2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Robust Volleyball Tracking System using Multi-View Cameras

Masaki Takahashi, Kensuke Ikeya, Masanori Kano, Hidehiko Ookubo, and Tomoyuki Mishina
Science and Technology Research Laboratories
Japan Broadcasting Corporation (NHK)
Tokyo, Japan
takahashi.m-iu@nhk.or.jp

***Abstract*— We have developed a real-time ball tracking system that can be used for volleyball games. Although a number of methods for visual object tracking have been proposed, tracking a fast-moving ball is still a challenging task because of the motion blur and the occlusion. We thus use a complementary tracking scheme in which tracking processes for multiple cameras help each other sharing the 3D position of the ball. The ball on each camera is accurately tracked by predicting its position at the next frame. The 3D ball positions measured by the system can be used for drawing the trajectory CG of a ball and for calculating statistical data related to ball movement. Evaluation results obtained using actual volleyball video sequences showed that the system would be effective for visualizing ball trajectories in live volleyball broadcasts.**

Keywords*— *sports video analysis; visual object tracking; multi-view cameras;

I. Introduction

Visual object tracking is an important task in computer vision because the technology for it can be used in various scenarios, such as surveillance, robotics, and human-computer interaction [1, 2]. Sports video analysis is one scenario for which visual object tracking is strongly demanded [3–5], and it has been widely studied thanks to the improvement of computer vision technologies and the development of camera devices. Most ball game audiences focus on the ball position and its movement. If the 3D position of a ball could be measured at every frame, the trajectory of the ball could be drawn by computer graphics (CG) and a variety of statistical data could be calculated. Robust tracking of a ball from a video sequence is thus a very important task, but the illumination change, occlusion, background clutter, etc. in actual video sequences make it very difficult.

Although many object tracking method have been proposed [6–8], for the following reasons it is still hard to track a ball automatically in volleyball games. First, the ball moves so fast that there is a huge motion blur in image obtained with the common 30 fps camera. The motion blur causes changes in the shape of a ball object and decreases its brightness, so it makes stable tracking difficult. Second, the ball moves freely in a 3D coordinate space, so it is hard to predict its position. Third, a ball tends to be hidden by players, so multiple viewpoints are needed for robust tracking. Moreover, the ball is much smaller than the players, so sufficient amounts of image features cannot be acquired from the small region.

Fig. 1. Example of a ball trajectory CG.

Considering those problems, we developed a ball tracking system that can be used for volleyball games. The system uses four HD cameras as sensors of ball position, and all camera images are processed in parallel to track the ball. Tracking results from those cameras are integrated and 3D ball position is calculated. The ball position is shared with all procedures, and is used to predict the ball position at the next frame. Thus even if ball detection failed in a certain camera, the tracking process can be continued by referring to the shared 3D ball position. This complementary tracking scheme is the main characteristic of our system.

We evaluated our system with video sequences of actual volleyball games and obtained results proving that the system can accurately measure positions of a ball and robustly track a ball for a long period. The ball tracking was performed in real-time, so the ball trajectory CG can be drawn for live volleyball broadcasts. Figure 1 shows a sample image of ball trajectory CG drawn by using our system's tracking results.

The remainder of this paper is organized as follows. Related work is described in Section II, the methods of ball tracking are explained in Section III, the results of the experiments are presented in Section IV, and the paper is concluded in Section V.

II. Related Work

Visual object tracking is an important theme, so many algorithms for it have been proposed [1, 2]. It is also important for sports video analysis because the positions of players and a ball attract much interest from audiences.

Visual object tracking is basically performed by matching the representation of a target model, and tracking methods based on discriminative classification have been proposed recently [9–13]. They build a model on the distinction of the target foreground against the background on the basis of their appearance features. Kalal et al. proposed a discriminative classifier learning method called Tracking Learning Detection (TLD) [10]. It simultaneously detects a target object in the video, learns its appearance, and tracks it. Multiple Instance Learning (MIL) tracker [11] uses a tracking-by-detection approach. It shows improved robustness to inaccuracies of the tracker and to incorrectly labelled training samples. Moreover, correlation filter based trackers proposed recently are regarded as state-of-the-art trackers [12, 13]. Bolme et al. [12] proposed a correlation filter based tracker called Minimum Output Sum of Squared Error (MOSSE), that uses classical signal processing techniques. Although these methods track targets accurately, it is hard for them to track volleyballs because they don't assume rapid change of the targets' positions and shapes.

Estimation of a ball position in 3D coordinates is a cue to robust tracking. However, the volleyball moves freely in real 3D coordinates, so it is difficult to measure the precise 3D position of a ball. Although many methods for measuring the 3D position of an object have been studied [6, 7], their results are unreliable because of the lack of multiple view-points.

When a ball is hidden by players, ball tracking tends to be interrupted by the detection of other noise objects. Thus, estimation of a hidden true value from system observations, which contain missing data and measurement errors, is another problem. Although some technologies have been proposed [14, 15] to solve this occlusion problem, it is still hard for them to estimate the ball's true position from single-view information. The Kalman Filter [16] is an estimation algorithm that assumes a linear and Gaussian model. Particle filter [17, 18], which is another estimation algorithm, assumes a non-linear and non-Gaussian model, so it is suitable for tracking objects whose movements are unpredictable. Although some particle filter based trackers are proposed [19, 20], tracking a ball from a complex background is still difficult, and the target cannot be reacquired after it is lost.

Small size leads to lack of feature points, so it is difficult for tracking methods which use gradient based features, such as SIFT and SURF [21], to detect small balls. In addition, those feature representations have high dimensions and their models are complicated, so their computational cost is high. Real-time computation as well as high accuracy is demanded for video analysis in live sports.

We propose a robust volleyball tacking method with multiple cameras. Ball positions from every pair of cameras are used to calculate a 3D position of a ball, and the position is shared with all cameras. Each process predicts the ball position at the next frame on the basis of the 3D ball position. This means that tracking processes can continue even when a tracking failed for a certain camera, and stable tracking can be achieved in total. This complementary scheme is the main characteristic of this system.

III. Proposed Method

A. Overview

The proposed system tracks a ball object from multi-view cameras. As seen in the system's signal-flow diagram (Fig. 2), it comprises three modules: one for ball detection, one for 3D position measurement, and ball position prediction.

The inputs of the system are video sequences shot by four fixed cameras. Those video sequences are processed in parallel, and balls in those images are detected every frame. A ball position in real 3D coordinates is calculated on the basis of those tracking results. The 3D position of the ball is used to draw a ball trajectory and to calculate variety of statistical data related to the ball movement.

The 3D position of a ball is also used for ball position prediction for the purpose of stable tracking. Even when ball tracking failed in a certain camera, ball position at the next frame can be predicted by sharing the 3D position of the ball among all cameras. This complementary architecture of the system contributes to robust ball tracking. We explain the details of each of the three modules in the following subsections.

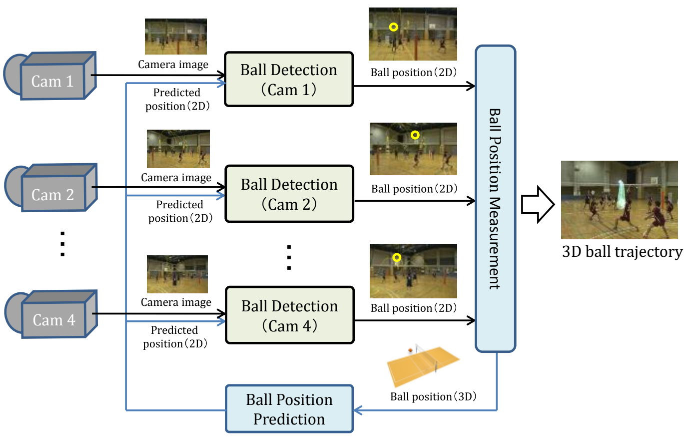

Fig. 2. Signal-flow diagram of the proposed system.

B. Ball Detection Module

The ball detection module detects a ball object in a 2D camera image every frame. A video sequence from a camera is input to this module, and a ball position in its image coordinates is measured. This module is processed in parallel among all four cameras.

The module detects a ball by creating an accumulated frame difference image defined as in equation (1), where $\boldsymbol{I}_t^{xy}$ denotes a brightness at position (x, y) on the input image at frame t, and S_t^{xy} denotes a given pixel value from the system at position (x, y). First, frame difference at every pixel is calculated, and the specific value (S_{max}) is given if the difference exceeds a threshold M. Only moving objects can be extracted by this operation. Next, morphological operations (close and open) are applied to eliminate small noises. Then, ball candidate objects at the frame image are extracted by a filtering process related to their area sizes and shapes. Big and small objects and objects with complicated shapes are excluded from the candidates by this filtering process.

Ball candidate object might not be a single object because other moving objects, such as players' bodies, could also be extracted. Thus, all candidate objects are accumulated over certain frames to prevent from miss-detection of those noise objects. Brightness of ball candidates of past frame is gradually decreased every frame as shown in equation (1).

$$S_t^{xy} = \begin{cases} S_{max} & if\ |I_t^{xy} - I_{t-1}^{xy}| > M \\ S_{t-1}^{xy} - 1 & otherwise \end{cases} \tag{1}$$

Figure 3 shows an example of an accumulated frame difference image. Brighter and darker areas in the image denote newer and older detected ball candidates, respectively. Old candidates are gradually expanded by morphological operations, which are applied every frame, to connect same candidate objects over consecutive frames.

A ball object is searched for only in a search region, which is drawn as green rectangle in Fig. 3. The brightest objects in the region are regarded as final candidates. The candidate object nearest to the predicted position is selected as a ball object at last. We explain the predicted position in detail in Section III. *D*.

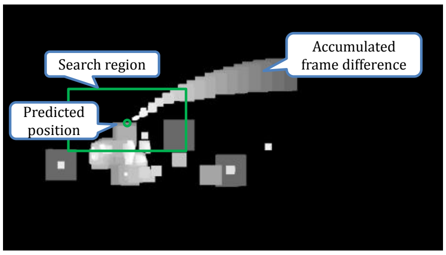

Fig. 3. Example of an accumulated frame difference image.

C. 3D Position Measurement Module

Ball positions in the two-dimensional image coordinates are measured for all cameras as described in the previous subsection. In this module, ball position in the real three-dimensional coordinates is calculated on the basis of those 2D positions of ball objects in the manner of triangulation.

A 3D position of the ball is measured as a midpoint between two lines-of-sight from a camera to the ball as shown in Fig. 4. The two red lines denote lines-of-sight from two cameras to the ball. A line-of-sight can be decided from two specific positions in 3D coordinates. Camera position, which can be measured in advance, is used as the one of the positions, and a ball position on a virtual plane is used as the other one. As shown in Fig. 4, we assumed the virtual plane to be at the same place as the volleyball net in the court.

Ball positions on the virtual plane can be calculated by projective transformation [22]. Figure 5 shows a concept of projective transformation from a camera image to the projected image on the virtual plane. Projective transformation could be performed as described in equation (2). Positions in the virtual plane $(X_{b1}, Y_{b1}, 0)$ were calculated from positions on the camera image (x_{b1}, y_{b1}) and homography matrix $\boldsymbol{H}$. Elements h_1 to h_8 in the $\boldsymbol{H}$ are calculated by manual pointing of more than four corresponding points between camera image and virtual plane beforehand. Every pixel in the camera image can be transformed to the virtual plane.

The 3D ball positions can be calculated with every two different camera image, so that six (= ${}_4C_2$) 3D ball positions can be acquired in total. An average position of those six positions is calculated at once, some outliers are then eliminated by calculating their distance to the average position. A final ball position is calculated by averaging remained positions.

A temporal history of ball positions in real 3D coordinates, which were output from our system, can be used to draw a ball trajectory CG on a video sequence. The data can be also used for calculating some properties related to the ball, such as ball speed, height, and touched positions by players. Variety of statistical data of the game can be calculated on the basis of the properties, so they are regarded as very important. Furthermore, the 3D position of a ball can be used for stable ball tracing in this system by predicting the ball position at the next frame as described in the next subsection.

$$X_{b1} = \frac{h_1 x_{b1} + h_2 y_{b1} + h_3}{h_7 x_{b1} + h_8 y_{b1} + 1} \qquad Y_{b1} = \frac{h_4 x_{b1} + h_5 y_{b1} + h_6}{h_7 x_{b1} + h_8 y_{b1} + 1} \qquad \boldsymbol{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \tag{2}$$

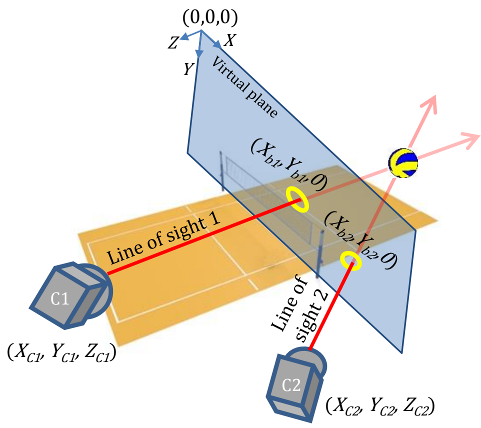

Fig. 4. Calculation of ball position in 3D coordinates.

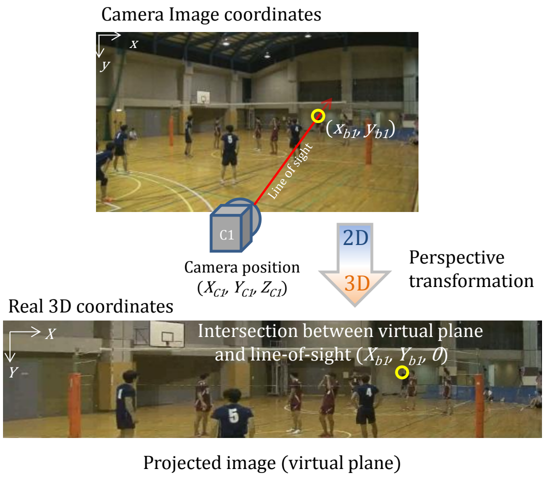

Fig. 5. Projective transformation from camera image to virtual plane.

D. Ball Position Prediction

The 3D position of a ball can be acquired by the two modules described in the preceding subsections. In this module, the system predicts ball position at the next frame on each camera image from the 3D position. The predicted position is used as a reference in the ball detection module, so this module contributes to accurate ball detection.

We used the particle filter to predict ball positions in camera images. The particle filter estimates a hidden true value by integrating weak estimations calculated with many particles [17, 18]. It can estimate the nonlinear and non-Gaussian motion of a target and thus it is suitable for tracking the unpredictable movement of a volleyball. We used straight-forward motion dynamics for the particle.

The measured 3D ball position is projected on each camera image as shown in Fig. 6. The likelihood of each particle is calculated on the basis of the distance between the particle and the projected position of the ball. Thus, the weight of particles near the projected position becomes high. Predicted ball position is calculated as a weighted expectation value among all particles. The predicted position is used as a reference for selecting a ball object. In addition, a ball search region on each camera image is set around the predicted position to prevent the system from catching noise objects that are far from the predicted position. In Fig. 6 the small green circles denote predicted positions and green rectangles denote search regions. Small yellow circles show ball positions detected in the past.

Moreover, the predicted ball position helps to reacquire a ball object after the system missed the ball in a certain camera. A ball is sometimes hidden by players' bodies. Even if a ball was missed in a camera image, a position of the search region is automatically updated referring to the predicted ball position. This means that all tracking process helps each other to achieve robust tracking by sharing the ball's 3D positions that were calculated with their tracking results. This complementary approach for robust tracking is the greatest characteristic of the proposed system.

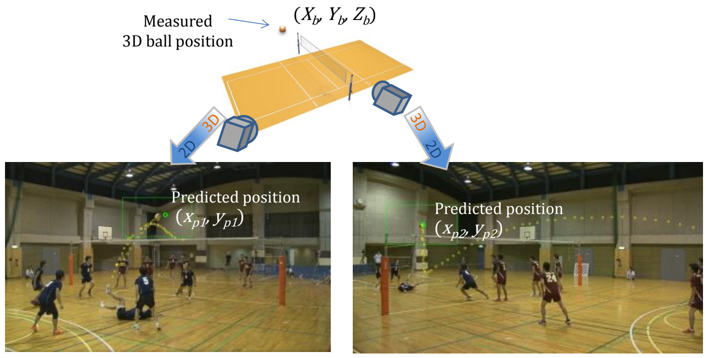

Fig. 6. Concept of ball position prediction.

IV. Experiment

A. Experimental Conditions

We evaluated the proposed system with actual videos of volleyball played by semi-professional players. We recorded a serve sequence (108 frames), and two game sequences (462 and 685 frames) for this experiment. Four fixed HD cameras (1920*1080 pixels, 30 fps) were positioned around a volleyball court as shown in Fig. 7.

The 3D positions of those cameras were measured in advance, and their homography matrices were calculated by pointing out more than four landmarks, such as edge points of the volleyball net, in all camera images. Owing to those calibrations, any points in 2D camera image coordinates can be transformed into 3D coordinates.

For evaluation, true ball positions on all camera images were annotated manually at each frame, and those 2D and calculated 3D positions were used as ground truth ball positions for this experiment. Four common PCs (3.5GHz processor, 16GByte memory) were used in this experiment. They processed video sequences of each camera in parallel. We evaluated our system in the following three kinds of evaluations.

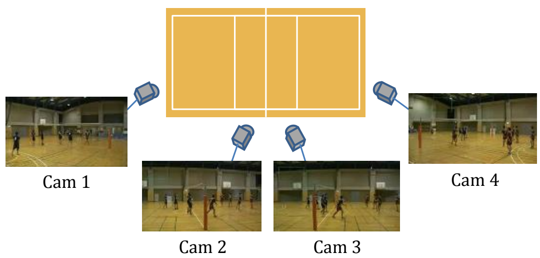

Fig. 7. Camera layout and their fields of view.

B. Tracking Accuracy

First, we evaluated the ball tracking accuracy of our system by comparing it with the accuracies of conventional visual object tracking methods. Three state-of-the-art tracking methods—TLD [10], MIL [11], KCF [12]—were used as comparison methods. Average error distances from ground truth in 2D image coordinates and their processing speed were measured for the serve sequence. The results are listed in Table I.

Our method achieved the best performance in terms of both the average error distance and processing speed. The three conventional methods could not track a ball until the last frame because they cannot reacquire the ball once they had lost it. Complicated background prevented those methods from robust tracking. Our proposed method has a complementary tracking scheme sharing the ball's 3D position; thus it could track a ball until the last frame and its error distance was small.

The processing time of our proposed method for a camera image was 31.2 msec. The speed was less than 33 msec/frame, which means that the system can track a ball in real-time. Real-time processing is necessary for a live sports broadcast system. Our fast computational ball detection function contributed to this fast processing speed. Other methods could not achieve real-time processing because their computational costs were high because of the online learning procedure.

TABLE I. AVERAGE ERROR DISTANCE AND PROCESSING SPEED.

	Average error distance [pixel]	Processing speed [msec]
TLD [9]	133.5	97.3
MIL [10]	102.8	81.7
KCF [11]	99.5	80.5
Proposed	59.3	31.2

C. Robustness of Tracking

Next, we evaluated robustness of tracking. We measured the success rate of ball tracking for the game 1 sequence. The rate of an un-complementary tracking method, which didn't predict ball position, was also evaluated for comparison. In that method, a ball was separately tracked on each camera without sharing 3D position of a ball. Table II lists the results.

The success rate of the un-complementary method was low for Cam 4 because the ball was frequently hidden by players in its field of view. The occlusions caused many tracking failures. The method had difficulty reacquiring the ball because it could not properly predict the ball positions without information from other cameras. Its average success rate was 79.8%, much lower than that of our method: 96.7%. Our complementary scheme of ball tracking contributed to robust tracking of a ball.

The proposed method was also compared to the Kalman filter. The Kalman filter predicts hidden states of a target with the linear and Gaussian model. Ball movement in volleyball is complicated because the ball moves to every direction in 3D space and the direction changes suddenly when the ball is touched by players. Our method predicts ball position by using a particle filter, which assumes a nonlinear and non-Gaussian model. Its result was better than the result of the Kalman filter.

Figure 8 shows the tracking results of the game 1 sequence for all four cameras. Small yellow points denote the detected ball positions. Even if the ball tracking for a camera failed, its ball search region was updated by referring to the ball's 3D position calculated from tracking results of other cameras. The system robustly tracked a ball through all frames of the playing scene, and it measured accurate 3D ball positions every frame.

TABLE II. TIME RATE OF BALL TRACKING [%].

	Cam 1	Cam 2	Cam 3	Cam 4	Average
Un-complementary method	76.9	76.7	97.5	68.1	79.8
Proposed (Kalman filter)	85.6	85.1	97.1	95.8	90.9
Proposed (Particle filter)	95.1	97.8	97.5	96.5	96.7

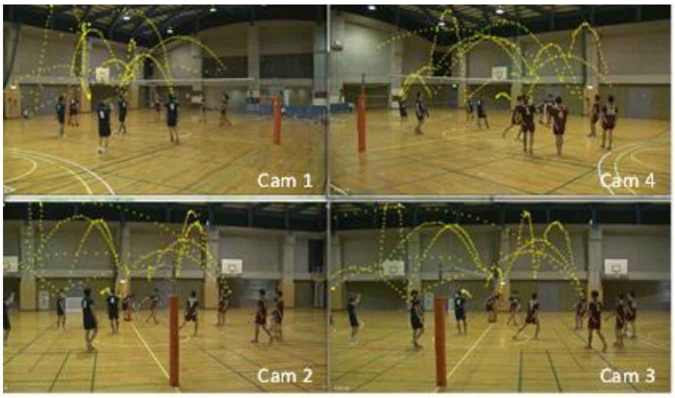

Fig. 8. Tracking results of game sequence.

D. Tracking Accuracy in 3D Space and CG Drawing

Finally, tracking accuracy in real 3D space was evaluated. Error distances from ground truth position in 3D coordinates were measured every frame, and their average was calculated. The results are listed in Table III.

Average error distance for the game 1 sequence was 22.2 cm and the distance for game 2 was 21.4 cm. There is little margin with two error distances and their average was 21.8 cm. This error distance in real-3D coordinates is about the size of an actual volleyball because the diameter of a volleyball is about 21 cm. This accuracy is reasonable for a system used to draw the trajectory CG of the ball. The ball position prediction module contributed to this high accuracy because it prevented the catching of noise objects far from the ball and reduced the error distance from the true ball position.

TABLE III. AVERAGE ERROR DISTANCE IN 3D COORDINATES.

Sequence	Average error distance [cm]
Game 1	22.2
Game 2	21.4
Average	21.8

We rendered a sample video sequence of ball trajectory CG assuming the use of a live volleyball broadcast. The CG can be drawn on any calibrated cameras because our system outputs ball positions in real 3D coordinates. Figure 1 and 9 show samples of output images of ball trajectory CGs.

The ball's speed and its bounding positions on the ground can be measured automatically because the real 3D position of the ball is calculated every frame. In addition, ball touching positions can be automatically detected by searching for a position where ball direction was immediately changed. A text display of ball speed for the spike scene and a CG effect of water spray at the received position were drawn in the image shown in Fig. 9.

Although ball trajectories and those specific positions can be measured automatically, some sort of manual operations for CG drawing and a certain period of rendering process are needed. We plan to improve those problems and use the system for rendering visual effects in live volleyball broadcasts.

Fig. 9. Sample image of ball trajectory CG.

V. CONCLUSION

A real-time ball tracking system that can be used for volleyball games has been developed. It robustly tracks a ball from video sequences shot by multiple cameras. The measured ball positions can be used for calculating various statistical data and making visual effects for a broadcast program. The system robustly tracks a ball by calculating real 3D ball position and predicting ball position at the next frame. We evaluated the system by using actual volleyball sequences shot by four cameras. The average error distance in 3D coordinates was 21.8 cm, which was about the diameter of a volleyball. This robust tracking was due to the complementary scheme of the proposed system. A sample video of ball trajectory CG was also created on the basis of 3D positions measured by this system. We plan to use the system for drawing ball trajectory CGs in live volleyball broadcasts in the future.

REFERENCES

[1] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 99, pp. 1–56, 2013.

[2] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking VOT2015 challenge results," in proc. of ICCV2015 Workshop, pp. 1–23, 2015.

[3] G. A. Thomas, "Virtual graphics for broadcast production," IEEE Computer, vol. 42, no. 7, pp. 42–47, 2009.

[4] C. B. Santiago, A. Sousa, M. L. Estriga, L. P. Reis, and M. Lames, "Survey on team tracking techniques applied to sports," in proc. of International Conference on Autonomous and Intelligent Systems. pp. 1–6, 2010.

[5] P. Spagnolo, M. Leo, P. L. Mazzeo, M. Nitti, E. Stella, and A. Distante, "Non-invasive soccer goal line technology: A real case study," 1st IEEE International Workshop on Computer Vision in Sports, in conjunction with CVPR, 2013.

[6] H. Chen, W. Tsai, Lee. S, and Yu. J, "Ball tracking and 3D trajectory approximation with applications to tactics analysis from singlecamera volleyball sequences," Multimedia Tools and Applications, vol. 60, no. 3, pp. 641–667, 2012.

[7] E. Morais, S. Goldenstein, A. Ferreira, and A. Rocha, "Automatic tracking of indoor soccer players using videos from multiple cameras," in Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on (pp. 174–181). IEEE, Aug. 2012.

[8] F. Yan, A. Kostin, W. Christmas, J. Kittler, "A novel data association algorithm for object tracking in clutter with application to tennis video analysis," in proc. of CVPR, 1, pp. 634-641, 2006.

[9] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in proc. of International Conference on Computer Vision (ICCV), 2011.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 37, no. 7, pp. 1409–1422, 2012.

[11] B. Babenko, M. H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 8, pp. 1819–1632, 2011.

[12] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in proc. of CVPR, pp.1–10, 2010.

[13] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in proc. of ICCV 2015, pp. 3038–3046, 2015

[14] I, Iman and K. Faez, "Object tracking with occlusion handling using mean shift, Kalman filter and edge histogram," Pattern Recognition and Image Analysis (IPRIA), 2015 2nd International Conference on. IEEE, 2015.

[15] H. Rezaee, A. Aghagolzadeh, M.H. Seyedarabi, and S. A. Zu'bi, "Tracking and occlusion handling in multi-sensor networks by particle filter," In GCC Conference and Exhibition (GCC), 2011 IEEE (pp. 397–400), 2011.

[16] R.E. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME - Journal of Basic Engineering pp. 35–45, 1960.

[17] Doucet, A., Freitas, N. D., and Gordon, N. 2001. Sequential Monte Carlo Methods in Practice. Springer.

[18] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," Signal Processing, IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002.

[19] R. Hess, A. Fern, "Discriminatively trained particle filter for complex multi-object tracking," in proc. of CVPR, pp. 240–247, 2009.

[20] T.S. Huang, J. Llach, and C. Zhang, "A method of small object detection and tracking based on particle filters," In 19th IEEE International Conference on Pattern Recognition, ICPR, 2008.

[21] D. G. Lowe, "Distinctive image features from scale in-variant key points," Int. Journal of Computer Vision, vol. 60, pp. 91–110, 2004.

[22] R. Baer, "Linear Algebra and Projective Geometry," Dover, 2005.