Robust Voting Algorithm Based on Labels of Behavior for Video Copy Detection

Julien Law-To, Olivier Buisson
INA
Institut National de l'Audiovisuel
Bry Sur Marne, France
(jlawto,obuisson) @ina.fr

Valerie Gouet-Brunet, Nozha Boujemaa
INRIA Institut National
de la Recherche et de l'Informatique
Rocquencourt, France
(valerie.gouet,nozha.boujemaa)@inria.fr

ABSTRACT

This paper presents an efficient approach for copies detection in a large videos archive consisting of several hundred of hours. The video content indexing method consists of extracting the dynamic behavior on the local description of interest points and further on the estimation of their trajectories along the video sequence. Analyzing the low-level description obtained allows to highlight trends of behaviors and then to assign a label of behavior to each local descriptor. Such an indexing approach has several interesting properties: it provides a rich, compact and generic description, while labels of behavior provide a high-level description of the video content. Here, we focus on video Content Based Copy Detection (CBCD). Copy detection is problematic as similarity search problem but with prominent differences. To be efficient, it requires a dedicated on-line retrieval method based on a specific voting function. This voting function must be robust to signal transformations and discriminating versus high similarities which are not copies. The method we propose in this paper is a dedicated on-line retrieval method based on a combination of the different dynamic contexts computed during the off-line indexing. A spatio-temporal registration based on the relevant combination of detected labels is then applied. This approach is evaluated using a huge video database of 300 hours with different video tests. The method is compared to a state-of-the art technique in the same conditions. We illustrate that taking labels into account in the specific voting process reduces false alarms significantly and drastically improves the precision.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithms

Keywords

Content-Based Video Copy Detection, Label of Behavior

MM'06, October 23–27, 2006, Santa Barbara, California, USA.

1. INTRODUCTION

Due to the increasing broadcasting of multimedia contents, finding similar videos or exact video copies is a new issue. The professionals of archives need to trace the use of their large video databases and for this, Content Based Copy Detection (CBCD) is an alternative to the watermarking approach for identification of video sequences.

Two similar videos which are not copies (the ties are different)

Two videos which are copies (one is used to make the other)
Source video: *Gala du Midem.* G. Ulmer 1970 (c) Ina

Figure 1: Copy / similarity.

In this paper, we focus on CBCD on large collections of videos which involves a content-based comparison between the original object and the candidate one [6, 10]. It generally consists of extracting few small pertinent features (called signatures or fingerprints) from the image or the video stream and matching them with the database. Several kinds of techniques have been proposed in the literature for the video retrieval: [9] uses a temporal fingerprints based on the cuts in a video sequence whereas [6] compares global descriptions of the video (motion, color and spatio temporal distribution of intensities). For still image retrieval, [3] defines fingerprints based on the wavelets to find replicate images on the web whereas [11] uses local descriptions on *points of interest*. Initially proposed for stereovision purposes, points of interest are sites in an image where the signal takes high frequency in several directions. Using such primitives is mainly motivated by the observation that they provide a compact representation of the image content

since limiting the correlation and redundancy between the detected features. When considering image transformations like geometric changes (cropping or shifting), signatures based on *points of interest* have been proven to be efficient for retrieving still images [2] and video sequences [10]. We will revisit them in section 2.

For a CBCD application, a crucial difficulty is the difference between a copy and a similarity: a copy is not an identical or a near replicated video sequence but rather a transformed video sequence. These transformations can strongly change the signal (gamma and contrast transformations, overlay, shift etc...) therefore a copy can be visually less similar than other kinds of similar videos. Some applications need to find similar videos like soccer games, or episode of a soap shows for video indexing but those detections are clearly false alarms in a CBCD application. The figure 1 shows very similar video but not copies and copies which are less similar.

We propose a concept that involves the estimation and characterization of trajectories of points of interest along the video sequence. Building trajectories of points in videos is a recent topic for video content indexing. At present, such trajectories are usually analyzed for modeling the variability of points along the video and then enhancing their robustness, for generic object recognition (see for example [5, 20]). We plan on taking advantage of such trajectories for indexing the spatio-temporal contents of videos. First, the redundancy of the local description along the trajectory can be efficiently summarized with a reduced loss of information and second, the trajectory properties will allow to enrich the local description with a spatial, dynamic and temporal behavior of this point. Analyzing the obtained trajectories allows to highlight trends of behaviors and then to assign a label of behavior to a local descriptor. The aim is to provide a *rich*, *compact* and *generic* video content description which can be used in a robust voting function for copy detection in large video databases. This voting function is based on a smart use of the signal description, the contextual information and the combination of relevant labels. Adding context to a local descriptor was recently proposed for still images. In [16], the authors use spatial context to enhance the matching of points of interest and in [1], the authors use spatial relation between the points of interest for increasing the quality of an object recognition algorithm. Similarly, the concept proposed by J. Sivic and A. Zisserman in [19] also involves points of interest in video sequences but the concept is different from our works because it is based on similarity more than on finding copies.

The following will be discussed in this paper: in section 2, we present our method to obtain the low-level description of the video sequences, i.e. to extract, to characterize points of interest and to estimate their trajectories along the sequences. Section 3 defines the concept of the final signal description, the temporal context and the labels based on the obtained low-level description. In section 4, the robust algorithm of retrieval based on the smart use of labels during the on-line retrieval step and on a spatio-temporal registration is presented. Section 5 is dedicated to the evaluation framework while section 6 presents the evaluation of our algorithm facing a state-of-the art technique ([10]) with different complementary tests in order to highlight the high performances of our system for CBCD.

2. BUILDING TRAJECTORIES OF LOCAL DESCRIPTORS

We present here the low-level description of the video sequences. Section 2.1 details the choice we made for interest point extraction and local characterization, while section 2.2 describes the algorithm for tracking these points. Though these techniques are classical, they do not represent the major contribution of this work.

2.1 Extracting and characterizing points of interest

The interesting properties of points of interest make them popular in the literature of Computer Vision and CBIR. The well-known Harris and Stephens detector [7] used to be described with local features, applied to gray value or color images. Many works have been done to make them robust to several image transformations. A recent performance evaluation [15] has shown that the SIFT descriptor [14] performs best for object recognition. More recently, points of interest have been extended to spatio-temporal signal [12]. Points of interest are relevant for precise retrieval in images, like objects or details. Associated to an adequate voting function, they are robust to occlusion and consequently are interesting for copy detection purposes where several geometric transformations of the image can occur, like cropping or shifting. Section 4 will present a voting function dedicated to CBCD.

We have not used the SIFT descriptor, first because it involves a high dimensional features set (128 items for each key point), making it incompatible with several hundred hours of videos (one hour represents $25 * 3600$ pictures, involving roughly 3×10^6 local descriptors). Second, this descriptor is invariant to several image transformations, making it efficient for object recognition but not optimal for tracking where consecutive frames differ by small transformations. We did not use a spatio-temporal local descriptor like the one described in [12] because the temporal part would really be not relevant for the tracking step describes below.

Therefore, the descriptor we employed is the Harris detector associated to a local description of the points leading to the following 20 dimensional signatures $\vec{S}$:

$$\vec{S} = \left(\frac{\vec{s_1}}{||\vec{s_1}||}, \frac{\vec{s_2}}{||\vec{s_2}||}, \frac{\vec{s_3}}{||\vec{s_3}||}, \frac{\vec{s_4}}{||\vec{s_4}||} \right)$$

where the $\vec{s_i}$ are 5 dimensional sub-signatures computed at 4 different spatial positions around the interest point. Each $\vec{s_i}$ is a differential decomposition of the gray level signal $I(x, y)$:

$$\vec{s_i} = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2} \right)$$

We use gaussian filters for computing the derivatives in order to reduce the noise. Such a description is invariant to image translation and to affine illumination transformations. In the remainder of the paper, this features space will be called the space S_{Harris}.

2.2 Tracking points of interest

Temporal approaches of feature point tracking exists for *point trajectory estimation*. Classically, the encountered techniques involve a cost function defined for three consecutive frames. Different linking strategies are applied to find the correspondences and optimize the trajectories. The most popular approach is probably the Kanade-Lucas-Tomasi (KLT) tracker, proposed in 1981 and fully developed later in [21]. It consists in defining good features by examining the minimum eigenvalue of each 2 by 2 gradient matrix, and in tracking them using a Newton-Raphson method of minimizing the difference between the two windows. Another approach is the one developed by Sethi and Jain in [18] called Greedy Exchange algorithm (GE). This algorithm is based on a cost function which penalizes the changes of direction and the magnitude of the speed vector. In [4], the algorithm "IPAN tracker" described is based on the idea of competing trajectories. The previous paper

also presents a performance evaluation of feature point tracking approaches. More recently, probabilistic and multi-solution tracking methods like particle filters in [8], inspired by the Kalman filter, has been developed to track the non-rigid objects and multiple objects or multiple points.

As we focus on low-cost computational techniques, the tracking algorithm we have chosen is basic and does not depend on the local description adopted. A L_2 distance is computed in S_{Harris} from frame to frame between all the local descriptors of the frame and all of those from 15 previous frames and the 15 next frames and for points that match, three decisions can be taken:

- matching only in the future: start of a new trajectory;
- matching only in the past: end of a trajectory;
- matching in the future and in the past: add the point to an existing trajectory.

3. LABELING BEHAVIOR OF POINTS

In this section, the choices made for building a higher level description of the set of videos, based on the low-level descriptors are presented. The different features spaces used for the off-line indexing are defined in this section.

3.1 Signal description

At the end of the trajectory building, a *low-level* description of the points of interest based on the signal must be associated to the trajectory. For each trajectory, we take the average of each component of the local descriptors in S_{Harris} as a low-level description of the trajectory. The descriptor obtained will be noted $\vec{S}_{mean}$. As the trajectory is computed from frame to frame, the local signatures may vary along the trajectory. To assess the representativeness of $\vec{S}_{mean}$ in the trajectory, we test on a sequence which lasts 1 hour, how many local signatures of the trajectory has a distance from $\vec{S}_{mean}$ lower than the matching threshold used during the trajectory building. 95 % of the points of the trajectories have a lower distance than this threshold. This evaluation confirmed that $\vec{S}_{mean}$ is relevant for characterizing a trajectory. A similar approach is described in [5]: the authors show that, on a trajectory, the SIFT descriptor has a quadratic variation depending on the viewing angle, and they take the average of the descriptors in the minimum zone of this variation. In the rest of the paper, the obtained feature space will be called the signal description space S_{Signal}.

3.2 Trajectory description

A higher level description of the local descriptors presented above can be obtained by exhibiting the geometric and kinematic behavior of the interest points along his trajectory. To do this, the following trajectory parameters are stored during the off-line indexing step:

- Time code of the beginning and of the end: $[tc_{in}, tc_{out}]$;
- Variation of the spatial position: $[x^{min}, x^{max}], [y^{min}, y^{max}]$.

In addition to these parameters, the mean local descriptors $\vec{S}_{mean}$ of S_{Signal}, associated to the trajectories, provides a richer description of the video content. Such a description is *generic*, because it is independent of the applications and therefore, it is computed *only once*, no matter what application is considered. In the remainder of the paper, this feature space will be called the trajectory parameters space S_{Traj}.

3.3 Definition of labels

From the description defined above, it is possible to exhibit trends of behaviors. For example, these categories can be considered:

- Moving / motionless points;
- Persistent / rare points;
- Fast motion / low motion points;
- Horizontal motion / vertical motion.

This list is just one example of categories of trajectories. By classifying the local descriptors according to their behavior, a label of behavior can be assigned to them. In the current version of this work, the categories of behaviors are simply obtained by thresholding the parameters defined in section 3.2.

This is a *higher level* description, because involving an interpretation of the video, and at the same time is a *specific* description of the video content. In the experiments perfomed for this paper, we chose two particular labels: the motionless and persistent points are used to define the label *Background* while the moving and persistent points define the label *Motion*. Those two labels are relevant for a CBCD system in a huge database as it was shown in a previous work but with a more simple voting function (see [13]). Their joint use during the voting process will be detailed in section 4.4. This section has exposed the extracted features from the off-line indexing and the next section presents their use in a robust voting function.

4. EFFICIENT RETRIEVAL ALGORITHM

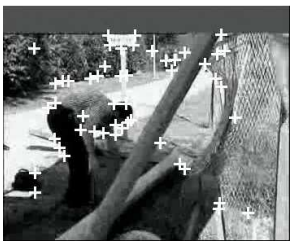
This section presents the algorithm of retrieval in details. This method uses the signal description (feature space S_{Signal}) for selecting candidates. Then it uses the trajectory information (feature space S_{Traj}) and the labels computed during the off-line indexing part for taking a decision. This voting function is robust to outliers and is the key of the whole retrieval system. This spatio-temporal robust registration is the main contribution of this paper.

4.1 An Asymmetric Technique

As the off-line indexing part needs long time computational (see section 6.3) and as the system of retrieval needs to be in real-time, the whole indexing process described in sections 2 and 3 cannot be done for the candidate video sequences. The retrieval approach is so asymmetric. The queries are descriptors from the feature space S_{Harris} computed depending on two parameters:

- period p of chosen frame in the video stream;
- number n of chosen points per selected frame.

For now, we use p and n constant but we can imagine different strategies for computing them. In the experiments shown here, we test two sets of values for those parameters. The advantage of the asymmetric technique is that the number of queries and the temporal precision can be chosen on-line, which gives more flexibility to the system. The main challenge of the asymmetric method was that we had on one side points of interest with a description from the feature space S_{Harris} and on the other side descriptors from the feature spaces S_{Signal} and S_{Traj}. Figure 2 illustrates the registration challenge with the crosses on the query points on the left and the representation of the S_{Traj} space on the right.



Candidate video
The + represent the queries
(Points of interest)

Original video in the database
The boxes represent S_{Traj}
$[x^{min}, x^{max}], [y^{min}, y^{max}]$

Figure 2: Illustration of the feature spaces involved in the asymmetric method.

4.2 Probabilistic Similarity Search

The candidate video clip is viewed as a series of K_S selected frames characterized by the time codes tc_l ($l \in [1, K_S]$) with p frames between each selected frame. On each selected frame, a number n of points of interest are extracted, described by a local signal description in the feature space S_{Harris} and characterized by a spatial position: $(x_{l,m}, y_{l,m})$ ($m \in [1, n]$). By searching the descriptors of the candidate video sequence from the feature space S_{Harris} in the feature space S_{Signal} using a probabilistic similarity search algorithm similar to the one detailed in [10], we measure the similarity between the candidate video and the video sequences in the database. Instead of using classical similarity queries (range queries, Knn queries), this technique applies probabilistic filtering rules on the feature space which allows fast search in huge databases with a reduced loss. This approach allows to choose some potential matches in the video database but this selection is not discriminant enough so a voting step based on a registration using a geometric model is necessary. The use of registration allows improvement which has been proved in [9] with temporal registration and in [11] with spatial registration. This registration is done using the features in the feature space S_{Traj} with the candidates found by the first step. For each query $(x_{l,m}, y_{l,m})$, the search processing in the reference descriptor space S_{Signal} returns a number $R_{l,m}$ of results. Each result $r_{l,m,r}$ ($r \in [1, R_{l,m}]$) has a value in S_{Traj} :

$$([tc^{in}_{l,m,r}, tc^{out}_{l,m,r}], [x^{min}_{l,m,r}, x^{max}_{l,m,r}], [y^{min}_{l,m,r}, y^{max}_{l,m,r}])$$

The registration is done by using those selected values associated to their labels and their descriptor in S_{Traj}.

4.3 Spatio Temporal Registration

If a candidate video S is made from the same source as a reference R in the database, there is a constant spatio-temporal offset between R and S. Therefore, during the decision algorithm, the goal is to estimate this offset. In a first step, a spatial registration is made on each frame. The idea is to count the number of queries which are compatible with a given offset. The offset or difference $d_{l,m,r}$ between the query $(x_{l,m}, y_{l,m})$ and the results $r_{l,m,r}$ is an interval-valued data:

$$d_{l,m,r} = ([tc^{in}_{l,m,r} - tc_l, tc^{out}_{l,m,r} - tc_l], \\ [x^{min}_{l,m,r} - x_{l,m}, x^{max}_{l,m,r} - x_{l,m}], [y^{min}_{l,m,r} - y_{l,m}, y^{max}_{l,m,r} - y_{l,m}])$$

During the off-line indexing part, each video sequence has been associated to unique number Id ($Id \in [1, N_{videos}]$) with N_{videos} the total number of video sequence in the video database. At each time code, the potential matches in the database are grouped by Id. For each Id, we evaluate the best offset. As the algorithm is based on interval-valued data, the possible offset O_f are also interval-valued data:

$$C_{Id,l,m}(O_f) = \begin{cases} 0 \; if \; \forall r, \; d_{l,m,r} \cap O_f = \{\emptyset\} \\ 1 \; else \end{cases}$$

This $C_{Id,l,m}$ compatibility measure is applied for each query at the time code tc_l and the optimal offset maximizes the number of compatible queries:

$$Cr_{Id,l}(O_f) = \sum_{m=1}^{n} C_{Id,l,m}(O_f) \quad (1)$$

The Cr criterion is not based on a simple intersection in order to be robust to outliers. To optimize Cr, the algorithm tests the different offset $d_{l,m,r}$ and their intersection to find the optimal offset $O_f(Id, l)$.

At the end of this step, we have an optimal score $Cr_{Id,l}$ associated to an interval-valued offset $O_f(Id, l)$ for each time code tc_l and each possible Id. The next section presents the use of the labels and their combination.

4.4 Combination of labels

The idea is to apply the previous registration strategy with different types of labels to improve CBCD but not in the same way. Similarly, A. Opelt in [17] uses different kinds of descriptors together to improve object recognition in still images and J. Sivic and A. Zisserman in [19] use two types of viewpoint covariant region to describe videos. In our case, the first step (see equation 1) is computed separately for the different labels because the information could be not relevant in the same way. For example, the label background presents a very accurate spatial position and a large temporal imprecision because of the persistence of the points whereas the label motion is more accurate in the temporal domain. By this step, we have for each frame a number of possible Id with a score for each label $Cr_{Id,l,labelX}$ considered and for each label, we have an interval-valued offset $O_f(Id, l, labelX)$. In order to combine the labels we need to make a fusion of the score for the same time code and Id and of the interval-valued offset. We use an heuristic for now to make this fusion by just multiplying the scores in case of compatibility as shown in the algorithm 1. The labels used are discussed after the algorithm.

Algorithm 1 Combination of labels.

if $(O_f(Id, l, \textit{Motion}) \cap O_f(Id, l, \textit{Background}) = \{\emptyset\})$ **then**
  $Cr_{Id,l} = Max(Cr_{Id,l,\textit{Motion}}, Cr_{Id,l,\textit{Background}})$
  $O_f(Id, l) =$ corresponding O_f
else
  $Cr_{Id,l} = Cr_{Id,l,\textit{Motion}} * Cr_{Id,l,\textit{Background}}$
  $O_f(Id, l) = O_f(Id, l, \textit{Motion}) \cap O_f(Id, l, \textit{Background})$
end if

For CBCD, we use the labels *Motion* and *Background*. Those two labels has been chosen because it seems natural that they are useful for a copy detection. The background is very robust and typical of a show and the motion is very discriminant. These labels have been tested and have proved their relevance in a previous work (see [13]) with a previous version of the vote. They still present good performances as it is shown in the evaluation (see section 6). Using the labels allows one to eliminate false alarms and at the same time, the feature space is smaller while the performances are better.

At the end of this step, for each frame, there is a number of potential Id that may correspond. Each Id has a score, one or two

labels and an interval-valued data which corresponds to an approximative spatio-temporal offset between the candidate clip and the potential matching video sequence from the database.

4.5 Propagation of the detected segments

This last step consists of aggregating in time the results of the previous step. So for each time code and for each Id, an intersection is computed from selected frame to selected frame and the score is cumulated in order to find the limits of the detected video segment (the first time code T_{in} and the last T_{out}). $Cr_{Id,T_{in},T_{out}}$ is the final score and $O_f(Id, T_{in}, T_{out})$ is the final estimated offset.

Algorithm 2 Propagation of the detected segments.

$l = T_{out} + p$
if $O_f(Id, T_{in}, T_{out}) \cap O_f(Id, l) = \{\emptyset\}$ **then**
  Detection is over
else
  $T_{out} = l$
  $Cr_{Id,T_{in},T_{out}} = Cr_{Id,T_{in},T_{out}} + Cr_{Id,l}$
  $O_f(Id, T_{in}, T_{out}) = O_f(Id, T_{in}, T_{out}) \cap O_f(Id, l)$
end if
$l = T_{in} - p$
if $O_f(Id, T_{in}, T_{out}) \cap O_f(Id, l) = \{\emptyset\}$ **then**
  Detection is over
else
  $T_{in} = l$
  $Cr_{Id,T_{in},T_{out}} = Cr_{Id,T_{in},T_{out}} + Cr_{Id,l}$
  $O_f(Id, T_{in}, T_{out}) = O_f(Id, T_{in}, T_{out}) \cap O_f(Id, l)$
end if

Labels are also used here to define a starting frame for this aggregation. In order to start from a very confident frame, the beginning of the aggregation must present all labels. In the case of the copy detection, we cannot start from a frame which just has the same part of the background, the frame must present motion matching. This is a way of avoiding the matching of the TV shows with the same background for example. For each Id, the final results show a number of possible spatio temporal offset evaluated at the precision of the final interval-valued data with a beginning time code, an end time code and a global score.

In this section, we have presented an efficient voting algorithm based on the high-level descriptors described in section 3. These choices allow to build a CBCD system robust to several transformations: by using the feature space S_{Signal}, we obtain a signal description robust to the noise due to the average of the features from S_{Harris} along the trajectory. This description is also invariant to affine illumination changes. The spatio-temporal registration using the feature space S_{Traj} allows an invariance to spatial and temporal shift of the video. The actual version of the retrieval system is not robust to zoom and to slow motion. Figure 3 presents the whole video copy detection Framework. In the two following sections, we will evaluate our approach on several hundreds hours of videos. The next section presents the framework of the evalutation while section 6 is dedicated to the results of the evaluation on different video benchmarks.

5. FRAMEWORK FOR THE EVALUATION

Evaluating a system of video copy detection is not obvious and this section presents our strategy for comparing our methods to others. Defining what a good retrieval is poses a problem. A perfect copy detection system should find all the copies in a video stream even with strong tranformations with a high precision. This section

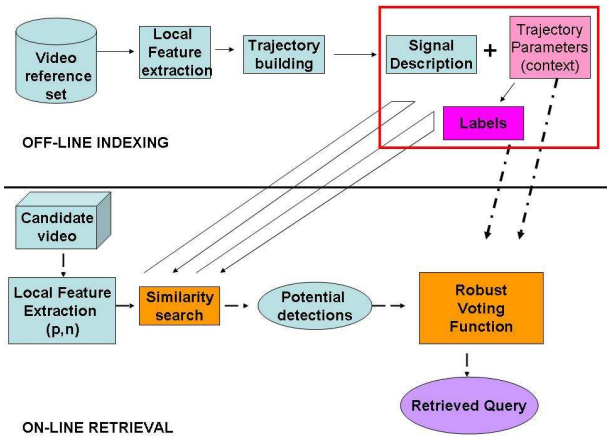


Figure 3: Video Copy Detection Framework.

presents the evaluation framework. The results of this evaluation are commented on in section 6.

5.1 Video Database

All the experiments are done on 300 hours of videos randomly taken from the video archive database stored at INA (the french *Institut National de l'Audiovisuel*). These videos are TV sequences from several kinds of programs (sports event, news show, talk show) and are stored in *MPEG-1* format with 25 frames per second and an image size of 352 x 288 pixels. To test the robustness of the system, we define different types of transformations and use those with different parameters to simulate the potential processing on the video sequences like crop, zoom, resize and shift. Noise and transformation of the contrast and the gamma can also occur. Example of transformations are shown in figure 5 on the left column. We have to notify that the value of the zoom was small (0.95 to 1.05) because our system is for now not robust to big zoom.

For the evaluation of the retrieval system, we have built two types of curves presented in detail in 5.3: the Receiver Operating Characteristic curves (ROC) and the precision-recall curves.

For the ROC curves, we use a video sequence from the database randomly transformed called *BenchPositiveAttack* and a video not in the database called *BenchNegative*.The video *BenchPositiveAttack* lasts 24 minutes (36000 frames) while *BenchNegative* lasts 3 hours (270000 frames).

For the precision-recall curves, we have built two video benchmarks: *Bench1min* and *Bench30*. To build *Bench1min* and *Bench30*, we have selected randomly 40 samples of video segments from the video database and we have transformed them randomly: for each segment, the transformations use different parameters. In *Bench1min*, each video sequence length is 1 minute and in *Bench30*, each video sequence has a random length from 10 frames to 30 seconds. Those videos have been inserted in 7 hours of videos not in the 300 hundred hours database as it is shown in the figure 4.

As the video sequences were taken randomly in a huge database of archives, they are very different: different in the quality of the encoding (some old videos present bad quality of pictures) and different in the kind of TV shows: sports events like soccer games, very old archives, recent commercials, recent news shows ... The robustness to re-encoding is also tested, because for computing the benchmark videos, the video segment is re-encoded twice with different encoding systems.

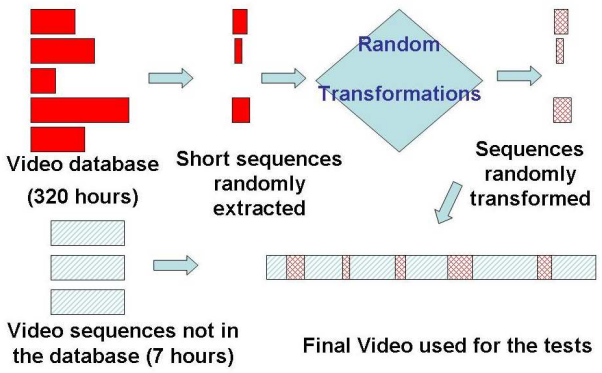


Figure 4: Video Benchmark Building.

5.2 A Reference Technique to Compare

As a reference, we use a symmetrical technique with local description: the same algorithm is applied to the database and the queries. This technique uses key frames based on the image activity and local descriptors based on the signal on points of interest. As shown in [10], this method presents high performance even on large video databases. We have implemented this technique as a reference rather than [6] which uses different global descriptions (color, motion and distribution of intensities) because they are not enough robust for our specific needs leading to lower performances, especially for short video sequences. A. Joly and al. [10] kindly provided us their code in order to compare the different techniques using exactly the same parameters. This technique uses a similar kind of local description based on points of interest. The main difference is that we have defined a temporal context for the local description and we use this context in the voting part. Another difference is that our technique describes the whole video sequence during the off-line indexing and not only the key images. In the asymmetric process, we can choose the temporal precision of the detection on-line by changing the parameters of the queries. This choice is impossible for the reference technique which needs to index the videos in the same way that it computes the queries.

5.3 Evaluation criteria

To evaluate our system, we build two kinds of curves: the ROC curves and the Precision Recall (PR) curves. In order to build the ROC curves wich present the true positive rate versus the false positive rate, *BenchPositiveAttack* ($N_{TotalFramesTrue}$ frames) and *BenchNegative* ($N_{TotalFramesFalse}$ frames) were used. By combining the results of the vote for these two videos, we can build the curves. In this case the true positive and false positive rates are defined as:

$$TruePositiveFramesRate = \frac{N_{TruePositiveFrames}}{N_{TotalFramesTrue}}$$

$$FalsePositiveFramesRate = \frac{N_{FalsePositiveFrames}}{N_{TotalFramesFalse}}$$

Another kind of evaluation is the PR curves. Those curves are more data dependent than the ROC curves. The precision strongly depends on the rate between the number of potential retrieved videos and videos not in the database. In our case this rate is very low: we use the two videos *Bench1min* and *Bench30*. Precision and recall are usually obtained from the following formulas:

$$Recall = \frac{N_{TruePositive}}{N_{AllTrue}}$$

$$Precision = \frac{N_{TruePositive}}{N_{AllPositive}}$$

When considering video segment retrieval, there are two possibilities to obtain them. $N_{TruePositive}$ can be the number of segments or the number of frames with a score higher than a threshold. It is important to notice the difference between detecting a segment and detecting the frames. Detecting the segment is fundamental for finding all the video copies in a video stream but detecting the frames is also important because the final human operator needs to have the best possible precision on the detection for controlling the results. Finding the frames is also more difficult and the results will be less apparent but more accurate for an objective evaluation of the precision.

6. EVALUATION FOR CBCD

The objective of this section is to demonstrate the relevance of the approach for CBCD proposed in sections 3 and 4. We first illustrate the retrieval results on the figure 5 based on the framework presented in section 5. Then we give a qualitative and quantitative evaluation of the retrieval system by computing ROC and PR curves. We also give some computational costs values and in a final part the relevance of the method on a real case is shown.

6.1 Peformance using ROC curves

The ROC curves were computed as described in 5.3. Since we only keep in the feature space S_{Signal} the descriptors associated to the labels *Background* and *Motion*, this feature space only involves 7.5 million features whereas the feature space used for the reference technique involves 17.5 million descriptors. For building the ROC curves and in order to compare to the reference technique we use the following parameters for the queries $p = 30$ (correspond to the average 0.8 key frame per second of the reference technique) and $n = 20$. Figure 6 shows that our method is more efficient than the reference technique with this evaluation.

In a CBCD application, at the end, the results are presented to a human operator who confirms or not the detection for the copyright management. Because of the large video database used (300 hours for the experiments but more in an industrial system), the *False Positive Frames Rate* must be very low. Missing a frame of *BenchPositiveAttack* can be corrected by finding close frames. Figure 6, shows that at the beginning of the curves the reference technique presents a better *True Positive Frames Rate* for the same *False Positive Frames Rate*; but the curves cross for false positive frames rate equal to 1 % and for a false positive frames rate equal to 2 % (which corresponds to an average of 0.5 false frames detected per second) the recall is 82 % for our technique compared to 72 % for the reference technique. This ROC curve presents an evaluation using only two video sequences and even if the videos are long videos (24 minutes and 3 hours), a CBCD system is supposed to deal with copies with different lengths in a video stream. The next section presents a more realistic evaluation closer to a real situation.

6.2 Performance with dedicated benchmark

These experiments are a way to evaluate the system in a simulated "real" situation: transformed video segments are randomly put into a long video stream and our goals are to find these video segments first and then to find them as precisely as possible. For this evaluation, we have created two video benchmarks (*Bench1min*

News show 1993 (c)

Brazil Vs France 1977 (c)

Chronique Bretonne 1970 (c) Ina.

Figure 5: Examples of copies retrieval. On the left, video from the video test (video sequences with transformations). On the right, retrieved video from the database.

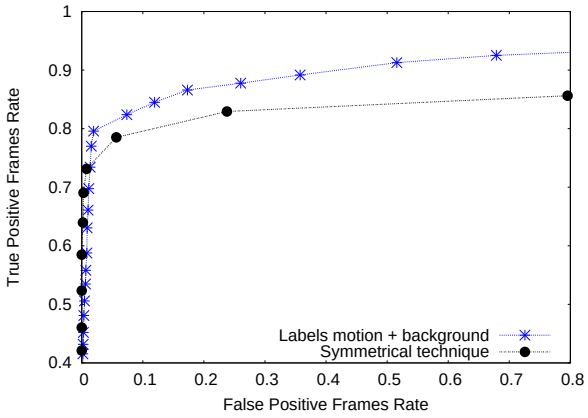


Figure 6: ROC curves.

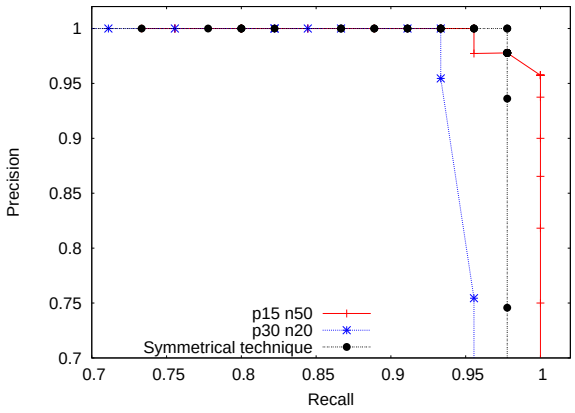


Figure 7: Precision recall by segment for *Bench1min*.

and *Bench30*) for computing the precision-recall curves which are described in section 5. We use two sets of parameters for our technique:

- $p = 30$ and $n = 20$ in order to have the same number of queries as the reference technique,
- $p = 15$ and $n = 50$ in order to test the improvement by increasing the number of queries.

6.2.1 *Retrieving video segments*

Here, we consider that a video segment is detected if at least one frame is detected which corresponds to our first goal: finding at least all the copies in the video stream. It is obvious that the most important component for the video copyright management is to find the most video segments possible. The figure 7 presents the result for the test called *Bench1min*.

Two remarks can be made:

- The two techniques present very good results for 1 minute length video sequences with a recall at the precision 95 % over 90% for the two techniques.
- By increasing the number of queries, we succeed in finding all the video sequences (100% compared to the reference technique with 97 %).

The second benchmark *Bench30* video is more difficult than the previous one because the video segments (which are taken from the same videos in the database as in *Bench1min*) can be very short (less than 1 second for some segments). Figure 8 shows the difficulty with curves lower than for the first benchmark:

- The fall of the precision occurs at 52% of recall for the reference technique whereas it occurs at 64% for our technique with the same number of queries.
- Increasing the number of queries is a way of increasing the recall rate but it costs a loss in the precision because some false alarms appear and the precision decreases at 44% of the recall.

For the two video tests, the recall for an acceptable precision is better with our technique using the specific vote: for a 90% precision, the rate of well retrieved segment is 100% for our technique compared to 97% for the reference technique for the video test *Bench1min*. The advantage of our technique is better highlighted by this curve with short video segments: 71% for our technique compared to 55% for the reference technique for 90% precision with the video test *Bench30*.

6.2.2 *Temporal precision of the detected segments*

For these experiments, the video segments are considered as consecutive frames and the precision and the recall are computed on the

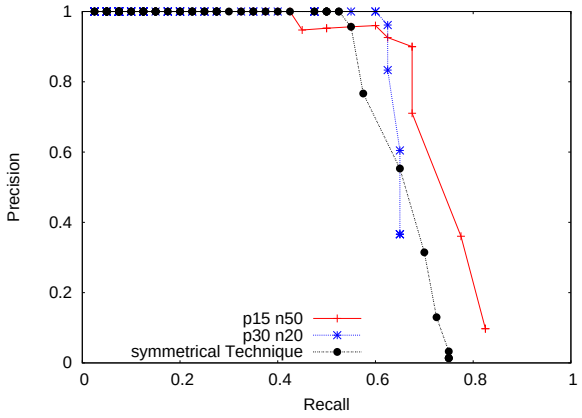


Figure 8: Precision recall by segment for *bench30*.

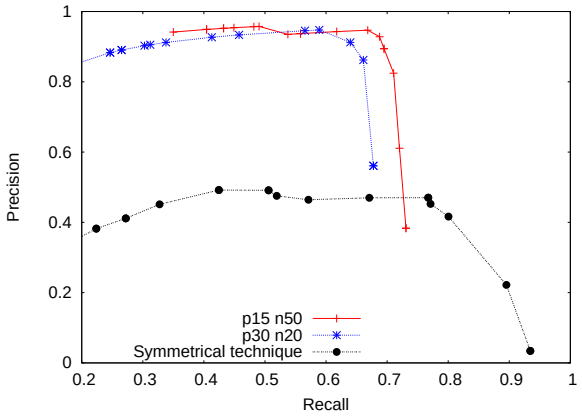


Figure 10: Precision recall by frames for *bench30*.

number of well retrieved frames and bad retrieved frames. This corresponds to temporal precision because the first experiment shows the quality in term of detected segments but these detected segments must be the most precise. If a video segment is detected with poor precision, it is a problem for the human operator to take the final decision. The curves 9 and 10 show that temporal precision.

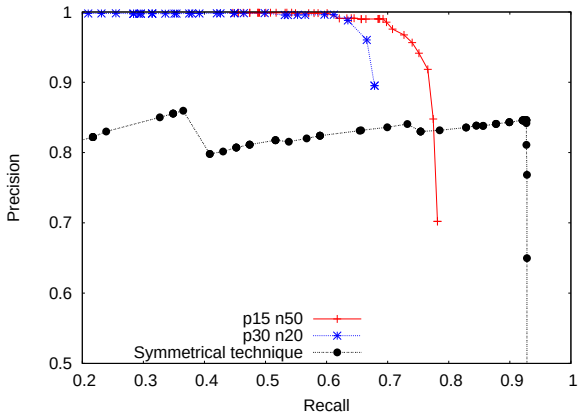


Figure 9: Precision recall by frames for *Bench1min*.

These two figures lead to the following observation:

- The reference technique "over detects" the segment: the detected segment is almost always too long, this explains the low precision in terms of detected frames.
- Our technique has a lower maximum recall but a much better precision which is very important in an industrial system with a human final operator.

For the test called *Bench30*, the over detection of the reference technique is also the reason of a very low precision: 45 % whereas the precision of our technique is 90 %. The difference is bigger than in the first experiment because of the very short video segments.

6.2.3 Conclusion of the evaluation

These video tests show that our technique is better than the reference technique which is a state-of-the art technique and especially for the short video segments. The precision is always better except when some false alarms appear due to the high number of queries so our technique is more efficient in terms of detected video segments. The use of labels allows to have a feature space smaller in terms of number of descriptors. Our technique is also more precise for the detected video segments which is important in an industrial system where a human operator must validate the results at the end of the retrieval process. Another advantage of our technique is that it is more flexible in the type of query parameter so we can choose the granularity of the detection which is not possible in the reference technique. The fact that the whole video is described during the off-line indexing process explains those performances even for the very small video segments.

6.3 Computational costs

As we work on a large set of videos, and as we want a final real-time system, we have to design fast strategies. This section gives some time computational information. The off-line indexing which consist of defining the trajectories parameters (see section 3) is 1.5 times slower than the real time with a standard PC (Pentium IV, 2.5 GHz, 1 Go RAM) but no computational optimizations have been applied on the code. The main CPU costs is the Harris points of interest detection whereas the building trajectories part is very fast. We have to remember that this description ie computing the features spaces S_{Traj} and S_{Signal}, is only done once and then from this first indexing step, it is possible to generate all the high-level descriptors required for the considered applications. The on-line part is really fast, 6 times faster than the candidate video length; this is why we are talking about real time: the system can work on continuous stream (TV channel) with just a constant delay. The next table 1 gives some values of the computational time for each step of the system given by the Linux *time* function.

6.4 A Hard Real Case

A french TV show uses video archives of singers from the 60's and mixes those video sequences with new videos of actual singers. This show is a perfect case for testing our technique. The searches were conduced using 20 hours of video from the INA archive corresponding to the video archives used for the TV show. Those 20 hours were indexed and put in our 300 hour video database. Figure 11 presents on the left, the video stream on the French TV and on the right, the retrieved videos from the 320 hour video database.

This test is an extreme case in the sense that the videos are really

Off-Line Indexing	300 hours of videos	
Computing S_{Traj} and S_{Signal} (see section 3)	450 hours	0.7 R.T.
Building the feature space	5 min	3600 R.T.
On-Line Detecting	7 hours of queries	
Computing queries (see section 4.1)	45 min	9 R.T.
Probabilistic search (see section 4.2)	15 min	28 R.T
Spatio-Temporal Registration (see section 4.3)	2 min	210 R.T
Combination and propagation (see section 4.4)	5 min	84 R.T.
Total	67 min	6 R.T.

Table 1: Computational Time: Measured time and time compare to Real-Time (R.T.).

heavily modified. In the detection (a) of figure 11, a second person is added to the scene with a shift of the old singer. The shift can sometimes be very high (b). The videos were also modified by the post-production in terms of color: the detections (c) and (d) show a strong transformation of gamma and illumination. The detection (e) presents a picture where only the singer was kept in the final video and the background is different which is a very hard case.

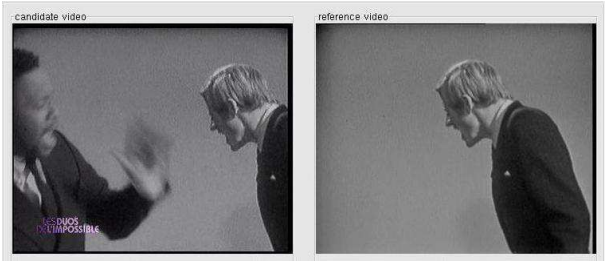
For this test, we have modified the voting algorithm for the propagation step. The beginning of the detection can contain two labels (*Background* and *Motion*) or only the label *Motion* because the transformation can add motion by adding a person for example but cannot delete the motion which is generally the interesting part of the video. By using our technique, we found short sequences not detected by the reference technique which confirms the increased precision already seen in 6.2. For a 2 hours show, we found 40 video segments not detected by the reference technique which correspond to 2 minutes and 51 seconds of video. Those added segments are short video sequences; the average length of those video segments is 4.3 seconds. Table 2 presents the results and sums up the improvements. Only one video segment is found by the reference technique and not by our technique. These results confirm the robustness of the method and the precision of the algorithm even for the short video sequences. The improvement (36 % in terms of video length) is very significant for a copyright management application.

This test which used a real TV stream, illustrates the acuracy and the performances of our method for video copy detection in a real situation.

Retrieved segments with reference technique					43
Retrieved segments with our technique					82
Good detections with reference technique					7min 53s
Good detections with our technique					10min 44s
Video segments found with reference technique					
Segments length L	$L < 1s$	$L > 1s$ $L < 5s$	$L > 5s$ $L < 10s$	$L > 10s$ $L < 20s$	$L > 20s$
Numbers of videos	0	11	10	17	5
Video segments found with our technique					
Numbers of videos	7	35	13	21	6
Segments added	7	24	4	4	1

Table 2: Results for a real case.

(a) Left: *Les duos de l'impossible* 2005, Right: *Palmarès des Chansons* R. Pradines 1966 (c) Ina..



(b) Left: *Les duos de l'impossible* 2005, Right: *Vient de Paraitre.* J. Guyon 1965 (c) Ina.

(c) Left: *Les duos de l'impossible* 2005, Right: *Sacha Show.* JP. Marchand 1963 (c) Ina.

(d) Left: *Les duos de l'impossible* 2005, Right: *Rendez Vous Avec.* F. Chatel 1961 (c) Ina.

(e) Left: *Les duos de l'impossible* 2005, Right: *Système deux.* C. Fayard 1975 (c) Ina.

Figure 11: Detection on Real cases: video queries on the left and videos detected on the right.

7. CONCLUSION AND PERSPECTIVES

In this paper, we have presented a robust and efficient approach for content-based video copy detection. It is based on two complementary contributions: the first is the smart description of local descriptors behaviors during the off-line indexing process by computing temporal contextual information and the second is the use of those informations in a robust voting function. The use of labels allows to compact the feature space with an efficient retrieval using a robust vote function. This vote function consists of a smart spatio-temporal registration of the queries with the features spaces computed during the off-line indexing process. It also uses the labels for improving the retrieval. The copy detection system is evaluated on different video benchmarks and finally on a real case. This evaluation is based on the number of video segments that are well detected despite the signal transformations (synthetic for the benchmark videos and real for the TV show) and also on the precision of these detections. These evaluations have shown a real improvement in terms of copy detection and in terms of precision of the detections facing a state-of-the art technique. Another advantage of this system is the flexibility of the parameters which allows one to choose the granularity of the detections in the on-line process. Last but not least the on-line retrieval method is faster than the real time on large video databases as proven in this paper.

Future work will consist of an automatic analysis of the trajectory parameters, potentially based on non-supervised classification methods in order to improve the labeling of the local descriptors. An improvement on the voting function can be done to make it robust even if there is slow motion or zoom. The model of the registration would be more complex in the spatio-temporal registration step. Another direction is to explore the nature of other local descriptions in order to use the complementarity of those descriptions in the same way that we use the complementarity of the labels of behavior in the voting function. This could be a way of reducing the number of false alarms and increasing the precision of copy detection.

8. REFERENCES

[1] J. Amores, N. Sebe, and P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. In *Conference on Computer Vision and Pattern Recognition*, 2005.

[2] S.-A. Berrani, L. Amsaleg, and P. Gros. Robust content-based image searches for copyright protection. In *ACM Intl. Workshop on Multimedia Databases*, pages 70–77, 2003.

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

[4] D. Chetverikov and J. Verest. Tracking feature points: A new algorithm. In *International Conference on Image Processing*, pages 1436–1438, 1998.

[5] M. Grabner and H. Bischof. Extracting object representations from local feature trajectories. In *1st Cognitive Vision Workshop*, 2005.

[6] A. Hampapur and R. Bolle. Comparison of sequence matching techniques for video copy detection. In *Conference on Storage and Retrieval for Media Databases*, pages 194–201, 2002.

[7] C. Harris and M. Stevens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 153–158, 1988.

[8] C. Hue, J. L. Cadre, and P. Perez. Tracking multiple objects with particle filtering. Technical report, IRISA, 2000.

[9] P. Indyk, G. Iyengar, and N. Shivakumar. Finding pirated video sequences on the internet. Technical report, Stanford University, 1999.

[10] A. Joly, C. Frelicot, and O. Buisson. Feature statistical retrieval applied to content-based copy identification. In *International Conference on Image Processing*, 2004.

[11] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Int. Conference On Multimedia*, 2004.

[12] I. Laptev and T. Lindeberg. Space-time interest points. In *International Conference on Computer Vision*, 2003.

[13] J. Law-To, V. Gouet-Brunet, O. Buisson, and N. Boujemaa. Local Behaviours Labelling for Content Based Video Copy Detection. In *International Conference Pattern Recognition*, 2006.

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *Interbational Conference on Computer Vision*, pages 1150–1157, Corfu, 1999.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *International Conference on Pattern Recognition*, 2003.

[16] E. N. Mortensen, H. Deng, and L. Shapiro. A sift descriptor with global context. In *Conference on Computer Vision and Pattern Recognition*, 2005.

[17] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Generic object recognition with boosting, 2004. Technical Report (submitted to [PAMI] 07/04).

[18] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9:56–73, 1987.

[19] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[20] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Conference on Computer Vision and Pattern Recognition*, June 2004.

[21] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Apr. 1991.