The Orange Labs Real Time Video Copy Detection System - TrecVid 2008 Results

Nicolas Gengembre, Sid-Ahmed Berrani
Orange Labs – Division R&D Technologies
4, rue du Clos Courtel
35510 Cesson Sévigné. France.
{nicolas.gengembre, sidahmed.berrani}@orange-ftgroup.com

October 27, 2008

Abstract

In this paper, we describe the content-based video copy detection system developed at Orange Labs. We also present the evaluation results of the TrecVid 2008 copy detection evaluation task. Target applications of video copy detection are mainly related to video copyright protection and video database management (e.g. duplicate detection). We have taken into account the constraints of these applications when building our system. We have in particular focused on the objective of achieving an optimal trade-off between effectiveness (i.e. robustness to transformations) and efficiency (i.e. rapidity). The 3 runs we have submitted to TrecVid 2008 are described and analyzed according to the evaluation criteria defined for the benchmark. The obtained results show that our goal of having an efficient and effective system has been globally reached.

1 Introduction

Huge and increasing amounts of videos are broadcasted through networks. This matter of facts raises the need of automatic video identification, for copyright protection as well as for near-duplicate detection. The most generic way to achieve this goal consists of analyzing the content of the videos, extracting *fingerprints*, and comparing these fingerprints.

A content-based video copy detection system is composed of two stages:

1. An off-line step during which fingerprints are computed from the referenced videos. Fingerprints are vectors that describe the visual/audio content of the video and are intended to be invariant to transformations the video may undergo. Fingerprints are also stored in an indexing structure in order to make similarity search efficient (i.e. rapid),

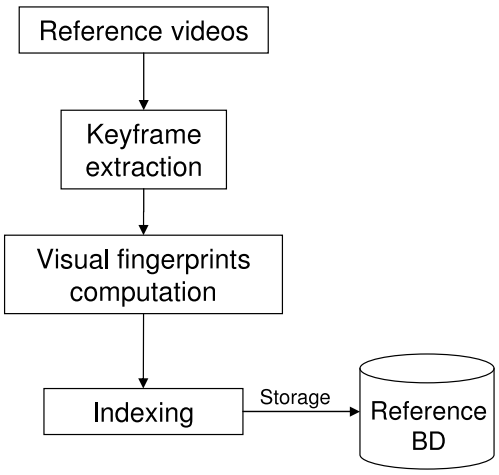

Figure 1: The Off-line step of our video copy detection system.

2. An on-line step during which suspicious videos are analyzed. Fingerprints are extracted from these videos and are compared to those stored in the reference database.

A content-based video copy detection system has to be therefore robust and efficient. Efficiency is essential for real-world applications that manage large referenced video databases and that have to check a large number of queries per day. Robustness to transformations that pirated videos may undergo is also very important as copied videos may have been modified (deliberately or not) through manipulations such as cropping, compression, color or contrast adjustments, etc.

This paper presents the content-based video copy detection system developed at Orange Labs. An overview of the system is given in Section 2. This section also gives a detailed description of the main processing steps. Section 3 presents the evaluation results obtained on the TrecVid 2008 dataset. Section 4 concludes the paper and outlines future extensions.

2 System Description

The general scheme of our system for video copy detection is presented in Figures 1 and 2. Our system extracts visual fingerprints from a subset of selected frames in the video. Off-line, fingerprints are extracted from the referenced videos. They are indexed and stored in a reference database. On-line, in order to check a suspicious video (the query), fingerprints are first extracted from the query video. These are used to query the reference database and similar fingerprints are found for each query fingerprint. An adaptive thresholding of similarity scores and a spatial coherence verification are applied in order to

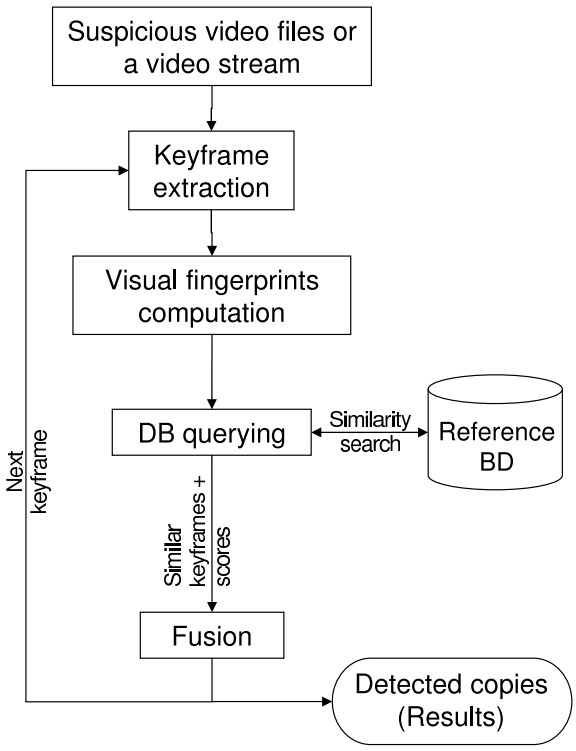

Figure 2: The On-line step of our video copy detection system.

decide whether a reference keyframe is similar or not to a query one. A probabilistic fusion module is also used in order to fuse the similarity search results. In addition, this module takes into account the temporal coherence between the keyframes of the query and the candidate keyframes from the referenced video.

The main important processing steps are detailed in the following subsections.

2.1 Keyframes extraction

A video is composed of a large set of ordered and highly correlated images. A first necessary step before describing the video is thus to select a subset of these images. The objective is to focus on a reduced set of images during the description step. This allows to significantly reduce the amount of data to store and to index and therefore increase the efficiency of the copy detection process.

These selected images are commonly called keyframes.

A number of methods can be found in the literature to solve the keyframe extraction problem [3, 7].

In our system, we first detect the boundaries of the shots. Then, each shot

is analyzed for keyframe extraction. The technique we have used relies on the variation of image values between each two consecutive frames. This quantity is cumulated over time to build an increasing signal. Changes in the evolution of this signal indicate frames with significant transitions. These changes are detected by a Page-Hinkley test as described in [1]. A keyframe is then defined as the median frame between two consecutive transition frames.

2.2 Fingerprint computation

The fingerprint we propose to use relies on local visual descriptors. Each keyframe is not considered as a whole but as a set of regions of interest. Around each region, a visual feature vector is computed. With this approach, our system is robust to transformations which discard parts of the frames (e.g. cropping or pattern inlay). If parts of the frames are missing, the remaining part are sufficient to match the copied video with the original one.

2.3 Indexing structure

During the off-line step, fingerprints are computed from the referenced videos. These fingerprints are used later on-line when a suspicious video is presented. Fingerprints are computed on the query keyframes and their similar fingerprints from the reference database are found.

The number of fingerprints in the reference database is generally huge as the number of referenced videos is very large. It is therefore important to index these fingerprints in order to accelerate the similarity search. The indexing structure we have used is derived from a hash table. Each fingerprint is mapped to a 32-bits word.

2.4 Scoring and thresholding

A referenced keyframe is assumed to be similar to a query one if the two keyframes share a minimum number of fingerprints. The number of shared fingerprints defines the similarity score and the threshold above which two keyframes are considered similar is computed using the *a contrario* method.

It is an adaptive and parameter-free method. It is based on a probabilistic approach that uses an a contrario modelling of the similarity. By these means, the resulting decision threshold is well-adapted to the number of fingerprints stored in the reference database, to the number of fingerprint in the query ketframe and also to the rareness of fingerprints.

A detailed description of the method is given in [5].

2.5 Spatial and temporal coherence verification

Once fingerprints of a referenced keyframe have been matched to those of a query keyframe, a spatial coherence verification between matched regions is performed. The objective is to remove possible false and random matches.

Number	Transformations
1	Cam Cording
2	Picture in picture type 1 (original video in front)
3	Insertion of patterns
4	Strong reencoding
5	Change of gamma
6	Random combination of 3 transformations amongst: blur, gamma, frame dropping, contrast, compression, ratio, noise
7	Random combination of 5 transformations amongst the list used for transformation 6
8	Random combination of 3 transformations amongst: Crop, shift, contrast, caption, flip, Insertion of pattern, picture in picture type 2 (original video behind)
9	Random combination of 5 transformations amongst the list used for transformation 8
10	Random combination of 5 transformations amongst all the transformations from 1 to 9

Table 1: List of the transformations that have been applied to the query videos.

Similarly, when keyframes of a referenced video have been identified at different times within a query video, the temporal coherence of the keyframes is checked. This is performed using a procedure that is based on a probabilistic Markovian framework. It makes use of the temporal consistency in order to compensate for possible mis-detections or false alarms during the fingerprint similarity searches. It shares some of its formalism with the Bayesian sequential filtering [2].

A detailed description of the method is given in [4].

3 Experiments

This section presents the results we have obtained with our system on the TrecVid 2008 dataset. We first briefly introduce the evaluation protocol that has been defined and the test dataset. We then describe our thee runs and finally we analyze the obtained results. Our results are also compared to the 48 other submitted runs.

3.1 Evaluation protocol

The TrecVid organization committee [6] has defined 10 different types of transformations, each of them being applied to 201 sequences, what leads to a total of 2010 queries. Two thirds of these queries are (or contain) sub-videos that

have been randomly selected from the video reference database. The others are not refreferenced videos. The reference database contains 206 hours of videos.

The applied transformations are numbered and are summarized in Table 1.[1]

A set of evaluation criteria has been defined. The two most important ones are the so-called MinNDCR and the computation time. The NDCR criterion[2] is defined as follows:

$$NDCR = \frac{FN}{134} + \frac{FP}{21.5}, \tag{1}$$

where FN corresponds to the number of false negative answers and FP is the number of false positive answers.

The NDCR allows to measure the ability to detect copies as well as to avoid false alarms. The smaller the NDCR, the better the robustness is.

These two criteria (i.e. NDCR and computation time) are combined and weighted with respect to a predefined use-case. In TrecVid 2008, as can be seen in the equation above, the NDCR has been configured so that false alarms are much more important than mis-detections.

The result of a video copy detection system consists of a list, possibly empty, of results for each query video. Each result is composed of the ID of a reference video, a confidence score, and the temporal location of the detected segment. Different tentative thresholds are used in order to cut the result list and to keep only videos with the highest confidence scores. The MinNDCR is therefore computed as the NDCR corresponding to the threshold achieving the best performance. We note that an optimal threshold is computed per transformation.

3.2 Description of the runs

We have submitted three sets of results. These correspond to three different runs during which our system was configured differently. The main internal parameter that has been modified is related to the number of fingerprints that have been extracted from each keyframe. However, this number is not necessarily the same for referenced keyframes and the query ones: reducing the number of fingerprints per reference keyframe allows to reduce the size of the reference database and hence increase efficiency, where increasing the number of fingerprints per query keyframe allows the improvement of robustness. A significant impact of transformations is the modification of saliency of regions of interest. Hence, increasing the number of fingerprints (and so the number of regions of interest) may increase the chance to keep detecting regions of the interest in the query that insure the detection.

Two different fingerprint databases have been computed : for the first one, 150 fingerprints have been extracted from the referenced keyframes, and for the second one, 200 fingerprints have been extracted.

[1] See document entitled "Final list of transformations" at the following url: *http://www-nlpir.nist.gov/projects/tv2008/active/copy.detection/final.cbcd.video.transformations.pdf*

[2] See document entitled "Final CBCD evaluation Plan TRECVid 2008" at the following url: *http://www-nlpir.nist.gov/projects/tv2008/Evaluation-cbcd-v1.3.htm*

Runs	Number of fingerprints per referenced keyframe	Number of fingerprints per query keyframe
Run 1	150	100
Run 2	200	200
Run 3	200	300

Table 2: Parameters of the three Orange Labs runs submitted to TrecVid 2008 Copy Detection Task.

Regarding the analyzing step, we have also chosen different numbers of fingerprints. The parameters used for the three runs are summarized in Table 2.

3.3 Results analysis

The obtained results are summarized in Table 3 and Table 4. Table 3 gives the performance of our three runs in terms of MinNDCR for each transformation. It also provides the average value of MinNDCR for all the transformations.

	1	2	3	4	5	6
Run1	0.484	0.186	0.096	0.412	0.027	0.44
Run2	0.48	0.188	0.111	0.433	0.061	0.435
Run3	0.498	0.148	0.119	0.435	0.042	0.425

	7	8	9	10	All
Run1	0.729	0.076	0.173	0.643	0.3266
Run2	0.754	0.077	0.184	0.643	0.3366
Run3	0.739	0.122	0.176	0.687	0.3391

Table 3: MinNDCR values obtained for the three Orange Labs runs and for the 10 transformations (columns). The last column (labeled *"All"*) is the mean value computed over all the transformations.

Table 4 gives for each run, the cumulated processing times over all the queries and the *acceleration factor w.r.t. real time* (AFRT). This factor equals the total length of queries (i.e. the sum of the queries durations, which equals $155,660$s) over the total cumulated processing time. The higher the acceleration factor, the faster the system is.

The three runs, and hence the three corresponding sets of parameters, lead to very similar results in terms of robustness. Of course, Run 1 which computes fewer fingerprints per keyframe is faster.

We have used the mean value of MinNDCR and the AFRT in order to assess the trade-off between effectiveness and efficiency of the runs that have been submitted to this campaign. Effectiveness has been defined by (1−MinNDCR) so that the higher the value, the better the robustness of the system is. We note

	Total processing time (s)	Acceleration factor w.r.t. real time
Run 1	41,588	3.74
Run 2	50,906	3.06
Run 3	57,269	2.72

Table 4: Processing times for the three Orange Labs runs.

that the definition of MinNDCR implies that (1−MinNDCR) cannot be greater than one, but it can be negative.

In Figure 3, the MinNDCRs and the AFRTs of the 15 most effective runs are plotted. Our three runs are plotted in red. They are respectively the fifth, sixth and seventh most effective ones among the 48 runs submitted by the TrecVid participants (22 different teams have submitted results among the 56 teams initially registered).

In addition to these good results, a major property of our runs is their very good efficiency: they are 5 to 30 times faster than the four most effective runs. This very good trade-off between effectiveness and efficiency is very important for real-world applications, specially when large reference databases have to be indexed.

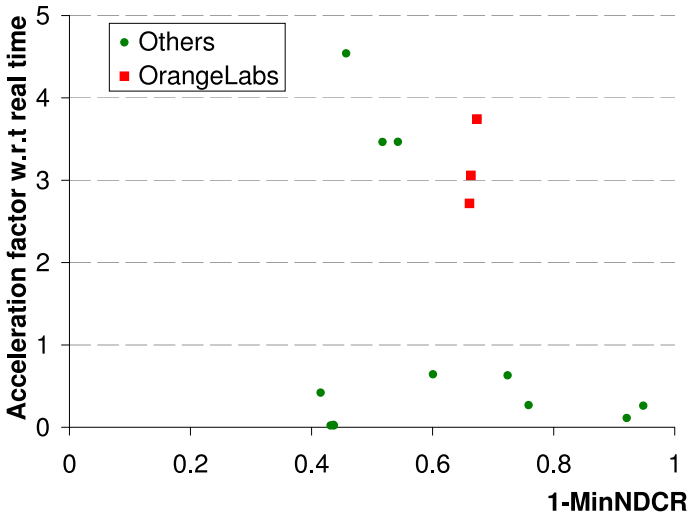

Figure 3: Efficiency vs. effectiveness: the best 15 runs of the TrecVid 2008. The Orange Labs runs are plotted in red, the others in green

On the other hand, if we focus on the results presented in Table 3, we can notice that most of the transformations for which our results are medium, involves changes in the aspect ratio (transformations 6 and 7, but also camcording, i.e.

type 1, for which some image warping appears). This can be explained by the way the regions of interest are computed. These are circular which is probably responsible for this decrease in effectiveness in these cases. The extension of our description scheme to elliptical regions of interest would probably significantly improve the overall performance of the system.

4 Conclusions and perspectives

This paper describes the content-based video copy detection system developed at Orange Labs and summarizes its performance on the TrecVid 2008 benchmark. The obtained results show a very good behavior both in terms of effectiveness and efficiency. Our system ensures an optimal trade-off between robustness to transformations and computation time. This is very important for industrial applications that might deal with large reference databases and where the system has to be tuned with respect to the application constraints.

Our future extension will focus on how to improve the effectiveness in particular to severe transformations like 6, 7 and 10. This however will only be done after a careful analysis of these transformations in order to assess whether this is worth or not. Indeed, some transformations deteriorate the visual content so that the video becomes completely useless. These transformations have therefore not to be considered.

References

[1] M. Basseville and A. Benveniste. Desgin and comparative study of some sequential jump detection algorithms for digital signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(3):521–535, June 1983.

[2] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000.

[3] B. Fauvet, P. Bouthemy, P. Gros, and F. Spindler. A geometrical key-frame selection method exploiting dominant motion estimation in video. In *Proceedings of the ACM International Conference on Image and VideoRetrieval*, Dublin, Ireland, July 2004.

[4] N. Gengembre and S.-A. Berrani. A probabilistic framework for fusing frame-based searches within a video copy detection system. In *Proc. of the ACM International Conference on Image and Video Retrieval*, Niagara Falls, Canada, July 2008.

[5] N. Gengembre, S.-A. Berrani, and P. Lechat. Adaptive similarity search in large databases application to video copy detection. In *Proc. of the 6th International Workshop on Content-Based Multimedia Indexing*, London, UK, June 2008.

[6] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proc. of the 8th ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, CA, USA, October 2006.

[7] Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrota. Adaptative key frame extraction using unsupervised clustering. In *Proceedings of the 5th International Conference on Image Processing*, volume 1, Chicago, IL, USA, October 1998.