CRIM'S CONTENT-BASED COPY DETECTION SYSTEM FOR TRECVID

Maguelonne Héritier, Vishwa Gupta, Langis Gagnon, Gilles Boulianne, Samuel Foucher, Patrick Cardinal

Centre de recherche informatique de Montréal (CRIM)
{Maguelonne.Heritier, Vishwa.Gupta, Langis.Gagnon}@crim.ca
{Gilles.Boulianne, Samuel.Foucher, Patrick.Cardinal}@crim.ca

ABSTRACT

Approach we have tested in our submitted runs: For visual- based copy detection, we find links between video shot key-frames using a probabilistic latent space model over local matches between the key-frame images. This facilitates the extraction of significant groups of local matching descriptors that may represent common semantic elements of near duplicate key-frames. For 2009, we have worked on an optimal representation of the test database. We first select the discriminant local descriptors. Then, we quantize the selected local descriptors into a hierarchical structure.

For audio based copy detection, we give results with two different feature parameters: 15-bit energy difference parameters similar to [1] and a feature-based mapping of test frames to query frames.

Differences we found among the runs: We submitted 1 run for the video only copy detection task (same run for Balanced and for nofa). Four runs were submitted for the "audio only" copy detection task :

- CRIM.a.NOFA.EnNN2pass: energy-diff parameter search rescored with nearest-neighbor mapping.
- CRIM.a.NOFA.NN22para: search using nearest-neighbor mapping.
- CRIM.a.BALANCED.EnNN2pass: lower threshold than for NOFA case.
- CRIM.a.BALANCED.EnNN22wt15: fuse Energy-diff parameters search (wt 15) with nearest-neighbor mapping search.

We fused the video submission from CRIM with each of the four audio only submissions to get four different submissions for audio+video copy detection task. The threshold was adjusted based on the results of 2008 a+v queries.

Relative contribution of each component of our approach: For visual-based copy detection, the probabilistic latent space model over local matches between the key-frame images produces a robust and accurate filtering process in relation to all possible local matches. It works well even if there are only a few local matches between the key-frames of the copied video in question. We have introduced a new method for SIFT quantizing. It improves the time computation performance while keeping a good precision for SIFT representation.

For audio only copy detection, the fingerprints obtained by mapping each test frame to the nearest query frame (NN-based fingerprints) reduced minimal NDCR by half over that obtained with energy-difference based fingerprints.

This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC)

What we learned about runs/approaches and the research question(s) that motivated them : Approaches based on local descriptor matching are efficient for video copy detection but very time consuming. Our method is more adapted when there is very little common visual information to establish a link between two key-frames. Video copy detection may not need such a good precision. For audio copy detection, mapping each test frame to the nearest query frame (NN-mapping) results in robust audio copy detection. The minimal normalized detection cost rate (NDCR) for even the worst case transformations is less than 0.03 for 2008 queries, and less than 0.075 for 2009 queries. The algorithm provides easy parallel processing on a graphics processing unit, leading to a very fast search.

Index Terms— video copy detection, audio copy detection, copy detection, near duplicate detection.

1. INTRODUCTION

Video copy detection or near-duplicate detection (NDT) in movies is a relatively new topic [2] as it offers an alternative to watermarking for copyright control, business intelligence, advertisement tracking and law enforcement investigations. Videos often contain audio. Sometimes the original audio is retained in the copied material, sometimes it is replaced by a new soundtrack. Nevertheless, audio is an important and strong feature for some application scenarios of video copy detection. Audio copy detection is used to monitor peer-to-peer copying of music or any copyrighted audio over the internet. It is also used to monitor advertisement campaigns over the TV and radio broadcasts.

Visual NDT often proceeds via a video summarization approach like reducing a video in a set of key-frames. The copy detection task then consists of finding near-duplicates in key-frame images [3][4][5]. Matching key-frames through a set of key-points is an interesting strategy as it is robust to occlusions and illumination changes. Also, invariant descriptors for the key-points provide robustness to view point change. Two different groups of approaches based on key-point matching techniques have been proposed in the literature. One group (e.g. [6][7]) filters out the outliers between the whole key-frames using robust matching methods such as RANSAC or Least Median of Squares (LMS). However, those fitting methods perform poorly when the ratio of inliers falls below 50%. This requires a large overlap between a pair of images for an efficient matching process. In practice however, key-frames of two similar video segments can differ significantly due to the presence of motion in the scene or the key-frame generation process. Also, RANSAC is not efficient if there are only few inliers between 2 near duplicate key-frames. The second more recent group of approaches seek to find common spatial patterns (e.g. [8], [9]). These approaches are mainly based on comparing key-point neighborhoods. However,

there is an ambiguity in the choice of the neighborhood size used for the comparison. Moreover, outliers can be present in the neighborhood. In fact, it is always possible to obtain erroneous matches due to the presence of common local structures. Some authors ([8], [9]) use an efficient representation inspired from text analysis called Bag Of Words (BOW), to represent neighborhoods. BOW represents a text document as a vector; counting the number of occurrences of different words as features. In [8], [9], descriptors are quantized into clusters which are analogous to words in a text document. The BOW representation has two shortcomings when dealing with ambiguities: polysemy (i.e. a word having two different meanings) and synonymy (i.e. two words with same meaning). BOW generative models capture the co-occurrence information between elements in a collection of discrete data by introducing a latent variable (i.e. a context value), in order to raise the ambiguities of the BOW representation. BOW generative models are used in natural language processing and statistical text analysis to discover topics in documents [10].

In [11] and [12], BOW generative models are used to extract and link place features and cluster recurrent physical locations (key-places) within a movie. It finds links between key-frames of a common key-place based on the use of a probabilistic latent space model over the possible local matches between the key-frames. This allows the extraction of significant groups of matching descriptors that may represent characteristic elements of a key-place. Here, we adapt this approach for the video copy detection task. The BOW is used to represent key-frame images. BOW generative model filters out uninformative matches, generated by very common image structures, and extract groups of matches that may represent structural elements representative of near duplicate key-frames. Inliers are extracted, whatever the outlier number, by using a latent value for each match. A latent value is a context value shared by a group of local matches that may represent a structural characteristic element (analogous to a topic for text document).

However key-points based methods are very time consuming. It requires a very big database representation. Each video from the test database is represented by a set of several key-frames which are also represented by a set of several local descriptors. For 400 hours video, we get a minimum of 100 millions local descriptors. Some authors ([8][9][5]) use prototype-based clustering such as K-means to quantize local descriptors into a limited number of prototype (about 200 000). This quantization introduces errors, since cluster prototype may be not well defined. The cluster prototypes are only a coarse representation of the clustered descriptors. Also, 200 000 prototypes is not enough to represent well all the variability that a 128 dimension size vector could have. We propose a quantizing method which reduces the 128 floating point representation into 17 short values. We also propose an efficient comparison function between two quantized descriptors.

When we look at the published papers in audio copy detection and in advertisement detection, we see that the two fields have evolved differently. In audio copy detection [13] [14], the emphasis is on speed, since we compare the copy with a huge repository of copyrighted audio. Small percentage of misses will not make a big difference as long as we capture most of the copying. It has to be robust under various coding schemes and distortions that speech may go through over the internet. A fast audio copy detection uses audio fingerprints. In audio fingerprinting, they use energy differences in consecutive bands to generate a feature with 32 bits. The search is speeded up by looking for exact match of these 32 bits in the stored repository. A more complete search is only performed around these matching frames. This process has been shown to be robust to many coding schemes and audio distortions over the internet [13].

In advertisement detection, the emphasis is more on finding all the ads broadcast in the campaign [15] [16] [17]. The process is speeded up by first using a fast search strategy that overgenerates the possible advertisement matches. These are then compared using a detailed match. The detailed match in many instances includes comparing video features, as in some instances, the same audio may be played even though the video frames may be different.

We have experimented with the copy detection and the advertisement detection algorithms. We published a paper on advertisement detection where we use a fast search followed by a detailed matching algorithm [17]. We also experimented with the energy difference parameter used in audio copy detection [1]. We found that this parameter is very robust to various query transformations in the 2008 TRECVID copy detection competition [18]. We experimented with a new parameter which maps frames of test audio to nearest query frames. We show that this mapping is robust to the query transformations and reduces the missed segments from 7.7% to 1.6% for 2008 queries. These new parameters can be computed using a graphics processing unit (GPU) resulting in an accurate and fast copy detection. Rescoring the segments found using the energy difference parameters with these new parameters results in 1.7% missed segments with only a small increase in computing.

This notebook paper is organized as follows. Section 2 give details about the methodology and implementation of our approach for the video copy detection task. Section 3 describes the audio copy detection system. Section 4 describes the fusion of the audio and video system. Section 5 presents the evaluation process and performance results on the TRECVID dataset 2009.

2. VIDEO COPY DETECTION SYSTEM OVERVIEW

Our approach is derived from our last year's approach [19] which finds links between video shot key-frames, based on a probabilistic latent space model over local matches between the key-frame images. This allows the extraction of significant groups of local matching descriptors that may represent common characteristic elements of near duplicate key-frames. We combine it with various pre-processing steps designed to accelerate and improve the matching process for any query type, as well as post-processing steps designed to find the copied video segment borders. Figure 1 illustates the global video copy detection system.

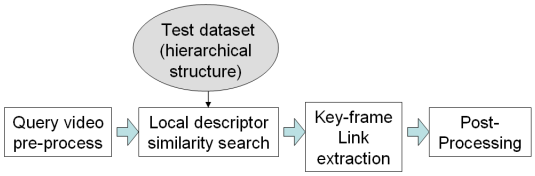

Fig. 1. *Video copy detection system.*

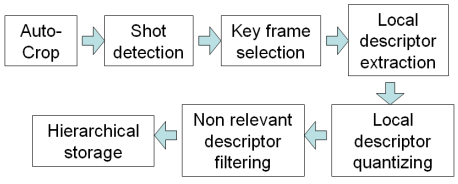

Fig. 2. *Pre-processes for video test database.*

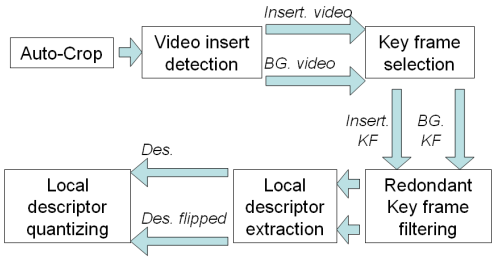

Fig. 3. *Pre-processes for query video.*

Most of the pre and post processing processes are similar as those from last year (shot detection, key frames selection, auto-crop, video insert detection, redundant key frame filtering, local descriptor extraction, post-processing). Figures 2 and 3 illustrate pre-processing steps for query videos and test dataset videos. We focused this year on an optimal representation of the test dataset. Indeed, we get a minimum of 100 million local descriptors to represent 400 hour of test dataset videos. We first introduce a non discriminant descriptor filtering step based on Latent Dirichlet allocation modeling. This eliminates about 40% of non relevant local descriptors from the test dataset. We propose a quantizing method for SIFT local descriptor which reduces the 128 floating point representation into 17 short values. This is combined with an efficient comparison function between two quantized descriptors. Then, quantized descriptors are stored in a hierarchical structure for fast retrieval.

2.1. Key-frame extraction and local descriptor extraction

Once the automatic shot transition detection is completed, each shot is then summarized in a few representative frames (key-frames). To this aim, we compute the overlap between images using a simple method based on camera motion estimation [20]. The algorithm finds the optimal frame path over the shot which then minimizes the overlap between frames. We extract local descriptors for each key-frame. First, Regions Of Interest (ROI) are automatically detected in the image with a difference of Gaussians (DOG) point detector from which we derive local descriptors using SIFT [21]. We use SIFT because it performs the best in terms of region representation specificity and robustness to image transformations [22].

2.2. Local descriptor quantizing

Each SIFT descriptor is in fact composed of 16 gradient histograms concatenated representing 16 independent regions. Therefore, each of these 16 regions can be represented independently. We define a vocabulary set to represent each region. We use normalized vectors on a regular grid in a four dimensional space to define our vocabulary. We chose a regular grid to be able to represent equally any possible region configuration. Therefore, we get a vocabulary of size 16 and we use 16 bit value to code each region. Each region histogram is projected in the vocabulary space. Let $\{V_i\}_{i=1..N_v}$ be the vocabulary set, X the normalized histogram projection in the vocabulary space and c_i the i-th bit of the corresponding code value C for X. For each i, if $d(V_i, X) < Threshold$, $c_i = 1$ else $c_i = 0$ where d is the L1 distance. We also project the vector composed by the norm values of each region histogram in the vocabulary space. Therefore, we obtain a vector composed of 17*16 bit size values to represent one SIFT descriptor. We can now use a very fast distance measure to make correspondence between descriptors. If two descriptors have the same sequence values, they are matched immediately, otherwise, a quick comparison function is applied. Let $\{C_0^j\}_{j=1..17}$ and $\{C_1^j\}_{j=1..17}$, two descriptors representation. A correspondence is set if $C_0^j \& C_1^j \neq 0$ for all j. The following scheme (figure 4) illustrates our coding scheme. We create an additional quantized

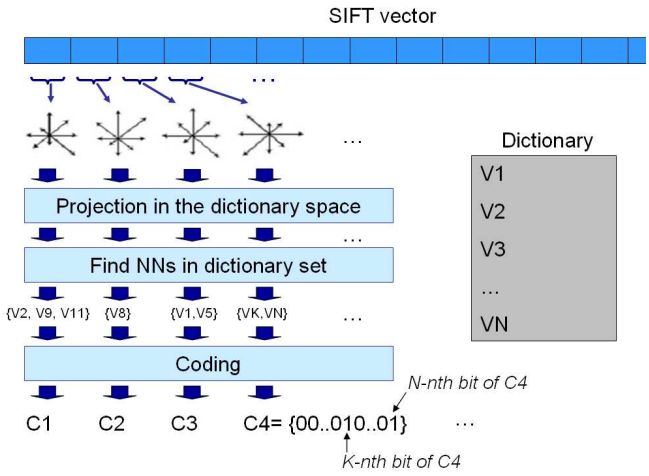

Fig. 4. *SIFT coding scheme.*

value (coarse quantized value) for each descriptor by using a similar coding process with a dictionary of size 2 associated with an index comparison function. We use this index to filter out quickly descriptors pairs for the finest comparison measure. The advantage of this quantization method is that the precision level can be defined by the vocabulary size or by the number of partitions of the descriptor vector (number of regions). The quantization error is limited because of the multi assignment approach to vocabulary vectors for each region.

2.3. Hierarchical representation

We use the 2 previously presented descriptor quantized value to create a hierarchical indexing structure. We first cluster all descriptors using the first quantized value (fine index). A new cluster is created if a quantized descriptor value from the dataset does not match any other clusters using the comparison function with the fine quantized value. The fine quantized value becomes the new cluster representation. We obtain about 4 million clusters. Each descriptor in the dataset is then assigned to one or several clusters. The clusters are themselves indexed using their coarse quantized value (coarse index). For descriptor similarity retrieval, we parse coarse indexes to select relevant cluster fine indexes. We then parse the selected fine indexes to select a cluster. All quantized values from the selected cluster are then compared with the query value.

2.4. Key-frame link extraction

We extract groups of local matches between near duplicate key-frames. We use the concept of Bag of visterms (BOV) for representing each key-frame where a visterm is a set of local descriptors participating in a local match. We then apply a generative probabilistic model to extract groups of local matches that represent a common structure representative of 2 near duplicate key-frames. We use the Latent Dirichlet Allocation (LDA) generative model [22], which is a new model derived from pLSA [10], to provide a discrete

discriminant analysis over matches. The significant extracted visterm distributions are seen as part of latent topics which are, in fact, typical structural elements of a key-frame. Latent topics are used as context values for visterms. A group of local matches (visterms) sharing the same latent topic constitutes a topic link across images. See [19],[11] and [12] for more details.

2.5. Descriptor filtering

In order to accelerate the linking process, we need to deal with the fewest possible number of local descriptors. One idea consists of eliminating the more common local descriptors which are not discriminative enough. For instance, local descriptors corresponding to straight lines or corners can be found in many images. This type of local descriptor is not specific enough to accurately describe an image. We apply the BOW generative models [22], over the quantized local descriptors from key frame test dataset. This provides a discrete discriminant analysis over the quantized local descriptors. It eliminates about 40% of non relevant local descriptors from the test dataset.

2.6. Post-Processing

Copied video segments are detected once links are extracted between the query and the reference key-frames. We apply RANSAC in the temporal domain in order to estimate the time shift and dilation between the times codes of the detected links. This step ensures that detected links are forming a coherent segment in time up to a translation and scaling factors. Finally, the shot boundaries from which the selected near-duplicate key-frame belongs to, define the time range of the near duplicate video segment. The confidence value is calculated from the number of local matches first extracted by the probabilistic latent space model and then selected by the video copy segment RANSAC estimation.

3. AUDIO COPY DETECTION SYSTEM

3.1. AUDIO Copy Detection System Overview

The overall system shown in Fig. 5 first computes the audio fingerprints of the audio query. We tried two different audio fingerprints. One fingerprinting method is based on energy differences in consecutive sub-bands [1] [13], and results in a very fast search giving good results. The other fingerprints are based on classification of each frame of the test to the nearest frame of the query[1]. These features result in even better performance. These features are slow to compute, but can be speeded up by parallel processing on a GPU.

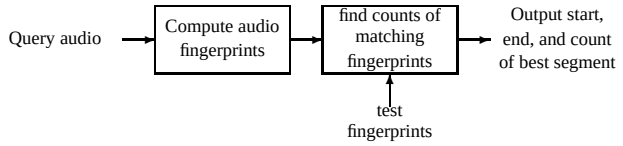

Fig. 5. *Audio copy detection algorithm using fingerprints.*

We use these fingerprints to find test segments that may be copies of the queries. Fingerprint matching is done by moving the query over the test and counting the total fingerprint matches for each alignment of the query with the test. One such alignment is shown in Fig. 6. In this alignment, the matching test segment is identified by the matching start frame (frame 4), the last matching frame (frame 8), and the number of fingerprint matches (3 matches). If we have 100 frames/sec, then the count/sec will be 3 * 100/(8-4+1) = 60. The best matching segment is the segment with the highest count. This is similar to the scoring used in [1].

Since the same query is matched against all the test segments, the total count is a good measure of match between the query and the test segment. However, when comparing matches for different queries, count/sec is more relevant, since the queries vary in duration from 3 secs to 3 minutes.

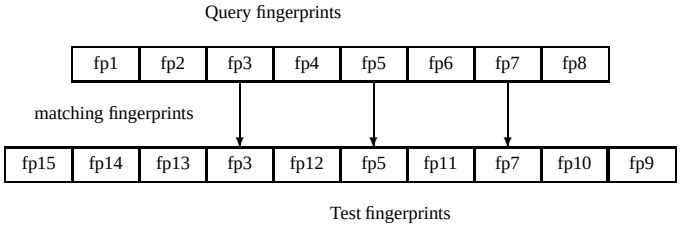

Fig. 6. *One example of matching query to a test.*

3.2. Feature Parameters for Audio Copy Detection

We experimented with two different feature parameters. The first feature corresponds to the audio fingerprint used in music search and other copy detection tasks [1] [13]. The fingerprint we used is similar to that used in [1]. These fingerprints have 15 bits/frame: The audio signal is lowpass filtered to 4 KHz and divided into 25 ms windows with 10 ms frame advance. A pre-emphasis of 0.97 is applied and then multiplied by a Hamming window before computing the Fourier transform. The spectrum between 300Hz and 3000 Hz is divided into 16 bands using mel-scale. A triangular window is then used to compute energy in each band. The energy differences between the sub-bands are used to compute the fingerprint. If $EB(n,m)$ represents the energy value of the n^{th} frame at the m^{th} sub-band, then the m^{th} bit $F(n,m)$ of the 15-bit fingerprint is given by

$$F(n,m) = 1,\ if\ EB(n,m) - EB(n,m+1) > 0,$$
$$Otherwise,\ F(n,m) = 0.$$

In the original formulation, Haitsma and Kalker [13] use 32 bits generated from consecutive sub-band and consecutive frame differences. Using 15 bits from one frame is more robust to bandwidth limitations and extraneous speech addition. With 15 bits, we see more frequent repetition of the fingerprints even for the transformed audio. We call this feature as *energy difference fingerprint*.

To search for a test segment that matches a query, we hash the fingerprints of the query. For example, if the fingerprint for frame k of the query is fp, then $hash(fp) = k$. For each frame j of the test, we keep a count $c(j)$ of query frame matches when the first frame of the query starts at frame j of the test. If the test frame t has a fingerprint $fp1$, then the count $c(t - hash(fp1))$ is incremented when $hash(fp1)$ exists. At the same time, we also update the first and the last matching test frames for query starting at test frame $t - hash(fp1)$. Since more than one frame can have the fingerprint $fp1$, $hash(fp1)$ can have multiple values, and therefore all the counts $c(t - hash(fp1))$ are updated. The maximum count $c(t_1)$ for some test frame t_1 and the corresponding start and end test frames gives the best matching test segment. As we can see, there

[1]These features are not strictly fingerprints, as their value changes when we change the query.

are only three operations involved per test frame. So, the computing to search for the best test segment that matches the query is trivial.

Note that, we search for a segment in the test that matches the query. Since the query is fixed, the count of number of fingerprint matches in a segment is a good measure of the match. However, when we want to use a threshold across many queries, then a better measure is the count/sec. The reason is simple. Query duration varies from 3 secs to several minutes. Therefore, the distribution of matching fingerprint counts for test segments will be very different when the query lengths differ. Using counts/sec across queries normalizes the counts and leads to fewer false alarms and higher recall rate. We compute a counts/sec threshold that gives minimal NDCR.

The second feature parameter maps each frame of the test to the closest frame of the query. For computing this measure of closeness, we compute 12 cepstral coefficients and normalized energy and its delta coefficients. The distance between the test frame and a frame of the query is defined as the sum of the absolute difference between the corresponding cepstral parameters. If $q_1, ..., q_n$ are the query cepstral parameters for a frame and $t_1, ..., t_n$ are the cepstral parameters for a test frame, then this distance is computed as $\sum_{i=1}^{n} |q_i - t_i|$. The test frame gets the frame number of the query frame closest to this test frame. We call this feature as *NN-based fingerprint*.

Computing the closest query frame for each test frame is compute intensive. Two possible alternatives can reduce the computing. One is to organise the frames in a binary tree and traverse it to find the best frame. The other choice is to use a graphics processing unit (GPU). GPU's are cheap, a GPU with 1 Gbyte of memory and 240 processors costs less than $500. The query, and the frames of the test can be transferred to the GPU and the calculation done in parallel in the GPU. Our implementation on a GPU has reduced the overall computing by a factor of over 100.

The search for the test segment that matches the query is trivial. As before, we keep a count $c(i)$ for each frame i of test as a possible starting point for the query. Assume that for each test frame i, $m(i)$ is the query frame that is closest to the test frame i. Then for each test frame i, we increment the count $c(i - m(i))$. We also update the starting test frame, and the last test frame corresponding to frame $(i - m(i))$. The count $c(j)$ then corresponds to the number of matching frames between the test and the query if the query started at frame j. The frame j with the highest count $c(j)$ and the corresponding start and end matching frames is the best matching segment.

4. AUDIO+VIDEO COPY DETECTION

In 2008, there were 10 transformations per video query, and 7 transformations per audio query. This resulted in a total of 70 transformations per query, or there were 70*201 total audio+video queries. One of the queries (query 166) had two test segments, so we removed this query. We ran fusion on these 70*200 a+v queries from 2008 in order to tune our algorithms. For merging audio+video queries, we used the audio submission of EnNN2pass. We merged the corresponding audio and video query into one if the test segments overlapped. Since we have more confidence in audio segmentation, we took the start and end of test segment from the audio query. The overall score was weighted addition of the two scores. If the test segments for video and audio did not overlap, then we only output the test segment with the highest weighted score. Only one test segment per audio+video query was output. We kept the weight of audio score as 1, while we varied the weight of the video score from 0,1,2,3,4 in order to find the optimal weight. We estimated the average min_NDCR over all the 70 transformations. Table 1 shows the average min_NDCR as we vary the weighting for the video score. From the Table, we can see that a weight of 2 is optimal with min_NDCR of 0.014. We also tried keeping the weight of video query when the test segments do not overlap as 2, and increasing the weight when the two test segments overlap. However, this strategy did not improve the min_NDCR. When we get one threshold for all the transformations, the min_NDCR increases to 0.015.

Table 1. *min_NDCR (averaged over 70 transformations) for NOFA case for fusion of audio+video queries from 2008 as we vary the weight of video score from zero to 4.*

Weight	0	1	2	3	4
minimal NDCR	0.017	0.016	0.014	0.016	0.017

The optimal weight for video for all four audio+video submissions turns out to be 2 in each case. For one threshold for all transforms, the min_NDCR when combined with NN22para is 0.011, when combined with EnNN22wt15 (fusion of energy diff and nearest-neighbor fingerprints), it is 0.008.

5. COPY DETECTION RESULTS

5.1. Dataset for Copy Detection

The test data for copy detection comes from NIST sponsored TRECVID 2008 and 2009 competitions [18] [23] [24]. All together, we have 385 hours of video + audio.

The queries for the 2009 submission also come from 201 original queries that are different from the 2008 queries. For 2009, only seven video transforms are used, for a total of 1407 video queries. For audio only also, there are seven transforms for a total of 1407 audio only queries. The audio+video queries are a combination of all the audio and video transforms, and therefore there are 201*7*7= 9849 queries. The seven video transformations for 2009 are shown in table 2. The seven audio transformations for 2008 and 2009 are shown in table 3.

Table 2. *Query video transformations used in TRECVID 2009.*

Transform	Description
T1	original video is inserted in font of background video
T2	insertions of pattern
T3	strong reencoding
T4	change of gamma
T5	decrease in quality
T6	3 post production transformations
T7	3 random transformations

For audio copy detection, the system was developed using audio queries from TRECVID 2008. These are 1407 queries (201 queries * 7 transforms). Query 166 occurred twice in the test, so it was removed from the development set.

5.2. Video only 2009 query results

During the final experimentation on the test database for the 2009 video copy detection tasks, we found that reading indexing data from the disk took 90% of the total processing time. We truncated our test database drastically in order to resolve this problem. We discarded all clusters with more than 150 descriptor value. This removed 60%

Table 3. *Query audio transformations used in TRECVID 2008/2009.*

Transform	Description
T1	nothing
T2	mp3 compression
T3	mp3 compression and multiband companding
T4	bandwidth limit and single-band companding
T5	mix with speech
T6	mix with speech, then multiband compress
T7	bandpass filter, mix with speech, compress

of the test database. Big clusters represent less specific descriptors. However, it results in an accurate precision detection algorithm (low false alarm) while the recall rate is between 30% and 75%. We noticed that about 30% of our submitted detection (see table 4) was considered as false alarm while the visual content was actually near duplicated. Indeed, we may have found redundant visual content at a different place in a movie. Redundant visual segments are most often grouped together on the timeline. Therefore, we generate new results (see table 5) with larger copy detected video segment (80 seconds were added before and after the initially detected copied segment).

Table 4. *Video only submitted detection results.*

Transform	1	2	3	4	5	6	7
N. queries	134	134	134	134	134	134	134
Miss rate	0.48	0.4	0.61	0.43	0.28	0.7	0.71
FA count	17	15	21	13	14	10	11
Mean F1	0.72	0.73	0.61	0.64	0.68	0.63	0.61
Mean time(s)	1374	796	989	765	780	1123	1136
Opt NDCR B	0.84	0.87	0.85	0.89	0.69	0.83	0.96
Opt NDCR NF	0.84	0.87	0.90	0.89	0.69	0.98	0.96

Table 5. *Video only detection results with larger copy detected video segment.*

Transform	1	2	3	4	5	6	7
N. queries	134	134	134	134	134	134	134
Miss rate	0.44	0.35	0.58	0.39	0.24	0.68	0.7
FA count	11	8	17	7	8	8	9
Mean F1	0.29	0.27	0.3	0.28	0.28	0.3	0.3
Mean time	1374	796	989	765	780	1123	1136

Our detection performance of our submitted result is slightly better than the median performance.

5.3. Audio only development on 2008 queries and results on 2009 queries

The audio only copy detection system was developed on 2008 queries. We give detailed performance figures and rationale for the four submissions for the 2009 queries. Basically, we experimented with the energy difference fingerprints and the NN-based fingerprints and their combinations.

5.3.1. Energy difference fingerprint

The copy detection using Energy difference fingerprints was run on 1400 queries from 2008 and 385 hours of test audio from TRECVID. The results were compiled for no FA case (Rtarget = 0.5/hr, CMiss = 1, CFA = 1000). We calculated the no FA result separately for each transform. We also give results when we use one threshold for all the transforms. This is the case in real life, where we do not know the transformation the query has gone through. (Also, this is the threshold we need to provide in our submission).

Table 6. *Minimal NDCR for no FA for energy diff fingerprints with one optimal threshold per transform for 2008 queries.*

Transform	1	2	3	4	5	6	7
min NDCR	.007	.007	.030	.022	.060	.053	.053

For no FA case, results for each transform are given in Table 6, where the decision threshold for each transform is computed separately. The first four transforms do not have any extraneous speech added, while the last three add extraneous speech to the query. For the first two transforms, the number of missed test segments are less than 1%. Even for transforms with extraneous speech added, the worst result is 6% missed segments. In no FA case, the minimal normalized detection cost rate (NDCR) corresponds to a threshold with no false alarms: all the errors are due to missed test segments corresponding to the queries. Table 7 shows minimal NDCR when we have one threshold for all the transforms. In this case the min NDCR more than doubles for the last three transforms.

Table 7. *Minimal NDCR for no FA for energy diff fingerprints with one optimal threshold for all transforms for 2008 queries.*

Transform	1	2	3	4	5	6	7
min NDCR	.015	.037	.037	.022	.127	.135	.165

Let us look at the distribution of counts for the matching test segments. For energy difference fingerprints, we only keep segments with counts greater than 30. Table 8 shows total number of test segments that match the queries with a given count. Over 350,000 test segments have a matching count of 35. However, if we reject test segments with counts less than 36, the minimal NDCR goes up significantly. This means that a significant number of correctly matching test segments have counts below 36. The counts for matching segments vary between 32 and 2300. The counts are consistent: the correct segment has higher count than the incorrect segments. However, across queries, these counts cannot be used to get good discrimination. For discrimination across queries, we use counts/sec.

Table 8. *Segments with matching counts N for the 1400 queries.*

count N	31	35	45	55	75	100
segments	738464	354898	133572	74480	16492	1796

The average query processing time for the energy difference fingerprints is 15 secs on Intel Core 2 quad 2.66GHz processor (we only use one processor). For searching through 385 hours of audio, this search speed is very fast.

Table 9. *Minimal NDCR for no FA for NN-based fingerprints with one optimal threshold per transform for 2008 queries.*

Transform	1	2	3	4	5	6	7
min NDCR	0.007	0	0.007	0.007	0.022	0	0.03

5.3.2. NN-based fingerprint

The copy detection using NN-based fingerprints was run on the same 2008 queries and 385 hours of test data. The results in Table 9 for one optimized threshold per transform are better than those in Table 6 for the energy difference fingerprints. Results for one threshold across all transforms are shown in first row of Table 10. These results are nearly the same as those for one threshold per transform, except for a small increase in min NDCR for transforms 3 and 4. One surprising result is that we do not miss any segments for transform 6 even though extraneous speech has been added to the queries with this transformation.

Table 10. *Minimal NDCR for no FA for NN-based fingerprints with one optimal threshold for all transforms. second row shows rescoring of energy diff results with NN-based features*

Transform	1	2	3	4	5	6	7
NN-based	.007	0	.015	.015	.022	0	.03
NN-based rescore	.007	0	.007	.007	.037	.03	.03

The computing for finding the query frame closest to the test frame is significantly higher than that for the energy difference fingerprint. To reduce computing, we programmed it in a GPU with 240 processors and 1 Gbyte of memory. The nearest neighbor computation lends itself easily to parallelization. The resulting average compute time per query is 360 seconds when we use 22 features (12 cepstral features + normalized energy + 9 delta cepstra). Even though these parameters are very accurate, they are much slower to compute than the energy difference parameters. As we reduce the number of features used to compute the nearest query frame, the results get worse. Table 11 gives the minimal NDCR for 13 features (12 cepstral features + normalized energy).

Table 11. *Minimal NDCR for no FA for NN-based fingerprint with one optimal threshold per transform, using 13 cepstral parameters.*

Transform	1	2	3	4	5	6	7
min NDCR	.007	0	.022	.022	.022	.007	.03

We can reduce the computing time by just rescoring the results from energy difference parameters with the NN-based features. Rescoring lowers average CPU time/query to 20 secs. Min NDCR is shown in the second row of Table 10. Compared to energy difference feature, min NDCR has reduced significantly.

Table 12 shows total number of test segments that match one of the queries and have a given count. The number of test segment matches with a given count drops dramatically with increasing counts. The count threshold for no FA is 23. Above 23, there are no false-alarm segments. Using counts/sec instead of counts does not reduce minimal NDCR. Counts itself are a good measure of copy detection, even across queries of different lengths. So the NN-based fingerprints generate very few false alarms, and the boundary between false alarms and correct detection is well marked.

Table 12. *Segments with matching counts N for the 1400 queries.*

count N	11	20	25	30	35	40
segments	12147	71	61	22	36	28

Since rescoring energy-difference fingerprints with NN-based fingerprints results in very fast compute times (20 secs/query) and low NDCR, we submitted one run for nofa (CRIM.a.nofa.EnNN2pass) and one for the balanced case (CRIM.a.balanced.EnNN2pass) using this rescoring. The only difference was the threshold: for nofa, the threshold corresponds to the score for correct detection just above the highest score for any false alarm. For balanced case, the threshold corresponds to highest score for any false alarm. Table 13 shows the results for 2009 queries. The results show optimal NDCR and actual NDCR using the thresholds from 2008 queries. First, the results for nofa and for balanced case are exactly the same. Second, the optimal and actual min NDCR are the same, except for a small difference for transforms three and six. The mean processing time is 20.5 secs. It turns out that these results are close to the best results for both computing speed and min NDCR.

Table 13. *optimal and actual NDCR for no FA and balanced cases for Energy-based fingerprints rescored with NN-based fingerprints for 2009 queries*

Transform	1	2	3	4	5	6	7
mean proc time	20.4	20.3	20.3	20.5	20.9	21.2	21
mean F1	.921	.936	.924	.89	.92	.90	.90
opt min NDCR	.052	.06	.067	.06	.06	.075	.082
actual min NDCR	.052	.06	.075	.06	.06	.09	.082

Since the results for NN-based feature search are the best and most reliable, we submitted one nofa submission using NN-based features computed using 22-cepstral features. Table 14 shows results for this case. Compared to the EnNN2pass submission, these results are slightly better for many transforms. However, the overall computing has gone up from 20.5 secs/query to 376 secs/query.

Table 14. *optimal and actual NDCR for no FA for copy detection with NN-based fingerprints for 2009 queries*

Transform	1	2	3	4	5	6	7
mean proc time	376	376	376	376	376	375	376
mean F1	.921	.93	.92	.89	.925	.88	.90
opt min NDCR	.052	.052	.067	.06	.052	.067	.075
actual min NDCR	.052	.06	.075	.067	.052	0.075	.082

5.3.3. Fusion of Energy difference and NN-based fingerprints

We fused the two results by combining the counts/sec from Energy diff fingerprint with counts from NN-based fingerprints. We multiplied by 15 the counts/sec to achieve a proper balance. For segments common in the two fingerprints (same query, overlapping test segment), we added the weighted scores and output the segment corresponding to the NN-based fingerprints. For segments not in common, we output the weighted score for the segment. The results for no FA case for 2008 queries are shown in Table 15. The results for

Table 15. *Minimal NDCR for fused results from the two fingerprints for no FA case (separate threshold per transformation).*

Transform	1	2	3	4	5	6	7
min NDCR	.007	0	.007	0	.022	0	.015

Table 16. *Minimal NDCR for fused results from the two fingerprints for no FA case (one threshold for all the transformations).*

Transform	1	2	3	4	5	6	7
min NDCR	.007	0	.007	0	.022	0	.022

no FA with just one threshold across all transformations is shown in Table 16. When we compare Tables 10 and 16, we see significant reduction in min NDCR due to fusion. If we average across all transformations, the min NDCR goes down from 0.016 to 0.008. Table 17 compares this averaged minimal NDCR for energy difference fingerprints versus NN-based fingerprints versus the fused results for 2008 queries. Note that rescoring results from energy diff features with NN-based features results in only a small increase in computing while reducing min NDCR from 0.077 to 0.017.

Table 17. *Comparison of averaged min NDCR across all transforms for different fingerprints when using one threshold for all transforms for 2008 queries.*

Method	minimal NDCR	avg CPU time
energy diff fingerprints	0.077	15 sec
energy diff + NN-based 2nd pass	0.017	20 sec
NN-based fingerprints	0.016	360 sec
fused results	0.008	375 sec

We also gave a submission using this fusion for the balanced case for 2009 queries. The results are shown in Table 18. The results are good except for the actual results for the transform seven. The compute time per query is 390 secs.

Table 19 summarizes the results for the four submissions for 2009 audio queries. For min and actual NDCR, we average the NDCR across all transformations in order to see relative advantage of each algorithm. The optimal min NDCR keeps going down with the improved algorithms. However, the actual min NDCR goes up for the fused results due to transform 7. This was due to one false alarm that was above the given threshold. This was brought about by the energy diff parameter. This was the primary reason for not submitting any runs with energy diff parameter alone, even though they are the fastest to compute.

5.4. Audio+video 2009 query results

We computed the audio+video query results as described in Section 4. We gave four audio+video submissions corresponding to the four audio only submissions. The results (optimal min NDCR and actual min NDCR averaged across all 49 transformations) are shown in Table 20. These results correspond to the following submissions:

- CRIM.m.NOFA.EnNN2pass: fuse 2009 video submission (weight 2) with audio only submission EnNN2pass.
- CRIM.m.NOFA.NN22para: fuse 2009 video submission (weight 2) with audio only submission NN22para.

Table 18. *optimal and actual NDCR for balanced case for copy detection with fusion of energy difference and NN-based fingerprints for 2009 queries*

Transform	1	2	3	4	5	6	7
mean proc time	390	389	389	389	390	389	390
mean F1	.921	.93	.92	.88	.925	.88	.90
opt min NDCR	.052	.052	.06	.052	.052	.052	.082
actual min NDCR	.052	.052	.06	.06	.052	.075	.137

Table 19. *Comparison of averaged min NDCR across all transforms for the different 2009 audio query detection submissions.*

Method Method	opt min NDCR	actual min NDCR	avg CPU time
energy diff + NN-based 2nd pass	0.065	0.068	20.5 sec
NN-based fingerprints	0.0607	0.066	376 sec
fused results	0.057	0.070	390 sec

- CRIM.m.BALANCED.EnNN2pass: fuse 2009 video submission (weight 2) with audio only submission EnNN2pass.
- CRIM.m.BALANCED.EnNN22wt15: fuse 2009 video submission (weight 2) with audio only submission EnNN22wt15.

As can be seen from Table 20, the actual min NDCR is close to the optimal min NDCR except for the first one. All the processing times are high due to the average of 995 sec processing time per video query. Overall, the system NN22para performed well for both audio and audio+video submissions.

Table 20. *Comparison of averaged min NDCR across all transforms for the different 2009 audio+video query detection submissions.*

Method Method	opt min NDCR	actual min NDCR	avg CPU time(sec)
CRIM.m.nofa.EnNN2pass	0.056	1.34	1016
CRIM.m.balanced.EnNN2pass	0.056	0.063	1016
CRIM.m.nofa.NN22para	0.055	0.06	1371
CRIM.m.balanced.EnNN22wt15	0.052	0.058	1385

6. CONCLUSIONS

For visual-based copy detection, approaches based on local descriptor matching are efficient for this task. It is robust to many transformations. However, local descriptor matching is very time consuming and we have to deal with a very big database if we want to maintain high precision. We introduce an efficient SIFT quantizing method and use it to build a hierarchical indexing structure for fast retrieval. However, we could not really take advantage of this approach this year because we encountered many problems while trying to efficiently store and swap from the disk our indexing structure. The probabilistic latent space model over local matches between key-frames allows a fast, robust and accurate filtering process among all possible local matches. This method is better adapted when there is very little common visual information to establish a link between two key-frames. Video copy detection may not need such a good precision. However, our results are close to the median performance

for visual copy detection and we get best results for several transformations when we combine with audio.

We compare copy detection results on audio queries from TRECVID 2008 task using two different audio fingerprints. Fingerprints derived from energy differences in consecutive bands take only 15 seconds/query and give good results. When we compute just one optimized threshold over all queries and average the min NDCR over all transformations, we get a value of 0.077 for no FA, i.e., we miss 7.7% of the test segments that match the queries. For NN-based fingerprints that map each test frame to the nearest query frame, for the same scenario, we get min NDCR of 0.016. In other words, average segment miss rate goes down from 7.7% to 1.6%. However, we need to use a GPU to get reasonable compute times, and the average compute time for one query increases to 360 seconds. However, if we just rescore the energy diff based results with NN-based features, the miss rate goes down from 7.7% to 1.7% while the computing increases from 15 secs to 20 secs. When we fuse the results for the two fingerprints, the min NDCR goes down from 0.016 to 0.008. In other words, the segment miss rate goes down from 1.6% to 0.8% when averaged over all transformations. However, we do not see a similar decrease for 2009 queries. For 2009 queries, the optimal min NDCR goes down from 0.065 (for rescoring with NN-based fingerprints) to 0.057 (fused results). However the actual min NDCR fluctuates around 0.070 due to the difficulty of picking an accurate a priori threshold.

When we combine audio + video queries, the min_NDCR varies between 0.015 and 0.008 for 2008 queries when we estimate one threshold for all the transformations. The best result is when we merge video submission with the fusion of energy-difference and nearest-neighbor fingerprints. For 2009 audio+video queries, the optimal min NDCR averaged over all transformations varied between 0.056 and 0.052. The actual min NDCR averaged over all transformations varies between 1.34 and 0.058. The reason is that it is difficult to come up with an a priori threshold from 2008 queries that will work well for the 2009 queries. The only system that worked well in all scenarios was NN22para where we use NN-based fingerprints for search. For NN-based fingerprints, the thresholds are more stable, resulting in low min NDCR for both 2008 and 2009 queries in all scenarios.

7. REFERENCES

[1] A. Saracoglu, E. Esen, T. Ates, B. Acar, U. Zubari, E. Ozan, E. Özalp, A. Alatan, T. Çiloglu, "Content based copy detection with coarse audio-visual fingerprints", cbmi 2009, 213–218.

[2] J. Law-To, L.Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, F. Stentiford, Video Copy Detection: a Comparative Study, Proc. 6th ACM international conf. Image and video retrieval, p 573–580, July 09–11, 2007, Amsterdam.

[3] W. Zhao, C.-W. Ngo, Hung-Khoon Tan and Xiao Wu, Near-Duplicate Keyframe Identification with Interest Point Matching and Pattern Learning. IEEE Trans. Multimedia 9(5): 1037-1048 (2007)

[4] N. Gengembre, S.-A Berrani. "The Orange Labs Real Time Video Copy Detection System - TrecVid 2008 Results" In TRECVID 2008 Gaithersburg, MD.

[5] M. Douze, A. Gaidon, H. Jegou, M. Marszalek and C. Schmid, "INRIA-LEAR's video copy detection system" Proc. TRECVID Workshop, November, 2008

[6] Y. Ke and R. Suthankar, Efficient near-duplicate detection and sub-image retrieval, ACM Multimedia Conf, 2004, pp. 869-876

[7] J. Yao and W.K Cham. Robust multi-view feature matching from multiple unordered views. Pattern Recog., Vol. 40, Issue 11 pp. 3081-3099, 2007.

[8] J. Sivic and A. Zisserman, Video data mining using configurations of viewpoint invariant regions, CVPR, 2004.

[9] J. Sivic and A. Zisserman, Video google: A text retrieval approach to object matching in videos, ICCV, 2003,

[10] T. Hofmann, Probabilistic Latent Semantic Indexing, SIGIR, 1999.

[11] M. Héritier, L. Gagnon and S.Foucher. "Places Clustering in Videos via Latent Aspects Modeling of SIFT Matches" IEEE Trans. Circuits and Systems for Video Technology, 19 (6) June 2009 : 832-841.

[12] M. Héritier, S.Foucher and L. Gagnon. "Key-Places Detection and Clustering in Movies Using Latent Aspects" In Proc. 14th IEEE International Conf. Image Processing (ICIP 2007), 2, pp. II-225 - II-228. San Antonio, TX.

[13] J. Haitsma, T. Kalker, "A highly robust audio fingerprinting system", [online] ismir2002.ismir.net/proceedings/02-FP04-2.pdf

[14] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification", Proc. Comp. Vision Pattern Recog., 2005.

[15] M. Covell, S. Baluja, and M. Fink, "Advertisement Detection and Replacement using Acoustic and Visual Repetition", IEEE Workshop multimedia sig. proc., Oct. 2006, pp. 461–466.

[16] P. Duygulu, M. Chen, and A. Hauptmann, "Comparison and combination of two novel commercial detection methods", Proc. ICME, 2004, pp. 1267–1270.

[17] V. Gupta, G. Boulianne, P. Kenny, and P. Dumouchel, "Advertisement Detection in French Broadcast News using Acoustic repetition and Gaussian Mixture Models", Proc. InterSpeech 2008, Brisbane, Australia.

[18] "Final CBCD Evaluation Plan TRECVID 2008", June 3, 2008, [online] www-nlpir.nist.gov/projects/tv2008/Evaluation-cbcd-v1.3.htm.

[19] M. Héritier, S. Foucher and L. Gagnon. "Video Copy Detection Using Latent Aspect Modeling Over SIFT Matches" In TRECVID 2008, pp. 8. Gaithersburg, MD.

[20] S. Porter, M. Mirmehdi and B. Thomas, Video indexing using motion estimation, ECCV, 2006.

[21] D. G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV, 2004

[22] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptor, IEEE Trans. PAMI, Vol. 27, pp. 1615-1630, 2005.

[23] W. Kraaij, G. Awad, and P. Over, "TRECVID-2008 Content-based Copy Detection", [online]. www-nlpir.nist.gov/projects/tvpubs/tv8.slides/CBCD.slides.pdf.

[24] A. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid". In Proc. 8th ACM International Workshop Multimedia Information Retrieval (Santa Barbara, California). MIR '06. ACM Press, New York. http://doi.acm.org/10.1145/1178677.1178722

[25] D. Blei, A. Ng and M. Jordan. Latent Dirichlet allocation, Journ. Machine Learning Res., Vol. 3, pp. 993-1022, 2003