Efficient Indexing for High Dimensional Data: Applications to a Video Search Tool

Thierry Urruty
LIFL – UMR CNRS-USTL n° 8022
University of Lille 1, France
00 33 3 20 43 47 38
urruty@lifl.fr

Fatima Belkouch
LIFL – UMR CNRS-USTL n° 8022
University of Lille 1, France
00 33 3 20 43 47 38
belkouch@lifl.fr

Chabane Djeraba
LIFL – UMR CNRS-USTL n° 8022
University of Lille 1, France
00 33 3 20 43 47 38
djeraba@lifl.fr

ABSTRACT

The emergence of numerical technologies in the audio-visual sector requires the use of powerful tools for accessing data. In this demonstration paper, we focus on content-based indexing and similarity search in very large audiovisual databases of business movie companies. This paper summarizes our analytical study and experimental results for two new indexing structures we propose. These structures are integrated in a search tool we detail briefly intended for professional users of large company film databases.

A video demonstration can be downloaded from www.lifl.fr/~urruty/urruty

Keywords

Video database, high dimensional indexing structure, search tool.

1.SCOPE OF THE TOOL

The expansion of multimedia information databases containing large multimedia digital libraries requires effective access techniques to data.

In our research project, we focus on content-based search within the framework of an application aiming to provide intelligent tools to retrieve video sequences in a large database. Database in the audio-visual sector have huge volumes of unexploited video constantly increasing. An interoperable, content-neutral description interface is needed to facilitate the search and indexing of this growing amount of multimedia data.

The MPEG standards group has formulated the MPEG-7 [7] multimedia content description interface, mainly using the XML format. When the XML format is used for describing audiovisual content, an important quantity of content descriptions is generated. Standard description is powerful enough to represent the complexity of descriptors. However, it is not designed for efficient matching and retrieval, when dealing with important number of descriptors and thousands of hours of audiovisual data., a major shortcoming of the normalized audiovisual content descriptions,based on XML format, and aims at developing a tool providing efficient audiovisual content search. Such emerging video database applications manipulate high dimensional data. In these applications, one of the most frequently used and yet inefficient operation is the ability to find video sequences that are similar to a given query. Queries and data (video sequences) are both represented by the same vector space model in order to match them together.

Our approach starts with audiovisual sequences already annotated by professionals of companies specialized in business films. It exploits the Mpeg-7 Standard [7] to describe the video content in XML files. First, we develop a vector space model adapted to the needs of professional users. We transform the video sequences in XML description files into vectors in a multidimensional space. These vectors are then organized efficiently using multidimensional indexing methods. Our contribution concentrates on the indexing structure. We stress that the techniques developed here have wider applicability in clustering high dimensional data, e.g., microarray data in computational biology.

As far as the search in high-dimensional space is concerned, several indexing methods have been proposed to deal with the high number of dimensions. The difficulties of dealing with high-dimensional spaces are collectively known as the "curse of dimensionality" [5]. Generally, most of the existing methods are not adapted to different workloads (different data sizes, dimensionalities, data distributions, selectivities, etc.) and to both types of queries: WQ (Window Query also called Range Query) and KNN query (K-Nearest Neighbors). For example, similarity based searches using the *Pyramid* technique [1] as indexing structure are not affected by high dimensional data. The performance of these algorithms depends on the data distribution and the position of the query in the space. A simple search can generate useless accesses to a very large volume of data, which impacts considerably the performance and makes the *Pyramid* technique less efficient. Two others methods *IDistance*[2] and *IMinMax*[4] are suitable for only one type of query, KNN and WQ respectively. A more recent method *P+Tree*[3] attempts to improve the *Pyramid* technique by reducing the space concerned by a search query. The dividing space method it uses is not efficient.

Consequently, the existing indexing techniques perform well for some databases and poorly for others. The performance of the algorithms generally depends on the workload and sequential scan remains an efficient search strategy for similarity search.

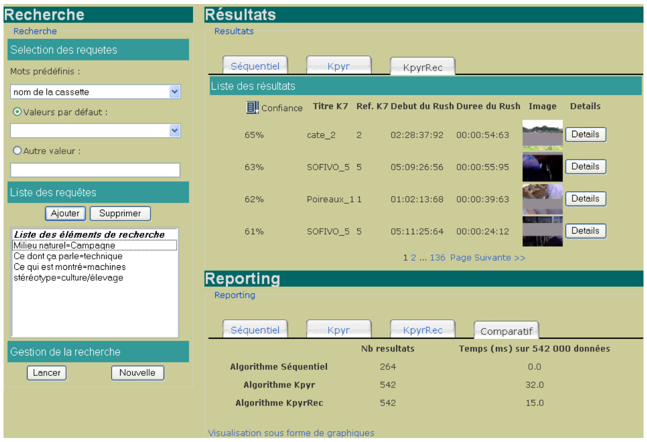
We present here a demonstration based on two indexing methods *Kpyr* [6] and *KpyrRec* on a real video database. The first technique, *Kpyr* provides the best conditions to apply the *Pyramid* Technique and its performance is affected neither by data size nor by data distribution. However, it is slightly influenced by the data dimensionality. Experimental results indicate that the response time of a search query depends mainly on the number of accessed data points. Thus, we propose a new indexing method called *KpyrRec* which is based on a recursive splitting algorithm on the data space. We show that dividing the space helps to perform faster searches. Surprisingly, we observe that one can dispense with the sequential scan. In extreme cases the sequential scan may outperform our technique; however, such cases do not correspond to actual situations.

We implemented our two method Kpyr and KpyrRec on our company film database showing the good performances of our methods.

2. DEMONSTRATION TOOL

2.1 Video Search Tool

A goal of our project is to build a search tool for video sequence retrieval. This search engine has been realized in order to be used on-line. It is intended for professional users in company films. Three steps are necessary: the first one concerns the search interface where the user specifies its query. The user can choose a category within a list concerning the application field. Categories are corresponding to the *company films* field. After choosing a category, our interface gives the end user a list of frequent keywords links to this category. For example, "Natural environment" category, we have "Sea/Ocean", "Country side", "Mountains" as linked frequent keywords. The end user can choose the categories and keywords he needs, before submitting his search. In Figure 1, we give a query example; we got four categories and keywords (see the video demonstration for more details).



Recherche

Recherche

Selection des requetes

Mots prédéfinis : nom de la cassette

● Valeurs par défaut :

○ Autre valeur :

Liste des requêtes

Liste des éléments de recherche
Milieu naturel=Campagne
Ce dont ça parle=technique
Ce qui est montré=machines
stéréotype=culture/élevage

Gestion de la recherche

Résultats

Resultats

Liste des résultats

Confiance	Titre K7	Ref. K7	Debut du Rush	Duree du Rush	Image	Details
65%	cate_2	2	02:28:37:92	00:00:54:63		Details
63%	SOFIVO_5	5	05:09:26:56	00:00:55:95		Details
62%	Poireaux_1	1	01:02:13:68	00:00:39:63		Details
61%	SOFIVO_5	5	05:11:25:64	00:00:24:12		Details

1 2 ... 136 Page Suivante >>

Reporting

Reporting

	Nb resultats	Temps (ms) sur 542 000 données
Algorithme Séquentiel	264	0.0
Algorithme Kpyr	542	32.0
Algorithme KpyrRec	542	15.0

Visualisation sous forme de graphiques

Figure 1. Video search engine tool

The second step of the search sends the end user query to the server that contains our database and our indexing structure. This query is then analyzed as presented in section 4.

The last step concerns the results of the search. Indeed, as shown in Figure 1, all video sequences resulting from this search appear in our result interface with some information related to each sequence. We display on the screen the pertinence value, the video title, the number of videotape linked to the same project, the beginning of the sequence in the videotape, its duration and a key image of the video sequence. A link has been added to allow end user needs to see the normalized audiovisual content description.

Below the frame of the ordered results returned, we have a report on time information, including the time to query for sequential scan, *Kpyr* and *KpyrRec*. This report shows the good performance of our two algorithms on a real ideo database.

2.2 Visualization tools

At the same time, we have the possibility to visualize the multidimensional vectors representing the video sequence of our database with the Window query done by the end user. These visualization tools are very powerful to understand the distribution of the video sequences in any category. Figure 2 (top) shows the parallel coordinate visualization of our database after transforming all video sequences into multidimensional vectors. In figure 2 (middle), we visualize only the data set of one cluster. W also see an example of a window query made by an end user; the two black lines represent the minimal and maximal limit of the Window Query. In Figure 2 (bottom), we show only the vectors included in the Window Query.

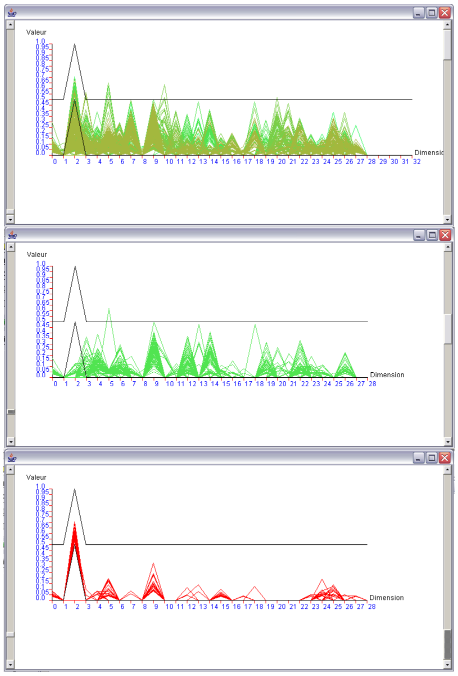


Figure 2. Parallel visualization

Our visualization tool also allows us to choose two interesting categories among all. As we show in Figure 3, we can choose the visualization of all data, one cluster, or only the vectors corresponding to the Window Query. Choosing "*Human*" and "*Nature*" as two dimensions allows visualizing the points corresponding to video sequences containing for example people and landscape. We note that the Window Query contains lots of points but only a few of them are considered as solutions as they have to be inside the WQ for all dimensions.

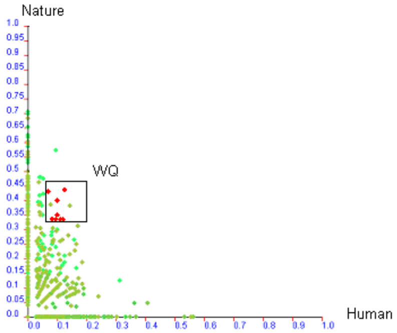


Figure 3. Visualization in dimensions "*Nature*" and "*Human*"

The visualization of all data points in the space enables us to know the distribution of all data points, the distribution of data within a cluster, and the distribution of data with respect to one or two dimensions. It permits to re-examinie either the number of clusters or our vector space model. Furthermore, visualization of the results compared to the WQ gives a preliminary overview of the results. That enables us to adapt the value of selectivity.

3.CONCLUSION

In this demonstration, we detail briefly our research objectives and show an application of our indexing methods in a video search tool. A demonstration video of our tool accompanies this paper showing the good response times using our indexing methods.

4.REFERENCES

[1] S. Berchtold, C. Bohm and H.-P. Kriegel, "The Pyramid Technique: Towards Breaking the Curse of Dimensionality", *in Proc. ACM SIGMOD Int. Conf. on Management of Data,* 1998, pp. 142-153.

[2] C. Yu, B.C. Ooi, K.-L. Tan and H.V. Jagadish, "Indexing the distance: An efficient method to Knn processing", *in VLDB*, September 2001, pp. 421-430.

[3] R. Zhang, B.C. Ooi and K.L. Tan, "Making the Pyramid Technique Robust to Query Types and Workloads", *in IEEE ICDE, 20th International Conference on Data Engineering*, Boston, USA, April 2004.

[4] B.C. Ooi., K.L. Tan, C. Yu and S. Bressan, "Indexing the Edges - A Simple and Yet Efficient Approach to High-Dimensional", *in 19th ACM SIGMOD SIGACT SIGART Symposium on Principles of Database Systems*, Dallas, USA, May 2000.

[5] Bellman, R.E. 1961. "Adaptive Control Processes". Princeton University Press, Princeton, NJ.

[6] T. Urruty, F. Belkouch and C. Djeraba, "*Kpyr*: an efficient indexing method", *in Proc.* Of IEEE International

[7] MPEG: Moving Picture Experts Group, http://www.chiariglione.org/mpeg/