3D Ball Motion and Relative Position Feature Based Real-time Start Scene Detection for Volleyball Game Analysis on GPU

Guixing Liang, Xina Cheng, and Takeshi Ikenaga

Graduate School of Information, Production and Systems, Waseda University,
Kitakyushu 808-0135, Japan.

liangguixing95@akane.waseda.jp

Abstract

Volleyball game analysis is essential for extracting useful information in various applications such as quantitative game and TV content. Detecting the start scene of each round of volleyball game is an important task eager to be solved for achieving automatic volleyball game analysis. To achieve high success rate and real-time process, crucial problem including effect of background noise, small difference between motions and redundant calculation of pixel likelihood need to be solved. This paper proposes a 3D ball motion and relative position feature based real-time start scene detection for volleyball analysis. Template update based color likelihood estimation method is proposed to eliminate the ball-like background noise. The relative position between ball and server feature is utilized to distinguish scenes with similar ball motion. Motion estimation is combined to reduce the redundant calculation for GPU acceleration. Experimental results are based on multi-view HDTV video sequences (2014 Inter High School Men's Volleyball Games, Japan). On the GPU device GeForce GTX 1080Ti, we achieve real-time processing in 60 fps sequences. The time consumption of feature extraction scheme is 16.37 ms /frame with a detection recall rate of 97.67% and a precision rate of 100% in average.

Keywords: Start scene detection, Volleyball analysis, GPU.

1. Introduction

Computer vision based sports analysis technologies are necessary in various applications like data mining and tactics analysis⁽¹⁾. In studies on game and performance analysis of volleyball, various efforts have been made to reveal the relationships between some predefined parameters and sports performance, such as player tracking⁽²⁾, action

recognition and event detection⁽³⁾. However, necessitated manual operations in data preparation are demanded in many of such the studies. The test sequences must be intentionally divided into game clips and other clips beforehand. As technology advances, automatic volleyball game analysis has become possible and it is eager for developing. Except for some professional analysis technique, automatically detecting the start scene of volleyball game is required urgently for achieving automatic volleyball analysis. Currently, the most common game start scene detection method is manually cutting, which is obviously unsuitable for analysis automation and limits the on-line practical application. With automatic game start scene detection, we can not only avoid human labor in data preparation, but also do some preprocessing like initialization for ball tracking and player tracking.

For automatic start scene detection, a related work utilizes player feature based method to detect the start scenes⁽⁴⁾ in actual badminton matches whose background is much simpler than the one of volleyball. This work firstly utilizes background subtraction method to locate players' position. Based on players' position, cubic higher-order local auto-correlation (CHLAC) is employed to represent features of postures and motions of players. Then, the serve scenes are detected by linear regression (MRA) from the CHLAC features. However, this work is based on sequences captured by a single ceiling camera. When it comes to the volleyball game, the circumstances become very complexed and many noises from the background occur. There are more players in the court and occlusion becomes a very serious problem and it is hard to locate players' position.

Cheng⁽⁵⁾ has proposed an automatic multi-view 3D ball recovery scheme in volleyball game, which can automatically detect the start frame and initialize the initial 3D ball position of tracking. This work uses multi-view

Fig. 1. Start scene detection feature extraction framework.

information, which is more robust to the occlusion problem and effective to the background noise problem. Moreover, from the real-time point of view, the key algorithms of likelihood calculation in this work are real-time oriented and the tracking system based on it has been implemented into real-time⁽⁶⁾. However, the tracking start scene is not the real start of game. This detection scheme only detects one start frame for tracking. For game start scene detection, ball motion information should be utilized. This detection scheme does not consider any inter-frame information and every frame is detected separately. In this case, the ball detection result of each frame has no relationship with other frames and it is sensitive to some ball-like noise which leads an unstable ball motion result. Scenes of volleyball game are very complex, and some different event scene can have similar ball motion. Distinguishing the start scene from scene with same ball motion is another difficulty for start scene detection of volleyball. Moreover, the large calculation amount is also a challenge for achieving real-time since it needs to scan and calculate the likelihood of enormous number of pixels.

In this paper, a 3D ball motion and relative position feature based real-time start scene detection for volleyball analysis is proposed. The template update based color likelihood estimation method solve the unstable result of ball motion problem. Meanwhile, the relative position between ball and server feature is utilized to distinguish the start scene from other scenes with similar ball motion. Motion estimation method is combined to reduce redundant calculation for acceleration. In the experiment, multi-view HDTV video sequences (2014 Inter High School Men's Volleyball Games, Japan) are utilized. The experiment results demonstrate the effectiveness of the proposed methods and shows the high accuracy and high process efficiency of our system.

This paper is arranged as follows. Section 2 covers the

framework of volleyball game start scene detection and the detail of the proposed methods. The experiments and the conclusion are described in section 3 and section 4.

2. Start scene detection methods

2.1 Framework

The 3D ball motion and relative position feature based real-time start scene detection system is implemented in our research. This system includes two steps. First is extracting feature from the input frames to generate feature vectors. Then, based on the feature, a classifier is trained and used for prediction. The feature extraction framework is shown in Fig.1.

In this system, All the pixels in the input 2-view video frames are raster scanned with a certain interval. $L(p; m)$ which is defined as pixel likelihood value of the scanning pixel p in camera m will be assign to the pixel to represent the probability of this pixel being the ball center. For each scanning pixel, we calculate its circle likelihood $L_{circle}(p; m)$, HSV color likelihood $L_{color}(p; m)$ and foreground likelihood $L_{foreground}(p; m)$, which are based on the shape, color, and foreground feature of volleyball⁽⁵⁾. Thus, the pixel likelihood $L(p; m)$ is represent as:

$$L(p; m) = L_{circle}(p; m) \cdot L_{color}(p; m) \cdot L_{foreground}(p; m). \quad (1)$$

The circle likelihood represents the shape feature of volleyball to distinguish circle object from other non-circular noise. Color likelihood is evaluated by comparing the distance of HSV color histogram between the sampled region around the scanning pixel and the prepared N volleyball templates. The foreground likelihood is for distinguishing the moving object from the background. The foreground likelihood is calculated from a binary mask image generated

Fig. 2. Template update method based color likelihood estimation

by background subtraction method.

After likelihood calculation, the pixel with a high likelihood value is chosen as the initial 2D ball candidates. For each initial ball candidate, the isolate coordinate judgement is applied using the evaluation method below:

$$w_i = \begin{cases} 1, & \text{is candidate} \\ 0, & \text{is not candidate} \end{cases} \quad (2)$$

$$\sum_{i=1}^{n \times n} w_i > thr_{moving}, \quad (3)$$

where w_i indicates whether the pixel i in the $n \times n$ search region is a candidate or not. thr_{moving} is the threshold to judge whether the center pixel of the search region becomes an isolated candidate.

With isolate coordinate judgement, the final 2D ball candidates can be filtered out by removing the isolated coordinates which are obviously erroneous candidates of 2D positions. The final 2D ball candidate will be projected into 3D space twice in different order to generate some 3D ball candidates. The detail 3D ball candidates evaluation method is mentioned in the conventional work⁽⁵⁾. The final 3D ball position is evaluated according to the special density and the distance of the 3D candidates. The group of 3D candidates with highest special density will be selected as the candidate group. And the point has smallest distance with other candidate points will be selected as the final 3D ball position.

Based on the 3D ball position, the region around the corresponding 2D ball position $P_b(m)$ of camera m will be sampled as a new template and updated into the template library. The server's 2D positions $P_{server}(m)$ of camera m which are used for relative position feature are also calculated around $P_b(m)$.

Fig. 3. Color histogram of new template from frame #56188, ball in frame #56189 and the prepared fixed template

The ball's 3D positions of past k frames and the relative positions are combined as the feature vector. Then, based on the feature vector, a classifier is trained and used for prediction on every frame.

2.2 Template update method based color likelihood estimation

For the situation that there are some similar noises in the complex background, the conventional likelihood estimation method utilizes shape feature color feature and foreground feature to detect candidate ball position in the input frames. The estimation method detects each frame separately without considering the relationship between frames. Sometimes there are some noises having a comparable likelihood with the target ball and it is hard to distinguish which is the ball we should detect. In this case, wrong detection of ball will occur which leads to the unstable detection of ball motion. For example, at the start of volleyball match, the staffs will pass the candidate ball to each other to prepare for the next round. So, the passing candidate ball becomes a noise with comparable likelihood with the target ball. To deal with such problem, the template update method based color likelihood estimation method is proposed. The template update method updates the detected

Fig. 4. Relative position feature

ball in the processing frame to the new templates library for the likelihood calculation of next frame. Combining the new templates, the ball's likelihood is enlarged like Fig. 2 shows and ball becomes more distinguishable from other noise.

Once a 3D ball position is detected, the region around $P_b(m)$ in the frame of camera m will be sampled as a new sample. We recalculate the likelihood of this sample and evaluate whether update it as a reference template or not. The likelihood calculation and evaluation methods are shown as below:

$$L_{sample} = 1 - e^{-\frac{1}{N} \sum_{n=1}^N d_n}, \quad (4)$$

$$L_{sample} > thr_{color}, \quad (5)$$

where d_n is the Bhattacharyya distance between the sampled color histogram of the observation region and the n_{th} histogram of the prepared templates.

The updated templates will be combined with the prepared templates for the color likelihood calculation of next frame. Thus, the color likelihood is represented as:

$$L_{color}(p; m) = 1 - e^{-\frac{1}{N+N_u} \sum_{n=1}^{N+N_u} d_n}, \quad (6)$$

where N_u is the number of new templates which is not greater than N . With template update method, information of ball's status in the last frame is utilized. Like the example shown in the Fig. 3, this figure shows the color histogram of the ball in the processing frame#56189, new updated template from last frame#56188 and one of the prepared fixed templates. the new templates updated from the frames close to the processing frame have higher similarity with the target ball. The likelihood of the target ball is enlarged and the noise will be eliminated. In this case, the target ball and noise are more distinguishable. Once a ball is detected, the probability of detecting this ball again in the next frame becomes higher. This makes the detection result of ball motion more stable.

Fig. 5. Relative position feature calculation

2.3 Relative position between ball and server feature

Ball trajectory represents the ball motion feature of volleyball game even, which is useful to distinguish most of the event scenes like passing and spiking. However, some scenes can have similar ball motion feature with serving like patting and throwing, which are generally existing before serving as the Fig. 4 shows. Part of the ball motion of patting is similar with serving ball motion. Only using ball motion feature is hard to distinguish these scenes. However, the relative position between server and ball feature is different. Combining relative position feature make it easy to distinguish ball motion of serving from other noise cases.

The serve's position calculation is based on the result of ball detection. After ball detection, an ROI (region of interest) around the 2D ball position in the frame will be generated. As Fig. 5 shows, in this ROI, color filter and background subtraction mask are used to generate a server mask. Color filter is utilized to distinguish the server from the audiences and the staffs. The background subtraction mask extracts server from the background and avoid the noise from the static background. According to the server mask, the 2D server position is calculated by the equation shown as below:

$$P_{server}(m) = \frac{\sum_{i=1}^{AP(m)} c_i}{AP(m)}, \quad (7)$$

here, $AP(m)$ is the available pixels in the server mask of camera m , and c_i is the coordinate of the available pixel i . The available pixels mean the pixels whose corresponding value in the server mask is not zero. After getting the server's position $P_{server}(m)$, the relative position feature $P(m)$ in the frame of camera m is represented as:

$$P(m) = P_b(m) - P_{server}(m). \quad (8)$$

Fig. 6. Motion estimation based redundancy removal

2.4 Motion estimation based redundancy removal

In the likelihood calculation, the input frames are raster scanned with a certain interval. Obviously, there are many redundant calculations on the background pixels. To reduce redundant calculation on the background pixels, the motion estimation based redundancy removal is combined before likelihood calculation.

In the motion estimation, the background subtraction mask which is prepared in the preprocessing part is used. As Fig. 6 shows, an $n \times n$ search window is used to do motion estimation on the mask for every pixel in a certain interval. After the estimation, for the scanned pixel p of camera m , an observation value $O(p; m)$ is assigned to it.

$$O(p; m) = \sum_{i=1}^{n \times n} \frac{v_i}{255} > thr_{moving}, \quad (9)$$

here v_i is the i_{th} pixel value in the background subtraction mask of camera m . By setting a threshold $> thr_{moving}$, this value is used to evaluate whether this pixel is selected as the target pixel. In our experiment the threshold is set as 1.

After the evaluation, only target pixels are involved to do further calculation. Since the foreground likelihood is included in the likelihood calculation and the calculation complexity of motion estimation is lower than the likelihood calculation, the number of pixels need to calculate is reduced after motion estimation. Since the foreground likelihood which is calculated from the background subtraction mask is included, the foreground likelihood of pixels without motion should be zero. Those pixels with zero likelihood will not be selected as ball candidates and the likelihood calculation of these pixels is redundant. After motion estimation, pixels without motion are reduced while pixels with motion are not affected. In this case, the motion estimation method is useful for acceleration without decreasing the detection accuracy.

Fig. 7. Examples of typical scenes in test sequences

3. Experiment

3.1 Experiment environment

This system is implemented on a host computer, whose CPU is Intel i7-6700K, 4.01GHz, with 32GB memory size. And a NVIDIA GeForce GTX1080Ti GPU device. For the third-party tools we use OpenCV 2.4.11 and CUDA 8.0. all program is written in C++ and CUDA C language. In our implementation, the frames data are transferred from host computer to GPU device. After feature extraction process finished, the 3D ball position and server's 2D position are copied back to the host device to generate the final feature vector. Training model is off-line process, then the trained module is used to do prediction for every frame and the prediction procedure is much less than 1ms/frame. So, we only focus on the feature extraction part. If needed, there are also some classifiers implemented on GPU⁽⁷⁾. Six kernels which are corresponding to the framework mentioned in section2 are included in our GPU implementation framework. K1: Preprocessing, is designed to covert the input RGB images into HSV and gray images. It also prepares the background subtraction mask and Sobel images for next processing. K2: Motion estimation which is corresponding to the motion estimation based redundancy removal method, selects the target pixels for likelihood calculation. K3: Likelihood calculation calculates the likelihood of the target pixels and generates ball candidates for evaluation. K4: Ball candidate evaluation evaluate candidates from K3 and gets the final 3D ball position and the corresponding 2D position in the camera frames. K5: Template update calculate the likelihood of the sampling region around the 2D ball position and updates the new sample to the new template

Fig. 8. Evaluation of one round

Table 1. Detection accuracy performance.

Criterion	C1 and C2		C3 and C4	
	Recall	Precision	Recall	Precision
Conventional ⁽⁵⁾	70.45%	81.58%	73.81%	81.58%
P1	86.36%	90.47%	90.47%	92.68%
P1+P2	95.45%	100%	100%	100%
P1+P2+P3	95.45%	100%	100%	100%

library for the calculation of next frame. K6: Relative position calculation calculate relative position between ball and server in 2D space.

3.2 Test dataset

In our experiment, the input of our test sequences are the whole video streams without any intentionally cutting from an official volleyball match (Semi-final Game of 2014 Japan Inter High School Game of Men Volleyball in Tokyo Metropolitan Gymnasium) which is a real match of high quality. The videos are taken by 4 cameras (C1, C2, C3, C4) synchronously at 4 corners of the game court. C1 and C2 are on the same side while C3 and C4 are on the same side. The shutter speed to make sure there is no motion blur in our sequences. The resolution is 1920×1080 and the frame rate is 60 fps and the dataset has more than 160000 frames for each view. There are totally 86 rounds of games in our sequences which covers a variety of event scenes like patting, passing, throwing, bouncing, serving, reparation scenes and so on. Moreover, same event with different appearances such as jump serve, normal serve is also included in our sequences like Fig. 7 shows. Experiment indicates that even in such a complex situation our system still works well. For start scene detection, we only use cameras of the opposite to extract feature, since the serving region is only visible by the

Table 2. Time consumption result.

	CPU+P1+P2 (ms/frame)	GPU+P1+P2 (ms/frame)	GPU+P1+P2+P3 (ms/frame)
Preprocessing	164	1.15	1.15
Motion estimation	Not included	Not included	4.59
Likelihood calculation	1038	19.58	8.28
Candidate evaluation	63	1.57	1.57
Template updating	31	0.22	0.22
Relative position calculation	96	0.56	0.56
SUM	1392	23.08	16.37

cameras of the opposite. Cameras of the serving side are unavailable due to the occlusion and out-of-frame problems. Thus, there are 44 rounds detected by C1 and C2, while 42 rounds are detected by C3 and C4.

3.3 Evaluation method

For evaluation of detection performance, we watch the sequences and manually label 10 frames in the start period of each round as the ground truth (start frames), which is from the server throws the ball to the ball reaches the top. Classifier does prediction on every frame. The evaluation method for detection of round is shown in Fig. 8, if there are continuous 7 frames predicted as the start frames, we regard this situation as one round is detected. And if there is at least 1 frame in these continuous 7 frames overlapping with the ground truth, we treat this detected round is a correct detection, otherwise it is a wrong detection since there is no overlapping with the ground truth.

For experiment, we separate the sequences into two parts, part 1 and part 2. Each part includes 43 rounds of start frames of the start scene which include 430 frames. These two parts become the positive samples for the classifier. The negative samples for training are randomly selected from others which are not the start scenes to solve the unbalance of positive and negative sample problem. Our experiment results are obtained in two steps. First, we use part 1 and randomly select 4300 frames from other not start scene as negative samples to train the Random Forest classifier. Samples of part 2 are used as test samples for prediction. Then, part 2 of the positive samples and the randomly

selected negative samples are used for training. After training, part 1 is used for prediction. Each step is repeat 3 times since the negative samples are randomly selected. For detection of each round, once one round is detected, the system will stop detection for several frames assuming it is in the game period. After several frames, detection for the next round starts.

Two standard evaluation criteria of recall and precision are used to evaluate detection performance of the start scene detection system. Adapting to our target, the definitions of recall and precision for our target are shown as below:

$$precision = \frac{correctly\ detected\ rounds}{detected\ rounds}, \quad (10)$$

$$recall = \frac{correctly\ detected\ rounds}{total\ rounds}. \quad (11)$$

Specifically, for our target, the recall rate evaluates how many rounds of game our system can correctly detect in the total round exist in our sequence. The precision rate represents correct detection rate in the detected rounds of the system.

3.4 Experiment result and consideration

In the experiment result of table1, P1 means template update method based color likelihood estimation while P2 is the relative position between ball and server feature. Conventional work⁽⁵⁾ reaches about 70.45% recall on C1&C2 and 73.81% recall on C3&C4, which means it failed to correctly extract the ball motion of some rounds from the frames and the motion result is sensitive to the background noise. Proposal 1 achieved a 86.36% recall on C1&C2 and 90.47% recall on C3&C4, which is much higher than conventional work⁽⁵⁾. Reason of this proposal works well is that the proposal enlarges the likelihood of the target ball in the frame and makes it stable to detect ball motion and will not easily affected by the background noise and randomly detect somewhere else. And the proposal 3 will not affect the detection accuracy. However, in proposal 1, the precision rate doesn't reach 100%, which means the classifier cannot distinguish some noise motion like patting and throwing which have similar ball motion with serving. And the classifier wrongly predicts those indistinguishable situations as serving. After combining proposal 2, the relative position between ball and server feature, the precision reaches 100%. That means the relative position feature makes those similar ball motion situations distinguishable. Moreover, the relative position feature also represents the relative motion between ball and server of start scene, which is helpful to provide more information for start scene detection. In this case, the

recall rate is also improved. However, the recall rate of C1&C2 still cannot reach 100%. That's because of the occlusion problem. In these failed cases, the ball is occluded by the referee in one view for several frames. In this case, we failed to construct 3D ball position since there are only two cameras used to do detection. And we cannot extract an available feature vector from the frame. This leads to the failed detection.

For the time consumption performance which is shown as table 2, CPU+P1+P2 means the feature extraction algorithms are implemented in CPU including proposal 1 and proposal 2. GPU+P1+P2 is a GPU version with proposal 1 and proposal 2 implemented. GPU+P1+P2+P3 means the implementation combining proposal 1, proposal 2 and proposal 3. The time consumption CPU version is 1392 ms/frame. After implementation in GPU without proposal3, the time cost is 23.08ms/frame. After proposal 3 is implemented, the time cost becomes 16.37 ms/frame which achieves real-time processing for 60 fps videos. Although motion estimation also needs to process all the pixels in a certain interval. But the estimation calculation is much simpler than likelihood calculation. After motion estimation, the number of pixels for likelihood calculation is reduced. So, the total time consumption is decreased. Combining the accuracy result and time consumption result, a conclusion that the proposal 3 works well for acceleration without affecting the detection accuracy can be drawn.

4. Conclusions

In this paper, a ball motion and relative position feature based real-time start scene detection system for volleyball game analysis is proposed. The template update method based color likelihood estimation method combines information from past frames to eliminates some ball-like background noise. This method updates detected ball region in past frame which have high similarity with ball in current frame as a new template for likelihood calculation of current processing frame. With templates updated, once a ball is detected, the risk of losing the ball decreases and the probability of keeping the success detection of the ball in the next frame becomes higher. This makes the detection result of ball motion more stable. Within a stable ball motion, the success detection rate of start scene is increased. The relative position between server and ball feature represents the relative motion between ball and server and makes the difference between different event scenes larger. This feature is generated from ball's position and server's position. It is

designed to provides more representative information from the input sequences and makes serving scene more distinguishable from other event scenes with similar ball motion. We also implement the algorithm on single NVIDIA GeForce GTX1080Ti GPU device for acceleration. For reducing the time cost in likelihood calculation part, considering foreground likelihood is included in the likelihood calculation, the motion estimation based redundancy removal method is combined to select target pixels for likelihood calculation and reduce the redundant calculation on the background pixels. With the motion estimation based redundancy removal method, the total time consumption is decreased and the detection accuracy is not affected. As the experiment result shows, our detection system reaches a 97.67% detection recall rate with the precision rate of 100% in average. And, the redundancy removal method is useful for acceleration. The final processing speed is faster than 16.37 ms/frame which achieves real-time on 60 fps sequences.

For the future work, two aspects including algorithm improvement and real-time volleyball analysis system building are under considering. On the one hand, for the improvement of algorithm design, a method which is robust to the occlusion problem should be considered. Applying the start scene detection framework on more datasets or extending the start scene detection framework to other sports like beach volleyball and tennis helps to achieve a robust start scene detection system. On the other hand, for the real-time volleyball analysis system construction, since current classifier used is based on CPU, combining the implemented feature extraction framework with the GPU based classifier, a completed real-time volleyball start scene detection system is achieved on GPU. What's more, combining the start scene detection work with other volleyball analysis studies like ball tracking and player tracking, an automatic volleyball game analysis system is possible to be realized.

Acknowledgment

This work was supported by KAKENHI (16K 13006) and Waseda University Grant for Special Research Projects (2018K-302).

References

- (1) G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao, "Event tactic analysis based on broadcast sports video," *IEEE Trans.Multimedia*, vol. 11, no.1, pp.49-67, 2009.
- (2) X. Zhuang, X. Cheng, S. Huang, M. Honda, and T. Ikenaga, "Motion vector and players' features based particle filter for volleyball players tracking in 3D space", *Pacific Rim Conference on Multimedia*, pp. 393-401, Springer, 2015.
- (3) Xina Cheng, Norikazu Ikoma, Masaaki Honda, Takeshi Ikenaga, "Ball State Based Parallel Ball Tracking and Event Detection for Volleyball Game Analysis", *IEICE Trans. Fundamentals*, vol. E100-A, No. 11, pp. 2285-2294, Nov. 2017.
- (4) Y. Fumito, K. Takumi, W. Kenji, and O. Nobuyuki, "Automated service scene detection for badminton game analysis using chlac and mra", in *World Academy of Science, Engineering and Technology*, vol. 62, 2010.
- (5) Xina Cheng, Norikazu Ikoma, Masaaki Honda, Takeshi Ikenaga, "Multi-View 3D Ball Tracking with Abrupt Motion Adaptive System Model, Anti-Occlusion Observation and Spatial Density Based Recovery in Sports Analysis", *IEICE Trans. Fundamentals*, vol. E100- A, No. 5, pp. 1215- 1225, May 2017.
- (6) Yilin Hou, Xina Cheng, Takeshi Ikenaga, "Real-time 3D Ball Tracking with CPU-GPU Acceleration Using Particle Filter with Multi-command queues and Stepped Parallelism Iteration", *Second International Conference on Multimedia and Image Processing (ICMIP 2017)*, Mar. 2017
- (7) A. Carpenter, "CUSVM: A CUDA Implementation of Support Vector Classification and Regression", patternsonscreen.net/cuSVMDesc.pdf, 2009

- (1) G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao, "Event tactic analysis based on broadcast sports video," *IEEE Trans.Multimedia*, vol. 11, no.1, pp.49-67,