MATCHING LOGOS FOR SLOW MOTION REPLAY DETECTION IN BROADCAST SPORTS VIDEO

Zhao Feng[1], *Dong Yuan*[2], *Wei Zhe*[2], *and Wang Haila*[1]

[1]France Telecom R&D Beijing, Beijing, 100190, China
{feng.zhao, haila.wang}@orange.com
[2]Beijing University of Posts and Telecommunications, Beijing, 100876, China
yuandong@bupt.edu.cn, weizhebupt@gmail.com

ABSTRACT

Slow motion replays are usually linked with semantically important highlights in broadcast sports video. A replay often happens between two logo transition sequences, which can be used to locate replays. In this paper, we present a new method to extract replays based on automatic logo template generation and logo searching. The method utilizes speeded-up robust features (SURF) to find repeating logo patterns and then to search those patterns in the video. Compared with the existed replay detection methods based on finding logo patterns, our method can handle more complex logo transition types. It can also be applied to the cases where different logo transitions are used in one match. We build a dataset consisting of 28 different logo transition types and 6 sport genres with 3907 minutes in total. Good experimental results have been achieved on the dataset and it validates the effectiveness and robustness of the proposed method.

Index Terms— Slow motion replay detection, SURF-based matching, logo template detection, logo searching

1. INTRODUCTION

Slow motion replays in broadcast sports video are important and reliable clues for highlights and key events, since they are usually chosen by human experts. Therefore, detection of replays could facilitate semantic-based high level video processing tasks such as highlight generation [1], event detection [2], video summarization [3], etc.

Some slow motion replay detection methods have been proposed in the literature. They can be broadly classified into three categories. The first category of method directly analyzes the inherent attributes of replay video segments and tries to use those attributes to differentiate the slow motion replays from the normal plays. Before high speed cameras are widely used, slow motion replays were usually produced by repeating frames recorded from standard cameras. A few detection methods are based on this assumption. Kobla et al. [4] proposed a replay detection method based on the macroblock type, motion vector and bit rate information. The detection was performed in the MPEG compressed domain to find the locations of repeating frames. Pan et al. [1] calculated the image differences between adjacent frames and search fluctuations in the differences to identify replays. These methods usually get low performance on replays that captured by high speed cameras and they can not provide accurate boundaries of replays.

The second category of method employs statistical techniques to perform shot classification for the purpose of detecting replay shots. Wang et al. [5] presented a method based on the difference of motions between replays and normal plays. A support vector machine (SVM) was trained with features of color ratio, shot length, mean color value and motion related feature to detect replay shots. Wang et al. [6] applied scene transition structure analysis on the shot classification results. Replays were extracted based on the generated shot label sequences by using some predefined rules. Ying et al. [7] employed HMM to classify replay and non-replay shots. The performance of this category is usually not satisfied and results are still not precise enough.

The third category of method tries to find the differences between replays and normal plays in terms of specific production behaviors, such as special video effects or logo patterns. Replays are then determined according to the existence of such behaviors. Babaguchi et al. [8] captured a digital video effect (DVE) model interactively for given videos. It was described by color and motion of the gradually changing boundary between adjacent shots. A replay was located by two DVEs detected with the model.

Based on the observation that more and more replays are sandwiched in between two certain logo transition sequences, some researchers try to use logo transition locations to identify replays. Pan et al. [9] used mean-square differences of intensity and color histogram as frame similarity measure to detect logo template and to search logos in videos. Duan et al. [10] employed the spatio-temporal mode seeking on selected logo sequences to capture the dominant color mode. This color mode was then used to perform EMD-based similarity matching for logo searching purpose. Tong et al. [11] used frame-to-frame difference with MSD to select candidate logo sequence. They calculated the average image of those candidate logo frames near to the center to obtain logo template. Template matching was performed based on color and shape features.

Huang et al. [12] first extracted gradual transitions and employed motion features to learn the logo template. A color representation was then computed from the logo template and was used to search logos. Dang et al. [13] semi-automatically extracted logo template sequences and detected replays based on template sequence matching. Han et al. [14] learned gradual transition patterns based on motion vector reliability classification from a database by SVM. Replay detection was performed by fusing the pattern matching results and slow motion detection results based on motion vector information.

The third category of method can usually obtain high accuracy on replay boundaries and the performance is better than above two categories. However, these methods often make assumptions on the logo transition sequences such as logo locations, moving patterns and speed, types of digital effects, etc. And nearly all of them assume that the two logo transition sequences before and after the replay are identical.

In this paper we propose an effective and reliable replay detection method based on logo template generation and matching by employing the fast and robust local feature SURF [15]. Our method falls into the third category method mentioned above. It follows the common framework that the first stage is logo template generation and the second stage is searching logo template in the video. In both stages SURF-based matching is employed to measure the similarity between video frames. The logo template consists of a frame in the video and the maximal detection length linked with the frame. We also present an efficient searching approach that avoids processing all the frames in the video.

The main advantages of our method are: First, it can handle more complex logo transition types regardless the logo color, location, moving pattern or digital effects such as wipe, dissolve, zoom, etc. Second, our method can apply to the situation where different logo transition sequences are used for replays in one match, as long as they are visually similar. The robustness of SURF contributes to this point. Last, the method achieves acceptable speed while keeping high performance in terms of recall and precision. The method also avoids complex operations and is easy to use.

The rest of this paper is organized as follows. Section 2 presents the logo template generation algorithm. Section 3 describes how to search the logos in the video, as well as the logo pairing rules. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. LOGO TEMPLATE GENERATION

Inspired by the previous work [9, 12], we use a fast shot boundary detection (SBD) algorithm [16] to extract all gradual transitions (GT) as candidate logo transition sequences. The minimum length of detected GTs is set to 15 empirically in order to cover more logo transitions. The SBD algorithm is run for a second time to extract only cuts. Then the zero-crossing-based slow motion detection algorithm [1] is applied to the cuts that are overlapped with GTs for further candidate refinement. Those cuts that are not detected as slow motion shots are removed. The remained cuts are sorted by the detection scores decreasingly and form the final candidate sets. Note that the detection performance of this algorithm on sports videos captured by high speed cameras is not high. But it is effective for refining the set of candidate logo transition sequences in our method.

In the final candidate set, we use the first GT's location in each cut as the beginning of logo transition sequence. The length of the sequence is 25 frames. The next step is to perform sequence matching to find the logo template. The similarity measure will be the number of matched points, which is generated by the SURF-based matching. Therefore, some matches are not expected such as the points on fixed logos or score information bars. A detection border is used to prevent such phenomenon. If a match is located near the top and bottom of the screen, it will not be counted into number of matches. The border is set to 1/5 of the video height in our experiments. The algorithm of logo template generation is defined as follows:

Step 1: Take the i-th sequence SEQ_i from the candidate sets LS. Extract SURF features for each frame F_{ip} in SEQ_i (p=1, 2, …, 25).

Step 2: Take the j-th sequence SEQ_j from the candidate sets LS (j=i+1, i+2, …, $|LS|$). Extract SURF features for each frame F_{jp} in SEQ_j (p=1, 2, …, 25).

Step 3: For each F_{ip}, computing the matches between every frame in SEQ_j using SURF features based on nearest-neighbour-ratio matching strategy. If F_{ip} can stably match 3 or more successive frames in SEQ_j, we mark it as a stable frame. Here the term 'stably match' means the number of matches that falling into valid region (inside the borders) is larger than 10.

Step 4: If 4 or more successive frames in SEQ_i are all marked as stable frame, we consider them as a stable match group. Find the group that has maximal length and record related information (index i and j, starting and ending frame in SEQ_i) in stable match group array SMG.

Step 5: Check elements in SMG, if $|SMG|$>=2 and the last two stable match groups have the same index for their first sequence, go to step 6. Otherwise if j<$|LS|$, increasing j with 1 and go to step 2. If j=$|LS|$ and i<$|LS|$-1, increasing i with 1 go to step 1. Otherwise it means all sequences are processed and output no logo template has been detected.

Step 6: For the last two elements in SMG, check the starting and ending frames to find whether they have overlapped part in the first sequence. If not overlapped or overlapped length is less than 3, output no logo template has been detected. Otherwise check all frames in the overlapped part to find the frame that has the maximal number of matches in the two stable match groups. Record this frame as logo template frame L_T. Then find the minimal length of successive frames that L_T can stably match in the two stable match groups. Record it as maximal detection length DL_M.

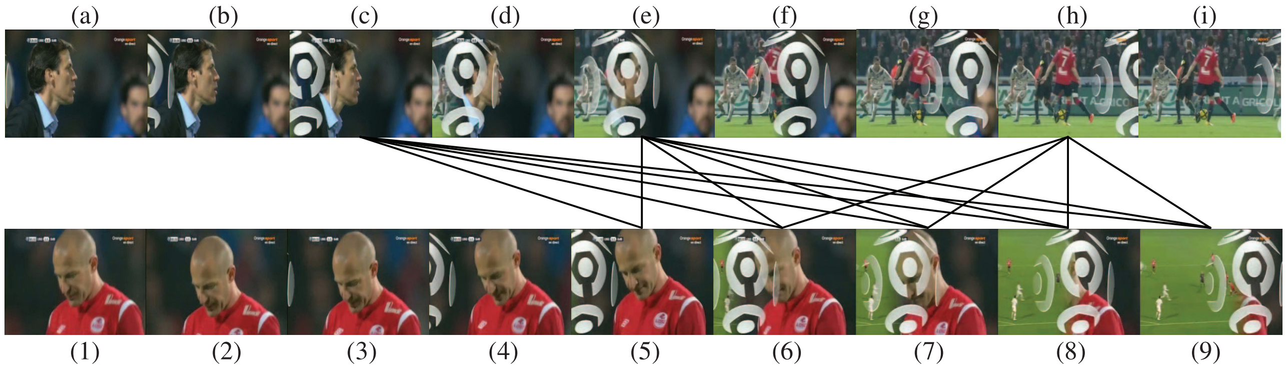

Fig. 1. Logo template generation by matching logo sequences (some frames are not shown for a clear vision).

Output logo template has been detected and record related information including frame index of starting and ending frame, frame index of L_T and maximal detection length DL_M.

Figure 1 illustrates the sequence matching and logo generation process. For simplicity we only use one stable match group. Let the first row of images be SEQ_i and let the second row of images be SEQ_j. Each frame in SEQ_i is matched with all the frames in SEQ_j. Frames (c) to (h) are the stable frames since each of them matches 3 or more successive frames in SEQ_j. Each line indicates a match and the match lines of (d), (f) and (g) are not drawn to avoid clutter. Then frames (c) to (h) form a stable match group. Suppose match (e)-(7) obtain the highest number of matches in the group. Frame (e) is recorded as L_T. Since frame (e) matches 5 successive frames (5)-(9) in SEQ_j, the DL_M is 5.

3. SEARCHING LOGOS IN THE VIDEO

Traditional logo searching approaches usually compare the logo template with all frames in the video. This is very time consuming. In our method we present a fast searching strategy that we only compare the logo templates with the frames within the detected GTs and we could use DL_M to further accelerate the searching process. After logo pairing process we try to find pairing logos in the nearby frames for the isolated logos. Experimental results verify our strategy is effective and efficient. The performance lost is only about 1% in recall but we obtain obviously speed increasing. The algorithm of logo searching and pairing is defined as below:

Step 1: Searching step S_{step} is set to (DL_M -1). Searching iteration number S_{iter} for one GT is set to ($\lfloor 25/S_{step} \rfloor$+1). Extract SURF features in L_T.

Step 2: Take i-th GT from all detected GTs set G (i=1, 2, …, $|G|$). Let the frame index of G_i's starting frame be F_{start}.

Step 3: Let the test frame index $F_{ind}= F_{start}+j* S_{step}$ (j=0, 1, …, S_{iter}-1). Extract SURF features from frame L_{test} with the index F_{ind}. Match L_T with L_{test} and get the number of matches n_1. Match L_{test} with L_T and get the number of matches n_2 (changing order of frames). If n_1>10 and n_2>10 then go to step 4. Otherwise if $j<S_{iter}$-1 then increase j with 1 and repeat step 3. Otherwise go to step 2.

Step 4: Match L_T with nearby 4 frames of L_{test}. If 2 or more frames get more than 10 matches, add F_{ind} in detected logo set LD. If $i<|G|$ go to step 2. Otherwise go to step 5.

Step 5: Checking all logos in LD with pairing rules to pair up the logos of same replays. Suppose A, B, C, D are four successive logos in LD. Pairing rules are defined as:

1. Difference of shot indexes A and B should less than 6. Difference of frame indexes A and B should less than 1300.

2. Difference of shot indexes C and D should less than 6. Difference of frame indexes C and D should less than 1300.

3. The difference of shot indexes A and B is less than the difference of shot indexes B and C.

4. The difference of frame indexes A and B is less than the difference of frame indexes B and C.

Logos A and B are paired up only if they conform to rule 1 and conform to at least one rule among rule 2, 3, 4. If logos A and B are paired up then record them in logo pair set LP, which is also the detected replay set. Otherwise put A into isolated logo set L_{iso} and continues pairing process from B. If all logos in LD are processed then go to step 6.

Step 6: For each logo in L_{iso}, search its nearby -750 ~ +750 frames using searching step S_{step} to find whether a new logo is paired with it. If a pairing logo has been found then record them into LP. Otherwise discard this isolated logo.

4. EXPERIMENTAL RESULTS

The sports video dataset used in our experiments consists of 6 sport genres of football, tennis, badminton, rugby, skiing and boxing. It contains 43 game matches and the total length is 3907 minutes. They are chosen from 2 years' broadcast program collections in Orange Sport Channel, Eurosport Channel and other sources. There are 28 types of logo transitions varying in shape, moving pattern and digital effects of wipe, dissolve, zoom, etc. The totally replay number is 1551. Most videos have a resolution of 512*288.

Table 1 presents the logo and replay detection results. Note that for football and rugby results, the numbers in the brackets means the numbers of totally different logos used in the matches. They are usually used for special events' replay such as goal. In our dataset, 6 types of such logos appear 32 times in 4 matches. This kind of logo would be difficult to

Table 1. Logo and replay detection results

Sport Genres	Total Logos	Missed Logos	False Alarm	Total Replays	Total Detected	Correctly Detected	Recall (Replay)	Precision (Replay)	Different logos found
Football	2622(26)	44	4	1313(13)	1284	1282	97.64%	99.84%	64
Tennis	210	9	0	105	100	100	95.24%	100%	0
Badminton	120	4	0	60	58	56	93.33%	96.55%	0
Rugby	38(6)	6	0	19(3)	16	16	84.21%	100%	0
Snowboard	30	0	0	15	15	15	100%	100%	0
Boxing	78	1	0	39	38	38	97.44%	100%	0
Total	**3098**	**64**	**4**	**1551**	**1511**	**1507**	**97.16%**	**99.74%**	**64**

detect without any prior knowledge. The other point is that a few logos for several replays are unexpectedly disappeared in some football matches. It makes the number of total logos not exactly two times of the replays number. The false detections of logo occur when logo template matches some signs in the playgrounds that contain similar logos or characters. But not all of them will generate false alarms due to the pairing rules.

Fig. 2. Logo samples

Figure 2 illustrates some logo templates detected in dataset and logos detected by them. In the third row, the 1st and 3rd images are logo templates and the 2nd and 4th images are logos detected by them. They are visually similar to the logo templates thus could be matched using SURF. From the last column of Table 1 we can see that total 64 such logos have been detected by our method. It is usually difficult for other methods to detect such logo pairs.

The tests are running on a Linux server with 2.0 GHz Intel Xeon CPU and 64G memory. For one 512*288 MPEG4 video file with 100 minutes, about 10 minutes are spent on video decoding averagely (a 100*100 YUV file for SBD and all frames in JPEG). The rest processing time varies from 10~30 minutes, which should be acceptable for most cases.

5. CONCLUSIONS

We propose a new method for slow motion replay detection. The method employs SURF-based matching to automatic generate logo template and search it in the video. An efficient logo searching and pairing strategy is also presented. Experiments on a large dataset achieve good results. It verifies that our method is robust against various logo transition types and digital video effects for replay detection tasks in different sport genres. Future work includes the study of how to detect multiple visually different logo transitions that occur in a same match.

6. REFERENCES

[1] H. Pan, P. van Beek, and M.I. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation," in Proc. of ICASSP'01, vol. 3, pp. 1649–1652, 2001.

[2] X. F. Tong, H. Q. Lu, and Q. S. Liu, "A three-layer event detection framework and its application in soccer video," in Proc. of ICME'04, vol. 3, pp. 1551–1554, 2004.

[3] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," IEEE Transactions on Image Processing, vol. 12, no. 7, pp. 796–807, 2003.

[4] V. Kobla, D. DeMenthon, and D. Doermann, "Detection of slow-motion replays for identifying sports videos," in Proc. of MMSP'99, pp.135–140, 1999.

[5] L. Wang, X. Liu, S. Lin, G. Y. Xu, and H.-Y. Shum, "Generic slow-motion replay detection in sports video," in Proc. of ICIP'04, vol.3, pp. 1585–1588, 2004.

[6] J. J. Wang, E. Chng, and C. S. Xu, "Soccer replay detection using scene transition structure analysis," in Proc. of ICASSP'05, vol.2, pp. 433–436, 2005.

[7] Y. Yang, S. X. Lin, Y. D. Zhang, and S. Tang, "A statistical framework for replay detection in soccer video," in Proc. of ISCAS'08, pp. 3538–3541, 2008.

[8] N. Babaguchi, Y. Kawai, Y. Yasugi, and T. Kitahashi, "Linking live and replay scenes in broadcasted sports video," in Proc. of MULTIMEDIA'00, pp. 205–208, 2000.

[9] H. Pan, B. Li, and M. I. Sezan, "Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions," in Proc. of ICASSP'02, pp. 3385–3388, 2002.

[10] L.-Y. Duan, M. Xu, Q. Tian, and C.-S. Xu, "Mean shift based video segment representation and applications to replay detection," in Proc. of ICASSP'04, pp. 709–712, 2004.

[11] X. F. Tong, H. Q. Lu, Q. S. Liu, and H. L. Jin, "Replay detection in broadcasting sports video," in Proc. of ICIG'04, pp. 337–340, 2004.

[12] Q. Huang, J. M. Hu, W. Hu, T. Wang, H. L. Bai, and Y. M. Zhang, "A reliable logo and replay detector for sports video," in Proc. of ICME'07, pp. 1695–1698, 2007.

[13] Z. H. Dang, J. Du, Q. M. Huang, and S. Q. Jiang, "Replay detection based on semi-automatic logo template sequence extraction in sports video," in Proc. of ICIG'07, pp. 839–844, 2007.

[14] B. Han, Y. Yan, Z. H. Chen, C. Liu, and W. G. Wu, "A general framework for automatic on-line replay detection in sports video," In Proc. of MM'09, pp. 501–504, 2009.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, 2008.

[16] S. G. Lian, Y. Dong, H. L. Wang, "Efficient temporal segmentation for sports programs with special cases," in Proc. of PCM'10, pp. 381–391, 2010.