuzzy terms or content-based retrieval. Moreover, it has to be approached globally, i. e. both for content-based searches and for associative searches.

Return of results. This step considers that query term weights and object attribute values have been used to calculate the relevancy of several results and to present them to the user relevancy ranked. The results are ought both to be listed quickly enough and to have a good quality.

Relevancy feedback. In traditional systems, when the user gets the query result, the querying ends. In more advanced retrieval systems, user reaction versus the data set is considered, iteratively, until the user is satisfied with the results. Concretely, the system will add or delete terms to the predicates, will re-weight the initial predicate terms in function of ignoring or selecting some objects from the query results. The user can even modify explicitly the query. These refinements of initial query are necessary due to the complexity of the object space and to the ambiguity of objects' representation, in order to locate accurately the searched objects.

The extraction of the descriptive attributes, respectively the recognition of objects in multimedia databases can be done manually (the user enters and indexes several properties and terms), automatically (by using pattern recognition algorithms or various calculations, such as color distribution histogram within an image), or in a hybrid manner (firstly, the system determines some values, and secondly, the user can correct or extend the detected values). These attributes/terms can be at physical level (representation, e. g. as HSM), at multimedia elements' level (graphics, video etc.), or to an application specific level (CAD/CAM object types, GIS types, images of some persons etc.).

Query Types

Multi-dimensional tries structures are similar to k-d trees, except that they divide object space. Each time when a partitioning is produced, a hypercube 2-split is produced as well, by selecting a dividing attribute. Due to the data independence of these structures, they are often unbalanced. As a result, the response time can be longer for some queries. The search mechanism is basically the one from the k-d trees, only the space partitioning is unequal. In many multimedia applications, the search can be considered as localization of desired objects in an nD space. Each of the dimensions can correspond to an attribute of media objects, which can be a physical attribute or a content term. Every multimedia object type will have its own attribute set, to which application specific attributes are added. Furthermore, the complex object class needs to be considered. It involves is-a relationships and attributes that can be atomic or compound (tuple, set, list). There are many kinds of queries for multi-key retrieval:

- *exact match*: all the values of the attributes that in the multi-key index are specified;
- *partial match* when just a sub-set of attributes is specified;
- *range* – in these queries a range for attribute values is given. They can be exact for some attributes and range-based for others;
- *partial range*, in which the ranges are indicated only for an attribute sub-set.

Another type of query is the best match, which can be expressed as a range or partial range query, and corresponds to an exact or partial specification that is combined with a distance function. These queries are known also as nearest neighborhood searches. They are especially useful in multimedia databases to address the fuzziness issues and to get the similar objects.

Content-based Retrieval

In content-based retrieval systems, the collections of multimedia objects are stored as digitized representations. The queries are characterized by fuzziness and ambiguity. Therefore, the user will not get from the beginning what s/he wants. There are two kinds of errors: *false results*, which are returned as query solution, and *absent results* that should be returned, but they are not in the query response. In information retrieval systems, to measure the success of a query two metrics are used:

- *precision*, which defines the ratio between the number of returned relevant objects and the total number of objects that are returned as query results;
- *recall*, which is the ration between the number of returned relevant objects and the total number of relevant objects in the collection.

Both metrics take values in range [0,1]. The aim is that they are closer to 1, but usually a compromise is accepted. The factors that influence them depend on the indexing scheme and on the application. Anyway, both values have only theoretical values, since the user is the one who will have the last word in appreciation of a query results.

Content-based Retrieval for Images

Informally speaking, the content of an image consists of each object of that image that is of interest from application viewpoint. These have various associated properties such as a descriptor of the form/location in which the object is found within the image and a descriptor of features, which expresses the characteristics of pixel or group of pixels (e. g. RGB values).

To be computerized, the images are split in mxn equal-sized rectangular cells that form an image grid. They are split furthermore, in homogeneous regions that are called segments. The segmentation process is presented further on. A connex region $\Re$ is a cell set, such that if $(x_1,y_1)\in\Re$ and $(x_2,y_2)\in\Re$, then it exists a cell sequence C_1, C_2, … , C_n in $\Re$ and $C_1=(x_1,y_1)$, $C_n=(x_2,y_2)$ and Euclidian distance between C_i and C_{i+1} is 1, for i ranging between 1 and n. A homogeneity predicate that associates with an image is a function H that gets a connex region $\Re$ from the image and returns true or false (for example, H is true is more than 100xδ% from the cells in the respective region have the same color - $\delta\in[0,1]$). An image segmentation according with the predicate H is defined as being a set of regions R_1, …, R_k, so that:

$R_i \cap R_j = 0$ for any $1\le i \ne j \le$ *and* $I=R_1 \cup \ldots \cup R_k$

$H(R_i)$ = true, for any $1\le i \le k$, *and*

for any $1 \le i \ne j \le k$, if $R_i \cup R_j$ is a connex region, $H(R_i \cup R_j)$=false.

In general, the reasoning that refers to an image does not consider all the pixels because their number is prohibitively high. A common approach is to transform the image matrix (or the segment's one) in a compressed representation. The most used techniques are *Discrete Fourier Transform* (*DFT*), *Discrete Cosine Transform* (*DCT) and Wavelet*.

A question to ask often about images is if two images or segments are similar. There are two approaches of similarity-based retrieval between two images/segments: *metrical* (distance function based) and *transformational* (transformation cost based).

In current databases, images are represented either as relationships, either as spatial data structures (for examples with generalized R trees), or with help of image transformations. Generalized R trees are similar with R trees, except that they contain a set of *Generalized Bounding Rectangles (GBR)*, which are represented by 2x(n+2) fields that correspond to the lower/upper bounds for each dimension.

Content-based Retrieval for Text

The most recent and efficient approach to access databases that contain text is Latent Semantic Indexing (LSI). Concretely, LSI associates with each document a limited-size vector vec(*d*) that contains frequency terms. Thus, the storage of documents in database becomes equivalent with storing the associated vectors. The basic idea of the technique is that similar documents have similar frequencies of words. The technique allows both the elimination of words and phrases that do not allow distinguishing between various documents and the identification of the ones that do so. It can identify also the similar words.

The most important problem to which LSI have to cope it is the large number of both documents M and terms N. It solves this by using the single value decomposition (SVD), which reduces the dimensions of frequency matrix (table). The reduction determines loss of information so SVD must ensure that only the least significant parts of the matrix are lost. Usually, the reduction is done at 200x200 for large collection of documents.

Content-based Retrieval for Video

The content of video can be seen as a tuple from the set of objects of interest within video, the set of activities, which are significant, and an application to show that are the objects and applications that are associated with each video frame. A video is a scene sequence, which are composed by shots that are frame sequences. A frame is a static image, a shot is a clip that presents a continuous action in time and space, and a scene and a sequence of shots with common semantics. A video library is a collection of 5-entry tuple: the identifier of video record, its content, the number of frames, a set of relations about the whole video and a placement application (which shows where are physically located the various parts of the record). Main queries refer segments, objects, activities and several features that appear in a video. To manipulate a video it will be divided logically in homogeneous segments. The video segmentation techniques get the record and determine where the shots have been concatenated, or spatially/chromatically composed. After segmentation the videos are presented compactly as FS or RS trees.

The basic idea behind FS trees (FS - frame-segment) is simple: firstly a table is created to associate with each object or activity a set of segments in which that object/activity appears in. The FS tree associates each node with the range in which the frames from the sub-tree that is dominated by that node are found. The indexing method associates each range with two tables, one with objects and one with activities, each of the elements of these tables is an ordered list of pointers to the nodes from the FS tree, which contain the respective objects/activities.

The RS trees are similar to FS trees with one major exception. The concepts of object vector and activity vector remain the same, but the (start, end)-frames, which are actually rectangles of e-s length and width zero, are stored instead in an R tree. That will be extended to show which are the objects/activities for each rectangle. The RS tree's advantage is that on each disk access they bring into memory more than one rectangle, those being in fact proximate rectangles.

Content-based Retrieval for Audio

As for images or videos, audio data can be characterized in two ways: by using metadata to explain the content (objects or activities) or by extracting specific features with signal processing techniques (for instance, frequency, amplitude, vibratory period etc.) The most common content-based indexing technique for audio is to segment the signal in time, to get small windows in which it can be considered homogeneous (amplitude, speed and wave length are constant). The segmentation is made with one constant step (window length) or by using a homogeneity predicate.

One the segmentation has been made, the audio signal can be seen as a window sequence w_1, w_2, …, w_n. For each significant window, k relevant features are extracted. Thus we get n points in a (k+3)-dimensional space (audio source file, the window and its duration add to the k features of the signal). Obviously, this approach is unrealistic since only ten minutes can produce 100.000 windows. Therefore, adapted compression techniques (DFT, DCT) have to be applied in this case too. A TV tree (Telescopic Vector tree) is used for indexing. These are similar to R trees, but they try to decide dynamically and flexible how to branch based on the data under examination – thus the terms that distinguish better between documents will be chosen. When the user addresses a query like: find every audio which have similar sound, a DFT will be applied to the query sound, then a search of the nearest neighborhood will be done.

Conclusions

Most multimedia data must be seen as being n-dimensional. Such data require special accessing and indexing techniques. The need for performance in information retrieval process imposes the use of multi-dimensional indices (k-d trees, tries, grid files, point quad trees, multi-key hash tables, MX-quad, R trees etc.) and media object clustering. These indices are used for associative retrieval (attribute-based). As this kind of querying is insufficient in multimedia databases, content-based retrieval structures had to be introduced. Inverse indices are the most used such a structure. These indexes rely on both metadata and content indexing terms. Metadata play an important role in multimedia systems for several reasons: exact query paradigm is not powerful enough in the multimedia context, the reduced efficiency of direct content-based search, inappropriate processing techniques, and the semantics of media data. There is many query types for multi-key based retrieval: exact match, partial match, range, and partial range.

In content-based retrieval systems, the collections of multimedia objects are stored as digitized representations. The queries are characterized by fuzziness and ambiguity. Therefore, the user will not get from the beginning what s/he wants. There are two kinds of errors: false results, which are returned as query solution, and absent results that should be returned, but they are not in the query response. Metrics as precision and recall can help to improve the query results, but the user will have the final word to appreciate them.

References

1. Aizawa, K., Nakamura, Y. - Advances in Multimedia Information Processing, *PCM 2004 : 5th Pacific Rim Conference on Multimedia Proceedings,* in *Lecture Notes in Computer Science*, Springer; 2005
2. Candan, K.S., Celentano, A. - Advances in Multimedia Information Systems*, 11th International Workshop MIS 2005 Proceedings,* in *Lecture Notes in Computer Science series,* Springer, 2005
3. Furht, B. - *Multimedia Technologies and Applications for the 21st Century*, Kluwer Academic Publishers, 1998
4. Hirzalla, N.B., Karmouch, A. - A multimedia query specification language*,* in *Nwosu K., Thuraisingham B., Bruce Berra P., Multimedia Database Systems, Design and Implementation Strategies*, Kluwer Academic Publishers, 1996
5. Khoshafian, S. - *Multimedia and Imaging Databases*, Morgan Kaufmann, 1995
6. Lee, K., Lee, Y.K., Berra, P.B. - Management of Multi-structured Hypermedia Documents: A Data Model, Query Language, and Indexing Scheme, in *Multimedia Database Management System - Research, Issues and Future Directions, Vol. 4, No.2,* Kluwer Academic Publishers, 1997
7. Royo, J.D., Hasegawa, G. - Management of Multimedia Networks and Services, *8th International Conference on Management of Multimedia Networks and Services, MMNS 2005, Barcelona, Lecture Notes in Computer Science Publisher,* Springer; 2005
8. Subrahmanian, V.S. - *Principles of multimedia Database Systems*, Morgan Kaufmann Pub. Inc., San Francisco, CA, 1998
9. Yu, C.T., Meng, W. - *Principles of Database query processing for advanced applications,* Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1998
10. *** - Multimedia Computing and Networking 2004, *Proceedings of S P I E (International Society for Optical Engine),* California, 2004

Interogare şi regăsirea informației în bazele de date multimedia

Rezumat

Sistemele de gestiune a bazelor de date multimedia pot fi văzute ca sisteme de stocare şi căutare în care volume mari de obiecte media sunt create, indexate, modificate, căutate şi regăsite. Există două feluri de regăsiri: navigația prin baza de date pentru a localiza datele dorite şi interogarea bazei de date, care găseşte datele dorite asociativ, bazat pe atribute sau bazate pe conținut. Procesarea interogărilor în bazele de date multimedia are următoarele stadii: specificarea şi rafinarea interogărilor, procesarea interogărilor (compilarea, optimizarea, execuția), generarea de rezultate ordonate după relevanță şi feedback-ul relevanței. În sistemele de regăsire bazate pe conținut, colecțiile de obiecte multimedia sunt stocate ca reprezentări digitizate. Interogările sunt caracterizate pe incertitudine şi ambiguitate . Ca urmare, utilizatorul nu va obține de la început ceea ce doreşte. Metricile (precizie, retur) şi tehnicile de indexare şi regăsire bazată pe conținut ajută, dar utilizatorul este acela care va avea ultimul cuvînt în aprecierea rezultatelor unei interogări.