INRIA LEAR-TEXMEX: Video copy detection task

Hervé Jégou, Matthijs Douze, Guillaume Gravier, Cordelia Schmid, Patrick Gros

INRIA LEAR-TEXMEX: VIDEO COPY DETECTION TASK

Hervé Jégou[1]*, Matthijs Douze*[2]*, Guillaume Gravier*[3]*, Cordelia Schmid*[2] *and Patrick Gros*[1]

email: firstname.lastname@inria.fr

[1]*INRIA Rennes* [2]*INRIA Rhône-Alpes* [3] *CNRS, IRISA Rennes*

ABSTRACT

In this paper we present the results of our experiments in the TRECVID'10 copy detection task [9] and introduce the components of our system. In particular, we describe the recent approximate nearest neighbor search method [6] we used to index the hundreds millions of audio descriptors. Our system obtained excellent accuracy and localization results, achieving the best performance on a few transformations, and this with a single kind of image descriptor. Moreover, the analysis evidences that our system can be significantly improved.

1. SUBMISSION OVERVIEW & INTRODUCTION

We have submitted four runs in total, two for each of the profiles: NOFA (No False Alarm) and BALANCED. The runs essentially differ in the video system and the fusion/scoring method. The main properties of the runs are the following:

Runid	Profile	Video
LAPIN	BALANCED	$\approx$ TRECVID'08
TRUITE	BALANCED	improved
MOUFLON	NOFA	$\approx$ TRECVID'08
BOUQUETIN	NOFA	improved

Our first two submitted runs, LAPIN and MOUFLON, were very similar to our TRECVID'08 submission [1] with respect to the video modality[1]. This submission obtained the top results in the copy detection evaluation in 2008. The system we used is detailed in [2].

We have built upon this video system and improved its performance, at the cost of efficiency. In particular, we have optimized some key parameters on a validation set to produce the video description for the TRUITE and BOUQUETIN runs. These two runs also give more importance to the audio modality, especially for the TRUITE run. This was not a good choice. Firstly, our audio system alone is not as strong as our video system alone. Secondly and most problematically, our audio system faced unexpected problems with the test set, see Section 3 for details. As a result, the TRUITE run, which was expected to be the best one in the BALANCED profile, was outperformed by the LAPIN run.

Another problem we observed is that we were not able to correctly set the threshold used in the computation of the actual Normalized Distance Cost Ratio (Act_NDCR). Our results for this actual cost are poor, despites excellent results with respect to the optimal cost (Opt_Min_NDCR). Apparently, several participants faced the same problem. A possible interpretation is that the difficulty of our validation set is not consistent with the test set of TRECVID'10.

The rest of this notebook paper is organized as follows. In section 2 we briefly review our TRECVID'08 video system. It is mainly based on the method we used in the TRECVID'08 evaluation [1], which obtained the top results during this campaign. We focus, in particular, on the improvements we introduced this year. In section 3 we present our audio detection system. The proposed method is based on a publicly available audio feature extraction package and can, therefore, be re-implemented quickly. Our matching system relies on a recent approximate nearest neighbor search method that is able to index many different types of descriptors, as long as they are compared with the Euclidean distance. This ANN method is efficient and accurate and uses limited memory resources. Finally, we will present the results from our pure audio and video runs separately, as well as the resulting fused runs. The fused runs are shown to significantly outperform each of the mono-media runs.

2. IMAGE-BASED COPY DETECTION

This section presents an overview of our video copy detection system, which is based on the frame matching[2] system we successfully used for 2008's evaluation [1, 2]. In the following, we review the different components of this system and detail the main differences with the one we used in 2008.

2.1. Frame representation using local descriptors

Frame subsampling. We have extracted one frame out of 10, which corresponds to 1.5 to 5 frames/second, depending on the video frame-rate. We do not use key-frame selection, as

[1]Here, "video" refers to the sequence of frames extracted from the video clip, as opposed to the "audio" modality

[2]Although the term "frame" is also used in audio, in this paper it is employed to refer to images extracted from the video stream.

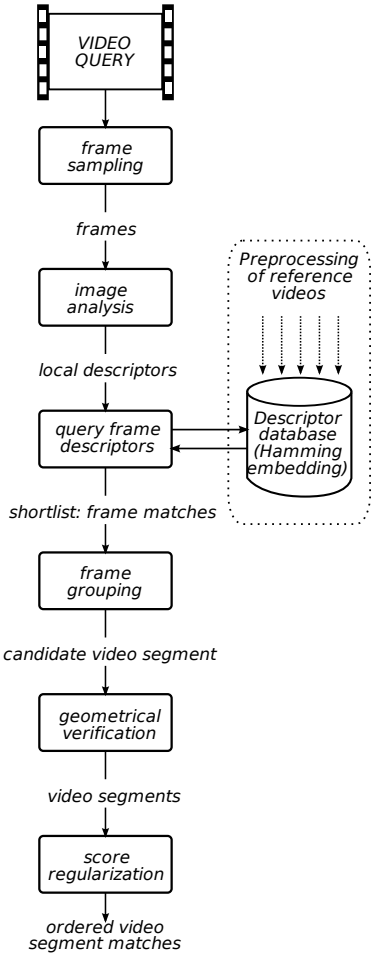


Fig. 1. Overview of video system (without audio)

we observed [1] that dense uniform sampling provides equivalent or better results.

Local descriptor extraction. We detect rotation-invariant Hessian-affine regions [7] in order to handle rotations, perspective distortions and scaling. However, as proposed in [2], we used the CS-LBP descriptor [3] instead of the SIFT descriptor to improve the efficiency of the descriptor extraction process. On average, we extract 350 descriptors per frame. The average frame description rate is about 0.6 frames per second, i.e., $4\times$ real time on one core.

2.2. Descriptor matching and frame scoring

Our frame matching system is the one described in [5]. It is an improvement of the Bag-of-features approach introduced by Sivic and Zisserman [8]. The key components are detailed below.

Visual codebook generation (off-line). The quantizer partitions the descriptor space. It is defined by a set of centroids. In the context of local descriptor matching, each centroid is usually referred to as a "visual word", belonging to a "visual vocabulary". Our visual vocabulary has been generated using the k-means algorithm learned on an independent set of 1M images. We have used $k = 200\,000$ visual words in all our experiments and runs.

Assigning the descriptors to visual words. Each local descriptor from the sampled frames is assigned to the closest visual word. This quantization step amounts to representing a descriptor by the corresponding centroid index $q(x)$.

Hamming Embedding. As shown in [5], the bag-of-words framework of [8] can be seen as a matching function that associates descriptors based on their quantization indexes. However, it tends to select incorrect matches, because quantization in high dimensional spaces groups together dissimilar vectors. To address this problem, a descriptor is represented by both the index $q(x)$ and a binary signature $b(x)$ of 48 bit, where $b(.)$ is the Hamming Embedding function associated with the visual word $q(x)$. It is designed so that the Hamming distance

$$h(b(x), b(y)) = \sum_{1 \leq i \leq 48} |b_i(x) - b_i(y)| \qquad (1)$$

between two descriptors x and y lying in the same cell approximates the Euclidean distance $d(x, y)$. A descriptor is now represented by $q(x)$ and $b(x)$. The descriptor matching function f_{HE} is then defined as

$$f_{\text{HE}}(x, y) = \begin{cases} w(h\,(b(x), b(y))) & \text{if } q(x) = q(y) \\ & \text{and } h\,(b(x), b(y)) \leq h_t \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where $h(.,.)$ is the Hamming distance defined in Equation 1, $h_t = 13$ is a fixed Hamming threshold and $w(.)$ is a weighting function that gives higher scores to smaller Hamming distances.

Weak geometric consistency. The idea of this method is to integrate geometrical information in the large index to filter out the descriptors that are not consistent with respect to a frame orientation and scale hypothesis. The method is detailed in [5], where the addition of a prior on the geometrical transformation is shown to improve the results. In the case of videos, the prior favors image matches without a global rotation.

Burstiness. The frame scoring method includes the method of [4] to handle visual bursts. Note that we did not use this extension in 2008. Burstiness refers to a statistical property of a random process, which in our case corresponds to the fact that a given visual element appears more times in an image than a statistically independent model would predict. In text retrieval algorithms, word burstiness is a well identified and carefully handled phenomenon. In the context of image search, burstiness corrupts the visual similarity measure, i.e.,

the scores used to rank the images. This is particularly critical for subtitles, as text on images is known to generate visual bursts. Although [4] shows the interest of handling burstiness in the context of image retrieval, on our copy detection validation dataset, the improvement brought by this method is limited.

Image matching improvements. The quality of frame matching is further improved as follows.

- Flipped images cannot normally be recognized by the image matching engine (except if the image content is strongly symmetric). Therefore, we also query a flipped version of the videos. This doubles the search time.
- We found that scaled images, especially embedded in clutter, are easier to recognize in a dataset of images that are also scaled down one half. Therefore, along with the normal image dataset, we maintain a dataset of half-sized images. This adds 30% to the search time.
- On the query side, instead of choosing only the nearest neighbor when assigning visual word, each descriptor is assigned to *several* nearest visual words. We perform multiple assignment for the query only, not for the indexed video dataset [4]. This limits the memory usage of the frame indexing structure. This multiple assignment strategy was used only in the runs TRUITE and BOUQUETIN;
- It is well known that using more image descriptors per image improves the matching quality with discriminative methods like Hamming Embedding. Therefore, we decreased the Hessian local description threshold from the default 500. For a threshold of 200 (resp. 100), the average number of descriptors per image becomes 620 (resp. 830), increasing the search time by a factor of 1.7 (resp. 2.3).

Compared to our TRECVID'08 participation, we have made some updates following a systematic evaluation of the impact of the different steps mentioned above. For this purpose, we generated a validation set of 100 videos extracted from the 1000 first videos of the TRECVID dataset, as described in [2]. Table 1 sums up the performance gains obtained with the various improvements on this validation set. It shows that adding flipped and half-sized images is important. Multiple assignment helps, but the number of assignment can be kept small, as the improvement saturates when assigning descriptors to 10 visual words. Lowering the threshold of the descriptor also helps, especially when describing half-size frames. Therefore, we settled for the combination: detector threshold is set to 100, multiple assignment to 3 visual words on the query side, for both the image and its flipped version. A detector threshold 200 and half-sized images with detection at threshold 100 were set on the database side. As a result, querying one frame takes about 20 seconds on a single core.

query	database		
	T200	T200+H200	T200+H100
T200	0.483		
T100	0.514	0.568	0.583
T100+F	0.627	0.719	0.738
T100, MA10	0.543	0.643	0.631
T100+F, MA10	0.683	0.749	0.737
T100, MA3	0.504	0.596	0.592
T100+F, MA3	0.650	0.755	**0.761**

Table 1. Performance evaluation of the search on our validation set (average precision) with various combinations of analysis methods on the database and query video sides. Tx=detection threshold x, MAx = multiple assignment to x visual words, F=query flipped videos as well, Hx=use half-sized image database as well, with detection threshold x.

Our method is multi-threaded: several queries are performed in parallel on the different CPU cores.

Temporal integration of matches. Frame matches are integrated into segment matches using a 1D Hough transform. The results hypotheses are then verified by estimating a spatiotemporal matching model. The method used is identical to [2], except that it was adapted to accommodate for video segments with different frame-rates (videos from the Internet Archive dataset have very different frame rates).

3. AUDIO-BASED COPY DETECTION

3.1. Pre-processing

The audio tracks extracted from TRECVID'10 are not homogeneous. The frame rates as well as encoding quality vary significantly from one track to another. We have not exploited this meta-data information, for instance, using consistency of frame rate or the number of audio channels to improve the search: we believe that this would not correspond to a real life scenario, as the pirate is likely to resample the waveform or to apply a stereo-to-mono/mono-to-stereo transformation.

In order to deal with this variability in a consistent way, we resample all tracks to 32000 Hz, and use the right stereo channel only.

3.2. Feature extraction: filter banks

The signal is ran through a pre-emphasis filter to compensate for the spectral slope and divided into overlapping short-term windows of 25 ms taken every 10 ms. In each window, the short-term spectrum is represented by log-energies at the output of a 48-channel filter bank, The filters are overlapping band-pass filters spread along the [300 Hz,3000 Hz] frequency range on a Mel scale. This representation gives a rough approximation of the signal's spectral shape in the frequency range considered while smoothing out the harmonic

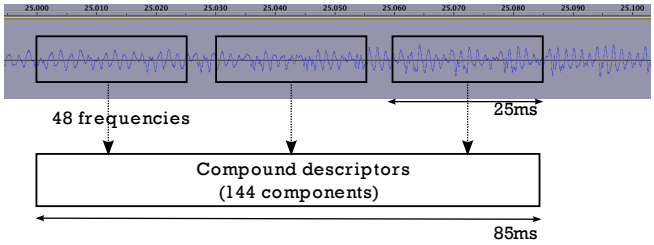


Fig. 2. Compound descriptor.

structure, if any, and is therefore robust to many spectral distortions. We have used the freely available `spro` software[3] for the generation of filter banks. More precisely, we have used the `sfbank` function off-the-shelf. Note that this software also includes an efficient implementation of the widely used MFCC descriptors.

3.3. Compound descriptors

At this point, the temporal consistency provided by a single filter bank is limited, as their temporal span is limited. This is problematic since the database is large: the filter banks themselves might be not discriminative enough to identify a matching hypothesis with sufficient reliability.

In order to increase the discriminative power of the descriptor, the temporal aspect is emphasized by concatenating several filter banks. For a given timestamp t, we extract 3 successive filter banks at timestamps $t-\delta$, t and $t+\delta$, producing a compound descriptor of dimensionality 144. We have performed a few experiments to decide on how to take into account this dynamic aspect (e.g., using derivatives of the filter bank with respect to time). Compounding the descriptors appeared a reasonable choice.

As illustrated by Figure 2, the resulting span of this descriptors is therefore 85 ms. This approach favors the temporal aspect by taking into account the dynamic behavior of the frequency energies, at the cost of an increased descriptor dimensionality.

We adopt the Euclidean metric to compare descriptors. For large vector databases it allows for efficient indexing algorithms, as the one described below. Note however that we have not performed a thorough evaluation of other possible distances/divergences because to our knowledge there is no efficient algorithm for such sophisticated metrics. In particular, we have not considered the metrics specifically developed to compare filter banks, such as the Itakura-Saito divergence or the log-spectral distance,

In order to take into account attacks on the volume (signal energy), the descriptor can be made invariant by subtracting its mean. We however feel that this choice is too radical. Instead, a partial normalization is applied: we subtract to the descriptor a fraction of its mean ($\times 0.8$ the mean), which provides an intermediate choice between discriminance and invariance. As our descriptors are compared with the Euclidean distance, subtracting part of the mean can be interpreted as a way of integrating, into the distance, a penalization term corresponding to the variation of energy.

[3] `http://gforge.inria.fr/projects/spro`

3.4. Approximate nearest neighbor search

As the exact search is not efficient enough, we have used approximate nearest neighbor search. Hereafter we briefly review the indexing method of [6], which finds the approximate k nearest neighbors using a source coding approach with limited memory. We first detail the Asymmetric Distance Computation (ADC) method to show the principle of compression-based search techniques. We will then briefly explain the more sophisticated variant we used in our submission.

Searching by compression: principle. For the sake of presentation, we assume that we search for the nearest neighbor (i.e., $k=1$), the extension more neighbors is obvious. Let $x \in \mathbb{R}^d$ be a query vector and $\mathcal{Y} = \{y_1, \ldots, y_n\}$ a set of vectors in which we want to find the nearest neighbor $\mathrm{NN}(x)$ of x. The ADC approach consists in encoding each vector y_i by a quantized version $c_i = \mathrm{q}(y_i) \in \mathbb{R}^d$. For a quantizer $\mathrm{q}(.)$ with K centroids, the vector is encoded by $b_\mathrm{c} = \log_2(K)$ bits, assuming K is a power of 2. We compute an approximate distance $d_\mathrm{c}(x, y_i)$ between a query x and a database vector:

$$d_\mathrm{c}(x, y_i)^2 = \|x - \mathrm{q}(y_i)\|^2. \quad (3)$$

The approximate nearest neighbor $\mathrm{NN}_a(x)$ of x is obtained by minimizing this distance estimator:

$$\mathrm{NN}_a(x) = \arg\min_i d_\mathrm{c}(x, y_i)^2 = \arg\min_i \|x - \mathrm{q}(y_i)\|^2, \quad (4)$$

which is an approximation of the exact distance calculation

$$\mathrm{NN}(x) = \arg\min_i \|x - y_i\|^2. \quad (5)$$

In contrast with the binary embedding method of [10], the query x is not converted to a code: there is no approximation error on the query side. To get a good vector approximation, K should be large ($K = 2^{64}$ for a 64 bit code). For such large values of K, learning a K-means codebook as well as assigning to the centroids is not tractable. Instead, the method of [6] uses a product quantizer, for which there is no need to explicitly enumerate the centroids. A vector $y \in \mathbb{R}^d$ is first split into m subvectors $y^1, ..., y^m \in \mathbb{R}^{d/m}$. A product quantizer is then defined as a function

$$\mathrm{q}(y) = \big(\mathrm{q}^1(y^1), ..., \mathrm{q}^m(y^m)\big), \quad (6)$$

which maps the input vector y to a tuple of indices by separately quantizing the subvectors. Each individual quantizer

$\mathrm{q}^j(.)$ has K_s reproduction values, learned by K-means. To limit the assignment complexity, $\mathcal{O}(m \times K_\mathrm{s})$, K_s is set to a small value (e.g. K_s=256). However, the set of K centroids induced by the product quantizer $\mathrm{q}(.)$ is large, as $K = (K_\mathrm{s})^m$.

The squared distances in Equation 4 are computed using the decomposition

$$d_\mathrm{c}(x, y)^2 = \|x - \mathrm{q}(y)\|^2 = \sum_{j=1,\ldots,m} \|x^j - \mathrm{q}^j(y^j)\|^2, \quad (7)$$

where y^j is the j^{th} subvector of y. The squared distances in the sum are read from look-up tables that are computed, prior to the search, from each subvector x^j and the k_s centroids associated with the corresponding quantizer q^j. The complexity of the table generation is $\mathcal{O}(d \times K_\mathrm{s})$. When $K_\mathrm{s} \ll n$, this complexity is negligible compared to the summation cost of $\mathcal{O}(d \times n)$ in Equation 4.

This approximate nearest neighbor method implicitly sees multi-dimensional indexing as a vector approximation problem: a database vector y can be decomposed as

$$y = \mathrm{q}(y) + \mathrm{r}(y), \quad (8)$$

where $\mathrm{q}(y)$ is the centroid associated with y and $\mathrm{r}(y)$ the error vector resulting from the quantization, called the *residual* vector. It is proved [6] that the square error between the distance and its estimation is bounded, on average, by the quantization error. This ensures, asymptotically, perfect search results when increasing the number of bits allocated for the quantization indexes.

The ADC indexing method is fully parametrized by the number of subvectors m and the total number of bits b_c.

Non exhaustive variant. Up to now, we have presented the ADC method. In practice, we use the IVFADC variant of [6], that avoids to exhaustively scan the dataset codes by using an inverted file structure. This requires an additional coarse quantizer. In addition to the numbers m and m' of bytes used to encode the vector, this variant requires two additional parameters: the number c of reproduction values of the coarse quantizer and the number v of inverted lists that are visited for a given query. The main advantage of this variant is that it computes the distance estimators only for a fraction (in the order of v/c) of the database, at the risk of missing some nearest neighbors if v/c is not large enough. Note finally that the memory usage per descriptor is increased by $\log_2(c)$ bits (typically rounded up to 4 bytes), due to the inverted file structure.

Parameters. As suggested in [6], we set set $b_\mathrm{c} = 8$ (i.e., $K_\mathrm{s} = 256$) for practical reasons (byte-aligned codes). Since our goal was to introduce little suboptimality by using approximate instead of exact search, we have parametrized our system so as to provide very good accuracy. We have used $m = 24$, which means that each descriptor is represented by $24 + 4 = 28$ bytes. A fixed number of $v = 128$ lists are visited out of $c = 16384$. These choices are probably too conservative, but our audio system requires significantly lower resources than our video system, and is therefore not the bottleneck in terms of efficiency. Even with these parameters, searching the neighbors for all the query descriptors is at least an order of magnitude faster than searching the local image descriptors for every frame.

3.5. Hough matching

Similar to what we have done in video, we exploit the fact that the transformations do not include any acceleration. Given a query, for each of its audio descriptors we first search for the 1000 approximate nearest neighbors using the approach exposed above. We then vote for several alignment hypotheses (a_b, Δ_T), weighting each match by a function of its rank (we choose $1/\log(1 + \text{rank})$). On our validation set, weighting brings a slight improvement at no cost in efficiency. The hypotheses with low scores are filtered. On output, this Hough matching system returns a maximum of 100 hypotheses per query.

3.6. Re-ranking and boundary detection

Similar to what is done in image or video retrieval, where the first stage is scalable but not very precise, we used a re-ranking stage to improve the scoring of the audio matches. For each query, this re-ranking stage processes several alignment hypotheses (a_b, Δ_T), where a_b is the database track and Δ_T is the difference between the query and the database timestamps.

We then use the whole set of descriptors and weight their distance to produce a score per time instant. This score is then filtered in time, and then used to detect the boundaries defining a matching segment. The score associated with the segment is computed from the individual scores for each time instant.

3.7. False silences

Our audio system had a problem with "false silences", i.e., the portion of the audio tracks containing almost no energy, leading to poor results on the audio part. A few videos were returned in first position (high scores) for many queries just because of segments containing those silences. We identified this problem analyzing the results provided by NIST. We were not able to identify this problem on our validation set, which was too "clean" for this problem to occur.

4. FUSION OF AUDIO AND VIDEO

Our fusion module is probably the weakest point of the system. This is because we did not used a consistent database for audio and video validation: TRECVID'09 for audio and our

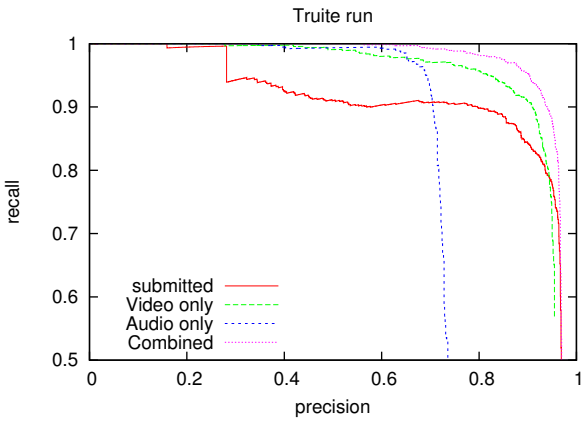


Fig. 3. Precision-recall plots of our run TRUITE, decomposed as audio and video and the new combination: the updated run (Combined) is exactly the same as the submitted run, but with a fix for the silences (filtered out).

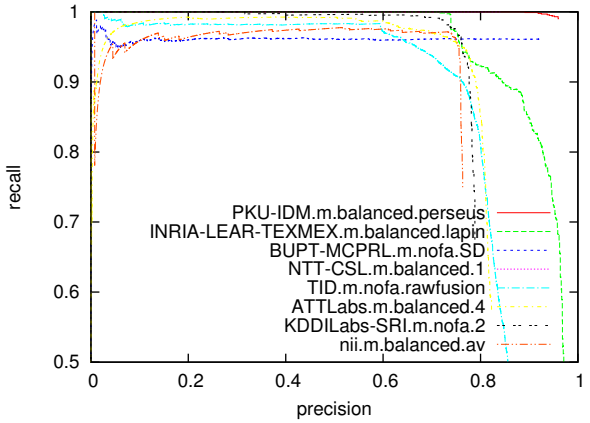


Fig. 4. Precision-recall plots of the best submitted runs of the 8 best participants.

validation set for video. Therefore, the fusion system was designed without any optimization performed on any validation set. In this section, we will therefore put an emphasis on the separate evaluation of audio and video.

4.1. Scoring and fusion strategy

The audio and video scores being regularized, we apply a scoring strategy very similar to the one we used in 2008, which severely penalizes the scores of the matching hypotheses which are not ranked first. On each modality, assuming that the best score obtained for the query is $\text{score}_{\text{best\_score}}$, we typically update the score of each matching video as:

$$\text{score}_{\text{vid}} = \text{score}_{\text{vid}} \times \left(\frac{\text{score}_{\text{vid}}}{\text{score}_{\text{best\_score}}} \right)^{\alpha} \qquad (9)$$

with $\alpha = 1$ for the BALANCED profile and $\alpha = 2$ for the NOFA profile. As in 2008, this choice seems to be fruitful. It is however not necessarily optimal: we have not optimized this score processing on the validation set. Results videos that are selected by both modalities are significantly favored, as the probability for an outlier to obtain a good score for both modalities is very low.

The fusion of the results from audio and video makes the assumption that the two modalities are equivalent with respect to search quality. Therefore, in our baseline method, we define a regularization function to produce scores having the same order of magnitude on average. However, we have not evaluated this point properly: as shown below, it appears that our video system is significantly better than our audio system.

4.2. Comparison of audio and video

In our two last runs TRUITE and BOUQUETIN, more importance has be given to the audio modality. This choice was unfortunate: not only did we face the problem of silences, but after correcting this the video system still appears to outperform the audio run on average. This is shown on Figure 3, which compares the performance of our audio system (with the "false silence" problem corrected), with the video system and the fused system. Here, the performance is measured in terms of mean average precision, see [2] for details. Figure 4 gives a comparison of the best submitted runs. Our video system appears to be obtain very good results, even used alone. The fusion with audio further improves the performance. This is mainly due to the fact that our fusion method significantly improves the score of a database video returned by both the audio and video systems.

We also show the performance of our submitted audio+video run TRUITE, where the problem with silences is visible: it produces a strong discontinuity at precision≈0.3. One can observe that our video system alone outperforms our submitted run, which means that the problem with audio had severe consequences on our overall submitted performance. This is especially the case for our runs TRUITE and BOUQUETIN, which gave more importance to the audio system.

5. RESULTS AND ANALYSIS

Figures 5, 6 and 7 presents the results generated by NIST for our three best runs. Our TRUITE run is significantly worse for the reasons exposed above and is not presented.

Despite our problem with the silences and our unfortunate choices in the fusion module, our runs obtained some excellent results in terms of precision. We obtained the best performance for a few transformation for the optimal NDCR measure. As shown in Table 2, relative to other participants, the transformations where we get better results are 20, 34, 35, 41, 42, 54, 55, 56. These correspond mostly to difficult video transformations (5, 6, 8) in combination with the hardest audio transformations (6, 7). The fact that are runs are compar-

transformation	our best run	NOFA rank (nb of ex-aequo)	our best run	balanced rank (nb of ex-aequo)
1	M.	3	L.	2
2	M.	3	L.	2
3	M.	3	L.	3
4	M.	4	L.	3
5	M.	2	L.	2
6	M.	3	L.	2
7	M.	2	L.	2
8	M.	3	L.	3
9	M.	3	L.	3
10	M.	3	L.	3
11	M.	4	L.	3
12	M.	3	L.	3
13	M.	4	L.	4
14	M.	3	L.	3
15	M.	5	L.	4
16	M.	4(1)	L.	4
17	M.	6	L.	4
18	M.	5	L.	4
19	B.	2	L.	2
20	B.	2(1)	**L.**	**1(1)**
21	B.	2(2)	L.	2
22	M.	5	L.	4
23	M.	5	L.	4
24	M.	6	L.	6
25	M.	7	L.	5
26	B.	4	L.	4
27	B.	2(2)	T.	4
28	B.	2	L.	4
29	M.	4	L.	4
30	M.	4	L.	4
31	M.	5	L.	4
32	M.	4(1)	L.	4
33	B.	2(2)	L.	2
34	B.	4(1)	**L.**	**1**
35	B.	4	**L.**	**1**
36	M.	4	L.	4(1)
37	M.	4	L.	4
38	M.	5	L.	4
39	M.	5(1)	L.	4
40	B.	3	L.	3
41	B.	**1**	L.	2(1)
42	B.	**1**	L.	2
50	M.	3	L.	3
51	M.	3	L.	3
52	M.	3	L.	3
53	M.	3	L.	3
54	B.	**1**	**L.**	**1**
55	B.	3	**L.**	**1**
56	B.	2	**L.**	**1**
64	M.	4	L.	3
65	M.	3	L.	3
66	M.	5	L.	5
67	M.	5	L.	3
68	B.	2	L.	3
69	B.	3	T.	3
70	B.	3	T.	3

Table 2. Ranking of our runs in Opt_Min_NDCR. The rankings are reported per participant, not per run (M.=mouflon, L.=lapin, B.=bouquetin, T.=truite). In bold: the transformations for which obtained the best result.

atively better for the most difficult transformations is not surprising, as our "bug" on silences has an impact independent of the transformation difficulty. Our timings are not good, but this is not surprising as we have tried to obtained the highest accuracy as possible.

NDCR Threshold: in our runs, we have significantly overestimated the threshold used in the computation of the actual NDCR (Act_Min_NDCR performance). As this parameter has a huge impact on the overall performance, our results for this actual cost are very poor.

6. CONCLUSION

We have learned a lot from our participation to TRECVID this year. Although we have obtained top performing results on a few difficult transformations, the analysis of our results shows that we can still obtain better performance on several points. First, our audio module is a prototype, which can be further improved. A simple filtering of the silences already gives a fair improvement with respect to our submitted runs. Second, we will put more effort in the future to the fusion of modalities, as this step has a critical impact on the final performance.

Concerning the evolution of the task, we believe that it would be interesting to evaluate audio and video separately, to identify the methods performing best in each modality and in the fusion step.

7. ACKNOWLEDGMENTS

This work was supported by the QUAERO project.

8. REFERENCES

[1] M. Douze, A. Gaidon, H. Jégou, M. Marszałek, and C. Schmid. INRIA-LEAR's video copy detection system. In *TRECVID Workshop*, November 2008.

[2] M. Douze, H. Jégou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. Multimedia*, 12(4):257–266, jun 2010.

[3] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, 2009.

[4] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, June 2009.

[5] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, February 2010.

[6] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(1):117–128, January 2011.

[7] K. Mikolajczyk and c. schmid. scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

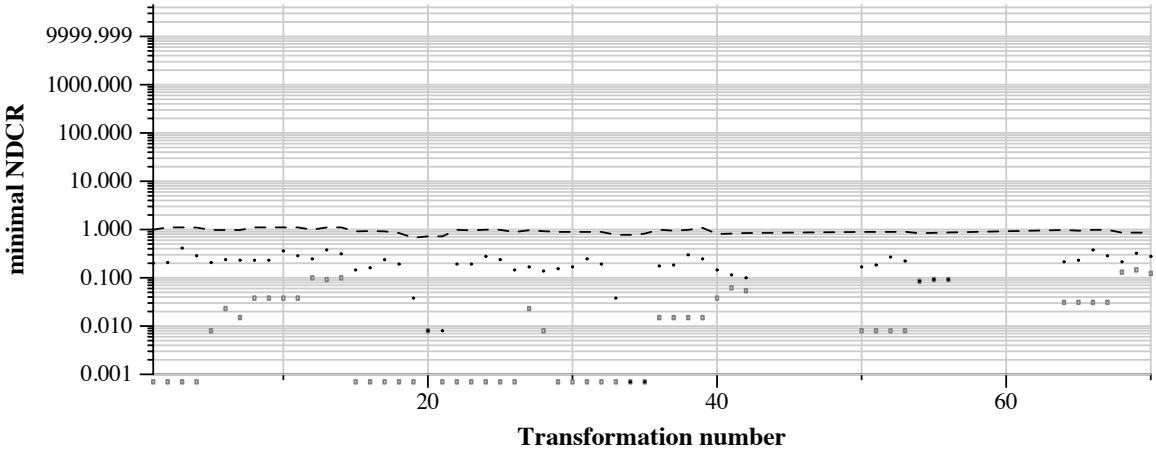
[8] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, October 2003.

[9] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR*, pages 321–330, 2006.

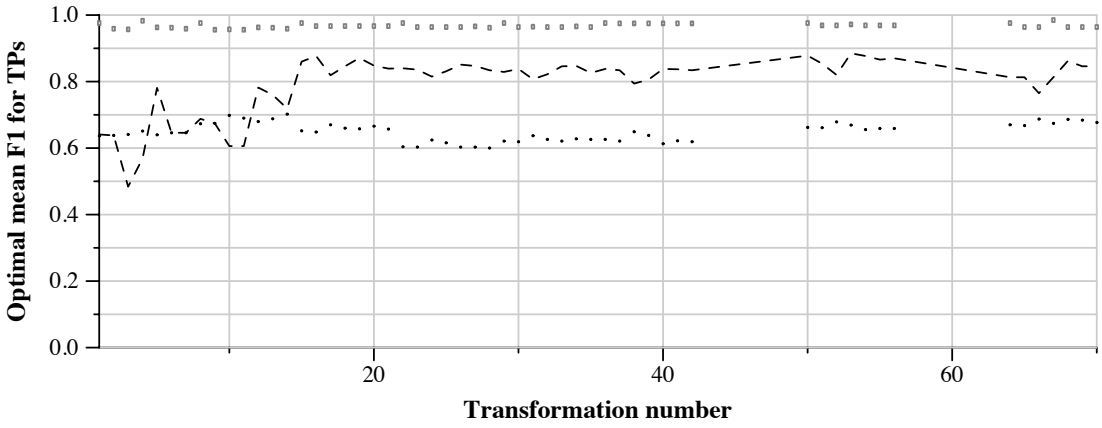
[10] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008.

TRECVID 2010: copy detection results (balanced application profile)

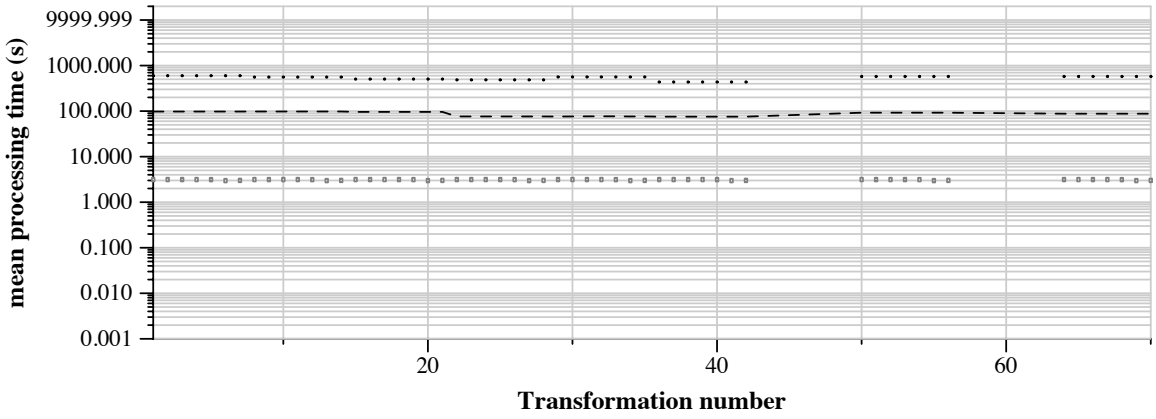
Run name: INRIA-LEAR-TEXMEX.m.balanced.lapin
Run type: audio+video



Run score (dot) versus median (---) versus best (box) by transformation



Run score (dot) versus median (---) versus best (box) by transformation

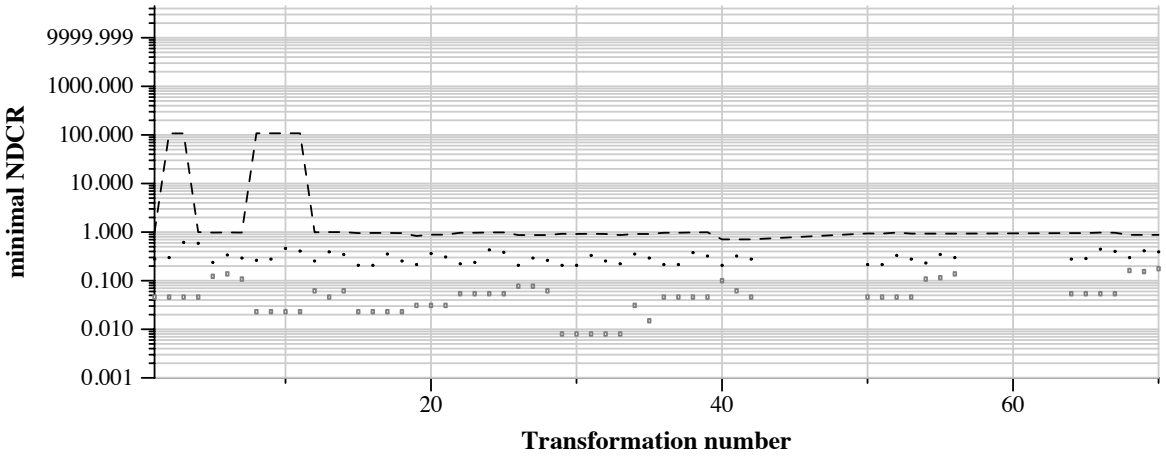


Run score (dot) versus median (---) versus best (box) by transformation

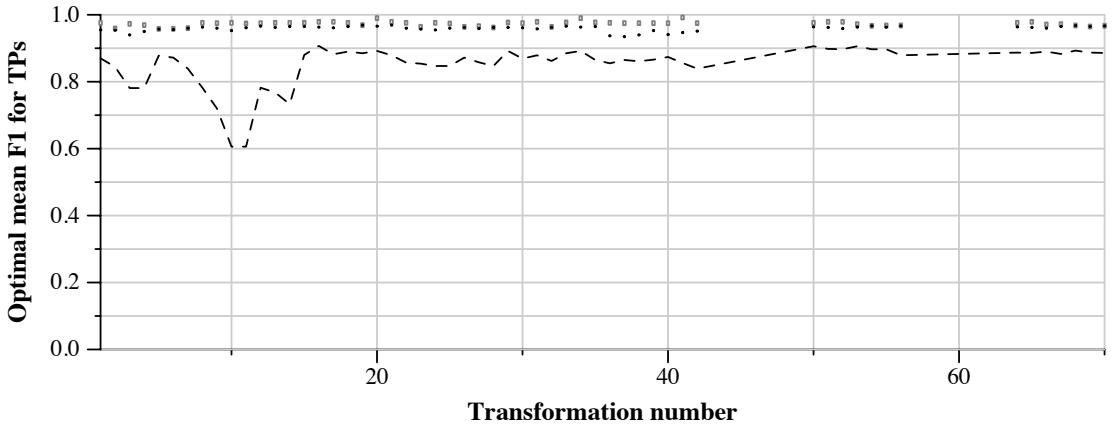
Fig. 5. Our LAPIN run

TRECVID 2010: copy detection results (no false alarms application profile)

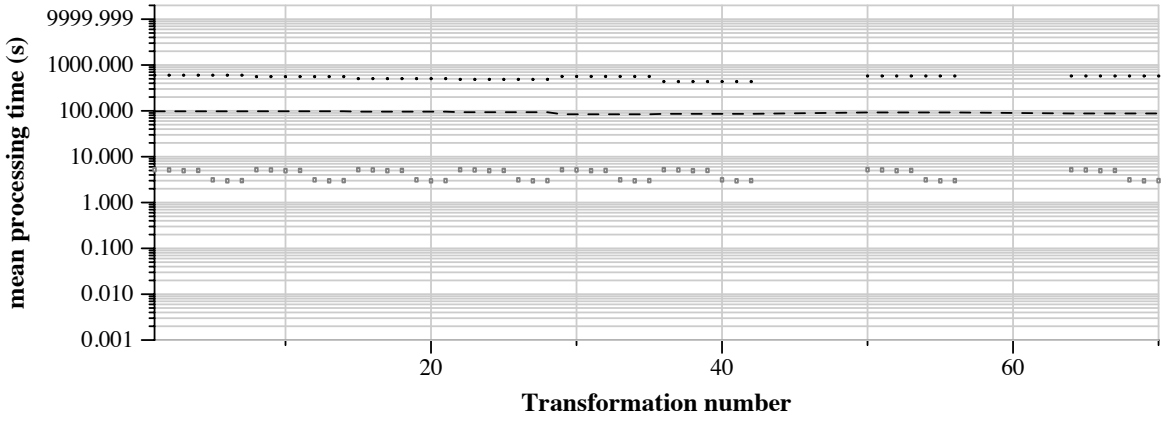
Run name: INRIA-LEAR-TEXMEX.m.nofa.mouflon
Run type: audio+video



Run score (dot) versus median (---) versus best (box) by transformation



Run score (dot) versus median (---) versus best (box) by transformation

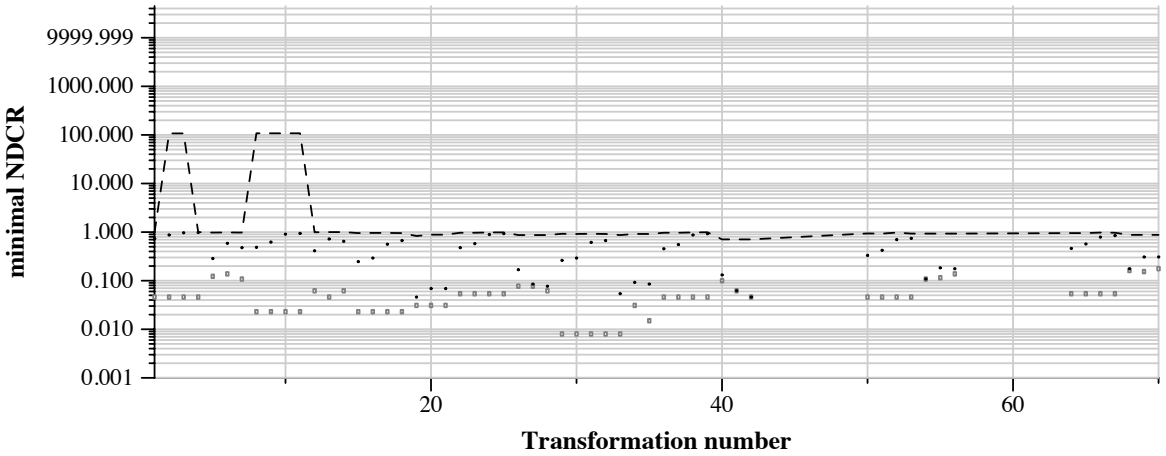


Run score (dot) versus median (---) versus best (box) by transformation

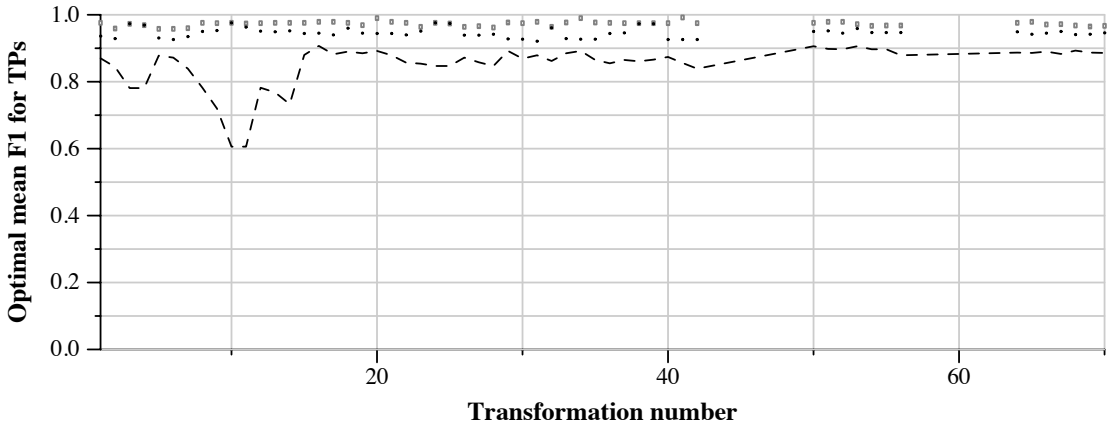
Fig. 6. Our MOUFLON run

TRECVID 2010: copy detection results (no false alarms application profile)

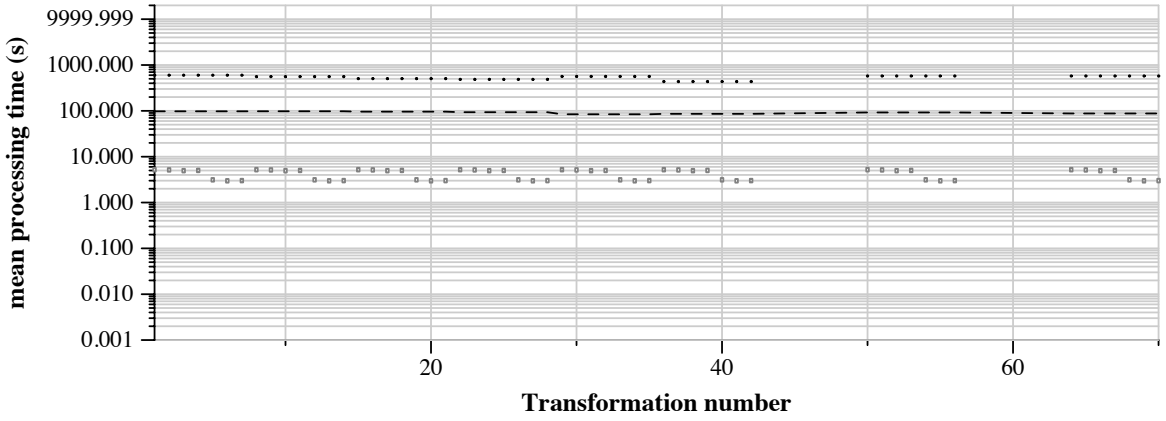
Run name: INRIA-LEAR-TEXMEX.m.nofa.bouquetin
Run type: audio+video



Run score (dot) versus median (---) versus best (box) by transformation



Run score (dot) versus median (---) versus best (box) by transformation



Run score (dot) versus median (---) versus best (box) by transformation

Fig. 7. Our BOUQUETIN run