University of Bradford at TRECVID 2008
Content Based Copy Detection Task

J. Chen and J. Jiang
{j.chen12, j.jiang1}@bradford.ac.uk
Department of EIMC, University of Bradford, UK

Abstract.

1. Briefly, what approach or combination of approaches did you test in each of your submitted runs? (please use the run id from the overall results table NIST returns)

   BradfordU_FhG.v.Juan: we present a novel method for spatial-temporal video copy detection based on adaptive masking.

2. What if any significant differences (in terms of what measures) did you find among the runs?

   No.

3. Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?

   Firstly, a dedicated video analysis is implemented for input videos, which ensures the accurate detection of complicated distortions query videos may undergo. Secondly, simple signatures are extracted for the benefit of time and space efficiency, and the frame mask is generated adaptively to reduce video temporal redundancy. Thirdly, a matching process is implemented to find video copies.

4. Overall, what did you learn about runs/approaches and the research question(s) that motivated them?

   The proposed video copy detection framework is effective, and robust against spatial and temporal variations.

1 Introduction

With the advances in high-performance networking and improvements in computing capability, efficient retrieval of multimedia data has become an important issue. Content-based retrieval technologies have been widely implemented to protect intellectual property rights (IPR) [1-5]. Watermarking and content-based copy detection is the main approaches towards the IPR protection. Watermarking inserts the identification of a document prior to distribution, while content-based copy detection searches the extracted signatures in an indexed database [6-9]. The primary advantage of content-based copy detection over watermarking is the fact that copies are detectable without previously embedded mark or existence of original material. Retrieval efficiency is the key issue in the application of multimedia search. Redundancy copies from the search result need to be identified and removed for the useful multimedia browsing. In addition, video copy retrieval is successfully applied in media tracking [10].

The rest of the paper is organized as follows. In section 2, the overview of our system for video copy retrieval is briefly described. Section 3 provides the process of video analysis. Section 4 describes the procedure of signature extraction. Section 5 reports the matching process. Experimental results on the implementation system and discussion are given in section 6. Finally, section 7 provides concluding remarks to finish this paper.

2 System overview

In the whole video copy retrieval process, pixel intensity is chosen among all the visual features, since it is simple and reliable. In addition, color information is not available in some black-white video segments. Our strategy for video copy retrieval is implemented in three major procedures, including video analysis, signature extraction and matching. The flowchart of our system for video copy retrieval is given in figure 1.

The video analysis process includes key frame extraction, template selection, temporal redundancy detection, video in video detection and video flip. Firstly, key frame is extracted to save the computation cost and increase the compression. Then, template selection is carried out to deal with different editing patterns within one video and facilitate the other three operations, which are temporal redundancy detection, video in video detection and video flip.

In the signature extraction process, each key frame is scaled down to be 9×11 and the corresponding frame mask is generated adaptively to reduce the temporal redundancy. For each key frame, the signature consists of 99-dimensional (99-D) pixel intensity in $[0,255]^{D=99}$, which is generated from its scaled down frame, and 99-dimensional (99-D) corresponding frame mask in $[0,1]^{D=99}$.

The matching process is mainly based on sequential match with adaptive sample step.

There is some difference of the procedures between target video and query video. Query video goes through all the procedures mentioned above, while target video does not go through static pixels detection, which is a sub-procedure of temporal redundancy detection, video in video detection and video flip in the video analysis process, all of which are unique transformations for query video. In addition, the frame mask is not generated for target video in the signature extraction process, in order to maintain the consistency for comparison in matching.

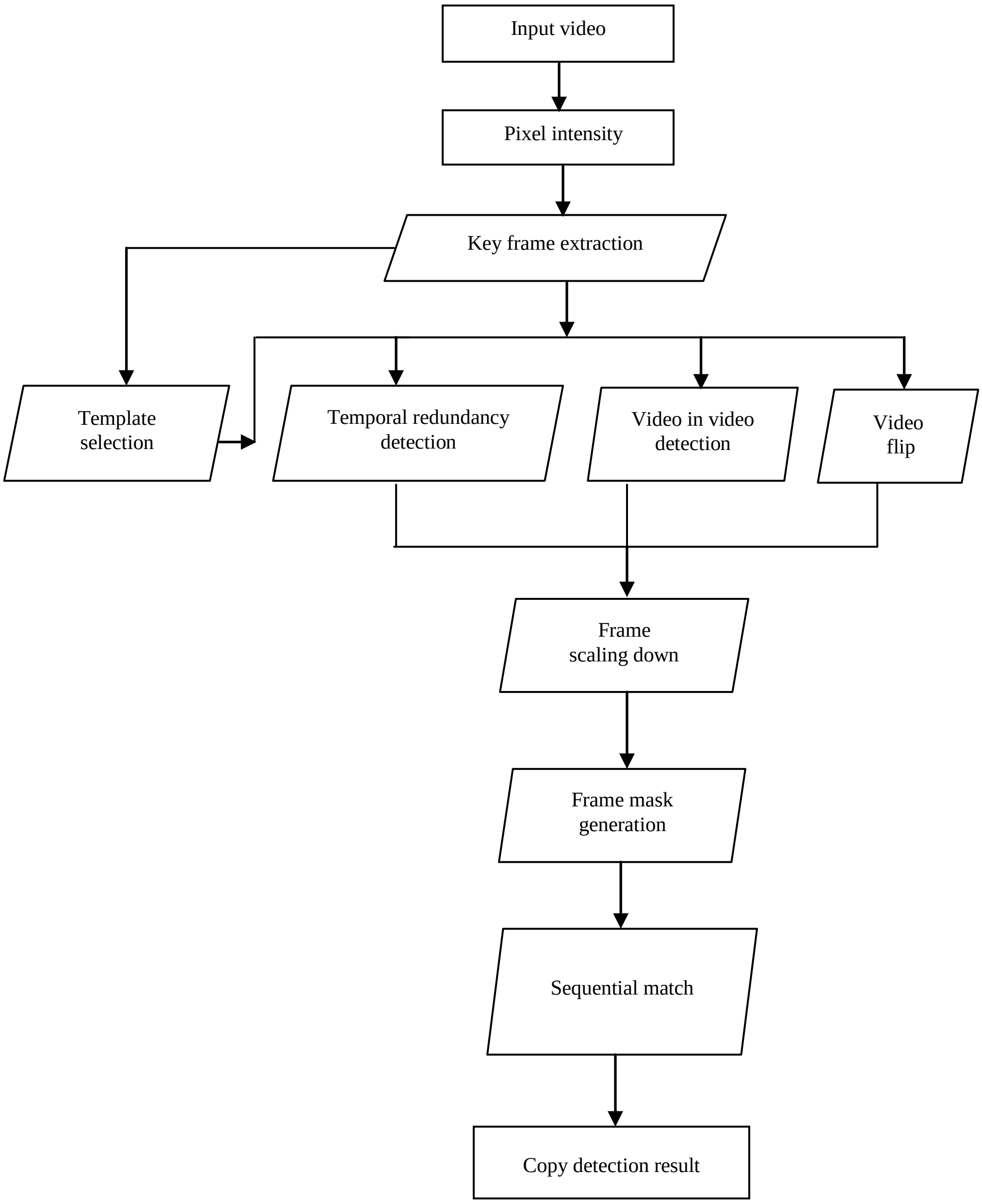

Figure 1 flow chart of video copy retrieval

3 Video analysis

3.1 Key frame selection

In order to handle both color frames and monochrome frames, as well as save computation load, we only choose the pixel intensity as the visual information. Standard deviation of the pixel intensity is used to filter out meaningless monochrome frames (e.g. white frames or black frames), whose standard deviations are very small. Among meaningful frames key frames correspond to the local maximum of motion activity defined in (1).

$$\alpha^* = \arg\max \left\{ \begin{array}{l} \alpha_s ; \alpha_s = \frac{1}{n} \sum_{k=1}^{n} \left| I_k^t - I_k^{t+s} \right| \\ s \in (1,10) \end{array} \right\} \tag{1}$$

where I_k^t is the k^{th} intensity value of the t^{th} frame, which is the key frame detected most recently, and at the beginning it is set as the first meaningful frame. n is the number of frame pixels. s is the frame step between previous key frame and the current frame being examined, and it is increased by 1 within a sliding window (size of 10 frames) until key frame is detected.

3.2 Template selection

Interest points on key frames are detected based on Harris interest point detector. Among the interest points detected, the four interest points that are the most leftward, rightward, top ward and bottomward are defined as boundary points. Area of interest is defined as the area embraced by boundary points and it lies in the range of $(x_{\min}, x_{\max})$ and $(y_{\min}, y_{\max})$ on the vertical and horizontal direction respectively. $x_{\min}$ and $x_{\max}$ are defined as the smallest and largest x-axis values of boundary points, which are on the vertical direction. $y_{\min}$ and $y_{\max}$ are similarly defined.

Since different editing methods may be used within one video sequence, area of interest varies from frame to frame. As a result, examining all the key frames based on the averaged positions of area of interest will cause propagation errors to the following operations, such as signature extraction and matching. However, examining each key frame based on individual area of interest is very vulnerable to noises and disturbances, such as the low quality video segments. Consecutive key frames transformed by the same editing method are considered as generated from the same template. The essential issue is how to associate key frames with templates. One flexible solution is to classify key frames into non-overlapping segments and determine the template label of each key frame within the segment. Thus, within the segment key frames with the same label are analyzed together and these key frames share the averaged positions of area of interest. In our experiment the number of template is up to 2, and the maximum size of segment is 500 key frames. This is just one of the reasonable choices in order to avoid complexity, which can be caused by too many template numbers, and ensure robustness, which can be affected by too few key frames within a segment.

Table I Template Label Decision

Index	Template One	Template Two	Position Difference	Template Label
1	Empty	Empty	N/A	1
2	Not empty	Empty	$D_1 \leq T_T$	1
3	Not empty	Empty	$D_1 > T_T$	2
4	Not empty	Not empty	$D_1 \leq D_2$	1
5	Not empty	Not empty	$D_1 > D_2$	2

Table I lists the criteria for template label decision, where D_1 defined in (2) means the position difference between current key frame and template number one, and T_T denotes the threshold value for decision. T_T is determined using the adaptive threshold selection method.

$$D_1 = \frac{\left|x_{\min}^c - x_{\min}^1\right| + \left|x_{\max}^c - x_{\max}^1\right| + \left|y_{\min}^c - y_{\min}^1\right| + \left|y_{\max}^c - y_{\max}^1\right|}{x_{\min}^1 + x_{\max}^1 + y_{\min}^1 + y_{\max}^1} \quad (2)$$

where $x_{\min}^c, x_{\max}^c, y_{\min}^c$ and $y_{\max}^c$ are the x-axis and y-axis values of area of interest from current key frame, while $x_{\min}^1, x_{\max}^1, y_{\min}^1$ and $y_{\max}^1$ are the x-axis and y-axis values of area of interest stored in template number one. The definition of D_2 is similar.

For case 1 listed in Table I, the 1^{st} key frame of each segment belongs to template number one, and its position information of area of interest is stored as following:

$$sx_{\min}^1 = x_{\min}^c, sx_{\max}^1 = x_{\max}^c, sy_{\min}^1 = y_{\min}^c, sy_{\max}^1 = y_{\max}^c, n^1 = 1 \quad (3)$$

$$x_{\min}^1 = sx_{\min}^1 / n^1, x_{\max}^1 = sx_{\max}^1 / n^1, y_{\min}^1 = sy_{\min}^1 / n^1, y_{\max}^1 = sy_{\max}^1 / n^1 \quad (4)$$

where $sx_{\min}^1, sx_{\max}^1, sy_{\min}^1$ and $sy_{\max}^1$ are the accumulators storing x-axis and y-axis values of area of interest in template number one. $x_{\min}^1, x_{\max}^1, y_{\min}^1$ and $y_{\max}^1$ are the buffers storing the averaged x-axis and y-axis values of area of interest in template one. n^1 is the accumulator storing the number of key frames belonging to template number one.

For case 2 and 3 in the decision table, the selection of template label depends on the position difference between current key frame and template number one. If D_1 is not more than the decision threshold, current key frame belongs to template number one; otherwise, it belongs to template number two. Once the decision is made, the template information is updated as following:

$$sx_{\min}^i += x_{\min}^c, sx_{\max}^i += x_{\max}^c, sy_{\min}^i += y_{\min}^c, sy_{\max}^i += y_{\max}^c, n^i += 1 \quad (5)$$

$$x_{\min}^i = sx_{\min}^i / n^i, x_{\max}^i = sx_{\max}^i / n^i, y_{\min}^i = sy_{\min}^i / n^i, y_{\max}^i = sy_{\max}^i / n^i \quad (6)$$

where i is template label.

Case 4 and 5 are situations that both template number one and number two have already been created. If D_1 is not more than D_2, current key frame belongs to template number one; otherwise, it belongs to template number two. Thus, the template information is updated accordingly.

3.3 Temporal redundancy detection

Temporal redundancy of video is mainly caused by insertions (e.g. caption or pattern) and black margins due to geometrical change (e.g. crop, shift or letter-box). Besides, motionless areas of video, which are consistent across several consecutive frames, also lead to temporal redundancy. Both of insertions and motionless areas are featured with static pixels of frame pictures. Black margins are featured with black pixels of frame pictures.

An averaged frame is generated from frames with the same template label within a segment, by averaging pixel intensity values. In this averaged frame, black pixels are defined as the pixels with very small intensity value, which are obtained by thresholding. In our case, the threshold T_B is determined by the adaptive threshold selection method. Since black pixels do not necessarily belong to black margins, the position information of area of interest is taken into consideration for further verification. The averaged area of interest is obtained by template selection procedure. In the averaged frame, margins are defined as the area shown outside the averaged area of interest. If black pixels occupy the majority number in the margin area, this margin is determined as black margin. As a result, the information of black margins is shared among the key frames with the same template label within a segment.

Difference frame is obtained by calculating the absolute difference values of pixel intensity between neighboring key frames with the same template label in a segment. Thus, an averaged difference frame is also generated with a segment, by averaging all the difference frames. In the averaged difference frame, static pixels are defined as the pixels with very small intensity value, which are obtained by thresholding. In our case, the threshold T_S is determined by the adaptive threshold selection method. As a result, the information of static pixels is shared among the key frames with the same template label within a segment.

3.4 Video in video detection

Video in video is a complicated transformation generated from the original video sequences, where two independent videos are playing back simultaneously in one frame picture. We refer the video displaying in the smaller part of frame picture as foreground video, and the video in the rest part as background video. The foreground video occurs mainly in five positions, top right, top left, bottom right, bottom left and center of the frame picture. Its scale usually varies from 50 percent to 20 percent of the original video frame. Our strategy for the detection of video in video is implemented in three steps. Firstly, the position of foreground video candidate is located using edge information. Secondly, foreground video is decided among candidates. Thirdly, the foreground video and background video are processed independently in terms of signature extraction and matching, which are defined in section 4 and 5, respectively.

The boundaries between foreground video and background video usually show distinguishing vertical line and horizontal line. Location of foreground video is to find the vertical line and horizontal line along which the accumulation number of edge points reaches maximum. Sobel edge detector is implemented to find horizontal and vertical edge point for its computation efficiency. On each pixel position, the status of being a horizontal and vertical edge point or not, is recorded as 0 or 1. If the pixel is determined as static pixel, it is set as a non-edge point.

On the top left of frame picture, the search of foreground video is to find the position with the maximum accumulation number of edge points defined in (7).

$$G_L^T(x,y) = \arg\max_{(x,y)} \left\{ \begin{array}{l} \sum\limits_{kx=x_T+h\cdot 20\%}^{x} \left(\sum\limits_{ky=y_L+w\cdot 20\%}^{y} \big(ex(kx,ky)+ey(kx,ky)\big) \right); \\ x \in (x_T + h\cdot 20\%, x_T + h\cdot 50\%); y \in (x_T + h\cdot 20\%, x_T + h\cdot 50\%) \end{array} \right\} \tag{7}$$

where $ex(kx,ky)$ and $ey(kx,ky)$ is status value of being vertical edge and horizontal edge on pixel position (kx,ky). h and w is the height and width of original frame picture, respectively. (x_T, x_B) and (y_L, y_R) is the range of frame pixels on vertical and horizontal direction, respectively, after the removal of black margins. The search of foreground video on top right, bottom left and bottom right of frame picture is similarly defined. As for the location of foreground video on the center, only top and left boundary lines are searched to save computation load, since the left and right boundary lines are symmetric, as well as top and bottom boundary lines.

After the location process, five candidates of foreground video are generated on five positions. The candidate with maximum G value defined in (7), which is larger than 15% of $(w+h)$, is chosen as the foreground video. The rest part of frame picture is the background video.

3.5 Video flip

Another common transformation from original video is to flip the original video along the vertical direction, causing the mirror effect. In order to handle this case, flipped scaled down frames are generated in signature extraction process defined in section 4, and these frames are independently processed in matching process defined in section 5.

4 Signature extraction

In the signature extraction process, each key frame is scaled down to be 9×11 and the corresponding frame mask is generated adaptively to reduce the temporal redundancy. For each key frame, the signature consists of 99-dimensional (99-D) pixel intensity in $[0,255]^{D=99}$, which is generated from its scaled down frame, and 99-dimensional (99-D) corresponding frame mask in $[0,1]^{D=99}$.

By the removal of black margins, the remaining part of frame is classified into non-overlapping equal size blocks. In our case, it is 9 blocks on the vertical direction and 11 blocks on the horizontal direction. The corresponding scaled down frame is obtained, whose pixel intensity value equals the averaged intensity value from each block. In scaled down frame, the significance value of pixel is defined as the percentage of non-static pixels inside each block. Larger the significance value is, more discriminative the pixel is. If video in video is detected in section 3.4, scaled down frames are generated from foreground and background video frame, respectively.

Frame mask is generated by the automatically selected threshold to filter out pixels with temporal redundancy.

5 Matching

The similarity between query video and target video is represented in two aspects: temporal similarity in terms of video segment pairs and spatial similarity in terms of frame pairs. Based on pixel intensity from scaled down frames and frame mask, our matching strategy is processed in three steps. Firstly, both target video and query video are sampled as video groups using adaptive step. Secondly, matched pairs of group are determined. Thirdly, matched pairs of frames are generated based on sequential match. Finally, decision is made using frame pair similarity and group similarity.

6 Experimental Results

The performance of our algorithm for the 10 types of transformations is summarized in table II.

Table II Performance of our algorithm

Transformations	1	2	3	4	5
Total_Queries	1508	1607	965	514	705
Mean_F1	0.728	0.794	0.789	0.839	0.783
Mean_proc_time	2231.43	2232.7	2228.97	2226.2	2229.04
Total_proc_time	448517	448773	448023	447466	448036
TP_count	36	43	46	53	54
Miss_count	98	91	88	81	80
FA_count	1472	1564	919	461	651
Min_NDCR	0.97	0.887	0.763	0.703	0.726
Decision_Threshold	0.953	0.942	0.942	0.94	0.942
Rfa	0	0.181	0.12	0.12	0.161
Pmiss	0.97	0.851	0.739	0.679	0.694
Type3_FAcount	735	785	553	270	404

Transformations	6	7	8	9	10
Total_Queries	320	278	1035	1337	800
Mean_F1	0.828	0.839	0.747	0.799	0.81
Mean_proc_time	2228.94	2226.72	2229.53	2229.28	2228.81
Total_proc_time	448016	447571	448135	448084	447990
TP_count	57	46	42	20	20
Miss_count	77	88	92	114	114
FA_count	263	232	993	1317	780
Min_NDCR	0.691	0.755	0.893	0.967	0.974
Decision_Threshold	0.928	0.925	0.95	0.939	0.948
Rfa	0.06	0.08	0.06	0.02	0.02
Pmiss	0.679	0.739	0.881	0.963	0.97
Type3_FAcount	100	22	449	503	397

7 Conclusions

Our strategy for video copy retrieval is implemented in three major procedures, including video analysis, signature extraction and matching. Firstly, a dedicated video analysis is implemented for input videos, so that different kinds of complex editing effect are well tackled. Secondly, key frames are scaled down to extract simple signatures, and the frame mask is generated adaptively to reduce video temporal redundancy. Thirdly, final matching is based on sequential match with adaptive sample step.

Acknowledgments. The authors wish to acknowledge the financial support under EU IST FP-6 Research Programme with the integrated project: LIVE (IST-4-027312).

References

[1] S.-A. Berrani, L. Amsaleg, and P. Gros, “Robust content-based image searches for copyright protection,” in Proc. ACMInt.Workshop on Multimedia Databases, 2003, pp. 70–77.

[2] A. JOLY, C. FRELICOT and O. BUISSON, “Robust content-based video copy identification in a large reference database ,” Second international conference on image and video retrieval, IL, USA, July, 2003, vol. 2728, pp. 414-424.

[3] Y. Ke, R. Sukthankar, and L. Huston, “Efficient near-duplicate detection and sub-image retrieval,” in Proc. ACM Int. Conf. Multimedia, New York, 2004.

[4] E. Chang, J. Wang, C. Li, and G. Wilderhold, “Rime—a replicated image detector for the world-wide web,” in Proc. SPIE Symp. Voice, Video, and Data Communications, 1998, pp. 58–67.

[5] A. Hampapur and R. Bolle, “Comparison of sequence matching techniques for video copy detection,” in Proc. Conf. Storage and Retrieval for Media Databases, 2002, pp. 194–201.

[6] C. Kim, “Ordinal measure of DCT coefficients for image correspondence and its application to copy detection,” in Proc. SPIE Storage and Retrieval for Media Databases 2003, Santa Clara, Jan. 2003, pp. 199–210.

[7] C. Kim, “Content-based image copy detection,” Signal Process. Image Commun., vol. 18, no. 3, pp. 169–184, Mar. 2003.

[8] R. Mohan, “Video sequence matching,” in Proc. Int. Conf. Audio, Speech and Signal Processing (ICASSP), vol. 6, Jan. 1998, pp. 3697–3700.

[9] A. Hampapur and R. M. Bolle, “Comparison of distance measures for video copy detection,” in Proc. IEEE Int. Conf. Multimedia and Expo (ICME), 2001, pp. 188–192.

[10] A. Hampapur, R. M. Bolle, “Feature based indexing for media tracking,” in Proc. IEEE Int. Conf. Multimedia and Expo (ICME), 2000, pp. 67–70.