Telefonica Research at TRECVID 2010 Content-Based Copy Detection

Ehsan Younessian[1], Xavier Anguera[2] Tomasz Adamek[2] and Nuria Oliver[2]
[1]School of Computer Engineering,
Nanyang Technology Univ., Singapore, Singapore
[2]Telefonica Research,
Via Augusta 177, 08027, Barcelona, Spain
[1]ehsa0001@e.ntu.edu.sg,{xanguera, tomasz}@tid.es

***Abstract*—This notebook paper presents the participation of Telefonica Research in the task of Video Copy Detection in TRECVID 2010. This is our second participation and, for this year, we have developed two local-based monomodal systems that we then combine using a score-based fusion to obtain a multimodal system output. We submitted 4 runs in total, whose main characteristics are described below:**

- **TID.m.[BALANCED/NOFA].fusion: These correspond to our main submission, both for the no false alarm and balanced profiles. They are based on the fusion between the local audio and local video monomodal systems.**
- **TID.m.BALANCED.videoonly: This submission is based on the monomodal video-based system using DART local features and with a temporal consistency postprocessing.**
- **TID.m.BALANCED.audioonly: This submission is based on the monomodal audio-based system using frequency-based audio local features.**

From these four systems submitted, two of them are processing only monomodal information (audio or video) and the fusion system takes the output of the previous two to output a fused result. Results for the monomodal systems in terms of NDCR are far from optimal, mainly due to an exces of false alarms that our monomodal systems still output. Results for F1 scores are very good for all cases. When combining the monomodal systems into he fusion the NDCR scores improve quite a bit as most false alarms are eliminated.

The proposed fusion turned out to work very well for combining our two monomodal systems. We will further investigate it to improve it for future evaluations.

I. Introduction

The final goal in the video copy detection task is to locate segments within query videos that occurs, with possible transformations, in a given reference video collection. Applied transformations can be inherent to the general video creation process, like encoding artifacts, video quality changing, etc. In addition, more complex transformations, which manipulate video content or its orientation, can be applied e.g., flipping, frame dropping, cropping, insertion of text/patterns like fixed banners or logos, etc.

As deduced from its definition, video copy detection can play an essential role in many applications for example search result redundancy removal, copyright control, business intelligence, advertisement tracking, law enforcement investigations, etc. In addition to the conventional method of watermarking, content-based copy detection is considered an alternative solution for video copy detection. In the watermarking approach irreversible information (i.e. watermarks) is embedded in the original video stream and is used to determine if a video is copied from another. One limitation of this approach is that the distributed videos should have been watermarked in the source, which adds an extra post-processing step for producing companies or individuals and not always can be done, as many times we do not have access to the source video. On the other hand through content-based approach, a set of content-based features are extracted from the video and are utilized to locate copied segments of query video from reference video dataset. With the content-based approach the content itself acts as the watermark. This approach is still a challenging topic because of various types of transformations applied and computational issues, although research on the area is progressing steadily.

In TRECVID video copy detection task, the focal point is to evaluate content-based approaches. The set of possible transformations are categorized into two groups: video and audio transformations. On the video modality there were a set of 8 possible transformations, and on the audio modality there were 7 possible transformations. The main differences from Trecvid 2009 evaluation were in the video transformations, where "simulated camcording" was added and one type of "picture in picture" was discarded. The audio transformations are combinations of 3 different categories that are intended to be typical of those that would occur in real world scenarios: (1) bandwidth limitation (2) other coding-related distortion (e.g. sub-band quantization noise) (3) variable mixing with unrelated audio content. Once transformed, queries were aligned and combined to create a multimodal query, which was the only condition evaluated (as opposed to last year's evaluation, where audio and video were also evaluated separately).

This year is our second participation in the TRECVID evaluation for video-copy detection and we are contributing a multimodal system based on two monomodal systems (one for audio and one for video) combined using a score-based fusion hat proved to work pretty well. Differently from last year's participation, this year we have focused on local-based

[1]E. Younessian participated in this project during a stay at Telefonica Research

features. We invested some time perfecting the global-based system presented last year [1], but did not submit any runs using it because it does not work well with the nature of data used in TRECVID. We have also spent some time perfecting the internal tools for processing the big amounts of data involved in the evaluation, which was a big problem in last year's submission for our team. Over all, we are very pleased with the results we obtained.

The implemented system for the visual domain aims at matching visual content across query videos and the reference video collection and determine corresponding visual similarity. It is done through two steps: offline indexing and online retrieval. Through offline indexing, we try to represent and index video data in an effective and compact form using our video local features, while through online retrieval step, we intend to link keyframes across query and reference videos using our proposed Visual Search Engine (VSE). Note that in video copy detection, because of much more amount of data which has to be processed than in general image copy detection, feature extraction and video indexing become a key point to develop an efficient and practical video comparison framework. After the keyframe matching step, we apply the temporal consistency enforcement upon linked keyframe pairs to find the copied video segment borders. Then final visual similarity score for each query-reference video pair is determined. Results on NDCR were not as good as expected, mainly due to the number of false alarms the system output (our tuning did not focus on totally eliminating them). On the other hand, the F1 scores turned to be well above median.

In the audio domain we implemented a variation of the fingerprinting method first proposed by [2] and then used by [3] for TRECVID 2009. We represent the acoustic data in the videos using a binary key extracted by comparing the differenced between adjacent frequency bands obtained from a short-term FFT transform of the signal at predefined intervals. In our acoustic retrieval system we first obtain a binary file containing the fingerprints of the audio track in each reference video, extracted every 20ms. Such fingerprints are then loaded and compared to the query fingerprints to obtain a putative matching segment start and end position, as well as a score which can be directly compared with all other videos. Likewise in the video-only system output, NDCR scores suffer from many resulting false alarms while F1 scores are very close to the best scores reported in the evaluation this year.

Both the audio and video monomodal systems output the Nbest reference videos they individually consider most similar to any given query, together with the segments supposed to match and their scores. The fusion step then takes over by analyzing these monomodal outputs to combine them into a multimodal output. The algorithm we used was inspired by [4] which uses both the ranking of each detected video in each modality as well as their matching score. The resulting fused output improves the NDCR scores from any of the individual modalities by reducing the number of false alarms detected, while not changing much the F1 score.

One of the teachings from TRECVID 2009 participation was the extreme importance of compacting the tools used to retrieve and manage the features extracted from the video collection. For this reason this year a big effort has gone into building tools that can process a video and store all necessary parameters in a compact and robust manner.

The rest of this paper is organized as follows. In the next sections we explain video and audio local feature preparation respectively. Next, we present and explain the official rvaluation results and finally we draw some conclusions and propose some future work.

II. TRECVID PROPOSED SYSTEMS

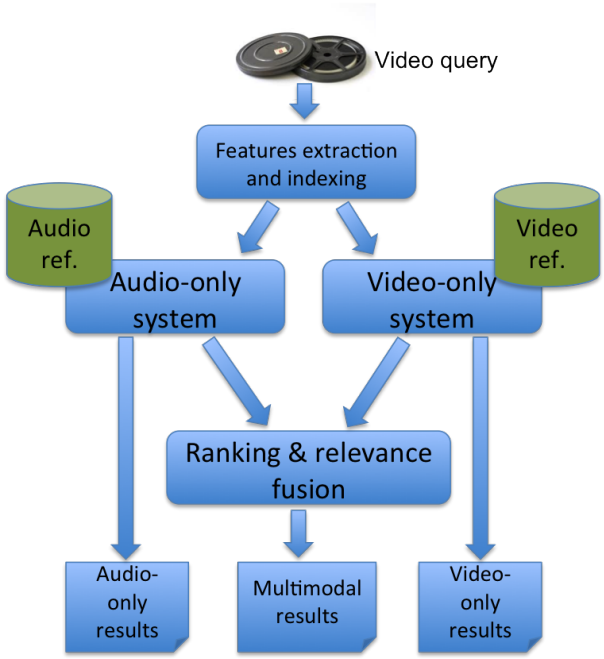

Fig. 1. *System presented for Trecvid 2010.*

As seen in Figure 1, this year's participation in Trecvid focuses on the fusion of two monomodal systems, one for audio-based and the other one for video-based video copy detection, both using local features. The video-based system is based on the recently proposed DART descriptors for keyframe matching [5], together with a post-processing step to take advantage of the temporal information in video [6]. The audio-based system uses frequency-based local features proposed by [2] and then used by [3] for TRECVID 2009. The fusion of both systems is a combination of rank and the normalized monomodal matching scores. In this section we first describe the two individual systems and then the fusion system.

III. VIDEO-BASED COPY DETECTION SYSTEM

Our video-based copy detection system is heavily based on local features and image-based retrieval in large collections of reference keyframes. The system consists of five main components: (i) key-frame extraction, (ii) local feature (keypoint) extraction, (iii) keypoint filtering, (iv) keyframe matching and (v) temporal consistency post-processing. The following sections provide detailed description of all these components.

A. Video local features extraction and post-processing

The local feature extraction and post-processing starts with video frame sampling in order to represent each video as a group of keyframes. Then, novel local features called DART [5] are extracted for every keyframe.

During the last year TRECVid experiments we observed that added banners and textual regions (inserted static text or dynamic subtitles) pose a serious challenge for the keyframe matching based on local features. Static banners (e.g. TV channel logo) can be easily matched across all videos where they appear causing many false alarms. Also, textual regions generate many keypoints, all with very similar descriptor components, leading also to false alarms. To alleviate such problems in this year system keypoints corresponding to static banners or inserted text are eliminated from further processing.

In the final step, only the most useful keypoints are selected for the keyframe indexing based on keypoints' scale and temporal stability. As a result, videos are represented with up to 400 most useful DART keypoints per keyframe.

1) Keyframe extraction: Similarly to other approaches relying on local-features, we sample the reference and query video frames in order to reduce the computational cost of the keyframe matching. We use uniform sub-sampling method with one frame-per-second sampling rate for both query and reference videos.

While the uniform sampling framework leads to more sampled frames and subsequently to a higher computational costs when compared to the non-uniform sampling methods, the uniform sampling does not depend on the performance of a shot boundary detection. Specifically, our choice was motivated by queries with complex scene transitions (fading in/out or dissolving, etc). and picture-in-picture in front of a background video where consistent shot boundary detection is very challenging.

2) DART keypoints extraction: In all submitted runs we used DART local features [5] developed at Telefonica Research. Typically DART performs better or comparable to Scale Invariant Feature Transform (SIFT) [7] and Speeded Up Robust Features (SURF) [8] in terms of repeatability, and precision vs recall [5]. Moreover, it is very attractive in the context of the video copy detection task because of its very low computational cost (6x faster that SIFT and 3x faster than SURF), and compactness (only 68 components).

On the other hand, it is worth mentioning that our keyframe matching module (described in section III-B) is generic and, except for the much higher computational cost, it could provide similar detection performances when used with other local feature extractors, such as the above-mentioned SIFT or SURF features.

In all our experiments the DART extraction tool was limited to extract up to 1200 keypoints per keyframe. These initial 1200 keypoints are later filtered using the text and static banners detection and the temporal consistency filtering described in the following sections.

3) Inserted text/patterns detection: In order to avoid false alarms caused by inserted static text or other patterns (e.g. banners) local features corresponding to such regions need to be excluded from the matching process. For this purpose we built a detector for static inserted text/pattern.

The detector operates by sliding a temporal window of 15 keyframes along the video. In our case this window size corresponds to approximately 15 seconds of video. For every keyframe an initial mask corresponding to static regions is created by finding pixels whose intensity has zero standard deviation within the temporal window surrounding the current keyframe. Then, dilation operator is applied to the initial mask in order to ensure an appropriate margins surrounding the static patterns and also to fill out possible inner holes. Conveniently, the method also removes keypoints close to black layout borders that are not very useful for matching.

It should be noted that the above mechanism was designed for videos of certain minimum length. In cases of very short videos (few seconds) showing relatively static scenes the method may detect most of the scene as being inserted static pattern and in consequence most of the corresponding keypoints will be eliminated.

4) Subtitle detection: Typical textual regions (e.g. dynamic subtitles or added text) generate many keypoints. Moreover, for most local features [7], [8], the generated keypoints have very similar descriptors. In order to limit the potential false alarms caused by this kind of keypoints the textual regions need to be detected and the corresponding keypoints excluded from the recognition.

Since typically close captions change many times over the duration of a single video, they will not be detected using the method relying on the zero value of standard-deviation described in the last section. Moreover, some videos in the TRECVid dataset contain moving texts or transparent texts which is not detectable using zero standard-deviation concept. Therefore we have developed a very simple yet effective dedicated subtitle region detector that relies on the analysis of the spatial density of vertical edges within every single keyframe.

In this method, first vertical edges (low-to-high or high-to-low transitions within every image row) are detected using the Sobel operator and binarized with respect to a predefined threshold. A pixel is classified as part of a textual region if the density of the edges within a sliding window centered at the pixel of interest is higher than a predefined threshold. Once all pixels are classified as text/no-text, the resulting initial mask is extended by applying the dilation operator within every row to ensure a secure margin around textual regions and also to fill out holes between or within letters. Since the above method relies only on the presence of vertical edges it works well for solid and transparent letters.

Our observations indicate that the above method is quite effective in detecting both, subtitles inserted at the bottom of the frame, as well as other inserted text such as the rolling cast part at the end of some videos or video title at the beginning. However, since the subtitles are much more common and the text detector false alarms at the bottom area of the image do not impact much correct image matching we have used two different sets of parameters for the dynamic text detector. For the bottom area (one fourth of the height) of the image we set the three parameters (thresholds controlling the

edge binarization, edge density, and the length of the sliding window) in a way to optimize high recall of the text detection, while in the remaining part the parameters were set to optimize high precision. Since the close caption text is expected to have smaller fonts than a text inserted in the upper part of the image the sliding window used for the bottom of the image was smaller than the one used for the upper part.

5) Keypoint tracking and filtering: As it was explained in section III-A2, the DART extractor was set to extract roughly 1200 keypoints per keyframe. Although in cases of very simple scenes the extractor may return less keypoints, and moreover many of the initial keypoints corresponding to static banners and text are filtered out, the number of the remaining keypoints is still prohibitively high for indexing. Fortunately our observations indicate that not all keypoints are equally useful and necessary for a successful recognition. In the past there were several methods proposed for selection of the most informative keypoints in static images [9]. However, we have observed that even as simple criterion as the scale at which keypoints were extracted provides very good indication about their usefulness for recognition. Moreover, as observed by other authors [10], it helps to use keypoints that are temporarily stable, i.e. can be tracked over time, when matching video frames. We combined both criteria to select only the most useful keypoints for further indexing. The objective is not only the reduction of the computational cost, but also reduction of the level of false matches that are created at the keypoint level.

Our keypoint importance measure u combines two criteria. According to the first criterion DART keypoints extracted at larger scales s are more important (more informative) than keypoints selected at lower scales. The second criterion includes the temporal stability of keypoints. Keypoints from a given keyframe are matched with the keypoints extracted from the previous and the next frames. The length of the detected keypoint trail measured as the number of the identified correspondences is taken as the indication of the keypoint's temporal stability. We will refer to this measure as the connectivity score c. Since in our current implementation the matching involves only the previous and the next frames, c can only assume values 0, 1, or 2.

In order to ensure low computational cost of the tracking the nearest neighbor search was implemented using kd-trees [11]. Two keypoints are considered as being matched based on the second nearest neighbor to the nearest neighbor ratio criterion proposed in [7]. Finally only matches that are consistent with the global transformation found using RANSAC algorithm are considered for the connectivity score.

Once the connectivity scores for all keypoints are determined we can compute their importance scores u using the following formula:

$$u(i) = s(i)/s_{MAX} + c(i) \tag{1}$$

, where $s(i)$ denotes the scale of keypoint i and s_{MAX} is the largest keypoint scale found within the current frame.

Finally, all keypoints are sorted according to their importance scores u and only the N-best keypoints with the highest value of u are retained for further indexing. Note that the importance measure u depends primarily on the connectivity score c. Only when two keypoints have exactly the same value of c the scales of the keypoints are influencing the ordering.

In all our experiments we set the keypoint budget to 400 keypoints per keyframe (while the DART extractor was set to extract 1200 initial keypoints per frame). Of course, if after removing keypoints corresponding to static banners and textual regions the number of the remaining keypoints is already smaller than the specified budget, the keypoint tracking and filtering step is skipped and all keypoints are used for indexing.

In most experiments, setting the budget to 400 keypoints per keyframe resulted in reducing the number of keypoints roughly by half. Additionally, we observed that by using the aforementioned spatial and temporal filtering results in more discriminative keyframe representation when compared to the use all initial DART keypoints.

B. Video-based matching algorithm

The video-based copy detection module operates in two stages. In the first stage all query keyframes are compared to all reference keyframes resulting in one ranked list of relevant reference keyframes for each query keyframe. The second stage involves a temporal post-processing where all ranked lists produced for keyframes from a single query video are combined using a voting mechanism.

1) Keyframe matching: The keyframe matching approach is inspired by the state of the art approaches to image retrieval based on Vocabularies of Visual Words [10], [12]. Similarly to these methods, our algorithm relies on local features (keypoints), hierarchical dictionaries of Visual Words [12], inverted file structures, and a spatial verification stage of the top ranked initial results [13].

First, in an off-line process a large number of descriptor samples is clustered into the Vocabulary of Visual Words, that defines the quantization of the descriptor space. From this moment every keypoint can be represented by its mapping to the closest Visual Word. Once the dictionary is created all reference keyframes are represented using an inverted file structure that stores all occurrences of visual words in reference keyframes.

The main difference between our method and the approaches mentioned ealier lies in the matching process itself. In our case the commonly used TF-IDF scoring mechanism [10] is replaced by clustering of the initial matching hypothesis in the pose space (limited to orientation and scale). The matching is performed in two stages: (i) initial ranking of reference images by voting in the limited pose space based on keypoints' poses (orientation and scale) and visual word IDs, and (ii) re-ranking of the initial results by performing more complex spatial verification based on the RANSAC algorithm [13].

For every query keyframe being processed the search results in a ranked list of the top matching reference keyframes. Only matches that have a consistent spatial consistency verified by the RASAC algorithm are included in the lists.

We have found that the inclusion of such a rudimentary spatial verification mechanism in the very initial stage of the recognition is very helpful in cases of small objects (e.g.

picture in picture) buried within complex scenes. It should be noted that the computational cost of our voting mechanism compares favorably to the TF-IDF scheme.

2) Temporal consistency post-processing: Given the search results for all query keyframes in a query video, in this step we obtain a list of the hypothesized matching reference videos given the video modality by applying an evolution of the temporal consistency algorithm we presented for TRECVID 2009, which is similar to the algorithm proposed by [6].

First, for every reference keyframe returned as possible match by any of the query keyframes in a query video we compute the relative difference Δt between the timestamp on the query keyframe and on the reference keyframe. We then insert the match into a differences histogram with bin resolution in milliseconds. Each histogram bin stores for each different reference video their number of matches at that time difference and the time of the first and last reference keyframes. Using such a histogram is a quick method to find matches that are consistent in time, as their relative time differences will usually be identical and therefore will create a peak in the histogram. We do this individually for every reference video appearing in the list of matching keyframes. Once finished, retrieving the top peaks in this histogram indicates which reference videos are most suitable matches for the given query.

During our tests with development data we noticed how sometimes the relative time difference for two video copies are prone to some jitter and therefore do not fall within the same exact histogram bin. This is due to the keyframe extractor we use which does not always return keyframes at the exact requested rate. For this reason we apply a post-processing to each of the histogram bins to include all those matches from the same video that fall within a short distance ϵ of the match (we used $\epsilon = 1$ second in our final submission).

Next, the best matching videos are retrieved and their final score is computed. First, the N reference videos (throughout our system $N = 20$) with highest keyframe matches for any particular histogram bin are retrieved. Then, for each of these videos we find the value of highest density of matches within a 10 seconds window along the matching region and normalize it to roughly return a final score between 0 and 1. This final score is used to re-rank the N-best videos output by the system, together with the matching region.

IV. Audio-based Copy Detection System

Given the good performance shown on TRECVID 2009 by [3] of the local acoustic features proposed initially by [2], this year we decided to implement our version of the features and matching system together with a novel postprocessing step to enhance the matching. After extracting the audio track from the video (either a query or a reference) we downsample the audio to 16KHz and bandpass limit it from 300 to 3KHz. In order to obtain the acoustic fingerprints we compute the FFT of the acoustic data every 10ms with an analysis window of 32ms and compute 16 mel-frequency bands on the data within the frequency bands mentioned above. Then each frequency band $F_i|i\epsilon 0 \ldots 15$ is compared with the next and a 15bit fingerprint vector v_f is obtained. Each position of the vector $v_f[i]$ is set to 1 if $|F_i| > |F_{i+1}|$ and to 0 otherwise. the obtained fingerprints are stored in binary format in main disk for each video extracted.

Comparison between reference and query is performed in a one-to-one basis (future work includes the indexing of all reference fingerprints for faster search). First we index the fingerprints corresponding to the reference into a hash table. If the fingerprints is composed of all 0's or all 1's we ignore it, as it most probably corresponds to an silence region, therefore non informative. In addition, in this step we also ignore those fingerprints that are repeated more than 10 times, which might indicate that although the spectrogram contains some information, it is the same across a long period of time, therefore it is thought to be non-discriminant for us.

Once all informative fingerprints from the reference video have been hashed, we use a similar method to the temporal consistency post-processing of video keyframes to locate regions within the reference video which contain sets of temporally aligned matches to the query fingerprints. For each query fingerprint we first retrieve the set of exact matches in the reference hash. In each case the time difference between reference and query is used to index it into the matches histogram. Differently from previously, for any given histogram position we create a new accumulator if the time distance between the current match and any previously indexed matches is bigger than 5 seconds. This way we ensure that false positive matches (with 15bit fingerprints we encounter quite a few of these) do not overextend the matching regions in the video. When a match is closer than 5 seconds from a previously inserted match the count of that accumulator is increased by one and the max/min matching timers are updated.

Upon finishing this step we find the accumulator with highest count and find an accurated similarity score given its reference-query relative alignment and matching region start and end times. For that we create a binary vector of matching query-reference pairs computed on the aligned fingerprint sequences between the start and end times and stored by the winning accumulator. Each position in the binary vector is set to 1 if the corresponding Hamming distance (bit-wise difference between the two fingerprints) is smaller than 4 (at most 3 bits are different). The final score is taken as the maximum of a 5 seconds running average computed along such binary vector.

Once all videos from the reference set have been compared with a single query video we output the ranked list of 20-best matching scores as an output of the audio-based algorithm.

V. Fusion of audio and video results

The fusion module in our video-copy detection system aims at processing the output of several partial outputs in order to obtain an enhanced final output for the system. The algorithm we used is able to process multiple partial inputs (not limited to two) and fuse their outputs into one. In the actual proposed system we are fusing the outputs from the local-features audio and video modules. The algorithm is inspired by [4] in that it uses the rank of each detected reference video within each

partial result, and it is extended to take also into account the score of each match as the rank alone turned not to be very effective when the number of partial results is small.

The algorithm takes as an input the partial outputs from the audio and video modules, composed of a ranked list of N-best reference video matches, together with the matching segments start and end positions, and their matching scores $S_{match}[i]|i\epsilon 0 \ldots N-1$. The first step of the algorithm normalizes the $S_{match}[i]$ scores to sum up to 1 (obtaining $\bar{S}_{match}[i]$). This is done so in order to make the comparison across modalities possible, as each modality has a different distribution of scores, dependent on how they are internally generated. Then, the resulting $\bar{S}_{match}[0]$ for both modalities are added together and stored for final normalization $S_{norm} = \bar{S}^{aud}_{match}[0] + \bar{S}^{vid}_{match}[0]$.

In addition to the normalized score, each partial match receives a score relative to their ranking within the partial list $S_{rank}[i] = \frac{N-i}{N}$ where the best match obtains a score 1, and the last in the list obtains a score $\frac{1}{N}$. The final partial score for each matching video is then defined as $S_{partial}[i] = \bar{S}_{match}[i] * S_{rank}[i]$.

Next, we insert all partial reference matching videos into a common/fused list, combining those matches that correspond to the same reference video and whose query and reference matching segments overlap at least 50% of the time. In these cases the resulting score is the sum of the $S_{partial}$ scores for each of the partial matches and the resulting start and end positions are the using of both matching segments. Once all partial queries have been inserted in the list, they are sorted and their final score is normalized by S_{norm} to avoid scores from being greater than 0. Note that if a reference video was found to be the best match for a query both in the audio and video modalities it will end up with a final score of 1. We are confident that in this case the match is not a false positive as both modalities strongly agree on it. Also, all false positives in either list will most probably not appear within the results from the other modality, therefore appearing with a lower rank in the final list.

As it can be seen in the results, this fusion strategy was very useful in reducing the rate of false positives in our results, therefore improving the NDCR metric. At the same time, the F1 metric was almost not affected by this fusion.

VI. TRECVID Participation Results

In this section we review the results obtained in our participation in Trecvid 2010 for each of the submissions, and we analyze the results when possible. As stated in the abstract, this year we submitted a total of 4 runs, covering 3 different systems in the balanced profile, plus one system on the no false alarm profile. The 3 systems submitted are: a) the results of the audio-only system; b) the results of the video-only system; c) the results of the fusion (submitted as nofa and balanced profiles). The following figures show the performances for these systems in the actual submitted working thresholds. In Fig. 2 and 3 the audio-only and the video-only runs are reported. Results in both cases are not as we expected in terms of NDCR. Analysis of these results brought to our attention that in this new set we are obtaining many more false alarms than in the development set we used, which caused a big increase in the NDCR. This is even more important in the audio-only system, where we observed that while in our development set we were able to discriminate well between videos, in the test set there were many videos with constant tones (or monotonous music) that were mistaken by others with similar characteristics. On the other hand, once a segment is detected we are obtaining for both audio and video-only systems quite good F1 scores, indicating that our algorithms for temporal consistency in video and the adjustment of the segment in audio wok as expected.

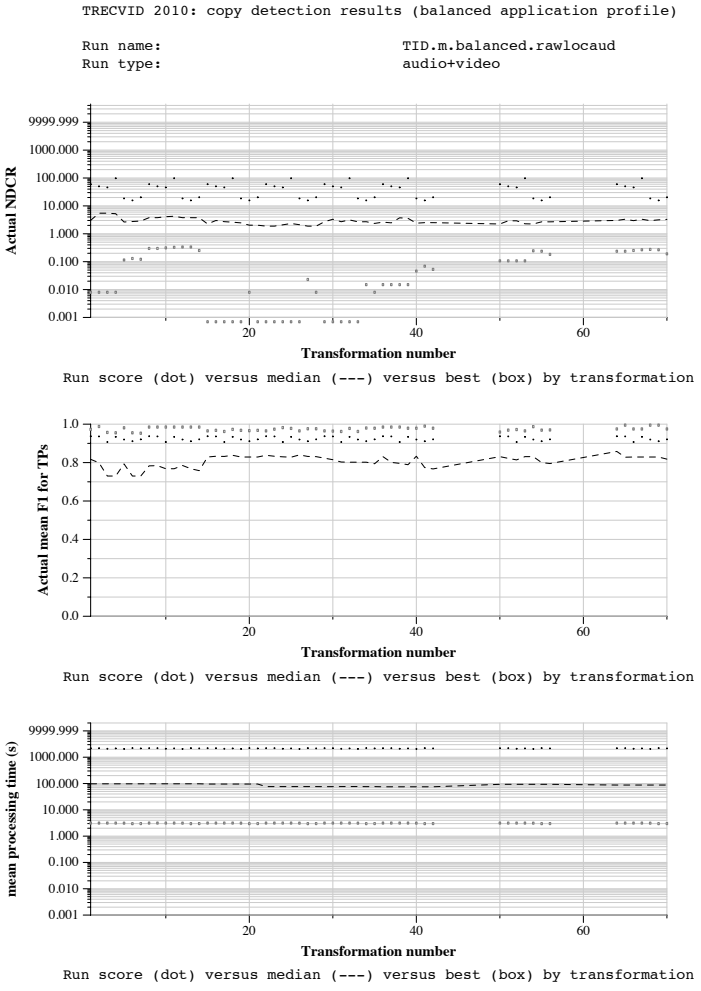

Fig. 2. *Results for local audio features, balanced profile, actual results.*

In figures 4 and 5 we show results for our main system submission, *i.e.* the fusion of the two individual modalities. Figure 4 shows results for the balanced profile, while Figure 5 refers to the no false alarms profile. In both figures we observe a big change in the NDCR scores in comparison

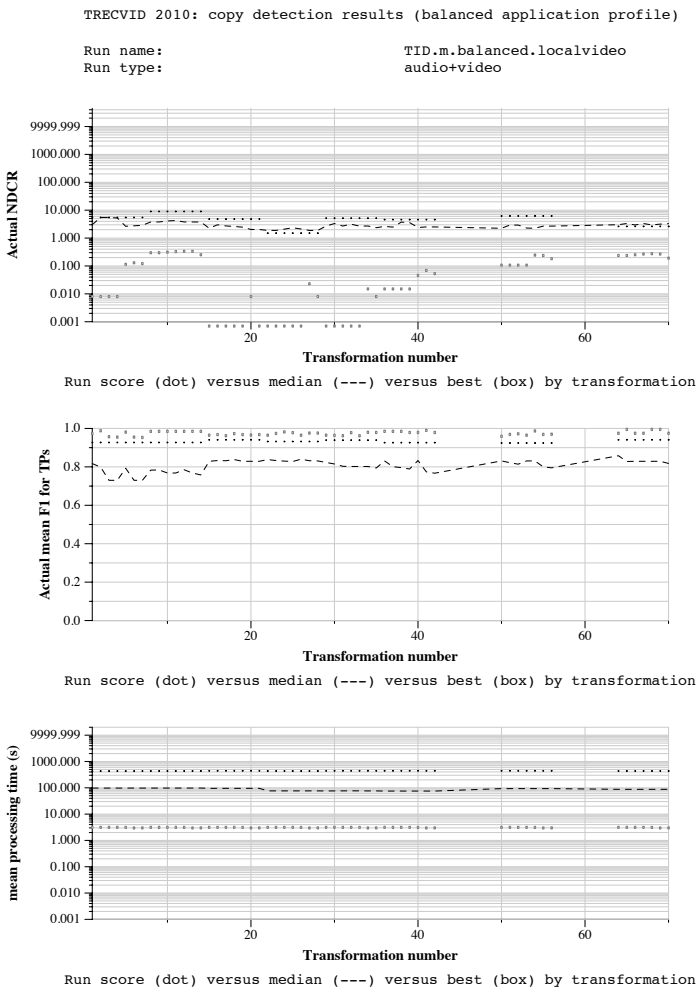

Fig. 3. *Results for local video features, balanced profile, actual results.*

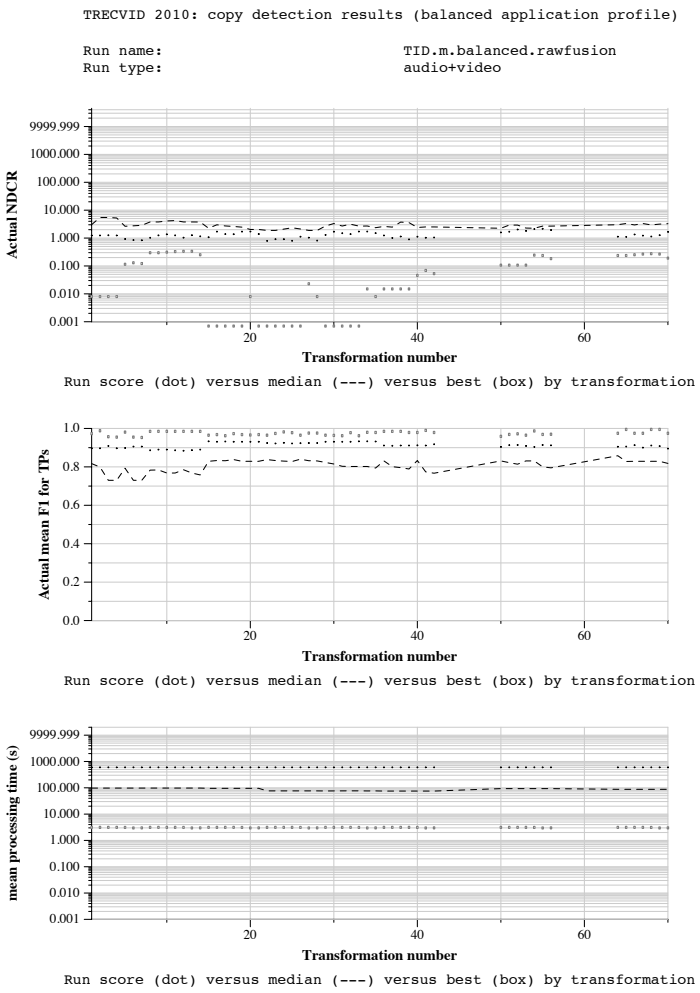

Fig. 4. *Results for the fusion system, balanced profile, actual results.*

with the monomodal systems, as now the fusion strategy of both individual outputs was able to reduce drastically the false alarms in our systems, therefore lowering the error in NDCR. This happened in both the balanced and the no false alarms profiles, where our scores are now better than median in NDCR. Note also how in the no false alarms profile we obtain for a few transformations very good results, close to the best ones. In terms if F1 scores we are more or less at the same level of results than in the individual systems. In the cases where we loose a bit of performance in F1 we think is due to the strategy followed in combining the output segments from each individual modality, which right now is performing a union of both segments and show be revisited.

In all the figures above we are obtaining runtime results that are very poor compared to the median of all other systems. In the audio-based system we are currently comparing each query video with each possible reference video, which has to be fetched from a network disk every time. Although such comparison is very fast, the time taken to fetch the features for each reference video and the accumulated time to process them linearly increases with the size of the reference database. We are currently not able to store all the audio fingerprints on memory, which would speedup the process immensely. In the video-only system we have in place an indexing system for individual keyframes. Given the big number of keyframes in the reference set of videos we had to split the processing among several machines, each one reporting times related to the querying of the query keyframes into the reference indexing as well as some common procedures for all. The total time reported is the accumulated time for all the machines and for all the query keyframes in a query video. Furthermore, given our computing restrictions, both in the audio and video systems we took advantage of the fact that we knew in the reference database which query videos and audios were

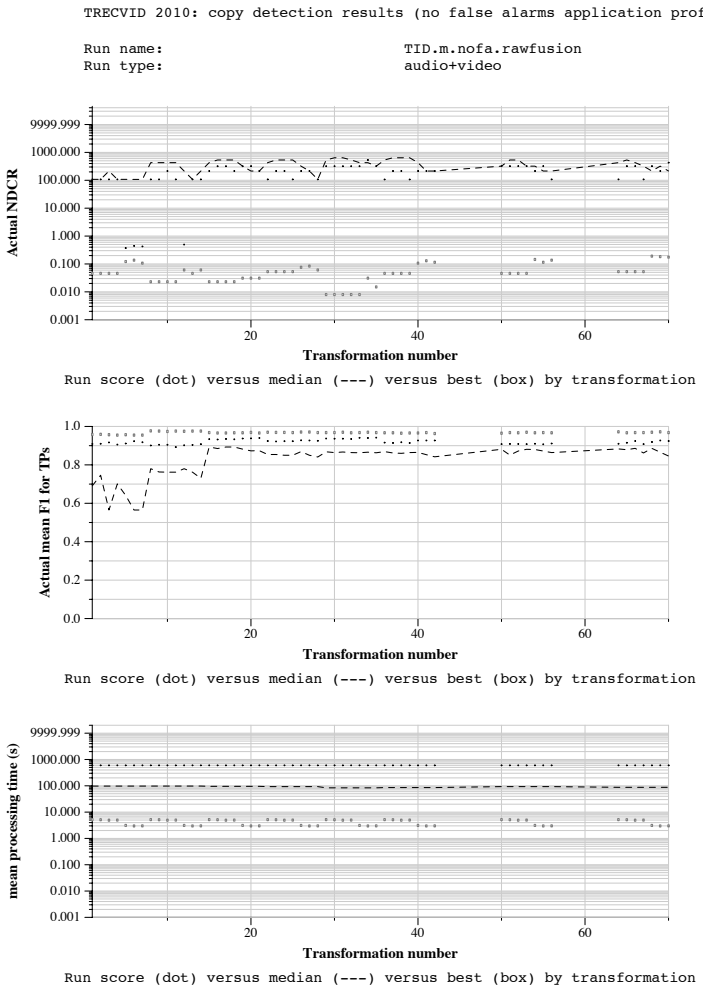

Fig. 5. *Results for the fusion system, no false alarms profile, actual results.*

repeated for combination into the final queries, therefore we just processed 1/8 of the total audio queries and 1/7 of the total video queries, and then reported equal results for each of the copies. In the case of fusion we combined the correct files from each of the two subsets. When reporting processing time we multiply by 7 and 8 the total times spent for every query in the video-only system and in the audio only system, respectively. We are planning for next evaluation to work on indexing strategies that allow us to speedup the querying at full query level, both for audio and video, as well as buying machines with more RAM memory to fit all databases in them.

VII. Conclusion and future work

This is the second year that Telefonica Research is participating on the Trecvid evaluations for video copy detection. This year we have reformulated our systems and created a multimodal system formed of two monomodal systems, one for audio and the other for video data. In both monomodal systems we are using local features, approach that has given good results in the past for Trecvid databases. In our system the video module uses DART features, which have been demonstrated to work better than SIFT and SURF, while also achieving higher processing speeds. For the fusion of the two modalities we propose an algorithm inspired on one used previously for image retrieval, which takes into account not only the ranking of the retrieved videos on the individual modalities, but also their output matching scores. Over all, we are very happy with our results, although our monomodal NDCR scores are a bit faulty due to the large number of false alarm that our systems produces. These are eliminated by the fusion, which gives us above than median performances for both NDCR and F1 scores. Our future work will involve making the system faster and more robust to false alarms.

Acknowledgment

Telefonica I+D participates in Torres Quevedo subprogram (MICINN), cofinanced by the European Social Fund, for Researchers recruitment. X. Anguera and T. Adamek have been partially funded by the Torres Quevedo program.

References

[1] X. Anguera, P. Obrador, T. Adamek, D. Marimon, and N. Oliver. (2009, November) Telefonica research content-based copy detection trecvid submission. Trecvid 2009 Online proceedings. [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/mesh.pdf

[2] A. K. Jaap Haitsma, "A highly robust audio fingerprinting system," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[3] M. Heritier, V. Gupta, L. Gagnon, G. Boulianne, S. Foucher, and P. Cardinal. (2009, November) Crims content-based copy detection system for trecvid. Trecvid 2009 Online proceedings. [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/crim.pdf

[4] X. Olivares, M. Ciaramita, and R. van Zwol, "Boosting image retrieval through aggregating search results based on visual annotations," in *Proc. ACM MM*, 2008.

[5] D. Marimon, A. Bonnin, T. Adamek, and R. Gimeno, "DARTs: Efficient scale-space extraction of daisy keypoints," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[6] Z. Liu, T. Liu, and B. Shahraray. (2009, November) http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/att.pdf. Trecvid 2009 Online proceedings. [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/att.pdf

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. European Conference on Computer Vision (ECCV)*, May 2006. [Online]. Available: http://www.vision.ee.ethz.ch/~surf/index.html

[9] G. Fritz, C. Seifert, and L. Paletta, "A mobile vision system for urban detection with informative local descriptors," in *ICVS'06*, 2006.

[10] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477. [Online]. Available: http://www.robots.ox.ac.uk/~vgg

[11] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *In Proc. IEEE Conf. Comp. Vision Patt. Recog*, 1997, pp. 1000–1006.

[12] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2006, pp. 2161–2168.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2007.