Indexing Local Configurations of Features for Scalable Content-Based Video Copy Detection

Sébastien Poullot
National Institute of
Informatics, Tokyo, Japan
and CEDRIC - CNAM, France
poullot.sebastien@free.fr

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

Shin'Ichi Satoh
National Institute of
Informatics
Tokyo 101-8430, Japan
satoh@nii.ac.jp

ABSTRACT

Content-based video copy detection is relevant for structuring large video databases. The use of local features leads to good robustness to most types of photometric or geometric transformations. However, to achieve both good precision and good recall when the transformations are strong, feature *configurations* should be taken into account. This usually leads to complex matching operations that are incompatible with scalable copy detection. We suggest a computationally inexpensive solution for including a minimal amount of configuration information that significantly improves the balance between overall detection quality and scalability.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.2.4 [**Systems**]: Multimedia databases

General Terms

Scalability, Mining, Design, Performance

Keywords

Content-based copy detection, video mining, scalability, local features

1. INTRODUCTION

Many new video programs are produced by recycling existing content, showing how widespread the use of copies is, both among professionals and amateurs. The term “copy” is employed here for any video that is directly or indirectly derived from original content. While initially motivated by copyright enforcement, the detection of these copies in video streams or databases can have many other applications of significant interest for content owners, providers and consumers. The identification, in a large video database, of all the video sequences that occur more than once (with various modifications) can make explicit an important part of the internal structure of the database, thus supporting content management, retrieval and preservation for both institutional archives and video sharing Web sites. Among these applications one could mention content segmentation, the extension of textual annotations from one video to another, the removal of lower-quality copies or advanced visual navigation in the results to a textual query or in the entire database. This shows how relevant content-based video copy detection (CBVCD) can be for structuring a video database.

Among all the copies found in TV broadcasts or on Web2.0 sites, full copies or exact copies are quite infrequent. A full copy is a reproduction of the entire original program, with possible changes of the visual aspect. An exact copy may exploit only an excerpt of the original program, but that excerpt is left unmodified. Most of the time, the visual aspect or the time line of the original content are subject to significant modifications in the copy-creation process (see e.g. Fig. 1). The representation of the videos and the decision procedure should be as robust as possible to the changes resulting from a copy-creation process in order to have a good detection rate. At the same time, they should remain very sensitive to all the other differences between videos in order to have as few false alarms as possible.

Experience with reused video content shows that the most frequent transformations concern gamma and contrast changes, scaling, cropping, blurring, compression artifacts and video inlays (logos, frames, text). The amplitude of gamma or contrast changes and of scaling is relatively low, in order to preserve the perceived quality. Cropping generally remains limited but video inlays can be quite complex and have a higher impact on the video frames. Another type of transformation, currently very popular both for TV shows and Web2.0 sites, is the editing of the time line. It is important to note that, with the advent of high definition (HD) video content, the nature and amplitude of some transformations can be expected to evolve. For example, with HD content, perceived quality can remain acceptable even with strong cropping or scaling.

Content-based video copy detection was already employed for structuring video databases. The method put forward in [18] focused on the elimination of video duplicates in Web search; global descriptors helped separating the least similar videos, then local descriptors allowed to refine duplicate detection. The proposal in [14] of compact embeddings and fast similarity self-joins mainly addressed the scalability issue in mining by CBVCD large video databases. The method was shown to be efficient and effective on a CBVCD benchmark and on Web2.0 content, but has to evolve in or-

LS-MMRM'09, October 23, 2009, Beijing, China.

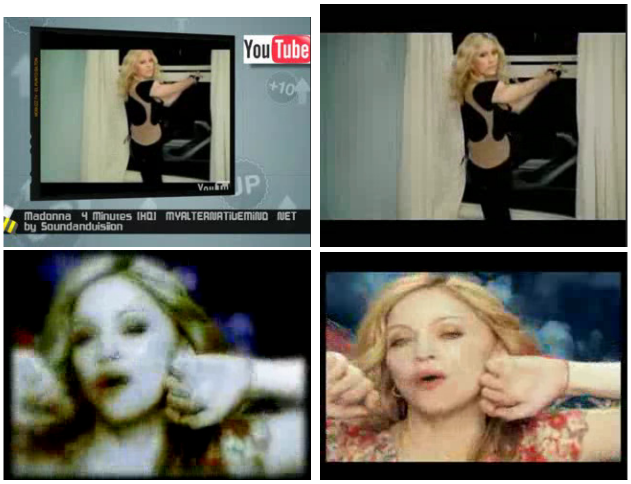

Figure 1: Original content (right) and associated copies (left) obtained by scaling with addition of a frame, logo and text (top) or by strong degradation of image quality (bottom).

der to address new challenges regarding both video transformations and scalability. Our goal here is to significantly improve the robustness of the method in [14] while maintaining or enhancing its scalability.

Most recent CBVCD methods rely on matching frames extracted from the videos and employ local features for describing these frames. To match two sets of local features (issued from two video frames), some proposals do not employ any information regarding the spatial configuration of the features but only count the number of similar features between the two sets. While very robust to geometric transformations, such methods may have difficulties in achieving both good recall and good precision for strong cropping or inlays that affect a large part of a frame. Other proposals rely on more complex matching operations that exploit the spatial configuration of the local features. These methods can more reliably find matches between small parts of frames (being thus more robust to strong cropping or inlays) but may be more sensitive to geometric transformations and have a significant additional cost.

We aim to find computationally inexpensive ways for taking into account the minimal amount of configuration information that allows to obtain both good recall and precision even for strong transformations. We consider that simple configuration information should best be employed during the off-line indexing stage, minimizing *a posteriori* filtering operations. For CBVCD, transformations like strong cropping and video inlays alter the longer-range structure of the frame but maintain part of the short-range structure. We shall then only consider *local* configurations because they are more likely to provide reliable information for matching.

The next section reviews part of the existing work on matching with local features and further discusses the role of configuration information in matching for copy detection. Our computationally inexpensive solution for improving matching with local configuration information is put forward in Section 3. An experimental evaluation of both the detection quality and the scalability of our proposal, with two different types of local features, is presented in Section 4.

2. LOCAL FEATURE CONFIGURATIONS

The most frequent transformations found in video copies concern gamma and contrast changes, scaling, cropping, blurring, compression artifacts and various types of video inlays (logos, frames, text). Frame matching using local features was found to be more robust than the comparison of global frame features in dealing with these transformations (see e.g. [7, 8]). Part of this increased robustness is due to specific invariance properties of local feature detectors and feature descriptions (e.g. gamma and contrast changes, scaling). Also, a significant contribution comes from the matching of local features, which provides robustness to the disappearance of some local features resulting from e.g. cropping and video inlays or from failures of the feature detector.

Simple solutions for computing a matching score between two frames (or images) only count the number of local features that are sufficiently similar in the two frames. If many frames in the database contain similar features but are not versions of a same original content (copies), good precision can only be obtained if the decision threshold (the minimal number of similar features) is relatively high. Note that precision is very important when CBVCD is employed in structuring a video database, since the presence of many false alarms can completely clutter the valuable information in the results. Strong cropping or video inlays are expected to be less infrequent for high definition than for normal definition videos. But with such transformations, only a relatively small share of the local features in the original frame are preserved in the copy. To achieve good recall in such cases, the decision threshold should be low, which in turn produces many false alarms that degrade precision.

To solve this critical problem, additional discriminant information (beyond the local descriptions of the individual local features) is needed. The main source of such information is the *configuration* of local features, that could allow to filter out sets of features having very different geometric organization. This would lead to a significant gain in discrimination power, allowing to diminish the decision threshold and thus to increase recall. Feature configuration information can also improve recall by directly contributing to the matching score of small sets of local features that have very similar configurations.

The typical approach for taking feature configuration information into account comprises two stages. First, the frames in the database that contain similar features to those in the query frame are retrieved. Then, the geometrical consistency with the query is verified for the resulting frames and those who do not reach a high enough score are discarded. A well-known solution for measuring geometrical consistency consists in estimating the parameters of a general affine transformation between the sets of local features using a random sample consensus (RANSAC [3]) algorithm, see e.g. [10, 7, 12, 6, 13]. While reliable, this solution is computationally very expensive, especially when the number of local features in each frame is high and the class of acceptable affine transformations is very general. Despite the use of restricted transformations (selected according to the application domain), of simplified approximate estimation algorithms [10, 13] and of a limited number of local features, the computation cost of this solution can hardly be considered compatible with scalable mining.

Rather than attempting to match sets of features corresponding to entire images, several methods focus on more *local* geometric constraints. One method in [16] (see also [17]) requires, for two local features to match, that 50% of their p nearest neighbors be similar and that angles defined by the relative positions of corresponding neighbors match. For local features whose descriptions include orientation information, it is suggested in [4], for computational complexity reasons, to employ instead the angles between the orientations of neighboring features.

A few recent proposals do not follow the two stage approach consisting of retrieval by similarity followed by filtering based on geometrical consistency, but employ instead a more *integrated* approach. In [9] a hierarchical decomposition of the image plane is employed, adding spatial information to the bags of features. The method in [5] focuses on weak geometrical consistency between the sets of features representing two images, by considering the orientation and scale information available in specific types of local feature descriptions. In [19] visual words are grouped together into "visual phrases" based on the detection of consistent configurations of local features.

It is also interesting to see what configuration information is employed by kernels that were recently developed for matching and also follow an integrated approach. The kernel in [15] augments the local description of each feature with a *local context* including data regarding the presence of other features in its neighborhood, in intervals defined in polar coordinates. In [1] a graph matching kernel is defined and applied to small graphs connecting a point to its nearest neighbors. The configuration information for a set of points is represented by the matrix of values of a local kernel between the positions of individual points in the set.

For exploiting feature configurations in the context of video mining by CBVCD, we consider important to follow an *integrated* approach and to employ *local* geometric information. An integrated approach is expected to improve the selectivity of the similarity self-join operations involved, thus increasing the efficiency of mining. Furthermore, this reduces memory requirements by avoiding the accumulation of potentially large amounts of intermediate data (corresponding to frames having similar sets of points in inappropriate configurations), inherent to the two stage approach.

Since strong transformations that remove or introduce many local features (like strong cropping or video inlays) alter the longer-range structure of the frame but maintain part of the short-range structure, we prefer to employ only *local* configuration information because it has more chances to provide reliable information for matching. The use of local information also contributes to keeping the complexity low, making the integrated approach affordable. Such local configuration information is particularly easy to include into the method put forward in [14] and should not significantly increase its space requirements, since that method was already exploiting n-tuples (triplets actually) of local features for indexing. The next section provides a detailed description of the new scheme.

3. PROPOSED INDEXING SOLUTION

We start by briefly presenting the description and indexing method put forward in [14], that serves as basis for our proposal. The detection of the video sequences that occur more than once (with various modifications) in a video database begins with the extraction of keyframes from all the videos, using an algorithm finding the maxima of the global intensity of motion (leading, on average, to 1 keyframe / second). Then, a similarity self-join operation is performed on the set of keyframe descriptions, based on a specific indexing method. Eventually, these links between individual keyframes allow to find the matching video sequences.

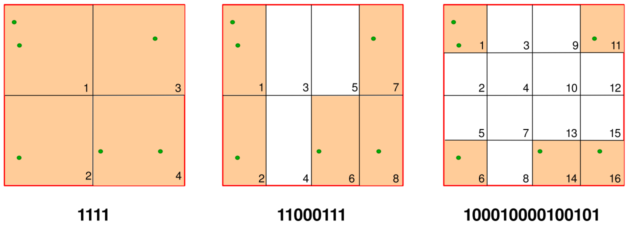

Figure 2: Glocal signatures for a set of 6 local features with 3 different quantizations (at a depth of 2, 3 and 4) of a 2D description space.

Instead of directly using the set of signatures (descriptions) of the local features extracted from a frame, in [14] this set is first embedded into a fixed-length binary vector. The embedding procedure is: (i) given the local features of a set of frames, the description space (not the image plane) is adaptively partitioned at a limited depth h, which produces 2^h cells that are numbered according to some consistent rule (see Fig. 2); (ii) for each frame, its Glocal signature is the binary vector where the bit i is set to 1 only if the description (signature) of at least one local feature of the frame falls within cell i. With the local features employed in [14] (from [6]), the distribution of the features in the description space is rather uniform, so this type of quantization is adequate. Furthermore, as shown in [14], the local features of *each* frame typically belong to different cells, so the loss of information in a Glocal signature is rather limited.

The Dice coefficient was employed to measure similarity between Glocal signatures, $S_{\text{Dice}}(\mathbf{g}_1, \mathbf{g}_2) = \frac{2\,|\mathcal{G}_1 \cap \mathcal{G}_2|}{|\mathcal{G}_1| + |\mathcal{G}_2|}$, where $\mathcal{G}_i$ is the set of bits set to 1 in the signature $\mathbf{g}_i$ and $|\cdot|$ denotes set cardinality.

For the similarity self-join operation, the database of Glocal signatures is divided into overlapping *buckets* (stored as inverted lists) such that, in each bucket, any two signatures are sufficiently similar. A self-join is then independently performed within each bucket. Following [14], a bucket is defined by a specific set of 3 bits that are set to 1 in at least one Glocal signature in the database. Every signature has several bits set to 1 (about 20 in [14]). Each signature can be stored in all the buckets that are defined by all the combinations of 3 bits set to 1 in the signature. This produces a redundant index. However, the number of buckets actually employed is much lower: the buckets into which a Glocal signature is stored are further selected by specific rules that only consider neighboring bits set to 1 (not separated by any other bit set to 1), 1-out-of-2 bits set to 1 (separated by 1 bit set to 1) and 1-out-of-3 bits set to 1 (separated by 2 bits set to 1). This allows to significantly reduce the redundancy of the index. Time and storage complexity depend on the total number of buckets and on the balance between the lengths of the different buckets.

To find the pairs of similar keyframes, the similarities (Dice coefficients) between Glocal signatures are computed within each bucket; if the similarity is above a decision threshold θ, the identifiers of both keyframes are stored as

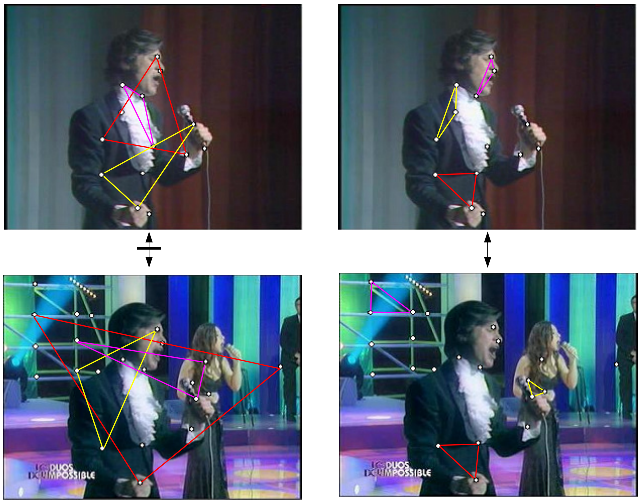

Figure 3: Feature triplets selected in the original frame (top line) and in the copy (bottom line), with the previous rules (left) or with the new locality constraint (right).

a link. At the end of this self-join operation, all the resulting pairs of connected keyframes are eventually used for recovering the matching video sequences.

3.1 Locality-based bucket definition

The rules employed in [14] for selecting the sets of bits set to 1 that define the buckets only depend on the representation of the Glocal signatures (the numbering of the cells in the partitioned description space). Some of the triplets selected by these rules are represented by triangles on the left side of Fig. 3, for an original keyframe (top) and for a copy (bottom) where the video inlay replaced a large part of the frame. It can be seen that the triplets link local features that are quite distant in the image plane and are unlikely to be preserved by strong cropping or video inlays. Since such transformations alter the longer-range structure of the frame but maintain part of the short-range structure, we suggest to take into account *locality* in the image plane when selecting the triplets that define the buckets where the Glocal signature of the frame is indexed.

The new selection and indexing procedure is: for each local feature f_i in the frame, (i) find its 6 nearest neighbors (6NN) in the *image plane*; (ii) the first triplet consists of f_i together with its 2NN and the corresponding bucket is identified by the numbers of the cells in *description space* where f_i and its 2NN are found; (iii) the second triplet consists of f_i together with its 3rd and 4th nearest neighbors, while the third triplet consists of f_i together with its 5th and 6th nearest neighbors; (iv) store (or index) the Glocal signature of the frame into these three buckets. This selection rule thus exploits *both* the positions of the features in description space and their neighborhood in the image plane.

Some of the triplets selected by the new rule are represented by triangles on the right side of Fig. 3, for an original keyframe (top) and for a copy (bottom). The impact of the locality constraint is obvious when comparing with the left side of the same figure. In the example presented in Fig. 3, the previous rules did not allow to find any common triplet between the original and the copy, so their signatures were not indexed in any common bucket and thus the copy remained undetected. The new rule does provide such common triplets (one of which is represented in the figure) and allows to detect the copy.

For the locality constraint to be meaningful and in order to cover well all the small salient areas of a frame, the number of local features considered in the frame should be high enough. A number bounded by 20, as in [14], appears insufficient, especially with HD video content. But an increase in the number of local features considered has an impact on the time and space complexity of the CBVCD-based mining operations. Indeed, the number of buckets necessarily increases with the number of local features per frame. Also, having more features per frame may require a finer partitioning of the description space, which implies longer Glocal signatures that take more space and require more time for computing Dice coefficients. At the same time, the length of the individual buckets is likely to diminish.

To quantify the possible evolution of time and space complexity, the impact of the different variables should be analyzed. Denote by N the total number of signatures in the database, by h the partitioning depth, by L the maximum number of local features per frame and by l the number of bits that define a bucket (l was set to 3 in the above-mentioned rules). The number of bits set to 1 in a Glocal signature is upper bounded by L. The length of every signature is then 2^h, the total number of possible buckets is $\binom{2^h}{l}$ and, if no selection rule is employed, every signature is present in $\binom{L}{l}$ buckets. If all the bits are set to 1 with equal frequency for the signatures in the database, then all the buckets have the same size, equal to $N\binom{L}{l}\binom{2^h}{l}^{-1}$. The number of similarity computations performed with the index is then approximately

$$n = \frac{N^2}{2}\binom{L}{l}^2\binom{2^h}{l}^{-1} \tag{1}$$

The storage requirements are $N\binom{L}{l}$ and maximal when $l = \frac{L}{2}$. Taking for l a value significantly higher than $\frac{L}{2}$ would make the similarity for two signatures in a same bucket too high (close to 1), which would severely restrict recall. The cost of finding the nearest neighbors of a local feature in the image plane also increases with l. For all these reasons, the value considered here is $l = 3$.

If the length of the Glocal signatures is much higher that the maximal number of local features per frame, $2^h \gg L$, then the signatures are sparse. It follows that the number of bits set to 1 in a Glocal signature can be considered independent of the size of the signatures 2^h and only dependent of L, so the space required for storing a signature and the time needed for computing the similarity between two signatures can then be considered fixed for a given L.

To maintain sparsity, h should be augmented when L significantly increases. For higher values of h (more cells are considered in the description space) the expected similarity between a keyframe and a transformed version of this keyframe will diminish since the transformations are more likely to move feature descriptions across borders between cells. So h should be kept under control. But the relevance and reliability of detection of the local features also diminishes when too many such features are employed for representing one frame, so the value of L should not increase too much either.

Table 1: Ratio between computation cost with different values for L, h and cost with $L = 20$, $h = 8$, for fixed $l = 3$

		h			
		8	9	10	11
L	20	1	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{64}$
	50	6.25	$\frac{6.25}{4}$	$\frac{6.25}{16}$	$\frac{6.25}{64}$
	100	25	$\frac{25}{4}$	$\frac{25}{16}$	$\frac{25}{64}$
	200	100	$\frac{100}{4}$	$\frac{100}{16}$	$\frac{100}{64}$

Table 1 shows the evolution of the computation cost with the values of L and h, the reference (used in [14]) being for $L = 20$, $h = 8$. It can be seen that, for comparable sparsity, the variation in computation cost is limited. For example, to keep sparsity close to 10%, when L increases from 20 to 100, h should increase from 8 to 10 and the corresponding increase in cost is $\frac{25}{16}$. Note that this comparison assumes that all the possible buckets are employed and does not consider the impact of the bucket selection rules. But the new locality-based rule was designed to provide a similar selection rate as the rules in [14].

3.2 Use of simple configuration information

Locality constraints reinforce robustness of the indexing scheme to transformations that alter the longer-range structure of the frame while keeping part of the short-range structure. Additional local geometric information should improve discrimination power and thus allow to reach both better detection precision and better recall.

A bucket is identified by using two neighbors (among the 6NN) of a local feature f_i in the frame. It is then natural to associate in that bucket, to the Glocal signature of the frame, data describing the relations between the feature f_i and the two neighbors. The data we add is the ratio between the shortest side and the longest side of the triangle formed in the image plane by the feature f_i and the two neighbors considered. This simple information is robust to translation, rotation and (isotropic) scaling, but not to more general affine transforms like scaling with very different ratios in two different directions (anisotropic scaling). An equivalent choice would have been the angle $\widehat{\text{neighbor}_1\, f_i\, \text{neighbor}_2}$, but the computation of the length ratio is less expensive. Since this information only considers the positions of the local features in the image plane and not their individual descriptions, it can be employed even with local descriptions that do not include any orientation information. Also, it is only dependent on the robustness of the local feature detector and not on the robustness of the feature description.

According to our indexing scheme, the Glocal signature of a frame is stored in every bucket selected by the locality-based rule, together with the ratio between the shortest side and the longest side of the triangle between the local features identifying that bucket. This is shown in Fig. 4. A similarity self-join is then performed in each bucket independently of the other buckets. This operation now involves a joint condition, including both the similarity between the Glocal signatures and the similarity between their corresponding ratio data. The threshold θ_r on the difference between ratios is given by the expected error of the local feature detector and by the required robustness to anisotropic scaling. This ratio information can be stored in low precision.

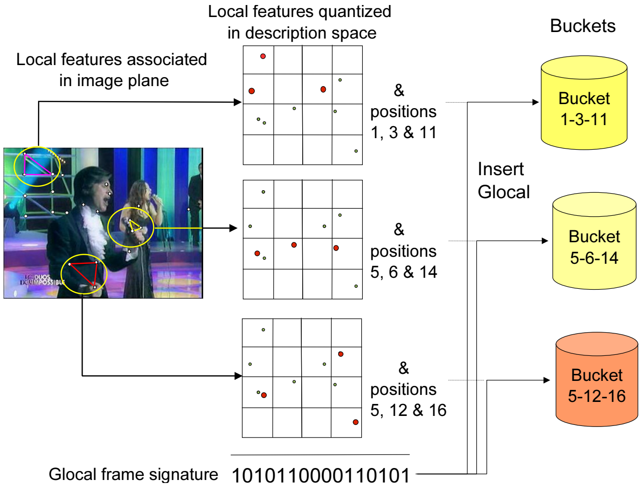

Figure 4: Bucket selection for a frame signature, using both feature location in description space and locality constraints in image plane.

The threshold θ_s on the Dice coefficient above which two Glocal signatures are considered to match is the decision threshold and has a key role in defining the balance between precision and recall. Two keyframes are considered to be in "copy" relation if their Glocal signatures collide in at least one bucket, their Dice coefficient is above θ_s and the difference between ratios in that bucket is lower than θ_r. Actually, the ratios are compared *first* and then, if their difference is $< \theta_r$, the Dice coefficient between the two Glocal signatures is computed. Since the comparison of two small precision numbers is much less expensive than the computation of the Dice coefficient between the two Glocal signatures (especially for long signatures), this pre-filtering using simple local configuration information actually saves significant computation time, at the expense of little additional space.

4. EXPERIMENTAL EVALUATION

4.1 Databases and experimental setup

Since a very large ground truth video database was not available, separate experiments were performed for measuring detection quality (precision and recall) and for assessing the scalability of the method. Quality was evaluated on the Trecvid 2008 CBCD benchmark[1]. This benchmark is composed of a 207 hours video reference database and 2010 artificially generated query video segments (or 43 hours), 1340 of which contain copies. One query can only correspond to one segment of one video from the reference database. A query can contain a copied part of a reference video but also video segments that are not related to the reference database. The queries are divided into 10 groups of 134 queries each, each group corresponding to one type of transformation ("attack"): camcording (group 1), picture in picture type 1 (group 2), insertion of pattern (group 3), strong re-encoding

[1] http://www-nlpir.nist.gov/projects/tv2008/

(group 4), change of gamma (group 5), 3 combined basic transformations (group 6), 5 combined basic transformations (group 7), 3 combined post-production transformations (group 8), 5 combined post-production transformations (group 9), combination of all transformations (group 10). The "basic" transformations are: blur, gamma, frame dropping, contrast, compression, scaling and noise. The post-production transformations are: crop, shift, contrast, caption, flip, insertion of pattern and picture in picture type 2. Since many queries include a left-right flip to which the local features employed here are not robust, the features were detected and described on both the initial queries and a flipped version of the queries (so there are 4020 queries and a total of 293 hours of video).

To evaluate the scalability of the method, a much larger video database of 3,000 hours was employed, obtained by continuously recording five different Japanese TV channels during several days. In this database there is significant redundancy (jingles, advertisements, credits, weather forecasts, news reports, etc.) but most transformations are not so strong and mainly consist of the insertion of logos or text and time line editing.

Two types of local features were employed in the experiments presented below. The first, introduced in [6] and also used in [14] will be called "Harris features" in the following. They rely on the improved Harris detector and on a 20-dimensional spatio-temporal local differential description. Since the Trecvid 2008 CBVCD benchmark includes strong changes in scale, for which these features are not adequate, SIFT [11] features were also employed. The distribution of the SIFT features in the description space being very unbalanced, the random projection method in [2] was employed in order to improve this distribution, resulting in 32-dimensional descriptions.

Three methods were compared: the reference method in [14], our method using locality in image plane for the definition of buckets (subsection 3.1, denoted by L) and the complete method also exploiting simple configuration information (subsection 3.2, denoted by LF). The reference method was only employed with Harris features, while the other two methods (L, SL) were evaluated with both Harris features and SIFT features. For all the methods, triplets were employed for defining the buckets, so $l = 3$. In all cases, the threshold on the difference between ratios is $\theta_r = 0.1$.

Most tests were performed on a laptop PC, having a Q8600 dual core CPU at 2.6 Ghz with 4 Gb of RAM. Even though the method can be efficiently parallelized, the results reported below only used one core. For the 3,000 hours database, a computer having an X7460 single core CPU at 2.66 GHz with 8 Gb of RAM was employed (column marked by * in Table 3 below).

4.2 Detection quality

The reference method denoted by 20_Harris_0.4 was employed with $L = 20$ and $\theta_s = 0.4$, as in [14]. For our new methods (exploiting locality, denoted by L_Harris_θ_s_L, and further using distance ratios, denoted by L_Harris_θ_s_LS), the upper bound on the number of local features per frame was $L \in [20, 200]$, the partitioning depth $h \in \{8, 9, 10\}$ and the decision threshold $\theta_s \in [0.2, 0.5]$. Fig. 5 shows a comparison between 20_Harris_0.4, 150_Harris_0.3_L and 150_Harris_0.3_LF regarding the recall (top) and the number of false positive detections (bottom).

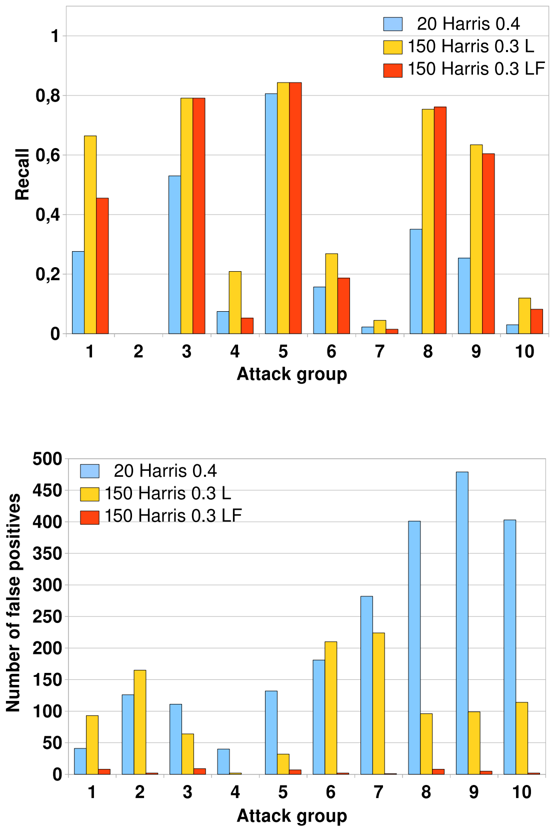

Figure 5: The recall (top) and number of false positive (bottom) detected with 3 methods: 20_Harris_0.4, 150_Harris_0.3_L and 150_Harris_0.3_LF.

It can be seen that recall is improved by using more local features and locality-based bucket selection (L); the addition of local geometric information (distance ratios in LS) does not have a further impact on recall. The most important contribution of the geometric information is the significant reduction of the number of false positives, despite a lower decision threshold θ_s. Fig. 6 illustrates some of the additional detections, showing that copies can be found in spite of the very strong transformations.

A comparison between the two types of features is shown in Fig. 7, for the method using both locality and distance ratios (LS). We set $L = 150$ and, since precision was high, the decision threshold was reduced to $\theta_s = 0.2$. With these parameter values, SIFT features provide on average slightly lower overall recall and equivalent precision.

Consider now the *types* of transformations. Group 2 appears to the most difficult and only 150_SIFT_0.2_LF provides some good detections. For this transformation (picture in picture type 1), the query contains a reference video scaled by a factor in $[0.3, 0.5]$ and displayed in front of a corner or of the center of another, unreferenced video. This is challenging because keyframe detection is driven by the unreferenced video, taking at least 75% of the frame, and also because the improved Harris detector lacks robustness to such changes in scale. Also, group 10 randomly combines all the other transformations, so some queries are also based on group 2.

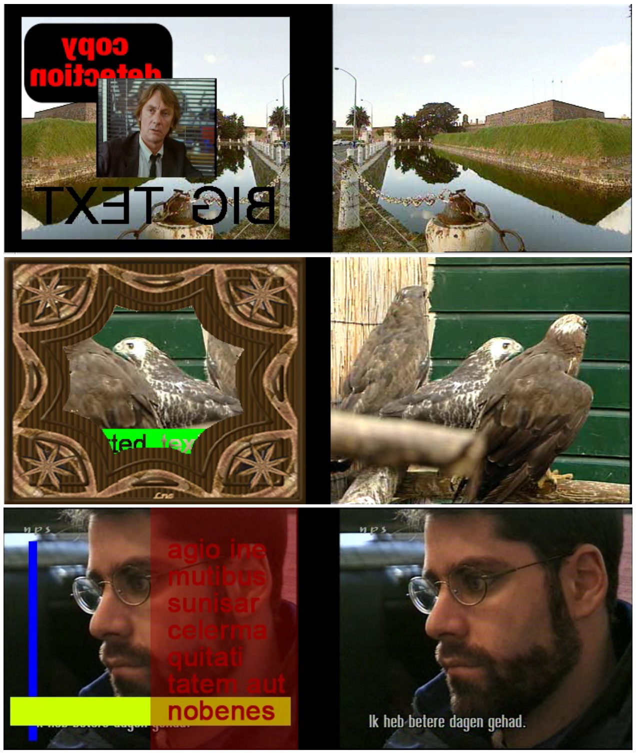

Figure 6: Three detections found with 150_Harris_0.3_LF but not with the reference method (20_Harris_0.4).

Recall is also low for group 4, corresponding to strong re-encoding (lower resolution and bitrate), and groups 6 and 7 that combine several quality degradation transformations. On the other hand, the results are good for camcording (with Harris features), insertion of patterns, change of gamma and combined post-production transformations, even when their amplitude is high (see e.g. Fig. 6).

While the Trevicd 2008 CBVCD dataset is too small for scalability evaluations, the time required for mining it is nevertheless a good indication of how efficient the methods are. Table 2 shows the time needed by 20_Harris_0.4, 150_Harris_0.3_L and respectively 150_Harris_0.3_LF for building the indexed database and then for the self-join operation.

Database construction includes the computation of Glocal signatures and of the buckets, but neither keyframe detection nor local feature extraction. The self-join operation consists in exploiting the buckets for performing the self-join over individual keyframes and then using the results for identifying matching video sequences. Note that the new methods employ here 7.5 more local features than the reference method (20_Harris_0.4). The use of pre-filtering by distance ratios in 150_Harris_0.3_LF make the self-join operation much faster.

4.3 Scalability

The same parameter values were employed for the reference method 20_Harris_0.4. The new methods were tested with both $L = 100$ and $L = 150$. The results of the evaluation performed with Harris features are shown in Table 3. The time needed for building the indexed database increases approximately linearly with L. The significant impact of the pre-filtering by distance ratios appears clearly. For the 3,000 hours database, the total time required (database construction and mining) is almost two times smaller for 150_Harris_0.3_LF than for 20_Harris_0.4, despite the fact that 7.5 more local features are employed by the first. The results obtained with SIFT are very similar.

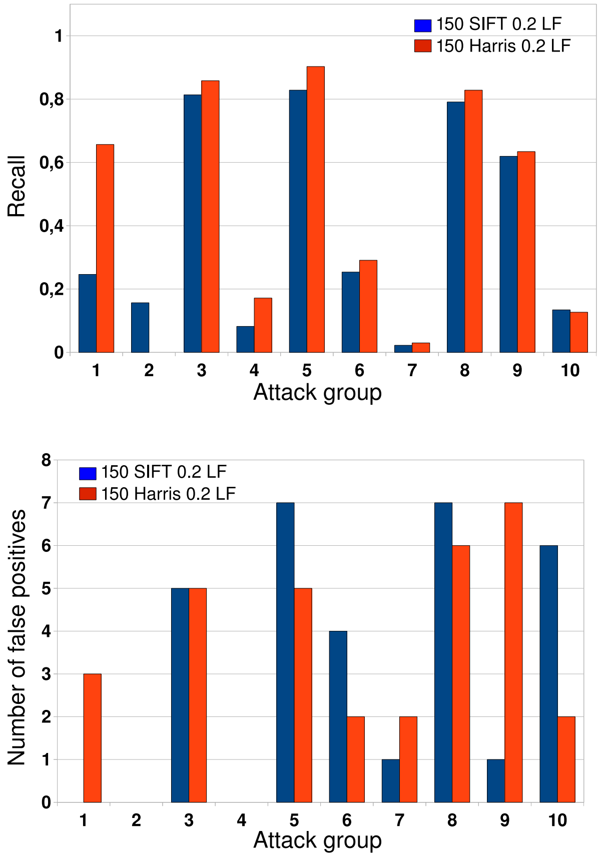

Figure 7: Comparison between 150_Harris_0.2_LF and 150_SIFT_0.2_LF regarding recall (top) and number of false positive detections.

5. CONCLUSION

Content-based video copy detection can provide useful information for structuring video databases, for large institutional archives as well as for video sharing Web sites. The challenge is to be both fast and reliable even when the transformations between original videos and copies are strong. We consider that a better compromise between reliability and scalability requires inexpensive ways of taking into account, in addition to the description of individual local features, information regarding the geometric configuration of these features in the image plane.

Since transformations like strong cropping and video inlays alter the longer-range structure of the frame but maintain part of the short-range structure, we suggest to take into account *locality* in the image plane when indexing the video (key)frames. We further include in the indexing and

Table 2: Time required for building and mining the Trecvid 2008 CBVCD database.

	operation	
method	build	self-join
20_Harris_0.4	4 min 44 s	8 min 02 s
150_Harris_0.3_L	40 min 45 s	13 min 23 s
150_Harris_0.3_LF	40 min 45 s	3 min 04 s

Table 3: Time required for building and mining the larger databases.

		database size	
method	operation	1000 hours	3000 hours*
20_Harris_0.4	build	15 min	21 min
	self-join	2 h 01 min	12 h 00 min
100_Harris_0.3_L	build	1 h 57 min	3 h 04 min
	self-join	7 h 30 min	40 h 00 min
100_Harris_0.3_LF	build	1 h 57 min	3 h 04 min
	self-join	1 h 17 min	0 h 54 min
150_Harris_0.3_L	build	3 h 11 min	5 h 01 min
	self-join	8 h 48 min	41 h 10 min
150_Harris_0.3_LF	build	3 h 11 min	5 h 01 min
	self-join	2 h 21 min	1 h 34 min

matching processes simple local geometric data, involving the nearest neighbors of a feature in the image plane. This data is selected to be as robust as possible to the most common types of image transformations.

An experimental evaluation of the detection quality of our proposal is conducted on the Trecvid 2008 copy-detection benchmark, with two different types of features, and shows a significant improvement over a previous method. The scalability is then assessed on larger databases of up to 3,000 hours of video and highlights the fact that computation time is much reduced by the pre-filtering operation exploiting the local geometric information.

6. REFERENCES

[1] F. R. Bach. Graph kernels between point clouds. In *ICML'08: Proc. 25th Intl. Conf. on Machine Learning*, pages 25–32, New York, NY, USA, 2008. ACM.

[2] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG'04: Proc. 20th annual Symp. on Computational Geometry*, pages 253–262, New York, NY, USA, 2004. ACM.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[4] V. Gouet and N. Boujemaa. Object-based queries using color points of interest. In *CBAIVL'01: Proc. IEEE Workshop on Content-based Access to Image and Video Libraries (CBAIVL'01)*, pages 24–30, Washington, DC, USA, 2001. IEEE Computer Society.

[5] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV'08: Proc. 10th European Conf. on Computer Vision*, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[7] Y. Ke, R. Sukthankar, and L. Huston. An efficient parts-based near-duplicate and sub-image retrieval system. In *Proc. ACM intl. conf. on Multimedia*, pages 869–876, 2004.

[8] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proc. 6th ACM intl. conf. on Image and video retrieval (CIVR'07)*, pages 371–378, New York, NY, USA, 2007. ACM.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: spatial pyramid matching for recognizing natural scene categories. In *CVPR'06: IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[10] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Intl. Conf. on Computer Vision (ICCV'99), Volume 2)*, pages 1150–1157, Washington, DC, USA, 1999. IEEE Computer Society.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Computer Vision*, 65(1-2):43–72, 2005.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR'07: IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[14] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *MM'08: Proc. 16th ACM intl. conf. on Multimedia*, pages 61–70, New York, NY, USA, 2008. ACM.

[15] H. Sahbi, J.-Y. Audibert, and R. Keriven. Incorporating context and geometry in kernel design for support vector machines. Technical Report 2009 D 002, Telecom ParisTech, Paris, France.

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

[17] L. Van Gool, P. Kempenaers, and A. Oosterlinck. Recognition and semi-differential invariants. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 454–460, 1991.

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

[19] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.