World Multiconference on Systemics, Cybernetics and Informatics

July 23-26, 2000
Orlando, Florida, USA

PROCEEDINGS

Volume I
Information Systems

Organized by IIIS

International Institute of Informatics and Systemics

Member of the International Federation of Systems Research

IFSR

Co-organized by IEEE Computer Society (Chapter: Venezuela)

EDITORS
Nagib Callaos
Mohamed Loutfi
Shogo Nishida
Dong-Ik Lee
Jesse Bemley
Marianella Aveledo

Access methods supporting content-based similarity retrieval of audio/video

Pasquale Savino
IEI-CNR
I-56126 Pisa, Italy

ABSTRACT

The paper provides an overview of the techniques used for similarity retrieval of multimedia objects. Particular emphasis is given to the problem of efficient retrieval and to the analysis of different access structures used.

Keywords: Access structures, audio/video retrieval, similarity retrieval, vector space model, metric space model, approximation techniques

1. INTRODUCTION

Content-based retrieval of multimedia information is attracting increasing interest, due to the wide diffusion of computer systems able to produce and process these type of data.

Historically, images were the first non-textual data type managed by a computer; retrieval techniques have been studied and experimented on images since the beginning of '80's. Initial approaches were either based on elementary techniques, such as those that require a textual description of image content and then basing the retrieval on the description, or on trying to support the retrieval through an interpretation of the image content. The use of image interpretation is very powerful when such a description can be provided; however, it has shown to be very ambitious in general environments and it has been successfully applied only to limited application domains. Afterwards, the increased processing power of computers and the availability of tools for the management of multimedia data lead to a significant increase of the amount of audio and video data managed by computers. The storage and retrieval of these new data types is remarkably more complex than is the case with images; in particular, new methods for the representation of their content and for their automatic indexing were needed. The retrieval of audio and video objects has also required the use of the temporal dimension in the object's representation, and query formulation and execution.

The main objective of a retrieval system is user satisfaction that, in a broad sense, is related to its capability of retrieving *all relevant* information in a *limited amount of time*. This means that the two key issues to be addressed by any retrieval system are *effectiveness* and *efficiency*.

In this paper we mainly address the problem of supporting an efficient retrieval of multimedia information; this is critical due to the *complexity* of descriptors of multimedia objects and due to the *large amount of data* contained in multimedia archives. In order to have an indication of the number of objects, we may consider that the Web contains over one billion pages, and most of them contain multimedia information. In particular, the paper addresses the problem of selecting appropriate access structures for an efficient retrieval. Related to this, we will discuss the question of whether we can use generic access structures for audio/video retrieval of if we need to define access structures specialized for this type of data.

2. RETRIEVAL OF NON TEXTUAL DATA

Retrieval of multimedia objects has, at the very general level, the same purpose and objective as the retrieval of textual data, i.e. to select a set of objects belonging to the multimedia archive, objects that satisfy user's information needs, and to disregard at the same time objects that do not correspond to these needs. The entire process is not precise, so that retrieved objects will contain some spurious results, while some relevant object will not be retrieved at all. The main problems to be addressed in order to support an effective retrieval of multimedia objects are related to (i) the definition of appropriate tools that allow the user to express his/her information needs, (ii) the possibility of using appropriate representations of the objects contained in the multimedia archive, as well as of the user's requests. and (iii) the definition of criteria to select objects that satisfy user's needs disregarding, at the same time, objects that do not address these needs. The central aspect to be taken into consideration is related to the representation used to describe the objects. An analysis of the different approaches adopted for multimedia retrieval leads us to distinguish three main categories according to the information used to index the multimedia objects. The first one uses a *keyword or text representation* of the objects, the second is based on *physical properties (features)* extracted from the

objects, and the third one uses conceptual information of the multimedia objects.

The first one is a quite straightforward approach that can use the standard access structures adopted for text retrieval. Most of the existing systems index the multimedia objects according to a certain number of physical characteristics (*feature indexing*) which could be used to discriminate among them. A typical example is color distribution in an image: an image with a dominance of certain colors is, in many cases, unrelated to images having a different set of dominant colors, and vice-versa. For example, images representing flowers may have dominant colors such are *red, yellow, green, etc.*. Then, the presence of these colors increases the expectation of the presence of flowers with respect to images that do not contain them. The physical characteristics of a multimedia object are usually represented through a *set of features*.

Features are media dependent because the physical properties of each media are different. For example, typical image properties used as features are *color, texture*, and *object shape*, while typical video properties are *motion vectors, camera position, etc.*. Features such as *signal intensity* and *pitch* can characterize an audio component. Features are also application dependent because, for particular uses, and considering objects belonging to a limited application domain, particular features can be used. For example, in a face recognition application, the features used could be the size of the eyes, their distance, the size of the nose and of the mouth, their relative distances, etc.

The similarity between multimedia objects is argued by the similarity of their features. The assumption on which the approach relies is that if two objects have similar feature values, then the objects are similar. It is obvious that this assumption cannot be verified for all objects, but is true only at a statistical level: there is a higher probability that two objects with similar features appear similar to a user than two objects with different features. An approach based on such an assumption is clearly very rough, and we can expect that the quality of retrieval will be limited by imprecision that is mainly due to the limited semantic meaning of some of the used features, which could not be significant for the user.

The object's indexing requires calculation of the value of each feature for all the objects contained in the multimedia archive M. The retrieval is performed through the following procedure: a query is represented through a multimedia object of the same type as those contained in the multimedia archive. The same features used to represent the object in the archive are extracted from the query object; then, the dissimilarity (and correspondingly the similarity) between each multimedia object and the query is computed, through a comparison of their representations. All objects in the multimedia archive M are then ordered in decreasing similarity values with the query, which corresponds to the query result.

Main phases of a generic system supporting multimedia object retrieval are: *archive population*, which allows one to insert multimedia data and to extract information about its content; *object storage*, which stores the objects and their representation and generates suitable *access structures* which support an efficient retrieval of multimedia objects, and *query formulation* and *execution*. In the following sections we will describe the main characteristics of access structures.

3. VECTOR MODEL

In this model, the assumption is that each object is represented as a point in an n-dimensional vector space V

$$x = (x_1, \ldots, x_n)$$

The dissimilarity between two generic objects $x, y \in V$, is measured as the distance between the objects as defined for the space V. This is usually given by the Euclidean distance L_2. However, the most general case is based on the use of the Minkowsky distance d_r, defined as follows:

$$d_r(x,y) = \left[\sum_{i=1}^{n} |x_i - y_i|^r \right]^{1/r}$$

that corresponds, for $r=1$ to the city-block metric, and for $r=2$ to the Euclidean distance. The similarity $sim(x,y)$ between two objects x and y can be obtained as a function of $d_r(x,y)$ ($sim(x,y)=G(d_r(x,y))$). In general terms, if we want to limit the values of $sim(x,y)$ to the [0,1] interval, we may impose the following constraints: $d_r \to \infty$ as $G(d) \to 0$ and if $d=0$ then $G(d) = 1$.

The vector model has been widely used in similarity retrieval due to its simplicity and to the properties of vector spaces. These properties have been used in the definition of efficient access structures.

Metric Space Model

In many cases, a vector representation is not adequate to represent the objects in the multimedia archive. For example, some similarity functions used to compare color distribution in an image satisfy the properties of metric spaces, but they do not comply to the properties of vector spaces.

However, even if the representation of the object is generic, the dissimilarity measure d between two generic objects x and y is defined as a distance metric function, i.e. a function having the following properties:

(i)	$d(x,y) = d(y,x)$	*simmetry*
(ii)	$0 < d(x,y) < \infty$, $x \neq y$ and $d(x,x) = 0$	*ositivity*
(iii)	$d(x,y) \leq d(x,z) + d(z,y)$	*triangle inequality*

Object representation with a distance measure that satisfies these properties is said to be compliant to a Metric Space Model (*MSM*). Examples of metric distances are the *edit distance* for text, *normalized overlap* for sets, *histogram distance* for images, and the *Hausdorf distance* for shapes. In the *MSM*, the representation of the object is not specified; this implies that it is not possible to take advantage, when the access structures are defined, of the specific representation of the objects. For example, in Vector Model the definition of access methods makes use of concepts such as splitting over one dimension, calculation of the centroid of "close" objects, etc. This cannot be done for the definition of access structures of objects belonging to the *MSM*. However, access structures have been defined for metric spaces, using only the symmetry, positivity, and triangula inequality properties. The *MSM* is a generalization of the *VSM*; indeed, a vector representation and a distance measure, based on the Minkowky distance, satisfy the positivity, symmetry, and triangular inequality properties. This implies that all access structures defined for the *MSM* can be used for the *VSM*, too. However, the access structures specifically studied for the *VSM* are expected to be much more efficient.

General models

Studies that have been conducted on the human perception of similarity, evidenced that perceptual notions of similarity do not adhere either to the vector model or to the metric space model. In particular, it has been shown that the properties of symmetry and triangular inequality are not verified. This implies that an exact evaluation of the similarity between objects in the multimedia archive and the query requires a complete scan of the archive: no general purpose access structure can be used, in order to filter out objects which are too much dissimilar with respect to the query. Generic similarity models require the adoption of ad-hoc access structures, specifically defined, when possible, for the distance measure. Another possibility consists in defining a distance function that approximates the true distance function, and that is compliant to the *MSM* or to the *VSM* models.

4. ACCESS STRUCTURES FOR SIMILARITY RETRIEVAL

Typical queries that can be expressed in order to retrieve multimedia objects are *range queries* and *k-NN queries*. They can be defined as follows:

Definition 4.1. *The range query q(r) requires the retrieval of all objects* $o \in M \mid sim(q,o) \leq r$.

Definition 4.2. *The k-NN query q(k) requires the retrieval of a set L of k objects* $(L = (o_1, \ldots, o_k))$ *such that* $L \subseteq M$ *and* $\forall o_i, o_j \mid (o_i \in M \wedge \notin L) \wedge o_j \in L$ *then* $sim(o_i, q) \leq sim(o_j, q)$.

A simple and trivial algorithm for the execution of *range* and *k-NN* queries requires a complete scan of the archive M. Such an algorithm, has a computational complexity which is linear with the cardinality of M; this may produce unacceptable response times for very large archives, especially when the cost for calculating $sim(o, q)$ is high. Thus, access structures are used to select the *k-NN* objects (or the objects within a "distance" r to q) without the need to access the entire archive containing the object's representation.
The main characteristics of access structures apt to support similarity retrieval of multimedia objects are as follows:

- *Search response time sub-linear with the dimension of the database.*
  We may measure the improvement of efficiency deriving from the use of the access structure by comparing the execution cost for a total scan of the database with the number of accesses needed when the access structure is used.
  If card(M) = N, and M is the number of database objects accessed when the access structure is used (M is also the number of computations of query/object similarity measures performed), then we define the *Index of Improvement* (*II*) as $II = M/N$.
  It is clear that a comparison between two access structures could be based on a comparison of their average values of *II*; the average is performed for different queries.
- *Support of range queries as well as k-NN queries.*
  In many applications both types of queries are needed and the access structure should improve the efficiency of both types of queries.
- *The access method should be adaptable to different types of data and similarity functions.*
  This is a particularly relevant problem. Indeed, when a new access structure is studied, it would be particularly useful if it could be adopted in many different situations. Furthermore, the behavior of the access structure is strongly dependent on the characteristics of the objects stored in the database; in particular, it strongly depends on the distribution of the distances between the objects stored in the database. The situation which is more difficult to be manage arises when the distribution of distances of objects is uniform.

Access structures for vector representations

Access structures, defined for objects that satisfy the properties of the *VSM*, are usually based on R-tree and its variants.

R-tree is a storage structure able to deal with spatial data, originally presented in [1]. In the proposal, it is assumed that an object, of arbitrary shape, can be reduced to (possibly multi-dimensional) rectangles by finding its minimum bounding rectangle. Like a B-tree structure, an R-tree is also balanced and all leaf nodes appear on the same level. An R-tree guarantees that the space utilization is at least 50%. However, dynamically built R-trees result in structures with excessive space overheads and "dead-spaces" in the nodes that may result in bad performance.

In R-trees, the concepts of *coverage* and *overlap* are important. Coverage of a level of an R-tree is defined as the total area of all the rectangles associated with nodes of that level. Overlap of a level of an R-tree is defined as the total area contained within two or more overlapping nodes. Obviously, minimization of both the overlap and the coverage is required for an efficient R-tree search.

For region data objects, zero overlap is in general not attainable. However, if we allow partitions to split rectangles, then zero overlap among intermediate node entries can be achieved. This is the main idea behind the R$^+$-tree [2]. Another variation of the R-tree, called R*-tree [3], does not change the properties of the basic structure, but rather has a more complex way of organizing rectangles into nodes so that overall performance is improved.

Access structures for metric representations

Most of the more commonly used distance and similarity measures between objects satisfy the definition of a *metric distance function*. Approaches based on so-called *metric trees* directly use pair-wise distances between objects to recursively partition the search space without considering positions of objects in a multidimensional space -- dimensionality of space need not even be known.

Several techniques, such as GNAT [4] and VP-trees [5Chi94], which basically differ in the criterion used for partitioning objects, have recently been proposed, with the common objective of designing a data structure that would exploit local features of the metric for solving similarity (proximity) queries [5].

- **VP-trees.** The basic construction of a VP-tree is to break the space up using spherical cuts. To build it, it is necessary to pick a point in the data set -- this is called the *vantage point*, hence the name VP-trees. Now, we may consider the *median* sphere centered at the vantage point with a radius such that half the remaining points fall inside and half fall outside it. For every other point it is evaluated if it is inside the sphere of if it is outside the sphere: in the first case the point is put in one branch, in the second one it is put into the other branch. The lower level branches are constructed recursively. This idea has been extended in [6] to the *n*-ary case and implemented as an indexing method for content-based image retrieval.
- **GNAT's.** The primary goal of the Geometric Near-neighbor Access Trees [7] is to have a data structure that reflects the intrinsic geometry of the underlying data. More specifically, the top node of the tree gives a brief summary of the data in a given metric space and, in descending the tree one gets a more and more accurate sense of this data geometry. The aim is achieved as a hierarchical *Dirichlet domain* (or *Voronoi diagram*) based structure -- given a number of points, $x_1, x_2, \ldots, x_k$, the Dirichlet domain of x_i consists of all possible points in the space which are closer to x_i than to any other point x_j ($j \neq i$). In order to minimize the number of distance computations while executing queries, a number of pre-computed distances is stored in the tree. However, the tree is built top-down, so the resulting structure is not very convenient for dynamic files.
- **M-tree.** An M-tree [8] can be viewed as a hierarchy of metric (ball) regions. A region is defined by a database object O_i and radius $r(O_i)$, which represents the maximum distance between O_i and any other object, including its region (if any), in the region of O_i. An M-tree is a multiway-branching tree, thus each node can contain several object entries which are all members of a region centered around a *parent object*, O_p, stored in a higher-level node. Notice that the region of objects from the root is assumed to be the entire universe, because these objects do not have any actual parent object. Each entry is represented by the object's features and, in the case of non-leave entries, by their region radii which restrict minimum regions in which all descendant objects and/or regions can be found. For efficiency reasons, *child to parent* object distances, computed during the tree construction phase, also form a part of the objects' entries.

Access structures for general representations

Current research activities try to transform new problems into other problems for which efficient implementations are already known, or to develop new access structures able to deal directly with some of the new problems.

The problem of selecting a suitable set of features to accurately represent objects is not always an easy task. An alternative approach, investigated in [9], suggests starting from an expert-defined similarity (distance) matrix, and assumes that each object is a point in a

"virtual" high-dimensional space. By means of a heuristic transformation technique, aiming to preserve pair-wise distances as much as possible, objects are mapped onto points in a lower k-dimensional space (k being user-defined), where SAMs can be used.
Presumably, any approximated distance measure must preserve the distance between objects as much as possible. It has been shown in [10] that no dismissals occur if the actual distance is *lower-bounded* by the actual distance in the distance space, more precisely:

$$d_{approximated}(O_i, O_j) \leq d_{actual}(O_i, O_j)$$

for any pair of objects from a given database.

Approximation techniques

Although there are several index structures that can support the execution of similarity queries, experience with their applications shows that the processing costs are still very high. For example, similarity retrieval in high dimensional data spaces tends to access most or even all similarity index tree nodes (sometimes designated as the *dimensionality curse*).
Another important fact is that the specification of a query object Q, needed as a reference object in similarity queries, is something users find quite difficult. This problem is easy to understand if a space of many dimensions is considered. Consequently, several queries need to be executed before a good query object is found. Initial retrieval steps are needed to find a more suitable query object. This iterative approach to query processing makes the total costs of a query execution even higher, while the need for exact answers is not always relevant.
The vagueness in query specification together with the high query execution costs led us to investigate the following idea [11]:

> *Provided an approximate similarity search can be performed much faster than the precise search, the approximate similarity retrieval can play a useful role in the global process of iterative similarity retrieval).*

In other words, since processing exact similarity queries is not always required, approximate answers would suffice, especially when obtained at much lower costs.
To investigate this idea, we have specified three approximation techniques in the environment of the M-tree. The approximations of the proposed techniques are bound, respectively, by:

- the relative distance error, ε;
- the statistically obtained fraction of the searched file best cases, ρ;
- the tangent of the expected search improvement curve, κ.

We have also defined measures to quantify the performance improvements and the quality of efficiency, and tested all of our approximation techniques by querying three different multidimensional data files.
Our results show that approximation through relative error is not very efficient. Approximation through distance distribution performs best, in fact, improvements in efficiency of even two orders of magnitude were observed for still quite high precision. However, a characteristic distance distribution is needed, and the quality of this distribution may influence the performance. If such a distribution is not available, the approximation through the slowdown of distance improvements is strongly recommended, because this method is simple, it does not need additional data, and it also performs very well. Though the techniques of approximation seem to be more efficient for smaller sets of nearest neighbors, approximation techniques also seem to perform well for skewed distance distributions, and they scale well to manage large data files.

5. CONCLUSION

The definition of appropriate access structures supporting similarity retrieval of multimedia objects is related to the mathematical properties of the used feature representation, and to the similarity measure adopted. In many cases, independently from the type of data (images, audio, video, etc.), features are represented by using a vector model with similarity measures calculated with an Euclidean distance. More complex situations are dealt using a more general model, based on metric spaces. In both cases, several access structures have been studied and are adopted in many commercial and experimental systems. The main limitation of the approaches adopted is limited performance – mainly for high dimensional spaces – which may require accessing approximately 10\% of the entire database. A research direction that is under investigation in order to limit this problem consists in accepting approximate results, yielding a significant performance improvement (in our experiments we observed an improvement of up to two orders of magnitude).

6. REFERENCES

[1] A. Guttman. "R-trees: A dynamic index structure for spatial searching". In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 47--57, Boston, MA, June 1984.

[2] T. Sellis, N. Roussopoulos, and C. Faloutsos. "The R*-tree: A Dynamic Index for Multi-Dimensional Objects. " *Proceedings of the VLDB87*, 1987, pp. 507-518.

[3] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles.", Proceedings of the *ACM SIGMOD*, 1990, pp. 322-331.

[4] S. Brin. "Near neighbor search in large metric spaces". In Proceedings of the 21st VLDB International Conference, pp. 574--584, Zurich, Switzerland, September 1995.

[5] J.K. Uhlmann. "Satisfying general proximity/similarity queries with metric trees". *Information Processing Letters*, 40(4):175--179, November 1991.

[6] T. Chiueh. "Content-based image indexing". In *Proceedings of the 20th VLDB International Conference*, pages 582--593, Santiago, Chile, September 1994.

[7] J. Nievergelt, H. Hinterberger, and K.C. Sevcik. "The Grid File: An Adaptable, Symmetric Multikey File Structure". *ACM TODS*, 9(1), March 1984, pp. 38-71.

[8] Ciaccia, P., Patella, M., Zezula, P., "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. " In: VLDB'97, *Proceedings of 23rd International Conference on Very Large Data Bases*. Morgan Kaufmann 1997, pp. 426-435

[9] C. Faloutsos and K.I. Lin. "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia data sets". *Proceedings of the ACM SIGMOD*, San Jose, CA, June 1995, pp. 163-174.

[10] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. "Fast sub-sequence matching in time-series databases". In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 419--429, Minneapolis, MN, May 1994.

[11] P. Zezula, P. Savino, G. Amato, and F. Rabitti. "Approximate similarity retrieval with M-tree". *The VLDB Journal*, 7(4), December 1998, pp. 275-293.