Z-grid-based Probabilistic Retrieval for Scaling Up Content-Based Copy Detection

Sébastien Poullot
Institut National de l'Audiovisuel
94366 Bry-sur-Marne, France
spoullot(a)ina.fr

Olivier Buisson
Institut National de l'Audiovisuel
94366 Bry-sur-Marne, France
obuisson(a)ina.fr

Michel Crucianu
CEDRIC - CNAM
292 rue St Martin
75141 Paris cedex 03, France
michel.crucianu(a)cnam.fr

ABSTRACT

Scalability is the key issue in making content-based copy detection (CBCD) methods practical for very large image and video databases. Since copies are transformed versions of original documents, CBCD involves some form of retrieval by similarity using as queries the descriptions of potential copies. To enhance the scalability of an existing competitive CBCD method, we introduce here three improvements of this retrieval process: a Z-grid for building the index, uniformity-based sorting and adapted partitioning of the components. Retrieval speed is significantly increased, enabling us to monitor with a single computer one TV channel against a database of 120,000 hours of video.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

General Terms

Algorithms, Experimentation, Performance.

Keywords

Content-based video copy detection, scalability, multidimensional index structures.

1. INTRODUCTION

The fast growth in the production of both professional and personal audiovisual content, together with the continuous multiplication of content diffusion channels, challenge the ability of legal owners to protect their rights. First, because the risk of intentional or accidental unauthorized (re)use of content significantly increases. Second, because the detection of the unauthorized (re)use of content faces a very serious scalability problem.

Copy detection is a key issue in protecting owners' rights and consists in finding whether a candidate document is issued from an original document stored in a content base. Existing solutions rely either on the use of watermarks (see [12]) or on signatures extracted from the content itself. Each of these alternatives has specific advantages and drawbacks. Watermarks can include various useful meta-data and can keep the computational costs of copy detection relatively low, but are not very robust to image transformations frequently performed during copy creation (blur, crop, add logos or frames, resize, etc.). Also, watermark-based copy detection cannot be used if copies of the original content were disseminated before the application of any mark, which is unfortunately the case for a large part of the existing content. Recent content-based copy detection (CBCD) methods for still images and video (e.g. [1], [8], [10], [9]) do not depend on the presence of marks and are more robust to image transformations. However, CBCD methods have a higher computational cost, so scalability is more difficult to achieve and will be our main focus in this paper.

Figure 1: Copy (left) and original image (right)

We focus here on the scalability of video CBCD and put forward several evolutions of the method described in [8, 9]. They allow us to perform searches significantly faster, so monitoring video streams against very large databases of original video content (120,000 hours of video) becomes feasible. In the next section, after a short state of the art regarding content-based image and video copy detection, we present the CBCD framework we employ. Our contributions regarding the index structure and associated search method are described in Section 3, then an experimental comparison with the method in [9] and evaluation on databases of up to 120,000 hours of video is shown in Section 4.

2. CONTENT-BASED COPY DETECTION

2.1 Similarity and copy detection

By *copy* we understand a document that is issued from the original and is perceived as being very similar to it. When

CIVR'07, July 9–11, 2007, Amsterdam, The Netherlands.

the copy is indistinguishable from the original (*near-exact* copy), simple solutions can be employed for detection. But in most cases of interest several transformations (filtering, addition of noise, cropping, addition of logos or frames, resizing, etc.) are applied to the original in order to obtain the copy. As in [7], we consider that a *copy* is the result of the application of a *tolerated transformation* to an original document. A transformation can be primitive or composed. A primitive transformation—usually not an invertible function—is defined by a primitive type (e.g. change in gamma, resizing, addition of noise) and associated parameters. A transformation is tolerable if the document obtained after its application is *similar enough* to the original to be considered a copy; this is, to a large extent, a subjective matter. However, the study of copies in a large audiovisual database has shown ([9]) that the probability of a transformation decreases when the "amplitude" of the transformation increases. The use of similarity for copy detection can allow to link documents that are copies of a common (but unavailable) ancestor, though none of these documents is a copy of another according to the definition above.

This definition of a copy implies that for CBCD we need to evaluate the similarity between the candidate document (potential copy) and the original; if the similarity is high, then the candidate is likely to be a copy of the original (source). For automatic evaluation we need a content description scheme and an associated metric that are as insensitive as possible to the tolerated transformations, while being as sensitive as possible to the differences between documents that are not linked by such transformations. Also, since we have an entire database of original documents to protect, the candidate document should be used as a query and every original document within the *similarity range* of this query should be returned as a possible source.

The similarity-based retrieval part of content-based image or video copy detection follows the query by example (QBE) paradigm of Content-Based Image (or video) Retrieval (CBIR, see [4], [2]). However, the specific goal of copy detection has an impact both on the choice of image descriptors and on the types of similarity queries employed.

2.2 Image and video CBCD

Existing proposals for image or video CBCD differ by the content description scheme and associated metric, and by the way similarity-based queries are processed.

Regarding content description, early proposals ([3], see also [5]) rely either on global image features or on image-block features. These features are appropriate for near-exact copy detection, but are not robust to the tolerated transformations we mentioned above, which are typically encountered in more general copy detection applications.

To achieve good robustness, later proposals, beginning with [14], [8], [1], describe every image by a set of *local* features and evaluate the similarity between two images as a score of the best match between the sets of features associated to the images. Part of the robustness comes from the fact that the local features employed are invariant to some of the transformations. Also, matching involves some form of voting, thus allowing for partial matches that provide robustness to other transformations. But the use of sets of local features requires a two-stage process: first, the individual features of the candidate image are used as queries for retrieving similar local features from the database of local features extracted from the original documents; then, matching is performed and the decision is taken. It was recently shown in [11] that the quality of the decisions can be significantly improved if a spatio-temporal registration method is employed.

When the amount of documents to protect is small or when few compact features are sufficient for reliable decisions (as is usually the case for near-exact copy detection), the size of the database of original documents is small and similarity-based retrieval can be performed fast enough by sequential search. However, in most cases of interest the size of the database is large and some index structure is needed for speeding up retrieval. Many multidimensional index structures and associated search methods were put forward for performing similarity-based retrieval and the recent monograph [13] presents a comprehensive review. Most of these solutions were developed for the retrieval of either all the items within a given range of ϵ around the query item (ϵ-range queries) or of the k nearest neighbors of the query item (kNN queries). While these types of queries are adequate for searches in spatial databases or for content-based image retrieval (CBIR), they may not be the most appropriate for content-based video copy detection.

Let us regard kNN queries first. In low-density regions they will retrieve neighbors that are too far from the query for this query to be considered a potential copy of any such neighbor. Alternatively, in high-density regions, kNN queries may not retrieve all the neighbors for which the query can be a copy. To see why ϵ-range queries might not be appropriate either, remember that, according to the study in [9] of copies present in a large audiovisual database, the probability of a transformation decreases when the "amplitude" of the transformation increases. By using a well-defined range, an ϵ-query will give equal importance to near and farther neighbors within the range, with a negative impact on performance, as shown in [9] (we return to this issue later). Following [8], we consider instead *probabilistic* queries based on a simple model of the effects of tolerated transformations, and we improve the indexing and retrieval scheme in [9].

2.3 Our framework for CBCD

We first provide a short overview of the CBCD framework we rely on, introduced in [9]; the reader should refer to [9, 7] for further details.

Content-based copy detection workflow. As shown in Fig. 2, a CBCD workflow has two main components, one offline and one online. The offline component builds the database of video signatures (called *reference signature database* in the following) extracted from the database of original documents (*reference content database*) and an index structure on this database. The local signatures we employ are described below. We consider that the original content database increases at a slow rate over time, so we do not need an incremental index construction method.

The online component checks whether a candidate video (from a stream or a database) is a copy of an original video stored in the reference content database. To do this, local signatures are first extracted from the candidate video. Then, each of these signatures is used as a query to retrieve similar signatures from the reference signature database. These similarity-based retrieval operations are critical for the scalability of the copy detection method; retrieval must have a very low costs even for large reference databases and

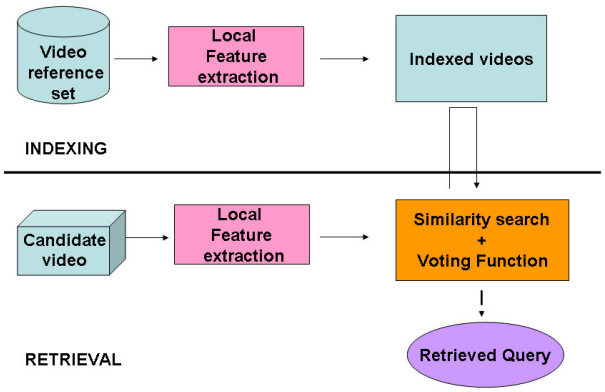

Figure 2: Workflow of a content-based video copy detection system

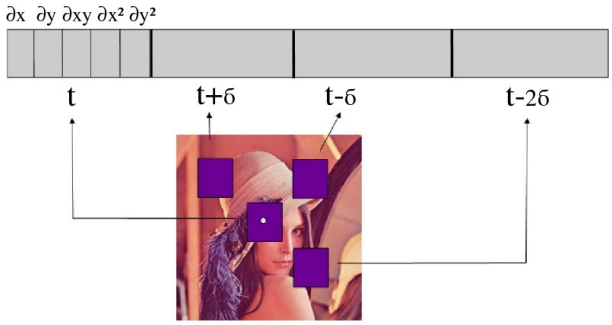

Figure 3: Spatio-temporal components of the 20-dimensional signature of one point of interest

complexity should be sublinear in the size of the database. The solution proposed in [9] is shortly described below and our improvements are presented in Section 3.

Matching is the last stage of the copy detection process. To decide whether the candidate video is a copy or not, we use the procedure described in [9]: the original videos corresponding to the retrieved signatures are identified, then the match between every such original video and the candidate video is evaluated via a vote over all their signatures.

This workflow is generic: it can accommodate any interest point detector, any local signature, any similarity-based retrieval method and any vote-based decision method.

Robust representation of video content. It was shown ([14], [8], [1]) that the use of local image features together with a vote-based decision scheme results in increased robustness to the tolerated transformations typically encountered in general copy detection applications. Together with scalability constraints, this motivates our choice of the local features introduced in [8].

To represent a video, keyframes are first extracted from the video stream; for a 25 fps (frames per second) video, we detect on average 1 keyframe per second (slightly depending on the type of program). For every keyframe, points of interest are obtained using an improved Harris detector [15]. Then, the neighborhood of every point of interest is described by the normalized 5-dimensional vector of first and second-order partial derivatives of the gray-level brightness. To obtain the 20-dimensional spatio-temporal signature associated to a point of interest in the frame at time t, the same type of description is computed for 3 other neighboring points in frames $t+\delta$, $t-\delta$ and $t-2\delta$ respectively, as shown in Fig. 3. Each component is coded on a byte, so an individual signature takes 20 bytes.

The use of the improved Harris detector and of the differential description makes the local features rather invariant to changes in contrast and gamma. The vote-based decision scheme provides significant robustness to other common transformations such as cropping, addition of logos or frames, combination with other frames.

Fast similarity-based retrieval. The local signatures representing the keyframes of the candidate video are used as queries to retrieve similar signatures from the reference signature database. These similar signatures are issued from videos that are potential original documents from which the candidate video was obtained.

Finding that the probability of a tolerated transformation decreases when the "amplitude" of the transformation increases, [8] proposed to model the effect of tolerated transformations on a signature by an isotropic multidimensional Gaussian probability density function, and to perform probabilistic retrieval based on this model. For any given query, the signatures issued from potentially original documents with respect to this query are distributed according to the same model. The 20-dimensional description space is partitioned into cells, which are accessed through an index.

Probabilistic retrieval consists in selecting a minimum number of cells such that their cumulated probability (following the model) is above a fixed threshold P_α considered acceptable for reliable copy detection. For a query $\mathbf{q}$ and a 1D standard deviation σ for the isotropic Gaussian model, the probability of a cell $(\mathbf{a},\mathbf{b}) = [a_1,b_1]\times\cdots\times[a_{20},b_{20}]$ is then

$$P(\mathbf{a},\mathbf{b}) = \prod_{i=1}^{20}\int_{a_i}^{b_i}\mathcal{N}(q_i,\sigma)(x)dx \qquad (1)$$

Fig. 4 shows a simple partitioning of a 2D square into cells and a query with an isotropic Gaussian model. The color of a cell represents the probability to find a potential original signature in that cell: the darker it is, the higher is the probability.

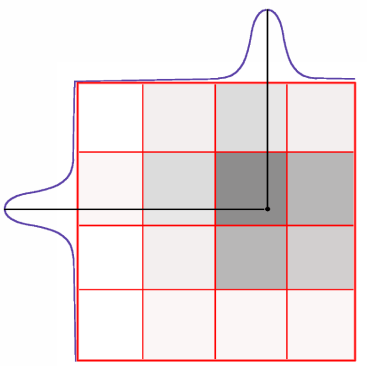

Figure 4: A probabilistic query following a 2D Gaussian pdf on a 2D grid. Darkness is proportional to the cumulated probability over the cell

The index employed in [9] is based on a Hilbert space-filling curve. We retain here the principle of probabilistic

retrieval and the isotropic Gaussian model, but we change the index structure and, accordingly, the search procedure. The details provided in the next section highlight the differences between [9] and the solution we put forward.

Our current application—monitoring the video streams of TV channels—does not require immediate answers to individual queries. Copy detection must indeed run at least in real time for a stream, but *deferred* real time is perfectly acceptable. This allows us to perform a sort of batch processing: we accumulate a large amount of queries (signatures from candidate videos), depending on the available RAM, on the size of the database and on disk latency, and process all these probabilistic queries on parts of the database that we progressively load in memory. It is nevertheless easy to adapt our architecture in order to process as fast as possible individual (or small sets of) queries for applications needing similarity-based retrieval or navigation.

New challenges for scalability. The largest reference content database on which the method in [9] was tested and achieved (deferred) real-time performance had 30,000 hours of video, represented by 1.6×10^9 signatures. To our knowledge, this was the largest database on which a CBCD method was evaluated. However, the video stream monitoring applications at the Institut National de l'Audiovisuel (whose main mission is to collect and store French radio and television broadcasts) will soon have to deal with databases of 250,000 hours of video, with a target of 1,000,000 hours. Also, the number of streams that should be monitored at the same time continuously increases. All this sets a new level of demand for the scalability of content-based video copy detection methods.

3. GRID-BASED INDEX AND COMPONENT-WISE PROBABILISTIC SEARCH

To further reduce the cost of retrieval with respect to the method in [9], we explored three ideas:

1. To speed up the computation of the keys (cell addresses) we wanted to make it simpler and link it more strongly with a component-wise search process; a Z-grid appeared as the most natural choice (Sections 3.1 and 3.2). This has a positive impact both on the cost of retrieval and on the cost of index construction.

2. The component-wise exploration of the description space encouraged us to begin with the most "convenient" components, i.e. those allowing the fastest elimination of unpromising space regions. In Section 3.3 we suggest to follow the order of decreasing uniformity of the projections of the signatures on the individual dimensions.

3. A good balance between the populations of the cells has a positive impact on the cost of retrieval, but should be compatible with the component-wise exploration and be achieved with as little additional complexity as possible. We attempt to find a compromise by partitioning along each dimension adaptively, but independently of other dimensions (Section 3.4).

As shown in Section 4, these improvements bring a significant speed up of similarity-based probabilistic retrieval.

3.1 Z-grid-based indexing

To find the potential original documents from which a candidate video could have been obtained, we perform probabilistic retrieval using signatures from the candidate video as queries. Space is partitioned into cells and probabilistic retrieval consists then in selecting a minimum number of cells such that their cumulated probability (according to an isotropic Gaussian model) is above the threshold considered acceptable for copy detection.

In [9] the description space ($[0, 255]^{20}$) is hierarchically partitioned into hyper-rectangular cells following a Hilbert space-filling curve, as shown in Fig. 5. At the top levels of the hierarchy, the dimensions of the description space are partitioned one after the other; an interval $[0, 255]$ is divided into 2 equal parts, $[0, 127]$ and $[128, 255]$ (first-order partitioning). At the lower levels, the *resulting spatial partitions* are further segmented by dividing intervals into equal parts (second and higher-order partitioning). At depth h in the hierarchy there are 2^h cells. In the index obtained and in the associated database, the resulting cells are ordered according to the Hilbert curve. Probabilistic retrieval follows the partitioning hierarchy: level after level, a spatial partition is discarded if its probability (1) is too low.

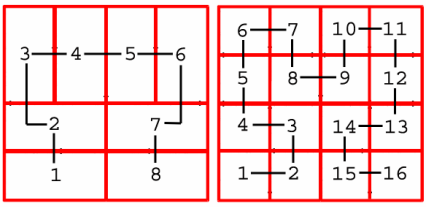

Figure 5: 2D space partitioning following the Hilbert curve at depth 3 (left) and 4 (right)

The Hilbert curve was originally chosen because it could guarantee that two cells that are neighbors in the index are also neighbors in the description space. However, this property is not required for the correctness of probabilistic retrieval, nor does it appear to make the retrieval process more efficient. Moreover, for high-dimensional spaces and higher-order partitioning it becomes difficult to compute the key in the index starting from the position in description space; the Butz algorithm employed is complex for high-dimensional description space and higher-order partitioning. Further difficulties arise from the fact that, as seen in Fig. 5, when the partitioning depth exceeds the number of dimensions, all the cells are not partitioned along the same dimensions.

To simplify the computation of the keys (cell addresses in the index) and to link it more strongly with a component-wise search process, we replace the Hilbert curve by the Z space-filling curve and we hierarchically partition the description space into hyper-rectangular cells following the Z-curve. Since cell order on the Z-curve is not used for controlling search but only for building the index and the associated database, in the following we refer to this indexing scheme as *Z-grid index*. Fig. 6 shows the corresponding space partitioning and highlights the fact that, whatever the depth, all the cells are partitioned along the same dimensions. The direct association between individual dimensions and levels in the hierarchy also allows us to put forward improvements based on 1-dimensional analyses (see 3.3 and 3.4).

The key of the cell containing a given signature is computed by following the partitioning hierarchy from top to bottom and finding, at every level, the appropriate value for a new bit in the key (going from the MSB to the LSB). Then, for a partitioning depth, the key of a cell is the binary value of the cell position along the Z-curve. Extension to higher-order partitioning is straightforward and has no negative impact on complexity. Fig. 7 shows the keys for a partitioning depth of 3 in 2D (as in Fig. 6).

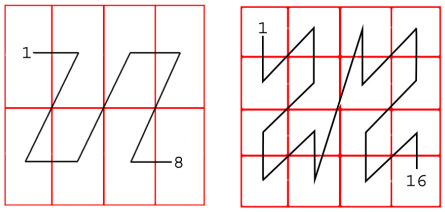

Figure 6: 2D Z-grid space partitioning at depth 3 (left) and 4 (right)

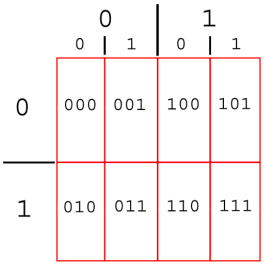

Figure 7: Computation of the keys at depth 3

For indexing based on the Hilbert curve, it was empirically found in [9] that the optimal depth h^* (for the fastest retrieval) was given by $h^* = \log_2 N$, N being the number of signatures in the reference database. In practice, using this value as an initial guess, we produce a few indexes at different depths and after some retrieval tests we retain the one providing the fastest retrieval.

Once the partitioning depth h^* is selected, to build the indexed database file we sort all the signatures of the reference database in ascending order of their key. Every line of the associated index file is a key and contains the line number in the indexed database file where starts the storage of the signatures belonging to the cell having this key. Table 1 gives an indication of the relation between the size of the reference content database and the sizes of the reference signature database and of the index. Note that index size doubles when partitioning depth increases by 1.

3.2 Z-grid-based retrieval

The retrieval of the signatures that are similar to a query is performed in two steps. We begin with a hierarchical search of cells such that their cumulated probability (following the model centered on the query) is above a threshold P_α selected for reliable copy detection. In the second step we scan all the signatures from these cells and filter out those that are too far from the query. The signatures retained identify the original videos to which the candidate will be matched in order to decide whether it is a copy or not.

Table 1: Size of the index for several sizes of the reference content database

Hours of video	Signatures	Database size (Gb)	Index size (Gb)
$10,000$	560×10^6	17	> 4
$60,000$	3.36×10^9	102	> 15
$120,000$	6.7×10^9	204	> 30

For the first step we use probabilistic queries based on an isotropic Gaussian model of the effects of tolerated transformations because, according to [8, 9], the probability of a transformation decreases when its "amplitude" increases. An ϵ-query, erroneously assuming that the distribution of transformed signatures is uniform within a hypersphere around the original signature, would give equal importance to all the neighbors within the range and thus return too many cells, with a negative impact on performance.

For hierarchical probabilistic search we need the minimal cumulative probability a cell should have in order to be retained during the search process. This calibration operation is carried out prior to the retrieval operations for copy detection. It consists in an iterative search procedure evaluating a random sample of queries and finds the new threshold P_c for the probability of a cell that guarantees that the cumulative probability of all the cells retained is above the threshold P_α needed for reliable copy detection.

Hierarchical probabilistic search is performed as follows: at every level from 1 to h^* we divide existing spatial partitions in 2 and update the probability of each new partition. If the probability of a partition is lower than the threshold P_c, we stop the search in the corresponding branch. We end up with a set of cells, each having the same volume $V/2^{h^*}$ (V being the volume of $[0, 255]^{20}$) and a probability $\geq P_c$.

The cells retained by the first step may contain many signatures that are too far from the query to be potential originals for this query. The second step filters out those signatures whose distance to the query is above a *reference range* (depending on the Gaussian model of the transformations). Moreover, if the number of signatures within this range is too high, then the query cannot be considered discriminant for the final decision; rather than just drop this query from the subsequent decision process, we prefer to keep it but only return the kNN signatures found.

The lower bound of the time complexity of the first step is logarithmic in N (the size of the reference database) and, since the selectivity of the index is relatively high, we consider that with an appropriate choice of the depth h, complexity remains sub-linear in N. This will be confirmed in Section 4. In the current implementation, the complexity of the second step is linear in the number of signatures in the selected cells; for retrieval to be systematically fast, both the mean value and the variance of this variable (for a diversified set of queries) should be small. From Table 1, one can note that for 560×10^6 signatures the partitioning depth is 29, meaning that we have $2^{29} = 536,870,912$ cells, almost as many cells as signatures. However, as shown in Fig. 8 for a database of 10,000 hours of video and a depth of 20, the distribution of the signatures in the description space is

far from being uniform. We believe this is a consequence of both the high redundancy within some types of videos, such as weather forecast reports or news shows, and the characteristics of the local descriptors employed (they were nevertheless designed to increase uniformity).

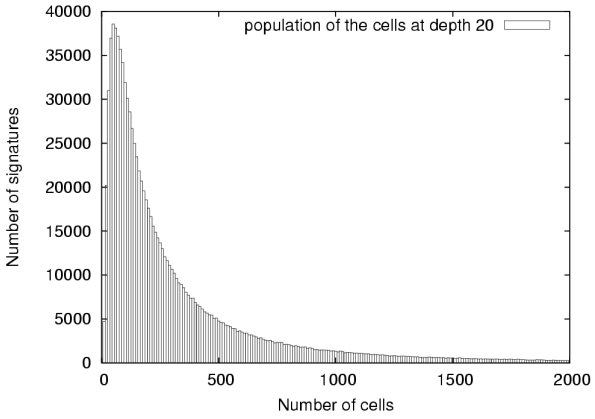

Figure 8: Number of cells as a function of their population of signatures

3.3 Sorting the components

Given the fact that data distribution is non-uniform, one may expect data-partitioning methods to provide better results than space-partitioning. In the context of this work, there are however several arguments in favor of our approach and we mention some of them very briefly. First, the signatures are relatively high-dimensional and, while data distribution is non uniform, there is no large empty region in the description space; these factors are known to have a strong negative impact on the selectivity of data-partitioning methods. Then, the hierarchical probabilistic search (first and potentially most costly step of retrieval) can be performed without any access to the index or database files. Both files are only needed during the second step, and the index file has a relatively small size. Last but not least, the retrieval process can be easily parallelized both by query and by space partition, providing more flexibility to an implementation on a cluster. Data-dependent space-partitioning methods, such as the well-known KDB-tree or the more recent LSDh-tree [6], were also devised for non-uniform data distributions. We considered that the potential gain did not compensate for the additional complexity introduced in the retrieval algorithm. Moreover, the complexity of the construction of a Z-grid-based index is $O(n \log(n))$, while for the LSDh-tree it is $O(d\,n \log(n))$ (d being the dimension of the description space). In the next two sections we put forward two improvements to our method in order to mitigate the effects of a non-uniform data distribution, with as little additional complexity as possible.

During the analysis of the 20-dimensional description space we also projected all the signatures on every dimension and displayed the resulting distribution, which was found to be relatively uniform for some dimensions, as in Fig. 9, and very non-uniform for others, as in Fig. 10.

To build the index, the description space is divided along every dimension once or several times. Defining an optimal order between dimensions can improve retrieval for two

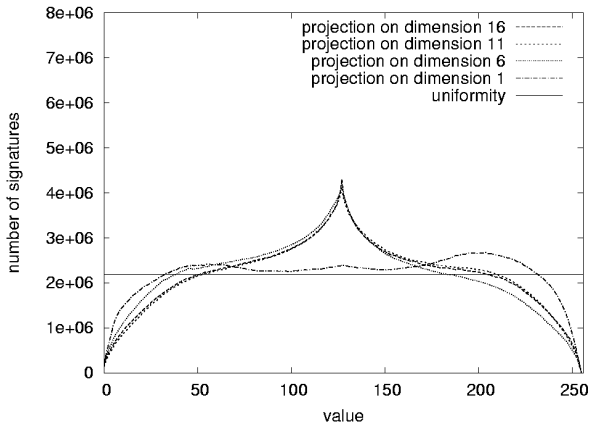

Figure 9: Distribution of the signatures projected on the 4 dimensions corresponding to ∂y for $t-2\delta$, $t-\delta$, t and $t+\delta$

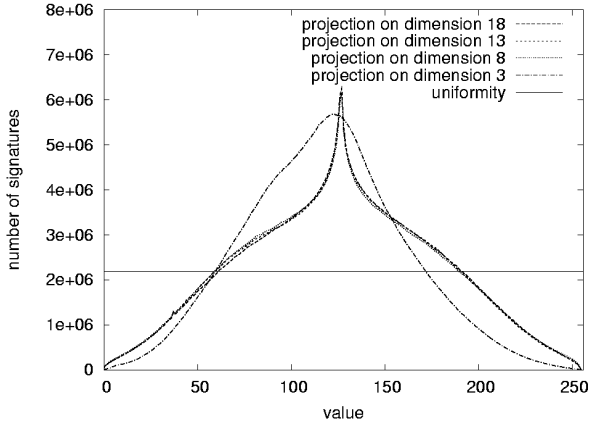

Figure 10: Distribution of the signatures projected on the 4 dimensions corresponding to ∂x^2 for $t-2\delta$, $t-\delta$, t and $t+\delta$

reasons. First, the hierarchical exploration of the 20 dimensions during probabilistic search should consider first those dimensions who, on average, allow for the fastest elimination of unpromising space regions. Exploration should then *begin with* the dimensions along which the distribution of the projected signatures is the most uniform. Second, according to the arguments presented in the previous section, the populations of the cells obtained at the optimal depth h^* should be as balanced as possible. This can be achieved by dividing *most* those dimensions along which the distribution is closest to being uniform. The two arguments converge to the following procedure: sort the dimensions by decreasing order of their *uniformity* and divide them in this order. The criterion we use for defining uniformity is $U_j = \sum_0^{255} |d_{ij} - \overline{d}_j|$, where d_{ij} is the number of signature whose orthogonal projections on dimension j fall in $[i, i+1)$ and $\overline{d}_j$ is the mean value of d_{ij} along dimension j. This is the L_1 distance between the actual histogram along dimension j and the histogram of a uniform distribution. The estimation of the uniformity was performed on 1,000 hours of randomly selected videos (but we found the estimates stable above 100 hours).

3.4 Adapting the boundaries

To further balance the populations of the cells with as little additional complexity as possible, we fit the partitioning along each dimension to the distribution of the signatures while keeping it independent of the other dimensions. Rather than systematically choosing the *middle* of the interval, we divide according to the *median*. Fig. 11 shows an example of this new partitioning at first order and then second order, for a dimension with high uniformity and a dimension with low uniformity (if needed).

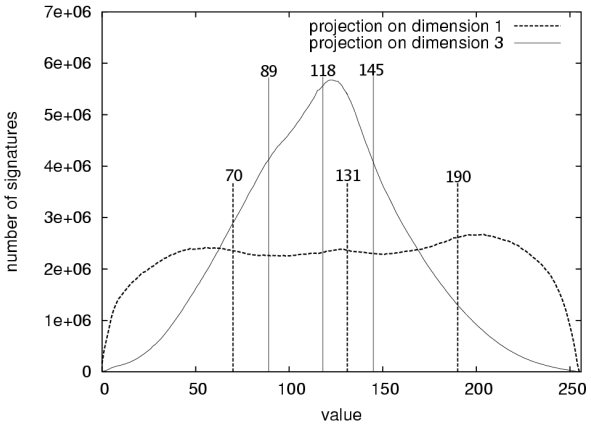

Figure 11: Median and quartiles for projections on dimensions 1 and 3

4. EXPERIMENTAL EVALUATION

Since the contribution presented above regards the index structure and the associated search method, the evaluation results provided here do not concern the entire copy detection process but rather the speed up in probabilistic retrieval for a fixed threshold $P_\alpha = 0.8$ considered acceptable for reliable copy detection (this value was used in [9, 7]).

4.1 Experimental setup

The reference content database we used for the evaluations contains 120,000 hours of MPEG1 video (352 × 288) provided by the Institut National de l'Audiovisuel (France). All types of television programs are present in this database. To see whether the complexity of the retrieval process is sublinear in the size of the database, we also use two smaller databases of 10,000 and 60,000 hours of video, obtained by a random selection of programs from the bigger database.

For each of the 3 databases, to obtain the set of queries we first perform a uniform random selection of 10,000 points of interest describing original videos. Then, a random modification following a 20-dimensional isotropic Gaussian law of $\mathbf{0}$ (vector) mean and a standard deviation $\sigma = 20$ is applied to the signature of each of these point of interest. This value for the standard deviation is estimated from a set of real transformations applied to original videos. Naturally, the same Gaussian law is used for the first step of the probabilistic retrieval, consisting in the hierarchical identification of cells having a probability of at least P_c for a given query. For the second step of the probabilistic retrieval (keep, in the selected cells, only the signatures that are close enough to the query) we employ a range of 6.7σ and a value of 100 for k. To allow a direct comparison, the same parameters are used for the method in [9, 7]. All our retrieval experiments were performed on the same PC, with a Xeon64 CPU at 3 GHz, 2 Gb of RAM, under Linux.

4.2 Evaluation results

We could only perform the comparison with the method in [9, 7] on the 10,000 hours database. Figure 12 shows the results with a partitioning depth of 29, considered optimal for the method in [9, 7] using the Hilbert curve, and with a partitioning depth of 31, optimal for the method put forward here. Our method based on the Z-grid, with sorted components and adapted boundaries, is almost 6 times faster than the one in [9, 7], using the Hilbert curve.

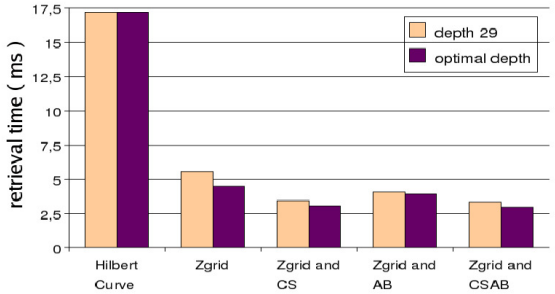

Figure 12: Comparison of mean retrieval times (miliseconds) for 1 query on the 10,000 hours database (CS stands for Components Sorting, AB for Adapted Boundaries)

Figures 13 and 14 display the results obtained with the Z-grid alone and with components sorting plus adapted boundaries, on the 60,000 and on the 120,000 hours databases. The larger the database, the stronger is the positive effect of the 2 improvements over the use of the Z-grid alone. When the populations of the cells are more balanced (with Z-grid plus CSAB), the depth of partitioning has a smaller impact on performance, so one can employ a smaller depth to reduce index size. From Fig. 15 we see that the complexity of the retrieval process is sublinear in the size of the database. Finally, we can mention that the speed up obtained enable us to monitor in (deferred) real-time one TV channel against the 120,000 hours database with a single PC like the one used in our experiments.

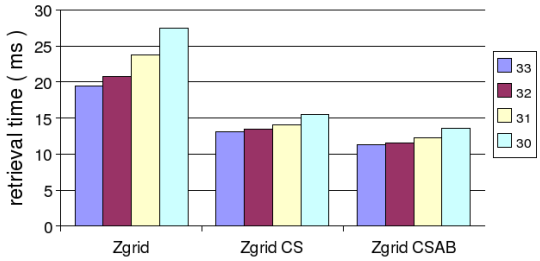

Figure 13: Comparison of mean retrieval times (ms) for 1 query on the 60,000 hours database

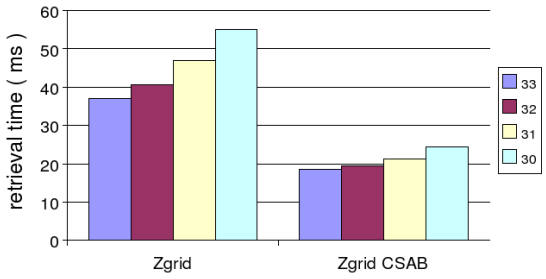

Figure 14: Comparison of mean retrieval times (ms) for 1 query on the 120,000 hours database

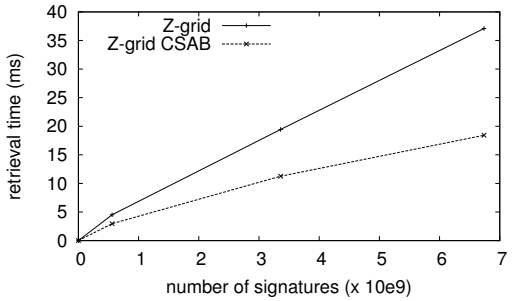

Figure 15: Retrieval time increases sublinearly with the size of the database

5. CONCLUSION

While content-based copy detection methods for still images and video show significant robustness to image transformations and do not depend on the presence of watermarks in the original content to be protected, their computational complexity also poses a scalability challenge. As an example, the Institut National de l'Audiovisuel will soon have to monitor a continuously increasing number of video streams against databases of more than 250,000 hours of original video programs. To go past the performance of existing methods, such as [7], we introduce here three improvements of the similarity-based probabilistic retrieval process: replacement of the Hilbert curve with a Z curve, uniformity-based sorting and adapted partitioning of the components. We show that these improvements provide a significant speed up in retrieval, enabling 1 PC to monitor one TV channel against a 120,000 hours database. Also, the complexity of the retrieval process is sublinear in the size of the database.

6. ACKNOWLEDGMENTS

The authors acknowledge the interesting discussions with Alexis Joly, Julien Law-To and Michel Scholl. The first author is partly financed by the ANRT.

7. REFERENCES

[1] S.-A. Berrani, L. Amsaleg, and P. Gros. Robust content-based image searches for copyright protection. In *Proc. 1st ACM intl. workshop on Multimedia Databases (MMDB'03)*, pages 70–77, New Orleans, USA, 2003. ACM Press.

[2] N. Boujemaa, J. Fauqueur, and V. Gouet. What's beyond query by example? In R. V. L. Shapiro, H.P. Kriegel, editor, *Trends and Advances in Content-Based Image and Video Retrieval*. Springer Verlag, 2004.

[3] E. Chang, J. Wang, C. Li, and G. Wilderhold. Rime - a replicated image detector for the world-wide web. In *Proceedings of SPIE Symposium of Voice, Video and Data Communications*, pages 58–67, November 1998.

[4] T. Gevers and A. W. M. Smeulders. Content-based image retrieval: An overview. In G. Medioni and S. B. Kang, editors, *Emerging Topics in Computer Vision*, chapter 8. Prentice Hall, 2004.

[5] A. Hampapur, K. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. In M. M. Yeung, C.-S. Li, and R. W. Lienhart, editors, *Proc. Conf. on Storage and Retrieval for Media Databases*, pages 194–201, December 2002.

[6] A. Henrich. The LSDh-tree: An access structure for feature vectors. In *Proceedings of the 14th International Conference on Data Engineering (ICDE'98)*, pages 362–369, Washington, DC, USA, 1998. IEEE Computer Society.

[7] A. Joly, O. Buisson, and C. Frélicot. Content-based copy detection using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*, 9(2):293–306, 2007.

[8] A. Joly, C. Frélicot, and O. Buisson. Robust content-based video copy identification in a large reference database. In *Intl. Conf. on Image and Video Retrieval (CIVR'03)*, pages 414–424, 2003.

[9] A. Joly, C. Frélicot, and O. Buisson. Discriminant local features selection using efficient density estimation in a large database. In *Proc. 7th ACM SIGMM intl. workshop on Multimedia Information Retrieval (MIR'05)*, pages 201–208, New York, NY, USA, 2005. ACM Press.

[10] Y. Ke, R. Sukthankar, and L. Huston. An efficient parts-based near-duplicate and sub-image retrieval system. In *Proceedings of the ACM international conference on Multimedia*, pages 869–876, 2004.

[11] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels of behavior for video copy detection. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 835–844, New York, NY, USA, 2006. ACM Press.

[12] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp. Advances in digital video content protection. *Proceedings of the IEEE*, 93(1):171–183, January 2005.

[13] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[14] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision (ECCV'02)*, pages 414–431, London, UK, 2002. Springer-Verlag.

[15] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535, 1997.