

Paper

Keep Your Eye on the Ball: Detection of Kicking Motions in Multi-view 4K Soccer Videos

Jianfeng Xu (member)†, Kazuyuki Tasaka (member)†

Abstract For automated game analysis, it is essential to detect the kicking motions of players in soccer videos in order to understand each player's actions. This paper presents a fast and accurate approach to detecting kicking motions with a ball-centric window in multi-view 4K soccer videos. Based on powerful object detection techniques like SSD or YOLOv3 and pose estimation techniques like OpenPose or CPN, we propose novel solutions to overcome two challenges in 4K soccer videos. The first challenge is that it is basically too computationally heavy to process the massive amount of data in multi-view 4K videos. The solution to this challenge is that we only process a small portion (i.e. a ball-centric window) of 4K video, benefiting from an object tracking technique and homography transformation. The second challenge is that kicking motions may be incorrectly detected due to two factors. One is the absence of depth information and the other is the inaccuracy of pose estimation. We fuse multiple views to avoid the depth problem. In addition, we propose enlarging the person areas to effectively improve the accuracy of pose estimation. The experiments on real data from the J1 League demonstrate that the proposed approach achieves both faster and more accurate detection of kicking motions than conventional methods.

Key words: sports motion analysis, multi-view detection, object detection/tracking, pose estimation, action detection.

1. Introduction

4K videos are becoming more accessible to consumers than ever before thanks to the release of many 4K consumer cameras and smart phones that support 4K video recording (e.g. iPhone6s and later models). Moreover, 4K content is available including the FIFA World Cup 2018 in 4K and streaming services from Netflix and YouTube. In addition, Strategy Analytics predicted that more than half of U.S. households are expected to have 4K-capable TVs by 2020[1]. Although we are entering a 4K video era, which provides better user experience, it takes much more time to process 4K videos than the lower resolution videos such as HD or SD. Furthermore, it obviously becomes even more challenging to process multi-view 4K videos with a limited computational resource[2].

In order to understand each player's actions in soccer games[3][4], many techniques are necessary such as player identification and highlight/event detection[5]. Among these, we focus on detecting the *kicking motions* of players, which are defined as from the time a player touches the ball and changes the speed or direction of the ball's movement. Such kicking motion detection also serves as a basis for statistical data analysis in soccer games such as ball possession, passes, shots, assists, saves, dribbling, and crosses. The detection of kicking motions basically requires detection of the soccer ball and players' poses[4] (especially the foot positions). As we know, soccer balls and human poses can be detected by deep learning approaches such as SSD[6]/YOLOv3[7] and CPN[8]/OpenPose[9]. Basically, the computational time for inference will increase as the resolution of the input layer in the neural network increases[10]. Therefore, it is still very challenging to detect objects and estimate poses with a limited computational resource in multi-view 4K soccer videos (30 fps) as shown in Fig. 1 because of the massive amount of data.

For object detection, a naive approach with high accuracy involves detecting objects with a sliding window (e.g., 416x416 in YOLOv3[7]), which has a very high resource demand. Another naive approach with fast processing is to resize each frame to a small one, which will result in low accuracy with many missing objects because the objects (especially the ball) in soccer videos are rather small as shown in Fig. 1. Basically, the processing area directly determines the processing time. Therefore, it is essential to reduce the processing area while retaining high accuracy. In this paper, we will accelerate object detection significantly by only

Received August 26, 2019; Revised December 16, 2019; Accepted January 16, 2020

† KDDI Research, Inc.
(Saitama, Japan)

This paper includes videos. Note that the videos are not viewable from this PDF file. The videos are available as separate files on the website that hosts this PDF file.

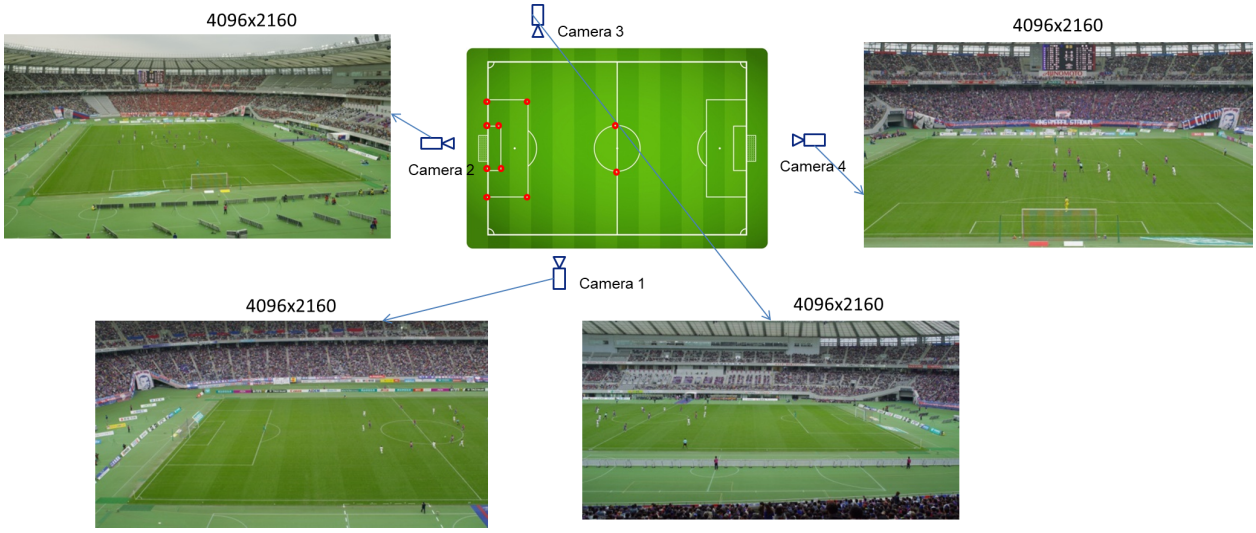

Fig. 1 Four synchronized 4K cameras are set around the soccer field to capture play in 30 fps during soccer matches. Red circles in the middle of top row denote the corresponding points used to calculate the parameters of homography transformation between cameras. The digits denote the width (pixel) x height (pixel).

processing the ball areas in multi-view videos (called *ball-centric window* in this paper). It is natural to focus on the ball area because the soccer ball is a prerequisite of the kicking motion. Moreover, a ball-centric window is commonly employed in most videos showing the highlights of soccer games, which actually dovetails with the end user's preference.

For pose estimation, we have observed that it is rather challenging to estimate pose accurately in our soccer videos because the person areas are very small with low image quality and motion blur. Fortunately, we have observed that person detection is rather accurate even in the original person areas and pose accuracy can be effectively improved simply by enlarging the person areas. In this paper, we use the object detection results from YOLOv3[7] and double the width and height of the bounding boxes of person areas for pose estimation (OpenPose[9]).

In addition, with a single view, the ball close to a player may not indicate that their physical distance is close due to the absence of depth information. This makes it difficult to avoid false positive detections from a single view. In this paper, we use multiple views with four synchronized cameras (as shown in Fig. 1) to greatly improve the accuracy of kicking motion detection. In a J1 League soccer match, the F-measure of kicking motion detection is significantly improved from 0.64 (single view) to 0.85 (four views).

Briefly, our main contributions are as follows.

- As far as we know, this paper presents for the first time an approach to detecting kicking motions in multi-view 4K soccer videos, which is both fast and accurate by overcoming two major challenges.
- As the solution to the first challenge, the proposed approach focuses on a ball-centric window to improve efficiency based on an object detection/tracking technique and homography transformation.
- For the second challenge, to improve the accuracy of pose estimation, we double the width and height of the person areas. To overcome the absence of depth information, we merge multiple views to detect any kicking motion. The experimental results obtained based on real data from a J1 League game demonstrate that the proposed approach achieves a high degree of accuracy in terms of kicking motion detection.

The remainder of this paper is organized as follows. After a brief survey of related work in Sect. 2, we describe the proposed approach in detail in Sect. 3. Sect. 4 reports our experimental results, followed by our conclusions in Sect. 5.

2. Related Work

Because we found no literature on detection of kicking motions in multi-view 4K soccer videos, we briefly survey techniques related to our approach, i.e., object detection, pose estimation, and the latest sports analysis techniques. There are survey papers that fully cover the object detection field[11] and sports analysis field[5].

2.1 Object detection

As a state-of-the-art technique in object detection,

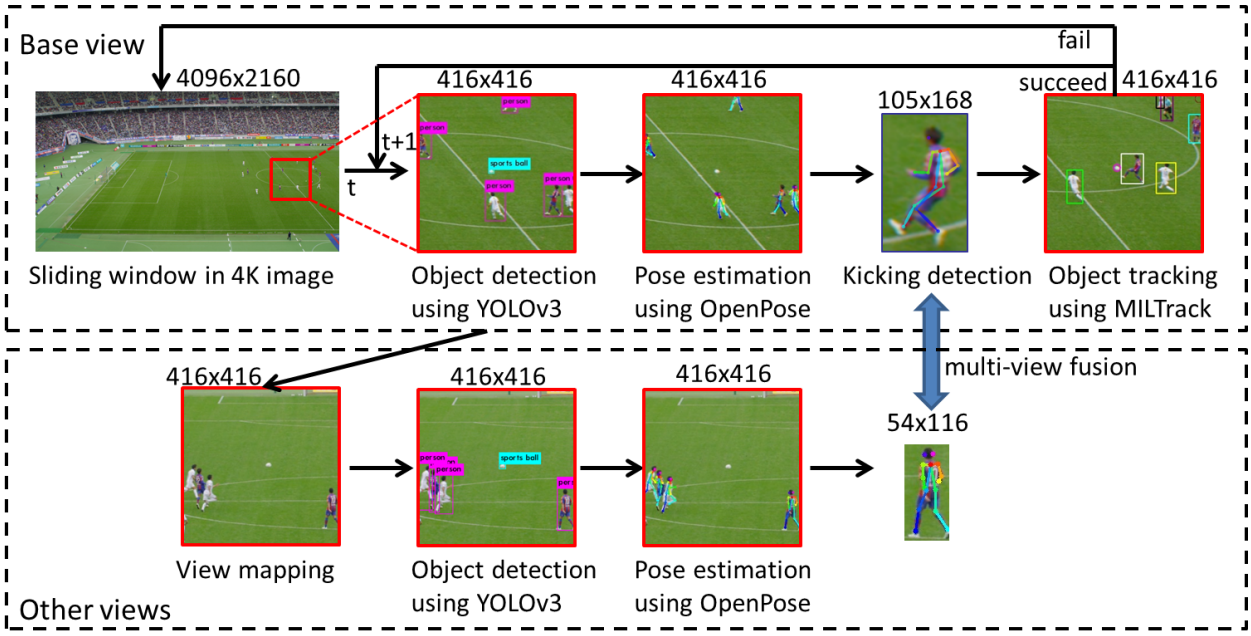

Fig. 2 Detection flow for kicking motions in multi-view 4K soccer videos. The digits denote the width (pixel) x height (pixel).

faster R-CNN[12] used a two-stage object detection strategy. It achieved very high detection accuracy within the most commonly used datasets such as COCO. However, the computational cost was still high, basically requiring faster R-CNN to run in GPU mode. Therefore, single shot detectors such as the YOLO family of object detectors[7][13][14] and SSD[6] were proposed, which ran much faster with reasonably high accuracy[15]. YOLO[13] was an end-to-end single convolutional neural network that detected objects based on bounding boxes prediction and class probabilities. However, it was still difficult to detect small-sized objects and achieve precise localization. Thereafter, SSD[6] was proposed for improving the YOLO-based method. With the introduction of multi-scale feature maps and the default boxes mechanism, SSD[6] can detect small-sized objects and also improve localization accuracy compared with YOLO[13]. On the other hand, YOLO has evolved and solved its initial problem. The latest version, YOLOv3[7], carries out detection on three different scales and utilizes a more powerful deep architecture that has more layers with residual blocks.

2.2 Pose estimation

The popular multi-person pose estimation techniques can be categorized into the top-down and bottom-up approaches. The top-down approaches such as CPN [8] and AlphaPose[16] first detect person areas with a bounding box using, for example, faster R-CNN[12] in AlphaPose, then estimate the human pose in each detected person area. These approaches are highly sensitive to the accuracy of the person detector and are known to have difficulty in estimating poses of persons where there is occlusion[17]. The bottom-up approaches such as OpenPose[9] first predict the heatmap of all body joints, then connect the body joints for each person. One additional advantage of bottom-up approaches is that there is little change in the computational cost regardless of the number of persons[9], which makes the method suitable for soccer games.

2.3 Sports analysis

As presented at five workshops on computer vision in sports held at CVPR or ICCV since 2013, many vision based approaches[18] were proposed to analyze ball possession[19][20], ball/player trajectory[21], player identification[22], pose estimation for sports action recognition[4] and activity recognition in sports applications[3]. The papers in the workshops also reported that a deep learning approach like[23] was particularly valuable. For example, the players and balls were detected using YOLO 9000[14] in[20] and other deep learning approaches[24].

A challenge in 4K sports videos is real-time processing, which is a basic requirement of such applications as live broadcasting[25] and AR/VR[26]. Resizing 4K videos to make them smaller is an efficient way[10][25] but may cause great loss of accuracy especially in the case of small objects. Another way is parallel computing, which directly requires more high-spec hardware[25]. This issue obviously becomes more serious for our multi-

view 4K soccer videos. To the best of our knowledge, few studies have investigated detection of kicking motions in multi-view 4K soccer videos[5], where it is essential but challenging to design a fast and accurate approach.

3. Proposed Approach

In this section, we will describe the main functions after a framework overview of our proposed approach.

3.1 Framework overview

Fig. 2 shows the flow of our approach, where we select camera 1 in Fig. 1 as our base view. First, we detect the soccer ball and players in a 416x416 region around the ball (i.e. a ball-centric window) in the base view. Second, we estimate the human poses from the detected bounding boxes of person areas, which are also enlarged for better pose estimation. Third, we detect the kicking motions by the distance between the ball and players. However, there are many false positives if only the base view is used. Therefore, we utilize other views as well. For these views, we calculate the parameters of homography transformation between cameras in advance. By doing this, we can get the ball-centric window using the ball positions mapped from the base view. Similar to the base view, we also detect the ball and players, estimate the human poses, and calculate the distance between the ball and players in other views. Therefore, we can fuse the information from all the views to detect kicking motions. In addition, to process the next frame, we track the soccer ball and players in the ball-centric window in the base view to accelerate processing of the next frame.

3.2 Object detection

Among many choices such as faster R-CNN[12], YOLOv3[7] and SSD[6], we selected YOLOv3 because it is fast and easy to use. The pre-trained model * is directly used without any fine tuning. This paper strives to reduce the processing area in object detection. The method is described as follows.

For the base view, when the first frame is input as shown in Fig. 2, we detect the ball and players from the entire frame with a resolution of 4096x2160 by a sliding window of 416x416. The only trick we use to improve efficiency is to mask out non-field areas, where the pixels out of the soccer field are set as zeros. Note that if there is no ball detection in the first frame, we will skip the frame and conduct the detection in next frames until we get a ball detection. After ball detection with YOLOv3[7] in a sliding window, we refine the position of the processing area with the center of the detected ball as a ball-centric window as shown in Fig. 2. For other frames, when the tracking succeeds, we only focus on a ball-centric window with the center of the tracked ball without sliding windows of the entire frame. If the tracking fails, we will conduct the same detection processing as the first frame. On the other hand, for other views, we basically do the same processing except that the ball-centric window comes from the view mapping using homography transformation, as shown in Sect. 3.5.

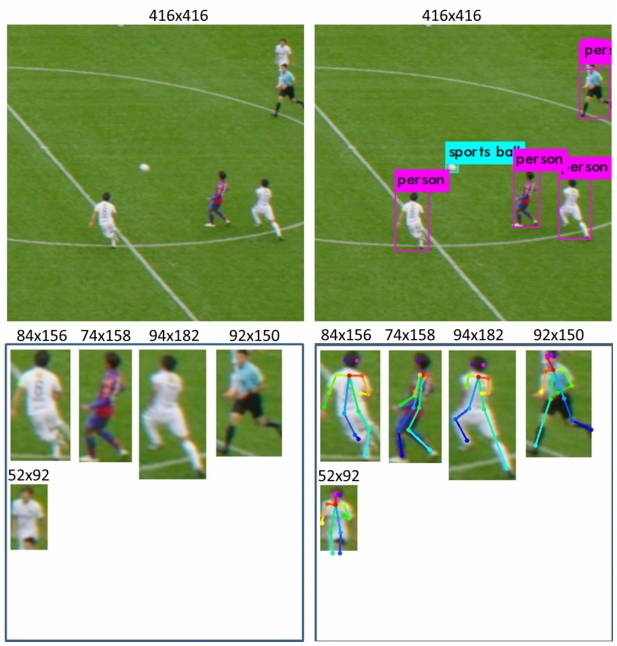

Fig. 3 Pose estimation procedure. Top-left: ball-centric window, top-right: object detection results, bottom-left: enlarged person areas, bottom-right:pose estimation results. The digits denote the width (pixel) x height (pixel).

The output of object detection is the detected bounding boxes of ball and players for each frame in each view. In this paper, the top-left corner and the bottom-right corners of the bounding box of the ball for the t-th frame in the v-th view are denoted as $(x_b^l(t,v), y_b^l(t,v))$, $(x_b^r(t,v), y_b^r(t,v))$ respectively. Similarly, the top-left corner and the bottom-right corners of the bounding box of the i-th person for the t-th frame in the v-th view are denoted as $(x_p^l(i,t,v), y_p^l(i,t,v))$, $(x_p^r(i,t,v), y_p^r(i,t,v))$ respectively.

3.3 Pose estimation

For both the base view and other views, we use OpenPose[9] to estimate the human poses. The pre-trained

* The pre-trained model is available at `https://pjreddie.com/media/files/yolov3.weights` on Aug. 26, 2019.

Table 1 Some statistical data of human size in training data of COCO dataset.

	width (pixel)	height (pixel)	area (pixel^2)
mean	88.42	133.87	22284.61
standard deviation	111.49	130.31	43375.83

model ** is directly used without any fine tuning. Note that the pre-trained model is trained on COCO dataset. Some statistical data of human size in training data are shown in Table 1, which is calculated from the bounding boxes in ground truth. The output of pose estimation is the joint positions of all detected players in Sect. 3. 2, where the j-th joint position of the i-th person for the t-th frame in the v-th view is denoted by $(x_j(i,t,v), y_j(i,t,v))$. Thanks to the ball-centric window, we only have limited number of detected players for pose estimation. The steps are shown in Fig. 3. In order to improve efficiency further, we crop the bounding boxes of persons and combine them. In order to improve accuracy, we double the width and height of the bounding boxes of persons. Finally, the joint positions will be mapped back to the ball-centric window.

Fact check: effect on person area. In our experiments, we found that enlarging the person area is an effective way to improve pose detection. As shown in Fig. 4, by doubling the width and height of the person areas, we can estimate more accurate poses that were not detected or were detected inaccurately in the original areas. If we compare the mean human size in training data as shown in Table 1 and the typical human sizes in our 4K soccer videos as shown in on top row of Fig. 4, the ratio is around twice. This is the reason why the width and height are enlarged with doubling not tripling or others. Note that although there are advanced super resolution techniques such as IDN[27], they are computationally heavy. In terms of cost performance balance, we prefer to use the simple resizing method.

3. 4 Object tracking

We use the MILTrack algorithm[28] to track the soccer ball and all the players in the ball-centric window. Because we only focus on the ball-centric window, there are some players who appear and disappear as shown in Fig. 5. By detecting the players in two frames, we can ascertain who appears and who disappears and assign new IDs to newcomers.

** The pre-trained model is available at `http://posefs1.perception.cs.cmu.edu/OpenPose/models/pose/coco/pose_iter_440000.caffemodel` on Aug. 26, 2019.

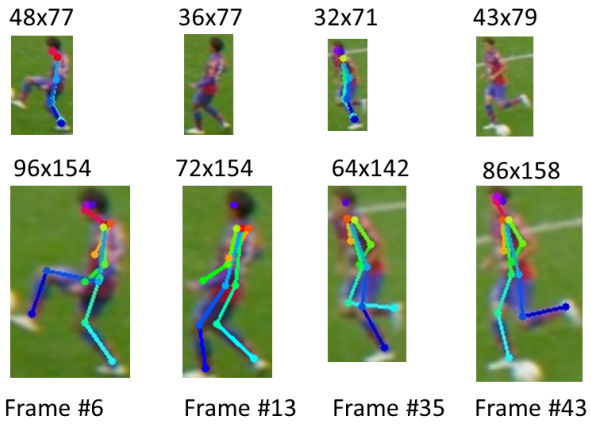

Fig. 4 Pose estimation by original size (top row) and enlarged size (bottom row). When the width and height of the person area are doubled, it becomes easier to estimate poses. The digits denote the width (pixel) x height (pixel). Note that the human sizes on top row here are very typical in our 4K soccer videos.

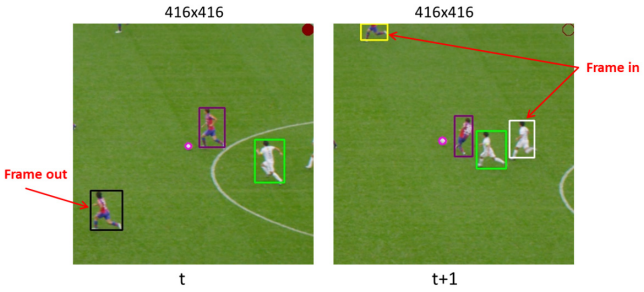

Fig. 5 There are some players who appear and disappear in the ball-centric window when tracking. The color of bounding box denotes person ID. The digits denote the width (pixel) x height (pixel).

3. 5 View mapping

We calculate the parameters of planar homography transformation between the base camera and any other camera in advance using the RANSAC algorithm[29]. Because the camera is static without pan, tilt, nor zoom movements, the parameters do not change during the soccer game and thus this operation is conducted only once. We select ten corresponding points in the soccer field to calculate the transformation parameters between two cameras as shown by the red circles in Fig. 1, which are manually marked in the first frame of each camera. The average reconstruction error of ten corresponding points from four cameras is 4.2 pixels.

With the parameters $\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ of homography transformation from base view vb to target view vt, we can map the ball center in each frame of

the base camera to the same frame of any other camera by

$$\hat{\boldsymbol{x}}_b^c(t, vt) = \boldsymbol{H} \cdot \boldsymbol{x}_b^c(t, vb) \quad (1)$$

where $\boldsymbol{x}_b^c(t, vb) = [x_b^c(t, vb), y_b^c(t, vb), 1]^T$ denotes the center of the detected ball in base view vb, $\hat{\boldsymbol{x}}_b^c(t, vt) = [\hat{x}_b^c(t, vt), \hat{y}_b^c(t, vt), 1]^T$ denotes the center of the mapped ball in target view vt. Note that $x_b^c(t, vb) = (x_b^l(t, vb) + x_b^r(t, vb))/2$ and $y_b^c(t, vb) = (y_b^l(t, vb) + y_b^r(t, vb))/2$.

When the ball center $\hat{\boldsymbol{x}}_b^c(t, vt)$ is mapped from the base view, we regard the 416x416 region with the center of mapped ball center as the ball-centric window in the target view, where the ball and players will be detected by YOLOv3[7] as described in Sect. 3. 2. Although the parameters $\boldsymbol{H}(vb, vt)$ are reasonably accurate, the ball may not be located on the field (i.e. in the air), which will cause large mapping errors. In some cases, the real position of ball $\boldsymbol{x}_b^c(t, vt)$ may be located outside the ball-centric window. Therefore, if no ball is detected in the ball-centric window, we will detect the ball again in the entire frame using a sliding window. If no ball is detected throughout the frame, this frame of this view will not be used in detection of kicking motions.

3. 6 Kicking detection

In each view, we calculate the Euclidean distances between the ball and the joints of players by

$$D_j(i,t,v) = \sqrt{(x_j(i,t,v) - x_b^c(t,v))^2 + (y_j(i,t,v) - y_b^c(t,v))^2}$$
$$D(i,t,v) = \min_j \{D_j(i,t,v)\} \quad (2)$$
$$D(t,v) = \min_i \{D(i,t,v)\} \quad (3)$$

where $(x_b^c(t,v), y_b^c(t,v))$ denotes the center of the detected ball in the v-th view, $(x_j(i,t,v), y_j(i,t,v))$ denotes the j-th joint position of the i-th detected player in the v-th view, $D_j(i,t,v)$ denotes the distance between the center of the detected ball and the j-th joint from the i-th detected player, $D(i,t,v)$ denotes the distance between the center of detected ball and the nearest joint in the i-th detected player, and $D(t,v)$ denotes the closest distance from all the detected players in the v-th view.

If the closest distance $D(t,v)$ is smaller than a threshold (set as 30 pixels) in at least two views as shown in Fig. 6, we decide that a kicking motion is detected at the t-th frame. Because we totally have four views but only require two views to satisfy the condition, the kicking detection is rather robust. Even if there are some errors in one or two views in the accuracy of the nearest joint position, we are still able to successfully detect the kicking motion. Although it is out of scope of this paper, we would like to mention an additional merit that we can potentially use this distance to improve the accuracy of the nearest joint position if it has larger error.

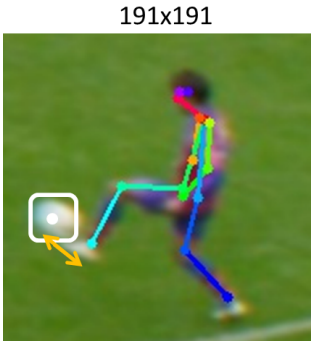

Fig. 6 The distance between the ball center and the nearest joint is used to detect kicking motions. The digits denote the width (pixel) x height (pixel).

4. Experiments

In terms of both speed and accuracy, we evaluate the proposed approach on real data from part of a J1 League soccer match. The data were captured by four synchronized 4K cameras with a frame rate of 30 fps as shown in Fig. 1. Note that we use a GeForce GTX 1080 Ti Graphics Card and the default setting of YOLOv3[7] and OpenPose[9].

4. 1 Accuracy evaluation

We compare the accuracy of kicking motion detection between a single base view and our approach. In total, there are 11 kicking motions in the test data. We define the detection as being correct if the detected kicking motion is located within the ground truth ± 2 frames. Otherwise, it is regarded as incorrect. With a single base view, we detect 14 kicking motions, where 8 motions are correct. With all four views, we detect 15 kicking motions, where 11 motions are correct. Table 2 shows the accuracy of kicking motion detection, which demonstrates that the performance of the proposed approach represents a significant improvement. Please watch our demo video called “Video 1”. All four failure cases come from over detection (i.e. false positive). The reason for over detection is that the closest distance $D(t,v)$ is very small in more than one view as shown in Fig 7. One possible solution is to use physical distance in 3D space, which is independent on any view.

4. 2 Speed comparison

We compare the computational time taken for object

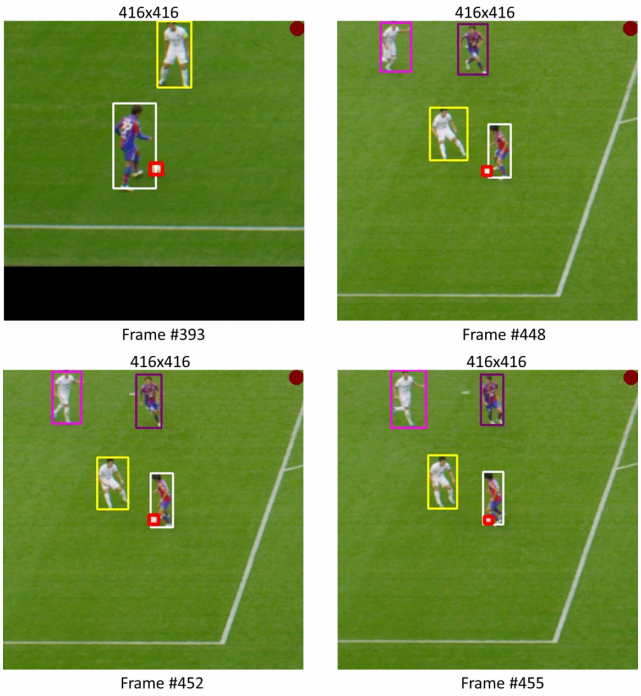

Fig. 7 All four failure cases in the test data. The distance between the ball and the nearest joint is very small. The digits denote the width (pixel) x height (pixel).

Table 2 Accuracy comparison of kicking motion detection between a single base view and all four views (proposed).

	precision	recall	F-measure
single base view	0.57	0.73	0.64
all four views	0.73	1.00	0.85

detection between a sliding window approach and our approach (ball-centric window). Here we exclude the loading and initialization time for the YOLOv3 model and OpenPose model. There are a total of 538 frames in the test data. The processing time of a 416x416 windows is 25 ms. Therefore, the total processing time of baseline for all frames from all four cameras is $25*538*4*(4096*2160)/(416*416) = 2,750,485$ ms. For the proposed approach, in the first frame, it takes 1075 ms from the base view and 30 ms for each other view. In other frames, it takes 90 ms for tracking from the base view and 30 ms for homography transformation for each of the other views. Therefore, the total computational time is $(1075 + 30 * 3) + 537 * (90 + 30 * 3) = 97,825$ ms. In other words, we achieved an approximately 28-fold increase in speed. If we separate the total computational time into base view and other views, the time for base view is $1075 + 537 * 90 = 49,405$ ms and the time for other views is $30 * 3 + 537 * 30 * 3 = 48,420$ ms.

4.3 Discussion

Limitations: The proposed approach has a number of limitations. Currently, we use the planar homography transformation between cameras, which is only suitable for sports with a planar field such as baseball and basketball. Another limitation is that the current approach cannot solve the occlusion problem. If the ball cannot be captured by any of the cameras, ball detection will fail. If body parts are occluded, pose estimation will partially fail. The third limitation comes from the still camera assumption. If the camera moves, it is necessary to calculate the parameters of homography transformation dynamically. Obviously, we need to detect the corresponding points automatically.

Base view selection: Although we fix the base view to one specific 4K camera in our experiments, there is no certain reason to do so in a general case. Basically, two requirements of base view are that (1) the ball should locate in all frames of the base view and (2) ball detection and tracking in the base view should be easier than other views. In our experiments, because requirement (1) is satisfied in all four views, we select our base view according to requirement (2), resulting in using Camera 1 as the base view. However, in a general case, the base view may change among four views according to requirement (1) and (2). For the bad aspect of fixing the base view, while the ball is missing at the base view, the system has to run the sliding-window process (even when it can be trackable at any of three other cameras), that results in high computation cost.

5. Conclusions and Future Work

This paper presents a novel approach to detecting kicking motions in multi-view 4K soccer videos. The approach was shown to be both fast and accurate. By using an object tracking technique and homography transformation, we were able to focus on a ball-centric window to significantly improve efficiency. By enlarging the person areas and fusing multi-view results, we were able to obtain much higher accuracy in the detection of kicking motion.

In the future, we will design an algorithm that will allow the corresponding points in calculating the parameters of homography transformation to be detected automatically. Also, we will propose a solution to solve the occlusion problem. Furthermore, we will apply our framework to other sports such as baseball and basketball. In addition, we would like to design an algorithm to choose a base view dynamically based on the visibility and/or ball-tracking performance without much additional computing cost.

References

1) "Wikipedia: 4k resolution," 2016.
2) G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.
3) Cem Direkoğlu and Noel E O'Connor, "Team activity recognition in sports," in *European Conference on Computer Vision*. Springer, 2012, pp. 69–83.
4) Masaki Hayashi, Kyoko Oshima, Masamoto Tanabiki, and Yoshimitsu Aoki, "Upper body pose estimation for team sports videos using a poselet-regressor of spine pose and body orientation classifiers conditioned by the spine angle prior," *Information and Media Technologies*, vol. 10, no. 4, pp. 531–547, 2015.
5) H. C. Shih, "A survey of content-aware video analysis for sports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1212–1231, May 2018.
6) Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
7) Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.
8) Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun, "Cascaded Pyramid Network for Multi-Person Pose Estimation," 2018.
9) Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh, "Real-time multi-person 2d pose estimation using part affinity fields," in *CVPR 2017*, 2017.
10) N. Tijtgat, W. V. Ranst, B. Volckaert, T. Goedemé, and F. D. Turck, "Embedded real-time object detection for a uav warning system," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 2110–2118.
11) J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, Jan 2018.
12) Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
13) Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
14) J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.
15) Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE CVPR*, 2017.
16) Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.
17) Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele, "Learning to refine human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
18) Matija Burić, Miran Pobar, and Marina Ivašić-Kos, "Object detection in sports videos," in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018.
19) Xinyu Wei, Long Sha, Patrick Lucey, Peter Carr, Sridha Sridharan, and Iain Matthews, "Predicting ball ownership in basketball from a monocular view using only player trajectories," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 63–70.
20) Rajkumar Theagarajan, Federico Pala, Xiu Zhang, and Bir Bhanu, "Soccer: Who has the ball? generating visual analytics and player statistics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1749–1757.
21) Shogo Miyata Hideo Saito and KosukeTakahashi DanMikami MarikoIsogawa HideakiKimata, "Ball 3d trajectory reconstruction without preliminary temporal and geometrical camera calibration," 2017.
22) Arda Senocak, Tae-Hyun Oh2 Junsik Kim, and In So Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1732–1739.
23) Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz, "Soccer on your tabletop," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4738–4747.
24) Vito Reno, Nicola Mosca, Roberto Marani, Massimiliano Nitti, Tiziana D'Orazio, and Ettore Stella, "Convolutional neural networks based ball detection in tennis games," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1758–1764.
25) Junjue Wang, Brandon Amos, Anupam Das, Padmanabhan Pillai, Norman Sadeh, and Mahadev Satyanarayanan, "Enabling live video analytics with a scalable and privacy-aware framework," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 3s, pp. 64:1–64:24, June 2018.
26) Wei-Tse Lee, Hsin-I Chen, Ming-Shiuan Chen, I-Chao Shen, and Bing-Yu Chen, "High-resolution 360 video foveated stitching for real-time vr," in *Computer Graphics Forum*. Wiley Online Library, 2017, vol. 36, pp. 115–123.
27) Zheng Hui, Xiumei Wang, and Xinbo Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.
28) B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 983–990.
29) Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

Jianfeng XU received the B.S. (with honor) and the M.S. degrees from Tsinghua University, China, in 2001 and 2004 respectively and the Ph.D. degree from the University of Tokyo, Japan, in 2007. He has been working at KDDI Research, Inc. since 2007 and now is a research engineer in Media Recognition Laboratory. His research interests include human motion analysis, sports analysis, and deep learning techniques.

Kazuyuki TASAKA received the M.S. and the Ph.D. degrees from Nara Institute of Science and Technology (NAIST), Japan, in 2004 and 2010 respectively. He has been working at KDDI Research, Inc. since 2004 and now is a group leader in Media Recognition Laboratory. His research interests include human action recognition and dynamic map platform.