Review

Pegah Rahimian* and Laszlo Toka

Optical tracking in team sports

A survey on player and ball tracking methods in soccer and other team sports

https://doi.org/10.1515/jqas-2020-0088
Received March 12, 2020; accepted January 7, 2022;
published online March 7, 2022

Abstract: Sports analysis has gained paramount importance for coaches, scouts, and fans. Recently, computer vision researchers have taken on the challenge of collecting the necessary data by proposing several methods of automatic player and ball tracking. Building on the gathered tracking data, data miners are able to perform quantitative analysis on the performance of players and teams. With this survey, our goal is to provide a basic understanding for quantitative data analysts about the process of creating the input data and the characteristics thereof. Thus, we summarize the recent methods of optical tracking by providing a comprehensive taxonomy of conventional and deep learning methods, separately. Moreover, we discuss the preprocessing steps of tracking, the most common challenges in this domain, and the application of tracking data to sports teams. Finally, we compare the methods by their cost and limitations, and conclude the work by highlighting potential future research directions.

Keywords: deep learning; image processing; optical tracking; player tracking; soccer; sports analytics.

1 Introduction

The success in every team sport significantly depends on the analysis of the semantics. Most team sports, such as football, basketball, and ice hockey, involve very complex interactions between players. Researchers and data analysts propose various methods for modeling these interactions. For this aim, they need to follow the movements of players and the ball from the video. However, this task is strenuous due to the large speed of players and of the ball in the playfield, and tracking usually fails in the cases of overlaps, poor light conditions, and low quality of the videos. During the past decades, computer vision researchers developed several optical tracking algorithms by analyzing video image pixels and by extracting the features of the objects of interest, such as players and the ball. On this data, the movement, action, intention, and gesture of the players can be analyzed.

The most common analysis is performed over player and ball tracking data, also known as trajectory data. The distilled knowledge can help coaches and scouts in several aspects, such as game strategy and tactics, goal analysis, pass and shot prediction, referee decisions, player evaluation, and talent identification. In order to automatize the end-to-end analytics procedure, the tracking methods require visual data (video frames) as the data source and produce tracking data (player and ball trajectories) for further data mining. The proposed methods majorly contribute to effectively evaluate the performance at individual and team levels in team sports. E.g., at the individual level, the characteristic style of a player, while at the team level, the combination of all players' trajectories can be evaluated.

The work in this paper is motivated by the following observations. First, researchers in sports analytics are continuously searching for the most accurate, but a cost-effective method for the player and ball tracking. The above-mentioned goals of tracking prove the importance of opting for an accurate method for extracting player and ball trajectories in sports analytics. Second, player and ball tracking are one of the broadest areas for research in sports analytics. In the literature, there are many published works without proper classification. Recently, the automatic feature extraction capability of deep learning in computer vision encourages sports analysts to experiment with neural networks for player and ball tracking tasks. Thus, a wider range of tracking options are available to the researchers and this survey helps them to choose

***Corresponding author**: **Pegah Rahimian**, Budapest University of Technology and Economics, Budapest, Hungary,
E-mail: pegah.rahimian@edu.bme.hu
Laszlo Toka, MTA-BME Information Systems Research Group, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary,
E-mail: toka.laszlo@vik.bme.hu

their suitable method depending on the task at hand. Furthermore, understanding all these methods requires deep knowledge of computer vision for quantitative analysts in sports, which is not realistic. Therefore, in this paper, we have the following goals: to provide a robust classification of methods for the two tasks of detection and tracking and to give insights about the applied computer vision techniques of extracting trajectories to the quantitative analysts in sports.

Several papers made attempts to present the myriad of state-of-the-art object tracking algorithms. A broad description of object tracking methods was given in Yilmaz, Javed, and Shah (2006), and a more recent one in Reddy, Priya, and Neelima (2015). Moreover, Dhenuka, Udesang, and Hemant (2018) presented a survey on multiple object tracking (MOT) methods, while a survey for solving occlusion problems was published in Lee et al. (2014). The first survey on the application of deep learning models in MOT is presented in Ciaparrone et al. (2019). All these surveys cover the description of tracking methods of generic objects, such as humans or vehicles. It was in Manafifard, Ebadi, and Abrishami (2017b) where the authors summarized the state-of-the-art player tracking methods focusing on soccer videos. Although, these surveys show the following shortcomings. Most of these papers are not dedicated to team sports and survey all kinds of object tracking algorithms. On the other side, the sport dedicated survey like Manafifard, Ebadi, and Abrishami (2017b), is too technical, suitable only for computer vision analysts, and dedicated to tracking.

This survey contributes to the state-of-the-art player and ball tracking methods as follows. First, the methods in detection and tracking tasks are classified separately. Second, this paper is not only listing the methods but also gives an insight about the computer vision techniques to the quantitative analysts in sports, who need the extracted trajectories for their quantitative models. Third, the application of deep learning in team sports is surveyed for the first time in the literature. Fourth, we provide a cost analysis of the methods according to their computational and infrastructure requirements.

This paper is organized as follows. In Section 2 we explain our paper collection process and the camera setup requirements of the published works. We list the methods for the player and ball detection in Section 3, and the player and ball tracking in Section 4. We evaluate the categorized techniques in terms of their applied theoretical methods and analyze their cost in Section 5, and finally, we conclude the work in Section 6.

2 Eligibility and data collection

This survey is conducted to help quantitative sports analysts choose the best method to create their own tracking data from sports videos. For this task, the eligible papers are collected from Science Direct, Google Scholar, Scopus databases, and ACM, IEEE, Springer digital libraries using the following keywords for filtering papers and minimizing bias: “sports analytics”, “soccer”, “player tracking”, “ball tracking”, “player detection”, “ball detection”, “deep learning for tracking”, “fixed camera”, “moving camera”, “broadcast sports video”. In the first round of collection, 125 papers have been identified and we carefully inspected their contributions in terms of (1) detection or tracking, (2) camera setup, and (3) deep learning-based or traditional methodologies. In order to make the best structure of this survey, we excluded the papers in which tracking was not the main focus. An example is a method called DeepQB in American football proposed by Burke (2019). This paper proposes a deep learning approach applied to player tracking data to evaluate quarterback decisions, which is clearly not a direct contribution in player tracking methods. As a result of filtering those papers and focusing on player or ball detection and tracking, 50 papers were eligible for this survey. Furthermore, we also classified eligible papers according to their camera setup as follows.

One of the most important criteria for the evaluation of the methods in this work is the required camera setup. Depending on the camera setup, the frame extraction methods are different. Several studies in sports video analytics are limited to a single fixed camera. In these methods preprocessing steps are simpler and faster, as they do not require time and location synchronization. However, as they need to cover the whole playfield, the frames are mostly blurry and difficult to use for detection (Arbues, Ballester, and Haro 2019; Needham and Boyle 2001; Rodriguez-Canosa et al. 2012; Sabirin, Sankoh, and Naito 2015). An alternative setting to improve resolution and accuracy is to use multiple fixed cameras. In these videos occlusion problems can be handled easily, as the occluded player or ball in one frame can be recognized with the frame captured by another camera from other angles (Ren et al. 2008, 2009; Wu 2008; Yazdi and Bouwmans 2018). Another option is to use multiple moving cameras, which makes the video processing more complex, but it provides more flexibility in the analysis. These types of video require significant synchronization effort, but finally, they produce longer trajectories, as the cameras try to follow ball controllers (Agelet Ruiz 2010; Alavi 2017; Xu, Orwell, and Jones 2004; Mondal 2014). In this

paper, we classify each of the cited papers according to their required video inputs in terms of the cameras being fixed or moving, and of their cardinality in the arena.

3 Player and ball detection

Tracking data, i.e., the exact location of the players and the ball on the field at each moment of the match, is the most important data for a quantitative model developer. Player and ball detection methods are computer vision techniques that allow the analyst to identify and locate players and the ball in a frame of a sports video. Detection methods provide the input to tracking, which would be a simple task if all players and the ball were totally visible in each frame and there were no occlusion. However, in real-world videos, most frames are blurry and continuous tracking fails due to e.g., occlusion, poor light, or posture changes. Therefore, the detection task should be combined with an appropriate tracking method to accurately track the players and the ball (see Figure 1).

In this section, we focus on detection methods that aim to find the bounding box of the players and the ball, and to localize the different detection features inside each bounding box. Bounding boxes are imaginary boxes around players and the ball (see Figure 1) that are used to separate each player and ball from other objects in a video frame. We classify detection methods into the categories of traditional and deep learning-based methods. As Figure 2 shows, while in the traditional methods the features of the input objects need to be described and extracted by the analyzer and depend on the detection algorithms, a deep learning method performs this process automatically through the layers of a neural network. Therefore, data quality, computational power, domain expertise, training time, and required accuracy specify the selection of the

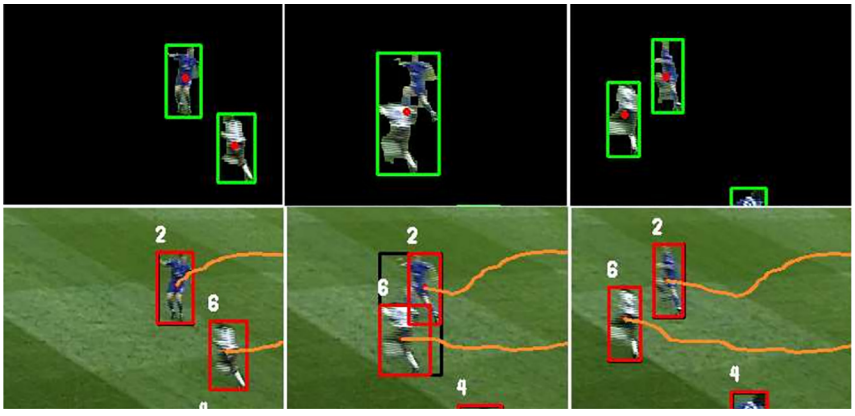


Figure 1: Player detection (top) and tracking (bottom) results from Xu, Orwell, and Jones (2004).

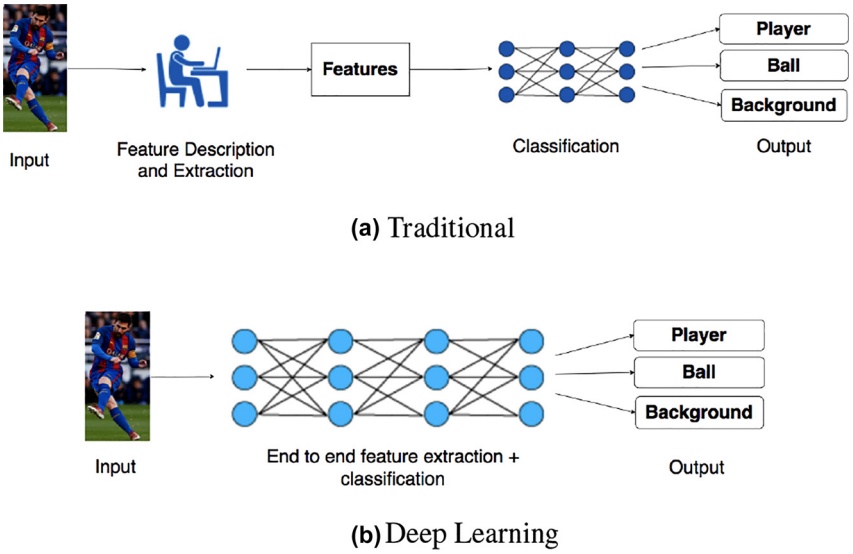


Figure 2: Player and ball detection workflow.

suitable choice of method to apply. We briefly describe each group of methods separately, and give a summary of published research papers, along with their important attributes, in Table 1.

3.1 Traditional methods for detection

In the traditional methods of detection, the features of players, ball, and playfield must be precisely described and extracted by the analyzer. In this section, we classify the methods according to their description of the features, and their extraction types.

3.1.1 Histogram of oriented gradients

Histogram of oriented gradients (HOG) is a feature descriptor and is essentially used to detect multiple objects in an image by building histograms of pixel gradients through different parts of the image. HOG considers these oriented gradients as features. An example of a calculating histogram of gradients is illustrated in Figure 3. As the first step, the frame is divided into 8×8 cells. For each cell, the gradient magnitude (arrows' length) and gradient direction (arrows' direction) will be identified. Consequently, the histogram containing nine bins corresponding to angles 0, 20, 40, ..., 160 is calculated. This feature vector can be used to classify objects into different classes, e.g., player, background, and ball. This method is used by Mackowiak et al. (2010) and Cheshire, Hu, and Chang (2015).

In these methods, the court lines can be detected with Hough transform, another feature extraction technique that searches for the presence of straight lines in an image. This algorithm fits a set of line segments to a set of image pixels.

3.1.2 Background modelling

Background modeling is another method for detecting players and the ball, and is a complex task as the background in sports videos frequently changes due to camera movement, shadows of players, etc. Most of the methods in the background modeling domain consider image pixel values as the features of the input objects. In the domains of player and ball detection, the following two methods are proposed by researchers for background modeling: Gaussian mixture model (GMM) and Pixel energy evaluation.

Gaussian mixture model (GMM): GMM is proposed by Ming, Guodong, and Lichao (2009) where playfield detection is performed first by taking the peak values of RGB histograms through the frames. This is because they assume the playfield is the largest area in the frames. Then each of these extracted background pixels is modelled by k Gaussian distributions; different Gaussians are for different colors. Thus, the probability of a pixel having value X_t can be calculated as:

$$P(X_t) = \sum_{i=1}^{k} \omega_i \eta(X_t) \tag{1}$$

where ω_i is the weight for the ith component (all summing to 1), and $\eta(X_t)$ is a normal distribution density function. Based on these probabilities and by setting arbitrary thresholds on the value of the pixels, the background pixels can be subtracted and the players or the ball will be detected. This algorithm cannot recognize players in shadows.

Pixel energy evaluation: Another background model is proposed by Mazzeo et al. (2008). In this method, the energy information of each point is analyzed in a small window: first, the information, i.e., mean and standard deviation, of the pixels at each frame is calculated. Then, by subtracting the information of the first image of the window and each subsequent image, the energy information of each point can be identified. Consequently, the slower energy points (static ones) represent the background, and higher energy points (moving ones) represent the players or the ball.

3.1.3 Edge detection

Edge detection is a method for detecting the boundaries of objects within frames as the features. This method works by detecting discontinuities in brightness. The researchers who choose this method for players and ball detection, mostly utilize the following two operators: Canny edge detector, and Soble filtering. Figure 4 demonstrates the edge detection methods on a sample frame of a player.

Canny edge detection: Is a popular method in OpenCV for binary edge detection (Figure 4(b)). Direkoglu, Sah, and O'connor (2018) proposed using the Canny edge detection method for extracting image data and features. However, there might be missing or disconnected edges, and it does not provide shape information of the players and the ball. Thus, given a set of binary edges, they solve a particular heat equation to generate a shape information image (Figure 4(c) and (d)). In mathematics, the heat equation is a partial differential equation that demonstrates the

Table 1: Review of playfield and player detection methods.

Reference	Playfield detection	Player detection	Team sport & camera type	Evaluation
Mackowiak et al. (2010)	Hough transform for court line detection	HOG descriptor	Football video broadcast	Performs well in SD and HD test sequences, different light condition, and various positions of the cameras; 78% precision
Cheshire et al. (2015)	Hough transform	Pedestrian detection with HOG & color-based detection	Basketball video broadcast	Miss rate: 70% for pedestrian detection
Ming, Guodong, and Lichao (2009)	Peak value of RGB	Adaptive GMM	Football video with single moving camera	Powerful segmentation result, but only in the absence of shadows
Mazzeo et al. (2008)	Background subtraction	Moving object segmentation by calculating energy information for each point	Football video with single stationary camera	Copes with light changes by proposing pixel energy evaluation
Direkoglu, Sah, and O'connor (2018)	Binary edge detection of court line with Canny edge detector	Using shape information of an object by solving heat diffusion equation	Hockey video with single stationary camera	Highly accurate method between 75 and 98%, but computationally less efficient in time required for detection
Naushad Ali, Abdullah-Al-Wadud, and Lee (2012), Rao and Pati (2015)	RGB color extraction if $G > R > B$	Sobel gradient algorithm	Football video broadcast	Accurately detects the ball when it is attached to the lines; but in crowded places, it fails to detect the player
Markoski et al. (2015)	–	Face recognition with adaboost	Basketball video with single moving camera	Detection accuracy: 70%
Zhu et al. (2006)	GMM	SVM for player classification	Soccer, hockey, American football video broadcast	Detection accuracy: 91%
Chengjun (2018)	Background subtraction	One-class SVM	Football video broadcast	Proposes automatic labeling of training dataset that significantly reduces cost and training time
GerkeKarsten and Schäfer (2015)	–	CNN for number recognition	Football video broadcast	Number level accuracy: 83%
Li et al. (2018)	–	CNN for classification & spatial transformer network for localization of Jersey numbers	Live football video with single moving camera	Number level accuracy: 87% & digit level accuracy: 92%

Table 1: (continued)

Reference	Playfield detection	Player detection	Team sport & camera type	Evaluation
Liu and Bhanu (2019)	Region proposal network	R-CNN for digit localization and classification	Footbal video with single pan-tilt zooming camera	Number level accuracy: 92% & digit level accuracy: 94%

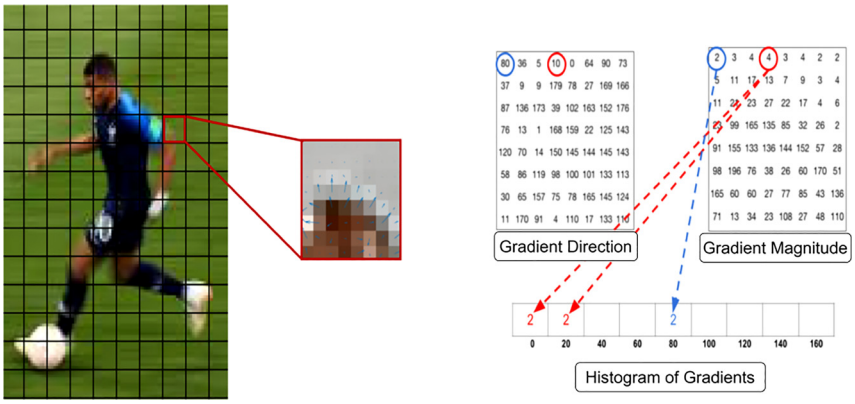


Figure 3: Calculating histogram of gradients.

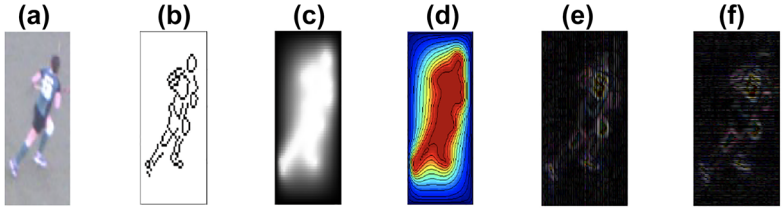


Figure 4: Edge detection methods: (a) Original frame, (b) binary edges with Canny method, (c) shape information image (Direkoglu, Sah, and O'connor 2018), (d) colored shape information image (Direkoglu, Sah, and O'connor 2018), (e) horizontal Sobel operator, (f) vertical Sobel operator.

evolution of a quantity like heat (here heat is considered as binary edges) over time. The solution of this equation is filling the inside object shape. This information image removes the appearance variation of the object, e.g., color or texture, while preserving the information of the shape. The result is the unique shape information for each player, which can be used for identification. This method works only for videos recorded with fixed cameras.

Sobel filtering: In the method by Naushad Ali, Abdullah-Al-Wadud, and Lee (2012) and Rao and Pati (2015), the Sobel gradient algorithm is used to detect horizontal and vertical edges (Figure 4(e) and (f)). The gradient is the vector with the components of (x,y) and the direction is calculated as $\tan^{-1}(\Delta y/\Delta x)$. Due to the similar color of the ball and the court lines, if the Sobel gradient algorithm is applied for background elimination instead of color segmentation, overlapping of the ball and court lines will not be a problem. However, general overlapping problems, e.g., player occlusion, cannot be handled with this method.

3.1.4 Supervised learning

In many proposed methods a robust classifier is trained to distinguish positive samples, i.e., players and/or ball, and negative samples, i.e., other objects or parts of the playfield. Any classification method, such as Support Vector Machine or Adaboost algorithms, can be trained for accurate detection of the players. Some examples of positive and negative sample frames are given in Figure 5.

Support vector machine: Several related works state that the advantages of SVM compared to other classifiers include better prediction, unique optimal solution, fewer parameters, and lower complexity. In the method

Figure 5: Positive (bottom) and negative (top) samples for training classifier.

of Zhu et al. (2006), the playfield is subtracted with a GMM. The results of background subtraction are thousands of objects, which SVM can help to classify into player and not player objects. However, in this method, the training dataset is manually labelled, which is time-consuming. In order to solve this problem, Chengjun (2018) proposed fuzzy decision making for automatic labelling of the training dataset.

Adaboost algorithm: Adaboost, short for Adaptive Boosting is used to make a strong classifier by a linear combination of many weak classifiers to improve detection accuracy. The main idea is to set the weights of the weak classifiers and to train on the data sample in each iteration until it can accurately classify the unknown objects. Markoski et al. (2015) used this algorithm for basketball players' face and body parts recognition. Although, they concluded that Adaboost is not accurate enough for object detection in sports events. Furthermore, Lehuger, Duffner, and Garcia (2007) showed that deep learning methods outperform the Adaboost algorithm for player detection.

3.2 Deep learning methods for detection

In the task of player detection, researchers usually use deep learning to recognize and localize jersey numbers. Most of the works in this area use a convolutional neural network (CNN) which is a deep learning model. The general architecture of CNN for digit recognition is illustrated in Figure 6. As the first step, players' bounding boxes should be detected. Then digits inside each bounding box should be accurately localized. These localized digits will be the input of CNN. Several convolution layers in CNN will assign importance to various features of the digits. Consequently, the neurons in the last layer will classify the digits from 0 to 9 classes. In this area, different works propose the following methods for improving the performance of detection: (1) how to localize digits inside each frame, (2) how to recognize multiple digits, (3) how to automatically label the training dataset, i.e., which benchmark dataset to use.

The first CNN-based approach for automatically recognizing jersey numbers from soccer videos was proposed by GerkeKarsten and Schäfer (2015). However, this method cannot recognize numbers in case of perspective distortion of the camera. To solve this problem, Li et al. (2018) used a spatial transformer network (STN) to localize jersey numbers more precisely. STN helps to crop and normalize the appropriate region of the numbers and improves the performance of classification. Another digit localization technique is region proposal network (RPN), which is a convolutional network that has a classifier and a regressor, and is trained end-to-end to generate high-quality region proposals for digits. RPN is used by Liu and Bhanu (2019) for classification and bounding-box regression over the background, person, and digits. While these methods can be more accurate than some traditional methods for player detection and they eliminate the necessities of manual feature description and extraction, they are also more expensive due to more computation and training

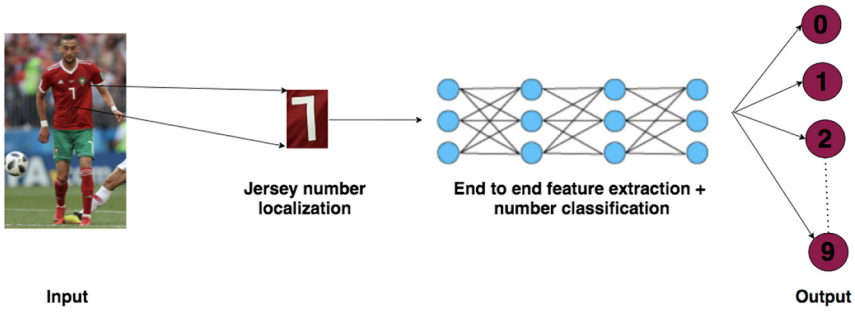


Figure 6: Neural network architecture for digit localization and detection.

time. Most of these methods require special versions of GPUs to be applied. Moreover, training and testing CNNs might be more time-consuming than running traditional methods.

4 Player tracking

Detection methods calculate the location of each player and the ball at each frame of the videos. There are always some frames for which the detection fails due to the blurriness of the frame, poor light conditions, occlusions, etc. In these cases, the detection methods cannot provide the location of the same player and ball in consecutive frames to construct continuous trajectories. Therefore, a player tracking method is needed to associate the partial trajectories, and to provide long tracking information of each of the players and the ball (see Figure 1). Player tracking involves the design of a tracker that can robustly match each observation to the trajectory of a specific player. This tracker can be designed for a single object or for multiple objects. The biggest challenge in tracking is the overlapping of players, namely the occlusion. Several studies suggested solutions for making a unique, continuous trajectory for each player by solving the occlusion problem. Those methods mostly follow filtering and data association. However, each method follows a different description for interest points (features) for filtering, and data association depends on the custom definition of probabilistic distributions. In this section, we survey the tracking methods classified by whether they are based on traditional or deep learning models.

4.1 Traditional methods for tracking

Same as the previously mentioned traditional detection models, the traditional tracking algorithms also require manual extraction and description of the player and ball features. The main categories of tracking methods in the literature of sports analytics are the following: point tracking, contour tracking, silhouette tracking, graph-based tracking, and data association methods.

4.1.1 Point tracking

The methods using point tracking mostly consider some points in the shape of the player and ball as the features, and choose the right algorithm (e.g., point distribution model (PDM), Kalman filter, particle filter) to associate those points through consecutive frames (see Figure 7).

Point Distribution Model: In these methods, the idea is to describe the statistical models of the shape of players and ball, called PDM. This method is used by several studies such as Mathes and Piater (2006); Hayet et al. (2005); Li and Flierl (2012). The shape is interpreted as the geometric information of the player, which is the residue once location and scaling are removed. As the first step, they extract the vector of features using two methods: Harris detector, or scale invariant feature transform (SIFT). Harris detector is the corner detection operator to extract corners and infer features of an image. Example results of Harris detector are shown as some points in Figure 7. SIFT is a feature detector algorithm to describe local features in images. These extracted features are detectable even under modifications in scale, noise, and illumination. Then, by learning the spatial relationships between these points, they construct the PDM to concatenate all feature vectors, i.e., interest points, of players (Figure 8). We provide a review and comparison of point tracking methods in Table 2.

Particle filter: All particle filter tracking systems aim to estimate the state of a system (x_t), given a set of noisy observations ($z_{1:t}$). Thus the goal is to estimate $P(x_t|z_{1:t})$. If we consider this problem as a Markov process, the solution can be found if the system is assumed to be linear and each conditional probability distribution is being modeled as a Gaussian. However, these assumptions cannot be made, as they decrease the accuracy of prediction. Particle filtering can help to eliminate the necessity of extra assumptions. This method approximates the probability distribution with a weighted set of N samples:

$$P(x) \sim \sum_{i=1}^{N} \omega^i (x - x_i), \tag{2}$$

where ω^i is the weight of the sample x_i. Now the questions are how to assign the weights, and how to sample the particles. Several studies suggested different methods for these questions.

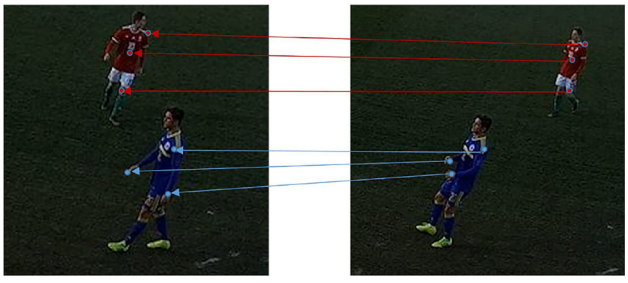
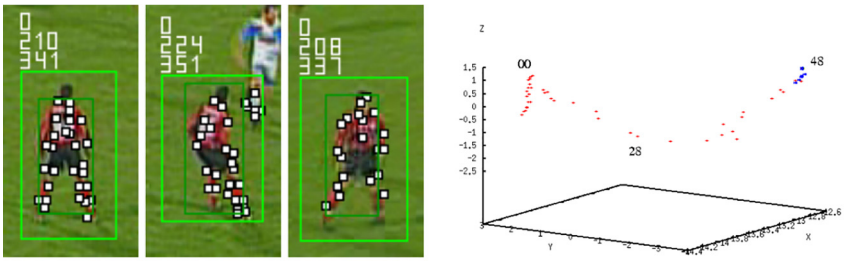


Figure 7: Point tracking.



(a) Players features from frames 0, 28, 48

(b) PDM corresponding to the normalized players' shape

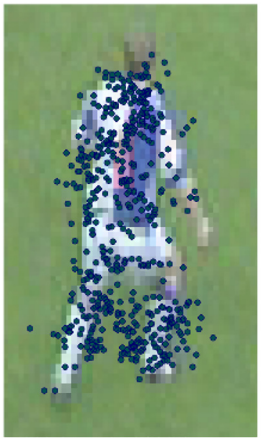
Figure 8: Describing shape by PDM from Mathes and Piater (2006).

Table 2: Review of tracking methods with PDM.

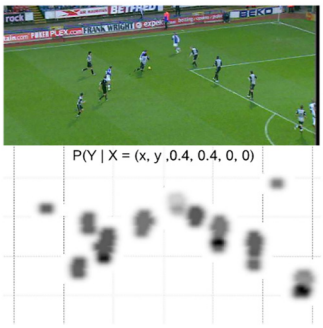
Reference	Tracking method	Point extraction method	Input video stream	Evaluation
Li and Flierl (2012)	Features tracking in consecutive frames	SIFT features	Football video with multiple stationary cameras	Average reliability of tracking, i.e., the number of correctly tracked players divided by the number of players in each frame is 99.7%; occlusion can be handled by comparing different viewpoints of cameras
Hayet et al. (2005)	Matching points of the PDM	Harris detector	Football video broadcast	Copes with the problem of rotating & zooming cameras by continuous image-to-model homography estimation; occlusion can be handled by interpolation in the PDM
Mathes and Piater (2006)	Points matching by maximum-gain using Hungarian algorithm	Harris detector	Football video broadcast	Can only track the non-rigid but textured objects in crowded scenes; occlusion can be handled by tracking sparse sets of local features

In the methods by Kataoka and Aoki (2011) and Manafifard, Ebadi, and Abrishami (2017a), particles are players' positions. Linear uniform motion is used to model the movement of particles, and the Bhattacharyya coefficient is applied for assigning weights, i.e., likelihood to each particle. In statistics, Bhattacharyya coefficient (BC) is a measure for the amount of overlap between two statistical samples (p, q) over the same domain x, and is calculated as $\mathrm{BC}(p, q) = \sum_x \sqrt{p(x)q(x)}$. In the works by Petsas and Kaimakis (2016) and Yang and Li (2017), each particle is estimated by the updated location of the player, knowing the last location plus a noise: $x_k = x_{k-1} + v_k$, which noise v_k is assumed to be i.i.d. following a zero-mean Gaussian distribution. Moreover, in Yang and Li (2017), particles are created based on color and edge features of players, and the weight of each particle is computed by contrast to the similarity between the particles and targets. Dearden, Demiris, and Grau (2006) introduced sample importance resampling to show that the shape of a player can be represented by a set of particles, e.g., edge, center of mass, and color pixels. Also, those points can represent a probabilistic distribution of the state of the player (Figure 9). Another method is proposed by de Pádua et al. (2015), in which players are detected by adaptive background subtraction method based on a mixture of Gaussians, and each detected player is automatically tracked by a separate particle filter and weighted average of particles. We show the above-mentioned methods for particle filtering in Table 3.

Kalman filter: (KF) method is mostly used in systems with the state-space format. In the state-space models, we have a set of states evolving over time. However, the observations of these states are noisy and we are sometimes



(a) Set of 500 particles for $P(x_t, y_t|y_t)$ of a player



(b) Posterior probability distribution function given the current state of a particle $P(y_t|x_t = x)$. Darker points represent higher probability

Figure 9: Particle filtering from Dearden, Demiris, and Grau (2006).

unable to directly observe the states. Thus, state-space models help to infer information of the states, given the observations, as new information arrives. In player and ball tracking, the observations of two inputs, i.e., time and noisy position measurements, continuously update the tracker. The role of KF is to estimate the x_t, given the initial estimate of x_0, and time-series of measurements (observations), $z_1, z_2, \ldots, z_t$. The KF process defines the evolution of state from time $t-1$ to t as:

$$x_t = Fx_{t-1} + Bu_{t-1} + \omega_{t-1}, \tag{3}$$

where F is the transition matrix for state vector x_{t-1}, B is the control-input matrix for control vector u_{t-1}, and ω_{t-1} is the noise following a zero-mean Gaussian distribution. A typical KF process is shown in Figure 10. As we can see, the Kalman filter and particle filter are both recursively updating an estimate of the state, given a set of noisy observations. Kalman filter performs this task by linear projections (3), while the Particle filter does so by estimating the probability distribution (2).

The following studies use Kalman filter for player and ball tracking: Makandar and Mulimani (2018), Kim and Kim (2009), and Liu, Liu, and Huang (2011). We summarize the KF methods in Table 4.

4.1.2 Contour tracking

Contour tracking for dynamic sports videos provides basic data, such as orientation and position of the players, and is used when we have deforming objects, i.e., players and

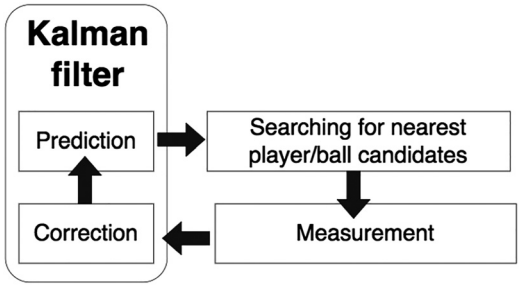


Figure 10: Typical Kalman filter process.

Table 3: Review of particle filtering methods.

Reference	Particles type	Weight assignment method	Input video stream	Evaluation
Kataoka and Aoki (2011)	Players' position & center of gravity	Bhattacharyya coefficient	Football video with single swing motion camera	Tracking rate for players: 83% & ball: 98%; occlusion handling by combining particle filter and real AdaBoost
Petsas and Kaimakis (2016)	Players' position	Weighted average of particles	Football video with single stationary camera	Not real-time; occlusion cannot be handled
Yang and Li (2017)	Color & edge features	Bhattacharyya coefficient	Football video broadcast	Occlusion handling by comparing color & edge features
Dearden, Demiris, and Grau (2006)	Edge points, center of mass, color pixels	Sample importance resampling	Football video from single moving camera	Overcomes the problem of non-linear and non-Gaussian nature of the noise model
Manafifard, Ebadi, and Abrishami (2017a)	Ellipse surrounded by the player bounding box	Bhattacharya coefficient	Football video broadcast	92% of accuracy; occlusion can be handled by combination of particle swarm optimization & multiple hypothesis tracking

Table 4: Summary of player and ball tracking methods with Kalman filter.

Reference	KF type	KF inputs	Input video stream	Evaluation
Makandar and Mulimani (2018)	KF with motion information	Players motion information (moving or static)	Volleyball video broadcast	Non-linear & non-Gaussian noise are ignored, which decreases the accuracy of tracking
J.-Y. Kim and T.-Y. Kim (2009)	Dynamic KF	Position & velocity of state vector	Football video broadcast	Copes with the problem of player-ball occlusion in KF
Liu, Liu, and Huang (2011)	Kinematic model of KF	Position state from mean-shift algorithm	Basketball video broadcast	KF is used to confirm the target location to empower mean-shift algorithm for ball tracking

ball, over consecutive frames. Figure 11 shows some examples of such contours. Many methods have been proposed to track these contours. In an easy approach, the centroid of these contours plus the bounding box of players will be obtained, and the player can be traced (Beetz et al. 2007; Hanzra and Rossi 2013). Researchers in this area tried to propose several methods for assigning a suitable contour to the players and the ball. Patil et al. (2018) find player's contours as curves, joining all the continuous points (along the boundary), having the same color or intensity. So they could track these contours and decide whether the player is in an offside position or not. Another method by Lefèvre et al. (2000, 2002), and Lin (2018) suggests snake or active contour tracking, which does not include any position prediction. In such methods, the algorithm fits open or close splines (i.e., a special function defined piecewise by polynomial) to lines or edges of the players. An active contour can be represented as a curve: $[x_t, y_t], t \in [0, 1]$ segmenting players from the rest of the image, which can be closed or not. Then this curve should be iteratively deformed and converged to target contour (Figure 12) to minimize an energy function and to fit the curve to the lines or edges of the players. The energy function is presented as physical properties of the contours, i.e., the shape of the contour, plus the gradient and intensity of the pixels in the contour. A review of contour representation of the above-mentioned tracking methods is in Table 5.

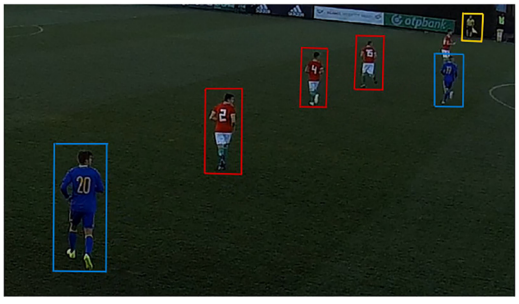


Figure 11: Contour tracking.

4.1.3 Silhouette tracking

When the information provided by contour and simple geometric shapes are not enough for the tracking algorithm, extracting the silhouette of the players and of the ball can provide extra information on the appearance of the object in consecutive frames. Unlike contours, the silhouette of a player is not a curved shape. Thus, it does not require deformation and convergence to the target shape of players and the ball. Instead, this method proposes some aspect ratios to describe the invariant shape. An example of this shape extraction for a specific player is illustrated in Figure 13. In such cases, shape analysis can help the tracking process as follows.

Shape matching: In the literature, the shape of an object is defined by its local features not determined or altered by additive contextual effects, e.g., location, scale and rotation. This method is mostly used for ball tracking. The problem in this area is that the shape of the ball varies

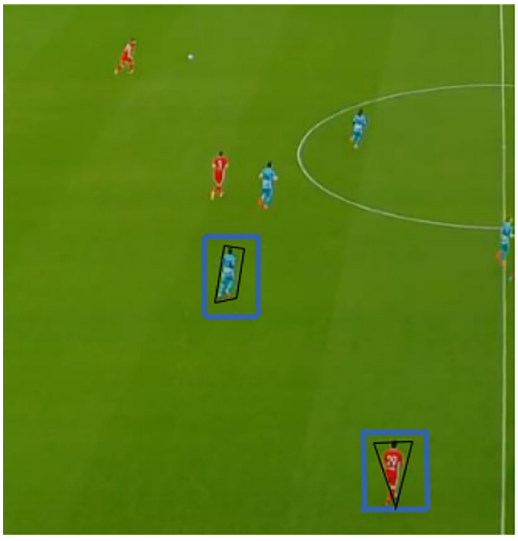


Figure 12: Active contour model for fitting curves to the players' edges and lines.

Table 5: Summary of contour tracking methods.

Reference	Contour representation	Tracking method	Input video stram	Evaluation
Patil et al. (2018)	A curve that joins all continuous pixels	Contour filtering for players & ball with Gaussian blurring	Football video with multiple stationary cameras	Fast tracking; occlusion can be handled by placing cameras on both sides of the field
Lefèvre et al. (2000, 2002)	Snake initialization	Snake deformation	Football video with single moving camera	Robustly solves occlusion
Hanzra and Rossi (2013)	Edge pixels form contour boundaries	Contour centroid tracking	Football video with 3 stationary and 1 moving cameras	Handles occlusion by comparing contour area of player & mean of that for all players
Beetz et al. (2007)	K-means clustering of pixels on marked regions	Multiple hypothesis tracker	Football video broadcast	Tracks players up to 20 min without getting lost; detection rate is over 90%
Lin (2018)	Motion curve of shooting arm	Iterative convergence of dynamic contour with Lagrange equation	Basketball video broadcast	Occlusion can be handled by minimizing the potential energy of the system image

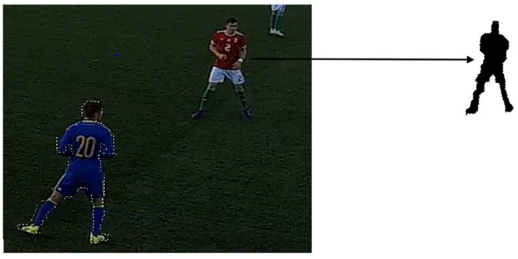


Figure 13: Silhouette tracking.

significantly in each frame, and does not look like a circle at all (Figure 14). Different studies suggest some aspect ratios, i.e., shape descriptors, to get the near-circular ball images. Chakraborty and Meher (2013) suggest using the degree of compaction C_d which is the ratio of the square of the perimeter of the given shape to the area of the given shape: $C_d = P^2/4\pi A$. Therefore, if $C_d > 50\%$, the shape can be filtered as a ball. Another shape descriptor is eccentricity, proposed by Naidoo and Tapamo (2006), and it is defined as the ratio of the longest diameter to the shortest diameter of a shape. The form factor indicates how circular an object is, and if the result is between [0.2, 0.65] they will consider it as a ball. Besides these shape descriptors, Huang, Llach, and Bhagavathy (2007) proposed using skeletons to separate a shape's topological properties from its geometries. To extract the skeleton for every foreground blob, they use the Euclidean distance transform. Table 6 shows a review of shape analysis in player and ball tracking methods.

4.1.4 Graph-based tracking

Some works explore graph-based multiple-hypothesis to perform player tracking. In these cases, a graph is constructed that shows all the possible trajectories of players, and it models their positions along with their transition between frames. The correct trajectory is found with the help of, e.g., similarity measure, linear programming, multi-commodity network flow, or the problem is modeled as a minimum edge cover problem. An example of graph tracking in consecutive frames is shown in Figure 15. The method shown by Figueroa et al. (2004) builds the graph in such a way that nodes represent blobs and edges represent the distance between these blobs. Then tracking of each player is performed by searching the shortest path in the graph. However, occlusion is difficult to be handled with

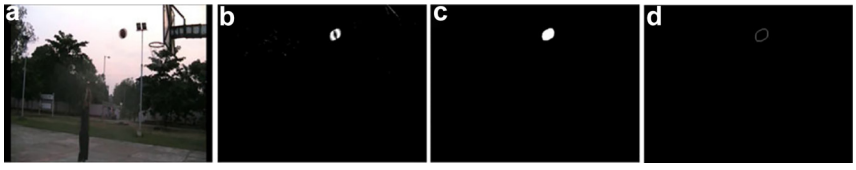


Figure 14: Shape of the moving ball from Chakraborty and Meher (2012).

Table 6: Summary of shape analysis in player and ball tracking methods.

Reference	Tracking method	Input video stream	Evaluation
Chakraborty and Meher (2013)	Shape, size and compaction filtering	Basketball video broadcast	93% of accuracy of ball detection and tracking; occlusion can be handled by trajectory interpolation with regression analysis
Naidoo and Tapamo (2006)	Moore-neighbour tracing algorithm	Football video with single stationary camera	Shape analysis in this method is failing in case of the shadow of players or the ball
Huang, Llach, and Bhagavathy (2007)	Euclidean distance transform	Football video broadcast	Occlusion cannot be handled

this method. Authors of Pallavi et al. (2008) used dynamic programming to find the optimal trajectory of each player in the graph. The proposed method by Xing et al. (2011) builds an undirected graph to model the occlusion relationships between different players. In Chen, Chang, and Hsiao (2017), the method constructs a layered graph for detected players, which includes all probable trajectories. Each layer corresponds to a frame and each node represents a player. Two nodes of adjacent layers are linked by an edge if their distance is less than a pre-defined threshold. Finally, the authors used the Viterbi algorithm in dynamic programming to extract the shortest path of the graph. Ball tracking with graphs was proposed in Maksai, Wang, and Fua (2015), where they build a ball graph to formulate the Mixed Integer Programming model, and each node is associated with a state, i.e., location of the ball at a time instance. Table 7 shows a review of node and edge representation, along with tracking methods defined on the graph.

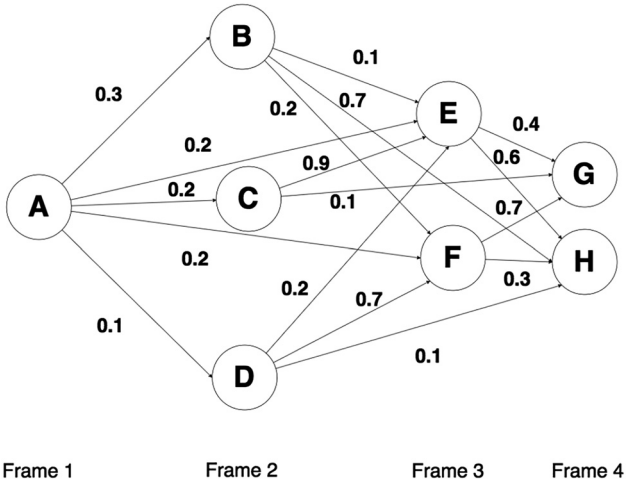


Figure 15: An example of weighted graph for player tracking in 4 consecutive frames.

4.1.5 Data association methods

Simulation-based approaches, including Monte Carlo methods and joint probabilistic data association, are usually used for solving multitarget tracking problems, as these methods perform well for nonlinear and non-Gaussian data models.

Markov chain Monte Carlo data association (MCMC): Septier et al. (2011) compared several MCMC methods, such as (1) sequential importance resampling algorithm, (2) resample-move, (3) MCMC-based particle method. The difference between these methods stems from the sampling strategy from posterior by using previous samples. Simulations show that the MCMC-based Particle approach exhibits better tracking performance and thus clearly represents interesting alternatives to Sequential Monte Carlo methods. The authors of Liu, Tong, and Li (2009) designed a Metropolis-Hastings sampler for MCMC, which increased the efficiency of the method.

Joint probabilistic data association (JPDA): The JPDA method can be used when the mapping from tracks to observations is not clear, and we do not know which observations are valid and which are just noise. In these cases, JPDA implements a probabilistic assignment. Abbott and Williams (2009) used JPDA to assign the probability of association between each observation and each track.

4.2 Deep learning-based tracking

Despite the effectiveness of traditional methods, they fail in many real-world scenarios, e.g., occlusion, and processing videos from several viewpoints. On the other hand, deep learning models benefit from the learning ability of neural networks on large and complex datasets, and they eliminate the necessities of features extraction by the human/expert. Therefore, deep learning-based trackers are recently getting much attention in computer vision. These trackers are categorized into online and offline methods: online trackers are trained from scratch during the test

Table 7: Summary of graph-based player and ball tracking methods.

Reference	Node representation	Edge representation	Tracking method	Input video stream	Evaluation
Figueroa et al. (2004)	Blobs	Distance between blobs	Minimal path of graph	Football video with multiple stationary cameras	The algorithm is tested for 3 players of defender, mid-fielder, and forwarder, and shows 88% of solved occlusions
Pallavi et al. (2008)	Probable player candidates	Candidates link between frames	Dynamic programming with acyclic graph	Football video broadcast	93% of accuracy in tracking & 80% of solved occlusions
Xing et al. (2011)	Player	Relationship ratio between 2 players	Dual-mode two-way Bayesian inference approach	Football and basketball video broadcast	Uses undirected graph to model the occlusion relationships & reports 119 mostly tracked trajectories & 12 ID switches
Chen et al. (2017)	Players position	Degree of closeness between players	Viterbi algorithm to find shortest path	Basketball video broadcast	88% of precision in player tracking; occlusion is handled by layered graph connections
Maksai et al. (2015)	Ball's location	State-time instant connection	Mixed integer programming	Football, volleyball, and basketball video with multiple cameras	97% of accuracy in player tracking & 74% in ball tracking

and are not taking advantage of already annotated videos for improving performance, while offline trackers train on offline data.

Several recent studies have attempted to assess the performance of deep learning methods in sports analytics. The core idea of all methods is to use CNN. However, each study proposes a different structure of the network and training method for increasing the performance. In this section, we summarize the state-of-the-art networks and their application in sports analytics. Table 8 is a brief review of these methods.

Visual geometry group (VGG): VGG-M is a CNN architecture, designed by the VGG at the University of Oxford. This network is used by several studies such as Kamble, Keskar, and Bhurchandi (2019); Arbues, Ballester, and Haro (2019). VGG-M is a small type of CNN, and its pretrained weights are publicly available. This network gets the image as input, and classifies the detected object as player, ball, or background, along with the probability of the classes. The architecture of VGG-M CNN is illustrated in Figure 16.

After the classification of the players and ball, the metric called intersection over union (IOU) is used to track them. IOU is the ratio of intersection of the ground truth bounding box from the previous frame (BB_A), and predicted bounding box in the current frame (BB_B), and it is calculated as in (4):

$$IOU = \frac{|BB_A \cap BB_B|}{|BB_A \cup BB_B|}, \tag{4}$$

where $\cap$ and $\cup$ are intersection and union in terms of the number of pixels. Thus, if the intersection is nonzero between consecutive frames, the player or ball can be traced.

Cascade-CNN: Is a novel deep learning architecture consisting of multiple CNNs. This network is trained on labeled image patches and classifies the detected objects into the two classes of player and non-player. Football and basketball player tracking using this method is suggested by Lu et al. (2017). The illustrated pipeline in Figure 17 shows the classification process and a dilation strategy for accurate player tracking with the help of IOU metric.

Table 8: Summary of deep learning methods and application in team sports.

Reference	Network structure	Input video stream	Required computational resource(s)	Performance
Lu et al. (2017)	Cascade CNN	Football and basketball video broadcast	Intel i7-6700HQ; NVIDIA GTX1060	AUC of player detection is 0.97
Kamble, Keskar, and Bhurchandi (2019)	VGG-M	Football video with multiple stationary cameras	MATLAB 2018a; intel i7; NVIDIA GTX1050Ti	87% of accuracy in player, ball, and event detection
Long (2019)	Full-convolution siamese NN	Football video broadcast	Matlab 2014a; intel i7; NVIDIA GTX 960 M	Mean value of target tracking effect of SiamCNN is 60%
Yoon et al. (2019)	YOLO	Basketball video with single moving camera	Intel i7; NVIDIA GeForce GTX 1080Ti	74% of precision in recognizing Jersey numbers; MAPE is at most 34%
Arbues, Ballester, and Haro (2019)	VGG-19	Basketball video with single moving camera	–	Detection precision: 98%; MOTA of tracking: 68%
Buric, Pobar, and Ivasic-Kos (2019)	YOLO	Handball video with multiple stationary cameras	12 core E5-2680v3 CPU; GeForce GTX TITAN	mAP in players & ball detection: 37%

AUC, area under curve; mAP, mean average precision; MAPE, mean absolute percentage error; MOTA, multi object tracking accuracy.

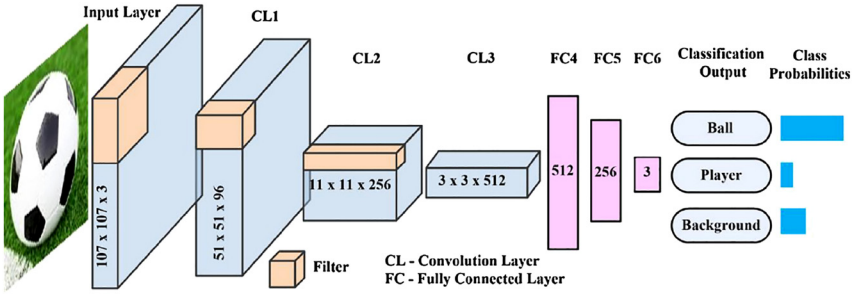


Figure 16: VGG-M CNN architecture from Kamble, Keskar, and Bhurchandi (2019).

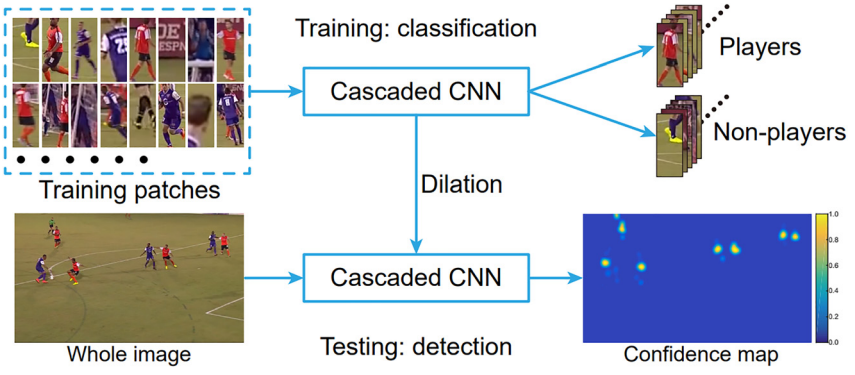
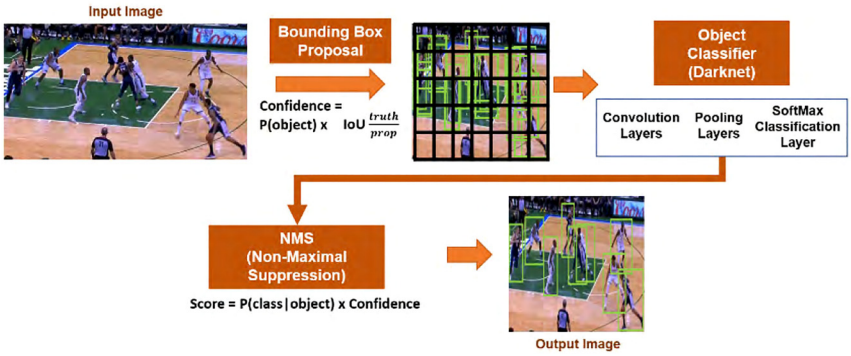
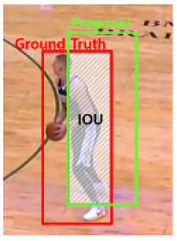


Figure 17: Classification process with Cascade-CNN from Lu et al. (2017).

YOLO: This network is used by Buric, Pobar, and Ivasic-Kos (2019) for handball player and ball tracking, and Yoon et al. (2019) for basketball player movement recognition. YOLO applies a single neural network to the full image. Then the network divides the image into cells and predicts bounding boxes and probabilities for each cell. The weights of the bounding boxes are the predicted probabilities. Then IOU metric can help for tracking purposes



(a) Player and ball classification with YOLO



(b) IOU evaluation for tracking

Figure 18: Player and ball tracking with YOLO from Yoon et al. (2019).

and solving the occlusion problem of the players and the ball (Figure 18).

SiamCNN: In this network, there are sister network 1 and sister network 2 with the same network structure, parameters, and weights. The structure looks like VGG-M except for the adjustment of the sizes of each layer. The inputs of SiamCNN are 3-color channels (R, G, B) from frames, and the output is the Euclidean distance between the characteristics/features of the inputs. Long (2019) used this network to extract players' characteristics through trajectories. Then they compare the similarities between search areas and a target template, so players can be tracked. The structure of this network is given in Figure 19.

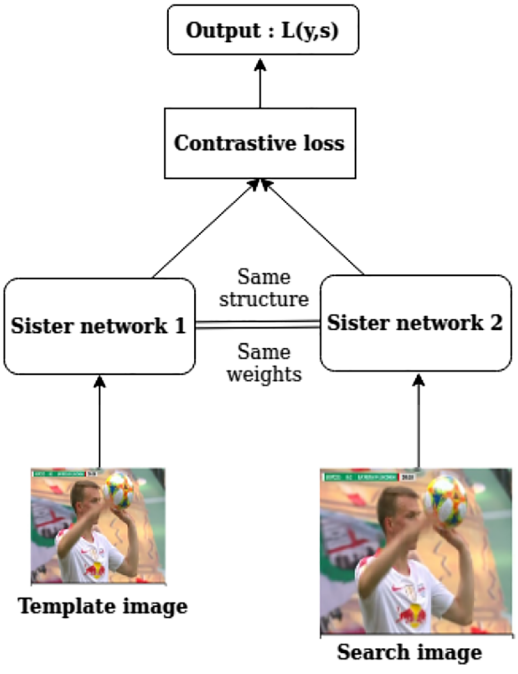


Figure 19: SiamCNN network structure for player tracking.

5 Evaluation and model selection

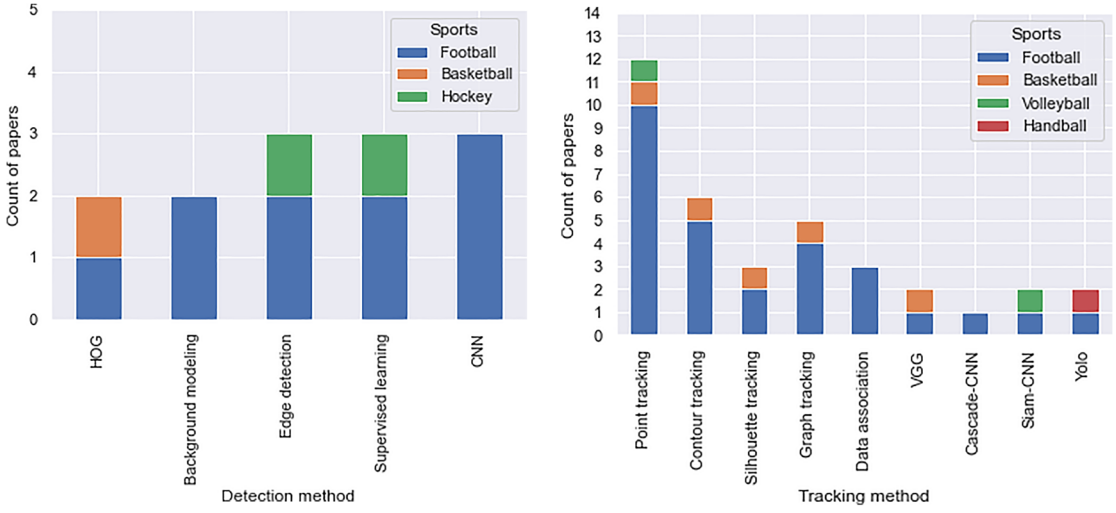
If a clean set of tracking information is not provided to a sports analyzer who is developing a quantitative model, his/her core task is to choose the most suitable method for tracking players and the ball, and construct the required

dataset for further analysis. In the detection and tracking domains, model selection, i.e., DL or traditional, heavily depends on the task at hand. The selection will be difficult by merely reviewing the performance metrics of the methods, as the tracking performance relies on the specific task at hand and the quality of the videos. However, there are some concrete criteria in this domain, which can help the analyst to rapidly choose the desired tracking method. Figure 20 compares the number of publications in detection and tracking domains categorized by team sports. Note that 74% of methods are applied on football videos, whereas deep learning methods (i.e., CNN, VGG, Cascade, Siam, Yolo) are covering only 20% of all publications. In this section, we review the benefits and drawbacks of each method, and compare them in terms of their estimated costs.

5.1 Deep learning-based versus traditional methods

In general, traditional methods are domain-specific, thus the analyzer must specifically describe and select the features (e.g., edge, color, points, etc.) of the ball, football player, basketball player, background, etc. in detail. Therefore, the performance of the traditional models depends on the analyzer's expertise and how accurate the features are defined. DL methods, on the other hand, demonstrate superior flexibility and automation in detection and tracking tasks, as they can be trained offline on a huge dataset, and then automatically extract features of any object type. In this case, the necessities of manual feature extraction are eliminated, and consequently, DL requires less expertise from the analyzer. In another point of view, DL models are more like a black box on the detection tasks. On the contrary, traditional methods provide more visibility and interpretability to the analyzer on how the developed algorithm can be performed in different situations such as sports types, lighting conditions, cameras, video quality, etc. So, traditional models can give a better opportunity to improve the tracker accuracy, when the system components are visible. Also in the case of failure, system debugging are more straightforward in traditional models than DL-based ones.

In addition to the pros and cons that are listed in this survey for each method, few criteria can help sports analysts to choose their desired method. Table 9 lists these criteria that can help analyzers to choose the suitable detection and tracking methods in the direction of DL-based or traditional ones.



(a) Detection task **(b)** Tracking task

Figure 20: Number of the published papers for each method categorized by their application in team sports.

Table 9: Model selection criteria.

Criteria	Deep Learning	Traditional
Availability of huge training dataset	✓	
Accessing to high computational power	✓	
Lack of storage		✓
Looking for cheaper solution		✓
Certainty and expertise in the object features		✓
Less domain expertise	✓	
Flexibility in terms of objects and training dataset	✓	
Flexibility of deployment on different hardware		✓
Short training and annotation time		✓

5.2 Cost analysis

The cost of the method is one of the most important characteristics of model selection for researchers and analysts: they are looking for a method with maximum accuracy and reasonable cost. Here we give an insight into the cost of the state-of-the-art methods, both for infrastructure and computation, and classify them into 3 categories: high, medium, low. The classification is based on the following facts. In the computational aspect, deep learning methods which require GPUs are more expensive than traditional methods with only CPUs. From an infrastructure perspective, different methods require different sets of camera settings to record the sports video. Methods that require a set of moving or stationary camera(s) to be set up in the arena are more expensive than the methods that can trace players and the ball on broadcast video. Table 10 shows the cost approximation of all methods along with their most significant limitations.

Table 10: Comparing cost of the methods.

Reference	Detection or tracking method	Cost approximation	Infrastructure requirements for sport video			Computational requirements		Limitation
			Moving camera(s)	Stationary camera(s)	Broadcast	GPU	CPU	
GerkeKarsten and Schäfer (2015)	Deep learning	Middle			✓	✓		Poor performance in perspective distortion of camera
Li et al. (2018)	Deep learning	High	✓			✓		Expensive manual force for labeling
Liu and Bhanu (2019)	Deep learning	High		✓		✓		High network training time
Yoon et al. (2019)	Deep learning	High	✓			✓		Very low accuracy
Arbues, Ballester, and Haro (2019)	Deep learning	High	✓			✓		Difficult network tuning
Kamble, Keskar, and Bhurchandi (2019)	Deep learning	High		✓		✓		Manual network parameters is needed to be assigned
Buric, Pobar, and Ivasic-Kos (2019)	Deep learning	High		✓		✓		–

Table 10: (continued)

Reference	Detection or tracking method	Cost approximation	Infrastructure requirements for sport video			Computational requirements		Limitation
			Moving camera(s)	Stationary camera(s)	Broadcast	GPU	CPU	
Lu et al. (2017)	Deep learning	Middle			✓	✓		Unrealistic uniform background color is assumed
Long (2019)	Deep learning	Middle			✓	✓		Too sensitive on number of convolutional layers
Mackowiak et al. (2010)	HOG	Low			✓		✓	Cannot detect occluded players
Cheshire, Hu, and Chang (2015)	HOG	Low			✓		✓	–
Ming et al. (2009)	GMM	Middle	✓				✓	Performs only in absence of shadow
Mazzeo et al. (2008)	Energy evaluation	Middle		✓			✓	High computational time
Direkoglu, Sah, and O'connor (2018)	Edge detection	Middle		✓			✓	High computational time and only with fixed camera
Naushad Ali, Abdullah-Al-Wadud, and Lee (2012), Rao and Pati (2015)	Sobel gradient	Low			✓		✓	Low performance in crowded places
Markoski et al. (2015)	Adaboost	Middle	✓				✓	Highly training time
Zhu et al. (2006)	SVM	Low			✓		✓	Time consuming due to manual labeling
Chengjun (2018)	SVM	Low			✓		✓	–
Li and Flierl (2012)	Point tracking	Middle		✓			✓	–
Hayet et al. (2005)	Point tracking	Low			✓		✓	Extracted trajectories are too short
Mathes and Piater (2006)	Point tracking	Low			✓		✓	Cannot track untextured objects
Kataoka and Aoki (2011)	Particle filter	Middle	✓				✓	–
Petsas and Kaimakis (2016)	Particle filter	Middle		✓			✓	Tracking fails in case of player jumping or falling
Yang and Li (2017)	Particle filter	Low			✓		✓	Inadequacy of identifying players
Dearden, Demiris, and Grau (2006)	Particle filter	Middle	✓				✓	Can track players only in image space, not in real-time moving camera system

Table 10: (continued)

Reference	Detection or tracking method	Cost approximation	Infrastructure requirements for sport video			Computational requirements		Limitation
			Moving camera(s)	Stationary camera(s)	Broadcast	GPU	CPU	
Manafifard, Ebadi, and Abrishami (2017a)	Particle filter	Low			✓		✓	Players' color features are pre-selected, but they are changing in each game
de Pádua et al. (2015)	Particle filter	Low		✓			✓	Lots of tracking id switches
Makandar and Mulimani (2018)	Kalman filter	Low			✓		✓	Non-linear, non-Gaussian noises are ignored
J.-Y. Kim and T.-Y. Kim (2009)	Kalman filter	Low			✓		✓	This algorithm fails in the frames with crowded players
Liu, Liu, and Huang (2011)	Kalman filter	Low			✓		✓	Shot event is required for ball tracking
Patil et al. (2018)	Contour tracking	Middle		✓			✓	Offside event is required for tracking
Lefèvre et al. (2000, 2002)	Contour tracking	Middle	✓				✓	High computational time
Hanzra and Rossi (2013)	Contour tracking	Middle	✓	✓			✓	Manual camera setting and zooming is required
Beetz et al. (2007)	Contour tracking	Low			✓		✓	–
Lin (2018)	Contour tracking	Low			✓		✓	Shooting arm is required for tracking
Chakraborty and Meher (2013)	Shape matching	Low			✓		✓	Long shot sequences are required for ball tracking
Naidoo and Tapamo (2006)	Shape matching	Middle		✓			✓	This algorithm fails in shadow
Huang, Llach, and Bhagavathy (2007)	Shape matching	Low			✓		✓	–
Figueroa et al. (2004)	Graph-based	Middle		✓			✓	–
Pallavi et al. (2008)	Graph-based	Low			✓		✓	Short tracking sequence as it focuses on solving occlusion
Xing et al. (2011)	Graph-based	Low			✓		✓	–
Chen, Chang, and Hsiao (2017)	Graph-based	Low			✓		✓	–
Maksai, Wang, and Fua (2015)	Graph-based	Middle	✓				✓	–

6 Conclusion and future research directions

According to a large number of cited papers in this survey, computer vision researchers intensively investigate robust methods of optical tracking in sports. In this survey, we have categorized the literature according to the applied methods and video type they build on. Moreover, we elaborated on the detection phase, as a necessary preprocessing step for tracking by conventional and deep learning methods. We believe that this survey can significantly help quantitative analysts in sports to choose the most accurate, while cost-effective tracking method suitable for their analysis. Furthermore, the combination of traditional and deep learning methods can be rarely seen in the literature. Traditional models are time-consuming and require domain expertise due to some manual feature extraction tasks, while deep learning models are quite expensive to run in terms of computing resources. As possible future work, research may aim to combine those methods to increase the performance of tracking systems, along with the robust quantitative evaluation of the games. Another avenue for future work might be to minimize the computational costs of tracking systems with the aid of sophisticated data processing methods. We hope that this survey can give an insight to sports analytics researchers to recognize the gaps of state-of-the-art methods, and come up with novel solutions of tracking and quantitative analysis.

Acknowledgment: Project no. 128233 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the FK_18 funding scheme.

Author contribution: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: Project no. 128233 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the FK_18 funding scheme.

Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

Abbott, R. G., and L. Williams. 2009. "Multiple Target Tracking with Lazy Background Subtraction and Connected Components Analysis." *Machine Vision and Applications* 20 (2): 93–101.

Agelet Ruiz, N. 2010. "Tracking of a Basketball Using Multiple Cameras." PhD Thesis, University of Polytecnica de Catalunya.

Alavi, A. 2017. "Investigation into Tracking Football Players from Video Streams Produced by Cameras Set up for TV Broadcasting." *American Journal of Engineering Research* 6: 95–104.

Arbues, A., C. Ballester, and G. Haro. 2019. "Single-camera Basketball Tracker through Pose and Semantic Feature Fusion." arXiv preprint, arXiv:1906.02042v2.

Beetz, M., S. Gedikli, J. Bandouch, B. Kirchlechner, N. v. Hoyningen Huene, and A. Perzylo. 2007. "Visually Tracking Football Games Based on Tv Broadcasts." In *International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 7, 2066–71.

Buric, M., M. Pobar, and M. Ivasic-Kos. 2019. "Adapting Yolo Network for Ball and Player Detection." In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, Vol. 1, 845–51.

Burke, B. 2019. "DeepQB: Deep Learning with Player Tracking to Quantify Quarterback Decision-Making & Performance." In *MIT SLOAN Sports Analytics Conference*. Boston: MIT Sloan.

Chakraborty, B., and S. Meher. 2012. "Real-Time Position Estimation and Tracking of a Basketball." In *IEEE International Conference on Signal Processing, Computing and Control*, 1–6.

Chakraborty, B., and S. Meher. 2013. "A Real-Time Trajectory-Based Ball Detection-And-Tracking Framework for Basketball Video." *Journal of Optics* 42 (2): 156–70.

Chen, L.-H., H.-W. Chang, and H.-A. Hsiao. 2017. "Player Trajectory Reconstruction from Broadcast Basketball Video." In *ICBIP Proceedings of the 2nd International Conference on Biomedical Signal and Image Processing*, 72–6.

Chengjun, C. 2018. "Player Detection Based on Support Vector Machine in Football Videos." *International Journal of Performability Engineering* 14 (2): 309–19.

Cheshire, E., M.-C. Hu, and M.-H. Chang. 2015. "Player Tracking and Analysis of Basketball Plays." In *European Conference of Computer Vision*. Education.

Ciaparrone, G., F. Luque-Sanchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. 2019. "Deep Learning in Video Multi-Object Tracking: A Survey." *Journal of Neurocomputing* 4: 1–42.

de Pádua, P. H. C., F. L. C. Pádua, M. T. D. Sousa, and M. d. A. Pereira. 2015. "Particle Filter-Based Predictive Tracking of Futsal Players from a Single Stationary Camera." In *SIBGRAPI Conference on Graphics, Patterns and Images*, 134–41.

Dearden, A., Y. Demiris, and O. Grau. 2006. "Tracking Football Player Movement from a Single Moving Camera Using Particle Filters." In *The 3rd European Conference on Visual Media Production – Part of the 2nd Multimedia Conference*, 29–37.

Dhenuka, M., K. Udesang, and D. Hemant. 2018. "Multiple Object Detection and Tracking: A Survey." *International Journal for Research in Applied Science and Engineering Technology* 6 (2): 809–13.

Direkoglu, C., M. Sah, and N. O'connor. 2018. "Player Detection in Field Sports." *Machine Vision and Applications* 29 (2): 187–206.

Figueroa, P., N. Leite, R. Barros, I. Cohen, and G. Medioni. 2004. "Tracking Soccer Players Using the Graph Representation." In *Proceedings of the 17th International Conference on Pattern Recognition*, 787–90.

GerkeKarsten, S., and M. Schäfer. 2015. "Soccer jersey Number Recognition Using Convolutional Neural Networks." In *The IEEE*

International Conference on Computer Vision Workshops, 734–41.

Hanzra, B. S., and R. Rossi. 2013. "Automatic Cameraman for Dynamic Video Acquisition of Football Match." In *IEEE Proceedings of the Second International Conference on Image Information Processing (ICIIP)*, 142–7.

Hayet, J., T. Mathes, J. Czyz, J. Piater, J. Verly, and B. Macq. 2005. "A Modular Multi-Camera Framework for Team Sports Tracking." In *IEEE Conference on Advanced Video and Signal Based Surveillance*, 493–8.

Huang, Y., J. Llach, and S. Bhagavathy. 2007. "Players and Ball Detection in Soccer Videos Based on Color Segmentation and Shape Analysis." In *Multimedia Content Analysis and Mining, International Workshop*. MCAM. Weihai: Springer.

Kamble, P., A. Keskar, and K. Bhurchandi. 2019. "A Deep Learning Ball Tracking System in Soccer Videos." *Opto-Electronics Review* 27 (1): 58–69.

Kataoka, H., and Y. Aoki. 2011. "Football Players and Ball Trajectories from Single Camera's Image." In *17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*.

Kim, J.-Y., and T.-Y. Kim. 2009. "Soccer Ball Tracking Using Dynamic Kalman Filter with Velocity Control." In *Sixth International Conference on Computer Graphics, Imaging and Visualization*, 367–74.

Lee, B., L. Liew, W. Cheah, and Y. Wang. 2014. "Occlusion Handling in Videos Object Tracking: A Survey." In *IOP Conference Series: Earth and Environmental Science*, Vol. 18, 1–5.

Lefèvre, S., C. Fluck, B. Maillard, and N. Vincent. 2000. "A Fast Snake-Based Method to Track Football Player." In *Proceedings of the IARP Conference on Machine Vision Applications (IARP MVA)*. Tokyo: RFAI publication, 501–4.

Lefèvre, S., G. Jean-Pierre, A. Piron, and N. Vincent. 2002. "An Extended Snake Model for Real-Time Multiple Object Tracking." In *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS)*. Ghent: Ghent University.

Lehuger, A., S. Duffner, and C. Garcia. 2007. "A Robust Method for Automatic Player Detection in Sport Videos." In *Orange Labs, Cesson-Sévigné*. Paris: Orange Labs.

Li, H., and M. Flierl. 2012. "Sift-based Multi-View Cooperative Tracking for Soccer Video." In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Li, G., S. Xu, X. Liu, L. Li, and C. Wang. 2018. "Jersey Number Recognition with Semi-supervised Spatial Transformer Network." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1864–7.

Lin, M. 2018. "Contour Tracking Algorithm for Dynamic Image of Basketball Shooting Arm." *Journal of Discrete Mathematical Sciences and Cryptography* 21 (2): 299–304.

Liu, H., and B. Bhanu. 2019. "Pose-guided R-CNN for Jersey Number Recognition in Sports." In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Long Beach: Computer Vision Foundation/IEEE.

Liu, J., X. Tong, and W. Li. 2009. "Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video." *Pattern Recognition Letters* 30 (2): 103–13.

Liu, Y., X. Liu, and C. Huang. 2011. "A New Method for Shot Identification in Basketball Video." *Journal of Software* 6 (8): 1468–75.

Long, T. 2019. "Research on Application of Athlete Gesture Tracking Algorithms Based on Deep Learning." *Journal of Ambient Intelligence and Humanized Computing* 11 (2): 1–9.

Lu, K., J. Chen, J. J. Little, and H. He. 2017. "Light Cascaded Convolutional Neural Networks for Accurate Player Detection." In *Proceedings of the British Machine Vision Conference (BMVC)*, 173.1–13. bmva Press.

Mackowiak, S., J. Konieczny, M. Kurc, and P. Maćkowiak. 2010. "Football Player Detection in Video Broadcast." *Computer Vision and Graphics, Lecture Notes in Computer Science* 6375: 118–25.

Makandar, A., and D. Mulimani. 2018. "Analysis of Multiple Object Detection Using Kalman Filter in Sports Video." In *IJSA Proceedings on National Conference on Computer Science and Information Technology*, 13–5.

Maksai, A., X. Wang, and P. Fua. 2015. "What Players Do with the Ball: A Physically Constrained Interaction Modeling." In *IEEE Conference on Computer Vision and Pattern Recognition*, 972–81.

Manafifard, M., H. Ebadi, and H. M. Abrishami. 2017a. "Appearance-based Multiple Hypothesis Tracking: Application to Soccer Broadcast Videos Analysis." *Signal Processing: Image Communication* 55: 157–70.

Manafifard, M., H. Ebadi, and H. M. Abrishami. 2017b. "A Survey on Player Tracking in Soccer Videos." *Computer Vision and Image Understanding* 159: 19–46.

Markoski, B., Z. Ivankovic, L. Ratgeber, P. Predrag, and D. Glusac. 2015. "Application of Adaboost Algorithm in Basketball Player Detection." *Acta Polytechnica Hungarica* 12: 189–207.

Mathes, T., and J. H. Piater. 2006. "Robust Non-rigid Object Tracking Using Point Distribution Manifolds." In *Pattern Recognition, DAGM, Lecture Notes in Computer Science*, Vol. 4174, 1–10.

Mazzeo, P.-L., P. Spagnolo, M. Leo, and T. D'Orazio. 2008. "Visual Players Detection and Tracking in Soccer Matches." In *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 326–33.

Ming, Y., C. Guodong, and Q. Lichao. 2009. "Player Detection Algorithm Based on Gaussian Mixture Models Background Modeling." In *Second International Conference on Intelligent Networks and Intelligent Systems*, 323–6.

Mondal, D. C. 2014. "Multi Camera Soccer Player Tracking." PhD Thesis, Rourkela, India, National Institute of Technology.

Naidoo, W. C., and J. R. Tapamo. 2006. "Soccer Video Analysis by Ball, Player and Referee Tracking." In *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT) on IT Research in Developing Countries*, 51–60.

Naushad Ali, M., M. Abdullah-Al-Wadud, and S.-L. Lee. 2012. "An Efficient Algorithm for Detection of Soccer Ball and Players." In *Signal Processing Image Processing and Pattern Recognition*. Jeju Island: Springer.

Needham, C., and R. D. Boyle. 2001. "Tracking Multiple Sports Players through Occlusion, Congestion and Scale." In *Proceedings of the British Machine Vision Conference*. Manchester: British Machine Vision Association.

Pallavi, V., J. Mukherjee, A. K. Majumdar, and S. Sural. 2008. "Graph-based Multiplayer Detection and Tracking in Broadcast

Soccer Videos.'' *IEEE Transactions on Multimedia* 10 (5): 794–805.
Patil, P., R. Salve, K. Pawar, and M. P. Atre. 2018. ''Offside Detection in the Game of Football Using Contour Mapping.'' *International Journal of Research in Engineering and Science (IJRES)* 6 (4): 66–9.
Petsas, P., and P. Kaimakis. 2016. ''Soccer Player Tracking Using Particle Filters.'' In *IEEE International Symposium on Signal Processing and Information Technology*, 57–62.
Rao, U., and U. C. Pati. 2015. ''A Novel Algorithm for Detection of Soccer Ball and Player.'' In *International Conference on Communications and Signal Processing*.
Reddy, K. R., K. H. Priya, and N. Neelima. 2015. ''Object Detection and Tracking – A Survey.'' In *International Conference on Computational Intelligence and Communication Networks (CICN)*, 418–21.
Ren, J., J. Orwell, G. A. Jones, and M. Xu. 2008. ''Real-time Modeling of 3-d Soccer Ball Trajectories from Multiple Fixed Cameras.'' *IEEE Transactions on Circuits and Systems for Video Technology* 18 (3): 350–62.
Ren, J., J. Orwell, G. A. Jones, and M. Xu. 2009. ''Tracking the Soccer Ball Using Multiple Fixed Cameras.'' *Computer Vision and Image Understanding* 113 (5): 633–42.
Rodriguez-Canosa, G. R., S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald. 2012. ''A Real-Time Method to Detect and Track Moving Objects (DATMO) from Unmanned Aerial Vehicles (UAVS) Ssing a Single Camera.'' *Remote Sensing* 4 (4): 1090–111.
Sabirin, H., H. Sankoh, and S. Naito. 2015. ''Automatic Soccer Player Tracking in Single Camera with Robust Occlusion Handling Using Attribute Matching.'' *IEICE Transactions* 98-D: 1580–8.
Septier, F., J. Cornebise, S. J. Godsill, and Y. Delignon. 2011. ''A Comparative Study of Montecarlo Methods for Multitarget Tracking.'' In *IEEE Statistical Signal Processing Workshop*, 205–8.
Wu, L. 2008. ''Multi-view Hockey Tracking with Trajectory Smoothing and Camera Selection.'' PhD Thesis, The University of British Columbia.
Xing, J., H. Ai, L. Liu, and S. Lao. 2011. ''Multiple Player Tracking in Sports Video: A Dual-Mode Two-Way Bayesian Inference Approach with Progressive Observation Modeling.'' *IEEE Transactions on Image Processing* 20 (6): 1652–67.
Xu, M., J. Orwell, and G. Jones. 2004. ''Tracking Football Players with Multiple Cameras.'' In *International Conference on Image Processing (ICIP)*, Vol. 5, 2909–12.
Yang, Y., and D. Li. 2017. ''Robust Player Detection and Tracking in Broadcast Soccer Video Based on Enhanced Particle Filter.'' *Journal of Visual Communication and Image Representation* 46: 81–94.
Yazdi, M., and T. Bouwmans. 2018. ''New Trends on Moving Object Detection in Video Images Captured by a Moving Camera: A Survey.'' *Computer Science Review* 28: 157–77.
Yilmaz, A., O. Javed, and M. Shah. 2006. ''Object Tracking: A Survey.'' *ACM Computing Surveys (CSUR)* 38: 45.
Yoon, Y., H. Hwang, Y. Choi, M. Joo, and H. Oh. 2019. ''Analyzing Basketball Movements and Pass Relationships Using Realtime Object Tracking Techniques Based on Deep Learning.'' *IEEE Access* 7: 56564–76.
Zhu, G., C. Xu, Q. Huang, and W. Gao. 2006. ''Automatic Multi-Player Detection and Tracking in Broadcast Sports Video Using Support Vector Machine and Particle Filter.'' In *IEEE International Conference on Multimedia and Expo*, 1629–32.

Supplementary Material: The online version of this article offers supplementary material (https://doi.org/10.1515/jqas-2020-0088).