CONTENT-BASED VIDEO COPY DETECTION USING NEAREST-NEIGHBOR MAPPING

Vishwa Gupta, Parisa Darvish Zadeh Varcheie, Langis Gagnon, Gilles Boulianne*

Centre de recherche informatique de Montréal (CRIM)
{Vishwa.Gupta, Parisa.Darvish, langis.gagnon, gilles.boulianne}@crim.ca

ABSTRACT

We report results on video copy detection using nearest-neighbor (NN) mapping that has been used successfully in audio copy detection. For copy detection search, we use a sliding window to move the query video over the test video, and count the number of frames of query that match the frames in the test segment. The feature in the test frame that we match is the frame number of the query that is closest to that test frame. This leads to good matching scores even when the query video is distorted and contains occlusions. We test the NN mapping algorithm and the video features that map test frame to the closest query frame on TRECVID 2009 and 2010 content-based copy detection (CBCD) evaluation data. For both these tasks, the NN mapping for video copy detection gives minimal normalized detection cost rate (min NDCR) comparable to that achieved with audio copy detection for the same task. For the TRECVID 2011 CBCD evaluation data we got the lowest min NDCR for 26 out of 56 transforms for actual no false alarm case.

1. INTRODUCTION

There are many applications of video copy detection: for copyright control, for monitoring advertisement campaigns of businesses, for monitoring ads of competitors for business intelligence, and for law enforcement investigations. Content-based video copy detection offers an alternative to watermarking. In watermarking, only the content that has been watermarked can be detected, while content-based copy detection can detect any copy for which there is a copy of the video in the search database.

The content-based video copy detection got a big boost with the TRECVID-2008 CBCD (content-based copy detection) evaluation [1], and has continued with TRECVID 2009, 2010, and 2011 evaluations. Many research labs have participated in these CBCD evaluations with many different algorithms for copy detection. The copy detection performance has improved significantly in the last 3 years. In TRECVID 2008, the emphasis was on copy detection with only a small penalty for false alarms. In TRECVID 2009 [2], the emphasis shifted to no FA (false alarms) case, with a penalty of 1000 for each false alarm. The no FA case was divided into two: optimal and actual. In the optimal case, a separate threshold for rejection per transform (see Sec. 5 for the list of transforms) is computed to minimize NDCR (normalized detection cost rate) [3]. In the actual case, one threshold for all the transforms is specified *a priori* by the participants. This threshold is then used to estimate the min NDCR for all the transforms. So the actual case is much more difficult than the optimal case. Also, it is closer to an actual application where the threshold has to be set a priori.

TRECVID 2009 is the last time when NIST evaluated audio, video, and audio+video copy detection separately. For this evaluation, CRIM achieved the lowest min NDCR for audio only copy detection [2] for all categories (optimal balanced, actual balanced, optimal no FA, actual no FA) and for all the seven transformations of the audio queries. Most of our results were around min NDCR of 0.06. This was primarily due to the nearest-neighbor mapping [4] that was used to map test frames to the nearest query frames. For TRECVID 2009 *Video only* copy detection, ATT labs [2][5] got the lowest NDCR for *optimal no FA* runs. Their min NDCR varied between 0.22 to 0.68[1]. The best *actual no FA* results were shared by three different labs (for different transforms) [2], and the min NDCR varied between 0.22 and 0.69. Even though CRIM got only average results in *video only* copy detection evaluations for TRECVID 2009, we still got the lowest min NDCR for *audio+video* evaluations for *actual balanced* and *actual no FA* runs for all the 49 transforms [2]. This was primarily due to the *audio only* copy detection results.

Because nearest-neighbor mapping gave such good results for audio copy detection, we decided to implement it for video copy detection also. For video copy detection, the idea is to map each frame of the test video to the nearest frame of the query video. We then move the query over the test and find the best matching test segment, i.e. the test segment with the highest number of frames that match the overlaid query. The frame match between the test frame and the query frame is by comparing the query frame number with the query frame number that is closest to the test frame we are matching.

The first step in video copy detection using nearest-neighbor mapping is to choose frame-based local video features that are suitable for search using nearest-neighbor

*Parisa Darvish is now with Genetec Inc, Montreal, pdarvish@genetec.com

[1] min NDCR represents weighted average of missed correct detections and false alarms detections. In no FA case, there are no false alarms, so min NDCR represents the percent of missed queries. So, between 22% and 68% of the queries were missed.

mapping. In 2010 TRECVID CBCD evaluations, Peking University [6] [7] got the lowest min NDCR. They achieved this by using many different detectors over local and global visual features, and then fused the results. It is difficult to pick one feature that stands out. However, NTT [8] also got very good results with only one feature set: temporally normalized local visual features. We used these features as our starting point. We computed seven discrete temporally normalized features per video frame and used these features to search for video copies. Using these features, we show that the search using nearest-neighbor mapping outperforms the search algorithm similar to that used by NTT. We also show that 16 unquantized features per frame perform much better than the seven quantized features with nearest-neighbor mapping. The min NDCR for video copy detection that we achieved for TRECVID 2009 data varies between 0 and 0.097. This is better than the average min NDCR of 0.06 that we achieved for audio copy detection in TRECVID 2009. This is better than the best TRECVID 2009 *video only* copy detection results (with min NDCR varying between 0.22 and 0.68) [2]. We also got good video copy detection results on TRECVID 2010 data. For the TRECVID 2011 CBCD task, we got the lowest min NDCR for 26 out of 56 transforms for the audio+video actual no FA task.

2. VIDEO COPY DETECTION SYSTEM OVERVIEW

The overall system shown in Fig. 1 first computes the video fingerprints of the test video. By video fingerprint of a test frame we mean the query frame number closest to this test frame. So the video fingerprint of a complete video is the sequence of query frame numbers that are closest to the sequence of test video frames. We tried two different video feature parameters to compute these fingerprints. One set of video feature parameters is based on the video features used by NTT for TRECVID 2010 [8]: we compute 16 averaged pixel values per frame per color (RGB). These values then go through local temporal normalization in a window of 10 frames, and the top 7 values (based on maximum deviation from the mean) are then selected for quantization. For each test frame, we then find the closest query frame based on these feature parameters. The metric for comparing the test and the query feature parameters is the sum of the absolute distance between the corresponding features. The query frame number closest to the test frame becomes the fingerprint for the test frame. In the second feature set, we used 16 normalized values per frame without quantization, and we computed the fingerprint for each test frame using these 16 unquantized features. These fingerprints gave us the best results.

We use these fingerprints to find test segments that may be copies of the query. We match the query and test fingerprints by moving the query over the test fingerprints and counting the total fingerprint matches for each alignment of the query with the test. One such alignment is shown in Fig. 2. In this alignment, the matching test segment is identified by the matching start frame (frame 3), the last matching frame (frame 7), and the number of fingerprint matches (3 matches). The total count of matches over all the aligned frames is a measure of confidence in the match. The best matching test segment is the segment with the highest count. We tried both counts and counts/sec as a confidence measure. It turns out that counts work much better than counts/sec. This is similar to our experience with audio copy detection [4].

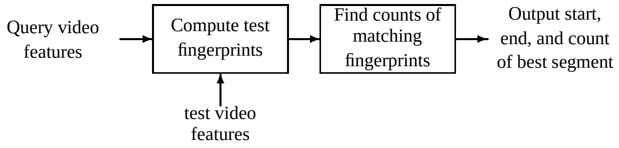
Query video features
Compute test fingerprints
Find counts of matching fingerprints
Output start, end, and count of best segment
test video features

Fig. 1. *Video copy detection algorithm using fingerprints.*

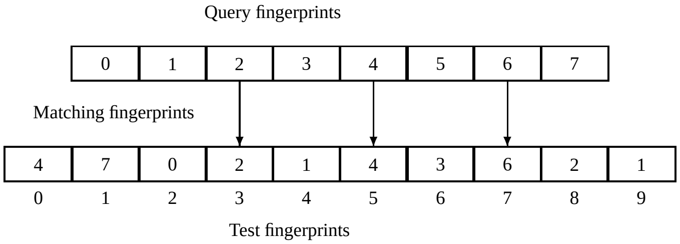

Fig. 2. *One example of matching query to a test.*

3. FEATURE PARAMETERS

As mentioned in Sec. 2, we experimented with two different feature parameters. The first feature parameter is similar to the temporally normalized and quantized features used by NTT [8]. These features are computed as follows: Let $v_c(p,t)$ represent RGB value of a pixel in a video frame at time t, where p = pixel coordinate, $c \in \{R,G,B\}$. We divide the frame into 16 sub-squares and compute raw RGB value $x_c(i,t)$ in each square as

$$x_c(i,t) = \frac{1}{|I_i|}\sum_{p\in I_i} v_c(p,t),$$

where I_i $(i = 1, 2, ..., 16)$ is a whole set of pixels in the i^{th} sub image. $|I_i|$ is the total number of pixels in the i^{th} sub image. The temporally normalized features $y_c(i,t)$ are computed from $x_c(i,t)$ using a 10-frame window as follows:

$$y_c(i,t) = \frac{1}{\sigma_c(i,t)}(x_c(i,t) - \mu_c(i,t)), \quad where$$

$$\mu_c(i,t) = \frac{1}{M}\sum_{j=-[M/2]}^{M-[M/2]-1} x_c(i,t+j), \quad and$$

$$\sigma_c(i,t) = \left(\frac{1}{M}\sum_{j=-[M/2]}^{M-[M/2]-1} (x_c(i,t+j) - \mu_c(i,t))^2\right)^{1/2}$$

are average and standard deviation computed over a time window of M frames.

For each color, we choose 7 features for each frame that have the largest deviation from the temporal mean, that is, we choose seven values of i that have the maximum values for $z_c(i,t)$ where

$$z_c(i,t) = |(x_c(i,t) - \mu_c(i,t))|.$$

Each of these 7 chosen $x_c(i,t)$ values is then quantized between 0 and 5. Each of these values is stored as a (value, position) pair. In other words, there are 21 (value, position) pairs per video frame.

We tried two different algorithms to search for a given query in the test set using these features. One algorithm was similar to that used by NTT [8] where we move the query over the test and count all the matching (value, position) pairs. The reason for counting matching values is that it leads to a very fast search. The test with the highest matching count is considered the best match, and the start and end of matches in the test correspond to the matching test segment. The total matching count of the (value, position) pairs is used as a confidence value. We call this search as *value-position matching*. The results for TRECVID 2009 for transforms 3, 4, and 5 (see Sec. 5) are shown in second row of Table 1. We use only transforms 3, 4, and 5 because they do not contain any flip, shift or picture-in-picture transforms. So we can use the extracted features directly for search.

The second search process we used with these features was the search using nearest-neighbor (NN) mapping (see Sec. 4). In this search, we map each test frame to the closest query frame. To compute the closest query frame, we augment the seven (value, position) pairs with pairs (-1, position) for the 9 missing positions, and then compute the absolute sum S between a test frame and a query frame as

$$S = \sum_{i=0}^{15} |(y_c^{'}(i,t) - q_c^{'}(i,k))|.$$

where $y_c^{'}(i,t)$ is the quantized value of $y_c(i,t)$ in position i for the test frame t, and $q_c^{'}(i,k)$ is the quantized value in position i for the query frame k. We label the test frame as the query frame number k that gives the lowest sum S. The nearest-neighbor k is efficiently computed on a GPU [9]. We then search for the test frame segment that gives the highest matching count (as shown in Fig. 2). Table 1 compares the min NDCR for the nearest-neighbor mapping (row 3) versus the value-position matching search (row 2). The nearest-neighbor mapping outperforms the value-position matching search. The reason is that when the query matches a test segment, the nearest-neighbors will be ordered sequentially leading to a high matching count. When the query does not match a test segment, the nearest neighbors will be random, leading to very small counts. This is similar to our experience with audio copy detection [4].

The second feature set we used was the unquantized features $y_c(i,t)$ for all 16 positions. For search, we used the nearest-neighbor mapping with these unquantized values. The third row in Table 1 shows the min NDCR for these unquantized features. These features gave the best results. So these features were used in the rest of the experiments.

Table 1. *Minimal NDCR for optimal no FA for different feature parameters and search algorithms.*

Transform	3	4	5
value-position matching	.052	.269	.067
NN mapping: discrete features	.007	.082	0.0
NN mapping: unquantized features	0.0	.037	0.0

The query goes through many transforms which affect the position of the feature parameters. For flip transform, we flipped the 16 feature vectors of each frame of the query. This leads to two feature sets per query: flipped and unflipped features. Each feature set is searched independently. Similarly, there were 5 picture-in-picture (PiP) positions (upper left, upper right, lower left, lower right, and center), and for each PiP position, there were three different sizes (0.5, 0.4, 03). This lead to 15 additional different feature sets for each of the flipped and non-flipped positions. So all together, we generate 32 different feature sets per query that are searched independently. We then retain the test segment that gives the highest matching count (using the nearest-neighbor mapping). Because of flip and picture-in-picture transforms, the search is 32 times slower. The time for one search (without flip or PiP) is 69 secs for searching the 400 hours of test video (95% of this time is for computing the closest query frame for each test frame in the GPU). If we can locate PiP transform automatically (by locating the edges using the Hough lines [10]), then we can search in roughly $69 * 4$ secs (for non-PiP, non-PiP flipped, PiP, PiP flipped cases), and $69 * 2$ secs when there is no PiP.

4. SEARCHING VIDEO QUERY IN TEST USING NEAREST-NEIGHBOR MAPPING

In video copy detection using nearest-neighbor (NN) mapping, the idea is to first map each frame in the test video to the closest query frame. We then search for the mapped test segment that matches the query best. The search for the mapped test segment that matches the query best is as follows. Each test frame is labeled as a query frame number corresponding to the query frame closest to the test. For example, in Fig. 3, the number inside each test frame corresponds to the query frame closest to that test frame. Frame 0 of the test matches frame 4 of the query, frame 1 of the test matches frame 1 of the query, ... We keep a count $c(i)$ for each frame i of test as a possible starting point for the query. In other words, count $c(i)$ corresponds to the total number of frames that match when the query is overlaid on top of the test starting with frame i of the test (same as shown in Fig. 2). The count $c(i)$ is computed incrementally as follows. Assume that for each test frame i, $m(i)$ is the query frame closest to the test frame i. Then for each test frame i, we increment the count $c(i - m(i))$ by 1. We also update the starting test frame, and the last test frame corresponding to query overlaid on top of the test starting with 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 was overlaid starting 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. The final matching counts for the search example are shown in the bottom row of Fig. 3. In this example, frame 3 of the test has the highest matching count. These counts are accumulated separately for each color (RGB) and then summed to get the final count.

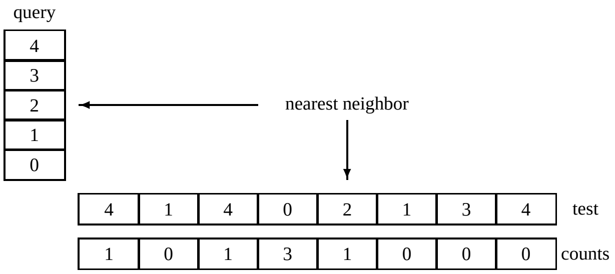

Fig. 3. *One example search using NN fingerprints.*

5. DATASET FOR VIDEO COPY DETECTION

The data for video copy detection for TRECVID 2009 comes from NIST sponsored TRECVID 2008 and 2009 CBCD evaluations [3] [1]. The video queries are from the TRECVID 2009 evaluations. In TRECVID 2009, there were 201 original video queries transformed 7 different ways (Transforms 2, 3, 4, 5, 6, 8, 10 in Table 2). Each original query is supposed to occur one or zero times in the test video. The test set for TRECVID 2009 consists of a total of 385 hours of video from TV broadcasts.

For the 2010 TRECVID copy detection evaluation, the test set consists of roughly 12000 videos from internet archives for a total of 400 hours of video. There are 201 original video queries (different from 2009) transformed 8 different ways (transforms 1 2 3 4 5 6 8 10).

6. VIDEO COPY DETECTION RESULTS

6.1. TRECVID 2009 results

Video copy detection using the 16 floating-point temporally normalized values per frame was run on 1407 video queries and 385 hours of test video from TRECVID 2009 CBCD evaluations. The min NDCR for the optimized no false-alarm case (Rtarget = 0.5/hr, CMiss = 1, CFA = 1000)[3] are shown in Table 3. Note that when we search 32 sets of features (Table 3) instead of one (row 4, Table 1), the min NDCR for transform 4 goes up from 0.037 to 0.052.

Table 2. *Query video transforms used in TRECVID CBCD evaluations.*

Transform	Description
T1	Cam Cording
T2	Picture in picture (PiP) Type 1: original video in front of background video
T3	Insertions of pattern
T4	Strong re-encoding
T5	Change of gamma
T6, T7	Decrease in quality: blur, gamma, frame dropping, contrast, compression, ratio, white noise
T8, T9	Post production transforms: crop, shift, contrast, caption, flip, insertion of pattern, PiP type 2
T10	Combination of everything

Table 3. *Minimal NDCR for optimal no FA for features with 16 unquantized values/frame using nearest-neighbor mapping.*

Transform	2	3	4	5	6	8	10
min NDCR	.022	0	.052	0	0	.037	.097

These results (Table 3) are probably the best published results on video copy detection for TRECVID 2009 [2]. In general, the min NDCR for video copy detection is significantly worse than for audio copy detection, but these results are better than the min NDCR we achieved on audio copy detection for the same task [4]. Our audio copy detection results from this paper are shown in Table 4. When we compare Table 3 with Table 4, we can see that min NDCR for video copy detection is significantly better than that for audio copy detection except for transform 10. The average min NDCR for video copy detection across all transforms is 0.03, while for audio copy detection it is 0.061.

Table 4. *Minimal NDCR for optimal no FA for* **audio** *copy detection using nearest-neighbor mapping.*

Transform	1	2	3	4	5	6	7
min NDCR	.052	.052	.067	.06	.052	.067	.075

6.2. TRECVID 2010 results

The TRECVID 2010 CBCD evaluations test set consists of completely new videos collected from the web. This new set of videos is characterized by a high degree of diversity in creator, content, style, production qualities, orginal collection device/encoding, language, etc - as is common in much of *web video*. In 2009, there were 838 test videos containing 385 hours of video from TV broadcasts. In 2010, there are over 12000 files containing 400 hours of video from internet archives. These videos are in general less than 4.1 minutes in duration. Many of these videos are slide shows with varying durations of each slide.

In compiling the copy detection results, we noticed that there were many duplicate test files for many queries. To compile the results correctly, we removed these duplicate files. The final results shown in Table 5 use features with 16 unquantized values/frame and search using the nearest-neighbor mapping.

Table 5. *Minimal NDCR for 2010 queries for optimal no FA for features with 16 unquantized values/frame using nearest-neighbor mapping.*

Transform	1	2	3	4	5	6	8	10
min NDCR	.6	.42	.04	.18	.03	.14	.19	.27

As we can see from Table 5, the min NDCR for optimal no FA case is significantly worse for 2010 data than for 2009 data. The reason is simple. In 2009 videos, there are no slide shows, while 2010 data has several slide shows. The feature parameters we have used are based on temporal variability. When there is no temporal variability, then the features are either zero or one. This leads to many more false matches. For 2009 data, the largest count for false alarms is 36, while the largest count for false alarms for 2010 data is 51. This affects significantly the picture-in-picture (PiP) transforms. Inherently, PiP transforms show significantly fewer matches than for videos without PiP. With the false alarm threshold going up, all the transforms with PiP (transforms 2, 8 and 10) are adversely affected. Transforms 4 and 6 have lower resolution, and they are similarly adversely affected. Transform 1 is camcording, and the video frames have a lot of jitter, leading to fewer matches and therefore they are also adversely affected by the higher threshold for false alarms.

The optimal no FA results shown in Table 5 use separate rejection threshold for each transform. In reality, we do not know *a priori* which transform is being used. So, in actual case, we can only use one threshold across all transforms. Table 6 gives results for one threshold across all transforms (actual no FA case). For 2009 queries, this threshold was 36, while for 2010 queries, it was 51. We notice that for 2009 queries, except for transform 10, the min NDCR is the same as it was for one optimal threshold per transform. For the 2010 queries, min NDCR has gone up for all transforms except for transform 5. This increase is primarily due to the slide shows, which result in higher threshold for the false alarms. We probably need to use static feature parameters for the slide shows in order to reduce these false alarms.

Table 6. *Minimal NDCR for 2009 and 2010 queries for actual no FA case with one threshold across all transforms.*

transform	1	2	3	4	5	6	8	10
2009		.02	0	.05	0	0	.04	.12
2010	.71	.46	.05	.19	.03	.16	.24	.29

6.3. TRECVID 2011 Audio+Video results

We used the video copy detection algorithms developed above in audio+video copy detection task for TRECVID 2011 CBCD evaluation. In TRECVID 2011 copy detection task [11], only audio+video queries were evaluated. We combined the audio+video submission by first generating separately audio and video submissions with multiple choices, and then combining the two. For audio submission, we used the algorithms outlined in [4]. We improved the audio submission by using three different audio features [12] and fused their results in order to improve audio copy detection. As shown in Table 7 original result, we obtained lowest optimal min NDCR (among all the participating research organisations) for 18 out of 56 transforms for no FA case and 21 out of 56 for the balanced case, and 22 out of 56 for the actual no FA case [13]. For audio copy detection, there was a bug in computing the query audio segment that matches the test audio segment. The revised results shown in Table 7 significantly improve the number of lowest min NDCR across all sites for all cases. The average F1 measure across all transforms also improved from 0.711 to 0.833 (F1 measures how well we locate the matching segments).

Table 7. *Number of transforms with lowest min NDCR across all sites for various copy detection cases: opt min no FA/Balanced, and actual min no FA.*

case	opt min no FA	opt min balanced	actual min no FA
original	18	21	22
revised	25	26	26

The optimal min NDCR for no FA case for each transform is shown in Table 8. In this Table, Transform T1 is (video transform 1, audio transform 1), transform T2 is (video transform 1, audio transform 2), ..., transform T8 is (video transform 2, audio transform 1), and so on. The seven possible audio transforms that an audio query can undergo are shown in Table 9.

7. CONCLUSIONS

We show that nearest-neighbor mapping of test frames to query frames works as well for video copy detection as it did for audio copy detection. We tried two different feature sets with this nearest neighbor mapping. One feature set was similar to that used by NTT [8] where we used 7 discrete features per frame per color. For this feature set, we showed that the nearest-neighbor mapping gave significantly better performance than the search similar to that used by NTT. The other feature set used 16 unquantized features per frame per color. We show that these unquantized features gave significantly lower min NDCR than the discrete features. The unquantized features work better as they map to the correct query frame more frequently. In other words, these unquantized features represent the video frame more accurately even under noisy condition.

Table 8. *Minimal NDCR for audio+video combined results for 2011 audio+video queries for optimal no FA case. The min NDCR in boldface are the best results across all sites.*

T	min ndcr	T	min ndcr	T	min ndcr	T	min ndcr
T1	.299	T15	**.007**	T29	**.007**	T50	**.037**
T2	.299	T16	**.007**	T30	**.007**	T51	**.037**
T3	.291	T17	**.007**	T31	**.007**	T52	**.037**
T4	.284	T18	**.007**	T32	**.007**	T53	**.045**
T5	.336	T19	**.007**	T33	**.007**	T54	**.037**
T6	.328	T20	.007	T34	.007	T55	**.037**
T7	.246	T21	**.007**	T35	.007	T56	**.045**
T8	.172	T22	**.03**	T36	.052	T64	.231
T9	.179	T23	**.03**	T37	.052	T65	.194
T10	.179	T24	**.045**	T38	.052	T66	.187
T11	.179	T25	**.03**	T39	.052	T67	.187
T12	.209	T26	**.03**	T40	.052	T68	.194
T13	.224	T27	**.037**	T41	.06	T69	.142
T14	.306	T28	.045	T42	.06	T70	**.067**

Table 9. *Query audio transformations used in TRECVID 2009/2010/2011.*

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

For the 2009 TRECVID CBCD data, the min NDCR for video copy detection varies between 0 and 0.097 depending on the transform. The averge min NDCR of 0.03 across all video transforms is better than the average min NDCR of 0.061 across all audio transforms that we achieved on audio copy detection for the same data. For the 2010 TRECVID CBCD data, the optimal min NDCR for video copy detection for no FA case varies between 0.03 and 0.7. The results on 2010 data are worse than 2009 data because of many slide shows where the temporal variability is zero for many consecutive frames. For the slide shows, we need to come up with new static features and modified search in order to detect them. In TRECVID 2011 CBCD audio+video evaluation, this NN-mapping algorithm lead to very good results, and we got lowest min NDCR across all sites for 25 out of 56 transforms.

8. REFERENCES

[1] 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.

[2] W. Kraaij, G. Awad, P. Over, “Slides: TRECVID 2009 Content-based Copy Detection Task” 2009, [online] www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2009.

[3] “Guidelines for the TRECVID 2009 Evaluation” 2009, [online] www-nlpir.nist.gov/projects/tv2009/

[4] V. Gupta, G. Boulianne, P. Cardinal, “ CRIM’s content-based audio copy detection system for TRECVID 2009”, Multimedia Tools and Applications, 2010, Springer Netherlands, pp. 1-17, DOI: 10.1007/s11042-010-0608-x.

[5] Z. Liu, B. Shahraray, T. Liu, “AT&T Research at TRECVID 2009 Content-based Copy Detection” 2009, [online] www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2009.

[6] W. Kraaij, G. Awad, “Slides: TRECVID 2010 Content-based Copy Detection Task” 2010, [online] www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[7] Y. Li, L. Mou, M. Jiang, C. Su, X. Fang, M. Qian, Y. Tian, Y. Wang, T. Huang, W. Gao, “PKU-IDM @ TRECVid 2010: Copy Detection with Visual-Audio Feature Fusion and Sequential Pyramid Matching”, [online] www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[8] R. Mukai, T. Kurozumi, K. Hiramatsu, T. Kawanishi, H. Nagano, K. Kashino, “NTT Communications Science Laboratories at TRECVID 2010 Content-Based Copy Detection”, Proc. TRECVID 2010, Gaitersburg, MD, USA.

[9] P. Cardinal, V. Gupta, G. Boulianne, “Content-based Advertisement Detection”, Interspeech 2010, pp. 2214-2217.

[10] O. Orhan, et al, “University of Central Florida at TRECVID 2008 Content Based Copy Detection and Surveillance Event Detection,” Proc. TRECVID 2008, Gaithersburg, MD.

[11] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, and A. Smeaton, “TRECVID 2011 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics”, [online] www-nlpir.nist.gov/projects/tvpubs/tv11.papers/tv11over view.pdf.

[12] S. Foucher, M. Lalonde, V. Gupta, P. Darvish, L. Gagnon, G. Boulianne,, “CRIM Notebook Paper TRECVID 2011 Surveillance Event Detection”, Proc. TRECVID 2011, Gaitersburg, MD, USA.

[13] W. Kraaij, and G. Awad, “TRECVID-2011 Content-based Copy Detection Task”, [online]. www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html