

Research Article

An Effective Slow-Motion Detection Approach for Compressed Soccer Videos

Vahid Kiani and Hamid Reza Pourreza

Machine Vision Laboratory, Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad 9177948944, Iran

Correspondence should be addressed to Vahid Kiani, vahid.kiani@rocketmail.com

Received 21 December 2011; Accepted 15 January 2012

Academic Editors: M. Pardàs and A. Prati

Copyright © 2012 V. Kiani and H. R. Pourreza. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Slow-motion replays are content full segments of broadcast soccer videos. In this paper, we propose an efficient method for detection of slow-motion shots produced by high-speed cameras in soccer broadcasts. A rich set of color, motion, and cinematic features are extracted from compressed video by partial decoding of the MPEG-1 bitstream. Then, slow-motion shots are modeled by SVM classifiers for each shot class. A set of six full-match soccer games is used for training and evaluation of the proposed method. Our algorithm presents satisfactory results along with high speed for slow-motion detection in soccer videos.

1. Introduction

Replays in soccer broadcasts cover most important contents of the video. Quick development of video compression techniques led to huge compressed video archives. Compressed domain video analysis on these archived videos can result in efficient video processing frameworks. However, noisy features are main challenge in compressed video analysis.

Nowadays, most sports broadcasters use logo transitions before and after replay shots [1–3]; however, detection of replay shots by slow-motion detection can bring robustness and generality to prevailing systems. Slow-motion replays could be produced by standard or high-speed cameras. Several approaches are proposed for slow-motion detection based on each production style. A slow-motion replay from a standard camera can be generated by repeating some normal frames or inserting morphed frames between two consecutive frames [4, 5]. Repeated or inserted frames result in special patterns in frame difference feature and could be detected easily in spatial [4–6] or compressed domain [7–9].

Recently, majority of broadcasters are using high-speed cameras for slow-motion generation to achieve finer presentation of fast movements. When high-speed cameras are used with recording frame rate higher than desired slow-motion frame rate, some frames must be dropped. Dropped frames could be detected by plentiful fluctuations in frame

difference feature [10]. In addition, slow-motion replays generated by dropped frames has higher mean of absolute frame difference than slow-motion replays generated by normal cameras [4]. With a high-speed camera, the slow-motion effect can also be generated by simply playing out the video at the normal speed. We call these slow-motion replays generated by high-speed cameras HISM replays. Detection of HISMs is very challenging because they do not result in any special pattern in visual features. Han et al. in [11] tried to detect HISMs by exploiting camera motion patterns and achieved to 72% precision and 66% recall rates. Wang in [12] used color and motion features to model HISMs with SVM classifiers and obtained 75% precision and 61% recall rates on sports videos. Yang and others in [13] improved the work of Wang et al. in [12] and used HMM models. They achieved 83% precision and 81% recall rates on soccer videos. In contrast, scene transition structures are used in [14] as a general method for replay detection and resulted in 74% precision and 86% recall rates on three soccer games. All prior works on detection of HISMs are proposed for spatial domain analysis and have low speed.

In this paper, we propose an efficient framework for slow-motion detection in compressed MPEG-1 soccer videos. Several color, motion, and cinematic features are used in our framework to achieve satisfactory results despite intrinsic

noise in compressed domain. The rest of the paper is organized as follows. Section 2 presents an overview of the proposed framework for slow-motion detection. In Section 3, the proposed framework is evaluated by precision and recall rates on soccer videos. Finally, Section 4 presents conclusion and future work.

2. Proposed Framework

Humans can discriminate between slow-motion and live shots by noticing the speed of players or ball motion in the scene [12]. However, robust detection and tracking of moving objects in soccer videos is a challenging task. On the other hand, color difference and motion features are correlated with object and camera motion in the scene. We try to use color, motion, and cinematic features for slow-motion modeling.

Shots in soccer video can be categorized into four classes: long, medium, close up, and out of field [6]. Close-up and out-of-field shots do not exhibit important events of the game. Thus, we do not consider close-up and out-of-field shots in our framework. In addition, motion-related visual features of long and medium shots are fully different. Therefore, we try to model slow-motion shots for each shot class independently.

Shot boundary detection and shot classification are two preliminary tasks in soccer domain and studied in several works [6, 15–18]. Any error in these modules can result in high error rates in slow-motion detection. For an unbiased evaluation of slow-motion detection module, in our framework shot boundary detection and shot classification are done by human. First, low-level color and motion features are extracted efficiently from compressed domain. Then, a GMM is used to model grass color adaptively in each video stream. Color, motion and cinematic features are extracted in the next step. Thereafter, a SVM classifier is trained for slow-motion detection in each shot class based on training data. Finally, trained SVM classifiers are used for classification of slow-motion shots in test videos.

2.1. Low-Level Information Extraction

In this step, low-level color and motion information are extracted from MPEG bitstream. A MPEG-1 video decoder in Java language modified to extract this information from compressed video by partial decoding of the bit-stream.

2.1.1. DC Sequence Extraction. MPEG-1 is a block-based video compression technique which exploits discrete cosine transform (DCT) and motion compensation error to reduce spatiotemporal redundancy in image sequence [19, 20]. In MPEG-1 standard, each picture is divided to 16×16 subimages called macroblocks (MBs). Each macroblock constitutes from four luminance and two chrominance blocks. For each block in MPEG sequence, the first DCT coefficient is called DC coefficient and contains mean intensity of the block [21]. Therefore, a down-sampled version of each picture could be constructed by approximating DC coefficients of picture blocks. We utilized method proposed in [22, 23] for

DC image extraction. The DC image of each picture in the sequence has a luminance plane Y and two chrominance planes: Cb and Cr . Color and object feature could be extracted from approximated DC image.

2.1.2. Motion Vector Extraction. We use only I-Pictures or P-Pictures of MPEG stream. In an I-Picture, all MBs are coded independently from other pictures and named intracoded MB. In a P-Picture, MBs can code as an intracoded MB or a forward-coded MB. A forward-coded MB is similar to previous reference picture (I-Picture or P-Picture) and is coded by its displacement to similar area in previous reference frame and its compensation error in DCT domain. We consider displacement of each forward-coded MB as its motion vector as follows:

$$x'(m, n, t - 1) = x(m, n, t) + MV_x(m, n, t), \quad (1)$$

$$y'(m, n, t - 1) = y(m, n, t) + MV_y(m, n, t), \quad (2)$$

where $x(m, n, t)$ and $y(m, n, t)$ are center point locations of MB(m, n) in current picture, $x'(m, n, t - 1)$ and $y'(m, n, t - 1)$ are center point locations of most similar area in previous picture, and $MV_x(m, n, t)$ and $MV_y(m, n, t)$ are horizontal and vertical components of MB(m, n) motion vector in current picture.

2.1.3. Grass Modeling. To model grass color, a temporal subsampling technique is used. In this manner, grass color is extracted adaptively from each video sequence by processing I-Pictures only. After processing each I-Picture, the next 20 I-Pictures are skipped and not processed. For each processed I-Picture, all pixels with color in the following range are considered as green:

$$\begin{aligned} 90 < Y < 140 \\ 110 < Cb < 130 \\ 90 < Cr < 120. \end{aligned} \quad (3)$$

Then, green pixels are grouped into disjoint connected components, and all connected components smaller than one-tenth of picture size are removed. Remaining pixels are candidate grass pixels.

After that, a Gaussian mixture model (GMM) is used to model grass color by using candidate grass pixels in the video. Several GMMs with one, two, three, and four Gaussians are used to model grass color, and optimal model with minimum error is used as final grass color model.

2.2. Feature Extraction. In this section, several color, motion, and cinematic features are extracted from compressed soccer videos to model slow-motion shots among long and medium shots.

2.2.1. Cinematic Features. Cinematic features are extremely exploited in soccer video analysis because they are lightweight and effective features. Similarly, we try to exploit cinematic knowledge of soccer videos for slow-motion detection.

(1) *Shot Length*. Considering long shots, most live scenes last more than 16 seconds. On the other hand, slow-motion medium shots are often longer than live medium shots [12]. So, shot length can be a promising feature for discriminating live shots from slow-motion shots. Therefore, the length of each shot is computed and denoted by $\text{Length}(i)$ for i 'th shot of the sequence.

(2) *Shot Type*. During a break, director may use several special consecutive shots to view a replay of last occurred event. This leads to common shot type patterns for replay scenes. Therefore, we use shot type of two prior and two next shots of the current shot as next features: respectively $S_{P1}(i)$, $S_{P2}(i)$, $S_{N1}(i)$, and $S_{N2}(i)$.

(3) *Repeated Frame*. In some slow-motion long shots, director freezes the scene for a short time during critical moments. As a result, detecting repeated frames in a short duration could be informative for slow-motion detection. Repeated frames can cause a low picture difference in DC image sequence. Therefore, we define repeated frame feature for t 'th picture in the sequence as:

$$\text{Repeated}(t) = \begin{cases} 1 & D_Y(t) < \text{Th}_{D_Y} \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where Th_{D_Y} is image difference threshold and D_Y is DC image difference. We define D_Y as:

$$D_Y(t) = \frac{1}{4 * M * N} \sum_{m=1}^{2M} \sum_{n=1}^{2N} |Y(m, n, t) - Y(m, n, t - 1)|. \quad (5)$$

The Y is luminance plane of DC image; M and N are number of rows and columns of MB grid (each macroblock contains four luminance blocks). Then, existence of frozen pictures in a shot can distinguished as

$$\text{Repetition}(i) = \begin{cases} 1 & \left(\sum_{t \in \text{Shot}_i} \text{Repeated}(t) \right) > \text{Th}_{\text{Repetition}} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

We used $\text{Th}_{D_Y} = 0.05$ and $\text{Th}_{\text{Repetition}} = 3$ in our experiments.

2.2.2. *Color Features*. Color features can indicate similarity of consecutive pictures in the sequence. On the other hand, some object features could also approximated by color features. Several color features are introduced in this section for slow-motion detection.

(1) *Grass Ratio Difference*. Grass ratio difference of consecutive pictures in each medium shot is correlated with camera and object motion in the scene. To compute grass ratio of each picture, constructed GMM grass model is applied on each pixel of DC image. Then, all blocks with probability higher than Th_{grass} are considered as grass pixels. Value of Th_{grass} is determined empirically for whole video dataset. Finally, the ratio of grass pixels in t 'th picture of the sequence

is called $G(t)$ and computed by dividing number of grass blocks by number of all blocks in the DC image.

We define mean of grass ratio difference in the shot as

$$M_{\text{Grass}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |G(t) - G(t - 1)|, \quad (7)$$

where $G(t)$ is ratio of grass-colored pixels in DC image of t 'th picture in the sequence.

(2) *Difference of Luminance Standard Deviation*. Camera and object motion in the scene can cause changes in luminance contrast of the picture. Variance is a measure for contract of image luminance [24, 25]. So, changes in standard deviation of luminance correlate with motion patterns in the scene. Therefore, mean of luminance contrast difference in each shot is defined as the next descriptor of the shot:

$$M_{\text{StdY}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{StdY}(t) - \text{StdY}(t - 1)|, \quad (8)$$

where $\text{StdY}(t)$ is standard deviation of pixels intensities in Y plane of t 'th DC image of the sequence.

(3) *Object Ratio*. For each shot type, size of biggest object in the picture can be related with distance from camera to objects. A small movement causes high motion magnitude in the picture when camera is close to objects and vice versa. The biggest object size in the picture can be approximated by biggest connected component's size in DC image. Consequently, we define next shot descriptor as

$$R_{\text{Object}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} R_o(t), \quad (9)$$

where $R_o(t)$ is ratio of biggest connected component in the DC image of t 'th picture with respect to DC image size.

2.2.3. *Motion Features*. Due to slower motion of objects in slow-motion scenes, the most promising features for slow-motion detection are motion-related features. In this step, we introduce several motion features for slow-motion shot modeling.

In [26], we proposed a method for motion vector reliability measurement and global motion estimation in compressed MPEG-1 sequences. For each motion vector, a reliability value between zero and one is extracted and called $\text{MV}_{\text{Reliability}}(m, n, t)$. In addition, global motion parameters, namely, $\text{GM}_P(t)$, $\text{GM}_T(t)$, and $\text{GM}_Z(t)$ are computed for each frame which denote pan, tilt, and zoom factor in t 'th picture of the sequence. Figure 1 shows reliable motion information extraction from a sample picture by the mentioned method. In Figure 1(b), intensity of each motion vector indicates its reliability.

(1) *Skipped Macroblocks Ratio*. In MPEG compression standard, MBs which are very similar to reference picture have no prediction error. These MBs are not coded in the MPEG bit-stream and called skipped MBs. Thus, ratio of skipped MBs



FIGURE 1: Reliable motion information extraction. (a) all motion vectors; (b) motion vectors reliability; (c) reliable motion vectors; (d) estimated camera motion.

in each picture shows picture similarity to reference picture. In slow-motion shots, we expect higher similarity between consecutive pictures; which results in higher skipped MB ratio in each picture. Therefore, we define ratio of skipped MBs in each picture as:

$$R_S(t) = \frac{1}{M*N} \sum_{m=1}^M \sum_{n=1}^M \text{Skipped MB}(m, n, t), \quad (10)$$

where t is index of current picture in the sequence, and Skipped MB for block $MB(m, n)$ of t 'th picture is defined as

$$\text{Skipped MB}(m, n, t) = \begin{cases} 1 & \text{skipped macro block} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Then, we define mean of skipped MBs ratio in the shot as

$$M_{\text{Skipped}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} R_S(t). \quad (12)$$

(2) *Reliable MV Magnitude*. Motion magnitude in each picture can indicate fast or slow movements of camera and objects in the scene [12]. However, in MPEG compressed sequences some motion vectors (MVs) are noisy and unreliable. Therefore, we try to discriminate reliable MVs from unreliable MVs and compute mean magnitude of reliable MVs.

The reliable MVs are distinguished by

$$\text{Reliable MV}(m, n, t) = \begin{cases} 1 & \text{MV}_{\text{Reliability}}(m, n, t) > \text{Th}_{\text{MVRel}}(t) \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

where $\text{Th}_{\text{MVRel}}(t)$ denotes adaptive threshold for reliable MVs detection which is computed as

$$\text{Th}_{\text{MVRel}}(t) = \max(1.2 * M_{\text{MVRel}}(t), 0.3). \quad (14)$$

The $M_{\text{MVRel}}(t)$ denotes mean reliability of motion vectors in t 'th picture of the sequence and is defined as

$$M_{\text{MVRel}}(t) = \frac{1}{M*N} \sum_{m=1}^M \sum_{n=1}^N \text{MV}_{\text{Reliability}}(m, n, t), \quad (15)$$

where M and N denote number of rows and columns of MB grid respectively. Then we define mean reliable motion magnitude of each picture as:

$$M_{\text{RMag}}(t) = \frac{1}{\sum_{m=1}^M \sum_{n=1}^M \text{Reliable MV}(m, n, t)} \times \sum_{m, n \in \text{reliableMV}} \sqrt{\text{MV}_x(m, n, t)^2 + \text{MV}_y(m, n, t)^2}. \quad (16)$$

Then, mean motion magnitude of the shot is defined as

$$M_{\text{ReliableMag}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} M_{\text{RMag}}(t). \quad (17)$$

(3) *Motion Magnitude Fluctuations*. When slow-motion shot is produced by frame-dropping technique, numerous fluctuations appear in mean motion magnitude feature. On the other hand, recently some directors play only critical seconds of a replay shot as slow-motion; while play other moments of the shot with live speed. The same pattern of numerous fluctuations can appear in these replay shots on slow motion moments. Local variance of values is an indicative measure

for magnitude fluctuations. Therefore, we moved a time window with length 10 on the mean motion magnitude feature and computed variance of values in each location as $M_{\text{magvar}}(t)$. Then, motion magnitude fluctuations feature is defined for each shot as follows:

$$M_{\text{Vibration}}(i) = \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} M_{\text{magvar}}(t). \quad (18)$$

(4) *Camera Motion*. Camera movement in slow-motion shot is slower than similar live shots [11]. Hence, we compute mean of each camera motion coefficient as in next features:

$$\begin{aligned} M_{\text{GMP}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_P(t)|, \\ M_{\text{GMT}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_T(t)|, \\ M_{\text{GMZ}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_Z(t) - 1|. \end{aligned} \quad (19)$$

The $M_{\text{GMP}}(i)$, $M_{\text{GMT}}(i)$, and $M_{\text{GMZ}}(i)$ indicate mean pan, tilt, and zoom factor in the shot, respectively.

(5) *Camera Motion Difference*. Speed of camera motion changes could also be slower in slow-motion shots with respect to similar live shots [11]. Therefore, the mean change of motion parameters for each shot is computed as

$$\begin{aligned} D_{\text{GMP}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_P(t) - \text{GM}_P(t-1)|, \\ D_{\text{GMT}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_T(t) - \text{GM}_T(t-1)|, \\ D_{\text{GMZ}}(i) &= \frac{1}{\text{length}(i)} \sum_{t \in \text{Shot}_i} |\text{GM}_Z(t) - \text{GM}_Z(t-1)|. \end{aligned} \quad (20)$$

2.2.4. *Semantic Features*. Soccer domain features which have special meanings for viewers are called semantic features. These features could also be used for slow-motion modeling in soccer videos.

(1) *Field-Side View*. Most slow-motion long shots contain shoot and offside events occurred near goal area. So, detection of field-side views can be helpful for detecting slow-motion long shots. Kolekar in [27], used a simple rule and grass ratio in three regions of the picture for field-view detection. Similarly, we divide each picture into three regions as shown in Figure 2. Then, we use following rule for view classification of each frame during long shots.

IF $|G_{\text{RUL}}(t) - G_{\text{RRUR}}(t)| > \text{Th}_{\text{FieldSide}}$ THEN $\text{FieldSide}(t) = 1$ ELSE $\text{FieldSide}(t) = 0$.

The G_{RUL} and G_{RRUR} denote grass ratio in left and right region of the picture, respectively. We used $\text{Th}_{\text{FieldSize}} = 0.7$ in our experiments.

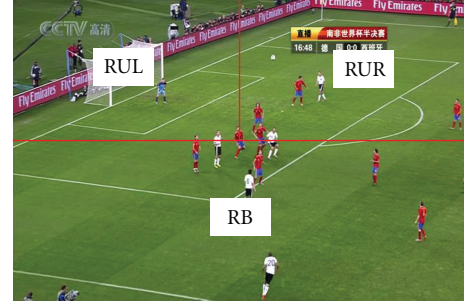


FIGURE 2: Partitioning of picture into three regions for field-side view detection.

TABLE 1: FUM-BSVD 2011 soccer video dataset.

Number	Game	Length
1	Germany versus England	01 : 33 : 22
2	Greece versus Argentina	01 : 34 : 34
3	Slovakia versus Italy	01 : 40 : 07
4	Germany versus Argentina	01 : 32 : 17
5	Germany versus Spain	01 : 34 : 31
6	Germany versus Uruguay	01 : 35 : 17

TABLE 2: FFMPEG encoder settings for soccer videos.

Parameter	Value
GOP Size	16 (fix)
B-picture count	0
Frame rate	25
Bit-rate	10240 kbps
Frame size	720 × 480
Maximum motion vector range	100 pixel

Then, we examine existence of field-side view in each long shot as

$\text{SideView}(i)$

$$= \begin{cases} 1 & \left(\sum_{t \in \text{last 50 pictures of Shot}_i} \text{FieldView}(t) \right) > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

For medium and close-up shots, the $\text{SideView}(i)$ is equal to zero.

2.3. *Slow-Motion Detection*. In this section, we model slow-motion shots for each shot class separately. A set of features is selected for each shot class based on aforementioned intuitive motivations. Since close-up and out-of-field shots do not show an important event in the game, we do not consider them in our framework. Figure 3 shows diagram of proposed classification algorithm.

During training phase, following features of long shots are given to an SVM model: shot-type features, repeated

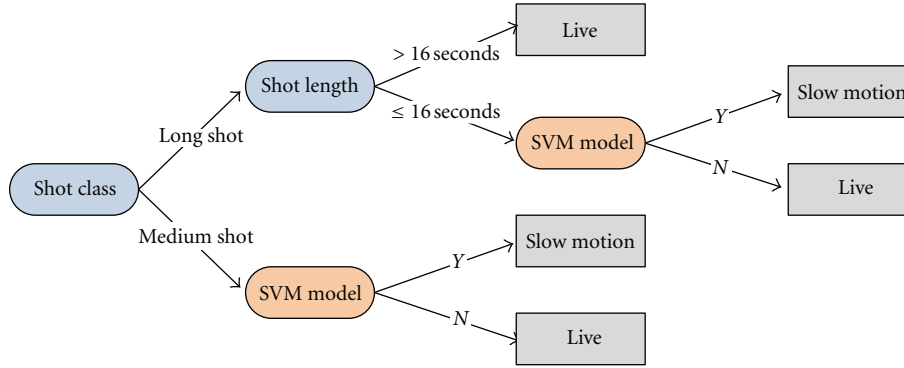


FIGURE 3: Diagram of the proposed classification method for slow-motion detection.

TABLE 3: Experimental results using tagged shot classes.

Data	Video	Medium shots					Long shots					Medium and long shots				
		TP	FP	FN	Precision %	Recall %	TP	FP	FN	Precision %	Recall %	TP	FP	FN	Precision %	Recall %
Train	1	83	1	1	99	99	9	3	9	75	50	229	11	24	95	91
	2	47	4	0	92	100	1	0	9	100	10					
	3	75	3	0	96	100	14	0	5	100	74					
	Total	205	8	1	96	100	24	3	23	89	51					
Test	4	61	20	26	75	70	6	3	6	67	50	167	35	70	83	70
	5	42	4	8	91	84	6	0	3	100	67					
	6	46	8	21	85	69	6	0	6	100	50					
	Total	149	32	55	82	73	18	3	15	86	54					

frame feature, camera motion features, camera motion difference features, and field-side view feature. In the test phase, long shots which last more than 16 seconds are considered as live shots. Shorter long shots are given to the SVM model for classification. RBF kernel function is used in this SVM model.

For medium shots, in the training phase the following features are given to another SVM model: shot length feature, shot type features, grass ratio difference feature, difference of luminance standard deviation feature, object ratio feature, skipped macro-block ratio feature, reliable MV magnitude feature, and motion magnitude fluctuations feature. In the test phase, all medium shots are given to the trained SVM model for classification. RBF kernel function is used in this SVM model.

3. Experimental Results

In this section, we evaluate performance of the proposed method on our soccer video dataset (FUM-BSVD 2011) of six soccer games captured from World Cup 2010. Table 1 shows six soccer games used in our experiments. The first three games are used as training set, and remaining games are used as test set. All slow-motion shots in these videos are produced by high-speed cameras. Each video sequence is encoded to MPEG-1 video format using FFMPEG video

library. Important encoding parameters are summarized in Table 2.

Table 3 shows experimental results of the proposed method on training and test data, respectively. Our algorithm achieved high accuracy of 95% precision and 91% recall on training data. Errors in training data are mainly in long shots. The proposed method achieved satisfactory results of 70% precision and 83% recall rates on test data. Comparing to [11, 12], our algorithm achieved higher accuracy by exploiting more features in slow-motion modeling.

The recall rate for long shots is low. Our experiments indicate that discriminating between slow-motion and live among long shots is a difficult task due to following reasons:

- (1) movement of camera and objects in some live shots is slow,
- (2) slow-motion shots of high-motion scenes contain intensive objects and camera movements,
- (3) a replay shot may also played in live speed.

In order to show robustness of the proposed method against errors in preprocessing stages, a hierarchical shot classification module with overall accuracy of 92% is designed. Shot classes detected by this module are fed into the slow-motion detection module demonstrated in Figure 3. As shown in Table 4, our algorithm achieved accuracy of 90% precision

TABLE 4: Experimental results using automatic shot classification.

Data	Video	Medium shots					Long shots					Medium and long shots				
		TP	FP	FN	Precision %	Recall %	TP	FP	FN	Precision %	Recall %	TP	FP	FN	Precision %	Recall %
Train	1	64	6	5	91	93	13	2	15	87	46	189	20	43	90	81
	2	44	5	3	90	94	1	0	10	100	09					
	3	53	7	3	88	95	14	0	7	100	67					
	Total	161	18	11	90	94	28	2	32	93	47					
Test	4	60	26	26	70	70	6	4	12	60	33	154	44	73	78	68
	5	38	6	5	86	88	5	1	4	83	56					
	6	40	7	17	85	70	5	0	9	100	36					
	Total	138	39	48	78	74	16	5	25	76	41					

TABLE 5: Speed of proposed method.

	Module	Time per frame
T1	Grass modeling	1.4 ms
T2	Color features extraction	19.6 ms
T3	Motion features extraction	16.7 ms
Total = MAX(T1 + T2, T3)		21 ms

and 81% recall on training data by using automatically detected shot classes. In this experiment, the proposed approach achieved accuracy of 78% precision and 68% recall rates on test data. Although such a small loss of accuracy is affordable, it could be reduced by exploiting more accurate shot classification methods.

Our proposed approach is more efficient than previous works due to compressed video analysis. Extraction of color and motion features from compressed video constitutes majority of proposed framework processing time; while training and classification by SVM classifiers is very fast. Table 5 shows processing time of time-consuming modules in proposed framework. Color features extraction and motion features extraction tasks could be done in parallel via multithreading. Using multithreading total processing time of the proposed method is 21 ms per frame or 47 fps which is nearly two times faster than realtime. When these modules run in sequence, the total processing time of the proposed method is 37.7 ms per frame or 26 fps which is faster than real-time. All video sequences in FUM-BSVD dataset are played at 25 fps.

Performance of proposed method on long shots can be improved by extracting better features from video. On the other hand, segmentation and tracking of moving objects in soccer scenes is an alternative way for improving this method. However, this process is problematic in broadcast soccer scenes due to moving camera and moving players.

4. Conclusion and Future Work

In this paper, we proposed an efficient method for slow-motion detection in compressed soccer videos. Exploiting a rich

set of color, motion and cinematic features led to high accuracy of the proposed method. In addition, direct extraction of low-level information from compressed domain significantly improved efficiency of our method. Our framework achieved 83% precision and 70% recall rates on test data. This framework could be used in soccer video summarization and retrieval tasks as a semantic feature extractor.

References

- [1] Q. Huang, J. Hu, W. Hu, T. Wang, H. Bai, and Y. Zhang, "A reliable logo and replay detector for sports video," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 1695–1698, July 2007.
- [2] Z. Dang, J. Du, Q. Huang, and S. Jiang, "Replay detection based on semi-automatic logo template sequence extraction in sports video," in *Proceedings of the ICME 4th International Conference on Image and Graphics (ICIG '07)*, pp. 839–844, August 2007.
- [3] Z. Zhao, J. Shuqiang, H. Qingming, and Z. Guangyu, "High-light summarization in sports video based on replay detection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1613–1616, July 2006.
- [4] E. J. Farn, L. H. Chen, and J. H. Liou, "A new slow-motion replay extractor for soccer game videos," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 8, pp. 1467–1481, 2003.
- [5] Y. Q. Yang, W. Gu, and Y. D. Lu, "An improved slow-motion detection approach for soccer video," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, pp. 4593–4598, August 2005.
- [6] 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.
- [7] W.-H. Chuang, D.-Y. Hsiao, S.-C. Pei, and H. Chen, "A rapid scheme for slow-motion replay segment detection," in *Proceedings of the 5th Pacific Rim Conference on Multimedia*, vol. 3331 of *Lecture Notes in Computer Science*, pp. 239–246, 2004.
- [8] V. Kobla, D. DeMenthon, and D. Doermann, "Identifying sports videos using replay, text, and camera motion features," in *Storage and Retrieval for Media Database*, vol. 3972 of *Proceedings of SPIE*, pp. 332–343, 2000.
- [9] V. Kobla, D. DeMenthon, and D. Doermann, "Detection of slow-motion replay sequences for identifying sports videos,"

- in *Proceedings of the IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 135–140, 1999.
- [10] H. Pan, P. Van Beek, and M. I. Sezan, “Detection of slow-motion replay segments in sports video for highlights generation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1649–1652, May 2001.
- [11] B. Han, Y. Yan, Z. Chen, C. Liu, and W. Wu, “A general framework for automatic on-line replay detection in sports video,” in *Proceedings of the 17th ACM International Conference on Multimedia (MM ’09)*, pp. 501–504, October 2009.
- [12] L. Wang, X. Liu, S. Lin, G. Xu, and H. Y. Shum, “Generic slow-motion replay detection in sports video,” in *Proceedings of the International Conference on Image Processing (ICIP ’04)*, pp. 1585–1588, October 2004.
- [13] Y. Yang, S. Lin, Y. Zhang, and S. Tang, “A statistical framework for replay detection in soccer video,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’08)*, pp. 3538–3541, May 2008.
- [14] J. Wang, E. Chng, and C. Xu, “Soccer replay detection using scene transition structure analysis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, pp. 433–436, March 2005.
- [15] C. Lang, D. Xu, and Y. Jiang, “Shot type classification in sports video based on visual attention,” in *Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC ’09)*, pp. 336–339, June 2009.
- [16] N. Zhenxing, G. Xinbo, T. Dacheng, and L. Xuelong, “Semantic video shot segmentation based on color ratio feature and SVM,” in *Proceedings of the International Conference on Cyberworlds (CW ’08)*, pp. 157–162, September 2008.
- [17] Y. H. Zhou, Y. D. Cao, L. F. Zhang, and H. X. Zhang, “An SVM-based soccer video shot classification,” in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC ’05)*, vol. 9, pp. 5398–5403, August 2005.
- [18] B. Han, Y. Hu, G. Wang, W. Wu, and T. Yoshigahara, “Enhanced sports video shot boundary detection based on middle level features and a unified model,” *IEEE Transactions on Consumer Electronics*, vol. 53, no. 3, pp. 1168–1176, 2007.
- [19] U. Srinivasan, S. Pfeiffer, S. Nepal, M. Lee, L. Gu, and S. Barrass, “A survey of MPEG-1 audio, video and semantic analysis techniques,” *Multimedia Tools and Applications*, vol. 27, no. 1, pp. 105–141, 2005.
- [20] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, “MPEG-1 video syntax,” in *MPEG Video Compression Standard*, pp. 1–470, Kluwer Academic Publishers, 1st edition, 2002.
- [21] H. S. Wong, H. H. S. Ip, L. P. L. Iu, K. K. T. Cheung, and L. Guan, “Transformation of compressed domain features for content-based image indexing and retrieval,” *Multimedia Tools and Applications*, vol. 26, no. 1, pp. 5–26, 2005.
- [22] Z. L. Tao, X. L. Huang, and X. Wang, “Shot boundary detection based on macroblock and DC image,” in *Proceedings of the International Conference on Management and Service Science (MASS ’09)*, pp. 1–3, September 2009.
- [23] X. Wang, S. Wang, and H. Chen, “An effective soccer video shot detection algorithm,” in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE ’08)*, pp. 214–217, December 2008.
- [24] K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, F. Sugaya, and Y. Nakajima, “Shot boundary determination on MPEG compressed domain and story segmentation experiments for TRECVID 2003,” in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID ’04)*, 2004.
- [25] C. Grana and R. Cucchiara, “Linear transition detection as a unified shot detection approach,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 483–489, 2007.
- [26] V. Kiani and H. R. Pourreza, “Robust GME in encoded MPEG video,” in *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia Processing (MoMM ’11)*, pp. 147–154, 2011.
- [27] M. H. Kolekar, K. Palaniappan, S. Sengupta, and G. Seetharaman, “Semantic concept mining based on hierarchical event detection for soccer video indexing,” *Journal of Multimedia*, vol. 4, no. 5, pp. 298–312, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

