INRIA@TRECVID'2011: Copy Detection & Multimedia Event Detection

Mohamed Ayari, Jonathan Delhumeau, Matthijs Douze, Hervé Jégou, Danila Potapov, Jérôme Revaud, Cordelia Schmid, Jiangbo Yuan

INRIA@TRECVID'2011: Copy Detection & Multimedia Event Detection

Mohamed Ayari, Jonathan Delhumeau, Matthijs Douze, Hervé Jégou,
Danila Potapov, Jérôme Revaud, Cordelia Schmid, Jiangbo Yuan

INRIA — email: firstname.lastname@inria.fr

***Abstract*— In this paper we present the results of our participation to the Trecvid tasks Copy Detection and Multimedia Event Detection. It focus, in particular, on the comparison of systems for the CCD task, by analyzing the importance of 1) the audio module, 2) the video module and of 3) the fusion module.**

I. Introduction

This notebook paper presents the results of INRIA at TRECVID'2011 for the copy detection (CCD) and multimedia event detection (MED) tasks. It focuses, in particular, on the copy detection task, in which we have obtained very good results.

Our CCD system is an improvement of the one we used 2010 [6]. Therefore we will mainly focus on this year's additions, and on the insight provided by the comparison of our runs, which are given below:

Run	Profile	Visual	Audio	Fusion	Cut
DEAF	balanced	yes	no	N/A	yes
AUDIOONLY	balanced	no	yes	N/A	yes
THEMIS	balanced	yes	yes	late	yes
ZOZO	balanced	yes	yes	late	no
DODO_{bal}	balanced	yes	yes	early	yes
TYCHE	nofa	yes	yes	late	yes
$\text{DODO}_{\text{nofa}}$	nofa	yes	yes	early	yes

Our submitted runs were designed to measure the contribution of the audio and visual content, and to evidence the impact of our early fusion module. Our best runs are those that include all the modalities and this new module, i.e., the DODO_{bal} and $\text{DODO}_{\text{nofa}}$ runs. The validation was done on the TRECVID'2010 set of queries, on which we obtained much better performance with early fusion than with late fusion module, as shown in the experiments of Section V. Our late fusion method is very similar to the one we used in 2010, except that we used the same logistic regression package as for the early fusion.

Although the other runs are suboptimal, the runs based on late fusion obtained better performance than the DODOs runs on a few transformations. In our opinion, this might be an artifact of the NDCR measure, which is very sensitive to the presence of a single false positive.

Comparing our runs on the validation set on our 2011's performance given by NIST leads to the following observations:

1) The performance of the DEAF and AUDIOONLY runs are comparatively poor. Combining the audio and visual modalities is *very* important.
2) Our pure visual system is better than our audio system.
3) The early fusion system improves a lot compared with the late fusion system.
4) The cost of a false positive being very high for the NDCR measure, a common choice, adopted by several participants, consists in returning a maximum of 1 result per query, in order to avoid false positive. Prior to submission, we considered this choice as safer. However, due to the presence of identical videos in the reference dataset, this choice raises the risk of missing a true positive. The "Cut" column in the table above indicates the only run, namely ZOZO, for which we kept more than one result when the first results had nearly identical scores. To our surprise, this run obtained better result than the THEMIS run, which is exactly the same system but keep the best hypothesis (if any).
5) The NDCR measure is strongly dependent on the rank of the first false positive appears, even in the balanced profile.

The MED system we developed is disjoint from the CCD. It mixes 3 modalities: audio, video and image. We submitted runs that combine all or part of them:

Run	Video	Audio	Image
3CHAN:	yes	yes	yes
MBH:	yes	yes	no
NOAUDIO:	yes	no	no
STILL:	no	no	yes

Overall we found that, to classify the events, the video (motion) descriptors were most useful. Other modalities do not necessarily improve the results.

The paper is organized as follows. Sections II and III describe our audio and visual matching systems for the CCD task, respectively. The new early CCD fusion system is detailed in Section IV, and our results are analyzed in Section V. Finally, we give a brief overview of our MED system in Section VI.

II. CCD – Audio matching: Babaz

This section describes the main components of BABAZ, which is a audio search system specifically designed for a copy detection setup such as the one considered in TRECVID [16], where the signal is the audio track of a video, i.e., that typically includes voices, silences and occasionally music. The copied audio tracks are transformed by different kinds of transformations, such as strong pass-band filter, compression, mixing, single- or multi-band companding, etc.

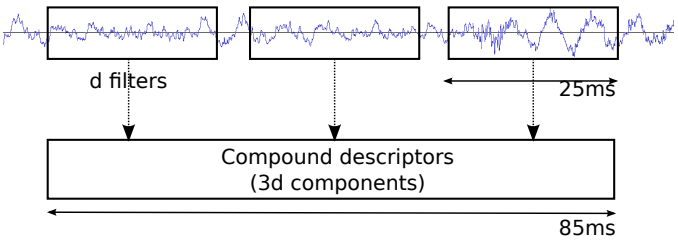

Fig. 1
COMPOUND DESCRIPTOR.

A. Pre-processing

The audio tracks extracted from an audio corpus are not necessarily homogeneous. Sample rates as well as encoding quality vary significantly from one track to another. It is in particular the case in the Internet Archive dataset used in the TRECVID's copy detection task, where the videos are mainly amateur videos captured and encoded by different devices and audio codecs. In order to deal with this variability in a consistent way, all the tracks are resampled to 32,000 Hz. We use the right stereo channel only when stereo is available.

B. Feature extraction: filter banks

Hereafter, we detail how we extract descriptors from the audio waveform. 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 overlapping band-pass filters. We use 40 filters spread along the [500 Hz,3000 Hz] frequency range on a Mel scale. As a result, the dimensionality of the descriptors is $d = 40$.

The representation based on these filters gives a rough approximation of the signal's spectral shape in the frequency range considered while smoothing out the harmonic structure, if any, and is therefore robust to many spectral distortions. We have used the freely available `spro` software[1] for the generation of filter banks. This software also includes an efficient implementation of the widely used MFCC descriptors. However, in our experiments, these descriptors are significantly outperformed by the filter banks.

C. Compound descriptors and energy invariance

The temporal consistency provided by a single filter bank is limited, as their temporal span is limited and only frequencies are considered. This is problematic since the database is large: the filter banks themselves might not be 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, as done in Serra's thesis [15] in a context of cover detection.

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

For a given timestamp t, 3 successive filter banks are extracted at timestamps $t - \delta$, t and $t + \delta$, producing a compound descriptor of dimensionality $3d$ (i.e., 120). We set $\delta = 30$ ms in order to avoid overlapping. We have performed a few experiments on a validation dataset 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 in Figure 1, the resulting span of this descriptors is 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.

Descriptors are compared with the Euclidean distance. For large vector databases it allows for efficient indexing algorithms. In order to take into account attacks on the volume (signal energy), the descriptor is finally made invariant by subtracting its mean.

D. Approximate nearest neighbor search

As the exact search is not efficient enough, BABAZ uses an approximate nearest neighbor search technique. Many methods exist for this task, such as the popular locality sensitive hashing [2] search algorithms and the FLANN package [12]. However, this step has a major impact on both efficiency and search quality, and only a few methods are able to search in hundreds of millions of descriptors with reasonable quality, as required by our method to index thousands of hours of audio.

BABAZ uses the IVFADC indexing method of [9], which is able to index billions of descriptors on a commodity server. It finds the approximate k nearest neighbors using a compression-based approach, and relies on an inverted structure to avoid exhaustive search. This approximate nearest neighbor method implicitly sees multi-dimensional indexing as a vector approximation problem. It is proved [9] that the square error between the distance and its estimation is bounded, on average, by the quantization error. This ensures, asymptotically, near perfect search results when increasing the number of bits allocated for the quantization indexes:

The main parameters of this method are the number of bytes b used per database vector and the number c of inverted lists associated with the partitioning of the feature space (learned by k-means). In our case, we set $b = 24$ bytes and use multiple assignment [9] on query side, leading to visit 16 inverted lists out of $c = 16,384$.

E. Scoring vote and reciprocal nearest neighbors

The search technique returns a list of k (approximate) nearest neighbors. A conventional method to exploit them consists in assigning a vote of 1 to all the corresponding audio tracks, or alternatively, a function of the rank or of the distance. Based on a recent state-of-the-art work [8] in image search, we adopt a different strategy, which is illustrated in Figure 2.

Denoting by $d_k(q)$ the distance between the query descriptor and its k-th nearest neighbor, the quantity $d_k(q) - d(q, i)$ is shown, based on a mutual information criterion [8] measured on image descriptors, to better reflect the quality of the match.

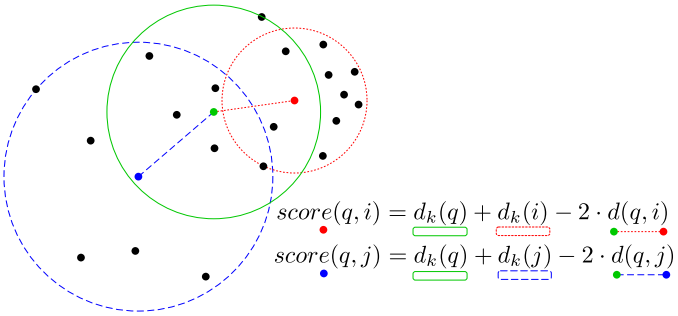

Fig. 2

RECIPROCAL NEAREST NEIGHBORS AND OF OUR VOTING STRATEGY.

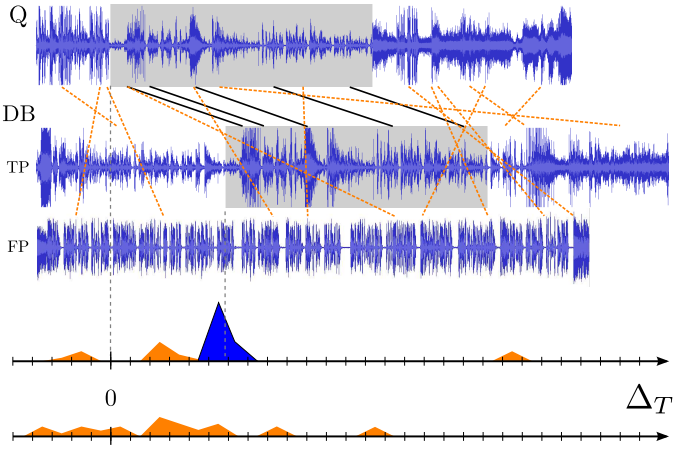

Fig. 3

ILLUSTRATION OF THE TEMPORAL HOUGH TRANSFORM: THE AUDIO MATCHES OUTPUT BY THE APPROXIMATE SEARCH ENGINE ARE COLLECTED AND SUMMED UP FOR EACH HYPOTHESIS (id, Δ_T). THIS DILUTES THE SCORES OF FALSE POSITIVES OVER TIME SHIFT HYPOTHESES.

This is also the case for our audio descriptors, so we adopt this weighting scheme.

The distance $d_k(q)$ is relative to the query. In order to symmetrize the relationship between the query and database descriptors, it is worth considering the *reciprocal* nearest neighbors of the database vector, or more specifically the typical distance between the database vector and its own k-nearest neighbors.

In practice, computing the reciprocal nearest neighbors is impractical: the audio descriptor database may contains up to billions of vectors. If exact nearest neighbor search is used, then it turns out that each database vector has to be submitted to the system. Although some approximate strategies [3] were proposed to compute the nearest neighbor graph, these approaches were only tested on up to 1 million vectors. However, we are not interested in the neighbors themselves, but in the typical distance of a database vector to its neighborhood. This reciprocal typical distance is estimated on a limited subset of 1 million vectors. In this case, the parameter k associated with the database vectors has to be adjusted to account for the smaller size of this subset.

F. Re-ranking

Finally, in the spirit of [10], the hypotheses are re-ranked based on exact descriptors to obtain the exact distances, in order to increase the precision of the proposed similarity. The difference with [10] is that we use the original descriptors and not only a compressed version of these.

G. Energy weighting

Video tracks contain many silences. Those are filtered when the signal and consequently the descriptor is zero. However, there are also many descriptors extracted on audio frames containing almost no energy, but which are not pure silence. Filtering audio segments with low energy may lead to loose some precious information, and reduce the accuracy of the localization. For this reason, we adopt a smoother strategy and multiply the score associated with the match with the energy of the query descriptor.

H. Hough matching

BABAZ assumes that the transformations do not include any acceleration. Given a query, for each of its audio descriptors we first search for the k approximate nearest neighbors and compute their weighting score based on the strategy exposed above. We then vote for several alignment hypotheses (a_b, Δ_T) using the scoring method introduced above. Compared with uniform voting, this brings a slight improvement at almost no cost in efficiency. The video hypotheses with low scores are filtered. On output, this Hough matching system returns a maximum of 40 hypotheses per query. Each database track is associated with a maximum of 3 Δ_T hypotheses.

I. Detection of boundaries

At this point, for each query we may have several alignment hypotheses (id, Δ_T), where id is the database track identifier and Δ_T is the difference between the query and the database timestamps. We use the whole set of descriptors and weight their distance to produce a score per time instant. This score is filtered in time and used to detect the boundaries defining a matching segment. Each segment receives a score computed from the individual scores for each time instant.

J. Shifted query

The audio descriptors are extracted every 10 ms, which leads to reduce the quality of the comparison if the sampling of the database track occurs in phase opposition, i.e., with a shift of 5 ms relative to the query track. To address this problem, we submit several shifted version of the query to the system. For instance, we create shifted versions of the query with shifts of 2, 4, 6 and 8 ms. This, obviously, significantly impacts the efficiency of the search by a significant factor, and should be used when high precision is required only.

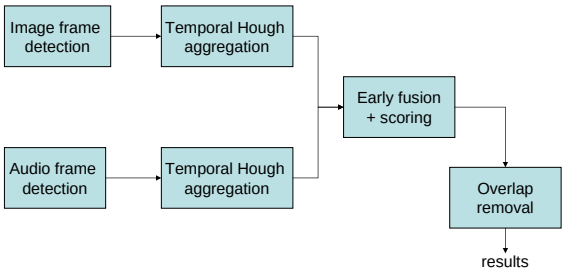

Fig. 4
SUMMARY OF OUR BEST APPROACH FOR CONTENT-BASED COPY DETECTION.

III. CCD – VISUAL MATCHING

We briefly describe our video indexing system, which did not change with respect to Trecvid 2010. For more detail, see [4] and [6].

A. Frame matching

Every 10th frame is extracted from the videos to analyze. The images are described with a Hessian-Laplace detector, followed by a CS-LBP descriptor [5].

Descriptors are quantized to a visual vocabulary of 200,000 words, and binary signatures of 48 bit [7] are computed on each point. The signatures are indexed in an inverted file system.

At query time, frames from the query video are analyzed similarly and matched against the inverted file. Database images that have most matches in common with the query are retained for the next stage.

B. Temporal aggregation

Alignment hypotheses between the query and database video sequence are generated with a 1D Hough transform (similar to the audio system, see II-H).

Hypothetic video sub-sequence matches are constructed from the frame matches used for each alignment estimated by the Hough transform.

IV. CCD – COMBINATION OF CHANNELS

In the following, we call a set of temporally consistent audio or video frames an *hypothesis*. As first step of the fusion process, low level audio and image frame matches consistent with the hypothesis parameters are gathered and precisely aligned using a robust time warping procedure (section IV-A). Then, different features are extracted based on the time warping result to describe various aspects of the hypothesis (IV-B). Finally, a classifier (IV-C) estimates the probability that the hypothesis is correct or not based on these features.

We comment the results of the independent channels, the fusion strategy (V) and how the fusion improves over a classical late fusion (IV-D).

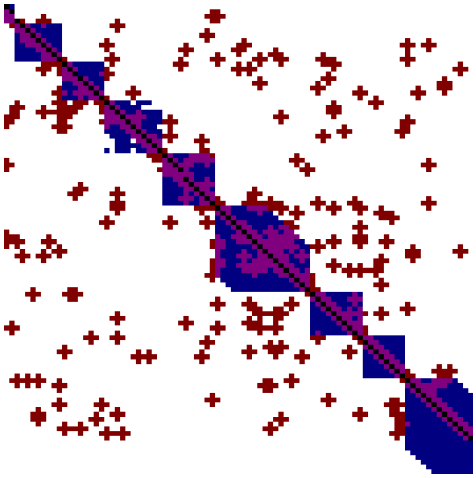

Fig. 5
EXAMPLE OF A COMPATIBILITY MATRIX BETWEEN A QUERY AND ITS GROUND-TRUTH COPY IN THE DATABASE. COLUMNS (RESP. ROWS) CORRESPONDS TO QUERY FRAME (RESP. RETRIEVED FRAMES) ORDERED BY TIME AND COLORED PIXELS INDICATE THE PRESENCE OF A LOCAL MATCH BETWEEN AUDIO DESCRIPTORS (RED PIXELS), IMAGE DESCRIPTORS (BLUE PIXELS) OR BOTH (PURPLE PIXELS). BLACK PIXELS SHOW THE OPTIMAL TIME WARPING PATH.

A. Robust time warping

1) The compatibility matrix: Given the time ranges on both query and database side, audio and image frame matches are gathered in order to build a compatibility matrix. In this matrix, each cell (u, v) describes the similarity between the query in the time range $[u.\Delta_q^I, (u+1).\Delta_q^I]$ and the database in the time range $[v.\Delta_{db}^I, (v+1).\Delta_{db}^I]$, where Δ_q^I and Δ_{db}^I are the duration of a query and a database image frame, respectively. An example of such matrix filled with image and audio frame matches is presented in Figure 5.

Because audio and image frame durations are not the same, an additional step is introduced to combine match scores: each audio frame matches is attached to the closest image frame match - the duration of an image frame is typically 400 ms, whereas audio frames only last 10 ms, so up to $(400/10)^2$ audio frame matches can be allocated to a single matrix cell. Furthermore, a geometric verification is processed beforehand on all image frame matches: an affine transform between the query and the database is estimated for the whole hypothesis (refer to [4] for more details), and frame matches not compliant with the affine transform are eliminated.

2) Cell-level similarity: Finally, a classifier computes a similarity score in each matrix cell, taking into account both audio and image matches in the cell. The classifier is a logistic regression, that takes the following form:

$$h(\mathbf{x}) = \frac{1}{1 + \exp(-\beta^\top \mathbf{x})},$$

where $\mathbf{x}$ is a feature vector describing both audio and image cell content (see below), and β is the vector of regression coefficients. The β is learned from a set of correct and incorrect matches using the standard maximum likelihood

estimates with iteratively re-weighted least squares. We use queries from Trecvid 2010 as a learning set.

The features $\mathbf{x} = [x_1 \dots x_4]^\top$ on which the classifier bases its decision are:

- The maximum cumulated score of consistent audio frame matches in the cell:

$$x_1 = \max_{\delta t} \sum_{a \in A / t_q(a) - t_{db}(a) = \delta t} \text{score}(a)$$

  where A the set of audio frame matches attached to the cell and $t_q(.)$, $t_{db}(.)$ yields the timestamp of audio match a on the query and database sides.
- The number x_2 of image key-point matches compliant with the estimated affine transform (ie. inliers); as well as two normalized versions x_3, x_4 of this number: the normalization factors are the total number of key-point in the query frame and the database frame.

3) Time warping: A robust dynamic time warping procedure finds the optimal path in the compatibility matrix. The proposed algorithm is robust to holes in the path thanks to a modification of the original algorithm: instead of computing the dynamic time warping using the standard formula $\text{score}(u,v) + \max(\text{DTW}(u-1,v), \text{DTW}(u,v-1), \text{DTW}(u-1,v-1))$, we define:

$$\text{DTW}(u,v) = \text{score}(u,v) + \max_{x<u, y<v} \text{DTW}(x,y). \quad (1)$$

The optimal path is retrieved accordingly by iterating on

$$\text{prev\_cell}(u,v) = \arg \max_{x<u, y<v} \text{DTW}(x,y), \quad (2)$$

where (u,v) starts at the bottom-right corner of the DTW matrix. The result is essentially different of the standard time warping for two reasons: (1) it allows jumping between non-contiguous cells; and (2) it enforces a one-to-one assignment constraint between query and database frames, because of the strict inequalities in eq. (2). The one-to-one constraint is important as some features used later to describe each hypothesis are the sum of matching image or audio frame scores, which can be over-estimated for still videos where each frame resembles all other frames.

In the example in Figure 5, the optimal path found by our algorithm is represented with black pixels. We call the resulting assignment *matching frames* in the following.

B. Hypothesis description

The next step involves the ranking of the considered hypotheses. For this task, features are extracted from the result of the dynamic time warping and are then once again fed into a logistic classifier. The extracted features can be distributed in three categories: features concerning image only, features concerning audio only and features concerning both modalities.

All features are simple and cheap to compute, hence not changing the computational cost of the proposed approach with respect to a baseline approach without classifier.

1) Image features: We extract several features to describe the quality of the matching image frames from a global viewpoint:

- the number of matching image frames;
- the cumulated score of matching image frames;
- the image match density, which is the average proportion of matching image frames per second;
- the maximum time lapse between two consecutive matching image frames (i.e. size of holes), on the database side. We experimentally observed incorrect hypotheses have less regularly spread frame matches;
- the ratio of matching image frames on the total number of image frame matches in the compatibility matrix;
- the fact that the query is flipped or not (the features has value in $\{0,1\}$);
- the plausibility of the geometric transform: each of the four coefficients specifying the affine transform (σ, r, α and α_2, see [4]) is assumed to follow a normal distribution $\mathcal{N}(0,\Sigma)$, and corresponding a-posteriori probabilities are extracted for each coefficient for various Σ.

Additional features are generated as normalized versions of the first two features above: the normalization factors include the number of image frames in the hypothesis time range, the total number of matching image frames in the compatibility matrix and the hypothesis' time length. Similar normalizations are also applied to equivalent features in the audio and multimodal descriptions.

Then, some more elaborate indicators based on the number and the spatial distribution of matched key-points are computed. In this perspective, we merge the points from the query frames into a single virtual query image, and similarly on the database side. The extracted features are:

- the matched area: we use a low-resolution accumulator (15×10 pixels) to measure what area of the database virtual frame is actually matched to the query frame. To that aim, the accumulator is initially set to zero. Then, each matched patch is "printed" in the accumulator image. We map pixel values with $x \mapsto 1 - s^{-x}, s > 1$ to mitigate the influence of high values. The final score is the sum over the accumulator pixels. We extract several features by varying the s;
- the Kullback-Lieber (KL) divergence between the spatial distributions P and Q of the matched key-points and all available key-points (in the database frame). The KL-divergence is calculated as

$$D_{KL}(P||Q) = \sum P(i) \log \frac{P(i)}{Q(i)}.$$

  As with the accumulator, we used a 15×10 map to quantify the distribution;
- every bin of the spatial distribution of the matching key-points P is used as features as well (based on a coarser 5×4 map). This is useful, as some parts in the image are more often incorrectly matched, for instance the top-right corner where the channel's logo is printed.

2) Audio features: Similarly, we compute several indicators to describe the global quality of the matching audio frames:

TABLE I

MOST EFFICIENT FEATURES IN THE ORDER OF SELECTION OF OUR BOOSTING-LIKE TRAINING PROCEDURE.

rank	feature name	transformation
1	matched area, $s = 1.5$	x
2	number of matching frames	$1/(1+x)$
3	KL divergence	x
4	matched area, $s = 4.5$	$\log(1+x)$
5	hypothesis length in s	x
6	hypothesis length in s	$1/(1+x)$
7	number of matching image frames	$1/(1+x)$
8	top-left corner of P distribution	$\sqrt{x}$
9	is query flipped?	x
10	ratio of matching image frames	x
11	max hole size between image frames	$1/(1+x)$

- the total number of matching audio frames;
- the cumlated score of matching audio frames;
- the audio match density, which is the average proportion of matching audio frames per seconds;
- the maximum time lapse between two consecutive matching audio frames (i.e. size of holes), on the database side;
- the ratio of matching audio frames on the total number of audio frame matches in the compatibility matrix.

3) multimodal features: Finally, a few multimodal indicators are also computed, in which audio and image matching frames are considered indifferently of their origin:

- the number of matching frames;
- the cumulated score of matching frames;
- the hypothesis length, measured between the first and last frame match;
- the maximum time lapse between two consecutive matching frames;
- the ratio of matching frames on the total number of frame matches in the compatibility frames.

C. Logisitic classifier and feature selection

As one can see, many features are available for the training (all in all, hundreds features are available, we omitted some of them for space reasons). Because of this overwhleming number, the training of the logistic classifier does not converge easily, and the resulting classifier yields poor results.

We propose is to use a greedy boosting-like procedure to select a small subset of all features. The algorithm proceeds as follows: at each iteration, it trains a classifier using all the already selected features and one additional feature, trying all non-selected features in turn. It keeps the feature maximizing the Average-Precision (AP) on a validation set. We iterate this until the AP does not increase more than a small fixed value.

We also incorporate in the feature pool transformations of the original feature values through 3 non-linear functions: $\log(1+x)$, $\sqrt{x}$ and $1/(1+x)$. We discovered that adding those artificially generated features improves the classifier performance up to 2% of AP.

The first features selected by our training procedure (i.e. the most efficient ones) are presented in Table I along with the associated transformation. We trained on the entire TRECvid 2010 query set, which represents about 40,000 positive hypotheses and more than 200,000 negative hypotheses. Overall, the training of the classifier takes a few hours on a standard computer. Note that we also tried using AdaBoost and a linear SVM, but the training was much slower without enhancing the result.

D. Late fusion

The easiest method to merge results is to combine the output scores of the audio and video systems. As a baseline for the early fusion, we implemented such a system.

In this late fusion approach, The image and audio systems are run independently and their hypotheses are merged to obtain the overall set of hypothesis and corresponding scores. No further analysis is done on the media themselves. In order to compare and fuse the scores from audio and visual systems, their distribution is required for false positives and true positives. Adjusting these scores on a common basis is performed by using a logistic regression on both inputs, which produces consistent output. This logistic regression is learned on the TRECVID'2010 validation set.

On audio, the logistic regression takes into account

- The log-score initially produced by the audio system;
- The log-scores of the best three hypotheses;
- The length of the matching segment;
- The length of the query and database tracks.

Taking into account the scores of the other hypotheses is motivated by the fact that the dynamics of the scores may vary significantly from one query to another.

At this point both systems have a set of hypothesis with similarly distributed scores. Some of those hypotheses are shared by both modalities and others are present in only one of them.

Intuitively, given the size of the database, a hypothesis shared between image and audio is likely to be true positive regardless of the scores: logistic regression is not appropriate for this last fusion.

Of the true positives found by only one media, a significant number of them were found by the image system only, whereas very few were found by the audio system only. Our overall strategy was therefore to use the video score as a baseline, adding the audio score and a significant bonus to the hypothesis shared with audio if either score was already positive, and ignoring the hypotheses found by the audio system only.

V. CCD – THE RESULTS

In this section, we first show how our system improved over the 2010 version. Then we compare the results with those of other participants.

A. Comparison with 2010

Figure 7 shows the precision-recall curve of our system, compared with the 2010 version (with corrected audio, see [6]) both are tested on the 2010 queries.

One can see that the precision is improved a lot due to the improved weightings (both in audio matching and the fusion

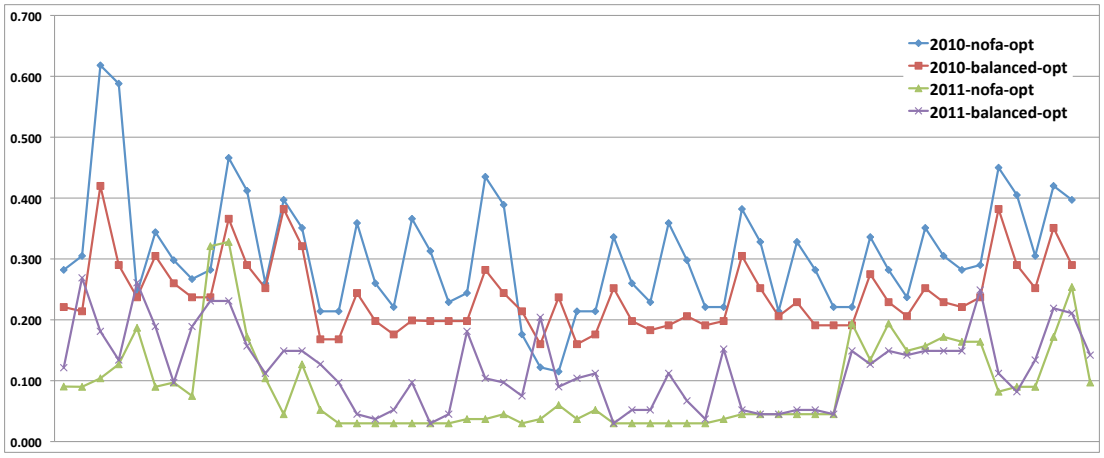

Fig. 6

CCD – COMPARISON OF THE SEARCH ACCURACY (NDCR) IN 2010 AND 2011 (RESULTS GENERATED BY NIST).

A \ V	1	2	3	4	5	6	8	10
1	1	5	3	1	4	1	3	2
2	1	4	3	1	2	1	4	1
3	2	2	3	2	2	1	4	1
4	2	2	3	1	2	1	3	1
5	1	1	3	1	3	1	4	2
6	1	2	3	3	4	2	4	4
7	1	1	2	1	4	1	4	1

Video transformations
1: camcording
2: insertion of orginal video in front of background
3: pattern insertion
4: compression
5: change in gamma
6: decrease in quality
8: post production
10: random transform

Audio transformations
1: none
2: mp3 compression
3: mp3 compression + companding
4: limit bandwidth + single-band companding
5: mix with speech
6: mix with speech + multiband compress
7: bandpass + mix with speech + compress

Overall best NDCR: 0.00 (easy) — 0.15 (hard)

TABLE II

RANK OF OUR BEST RUN, IN THE NOFA PROFILE, PER TRANSFORMATION. THE BACKGROUND COLORS OF THE CELLS GIVE AN INDICATION ON HOW HARD THE TRANSFORMATION IS: IT ENCODES THE BEST PERFORMANCE (NDCR) OBTAINED OVER ALL PARTICIPANTS.

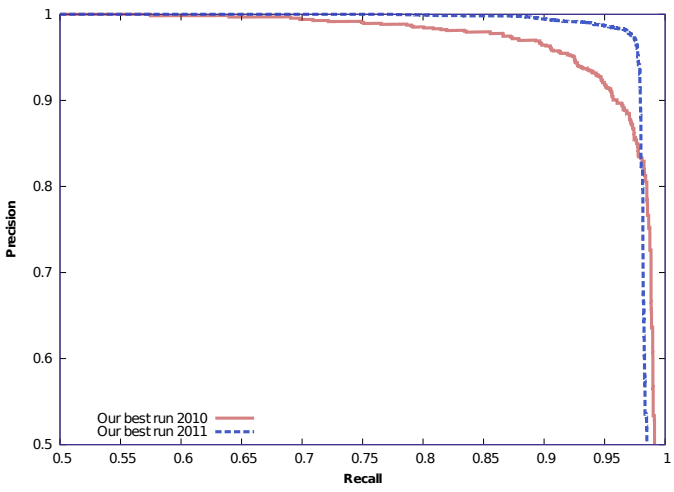

Fig. 7

COMPARISON OF THE PERFORMANCE OF OUR 2010'S WITH OUR NEW SYSTEM FOR 2011: PRECISION-RECALL.

module) introduced this year. This comes at a slight cost in recall.

Between 2010 and 2011, the reference videos did not change. There are the same number of queries and they are generated exactly in the same way. Therefore, it is significant to compare the NDCR measures. Figure 6 shows the comparison on our runs, which is clearly in favor of the new system.

Especially the NOFA profile improved a lot. This is consistent with the improvement of precision. In contrast, the balanced profile did not improve as much. When comparing with other participants, our runs in this profile are clearly weaker.

B. Comparison of variants

The table below provides a comparison of the runs we submitted for the balanced profile to evaluate the relative importance of the components of our system (audio and video core systems, late/fusion module). The performance is compared using the mean and median of opt-NDCR over all transformations.

Run	channels	Fusion	Cut	mean	median
DEAF	v	N/A	yes	0.258	0.209
AUDIOONLY	a	N/A	yes	0.406	0.425
THEMIS	a+v	late	yes	0.211	0.219
ZOZO	a+v	late	no	0.194	0.200
DODO_{bal}	a+v	early	yes	0.144	0.134

As observed in this table, the individual performance of the DEAF and AUDIOONLY runs are poor: combining the audio and visual modalities is important. The better performance of the ZOZO run over the THEMIS run demonstrates that keeping more than 1 result is interesting if these results have similar yet high scores. Finally, as validated on 2010 set, our new early fusion module brings a fair improvement over the late fusion module.

C. Comparison with other participants

Table II shows how our runs perform with respect to others. Relative to the other participants, our system is especially efficient on difficult attacks like camcording (1) or image based transforms (4, 6, 10). On easy ones like changing gamma or inserting patterns (3, 4), the system is relatively less efficient. This is probably because it is tuned to be very invariant to strong attacks. This comes at a price in terms of discriminative power (and computing cost...).

VI. MULTIMEDIA EVENT DETECTION

The goal of multimedia event detection is to classify video clips into event categories, such as birthday party, getting a vehicle unstuck and grooming an animal. Our approach combines classifiers for three modalities—video, audio and still images. The individual classifiers are described in sections VI-A, VI-B and VI-C. The combination of classifiers is presented in section VI-D. Results are discussed in section VI-E.

A. Video Features

The motion information in the video clips is described with dense trajectories [17], which have shown to obtain state-of-the-art results for video classification. The method extracts dense trajectories by sampling points densely in each frame and tracking them with a dense optical flow field. Trajectories are described with motion boundary histogram descriptors (MBH) [17], [1]. MBH descriptors encode the relative motion between pixels and are robust to camera motion. Derivatives are computed separately for the horizontal and vertical components of the optical flow and are quantized in a histogram.

Trajectories and their description are computed with an on-line available code [2]. We use the following parameters for trajectory extraction: 8 spatial scales, a spatial sampling stride of 8 pixels, a trajectory length of 20 frames and a dense point re-sampling every 5th frame. The MBH descriptor is of dimension 192, i.e., MBHx and MBHy components are represented by 96 dimensions each corresponding to a 2x2x3 spatio-temporal grid with a 8 bin histogram in each cell.

A video clip is represented by a bag-of-features (BOF). To construct the codebook we randomly select two million MBH descriptors from 2000 training videos and compute 4000 clusters ("visual words") with k-means. A MBH descriptor is assigned to the closest cluster center using Euclidean distance. The BOF represents the frequency per visual word and is normalized with the L1 norm. Video classification is performed with a non-linear χ^2 kernel. We train a one versus all support vector machine (SVM) classifier for each event. The parameter γ is set to the average distance between training examples and the parameter C is computed using 5-folds cross-validation.

We train each classifier with a subset of the training videos, i.e., approx. 200 positives and 6500 negatives for each event. The negative video clips correspond to roughly 2000 videos from the class NULL and 4500 clips form the other event classes.

B. Audio Features

The audio signal is described with Mel-frequency cepstral coefficients (MFCC) [14], which are widely used in speech recognition and music genre classification. For our system we use an on-line available code [3]. We apply a stereo-to-mono transformation to our audio signals by averaging the left and right stereo channels. We compute 32ms time-window MFCC descriptors of size 30 with 50% of window time overlap.

Audio features are also quantized with a bag-of-features representation. We extract 4000 clusters with k-means from 500k audio signals. To obtain the audio classifiers, we train an SVM classifier with a non-linear χ^2 kernel. The training set consists of the audio signals of the videos used to train the video classifier. We use a one versus all SVM, the parameter γ is set to the average distance between training examples and the parameter C is computed using 5-folds cross-validation.

C. Image Features

For the image classifier, we extract image features for every 10th frame of a video. For each image we extract SIFT descriptors [11] on a dense grid at 5 scales with horizontal and vertical steps of 4 pixels. The dimension of the descriptors is reduced using PCA from 128 to 64 dimensions. The descriptors of an image are, then, aggregated into a Fisher vector [13]. Here, we use a Fisher vector based on a Gaussian mixture model with 64 Gaussians—shown to be a good trade-off between computational efficiency and classification performance. A linear one versus all SVM classifier is, then, trained on the Fisher vectors. We use a subset of 1000 positive and 5000 negative frames for training each event classifier. The positive frames are obtained from approx. 100 videos and the negatives from 5000 videos. The C parameter is selected using 5-fold cross-validation (separately for each event category). We ensure that frames from a video are in the same fold.

To assign a label to a video clip, we score every 10th frame for a given event and, then, use the maximum frame scores as a confidence value for a video clip and event class.

[2] http://lear.inrialpes.fr/people/wang/dense_trajectories

[3] http://labrosa.ee.columbia.edu/matlab/rastamat/

Event	min NDC				rank
	(1) v + a + i	(2) v + a	(3) v	(4) i	
E006 birthday party	0.8702	**0.7165**	0.7533	1.0004	8
E007 changing a vehicle tire	0.7825	0.7500	**0.7323**	0.9240	8
E008 flash mob gathering	**0.4332**	0.4337	0.4809	0.5858	7
E009 getting a vehicle unstuck	0.6366	**0.5650**	0.6077	0.9056	8
E010 grooming an animal	0.8058	0.8101	**0.8039**	0.9911	8
E011 making a sandwich	0.9154	0.8892	**0.8572**	0.9317	10
E012 parade	0.5948	**0.5514**	0.5950	0.9263	6
E013 parkour	0.4489	**0.4475**	0.4789	0.9033	6
E014 repairing an appliance	0.6711	0.5099	**0.5094**	0.8396	8
E015 working on a sewing project	0.8304	**0.8027**	0.8105	0.9795	10

TABLE III

RESULTS FOR MED: NORMALIZED DETECTION COST (LOWER IS BETTER) AND RANK OF OUR BEST RUN WITH RESP. TO PARTICIPANTS (A TOTAL OF 19).

D. Fusion of multi-modal classifiers

The classifiers are combined with a weighted sum of the probabilities obtained by applying a sigmoid function to the scores. The combination of the video, audio and still image modalities is given by:

$$P(v_i/e_k) = \alpha P_{vid}(v_i/e_k) + \beta P_{aud}(v_i/e_k) + \gamma P_{im}(v_i/e_k) \quad (3)$$

where v_i is a video, e_k is an event and $\alpha, \beta, \gamma \in \mathbb{R}$ are the weights. These weights are learned on a validation set of 2400 videos to maximize the classification performance evaluated by the mean Average Precision (mAP). For our dataset we obtained $\alpha = 0.63, \beta = 0.27, \gamma = 0.1$. We can observe that most weight is given to the motion information. If only motion and audio information is used, we apply the same ratio between video and audio modalities, i.e., $\alpha = 0.7, \beta = 0.3$. In case of a video with no audio signal or with a weak energy one, the final probability ignores it and rebalances the weights. The submitted threshold was estimated on the validation set to produce an operating point with a 1:12.5 false alarm to miss ratio.

E. Analysis of results

We submitted four runs to the official evaluation. They correspond to different combinations of video, audio and still image modalities:

1) 3CHAN: a combination of video, audio and still image
2) MBH: a combination of video and audio
3) NOAUDIO: video only
4) STILL: still image only

Table III summarizes the results. Overall our runs rank about 8th out of 19 participants. We can observe that the combination of video and audio features performs best. The addition of audio improves the performance over video alone six times out of ten. It does not help or slightly degrade the results if the audio information is not significant for the event. Still images do not add significant information. This is probably due to our relatively weak still image classifier. It is clearly not optimal, as there are arbitrary frames in the positive video examples. Future work will include the selection of keyframes when training the classifier.

VII. ACKNOWLEDGMENTS

This work was partially funded by the QUAERO project supported by OSEO and the European integrated project AXES.

REFERENCES

[1] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2006.
[2] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Symposium on Computational Geometry*, pages 253–262, 2004.
[3] W. Dong, M. Charikar, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW: Proceeding of the International Conference on World Wide Web*, 2010. to appear.
[4] M. Douze, H. Jégou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. on Multimedia*, 12(4):257–266, jun 2010.
[5] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, 2009.
[6] H. Jégou, M. Douze, G. Gravier, C. Schmid, and P. Gros. INRIA LEAR-TEXMEX: Video copy detection task. In *Proc. of the TRECVid 2010 Workshop*, Gaithersburg, United States, 2010.
[7] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, February 2010.
[8] H. Jégou, M. Douze, and C. Schmid. Exploiting descriptor distances for precise image search. Research report, INRIA, June 2011.
[9] 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.
[10] H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg. Searching in one billion vectors: re-rank with source coding. In *International Conference on Acoustics, Speech, and Signal Processing*, Prague Czech Republic, 2011.
[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
[12] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, February 2009.
[13] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.
[14] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. PTR Prentice Hall, 1993.
[15] J. Serrà. *Identification of versions of the same musical composition by processing audio descriptions*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2011.
[16] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, 2006.
[17] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2011.