Multimodal Video Copy Detection Applied to Social Media

Xavier Anguera
Multimedia Research Group,
Telefonica Research
Via Augusta 177
Barcelona, Spain
xanguera@tid.es

Pere Obrador
Multimedia Research Group,
Telefonica Research
Via Augusta 177
Barcelona, Spain
pere@tid.es

Nuria Oliver
Multimedia Research Group,
Telefonica Research
Via Augusta 177
Barcelona, Spain
nuriao@tid.es

ABSTRACT

Reliable content-based copy detection algorithms (CBCD) are at the core of effective multimedia data management and copyright enforcement systems. CBCD techniques focus on detecting videos that are identical to or transformed versions of an original video. The fast growth of online video sharing services challenges state-of-the-art copy detection algorithms as they need to be: able to deal with vast amounts of data, computationally efficient and robust to a wide range of image and audio transformations. In this paper, we present two related multimodal CBCD algorithms that effectively fuse audio and video information by means of a compact multimodal signature based on audio and video global descriptors. We validate our algorithms with a benchmark database (MUSCLE-VCD) and obtain over a 14% relative improvement with respect to state-of-the-art systems. In addition, we illustrate the performance of our approach in a video view-count re-ranking task with YouTube data.

Categories and Subject Descriptors:
H.2.8 Information Technology and Systems: Database Applications
H.3.3 Information Technology and Systems: Information Search and Retrieval

General Terms: Algorithms, Experimentation.

Keywords: Video Copy Detection, Youtube, Multimedia Information Retrieval, Near-duplicate Videos, Multimodal Processing, Video Search.

1. INTRODUCTION

In today's digital world, we face the challenge of developing efficient multimedia data management tools that enable users to organize and search multimedia content from vast public repositories whose content is constantly growing. Increasing storage capabilities at low prices enable the archival of most of the professionally and user generated multimedia content.

The detection of video duplicates in a video database is one of the key technologies in multimedia management. It is particularly relevant today due to the large number of user generated content (UGC) available in video sharing sites. Its applications include storage optimization, copyright enforcement, improved video search and concept tracking.

Traditionally, watermarking techniques have been used to detect copies in images, audio or video [3, 7]. These techniques insert information (watermarks) into the media that is not noticeable by the user and that is later used to proof the authenticity of the media. One limitation of this approach is that the watermarks need to be included when generating the material, which is typically not the case in UGC. Alternatively, Content-Based Copy Detection (CBCD) techniques do not add any watermarks into the media: the *media itself* is the watermark.

CBCD techniques analyze the content and extract a set of features that are then used to compare it with putative duplicates. In this paper, we focus on the detection of Near-Duplicate Video Clips (NDVC) via a novel multimodal CBCD algorithm.

The definition of near-duplicate video clips varies, depending on the researcher. In [23], a near-duplicate video is defined as an identical or very similar video to the original one, where the authors include a set of possible transformations (*e.g.* file formats, encoding parameters, photometric variations, edition operations, changes in length and several frames adition/removal, etc.) applied to the original video. In [17, 19], the definition is expanded to include videos that contain the same scenes, even when shot by different cameras. Finally, Basharat *et al.* [2] also consider the semantic similarity of the videos, *i.e.* videos that contain the same semantic concept but with different scenes. The definition used in this paper is that proposed by Wu *et al.* [23], as it captures a wide range of transformations that are typically found in real multimedia databases.

The application of CBCD algorithms to videos has been usually approached from a video analysis perspective, by extracting and comparing features from the videos. Local features, typically computed at a key-frame level [14, 20], achieve good performance at high computational cost. Conversely, researchers have also investigated the use of global features ([8, 6]), extracted from low level features in the video, to conform a *fingerprint* or signature of the video. Video fingerprints allow for fast video comparisons at the expense, however, of lower performance.

In this paper, we propose two efficient multimodal algorithms for CBCD that use global features that are extracted from the audio and video tracks in order to create audio and video descriptors. Once the descriptors have been computed, the information in all modalities is combined in order to determine whether the two videos are near-duplicates or not. The proposed algorithms achieve a performance similar to or better than state-of-the-art monomodal CBCD local

WSM'09, October 23, 2009, Beijing, China.

algorithms, but with significantly lower computational cost. In addition to validating our algorithms on the MUSCLE database, We apply our approach to view-count re-ranking task of YouTube videos.

The rest of this paper is structured as follows. Section 2 gives an overview of the most relevant related work. The proposed algorithms for content-based video copy detection are presented in Section 3. Section 5 summarizes the experimental evaluation of such algorithms using a benchmark database. In Section 6 we apply the proposed approach to a user view-count re-ranking task using YouTube data. Finally, our conclusions and directions of future research are presented in Section 7.

2. RELATED WORK

In this section, we review the most relevant previous work on content-based copy detection, including multimodal and image-based approaches.

2.1 Image-based CBCD Algorithms

The vast majority of previous CBCD work has focus on analyzing the video modality. Depending on the nature of the features used in the analysis, we find authors that have used: *local* features extracted from video keyframes; *global* features computed from the low level features; and *a combination of* local and global features.

Local Feature-Based CBCD Algorithms

The work that has focused on local features typically applies single image analysis/comparison techniques to CBCD and requires an extra step in order to align the features from the two videos under analysis. These algorithms tend to achieve good matching performances at the cost of high computational burden. Recently, the main focus in this area has been on reducing the computational costs while maintaining the performance figures [20, 14, 18, 24]. We direct the reader to [11] for a comparative study of the performance –both in terms of accuracy and speed– of several global and local algorithms on the same database. They show how a local algorithm that was previously proposed by the authors achieves the best accuracies and speed performance.

Global Feature-Based CBCD Algorithms

The use of global features for CBCD is of interest for researchers and for commercial applications, because the extraction and comparison of global video signatures is usually faster than using local features. However, global algorithms have typically been linked to poor performances. In addition, they require a minimum video length in order to reduce false alarms in the comparisons. Initial work in this area comes from [9, 13]. Shortly afterwards, Hampapur *et al.* [5, 6] present the first exhaustive analysis of multiple global features for CBCD, pointing out their pros and cons. In [21], they propose a signature that is derived from a directed graph that is constructed from the quantization of the total brightness between adjacent frames in a video segment. Hoad *et al.* [8] offer a thorough review of global approaches and propose a system using a shot-length-based signature and a centroid-based signature. They propose to fuse all the signatures into one signature by interleaving the values for each frame. Unfortunately, the proposed matching methods are limited to work with full queries against videos of similar or longer length. Therefore, they are unable to search for subqueries within the queries, as done in this paper. Finally, Wu *et al.* [23] recently proposed a hierarchical approach, mixing both local and global features for the task of web video search.

2.2 Multimodal Multimedia Analysis

Recently, a number of authors have looked at videos with a multimodal perspective. In [22] and [25], the authors propose systems for tracking news stories that use both visual features and text transcripts. The most relevant multimodal paper to our work is the work by Naphade *et al.* [12], where the authors propose using both the audio and video tracks to support audio-visual queries. On the audio side, they use dynamic time warping (DTW) in order to align the Mel frequency cepstral coefficients (MFCC) from the audio tracks of the two videos being compared. On the video side, they compare global features (color histogram, edge direction histogram, motion magnitude and motion direction histograms) independently. The fusion of all aligned streams is done by means of a weighted sum. Unfortunately, they assume that the optimum alignments of all independent streams belong to the same DTW path, as they consider the optimum path value for each stream. The work presented in this paper addresses this shortcoming. Finally, [4] implements a monomodal system to compare videos by means of local descriptors, but proposes an adaptive and parameter-free method for decision making using several inputs, which could be easily extended to multimodal data.

3. MULTIMODAL NDVC DETECTION

We propose two related multimodal video copy detection algorithms to detect near-duplicate videos by fusing information from the audio and the video tracks. The goal of the algorithms is to determine whether two videos are near-duplicates of each other or not. The algorithms work in two steps. First, global descriptors are extracted in both the video and audio tracks independently. Second, the similarities between the audio and video tracks are computed and fused in order to return a final score that determines the similarity between both videos.

The first matching algorithm (*full query*) is optimal when the goal is to determine whether the entire query video, as a whole, appears in any of the videos in the database. The second algorithm (*partial query*) is designed to allow for portions of the query video to appear in the video database. The latter case is very relevant in video sharing sites where UGC is pervasive. Note that it is technically more challenging than the first case.

3.1 Audio Descriptor Extraction

The global audio descriptor is extracted by analyzing the acoustic changes that take place in the audio track of the video. A one dimensional signal is created from the audio track whose minima correspond to places where there is a change in the speaker or in the sounds being recorded.

First, video and audio tracks are separated into independent streams. The audio track is downsampled to 16KHz, 16 bits/sample and converted to 24 Mel Frequency Cepstrum coefficients (MFCC), including cepstrum derivatives, acceleration and energy, computed every 10 milliseconds. These features are typically used in speech processing and acoustic segmentation techniques. In particular, we use the Bayesian Information Criterion (BIC) [16] as the basis to obtain a measure of how similar the audio is in both sides of the analysis point (and hence whether there is a change in the audio signal or not).

In our present work, BIC is used as in [1] for acoustic segmentation by computing the acoustic metric for frame i:

$$\Delta BIC(i) = \log L(X[i-W, i+W]|M_{ab}) - \log L(X[i-W, i]|M_a) - \log L(X[i, i+W]|M_b) \quad (1)$$

where $X[i, j]$ is an n-dimensional sequence corresponding to the MFCC features extracted from the audio segment from time $t = i$ to $t = j$; $M_{a,b,ab}$ are three Gaussian Mixture Models (GMM) composed of a weighted sum of Gaussian Mixtures, such that the number of Gaussians in M_a and M_b is the same and half of the number of Gaussians in M_{ab}; and W corresponds to the analysis window, set to 100 MFCC frames (one second of audio). The ΔBIC acoustic metric is computed every 100 milliseconds in a sliding window along the input signal, obtaining an acoustic change signature with 10 points for every second of video, which accounts for a 0.16kbps signature stream. Figure 1a shows an example of the ΔBIC acoustic change pattern extracted for a 173 second long video. The local maxima in the plot represent places where the surrounding audio is maximally similar. Conversely, the local minima represent acoustic change points; small changes within the same acoustic condition (positive value) or large changes from one acoustic condition to another (negative value).

3.2 Video Descriptors Extraction

The global video descriptors are extracted by considering the temporal visual changes occurring in the video. Three features are extracted (hue, luminance and highlights centroid variation) from which two descriptors are generated.

3.2.1 Hue and Luminance Variation (HLV)

The first descriptor of video variation consists of the change in color and luminance, in a similar way as the audio signature. A similar approach was proposed by Hoad and Zobel [8], where they considered all color channels (YCrCb) in order to detect video variations. Note that direct comparison of video signals is avoided by using the *change* in color and luminance, instead of the color itself. The authors show that only two channels (Hue and Value) are needed in order to detect near-duplicates. Therefore, we first convert the input video signal to the HSV (Hue, Saturation, Value) color space. Next, we compute the Hue and Value histograms at each frame H_{x_i} and compare them with the histograms from the previous frame, using a simple histogram intersection measure:

$$I(H_{x_i}, H_{x_{i-1}}) = \frac{\sum_{l=1}^{M} min(H_{x_i}(l), H_{x_{i-1}}(l))}{\sum_{l=1}^{M} H_{x_i}(l)} \quad (2)$$

where M is the number of bins in the histograms. Note that $I = 100\%$ implies no change at all, *i.e.*, exactly the same video frame, and $I = 0\%$ implies maximum change, *i.e.*, in checkerboard type images, changing from white to black.

The comparison is performed on a frame by frame basis without any temporal down-sampling. Therefore, this technique allows to track typical consumer camera variations that expand over multiple frames such as auto white balance or auto contrast.

Figure 1b depicts an example of the combined Hue and Value change features for a 173 sec long video. Sudden drops in this histogram intersection plot indicate scene changes, while ramps indicate fade-in/outs or other progressive changes.

3.2.2 Highlights Centroid Variation(HCV)

This descriptor tracks the highlights (areas of high luminosity) in the image, since the motion of high luminosity areas should be very similar between a video and a transformed version of it. This feature was first proposed by [8] in conjunction with a shadow centroid variation feature (areas of low luminosity). The authors found that the highlight centroid feature was enough for near-duplicate detection.

The image is first down-sampled by a factor of 8, both in the horizontal and vertical dimensions. A histogram of the Value channel is computed and the top 10% pixels in the histogram (*i.e.*, brightest pixels) are used to calculate the highlight centroid. The centroid is normalized to the diagonal of the frame, in order to be resilient to size change degradations in any of the axes (both horizontal and/or vertical). The centroid of the highlight region is then compared to the centroid of the highlight region from the previous frame. The change in the centroid position from frame to frame is calculated in absolute value, in order to be resilient to the horizontal flip of the entire (or part of the) video clip[1].

Therefore, this descriptor (see example in Figure 1c) tracks the movement of the highlight centroid from frame to frame in absolute numbers. If the highlight region remains static from frame to frame, this will result in zero absolute change; if it pans slowly, it will generate a roughly constant value; and if there is a scene cut, it will typically generate a high peak at the frame where the cut takes place.

3.3 Audio/Video Descriptor Postprocessing

Once the audio and video descriptors have been extracted, they are post-processed in order to obtain signals well suited for their fusion.

The audio descriptor contains most of the fingerprinting information in the low frequencies, with a high frequency component which is basically noise. Therefore, a low pass filter is applied: an average window with width of four frames, centered in each frame. The same filtering is applied to the highlights centroid variation (HCV) signature.

Conversely, the hue and luminance variation feature (HLV) contains the fingerprinting information in the peaks of the signal, indicating the points where a change of scene happens. In this case, the a derivative filter is applied (after computing the absolute value of the signals) using the following regression formula:

$$x_{filtered}[n] = \frac{\sum_{i=1}^{\Theta} i(x[n+i] - x[n-i])}{2\sum_{i=1}^{\Theta} i^2} \quad (3)$$

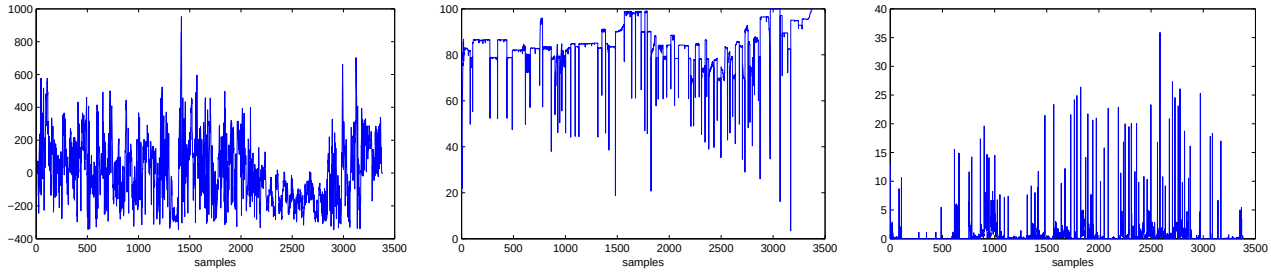
where $x[n]$ is the hue or luminance variation feature at frame n and $\Theta = 2$. This filtering does a windowed differential sum for each point where the further the samples (in time), the larger their weight.

All feature streams are then normalized to zero mean and unitary variance. Finally, as will be seen in the next Section, the number of samples extracted per second in the video channel is larger than in the audio channel. Therefore, the audio descriptor is warped to fit the video descriptor by means of a linear warping.

4. QUERY MATCHING ALGORITHMS

In this section, we describe two multimodal full or partial query matching algorithms that combine the previously described audio and video descriptors in order to detect

[1]Described as one of the transformation types in the MUSCLE-VCD dataset.



a) Audio variation b) Hue and Luminance c) Highlights Centroid variation

Figure 1: *Examples of the proposed signatures for a* 173 *sec long video.*

NDVC. In the following, we will denote the input video as the *query* whereas the video that the input is tested against will be the *queried* video.

4.1 Full-query Matching Algorithm

The full-query matching algorithm is appropriate when the *entire input query* is expected to be found in the queried video. The queried video might be a near-duplicate of the query of similar or longer duration, containing the query inside. The proposed algorithm is extremely fast when searching for NDVC where both the query and queried videos are compared in their entirety. It is particularly suitable to detect advertisements in broadcasted material, where predefined advertisements (*i.e.*, query) are detected in life broadcast video (queried video).

Once the audio and video descriptors have been extracted from the videos to be compared[2], the Generalized Cross Correlation with Phase Transform (GCC-PHAT) [10] between the descriptors is computed. The GCC-PHAT implements a frequency domain weighting of the cross-correlation between two signals, given by:

$$R^{GCC}_{x_1,x_2}(m) = \mathcal{F}^{-1}(X_1(w) \cdot X_2^*(w) \cdot \psi_{PHAT}(w)) \qquad (4)$$

where $X_1(w)$ and $X_2(w)$ are the signals being compared in frequency domain (usually converted from time domain using Fast Fourier Transform, FFT); $\mathcal{F}^{-1}$ is the inverse of the FFT and $\psi(w)$ is a weighting function:

$$\psi_{PHAT}(w) = \frac{1}{||X_1(w) \cdot X_2^*(w)||} \qquad (5)$$

where $||\cdot||$ indicates the modulus. The GCC-PHAT returns a normalized cross-correlation plot (with values between 0 and 1) with a maximum at $m_{\max}$, *i.e.,* the delay for which both signals are most similar.

The GCC-PHAT metric is computed for each of the audio and video descriptor pairs from the two videos, obtaining a cross-correlation plot for each of them. All resulting plots are added together to form a multimodal cross-correlation plot. The maxima of this plot correspond to the temporal delays with the optimal alignment for all modalities. Once the point of optimum alignment (*i.e.,* global maximum) is found, $m_{\max}$, the similarity score between both videos, $D(v_1, v_2)$, is defined as the weighted-normalized scalar product of all descriptors in the optimum delay:

$$D_{v_1,v_2}(m_{\max}) = \sum_{i=1}^{N} W_i \cdot \frac{x_1 \cdot x_2[m_{\max}]}{|x_1^i||x_2^i[m_{\max}]|} \qquad (6)$$

where $W_i$indicates an *a priory* weight assigned to each available signature i ($i = 1..N$); $x_2^i[m_{\max}]$ is the signature in the query video, delayed to the optimum match delay; and $|\cdot|$ is the $L2$-norm of each signature vector: $|x| = \sqrt{\sum_n x^2[n]}$.

Note that if the point of optimum alignment leaves points in either signal without a counterpart, they are matched with 0. The tree signatures previously described are used for the full-query matching algorithm: one acoustic (acoustic variation) and the two computed from the visual part (hue and luminance variation and highlights centroid variation). Therefore, $N = 3$ and the weights are set to $W_1 = 0.5$ for the acoustic and $W_{2,3} = 0.25$ for each visual descriptor in order to give equal importance to the audio and visual tracks.

In the case of NDVC, it is expected that the original and near-duplicate videos will be of similar length: possible alterations include adding a few frames at the beginning and/or the end due to editing modifications. In order to enforce this constraint, a penalty is applied to the similarity score given by Eq. 6 according to the difference in length between both videos:

$$D'_{v_1,v_2}(m_{\max}) = D_{v_1,v_2}(m_{\max}) \cdot \frac{\max(s_{v_1}, s_{v_2})}{\min(s_{v_1}, s_{v_2})} \qquad (7)$$

where s_{v_i} corresponds to the length (in video frames) of video v_i. The resulting normalized score D'_{x_1,x_2} ranges from a negative value (when the two signals are completely different) to 1 when the signals are identical. A similarity threshold θ_{full} needs to be defined in order to determine the minimum score between any two videos to be considered near-duplicates (see subsection 5.1.1 for a related discusssion).

4.2 Partial-query Matching Algorithm

The partial-query matching algorithm is more flexible than the full-query matching algorithm because it is able to find *partial* sequences of the query that are embedded in the queried videos. It is useful when only parts of the query are copies (altered or not) of a segment of the queried video. This flexibility in the matching comes at a higher computational cost, as no information is known *a priory* regarding which sequences from either video match each other. Therefore, we propose an iterative algorithm that finds possible matching sequences, and their length.

The partial-matching algorithm takes the same input as the full-query algorithm: the audio and video descriptors from the two videos to be compared. The query signal is

[2]Note that this is typically done offline for all videos in the system.

split into windowed segments of size $T = 1500$ video frames, sliding along the signal every $S = 300$ video frames. Each segment is compared to the entire queried video at each available modality by means of the GCC-PHAT measure in Eq. 4. The cross-correlation measures are fused together into one cross-correlation plot. However, in this case the n-best ($n = 3$ in the system presented) maxima of the fused cross-correlation plot are found and the n-best delays are saved into a delays histogram. This process is repeated in all windowed segments of the query. The use of a delays histogram follows the rationale that the delays defining segment matches should remain constant throughout the entire segment, ending with a peak in the histogram, whereas non-matching segments should return random delays. Note that these delays are stored in the histogram in *absolute* terms – with respect to the beginning of the query signal.

Next, the best alignment delays are determined from the delay histogram by finding its global maximum (max_count) and selecting delays within a [max_count-1, max_count] range. For each delay, the query and queried video windowed segments –in each of the available modalities– are retrieved and the weighted dot-product is computed (given by Eq. 6). The similarity between the segments corresponds to the largest dot-product. As in the full-query matching algorithm, a threshold θ_{part} needs to be defined that sets the minimum similarity between two segments to be considered near-duplicates, which will be discussed in subsection 5.1.2.

5. EXPERIMENTAL EVALUATION

We applied the proposed algorithms in two different settings. First, they were tested using a well known benchmark database in order to compare their performance with the published state-of-the-art in video copy detection. In addition, the full-query matching algorithm was applied to a real NDVC detection task using Youtube [3] data.

5.1 MUSCLE-VCD Benchmark Evaluation

The benchmark evaluation was performed using the CIVR 2007 publicly available Video Copy Detection database [4]. This database is composed of 80 hours of videos from various sources and two sets of queries.

The first set, ST1, contains 15 near-duplicate query videos (2h 30min) that are the result of applying a video and/or audio transformation to corresponding original videos existing in the database. The duration of each of the query videos ranges between 6min and 45min and contains some queries that do not exist in the database, and therefore should not return any near-duplicate.

The second set, ST2, consists of 3 query videos (45min): a total of 21 excerpts of original videos existing in the database, inserted in various locations of the queries, within videos that are not in the database. Various transformations have also been applied to excerpts in ST2. Their lengths range from 24sec and 123sec.

After a very careful analysis of the data set, we found a mismatch in 3 excerpts of ST2 of at least 3 video frames between the audio and video tracks of the query excerpts and their database counterparts. Such mismatches might have been introduced by the software used to create the ST2 queries and were considered out of the scope of our research. Therefore, the mismatched excerpts were not considered in our results. The excerpts belonged to: query 1, one excerpt, paired with video 43; and query 3, two excerpts, paired with videos 46 and 16, respectively.

[3]www.youtube.com

[4]MUSCLE-VCD-2007, www-rocq.inria.fr/imedia/civr-bench

Table 1: CIVR'07 MUSCLE database results

Method	Q_{ST1}	Q_{ST2}
Best CIVR 2007	86%	86%
Best proposed by [15]	93%	86%
Proposed algorithm	**100%**	**88%**

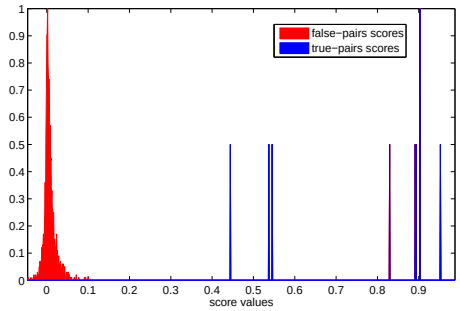


Figure 2: *MUSCLE-VCD ST1 scores histogram.*

The precision and recall figures are computed for both of the proposed algorithms, in addition to the scores proposed by the MUSCLE-VCD database for comparison with previous work: query quality for ST1 and segment quality for ST2:

$$Q_{ST1} = \frac{\#\text{ Correct\_queries}}{\#\text{ Queries}} \quad (8)$$

$$Q_{ST2} = \frac{\#\text{ Correct\_segments - }\#\text{ False\_alarms}}{\#\text{ Query\_segments}} \quad (9)$$

where $\#Queries$ and $\#Query\_segments$ are the total number of queries and segments to be detected in the data set.

All experiments reported in this section are conducted using the multimodal algorithms described in Section 3. The three audio and video signature streams presented earlier were extracted from all videos for this test.

5.1.1 Full-query Algorithm Evaluation

Table 1 compares the results given by the proposed algorithms with the best results obtained in the CIVR'07 benchmark evaluation (freely available from the MUSCLE-VCD website) and those reported by [15] in ACM MM'08.

On the ST1 task, the proposed algorithm obtains a query quality of $Q_{ST1} = 100\%$, as it correctly detects all videos given their transformed queries. Both precision and recall are 1. Note that the threshold θ_{full} in Eq. ref had to be set. Figure 2 shows the normalized histograms of the scores for query-video comparisons between near-duplicate pairs (*true tests*) and non duplicates (*false tests*), according to the ground truth. Note that there are only 10 out of 15 queries in the ST1 set that have a near-duplicate in the database. Hence, the histogram contains very few points.

The scores for the *true tests* and the *false tests* are very far apart. Therefore, the threshold θ_{full} could be set to any value between 0.10 (maximum value of the *false tests*) and 0.44 (minimum value of the *true tests*) in order to achieve 100% quality (equivalently, recall and precision of 1).

The length mismatch penalty factor in Eq. 7 was applied to penalize the videos of very different length when compared to the query's length. All queries in the MUSCLE-VCD benchmark have similar lengths to those of their near-

duplicate pairs. Additionally, we applied our algorithms without the length penalization in order to understand the impact that this penalization had on the results: we obtained the same performance although the similarity scores of true and false tests were closer to each other (between 0.26 and 0.45).

In our experience with the MUSCLE-VCD ST1 dataset, the proposed multimodal full matching algorithm has shown to be accurate and robust to compare full queries.

5.1.2 *Partial-Query Algorithm Evaluation*

The partial-query matching algorithm benchmark evaluation was performed using the ST2 set on the MUSCLE-VCD database. Taking out the faulty subsegments (as previously explained), we used 18 subsegments for the test, divided into 3 queries. Results for this evaluation are summarized in the third column of Table 1, where we compare the obtained results with those presented in [15] and the best reported results in the CIVR'07 evaluation. The segment quality obtained is 88% (with precision and recall being 1 and 0.88, respectively), a 14% relative improvement over the two compared systems –representative of the state-of-the-art in this domain.

In this case, the similarity scores of the true and false video pairs are much close to each other than in the ST1 set, given the increased difficulty of this task. The similarity threshold was set to $\theta_{part} = 0.5$. The proposed algorithm did not produce any false alarms, but missed two true video-pairs which had lower scores than the threshold. Analysis of results indicated that the two missed video-pairs corresponded to short queries (shorter than 30 seconds). Note that the minimum sub-query length is directly proportional to the analysis window: the proposed algorithm is designed to successfully analyze queries of at least 30 seconds of length. We are currently working on a variation of the algorithm that would handle very short queries as well.

5.2 Monomodal vs Multimodal Performance

The use of multiple modalities should make the CBCD system more robust to degradations in the signals and therefore increase its performance. Usually, transformations occurring in one modality do not tend to affect the other modalities to the same degree –more so if they are orthogonal, such as audio and video. In this section, we analyze the impact of using multiple descriptors in the proposed CBCD system.

Figure 3 shows an exemplary cross-correlation of one of MUSCLE-VCD videos when compared with one of the provided queries, where the true delay position between both videos is 694 frames. The first three plots show the cross-correlation of the monomodal streams and the last one corresponds to the fusion (weighted sum) of all the signatures.

Note how the *multimodal fusion* correlation is the only one with a global maximum at the true delay position. All the monomodal plots have higher peaks at different (erroneous) delays. In addition, the monomodal correlations (especially the one corresponding to the highlights centroid variation feature) have spurious peaks that disappear in the fusion correlation due to the complementary nature of the combined features.

Furthermore, some of the transformations applied to the near-duplicate videos might eliminate one of the modalities or are invariant to the features that are used to characterize the videos. In these cases, the use of multiple signatures and modalities enables the detection of the near-duplicates, which would have been impossible using only that modality.

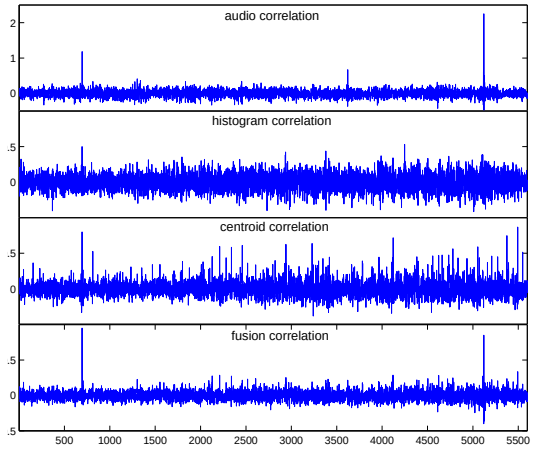


Figure 3: *Monomodal vs. multimodal cross-correlations comparison.*

Table 2: Multimodal vs. monomodal performance

Test	Metric	Audio	Histog.	Centroid	Multim.
ST1	Q_{ST1}		0.86		
test	Prec.	1	1	1	1
	Recall		0.86		
ST2	Q_{ST2}	-1.05	0.5	-3.83	0.88
test	Prec.	0.15	0.9	0.13	1
	Recall	0.22	0.55	0.72	0.88

An example of this situation is video 30 and query 1 from ST2 in the MUSCLE-VCD benchmark, where the audio has been eliminated from the near-duplicate version. Another example is a color video that has been transformed to pure black and white, such that the Hue variation histogram intersection feature would had been flat.

Table 2 shows the results of running the system using only one of the modalities versus using the proposed multimodal approach. For the ST1 task we used a threshold $\theta_{full} = 0.27$ whereas for the DT2 task we used $\theta_{part} = 0.5$. While both monomodal and multimodal approaches achieve high performances in ST1, the value of our multimodal approach is illustrated in more challenging ST2, where the monomodal algorithms perform significantly worse.

Finally, the processing times of the multimodal algorithm were of 12.36s and 29.58s per query for the ST1 and ST2 sets respectively running on a Quad-core PC 2.40GHz with Ubuntu Linux. Indexing techniques and source code optimization could certainly be applied to the research software in order to speedup processing.

6. YOUTUBE VIEW-COUNT RE-RANKING APPLICATION

After validating the proposed CBCD algorithms in a benchmark video-copy detection task, we evaluate the performance of the full-query matching algorithm in a real-world task with a larger number of videos and uncontrolled conditions. In particular, we turn our attention to a *YouTube video user-views re-ranking task*.

Users of YouTube can rank the retrieved videos by the number of views or by the people that have added them to their favorite lists. The top videos in these rankings are the

Table 3: YouTube Dataset: most seen and favorite videos

YRank	Query	# videos	# views
1	Avril Lavigne Girlfriend	329	114,544,079
2	Evolution of Dance Judson Laipply	239	112,947,349
3	Chris Brown With You	406	86,944,392
4	Leona Lewis Bleeding love	396	82,142,709
5	Jeff Dunham Achmed the dead terrorist	139	82,130,969
6	Rihanna Don't Stop the music	350	82,107,836
7	Charlie bit my finger	247	81,598,077
8	Alicia Keys No one	355	76,151,467
9	Timbaland Apologize	316	72,859,854
10	Miley Cyrus 7 things	338	69,077,504
11	Chris Brown Kiss Kiss	463	65,049,438
12	Jonas brothers SOS	311	60,012,466
13	mysterious ticking noise	313	58,833,782
14	Jonas Brothers Burnin up	436	58,691,354
15	Timbaland The way I are	374	57,694,983
16	Crank dat Soulja Boy Spongebob	292	56,522,654
17	White & nerdy	494	44,951,916
18	ok go Here it goes again	347	44,367,541
19	Charlie the unicorn	247	33,908,363
20	daft hands	325	28,578,798
21	jizz in my pants	330	27,718,250
22	the mean kitty song	346	17,745,038

most popular videos among YouTube users. These listings, however, are not error-free. In particular, the videos that have more than one copy or near-duplicate in the system have fewer chances to rank in the top positions when compared to unique videos, as they probably share user clicks with their near-duplicates. Note that all the near-duplicates of a particular video typically share the keywords associated to them. The user-views re-ranking task aims at *grouping together all duplicate or near-duplicate videos and use their cumulative view-count in order to create a re-ranked and more accurate list of most-viewed YouTube videos.*

We downloaded a subset of the 20 most viewed and the 20 most favorite videos in YouTube at *all times*. For each video in the list, we formulated the query that would return that video and all the videos related to it, including duplicate videos, in a very similar manner as done by [23]. From the list of 20 + 20 videos, we obtained a unique list of 27 video queries. After careful analysis, we eliminated 2 queries for explicit content and 3 additional queries because they were very vague and did not return the original video among the results (*e.g.,* "amazing guitar" or "shoes"). Finally, 22 queries were submitted to YouTube (listed in Table 3). For each query, we used the Google Gdata API[5] to retrieve the list of videos related to that query with their corresponding view-count. This was done at the beginning of March 2009. Although the view-counts change constantly, their Youtube ranking order (YRank, in column 1) remained stable until the end of this task.

Column 3 in Table 3 shows the number of videos returned by YouTube for each query. It ranges from 139 to 494 videos. These numbers differ from those reported by [23] for two reasons:

- In our experiment, we collected videos with at least 1 user view. We did not consider videos without any views as this means that either they are very new or no one usually reaches them with their textual queries, and ultimately, they will not affect our user-views re-ranking test.
- At the time that [23] was published and to the best of our knowledge, YouTube was not taking any actions to avoid exact duplicates from being posted on the site. However, nowadays YouTube issues an alert to the user when posting an exact duplicate and exact-duplicates are not shown in the search results. This reduces the number of retrieved duplicate videos from the percentages reported in [23]. However, near-duplicate video clips (NDVC) do not seem to be filtered by YouTube's algorithms.

[5]http://code.google.com/apis/YouTube

Table 4: YouTube user-viewcount re-ranking results

YRank	% dupl.	# dupl. views	new order	Δorder
1	9.45%	122,229,521	2	-1
2	1.68%	112,955,647	4	-2
3	1.48%	91,412,971	10	**-7**
4	13.67%	121,655,890	3	+1
5	15.21%	99,454,675	6	-1
6	6.59%	93,407,477	8	-2
7	5.64%	86,865,850	11	-4
8	16.38%	92,412,368	9	-1
9	20.31%	144,080,515	1	**+9**
10	1.18%	69,279,853	13	-3
11	22.07%	95,238,804	7	+4
12	13.22%	65,162,784	14	-2
13	8.65%	59,159,077	17	-4
14	1.37%	60,254,493	15	-1
15	14.47%	77,833,150	12	+3
16	8.24%	60,198,244	16	=
17	14.40%	109,644,280	5	**+12**
18	0.86%	44,543,320	18	=
19	4.47%	35,898,816	19	=
20	9.58%	29,619,923	21	-1
21	25.53%	30,863,043	20	+1
22	17.39%	20,428,873	22	=

We computed the acoustic and visual features for all retrieved videos and used the full-query algorithm –described in Section 4.1– to compare the most seen video in each query with the rest of the videos retrieved by that query.

In this case, the similarity threshold $\theta_{YouTube}$ was set to 0.20 which was experimentally found to minimize the false-alarm rate at the expense of a non-zero miss rate in some queries.

Table 4 shows the re-ranking results. Column 1 corresponds to the original YouTube ranked list as shown in Table 3. Column 2 indicates the percentage of duplicate or near-duplicate videos (NDVC) found for that query. Note how the % of near-duplicate videos ranges from 0.86 to 25.53. Query number 18 has the lowest percentage of near-duplicate videos. Interestingly, this is a musical video clip (typically prone to many copy uploads) where the most popular version is a home-made version done by the artists. In this case, the query retrieves a number of *imitation* videos that are *semantic* near-duplicates of the original (*i.e.,* very similar audio with completely different video) and hence beyond the scope of our work in this paper.

Column 3 in Table 4 shows the resulting ranking once the view-counts for NDVC are added together. Finally, the last column shows the change in the ranking for each the

videos with respect to the original YouTube ranking. Note that finding a lot of NDVC does not necessarily imply a big jump in the re-raking because the NDVC might have very few views. This is the case of the video initially ranked at position 21. It has the highest percentage of NDVC (25.53%) but its new ranking is only one position higher. However, there are other videos with very significant changes in their rankings, such as the videos originally placed in positions 9 and 17, that move up to positions 1 and 5, respectively. Their new cumulative views are 144 and 109 million, $\geq 100\%$ larger than the original number of views listed in Table 3.

While the YouTube test presented here was performed in a reduced set of videos returned by a query, we believe that it is indicative of both the capabilities and limitations of the proposed algorithm in a real-life setting. We are working on a large-scale deployment of our algorithm in the context of an shared video search task. We believe that NDVC algorithms will be at the core of video search in shared video sites.

7. CONCLUSIONS AND FUTURE WORK

Copy detection techniques are at the core of multimedia data management and copyright enforcement. They focus on detecting the videos that are either identical copies (duplicates) or transformed versions (near-duplicates) of an original video.

In this paper, we have proposed two multimodal algorithms for content-based video copy detection (CBCD) that fuse three audio-visual descriptors. The first algorithm (full-query) focuses on comparing two videos where one of them might appear in its totality inside the other video. The second algorithm (partial-query) addresses the situation where only part of a video might be copied and appear inside another video. Tests reported on the MUSCLE-VCD benchmark database (MUSCLE-VCD) show the superiority of the proposed algorithms to state-of-the-art published results by, at least, 14% of relative improvement. Finally, we illustrate the performance of the full-query algorithm in a view-count re-ranking task using YouTube data. Future work will focus on carrying out a large-scale deployment of our approach in the context of Internet video search on social media sites, dealing with possible asynchronies between the audio and video tracks, on automatically setting the weight of the mono-modal streams according to their relevance and on exploring additional features that could be used in our framework.

8. REFERENCES

[1] J. Ajmera, I. McCowan, and H. Bourlard. Robust speaker change detection. *IEEE Signal Processing Letters*, 11(8):649–651, 2004.

[2] A. Basharat, Y. Zhai, and M. Shan. Content based video matching using spatiotemporal volumes. *Journal of Computer Vision and Image Understanding*, 110(3):360–377, 2008.

[3] I. Cox, J. Kilian, F. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[4] N. Gengembre, S.-A. Berrani, and P. Lechat. Adaptive similarity search in large databases - application to image/video copy detection. In *Proc. Content-Based Multimedia Indexing Conference*, 2008.

[5] A. Hampapur and R. M. Bolle. Comparison of distance measures for video copy detection. Technical report, IBM Thomas J. Whatson Research Center, 2001.

[6] A. Hampapur, K.-H. Hyun, and R. Bolle. Comparison of sequence matching techniques for video copy detection. In *Proc. Conf. on Storage and Retrieval for Media Databases*, 2002.

[7] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings IEEE, Special Issue on Identification and Protection of Multimedia Information*, 87(7):1079–1107, 1999.

[8] T. Hoad and J.Zobel. Detection of video sequences using compact signatures. *ACM Transactions on Information Systems*, 24(1):1–50, January 2006.

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

[10] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-24(4):320–327, August 1976.

[11] J. Law-to, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Buoujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proc. CIVR*, 2007.

[12] M. R. Naphade, R. Wang, and T. S. Huang. Supporting audiovisual query using dynamic programming. In *Proc. ACM Multimedia*, 2001.

[13] M. R. Naphade, M. M. Yeung, and B.-L. Yeo. A novel scheme for fast and efficient video sequence matching using compact signatures. In *Proc. SPIE, Storage and Retrieval for Media Databases*, pages 564–572, 2000.

[14] S. Poullot, M. Crucianu, and O. Buisson. Fast content-based mining of web 2.0 videos. In *Proc. PCM, also in LNCS 5353*, 2008.

[15] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *Proc. ACM Multimedia*, 2008.

[16] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[17] H. T. Shen, X. Zhou, Z. Huang, J. Shao, and X. UQLIPS: a real-time near-duplicate video clip detection system. In *Proc. VLDB*, pages 1374–1377, 2007.

[18] H. sik Kim, J. Lee, H. Liu, and D. Lee. Video linkage: Group based copied video detection. In *Proc. CIVR*, 2008.

[19] M. Takimoto, S. Satoh, and M. Makauchi. Identification and detection of the same scene based on flash light patterns. In *Proc. IEEE-ICME*, pages 9–12, Los Alamitos, CA, USA, 2006.

[20] H.-K. Tan, X. Wu, C.-W. Ngo, and W.-L. Zhao. Accelerating near-duplicate video matching by combining visual similarity and alignment distortion. In *Proc. ACM Multimedia*, 2008.

[21] M. Toguro, K. Suzuki, P. Hartono, and S. Hashimoto. Video stream retrieval based on temporal feature of frame difference. In *Proc ICASSP*, 2005.

[22] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Novelty detection for cross-lingual news stories with visual duplicates and speech transcripts. In *Proc. ACM Multimedia*, 2007.

[23] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *Proc. ACM Multimedia*, 2007.

[24] X. Wu, Y. Zhang, S. Tang, T. Xia, and J. Li. A hierarchical scheme for rapid video copy detection. In *Proc. IEEE Conf. on Applications of Computer Vision*, 2008.

[25] Y. Zhai and M. Shah. Tracking news stories across different sources. In *Proc. ACM Multimedia*, 2005.