Scalable Mining of Large Video Databases Using Copy Detection

Sébastien Poullot
Institut National de l'Audiovisuel and CEDRIC - CNAM, France
spoullot@ina.fr

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

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

ABSTRACT

Mining the video content itself can bring to light important information regarding the internal structure of large video databases, compensating for a lasting absence of extensive and reliable annotations. Many valuable links between video segments can be identified by *content-based copy detection* methods, where "copies" are transformed versions of original video sequences. To make this approach viable for large video databases, we put forward a new mining method relying on the definition of a compact keyframe-level descriptor and of a specific index structure. The performance obtained in detecting links between video segments is evaluated with the help of a ground truth and several illustrations are given. The scalability of the approach is then demonstrated for databases of up to 10,000 hours of video.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.2.4 [**Systems**]: Multimedia databases; H.2.8 [**Database applications**]: Data mining

General Terms

Design, Performance

Keywords

Video mining, content-based copy detection, similarity join

1. INTRODUCTION

Given the exponential growth of institutional and user-generated multimedia archives, organizing these archives is an important challenge for the next years. Making explicit the internal structure of a large multimedia database supports content acquisition and management, content retrieval using various criteria and content preservation. Traditionally, the organization of multimedia content relies almost exclusively on textual metadata (keywords, descriptive notes). The semantic level of such metadata is appropriate for answering user queries and its low volume makes this approach easily scalable to large databases.

But the use of textual metadata has significant limitations. First, the direct provision of extensive and reliable metadata can not parallel the growth in content volume. National archives who collect, register the copyright, annotate and preserve radio and television broadcasts (like the Institut National de l'Audiovisuel, INA, in France) usually record very large volumes of data every day (for INA, about 1,000 hours daily in 2008) and cannot perform massive professional annotation tasks. User Generated Content (UGC) websites attempt to obtain relevant metadata by letting the large public collaboratively provide keywords or thematic classifications for the content. But only the most popular content gets annotated and the quality of these annotations is typically low (noise, spelling errors, ambiguous terms, etc.). Furthermore, the answers to a query are cluttered with countless versions of roughly the same content. Then, language and cultural barriers can significantly restrain the interest of both user-generated and professional annotations for content organization and retrieval.

A complementary approach consists in extracting information from the multimedia content itself. Regarding video content, which is our focus here, content-based mining methods can support many applications for institutional or user-generated multimedia archives and there have been recently many developments in this domain. Most of them address the problem of tracking news subjects by looking for near-exact copies [8], [15] or for less similar representations of the same scene or event [9], [11], [16], [12], [14]. Others focus on TV commercials [4] or on the elimination of near-duplicates in web search [13]. Some of these proposals are further considered in Section 2, together with an extended range of applications of content-based mining.

For the identification of content links between video sequences, supporting the range of applications described above, Content-Based Copy Detection (CBCD) is a very relevant tool. Actually, most of the recent video mining developments just mentioned are CBCD-related methods. By copies we understand potentially transformed versions of original video sequences. The transformations belong to a large family and their amplitude varies significantly (e.g. Fig. 1). But CBCD methods that are robust to a wide range of transformations are also computationally expensive, and the cost of Video Mining by content-based Copy Detection (VMCD in the following) is even higher. This explains why in most cases the domain is narrow (e.g. news shows), the volume of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-303-7/08/10 ...$5.00.

data is reduced and/or the amplitude of the transformations taken into account is strongly limited (near-exact copies).

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

We address here the scalability problem for VMCD, considering generic video content and a relatively large family of transformations of higher amplitude, as in Fig. 1. The goal is to find all the occurrences of short (a few seconds) video excerpts from any video document of a large database. The definition of a new keyframe descriptor and of an adapted index structure allows us to mine a 10,000 hours video database in less than 83 hours. The mining process produces a graph having as nodes all the video sequences that occur more than once (with various transformations) in the database, while the edges are the links found between such occurrences. By further processing this graph, different types of structural components can be identified in the video database, which supports several applications.

Section 2 considers possible applications of video mining and selected relevant work; then, a new mining approach is introduced and motivated. This new approach is presented in Section 3. First, a new keyframe description scheme, "Glocal", is put forward; it embeds the set of local signatures describing a keyframe in a fixed-size and compact binary vector. Then, a new solution is proposed for indexing Glocal signatures to support mining operations. The entire mining process producing the content links between video sequences is described. This mining approach is assessed in Section 4: the precision in detecting the content links is measured on a ground truth, then the results obtained on two specific databases are illustrated and the scalability is evaluated on larger databases of up to 10,000 hours of video.

2. CONTENT-BASED VIDEO MINING

The early approaches to video mining in [8] and [15] focus on news shows and develop CBCD solutions. While the studied databases are not very large and the descriptors employed are rather compact (limiting the range of transformations that are compatible with the detection of copies), they demonstrate the relevance of CBCD for video mining. Subsequent proposals for mining databases of news shows put forward methods that do not actually aim at the detection of transformed videos. In some cases the occurrences of the same *scenes* (shot by different cameras, from possibly very different viewpoints) are detected, either by using the discontinuities in the trajectories of interest points [9] or by employing flash patterns [11]. The video *sequence* descriptors are very compact, so relatively large video databases can be processed, but the application of these solutions is limited to rather specific video content. Others focus on tracking news *subjects* by using both video keyframe similarity and automatic audio transcriptions [16], [12], [14]. The keyframe descriptions employed allow to some extent to find views of a same scene from different angles, but rather small databases are processed.

For large archives like INA, the identification of content links between video sequences would allow to segment the content and to extend textual metadata from one video sequence to others (while avoiding to separately annotate several versions of the same content). It further supports such services as visual navigation in the database, broadcast programming analysis or media impact evaluation that often helps deciding which video should be annotated. For example, very short sequences that are near-identical copies usually correspond to broadcast design sequences; they allow to identify the channel, then to segment the broadcast and identify individual shows, thus providing annotations and supporting data management and retrieval. Longer and frequent sequences that are also highly similar typically represent TV commercials or, with a different temporal pattern, brief flashes from news agencies; they can provide information for media impact evaluation. Longer, infrequent and more strongly transformed sequences correspond to reused excerpts from shows or movies; they provide information about the type of program, help the transfer of annotations from one program to another and also support broadcast programming analysis.

UGC websites can also exploit content links between video sequences for specific purposes. Since the uploaded content usually consists of copies of video broadcasts rather than original, user-created content, there are many near-duplicates in the stored content, at various levels of quality, with different segmentations and quite different annotations. The lower-quality near-duplicates of a same video sequence can be removed and the links established between duplicate segments in longer sequences can support advanced navigation. Also, the textual metadata associated to the remaining sequence can be improved (richer annotations with less noise) by exploiting the metadata of the near-duplicates. A related application, proposed in [13], is the elimination of video near-duplicates from the results returned by a Web search engine. Inexpensive descriptors are used to separate the least similar videos, then local descriptors allow to refine duplicate detection. Even with this improvement, the removal of duplicates remains computationally intensive.

The video mining stage in [10] allows in principle to identify content links at a sub-frame level, e.g. same or similar objects in different scenes. However, the size and number of descriptors per frame, the cost of matching and tracking, together with the very large number of links this approach can generate, significantly reinforce the scalability challenge. For the case where it is applied to CBCD, the scalability of this description scheme was studied in [3]. A solution inspired by association rule mining was proposed in [6] for finding the *frequent* itemsets (sets of local descriptors with additional information regarding their spatial configuration) and the corresponding content links in a video.

The above-mentioned applications of VMCD have different requirements in terms of length of the detected sequences, of nature and amplitude of the transformations applied and of the particular categories of video programs that should be analyzed. Rather than developing entirely specific mining frameworks for specific applications and/or type of content (which is not always easy to define), it can be useful to have a single mining framework that is able to deal with

a very large volume of general video content and find links between substantially transformed sequences. The result of mining is a large graph having as nodes all the video sequences that occur more than once (with various transformations) in the database, while the edges are the links found between such occurrences. False detections can significantly overload the graph and connect many disconnected components; good precision (low rate of false detections) is thus even more important for mining than for CBCD. The information required by specific applications can be obtained by processing the graph. Clearly, the scalability challenge is stronger when considering a large volume of general (non specific) video content and a wide range of transformations (gamma and contrast changes, filtering, cropping, scaling, insertion of logos or frames, addition of noise) with potentially high amplitude (see Fig. 1). This is the challenge we address in the following.

Approaches to VMCD. A natural solution would be to consider a general and scalable method for content-based copy detection and apply it to the mining problem. CBCD methods usually have an off-line indexing component and an online detection component. The indexing component computes the signatures for all the keyframes in the database of original video content, then stores these signatures in a reference database and creates an index supporting fast similarity-based retrieval from this reference database. The detection component extracts keyframes from a video stream and computes their signatures, retrieves similar signatures from the reference database and decides whether the similarity between sequences of keyframes is sufficient for the input sequence to be considered a copy of an original sequence.

To directly apply a CBCD method to mining, the reference database and the corresponding index must be created first, then the signatures of every keyframe are used to query the database and potential copies are identified.

The CBCD method described in [5] can monitor in *deferred real time*, with one PC, one TV channel (video stream) against a database of 120,000 hours of video. It copes with the set of transformations mentioned above (with amplitudes found by studying real copies of videos stored at INA) and detects about 85% of the transformed sequences of at least 3 seconds with good precision. This is the fastest solution we know about for such large databases. Recent improvements allowed to obtain the same performance in terms of speed and recall with a larger database of 280,000 hours of video (860×10^6 keyframes, 16.4×10^9 local signatures). But the direct application of this method to mining would still be unacceptably slow since it would take about 3.5 years to mine the 280,000 hours database with 1 PC. This motivates the new framework described in the next section.

To facilitate the comprehension of the new framework, the proposal in [5] and its recent improvements must be briefly described. Videos are represented in [5] as sequences of keyframes, where each keyframe is described by a set of local spatiotemporal signatures of relevant interest points. To find out whether a keyframe in the monitored stream is a copy (transformed version) of a keyframe in the database, all the signatures describing the candidate keyframe are used as queries and the returned signatures from the reference database take part to a vote-based decision process. A coarse Z-grid index with component-wise probabilistic retrieval is proposed in [5] in order to provide the answers to these similarity queries at a low cost.

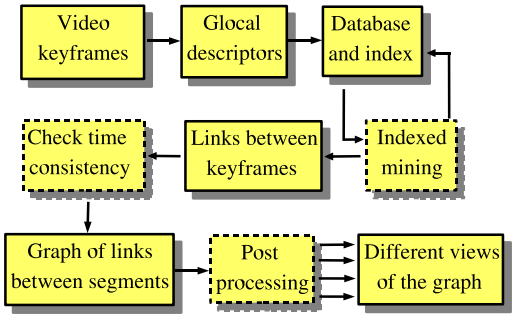

Figure 2: Proposed workflow for VMCD

3. VIDEO MINING FRAMEWORK

The problem we address can be described as a self-join operation on a database of video sequences, returning all the pairs of sequences that satisfy the selection criteria ("matching" sequences). The primary criterion is similarity, but other criteria can be added, depending on the data or application (e.g. exclude multiple occurrences that are close together in a short time interval, or only consider one type of broadcasts). When the additional criteria are more selective than similarity, they should be employed first to simplify the problem. To remain as general as possible, we only use one such criterion (avoid temporal proximity in a stream, see subsection 3.3) that it is not critical to the approach.

Rather than directly looking for similar sequences, the mining process suggested here first retrieves the pairs of similar keyframes and then employs them to find similar sequences (see Fig. 2). This solution brings more flexibility to the last stage of mining and can be readily used to mine databases that contain both images and videos.

To meet the scalability requirements, we (1) define and employ compact frame-level signatures instead of sets of interest point signatures, and (2) build an index to support mining. Compact frame-level signatures—"Glocal" in the following—allow to directly compute similarity at the level of frames and avoid the large volume of intermediate results (between retrieval by similarity and the vote-based decision) often associated to the direct use of interest point signatures. They provide a better compromise between detection time and quality, and make a redundant index affordable. The frame-level signatures and the redundant index allow to keep the similarity computations *local* in database, minimizing exchanges between main memory and mass storage.

3.1 Glocal description scheme

Good robustness to the typical transformations between original videos and copies is obtained in [5] by the use of local descriptors. Each keyframe is described by the local spatiotemporal signatures of at most 20 (and sometimes slightly less) interest points found by the improved Harris detector. The use of multiple local signatures for every keyframe and of a specific matching provides robustness to cropping or insertions. But this is also expensive in terms of time and storage. It is important to find a *frame-level* description scheme that keeps as much relevant information as possible and allows to include part of the vote-based decision in a simple computation of the similarity between two keyframes, while significantly reducing computation and storage cost. In [5] the spatiotemporal signature of an interest point in keyframe

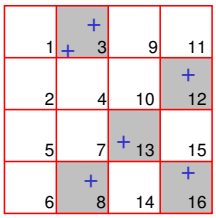

Figure 3: Construction of a Glocal signature in a 2-dimensional description space

t is composed of the normalized 5-dimensional vector of first and second-order partial derivatives of the gray-level brightness for this point and for 3 other neighboring points in the frames $t+\delta$, $t-\delta$ and $t-2\delta$. Such a signature belongs to the 20-dimensional description space $[0,255]^{20}$. The distribution of the signatures covers well this space, so indexing is based on hierarchical partitioning of the description space (not of the image plane) into hyper-rectangular cells. Every local signature is defined by a precise position and falls within one such cell. Only the coarse information given by the numbers of the cells to which the local signatures belong is employed for defining a frame-level signature.

The 20-dimensional description space is partitioned at a limited depth h (at every level, a new interval is partitioned in two), which produces 2^h cells. To every cell a position is assigned (following some numbering scheme) in a binary vector of fixed dimension 2^h. The *Glocal* signature of a keyframe is such a binary vector, where the i-th bit is set to 1 if the local signature of at least one interest point from that keyframe falls within the cell i, or else left to 0. The Glocal signature is a fixed-size embedding of a set of local signatures. A simplified example (with interest point signatures in a 2-dimensional space) is given in Fig. 3, for a keyframe containing 6 interest points. The squares represent the partitioning of the description space (not the image plane) at depth 4. The numbers of the cells are shown in the corners. The local signatures are the + marks within the squares. The resulting Glocal signature is **0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1**.

The information provided by a Glocal signature consists in the positions of the bits set to 1 and is a coarse approximation of what a *set* of local signatures would bring. This description scheme would be inadequate if the local signatures of many interest points belonging to a same keyframe would fall in a same 20-dimensional cell (on average). A very significant amount of information would then be lost. By studying how many bits are set to 1 in a Glocal signature, we obtained an average of 17.4 for a partitioning depth of 8, close to the average number of local signatures in a keyframe. This shows that the distribution is convenient, so the loss of information is limited, and that the partitioning of the space at this depth is already appropriate. Partitioning at a higher depth cannot diminish this average.

Glocal signatures are compact: considering a partitioning depth of 8, every keyframe is described by 2^8 bits (or 32 bytes) with a Glocal signatures, while the description in [5] requires 20 local signatures of 20 bytes each, for a total of 400 bytes. This gain allows the use of a redundant indexing scheme (Section 3.2) that supports faster mining, without increasing the total storage requirements.

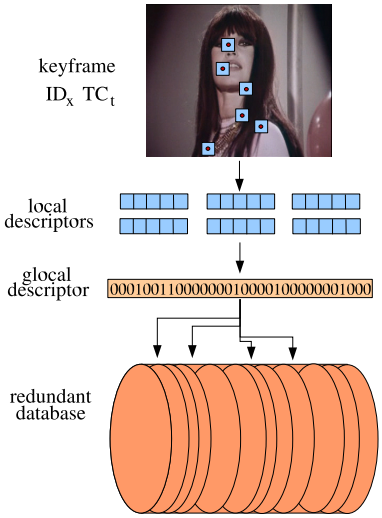

Figure 4: Obtaining the Glocal signature database

Since the Glocal signatures are binary, a natural choice for measuring their similarity is the Dice coefficient (S_{Dice}):

$$S_{\text{Dice}}(\mathbf{g}_1,\mathbf{g}_2)=\frac{2\,|\mathcal{G}_1\cap\mathcal{G}_2|}{|\mathcal{G}_1|+|\mathcal{G}_2|} \qquad (1)$$

where $\mathcal{G}_i$ is the set of positions of the bits set to 1 in the signature $\mathbf{g}_i$ and $|\cdot|$ denotes set cardinality. S_{Dice} is directly related to the Jaccard coefficient. When the number of bits set to 1 is about the same for every signature, S_{Dice} is almost identical to the overlap coefficient and can also be related to the Hamming distance.

3.2 Indexing Glocal signatures

The first stage of the video mining process, that consists in identifying the links between individual keyframes, is the most expensive. If $\mathcal{D}$ is the database of Glocal signatures and θ is the similarity threshold above which one keyframe is considered a transformed version of the other, then the following set should be found:

$$\mathcal{K}_\theta=\{(\mathbf{g}_i,\mathbf{g}_j)\,|\,\mathbf{g}_i,\mathbf{g}_j\in\mathcal{D}, S_{\text{Dice}}(\mathbf{g}_i,\mathbf{g}_j)>\theta\} \qquad (2)$$

This corresponds to a *similarity join* on the database of Glocal keyframe signatures and its time complexity would be $O(N^2)$ if the similarity was computed for every pair of signatures (N is the size of the database).

The efficient computation of similarity joins was addressed in the information retrieval and in the data management literature, e.g. [7], [1], [2]. To find an appropriate solution for speeding up the first stage of the mining process, our prerequisites and the main characteristics of the data should be made explicit. The main requirement concerns the level of scalability: we intend to mine databases of more than 5000 hours of video ($\geq 18\times10^6$ Glocal signatures) much faster than by the direct application of existing CBCD methods. To further reduce the time required for larger databases, the method should fully support a parallel implementation by requiring a limited volume of data exchanges.

As seen in subsection 3.1, the Glocal signatures are compact (2^h bits per frame for a partitioning depth of h) and sparse, but not as sparse as the text descriptors typically employed in information retrieval (e.g. [2]). Furthermore,

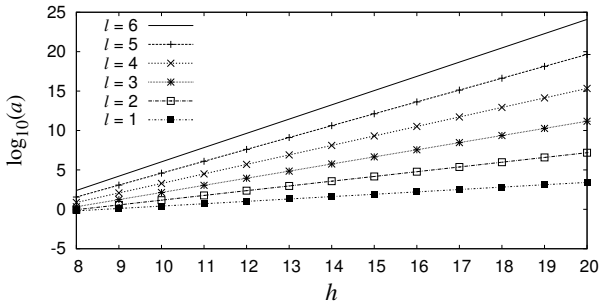

Figure 7: Impact of h and l on speedup

about the same number of bits is set to 1 in every Glocal signature, while for text descriptors the number of non-zero components is usually highly variable. Also, the Glocal signatures cover well the description space, even if the probability of being set is not the same for all the bits. These characteristics of Glocal signatures and of their distribution, together with the requirements mentioned above, do not allow a direct application of the proposals in [2] or [1].

Principle of the indexing scheme. We aim to accelerate the discovery of $\mathcal{K}_\theta$ defined by (2) by avoiding to compute the similarity for every pair of signatures, while keeping storage requirements rather low. The indexing scheme put forward here is inspired by both inverted lists and hashing. It is based on dividing the database of Glocal signatures into segments such that, in each segment, the similarity between any two signatures is above a threshold, and then performing the similarity join independently for every segment. In a large database the signatures are scattered rather than grouped into compact and well separated clusters, so different segments must overlap in order to guarantee that all the links are found; redundancy can be a source of inefficiency if the overlap is too high. An important requirement is that every segment should hold into main memory in order to avoid expensive intermediate disk accesses.

A segment is defined by a specific set of bits set to 1 (called "sentence" below) in the representation of Glocal signatures. The segment ("bucket" next) consists of all the Glocal signatures in the database that contain this sentence; it can be stored as an inverted list. To complete the description of the indexing scheme we must specify (i) at what depth should the description space be partitioned in order to define the Glocal signatures (subsection 3.1), (ii) how many bits the sentences should have, (iii) how many different sentences should be used for indexing a Glocal signature and (iv) how should these sentences be selected.

Speedup estimation. Let N be the total number of signatures in the database. If the similarity is evaluated for every pair of signatures, the total number of similarity computations is $\frac{N(N-1)}{2} \approx \frac{N^2}{2}$. A comparison with the index-based solution will show the impact of the partitioning depth h and of the length l of the sentences. The number of bits set to 1 is about the same for every Glocal signature, in our case ≥ 19 for $h \geq 8$, and is upper bounded by the maximum number of interest points per keyframe (20 here), so it does not increase with h for $h \geq 8$; it will be denoted by L. It follows that the length of every signature is 2^h, the total number of different buckets (possible sentences) is $\binom{2^h}{l}$ and every signature is present in $\binom{L}{l}$ buckets. If all the bits are set to 1 with the same frequency for the signatures in the database, then all the buckets have the same size, equal to $N\binom{L}{l}\binom{2^h}{l}^{-1}$. Consequently, the number of similarity computations performed with the index is approximately $\binom{2^h}{l}\frac{N^2}{2}\binom{L}{l}^2\binom{2^h}{l}^{-2} = \frac{N^2}{2}\binom{L}{l}^2\binom{2^h}{l}^{-1}$. The estimated speedup a obtained by using the index is then

$$a = \binom{2^h}{l}\binom{L}{l}^{-2} \tag{3}$$

The space required for storing a Glocal signature and the time needed for computing the similarity between two signatures can then be considered fixed and independent of h. As shown in Fig. 7 for $l \in \{1, \ldots, 6\}$ and $h \in \{8, \ldots, 20\}$ (with $L = 20$), the speedup increases with both l and h. But the storage requirements are $N\binom{L}{l}$, so they augment when l grows from 1 to 10 and then decrease. Taking for l a value between 15 and 20 would make the similarity for two signatures in a same bucket higher or equal to $\frac{2\times 15}{20+20} = 0.75$, according to (1), which would severely restrict recall. Moreover, when h increases (the partitioning of the description space is stronger) the similarity between a keyframe and a transformed version of this keyframe will diminish. The similarity threshold θ used in (2) for establishing a link between two keyframes could be reduced accordingly, but this would augment the overlap between true positives and true negatives. For these reasons, in the system developed here both the sentence length l and the partitioning depth h are close to their lower bounds shown in Fig. 7.

To further save computation time and storage, it is possible to reduce both the number of buckets and their size by assigning each signature to a relatively small share of the $\binom{L}{l}$ buckets it can belong to. This can be performed by defining rules for selecting sentences (sets of bits set to 1) in a signature; then, all the signatures containing a given sentence are assigned to the corresponding bucket.

Bucket selection. The rules considered here for selecting sentences are: neighboring bits (not separated by any other bit set to 1), 1-out-of-2 bits (separated by only one bit set to 1), 1-out-of-3 bits and 1-out-of-4 bits. As an example, for the Glocal signature **0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1**, the positions of the bits set to 1 are 3, 8, 12, 13 and 16, so the sentences of length 2 of neighboring bits are 3-8, 8-12, 12-13 and 13-16, those of 1-out-of-2-bits are 3-12, 8-13 and 12-16, those of 1-out-of-3-bits are 3-13 and 8-16, while the sentence of 1-out-of-4-bits is 3-16. The reduction of the number of buckets to which a signature is assigned is significant: e.g. with $L = 20$ and $l = 3$ every signature is only assigned to 48 of the $\binom{20}{3} = 1140$ buckets in which it would otherwise be present. But if this reduction is too strong, some pairs of signatures corresponding to a keyframe and to its transformed version may no longer be placed together in *any* remaining bucket, so only part of $\mathcal{K}_\theta$ would be found.

To evaluate these rules and find appropriate values for h and l, we explore their impact on an *automatically* generated set of transformed videos ("copies"). Since the available ground truths are small, they are *only* employed for the final evaluation in Section 4. Instead of using a ground truth, we automatically generate a large set of copies, compute the Glocal signatures of the original keyframes and of their transformed versions, then measure the impact of different parameters on the quality of copy detection. The transformation parameters should exceed the ones already

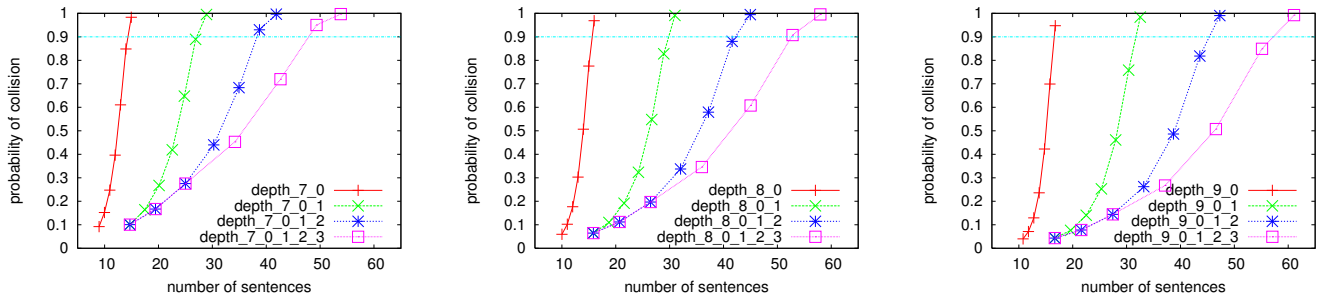

Figure 5: Estimated probability of collision between the Glocal signature of an original keyframe and that of an automatically generated copy, obtained at depth 7 (left), 8 (middle) and 9 (right), respectively.

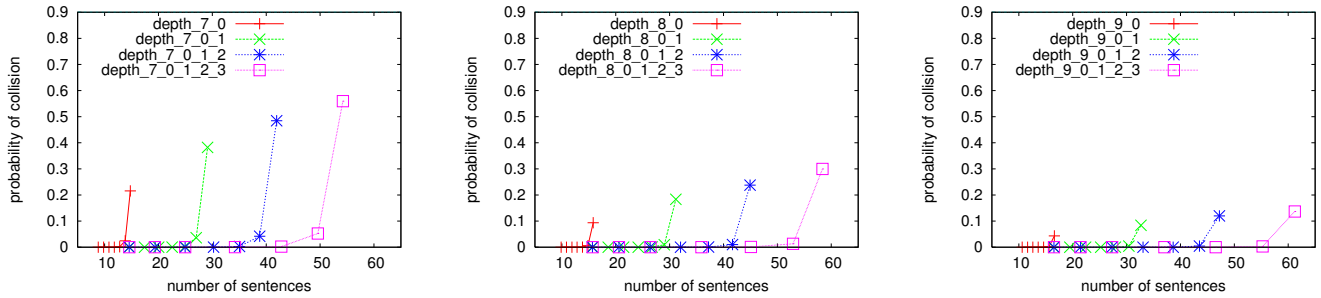

Figure 6: Estimated probability of collision between the Glocal signatures of two random keyframes, obtained at depth 7 (left), 8 (middle) and 9 (right), respectively.

observed but remain within reasonable bounds with regard to visual perception. From a diverse video collection of 300 hours, 100 hours of copies are automatically generated, using combined transformations including gamma and contrast changes, scaling and addition of Gaussian noise.

Collision analysis. Using this large dataset, we estimate the probability for the Glocal signature of an original keyframe to "collide" (i.e. be assigned to at least one bucket together) with the signature of an automatically generated copy; the results are shown in Fig. 5 for $h \in \{7, 8, 9\}$. Fig. 6 displays the results obtained when the estimation is performed for randomly selected pairs of signatures (moreover, two such keyframes are always taken from different broadcasts). Each curve on the graphs corresponds to the addition of one more selection rule: "0" stands for neighboring bits only, "0_1" for neighboring bits and 1-out-of-2 bits, "0_1_2" adds the selection of 1-out-of-3 bits and "0_1_2_3" adds the selection of 1-out-of-4 bits. Each point on a curve corresponds to a value for the length of the sentences: 2 for the point at the top, 3 for the point below, and so on up to 8 for the point at the bottom. The abscissa represents the total number of different sentences obtained for a signature.

The similarity between two signatures is computed and compared to the threshold θ only if the two signatures collide. When keyframes are compared to their copies (Fig. 5), the estimated probability of collision should be as high as possible; a value of 1 would guarantee that all of $\mathcal{K}_\theta$ is found. When random keyframes are compared, the estimated probability of collision should be as close to 0 as possible in order to save similarity computations; but a strictly positive value doesn't lower the precision, since the similarity between signatures that collide is always compared to θ. Fig. 5 and Fig. 6 suggest that a partitioning depth $h = 8$ or $h = 9$, the set of rules "0_1_2" and a sentence length $l = 3$ can lead to a good compromise: a relatively high recall (almost all of the corresponding $\mathcal{K}_\theta$ is found) and a very strong reduction in the number of buckets.

3.3 Finding links between keyframes

The role of the buckets is to avoid comparing each signature to every other signature in the database; the full similarity between two signatures is only computed if it is above a threshold. With $L = 20$ and $l = 3$, the similarity between two signatures in a same bucket is at least $\frac{2\times3}{20+20} = 0.15$, as given by (1); while the mining process is significantly accelerated, this value is too low to remove false detections. A higher value could be obtained by increasing the length of the sentences that define the buckets, but this has a negative impact on recall, as seen in the previous subsection. Consequently, within each bucket, the similarity is evaluated for every pair of Glocal signatures and compared to a threshold θ (defining $\mathcal{K}_\theta$) whose value is significantly higher than 0.15 (see Section 4). A link is established between two keyframes if the similarity between their signatures is above θ.

A preliminary analysis has shown that a strong temporal redundancy could be found in many broadcasts (such as talk shows or weather forecasts). If mining was directly applied, many of the links would be established between keyframes that are close in time and belong to a same broadcast; these uninteresting links would then overload the subsequent filtering stages (such as the temporal consistency check). This problem can be solved in many different ways, depending

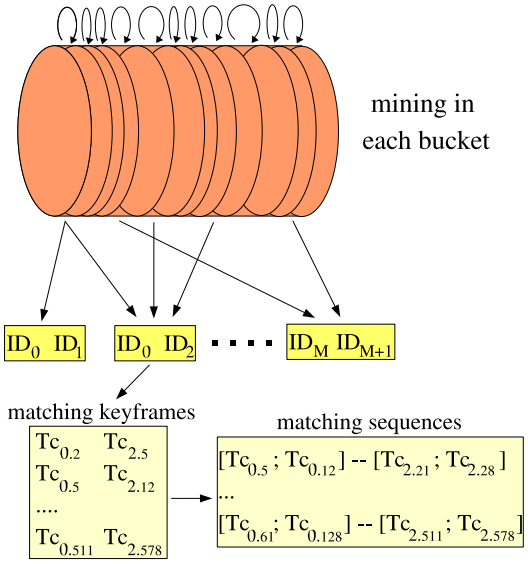

Figure 8: Finding links between video sequences

on the representation of the data. The videos stored in the archives we employed are already segmented into relatively short broadcasts (between a few minutes and three hours), each having its own ID. To every Glocal signature in the database is associated the ID of the video broadcast and the time code Tc of the keyframe it is issued from; time codes are relative to the beginning of the broadcast. The link between keyframe n from broadcast x and keyframe m from broadcast y is specified by the corresponding pairs of IDs $(\mathrm{ID_x}; \mathrm{ID_y})$ and time codes $(\mathrm{Tc_{x,n}}; \mathrm{Tc_{y,m}})$. To avoid the generation of links between keyframes that belong to a same broadcast, the signatures that are in a same bucket and have the same ID are never compared to each other.

3.4 Finding links between video sequences

The connections identified between individual keyframes are used to delimit and link together video sequences that are transformed versions of a same content. Starting from two connected keyframes, two joined sequences are built by the stepwise addition of other connected keyframes (with increasing time codes) that verify temporal consistency conditions. These conditions make the detection more robust to the absence of a few connected keyframes and to the presence of some false positive detections.

The first requirement is that the temporal gap between the last keyframe in a sequence (with time code $\mathrm{Tc_{x,l}}$) and a candidate keyframe to be added to the same sequence (with time code $\mathrm{Tc_{x,c}}$) should be lower than a threshold τ_g:

$$\mathrm{Tc_{x,c}} - \mathrm{Tc_{x,l}} < \tau_g \tag{4}$$

Gaps are due to the absence of a few connected keyframes, as a result of post-processing operations (addition or removal of several frames), of instabilities of the keyframe detector or of false negatives in the detection of connected keyframes.

The second requirement bounds the variation of temporal offset (jitter) between the connected keyframes of two different sequences of $\mathrm{ID_x}$ and $\mathrm{ID_y}$. Jitter is caused by post-processing operations or by instabilities of the keyframe detector. If $\mathrm{Tc_{x,l}}$ is the time code of the last keyframe in $\mathrm{ID_x}$, $\mathrm{Tc_{y,l}}$ the time code of the last keyframe in $\mathrm{ID_y}$ and $\mathrm{Tc_{x,c}}$, $\mathrm{Tc_{y,c}}$ are the time codes of the candidate keyframes, then the condition for upper bounding the jitter by τ_j is:

$$|(\mathrm{Tc_{x,c}} - \mathrm{Tc_{x,l}}) - (\mathrm{Tc_{y,c}} - \mathrm{Tc_{y,l}})| < \tau_j \tag{5}$$

The candidate keyframes are added at the end of the current sequences only if both the gap and the jitter conditions are satisfied. The third condition is that sequences should be longer than a minimal value τ_l to be considered valid; this removes very short detections, typically false positives.

From the indexed database of Glocal signatures representing the keyframes extracted from a collection of videos, the mining process brings out pairs of matching sequences (transformed versions of a same content). This is summarized in Fig. 8. The results can be represented as a graph where every node is a video sequence for which at least a match was found and every edge is such a match.

4. EXPERIMENTAL EVALUATION

To assess the proposed mining approach, precision and recall are measured on ground truth databases, then some of the results obtained on two specific databases are presented, and eventually the scalability is evaluated on larger databases of up to 10,000 hours of video. This mining method can be easily parallelized; nevertheless, all the experiments were performed with a sequential implementation, on a PC having a 3 GHz CPU and 4 Gb of RAM.

Previous experience with CBCD suggested to select a jitter threshold τ_j of 15 frames (0.6 seconds). A gap threshold τ_g of 100 frames (about 4 seconds) allows to avoid an excessive fragmentation of the resulting sequences. A basis value of 4 keyframes (about 4 seconds) for the minimal length τ_l removes short, usually irrelevant detections.

4.1 Evaluation on ground truth databases

The first ground truth database employed consists of 30 hours of original broadcasts issued from the INA archive, together with 20 minutes of copies (short excerpts) obtained from a UGC website for tests. At least one sequence from every broadcast is present, transformed, in the set of copies. Since the amount of data is relatively limited, all the links between original and transformed video sequences were manually checked. The databases being small, Glocal signatures were computed at a partitioning depth $h = 8$.

Several experiments were performed to find an appropriate value for the similarity threshold θ used for establishing links between individual keyframes. Previous experience with the CBCD system in [5] has shown that a keyframe could be safely considered a copy of another if it at least half of the interest points detected did not change completely. This suggested that a good initial guess could be $\theta = 0.5$, which resulted in 46 true positives, 30 false positives and 6 false negatives; recall is 0.88 and precision 0.6. The value obtained for the precision is too low, so for a large video database the volume of false alarms could be too high. With $\theta = 0.55$ there are 43 true positives, 2 false positives and 8 false negatives, which gives a recall of 0.84 and a precision of 0.95. Recall is slightly lower than before but precision significantly increases. The false negatives are either very short, or very noisy, or strongly compressed sequences (with salient MPEG block artifacts). The similarity threshold θ is set to 0.55 for all the other experiments.

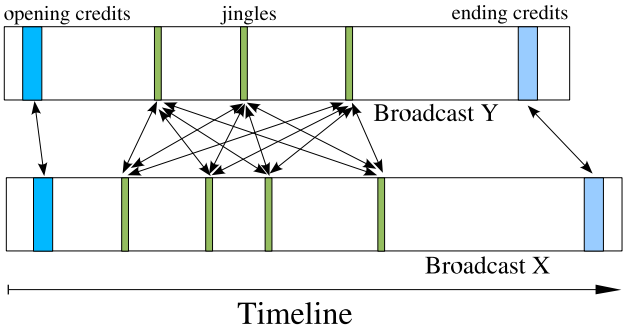

Figure 9: Pattern of broadcast design sequences

This value for the threshold had to be evaluated on another, independent ground truth. We employed the public video copy detection benchmark[1] of CIVR 2007, that provides a database of 80 hours and two sets of queries: ST1 are copies of entire videos from the database, while ST2 consists of copy excerpts inserted into longer videos external to the database. ST1 and ST2 were added to the database and the mining operations performed. We measured the precision and recall in finding the links between the queries in ST1 and ST2 and the 80 hours database. With $\theta = 0.55$, recall is 0.8 for a precision of 0.96. The database of Glocal signatures, the buckets, the links between keyframes and between sequences easily hold in main memory for such small databases, so mining using the indexing scheme only takes 20 seconds. For comparison, if the indexing scheme is not employed, recall and precision remain the same for $\theta = 0.55$, but mining now requires 23 minutes.

To measure the effectiveness of the Glocal descriptor in detecting copies, Table 1 compares our results to those of the CIVR 2007 CBCD competition[1], using the same performance measures. In this case, lowering the θ threshold to 0.45 increases recall without reducing precision, but we nevertheless use $\theta = 0.55$ for the following experiments since we consider the first ground truth to be more representative.

Table 1: Comparison for CBCD

Method	ST1 score	ST2 segment score
Best CIVR 2007	0.86	0.86
Ours with $\theta = 0.55$	0.73	0.71
Ours with $\theta = 0.45$	0.93	0.86

4.2 Mining results on two databases

It can be revealing to visually explore the results obtained on real-world databases in two application contexts.

The first database consists of 1,000 hours of video from the INA archive, to which were added the two ground truth databases. The largest share of the sequences occurring more than once are broadcast design sequences and especially jingles, that are very brief and follow the pattern shown in Fig. 9. To better focus on the other detections, the jingles were set aside; the resulting graph had 6,794 nodes and 12,142 edges. Then the sequences shorter than 7 seconds (mostly opening or closing credits) were also removed, to produce the graph shown in Fig. 10, with 2,757 nodes and 3,057 edges. In this picture, if two sequences are issued from a same broadcast (same ID) and have a non-zero overlap, then they are represented by a single node. So, if a node X is linked to nodes Y and Z, and these two links correspond to different duplicate sequences having a very small overlap ($< \tau_l$), then there is no direct link between Y and Z.

In Fig. 10 the nodes of some subgraphs are illustrated by connected keyframes. Cliques are found when transformed versions of a long enough sequence appear in several broadcasts; they represent here opening or closing credits (> 7 seconds), advertisements or excerpts from news shows. Connected subgraphs that are not cliques correspond to repurposed videos edited in various ways; they support broadcast programming analysis and the transfer of annotations. Graph representations are an effective tool for navigating large sets of results. The interface allows additional filtering based on meta-data and on characteristics of the links.

The second database contains only videos obtained from a UGC website, for tests. It consists in the top 925 videos (for a total of 63 hours) returned by a search with the keyword "Madonna". Since the database is small, all the data holds in main memory so mining required 42 seconds. No broadcast design sequence (jingle, credits) was found.

The resulting graph, having 477 nodes and 1978 edges, is shown in Fig. 11. Three subgraphs are illustrated by connected keyframes. The clique in subgraph A corresponds to transformed versions of a same video sequence; the "tail" contains heavily edited versions of the same clip that include many new fragments. In the "double star" subgraph B, the centers of the stars are two very different versions of a video-clip (same piece of music) that group together video sequences also found in the outer nodes. Even if some outer nodes have links to both centers, no overlap is long enough ($\tau_l = 4$) to establish a link between the centers. The largest connected subgraph C contains sequences from the many different video-clips associated to the "4 minutes" single. The same videos were cut into small segments (some of which are shorter than $\tau_l = 4$) that were then transformed and assembled together. Few of these versions are official releases, the others are probably created by enthusiasts.

The most frequent transformations in this specific database are the reassembly of short sequences, the speedup or slowdown (by as much as 20%), strong compression and scaling. The amplitude of the transformations is higher than for the INA archive. We found no wrong link in the results (precision is 1) but, given the size and the nature of the database, recall could not be measured. Fig. 11 (bottom left) also shows four examples of links that were successfully found between keyframes despite strong transformations.

4.3 Scalability evaluation

To measure the time required for mining and evaluate the scalability of the method, three larger databases of 2,000, 5,000 and 10,000 hours were created by taking miscelaneous videos from the INA archive. The Glocal signatures were obtained by partitioning the description space at depth $h = 9$. Table 2 shows the time required by each stage of the mining process. Database construction includes the computation of the Glocal signatures and the creation of the buckets. The largest share of the time required for mining corresponds to the identification of links between individual keyframes.

For the construction of the database, most the time (95% for $h = 9$) is taken by access to mass storage. The 10,000

[1] `http://www-rocq.inria.fr/imedia/civr-bench/`

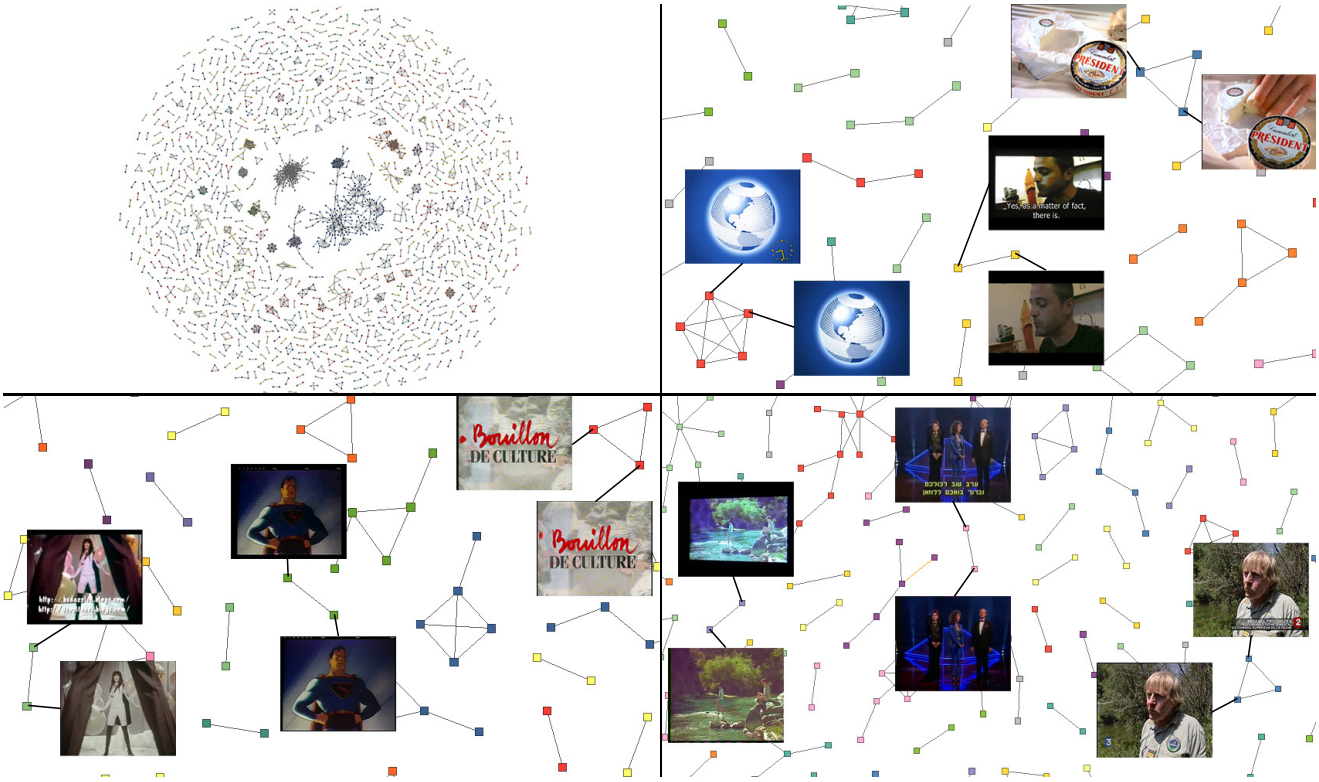

Figure 10: Global view of the graph found on the 1,000 hours database (top left) after removal of most of the components corresponding to broadcast design sequences, and illustrations for several subgraphs

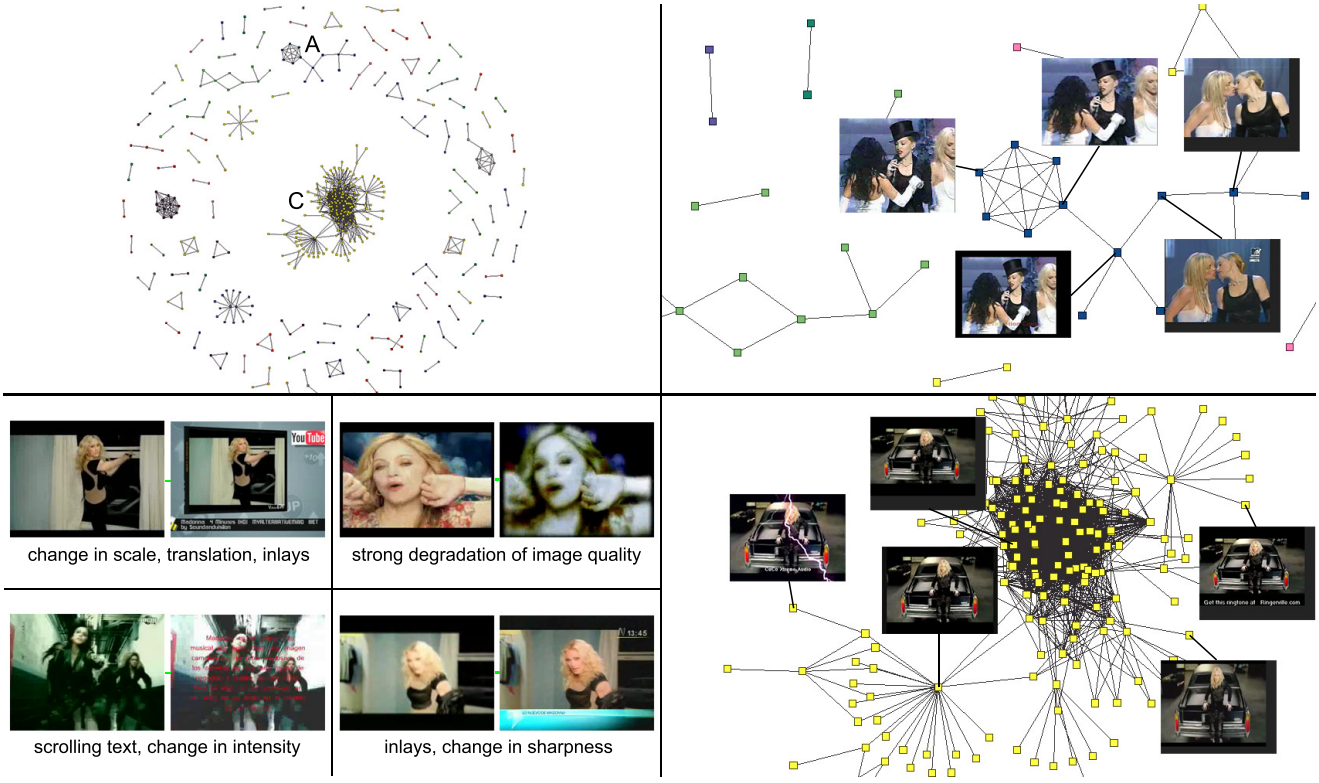

Figure 11: Graph found (top left) on 925 videos obtained from a UGC website (for tests), illustrations for subgraphs A and C, and 4 examples of links found between keyframes despite strong transformations

hours database requires twice the time needed for the construction of the 5,000 hours database because all the work can be done in main memory for the smaller database, while for the 10,000 hours the two halves are processed in main memory independently, one after the other (this also shows how easy it is to parallelize the process).

Table 2: Time required for mining 3 databases

Database size	2,000 h	5,000 h	10,000 h
Nb. of keyframes	5.8×10^6	14.5×10^6	28.7×10^6
Base construction	2h35min	3h38min	7h00min
Linking keyframes	5h40min	14h59min	55h
Linking sequences	1h15min	7h15min	20h35min

For comparison, if the similarity was computed for every pair of Glocal signatures, mining the 10,000 hours database would require about one year. The CBCD solution in [5] can also be applied by using every keyframe as a query; it needs about 20 days to mine 10,000 hours.

5. CONCLUSION

Making explicit the internal structure of a large video database can provide relevant information for content acquisition, management, retrieval and preservation, in both institutional and user-generated multimedia archives. The content links between video sequences, identified by content-based copy detection methods, are an important part of the structure of the database and support a wide range of applications. We focused here on the difficult scalability problem raised by the use of content-based copy detection for mining large video databases. We considered generic video content and a fairly large family of transformations of realistic amplitude between video sequences and their "copies".

The direct application of a state of the art copy detection method to video mining results in an unacceptably long processing time for large databases. This motivated the introduction of a new approach, relying on the definition of compact keyframe-level descriptors and of an adequate index structure supporting mining operations. This approach was shown to provide reliable results on two ground truths and its scalability was demonstrated for databases of up to 10,000 hours of video. It makes video mining by copy detection feasible in a reasonable time for relatively large databases. An extensive study of the characteristics of the extracted sub-graphs and the use of machine learning methods can lead to a reliable automatic classification of the identified links; this is a subject of current work.

6. ACKNOWLEDGMENTS

This work is part of the Sigmund project financed by the French National Research Agency (ANR).

7. REFERENCES

[1] A. Arasu, V. Ganti, and R. Kaushik. Efficient exact set-similarity joins. In *Proc. 32nd intl. conf. on Very Large Data Bases (VLDB'06)*, pp. 918–929. VLDB Endowment, 2006.

[2] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *Proc. 16th intl. conf. on World Wide Web (WWW'07)*, pp. 131–140, New York, NY, USA, 2007. ACM.

[3] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proc. 6th ACM intl. Conf. on Image and Video Retrieval (CIVR'07)*, pp. 549–556, Amsterdam, The Netherlands, 2007. ACM Press.

[4] J. M. Gauch and A. Shivadas. Finding and identifying unknown commercials using repeated video sequence detection. *Computer Vision and Image Understanding*, 103(1):80–88, 2006.

[5] S. Poullot, O. Buisson, and M. Crucianu. Z-grid-based probabilistic retrieval for scaling up content-based copy detection. In *Proc. 6th ACM intl. Conf. on Image and Video Retrieval (CIVR'07)*, pp. 348–355, Amsterdam, The Netherlands, 2007. ACM Press.

[6] T. Quack, V. Ferrari, and L. J. V. Gool. Video mining with frequent itemset configurations. In H. Sundaram, M. R. Naphade, J. R. Smith, and Y. Rui, eds., *CIVR*, *LNCS* vol. 4071, pp. 360–369. Springer, 2006.

[7] S. Sarawagi and A. Kirpal. Efficient set joins on similarity predicates. In *Proc. 2004 ACM SIGMOD intl. conf. on Management of data (SIGMOD'04)*, pp. 743–754, New York, NY, USA, 2004. ACM.

[8] S. Satoh. News video analysis based on identical shot detection. In *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME'02)*, pp. 69–72, 2002.

[9] S. Satoh, M. Takimoto, and J. Adachi. Scene duplicate detection from videos based on trajectories of feature points. In *Proc. intl. workshop on Multimedia Information Retrieval (MIR'07)*, pp. 237–244, New York, NY, USA, 2007. ACM.

[10] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. 9th IEEE Intl. Conf. on Computer Vision (ICCV'03)*, pp. 1470–1477, Washington, DC, USA, 2003. IEEE Computer Society.

[11] M. Takimoto, S. Satoh, and M. Sakauchi. Identification and detection of the same scene based on flash light patterns. In *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME'06)*, pp. 9–12, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[12] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Novelty detection for cross-lingual news stories with visual duplicates and speech transcripts. In *Proc. 15th ACM intl. conf. on Multimedia*, pp. 168–177, New York, NY, USA, 2007. ACM.

[13] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *Proc. 15th ACM intl. conf. on Multimedia*, pp. 218–227, New York, NY, USA, 2007. ACM.

[14] X. Wu, W.-L. Zhao, and C.-W. Ngo. Near-duplicate keyframe retrieval with visual keywords and semantic context. In *Proc. 6th ACM intl. Conf. on Image and Video Retrieval (CIVR'07)*, pp. 162–169, New York, NY, USA, 2007. ACM.

[15] F. Yamagishi, S. Satoh, and M. Sakauchi. A news video browser using identical video segment detection. In K. Aizawa, Y. Nakamura, and S. Satoh, eds., *PCM (2)*, *LNCS* vol. 3332, pp. 205–212. Springer, 2004.

[16] Y. Zhai and M. Shah. Tracking news stories across different sources. In *Proc. 13th ACM intl. conf. on Multimedia*, pp. 2–10, New York, NY, USA, 2005.