FEATURE STATISTICAL RETRIEVAL APPLIED TO CONTENT-BASED COPY IDENTIFICATION

Alexis Joly[1,2]*, Carl Frélicot*[1] *and Olivier Buisson*[2]

[1] Labo. d'Informatique-Image-Interaction, Université de La Rochelle, 17042 La Rochelle, France
[2] Département Recherche et Études, Institut National de l'Audiovisuel, 94366 Bry/Marne, France
{ajoly,obuisson}@ina.fr, carl.frelicot@univ-lr.fr

ABSTRACT

In many image or video retrieval systems, the search of similar objects in the database includes a spatial access method to a multidimensional feature space. This step is generally considered as a problem independent of the features and the similarity type. The well known multidimensional nearest neighbor search was also widely studied by the database community as a generic method. In this paper, we propose a novel strategy dedicated to pseudo-invariant features retrieval and more specifically applied to content-based copy identification. The range of a query is computed during the search according to deviation statistics between original and observed features. Furthermore, this approximate search range is directly mapped onto a Hilbert space-filling curve allowing an efficient access to the database. Experimental results give excellent response times for very large databases both on synthetic and real data. This work is used in a TV monitoring system including more than 13000 hours of video in the reference database.

1. INTRODUCTION

Content-Based Copy Identification (CBCI) schemes are an alternative to the watermarking approach for persistent identification of images and video clips. As opposed to watermarking, the CBCI approach only uses a content-based comparison between the original object and the candidate one [1, 2]. It generally consists in extracting few small pertinent features (also called *signatures* or *fingerprints* [3]) from the image or the video stream and matching them with a *DataBase* (DB). As for many content based retrieval systems, one of the difficult task is the cost to search similar objects in a large DB. To mitigate this problem, many similarity search systems use a spatial access method in a multidimensional space feature. When objects are already vectors (signatures for example), multidimensional access methods can be used directly or after dimension reduction techniques, e.g. in [4]. For other complex similarity metrics, embedding methods [5] allow the mapping of a given set of objects with a similarity function between them into a multidimensional embedding space. However all these methods consider the multidimensional access method in the feature space as a *black box* receiving a range query or a k-NN (*Nearest-Neighbors*) query. In this paper, we propose a new multidimensional access method dedicated to pseudo-invariant features retrieval and applied to a CBCI video scheme. Section 2 discusses the specificities of CBCI and how those can be used to reduce the response time of a multidimensional access method. In section 3, we present our new database strategy based on statistical range queries mapped onto Hilbert's space filling curve. Experiments are presented in section 4, both with synthetic data and with a real large DB of local signatures.

2. CBCI SPECIFICITIES AND STATISTICAL BASED QUERIES

In many content based image retrieval schemes, the retrieval is processed by a k-NN query or a range query in the feature space. The idea is to find the features that are the most similar to a requested one. This has often been extended to the CBCI problem [6] though the expected result differs by: *(i)* a copy is defined by a set of tolerated transformations of the original object, *(ii)* the search only consists in finding the original signature if the query is a copy of a referenced object. The difference of query type is fundamental regarding the search complexity [7] which highly depends on the spatial range mapped by the query. In a k-NN search, this spatial range depends on the DB size and on the local points density around the query. In a CBCI scheme, it depends on the distortions between the original signature and the signature of the copy. For example, if the signature is quite invariant to some expected transformations, the spatial range of the query can be strongly limited. In the past few years, the use of approximate NN queries proved that small losses in quality can be traded for high response time gains [8]. However, response times are often linear versus the DB size when increasing amount of data is a major stake for rights protection systems.

Most spatial access methods in multidimensional feature space comprise a *filtering step* and a *refinement step* [9, 10]. The *refinement step* is generally the same process which would be used with a naive sequential scan but it is applied only to parts of the DB selected by the *filtering step*. For CBCI schemes, we propose to adapt the *filtering step* to the expected distortions with statistical based queries. For a given query signature, the idea is to predict a region of the space where the probability to find the enventual referenced signatures of same object is superior to a fixed threshold α. Formally, for a given query Y and a user defined probability α, the *filtering step* consists in finding the set S_α of all signatures contained in a hyper-volume V_α such as:

$$\int_{V_\alpha} p(X|Y)\, dV \geq \alpha \qquad (1)$$

where $p(X|Y)$ is the probability density function that X and Y are signatures of same object, given Y.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

The *refinement step* is then processed only on S_α. This process is commonly a k-NN search, a range search or both. If the DB contains several copies of some objects (repeat in a TV DB for example), the k-NN search sets the maximum number of potential copies and the range search radius sets the decision threshold. When local signatures are used [2, 6], the final result is a consolidation of many partial results. Each partial result contains many candidate signatures, and possibly the whole S_α. The *refinement step* could then be only the computation of the distances or of a probability for each point in S_α.

CBCI applications such as copyright protection or broadcasting checking generally do not require an immediate response. Even for a TV monitoring application, e.g. in [2], the system must be on average sufficiently fast but the response to a query can be delayed. This tolerance allows to gather several queries and to avoid many disk accesses when the DB exceeds primary storage size (see details in section 3). Finally, insertions or deletions in the DB are not constantly required; the DB can be only searched during *in-line run* and costly *off-line* processes are affordable.

3. DATABASE STRATEGY

Multidimensional indexing using Hilbert's space filling curve was originally suggested by Faloutsos [11] and fully developed by Lawder [12]. The principle of our retrieval method is quite similar to Lawder's one: The query is mapped to Hilbert's curve coordinate and it is converted into several curve sections. The refinement step then consists in scanning the data points belonging to these sections. Hilbert's curve clustering property limits the number and the dispersion (on the whole curve) of these sections reducing the number of memory accesses. However our method differs in several main points. Lawder's filtering step requires the use of state diagrams to compute the mapping to Hilbert's curve which limits the dimension to about 10 because of primary storage considerations. Furthermore, only hyper-rectangular range queries are computable. Our statistical based filtering step uses Butz algorithm [13] for the mapping and requires little memory. The DB is physically ordered according to points position on Hilbert's curve and it is locked during the whole in-line search stage. For a given query, once the curve sections have been identified, the corresponding sets of successive points in the DB are localized by an index table. Then the refinement step sequentially scans each set of successive points.

The DB is stored in a file and is loaded in primary storage at the beginning of the in-line stage. When the DB exceeds primary storage size, it is cyclically loaded in several memory size blocks and several queries are searched together (see subsection 3.2). Subsection 3.1 describes the proposed filtering step where we assume that the D components of the signature are independent:

$$p(X|Y) = \prod_{j=1}^{D} p(x_j|y_j)$$

3.1. Filtering step

The K-th order approximation of Hilbert space-filling curve in a D-dimensional grid space H_K^D, is a bijective mapping: $\left[0, 2^K - 1\right]^D \leftrightarrow \left[0, 2^{KD} - 1\right]$. The main property is that two neighboring intervals on the curve always remain neighboring cells in the grid space. The reciprocal property is generally not true and the quality

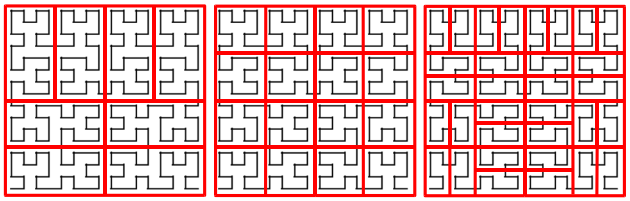

Fig. 1. Space partition for D=2 and K=4 at different depths – from left to right: p=3,4 and 5

of a space filling curve can be evaluated by its ability to preserve a certain locality on the curve.

Some intermediate variables of Butz algorithm, allow to easily define the space partition corresponding to the regular partition of the curve in 2^p intervals [14]. Parameter $p \epsilon [1, KD]$ is called the *depth* of the partition (by analogy to KD-trees). Like illustrated on Figure 1, the space partition is a set of 2^p hyper-rectangular blocks (called p-blocks) of same volume and shape but of different orientations.

For a p-partitioned space, inequality (1) may be satisfied by finding a set B_α of p-blocks such as:

$$\sum_{i=1}^{card(B_\alpha)} \int_{b_i} p(X|Y)\, dV \geq \alpha \qquad (2)$$

where $B_\alpha = \{b_i : i \in [1, card(B_\alpha)]\}$ and $0 \leq card(B_\alpha) \leq 2^p$.

In practice, $card(B_\alpha)$ should be minimum to limit the cost of the search. We refer to this particular solution as B_α^{min} whose computation is not trivial because sorting the 2^p blocks according to their probability is not affordable. Nevertheless, it is possible to identify quickly the set of blocks with a probability greater than a fixed threshold t :

$$B(t) = \left\{ \{b_i\} : \int_{b_i} p(X|Y)\, dV > t \right\}$$

and the corresponding probability sum :

$$P_{sup}(t) = \sum_{i=1}^{card(B(t))} \int_{b_i} p(X|Y)\, dV$$

Since $card(B(t))$ decreases with t, finding B_α^{min} is equivalent to finding t_{max} verifying:

$$\begin{cases} P_{sup}(t_{max}) \geq \alpha \\ \forall t > t_{max}, P_{sup}(t_{max}) < \alpha \end{cases} \qquad (3)$$

As $P_{sup}(t)$ also decreases with t, t_{max} can be easily approximated by a method inspired by Newton-Raphson technique.

Parameter p is of major importance since it directly influences the response time of our approximate method

$$T(p) = T_f(p) + T_r(p)$$

The filtering time $T_f(p)$ is strictly increasing because the computation time of B_α and the number of memory accesses increase with p. The refinement time $T_r(p)$ is decreasing because the *selectivity* of the filtering step increases, i.e $card(S_\alpha)$ decreases with p. $T(p)$ has generally only one minimum at p_{min} which can be learned at the beginning of the in-line stage in order to obtain the best mean response time on several queries.

3.2. Disk strategy

When the DB exceeds memory size, several, say N_{sig}, signatures are searched together. At the beginning of the in-line stage, the Hilbert's curve is split in 2^r regular sections ($0 \geq r \geq p$), such as the most filled section fits in memory. The filtering step is processed for each signature during a first stage. Each section is then sequentially loaded and searched by the refinement step. The mean total process time per query is given by:

$$\overline{T}_{tot} = \overline{T} + (T_{load}/N_{sig}) \quad (4)$$

where T_{load} is the loading time for the entire DB. This additional time introduces a linear component in the response time against DB size, however it can be neglected in most cases by adjusting N_{sig}. In our system, N_{sig} is automatically set to obtain a constant mean loading time per query whatever the DB size is (see 4.3).

4. EXPERIMENTS AND RESULTS

Experiments were computed on a Pentium IV (CPU 2.5 GHz, cache size 512Kb, RAM 1.5 Gb). Response times were obtained with unix `getrusage()` command. For comparison, we implemented our own version of the sequential scan method, which loads the entire DB in main memory. When the DB does not fit in memory, a similar disk strategy than above described is used. Similarly to (4), the mean total process time per signature is given by: $\overline{T}_{totsequ} = \overline{T}_{sequ} + T_{load}/N_{sig}$. Since the loading time is identical for both methods and can be adjusted as seen previously, it is not taken into account. The sequential scan and the refinement step were performed with L^2-metric. As signature components are coded on one byte, all measures refer to the space $[0, 255]^D$ and it was assumed that:

$$p(x_j|y_j) = p(x_j - y_j) \ , \ \forall j = 1, D$$

4.1. Synthetic database

In order to assess the performance of the method as the DB size grows, we generated signatures DBs of different size containing random 20-dimensional signatures drawn from the uniform distribution on [0,255]. For each DB, 1000 signatures were randomly selected, distorted and searched with both the proposed method and the sequential scan. Each signature component was independently corrupted by an additional zero-mean gaussian noise with arbitrary standard deviation $\sigma_j = 22.5, \ \forall j = 1, D$. The probability of the statistical based query was fixed to $\alpha = 0.96$. The refinement step process was the same than the sequential scan process and was a 5-NN search. Note that, only the time cost of this step is important. A signature is considered to be *retrieved* if it is selected by the filtering step. Based on the different DB sizes, we obtained the following 95% confidence interval for the *retrieval rate*: $r = 0.957 \pm 0.007$, to which expected value 0.96 belongs.

Let $\overline{T}_{sequ}$ and $\overline{T}$ be the mean response times for the sequential scan and the proposed method. $\overline{T}_{sequ}$ is known to be a linear function of the DB size N. Figure 2 shows the response time gain, defined by $G = \frac{\overline{T}_{sequ}}{\overline{T}}$, as a function of the DB size. The linear behavior of G in log-log scale, corresponds to a sub-linear behavior for the response time that we can graphically evaluate to $\overline{T} = 2.046 \ 10^{-6} \times N^{0.38}$ *sec*. Therefore, the larger N, the more profitable the proposed method, compared to a linear search as the sequential scan or most approximate methods are.

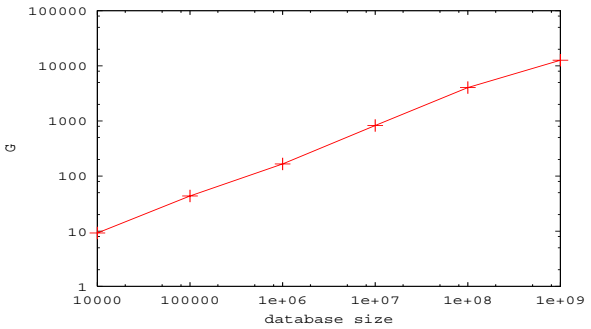

Fig. 2. Time gain G *versus* DB size (log-log scale)

4.2. Real database

The real signatures DB was obtained using the Content Based Video Copy Identification scheme described in [2]. It contains 434,240,861 20-dimensional signatures whose components are local differential descriptors extracted in key images around interest points. This represents 11.04 Gb corresponding to 8042 hours of (color and black&white) TV video extracts from various programs: news, sport, shows, movies. Probability density functions $p(x_i - y_i)$ were estimated by extracting signatures both in original sequences and in distorted ones at the corresponding positions. We focused on four image distortions: resizing ($factor = 0.8$), gamma correction ($factor = 0.5$), zero-mean gaussian noise addition ($\sigma_n = 20.0$), and imprecise interest points location (by a 1 pixel shift operation). The *distorted signatures* were then searched both with sequential scan and with our method initialized with estimated cumulative distribution functions and $\alpha = 0.96$. Again, the refinement step was a 5-NN search for both methods. Table 1 reports the observed retrieval rates r, the time gains $G_{dist.}$, the time responses $t_{dist.}$. In addition, the means of distribution parameters $\overline{\mu}_{dist.} = \frac{1}{D}\sum_{j=1}^{D}\mu_j$ and $\overline{\sigma}_{dist.} = \frac{1}{D}\sum_{j=1}^{D}\sigma_j$ are given. Small losses in retrieval (8.3 % in the worst case) allowed us to obtain significantly low response times ($< 86 \ msec.$), as compared to the litterature [6]. Note that the relative error $\frac{|r-\alpha|}{\alpha}$ (4.5% in worst case) is due to the underlying model assumptions.

distorsion	resize	gamma	noise	shift
r (%)	91.7	94.3	94.6	93.2
$G_{dist.}$	327	26509	4532	1090
$t_{dist.}$ $(msec.)$	85.9	1.06	6.20	25.78
$\overline{\mu}_{dist.}$	0.122	-0.597	0.016	0.546
$\overline{\sigma}_{dist.}$	22.084	4.347	10.607	16.757

Table 1. Results for real image distortions

The final result deals with the gain G behavior with respect to an arbitrary standard deviation of the additional noise corrupting the signatures ($\sigma_j = \sigma, \ \forall j = 1, D$). Figure 3 shows G as a function of σ. $\overline{T}_{sequ}$ does not depend on σ and is is about $28.10 sec.$ The high decrease of the G when σ grows shows that the *severity* of expected distortions is decisive for the response time. This explains the variability of the response time for the different distortions in Table 1. However, the gain remains superior to one. This is due to the learning of p which guarantees that our method is better than a sequential scan (equivalent to $p = 0$).

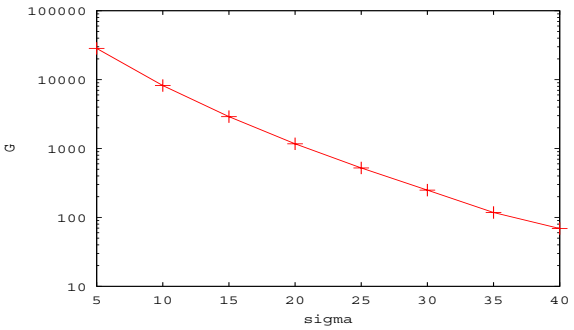

Fig. 3. Time Gain G *versus* σ (log scale)

4.3. TV monitoring system

Our method is integrated in a TV monitoring prototype watching a growing TV archive DB. It includes today about 14,000 hours of video (750,000,000 signatures identical to those described in subsection 4.2). The signature stream of one TV channel is continuously processed by one single Pentium IV (without TV capture). Statistical estimation of probability density functions $p(x_j - y_j)$ are computed as in subsection 4.2 but with a whole set of image distortions simply determined by watching TV and asking TV archivists. The number of signatures together searched, N_{sig}, is increasing proportionally with DB size in order to have a constant mean loading time per query which is about $T_{load}/N_{sig} = 4.5\ ms$. For a 14,000 hours DB the total loading time is $T_{load} = 16\ min.$ with $N_{sig} = 210,000$. The corresponding delay between broadcasting and results is 3 $hours$ 45 min. The total mean response time per query is $\overline{T}_{tot} = \overline{T} + (T_{load}/N_{sig}) = 20.2 + 4.5 = 24.7\ ms$. The mean needed time to search for 1 hour of video is 21 min. Figure 4 shows results obtained while monitoring a french TV channel.

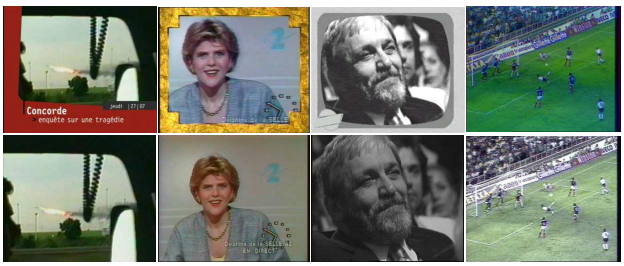

Fig. 4. Broadcast (up) and retrieved videos (bottom)

5. CONCLUSION AND PERSPECTIVES

The proposed statistical features search strategy applied to CBCI problem gives excellent response times. The sub-linear behavior of the response time with the DB size allows to control very large audiovisual DBs even with high signatures rate. This method could be easily applied to other applications involving large pseudo-invariant feature DBs, such as biometrics or object recognition. Investigations in statistical modeling of data and distortions should improve the method. However, investigating the signature itself certainly could be more profitable: a reduction of the sensitivity to distortions would improve both the retrieval efficiency and the response time. The independence between components is also a common objective. Future works will compare different signatures according to these objectives. The impact of independent component analysis, embedding methods or kernel based methods is another perspective.

6. REFERENCES

[1] A. Hampapur and R. Bolle, "Comparison of sequence matching techniques for video copy detection," in *: SPIE Conference on Storage and Retrieval for Media Databases*, 2002.

[2] Alexis Joly, Olivier Buisson, and Carl Frélicot, "Robust content-based video copy identification in a large reference database," in *International Conference on Image and Video Retrieval*, 2003, pp. 414–424.

[3] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Int. Conference on Visual Information and Information Systems (VISUAL)*, 2002, pp. 117–128.

[4] S-C.S. Cheung and A. Zakhor, "Fast similarity search on video signatures," in *International Conference on Image Processing*, 2003.

[5] G. R. Hjaltason, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 530–549, May 2003.

[6] P. Gros S.-A. Berrani, L. Amsaleg, "Robust content-based image searches for copyright protection," in *ACM International Workshop on Multimedia Databases*, 2003, pp. 70–77.

[7] D. A. Keim C. Böhm, S. Berchtold, "Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases," *ACM Computing surveys*, 2001.

[8] Roger Weber and Klemens Böhm, "Trading quality for time with nearest-neighbor search," *Lecture Notes in Computer Science*, vol. 1777, pp. 21–35, 2000.

[9] Alberto Belussi and Christos Faloutsos, "Estimating the selectivity of spatial queries using the 'correlation' fractal dimension," in *Proc. 21st Int. Conf. Very Large Data Bases, VLDB*, Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, Eds. 11–15 1995, pp. 299–310, Morgan Kaufmann.

[10] Ashraf Aboulnaga and Jeffrey F. Naughton, "Accurate estimation of the cost of spatial selections," in *ICDE*, 2000, pp. 123–134.

[11] C. Faloutsos and S. Roseman, "Fractals for secondary key retrieval," 1989, pp. 247–252.

[12] J.K. Lawder and P.J.H. King, "Querying multi-dimensional data indexed using the hilbert space-filling surve," *SIGMOD Record*, vol. 30, pp. 19–24, 2001.

[13] A. R. Butz, "Alternative algorithm for hilbert's space-filling curve," *IEEE Transaction on Computers*, vol. C, no. 2, pp. 424–426, 1971.

[14] J. Lawder, "The application of space-filling curves to the storage and retrieval of multi-dimensional data," Technical report jl/1/99, Birkbeck College, University of London, 1999.