Machine Vision and Applications (2018) 29:187–206
https://doi.org/10.1007/s00138-017-0893-8

ORIGINAL PAPER

Player detection in field sports

Cem Direkoglu[1] · Melike Sah[2] · Noel E. O'Connor[3]

Received: 21 December 2015 / Revised: 23 September 2017 / Accepted: 7 November 2017 / Published online: 5 December 2017

Abstract
We describe a method for player detection in field sports with a fixed camera setup based on a new player feature extraction strategy. The proposed method detects players in static images with a sliding window technique. First, we compute a binary edge image and then the detector window is shifted over the edge regions. Given a set of binary edges in a sliding window, we introduce and solve a particular diffusion equation to generate a shape information image. The proposed diffusion to generate a shape information image is the key stage and the main theoretical contribution in our new algorithm. It removes the appearance variations of an object while preserving the shape information. It also enables the use of polar and Fourier transforms in the next stage to achieve scale- and rotation-invariant feature extraction. A support vector machine classifier is used to assign either player or non-player class inside a detector window. We evaluate our approach on three different field hockey datasets. In general, results show that the proposed feature extraction is effective and performs competitive results compared to the state-of-the-art methods.

Keywords Feature extraction · Heat diffusion · Player detection · Field sports

1 Introduction

Sport video analysis is an important and active topic in computer vision. In particular, many works focus on field sports such as soccer, American football and field hockey, which are very popular outdoor sports around the world. There are many possible applications of analyzing field sport videos such as event detection and player/team activity analysis. These high-level applications require low-level structural procedures, specifically player detection, classification and tracking. Player detection is usually the fundamental step in sport video analysis. There are two possible sources of sport videos: TV broadcasts and fixed cameras around the playground. In this paper, we focus on player detection in field sports using a fixed camera infrastructure. However, for completeness, in the following we review player detection techniques based on the both sources.

✉ Cem Direkoglu
cemdir@metu.edu.tr

Melike Sah
melike.sah@neu.edu.tr

Noel E. O'Connor
Noel.Oconnor@dcu.ie

[1] Department of Electrical and Electronics Engineering, Middle East Technical University - Northern Cyprus Campus, Kalkanli, Guzelyurt, North Cyprus, via Mersin 10, Turkey

[2] Department of Computer Engineering, Near East University, Nicosia, North Cyprus, via Mersin 10, Turkey

[3] INSIGHT Centre, Department of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland

1.1 Using the TV broadcast

Field sports are played outdoors on a large playground which is an almost homogeneous region. Most player detection techniques assume the existence of a dominant color (e.g., a tone of green) on a field of play and use this characteristic to assist player detection algorithms. The dominant color feature has been used in TV broadcast videos for player detection [1–4]. Liu et al. [1] learn the dominant color by accumulating HSV color histograms in a broadcast video. Then, the dominant color is used to segment the playfield. According to the area of the segmented region, they classify view types, and player detection is performed in global (i.e., distance) view type, which is achieved by running a boosted cascade of Haar features [5] on non-playfield regions. Khatoonabadi and Rahmati [2] use RGB color histograms to determine the dominant color and detect the playground in broadcast videos. The field line markings are detected in a second step using the Hough transform. Finally, some restric-

tions such as area and ratio of major length to minor length are applied to the remaining regions to detect players. Beetz et al. [3] model color classes on the playground (i.e., the field is green and the lines are white) using a mixture of Gaussians in RGB space and use this model to segment the playfield regions. Next, they use special templates to detect players based on color distributions, compactness and vertical spacing of the remaining regions. A comprehensive survey on player detection using the TV broadcast is given in [4].

Using broadcast cameras, however, cannot allow us to address some specific tasks such as team activity and strategy analysis, evaluation of player performances, 2D/3D reconstructions and visualizations of player actions. This is because the broadcast camera usually only captures a specific region (such as ball locations) and many players may not be in that region. Using broadcast cameras also suffers from inaccurate player detection because of camera motions, occlusions, etc.

1.2 Using fixed cameras

Fixed multi-camera systems usually cover all locations on the field of play and therefore capture all players simultaneously. Background subtraction is a common method for player detection with a fixed camera infrastructure [6–10]. To consider problems in outdoor scenes such as changes of illumination, shadows, background objects, these methods need to frequently update the background representation model. Some statistical adaptive methods [11–14] have been proposed, but these methods only work well for simple scenes with slow changes of illumination. These approaches can also easily incorporate objects that stop moving for a certain time into the background model. In field sport, it is common to have players (e.g., goalkeeper) that stand still for many video frames. Figueroa et al. [7] pointed out that applying a median filter along the pixels of some consecutive frames for background modeling can increase the tolerance to illumination changes and facilitate still player detection in comparison with statistical adaptive methods. Carr et al. [10] created shape-specific occupancy maps on the ground plane using the foreground regions after background subtraction for player detection. This approach increases the tolerance to shadows, but can only identify isolated individuals. Xu et al. [6] integrated the dominant color and geometry information of the field to assist background subtraction for player detection. Vandenbroucke et al. [15] proposed a player detection technique based on color image segmentation instead of using the temporal information. However, all of the methods based on background subtraction and image segmentation fail when a single segmented region contains multiple players or when a single player is segmented into multiple regions.

2 Our motivation and contribution

Player detection algorithms have to face challenging situations in field sports such as variability of lighting and weather conditions, geometric variations of the players in images such as scale and rotation depending on the camera view point. Players may appear at different scales, resolution and orientation depending on the distance to camera and direction of their movement. Player appearance is also strongly influenced by the team uniform and illumination, since there is a wide range of player uniform colors and textures. We propose to address these problems and introduce an approach for player detection within a fixed camera infrastructure in field sports. We evaluate our approach on field hockey, where the top view playground and the camera configuration is shown in Fig. 1a. A sample frame from one of the camera views is also shown in Fig. 1b. We constrain the pose to standing, walking, running and bending. A player corresponds to any human on the playground including both team players and referee.

The proposed approach is based on a sliding window technique on an individual image. Given a video frame, we compute a binary edge image. There may be edges detected outside the playground because of audience and advertisements. These edges are removed by a geometry-based playground mask to restrict further processing and accelerate detection speed. Since the playground is almost homogeneous in field sports, the remaining edges belong to the field markings, players and noise on the playground. The detector window is then scanned across the edge regions. The window dimensions are determined based on known camera geometry and prior information of the target object class.

Given a set of binary edges in a sliding window, we introduce and solve a particular diffusion equation to generate a shape information image. The proposed diffusion to generate a shape information image, inside the detector window, is the main theoretical contribution and the key stage in our new algorithm. Despite the missing edges of an object because of low resolution or noise, the proposed diffusion can fill inside the object's shape while preserving the shape information. It removes the appearance variations (i.e., color and texture) of an object. It also enables to use polar and Fourier transforms in the next stage to achieve scale- and rotation-invariant feature extraction. The heat diffusion analogy has been deployed before in various ways in image processing and computer vision. It has been used for: image smoothing and enhancement [16], region-based image segmentation [17], skeletonization [18], multi-scale shape description [19] and motion analysis [20,21]. However, this is the first time a particular heat diffusion equation is used for estimating shape over the binary edge maps.

After the proposed features are extracted, a support vector machine classifier is used to label either player or non-player

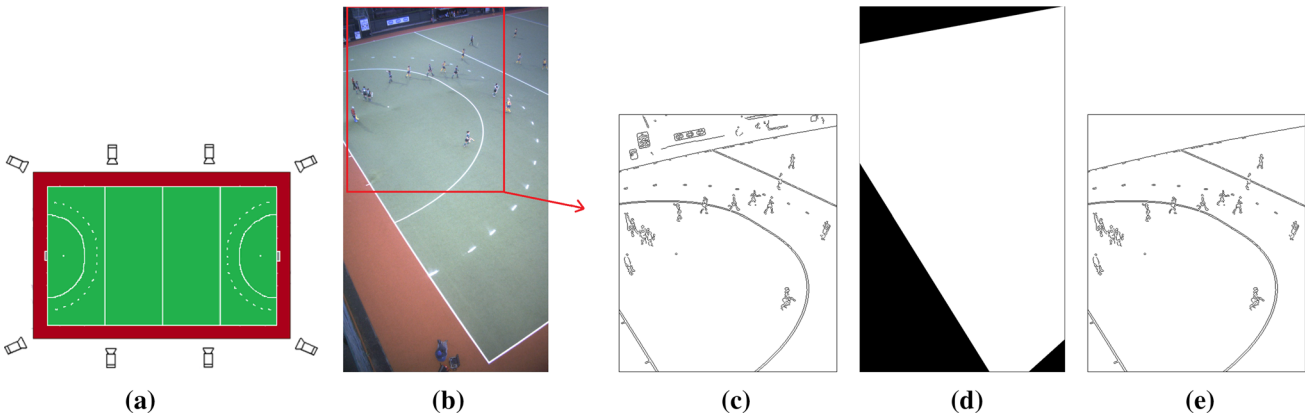


Fig. 1 **a** Top view of the playground with camera locations. **b** A sample frame of dimensions 959 × 539 from a camera view. **c** Binary edge image from the red box in sample image. **d** The geometric mask of the playground. **e** Binary edge image from the red box after masking operation (color figure online)

class at each window location. We evaluate our approach on a field hockey dataset on different camera views. The results show that our approach is effective, and in general it performs better than the state-of-the-art techniques for player detection.

3 Region of interest selection

The search region is estimated based on edge features derived from the image data and known playfield geometry. The first step in our approach is binary edge detection using the Canny method [22]. Canny edge detection is perhaps the most popular edge detection technique at present. The first requirement is to reduce the response to noise with Gaussian filtering. Then, a finite difference edge finder is applied to compute the gradient magnitude. Then, non-maxima suppression (peak detection) is applied to the gradient magnitude image that retains only those points at the top of the ridge, while suppressing others. Finally, hysteresis thresholding is used, which involves two thresholds, to obtain binary edges. In our experiments, the standard deviation of the Gaussian filter is 0.4. The finite difference edge finder is the Sobel operator. The thresholds to obtain the binary image are determined automatically. In this process, the non-maxima suppressed image is thresholded by the scaled median value of the gradient magnitude image. The upper and the lower thresholds are determined as $T_H = c \times \text{median}(G)$ and $T_L = T_H/2$, where median(G) is the median value in the gradient magnitude image G and c is a scale factor for threshold selection which is a positive constant. Higher values of c cause higher values for the thresholds. However, the threshold values must be smaller than the maximum intensity values of the image which we are thresholding. In this evaluation, the optimum value for c is 9 determined experimentally. The ratio between high, T_H, and low, T_L, thresholds is 2.

Since the playground is homogeneous in field sports, the edges mostly belong to the field markings, players and noise on the playground as shown in Fig. 1c. There are also edges detected outside the playground because of audience and advertisements. These edges are removed by a geometry-based playground mask. The geometry-based mask has been similarly used by [6] to assist player segmentation. The geometry-based mask is obtained by using homography transformations from the image plane to the top view ground plane (i.e., 2D to 2D plane transform). Suppose that H is the transformation matrix from the image plane to ground plane and C is the coordinate range of the ground plane. If an image point, $\mathbf{x} = (x, y)$, is in the ground plane coordinate after the transformation, it is one, otherwise it is zero. The geometry-based mask can be represented as follows $M = \{(x, y)|H(x, y) \in C\}$. The binary geometry-based mask image and the binary edge image after the masking operation are shown in Fig. 1d, e. To further accelerate the detection, we process only the window regions which include a significant number of edge points. If the total number of edge points, inside a window, is higher than a pre-defined threshold (i.e., $T = 10$ in our experiments), we employ feature extraction and classification.

4 Window aspect ratio and dimensions

The window aspect ratio and dimensions are determined based on prior information of the target object class and known camera geometry, respectively. The window aspect ratio (height divided by width) is 1.6 which is estimated experimentally based on annotated player regions to cover poses such as standing, walking, running and bending. The window dimensions are determined during the scanning process using the camera geometry. Each scanning point is assumed to be the bottom middle point of the window, and

this point is projected from the image coordinates to world coordinates onto the ground plane. This is a 2D to 3D inverse perspective transformation with height equal to zero meters. Then, we make the height 1.8 m in the world coordinate system, assuming that players are 1.8 m tall, and project back to the image coordinates (i.e., 3D to 2D direct perspective projection). The projected point is the top point of the window. We can compute the window height in pixels using the top and bottom points of the window. The width can be calculated using the aspect ratio (i.e., 1.6). However, instead of computing the width at each scanning, we select one of the pre-defined window dimensions depending on the height of the window. These pre-defined window dimensions are estimated using the annotated player regions. If the height is less than 40 pixels, the window dimensions are 40×25. If the height is between 40 and 48 pixels, the window dimensions are 48×30. If the height is between 48 and 56 pixels, the window dimensions are 56×35, and finally if the height is between 56 and 64 pixels, the window dimensions are 64×40.

5 Shape information image generation using a heat equation

Here, we introduce the key stage and the main theoretical contribution in our algorithm. In each detector window, there can be missing or disconnected edges of an object due to low resolution, noise, etc. If there is a player in the window, it means there are also edges because of the team uniform texture and style. Edge detection is a low-level feature extraction, and it does not give any object shape information. We address these problems by solving a particular heat diffusion equation in the window region. The proposed diffusion generates a shape information image of an object. The heat diffusion analogy has been used before in image processing and computer vision such as for image smoothing and enhancement [16], region-based image segmentation [17], multi-scale scape description [19], skeletonization [18] and motion analysis [20,21]. However, this is the first time a particular heat diffusion equation is used for shape estimation over the binary edge maps. First, we explain the basic concept of heat diffusion and then describe the proposed diffusion problem to generate a shape information image.

5.1 Basic concepts of heat diffusion

Conduction or diffusion is the flow of heat energy from high- to low-temperature regions due to the presence of a thermal gradient in a body [23]. The change of temperature over time at each point of a two-dimensional material is described using the general heat diffusion equation,

$$\frac{\partial T(\mathbf{x},t)}{\partial t} = \alpha \nabla^2 T(\mathbf{x},t) = \alpha \left(\frac{\partial^2 T(\mathbf{x},t)}{\partial^2 x} + \frac{\partial^2 T(\mathbf{x},t)}{\partial^2 y} \right), \tag{1}$$

where $\partial T(\mathbf{x},t)/\partial t$ is the rate of change of temperature and $(\mathbf{x},t) = (x,y,t)$ is space and time vector, ∇^2 is the spatial Laplacian operator for the temperature, α is called thermal diffusion coefficient of the material and a larger value of α indicates faster heat diffusion through the material. The solution of this equation provides the temperature distribution over the material body, and it depends on time, distance, properties of material, as well as specified initial and boundary conditions.

Initial conditions specify the temperature distribution in a body, as a function of space coordinates, at the origin of the time coordinate ($t = 0$). Initial conditions are represented as follows,

$$T(\mathbf{x}, t = 0) = F(\mathbf{x}), \tag{2}$$

where $F(\mathbf{x})$ is the function that specifies the initial temperature inside the body. Boundary conditions specify the temperature or the heat flow at the boundaries of the body. There are three general types of boundary conditions: Dirichlet, Neuman and Robin. Here, we explain the Dirichlet conditions, which is used in our algorithm. In the Dirichlet condition, temperature is specified along the boundary layer. It can be a function of space and time, or constant. The Dirichlet condition is represented as follows,

$$T(\mathbf{x},t) = \Phi(\mathbf{x}), \tag{3}$$

where $\Phi(\mathbf{x})$ is the function that specifies the temperature at the boundary layer. A tutorial on heat diffusion theory is also given in [24,25].

5.2 Proposed heat diffusion problem and solution

Given a set of binary edges in a sliding window, we propose and solve a heat diffusion equation. The solution of the proposed equation fills inside the object shape while preserving the shape information. Therefore, it removes the appearance variations (i.e., color and texture) of an object. The proposed equation is given below,

$$\begin{aligned} &\tfrac{\partial I(\mathbf{x},t)}{\partial t} = E(\mathbf{x}) \nabla^2 I(\mathbf{x},t) \\ &\text{with} \quad \begin{cases} I(\mathbf{x}, t=0) = 1 - E(\mathbf{x}), & \text{initial condition} \\ I(\mathbf{x},t) = 0, & \text{boundary condition} \end{cases} \end{aligned} \tag{4}$$

where E is a binary edge image of a space vector $\mathbf{x} = (x,y)$ and in diffusion theory it is known as the diffusion coefficient. In this equation, the diffusion coefficient $E(\mathbf{x})$ is space variant (i.e., non-uniform) where the edge positions are zero and

the rest of the positions are one. I is a solution that is a real-valued function of a space and time vector $(\mathbf{x}, t) = (x, y, t)$. The solution, I, depends on the diffusion coefficient, as well as the initial and boundary conditions over a bounded region of interest. The initial condition is a binary image where the edge positions are one and the rest of the positions are zero $(1 - E(\mathbf{x}))$. The boundary condition is Dirichlet which has a specific solution, $I(\mathbf{x}, t) = 0$, at the boundaries of the window. The proposed diffusion problem has a steady-state solution since it is a linear and homogeneous diffusion equation [23] with a space-variant diffusion coefficient. In this work, the numerical solution is obtained using a multigrid solver [26] since it is computationally more efficient than iterative methods. Figure 2a–d shows shape information images generated for the given samples, where the top five samples represent players and the bottom five samples represent background (non-players). The solution of the proposed diffusion enables the use of polar and Fourier transforms in the next stage to achieve scale- and rotation-invariant feature extraction.

In Fig. 3, we compare the proposed diffusion with the morphological operation for the object shape estimation using the binary edges. In morphological operation, first the closing operation (i.e., dilation and then erosion) is applied to the binary edge map using a pre-defined structuring element, and then we fill the small regions inside the object. To visual inspection, it is seen that the proposed diffusion can estimate the player shapes better than the morphological operation. The morphological operation is more sensitive to missing edges in comparison with the proposed diffusion. In morphological operation, if the size of the structuring element is small, it cannot handle missing edges (i.e., the missing parts of the boundary) well and fails to estimate object shape. On the other hand, if the size of the structuring element is large, it can disturb and remove the important curvatures of the shape, and again it may fail to estimate the object shape.

Figure 4 also shows the behavior of the proposed diffusion in case of presence of the field lines edge pixels in the background (i.e., background clutter). In this case, we may observe failures in player shape estimation.

6 Scale- and rotation-invariant feature extraction

As we described in Sect. 4, the detector window's dimensions change depending on the player location on the playground. The orientation of a player may vary depending on the direction of movement, as well as the camera view point. For example, in Fig. 2a, b, although they are upright their orientation is different. Players' scales may also differ at each detector window even if the window's dimensions are the same. To overcome these problems, the coordinates of each window image are polar mapped [27,28] onto an image of fixed dimensions, i.e., 32×32. In the polar-mapped image, rotations appear as translations and image dimensions are the same for all samples. Consider the polar coordinate system (r, θ), where $r \in \Re$ denotes radial distance from the center of the window image (x_c, y_c) and $0 \leq \theta \leq 2\pi$ denotes angle. Any point $(x, y) \in \Re^2$ can be represented in polar coordinates as follows,

$$\begin{aligned} r &= \sqrt{(x - x_c)^2 - (y - y_c)^2} \\ \theta &= \tan^{-1}\left(\tfrac{y - y_c}{x - x_c}\right). \end{aligned} \tag{5}$$

There are two principal methods for mapping a rectangular image to a circle in the polar transform. The image can either be fitted within the circle or the circle can be fitted within the boundaries of the image. The main problem with fitting the circle within the boundaries of the image is losing the information in the corners. Since we want to use all information in the window image, we use the method that fits the image within a circle. In this method, all pixels will be taken into account, but some invalid pixels will also be included, which fall inside the circle but outside the image. In our algorithm, these invalid pixel values are set to zero. Figure 2e shows the polar transform of the shape information image for each sample. For better visualization, Fig. 2f shows the color-mapped polar transforms.

Then 2D Fourier transform is applied to the polar-mapped image, as given below, to compute the Fourier magnitude, which removes these translations.

$$F(k, l) = \frac{1}{MN} \sum_{r=0}^{M-1} \sum_{\theta=0}^{N-1} P(r, \theta) e^{[-j2\pi(kr/M + l\theta/N)]}, \tag{6}$$

where $F(k, l)$ is the Fourier transform of the polar-mapped image $P(r, \theta)$ of size M×N. The resultant Fourier magnitude image, $|F(k, l)|$, is translation invariant which means that it is player rotation invariant. Applying the Fourier transform over polar-mapped image to achieve rotation invariance is not a novel approach. First, it has been introduced as a part of the Fourier–Mellin transform algorithm [29] that performs rotation-, size- and translation-invariant image feature extraction in 2D space. Later, it has been utilized by the well-known region-based shape description techniques [27,28]. These techniques apply polar and Fourier transforms to the binary images of the objects to achieve the rotation invariance; on the other hand, we apply these transforms to the solution of the proposed heat diffusion equation. Figure 2g shows the Fourier transform magnitude images for each polar transform sample.

To achieve scale invariance of the object, all of the Fourier magnitude values are divided by $|F(0, 0)|$, the DC value of the image that corresponds to the average brightness. In our

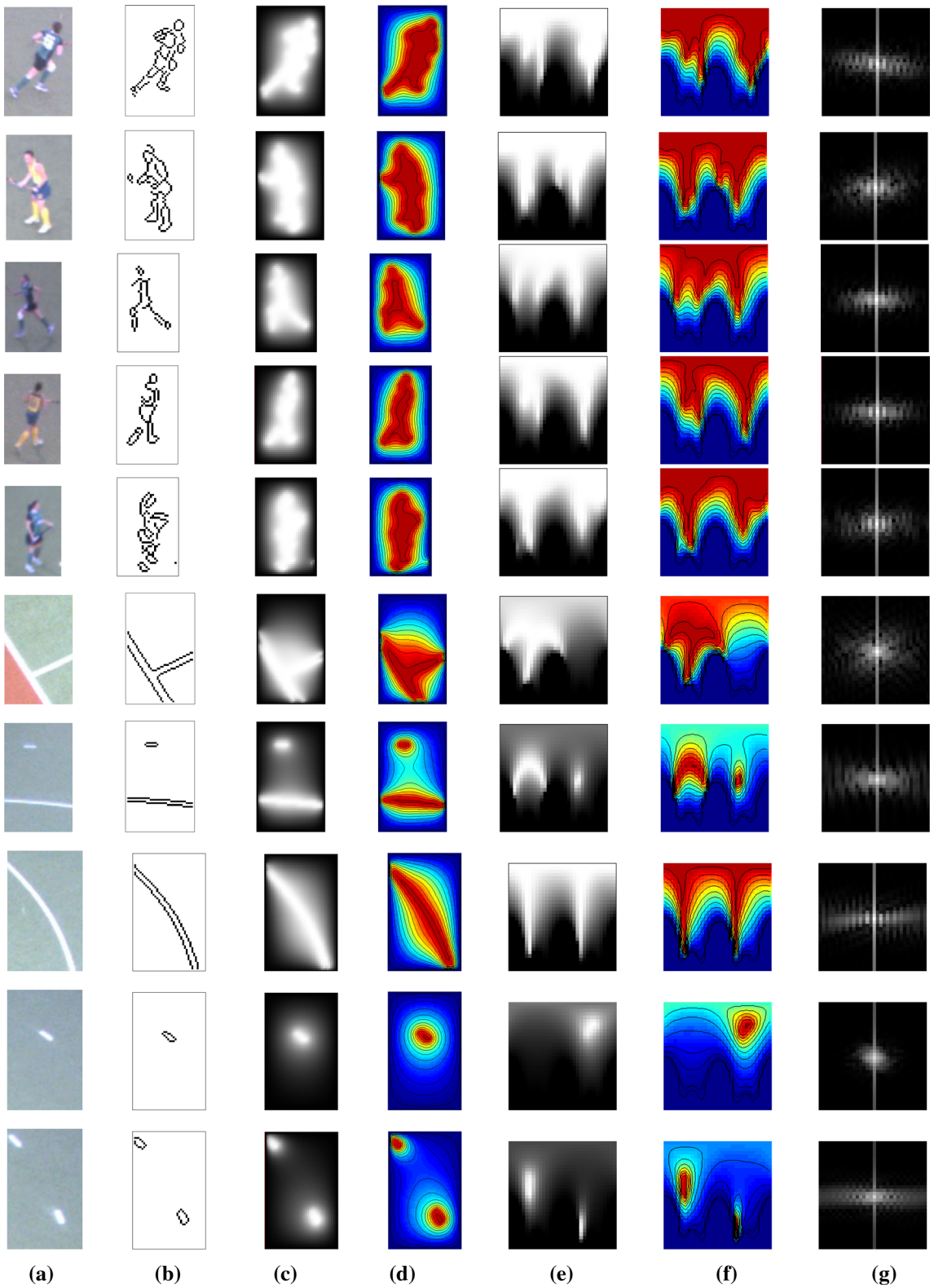


Fig. 2 **a** Top five samples are for players and the bottom five samples are for non-players. **b** Binary edges. **c** Shape information image. **d** The color-mapped shape information image. **d** The polar transform image. **e** The color-mapped polar transform image. **f** The Fourier magnitude image

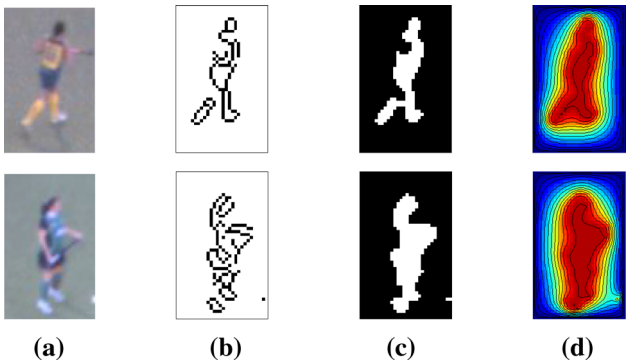


Fig. 3 Comparison of morphological operation and the proposed diffusion for shape estimation. **a** Samples, **b** binary edges, **c** binary object image after morphological operations, **d** shape information image after the proposed diffusion

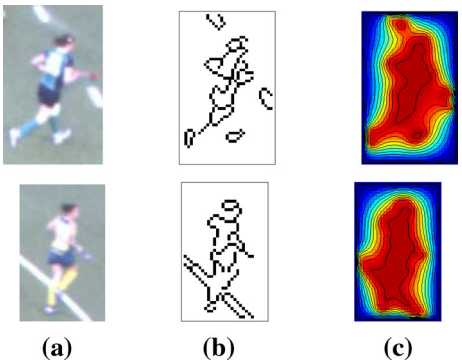


Fig. 4 Behavior of the proposed diffusion in the presence of field lines in the background. **a** Samples, **b** binary edges, **c** shape information image after the proposed diffusion

implementations, the Fourier magnitude image is shifted in a way that the DC value is displayed in the center of the image. Distance from this center point represents increasing frequency. The lower frequency components of the Fourier descriptor capture the general shape properties of the object, and the higher frequency components capture the finer detail. For efficient shape description, only a small number of the descriptors should be selected for shape representation. In our work, a shape information image and its polar transform are a smooth distribution, and most of the shape information is contained in the low-frequency components. To select the lower frequency components as descriptors, we draw a circle around the center point (i.e., DC value point) with a predefined radius and choose all of the descriptors within the circle, except the descriptor in the center point, to represent shape. We form a one-dimensional vector with these features; in our experiments, the radius of the circle is 5 which results in 100 features for shape representation. It is important to note that we choose Fourier-based shape description because it is proven that Fourier descriptors are easy to compute and robust in 2D shape classification [27,28].

7 Classification using the shape features

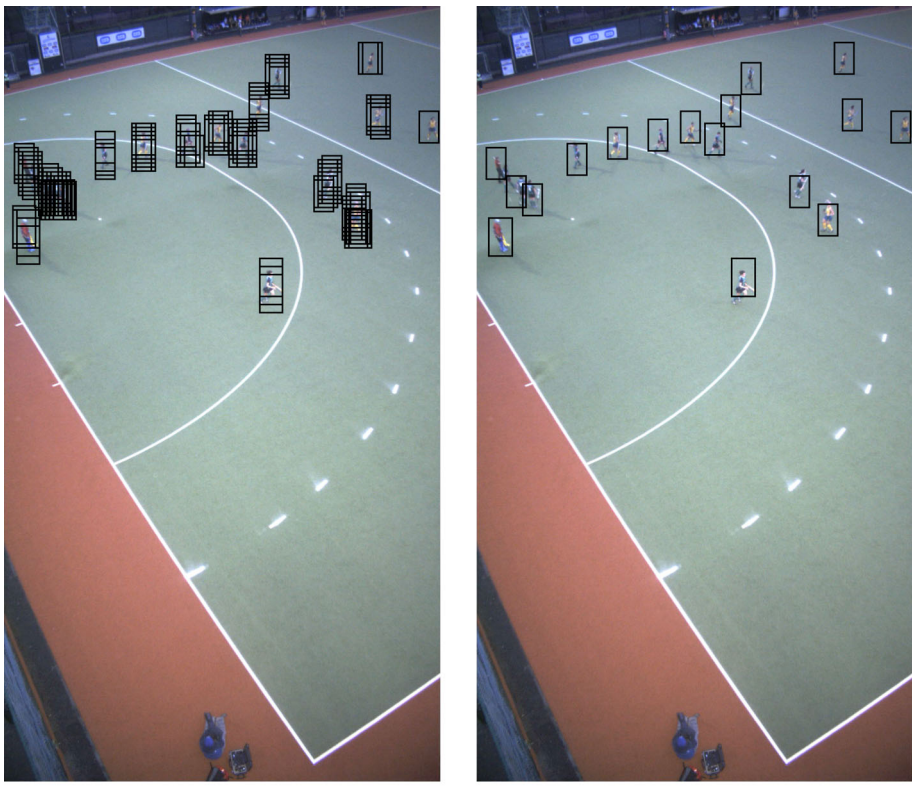
A support vector machine (SVM) with a Gaussian radial basis function kernel is used to label either a player or non-player in each detector window. Our experiments show that the proposed features achieve better results with the Gaussian kernel in comparison with other possible kernel functions in SVM. The scaling factor of the Gaussian kernel is 2.1. The upper bound on the Lagrange parameters is 5. These parameter values are selected using the cross-validation on the training set. In addition, we use the sequential minimal optimization method to find the separating hyperplane since we have a large training set and this method is computationally efficient. Our detection system takes an image and returns a set of bounding boxes (BB) and a confidence value for each detection. Then, non-maximal suppression is applied for merging nearby detections, using the confidence values, to determine the final detections. In our method, the confidence value is the SVM decision value. The non-maxima suppression is a pairwise max (PM) suppression [30] which greedily selects high-scoring detections and discards detections that significantly overlap with a previously selected detection. The overlap is measured as follows:

$$\Gamma_{ij} = \frac{\text{area}(\text{BB}_i \cap \text{BB}_j)}{\text{area}(\text{BB}_i \cup \text{BB}_j)}, \tag{7}$$

where Γ_{ij} is the overlap measure between two different bounding boxes BB_i and BB_j. In our experiments, non-maximal suppression is applied if $\Gamma_{ij} > 0.2$ (more than 20% overlap). Figure 5a, b shows detections before and after the non-maxima suppression, respectively, with the proposed method.

8 Evaluation and results

The proposed approach is validated on a field hockey dataset. There are eight fixed cameras around the field in order to cover the entire playground, and each camera is mounted on a pole 20 m high. In this paper, we present results for three different camera views, two of them (Camera 1 and 2) are corner view cameras and one of them (Camera 3) is a side view camera. Figures 5, 6 and 7 show example frames, respectively, for Camera 1, 2 and 3. The dimensions of the frames are 959×539. For training, we collect 1375 player samples from different camera views with variation of appearance, scale, rotation and pose. The non-player samples are different for each camera view since each camera view has a different background image (i.e., the image with no players in the scene). The edge regions are scanned after geometry-based masking to extract and collect non-player features. As a result, Camera 1, 2 and 3 have, respectively,



(a) **(b)**

Fig. 5 Detections **a** before and **b** after non-maximal suppression

13,420, 12,514 and 3111 non-player samples for training. For testing, we prepare a dataset for each camera view by manually labeling the ground truth BBs. There are 4526, 4780 and 2407 players labeled in Camera 1, Camera 2 and Camera 3 datasets, respectively, in 301 consecutive frames for each view. In total, 11,713 player BB locations are manually labeled from three different camera views for evaluation.

We evaluate our approach while comparing with nine different methods: A background subtraction (BS) method [7], the histogram of oriented gradients (HOG) features [31] describing the human body shape, the deformable part-based model (DPM) [30] that also use the HOG features to describe human body shape with a part-based approach, using a pre-trained convolutional neural network (CNN) (AlexNet) as a feature extractor [32], with the selective search method that is used for detecting objects using hierarchical grouping and SVM [33], the PSHOG model which combines the proposed shape proposal with the HOG features. In addition to these methods, we also perform comparison with the different shape representations. For example, in our algorithm, instead of using the proposed shape proposal we use binary foreground mask of an object appear after morphological operations to the binary edge image (abbreviated with MORPH on the graphs and tables). This representation is shown in Fig. 3c. We also compare with the smoothed version of the binary foreground mask. A Gaussian filter is used for smoothing, and this method is abbreviated with Gauss in the evaluations. In addition, we compare with the shape that appears after passing the binary foreground mask through sigmoid function to get values between 0 and 1. This method is abbreviated with SIGD in evaluations.

The BS method [7] is a commonly used method for player detection with a fixed camera. This method, in our evaluation, extracts the background image by applying a median filter along the pixels of 70 consecutive frames. Then, the difference between the current and the background image is computed, and a threshold is applied to the difference image for binarization. The threshold value to binarize the difference image is 13. The next step is morphological filtering (i.e., opening and closing) to eliminate noise and connected pixels labeling to define players regions. The parameter values in the BS method [7] are determined experimentally using an additional validation set.

The HOG + SVM [31] is combined with the sliding window technique in an individual image for player detection. The region of interest, the window aspect ratio and dimensions are determined as described in Sects. 3 and 4. The estimated window dimensions are normalized to 56×35. Then, we compute the HOG features in this region. In our experiments, the number of orientation bins is 7, the cell size is 10×10 pixels, the block size is 2×2 cells, the stride of the blocks is 10 pixels, and the L2 norm is used to normalize

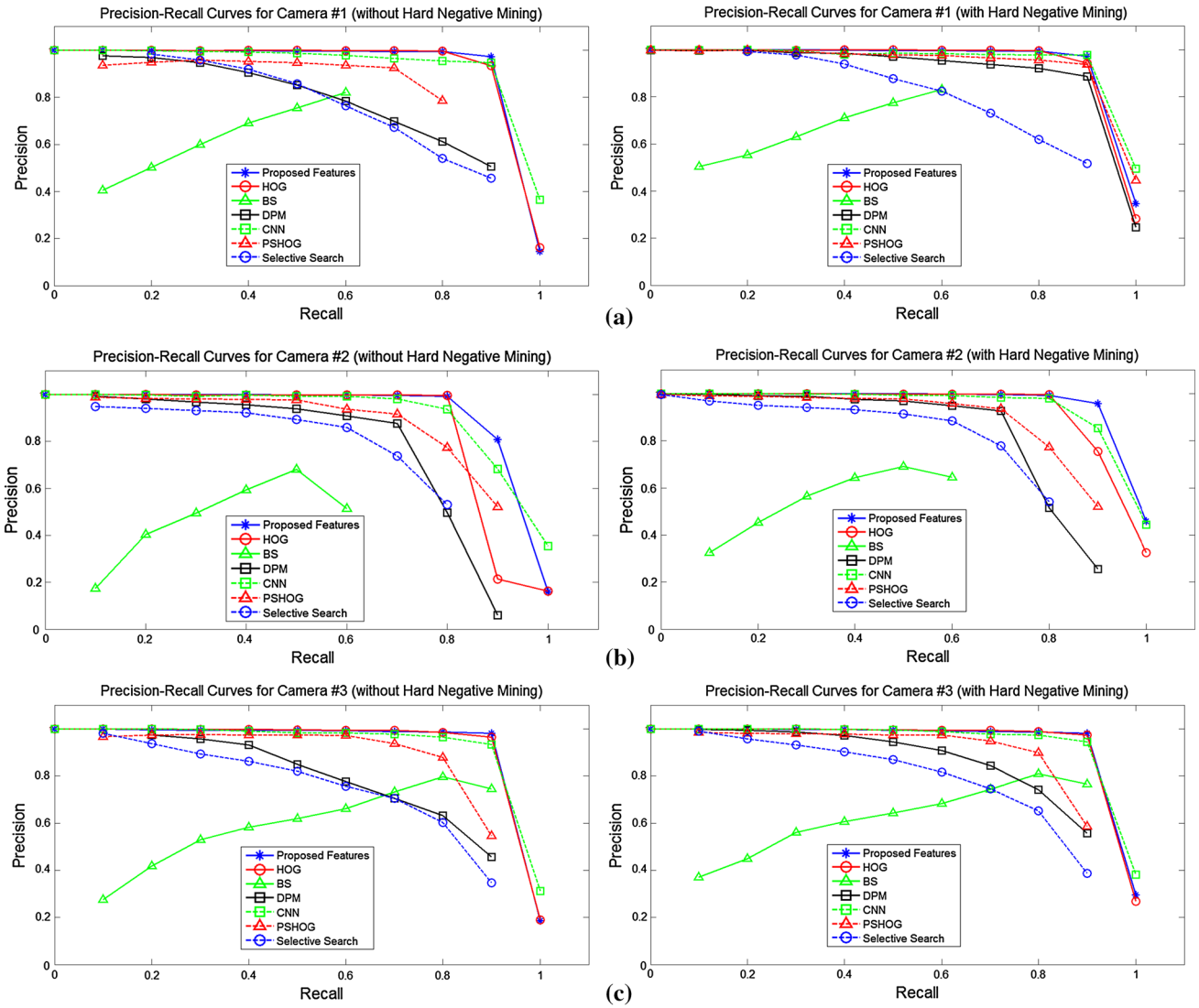


Fig. 6 Precision–recall curves comparison with other techniques (left—without hard negative mining, right—with hard negative mining): **a** Camera #1 dataset, **b** Camera #2 dataset and **c** Camera #3 dataset

contrast for each block. The feature vectors for all blocks are concatenated to yield a final feature vector, and the dimension of the final feature vector is 420. In SVM, linear kernel function (i.e., dot product) is used to map training data into kernel space. Our experiments show that, in our datasets, the HOG features achieve the best results with the linear kernel in comparison with other possible kernel functions. The upper bound on the Lagrange parameters, in linear SVM, is 0.15. The SVM parameter value is determined using the cross-validation on the training set.

The DPM + LSVM [30] is also combined with the sliding window technique in an individual image for player detection. In DPM, the person model is defined by filters such as the root filter (i.e., whole body filter) and part filters (i.e., head filter, right shoulder filter). These filters score sub-windows of a feature pyramid for person detection, where the feature pyramid is computed by computing the image pyramid. The number of levels in the pyramid is 5. The pyramid approach also makes this model scale invariant. Responses from root filter and part filters are computed at different levels in the pyramid to increase the performance as discussed in [30]. They use HOG features, but the lower dimensional ones that are obtained after principal component analysis (PCA). The dimension of the HOG features representing this model is 36, with 9 orientations and 4 normalizations. We trained DPM filters with the same samples that we used in our approach and HOG. The Latent SVM is used for training and classification of the person. These parameter values are determined experimentally on a different validation set. We use the original MATLAB codes implemented by authors [34] for comparison.

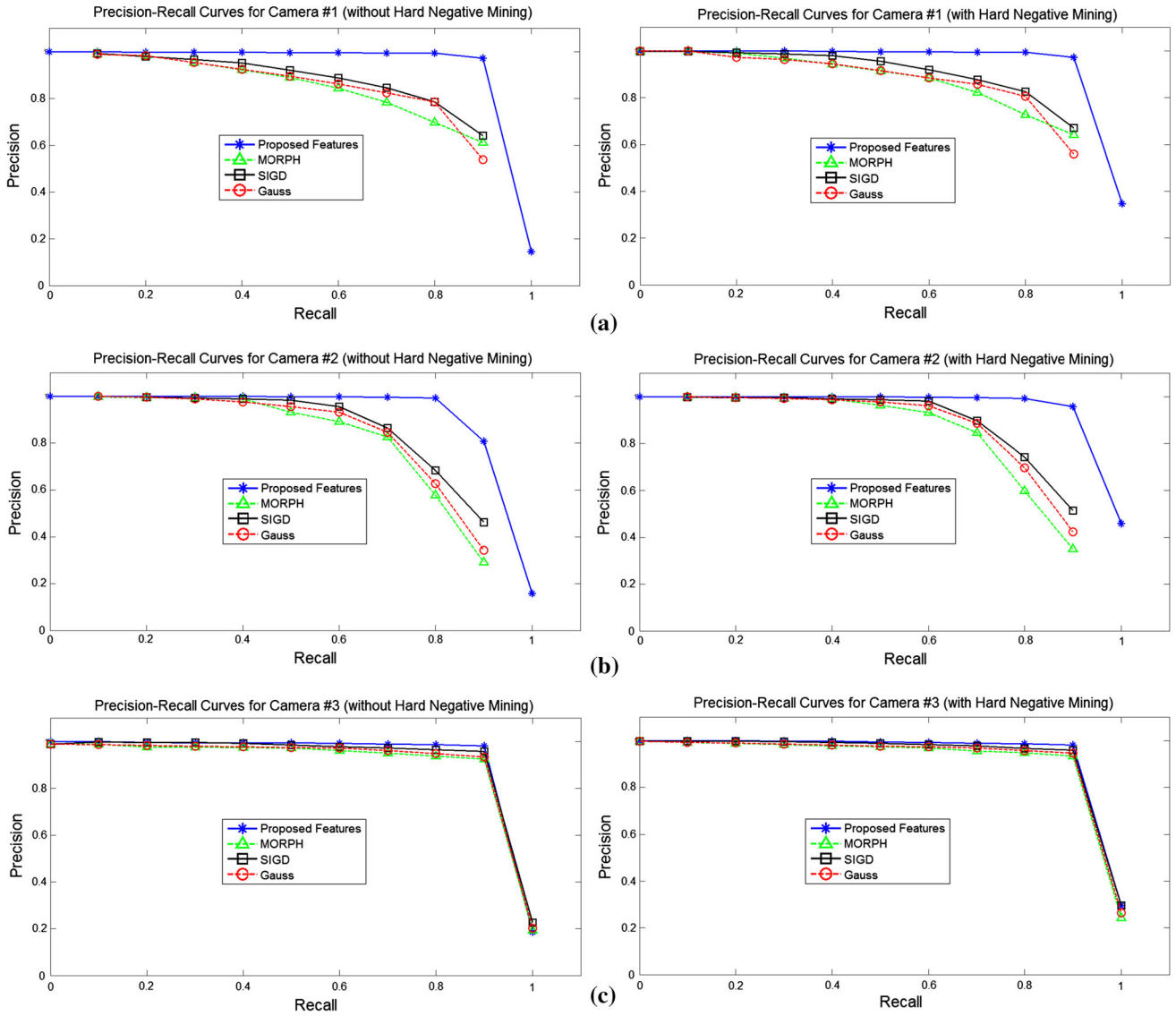


Fig. 7 Precision–recall curves comparison with the other shape proposals (left—without hard negative mining, right— with hard negative mining): **a** Camera #1 dataset, **b** Camera #2 dataset and **c** Camera #3 dataset

Convolutional neural network (CNN) is a type of feed-forward artificial neural network. Nowadays, CNN is the state-of-the-art tool for image classification. Therefore, we compare the proposed method with the CNN method. CNNs are trained using large image collections of diverse images, and they learn rich image features from these collections. One of the major drawbacks of the CNN method is the long time needed to train deep networks. However, without investing time and effort into training, a pre-trained CNN can be utilized as a feature extractor, which we perform as a comparison with the proposed method. Instead of using the proposed method for feature extraction, we apply CNN as a feature extractor [35] using the Matlab instructions given in [32]. In particular, we keep the proposed system architecture the same, but use CNN features instead of diffusion features. We use AlexNet [36] pre-trained network as a feature extractor. The training data consist of 1375 player images from three different camera views with varying appearance, scale, rotation and pose. Non-player samples consist of 13420, 12514 and 3111 images for Camera 1, 2 and 3, respectively. For CNN training, first these images are re-sized to AlexNet image requirements (i.e., 227×227), since we fine-tune AlexNet [36] pre-trained network as a feature extractor. We extract features from the last layer of the CNN (i.e., fc7 layer of the AlexNet which is the last layer before classification). We use stochastic gradient descent (SGD) method for CNN training. We also use the suggested training parameters of AlexNet [36]. The CNN training parameters such as the maximum number of iterations, learning rate, step size, weight decay, momentum and gamma are set to 40,000, 0.001, 1000,

0.0005, 0.9 and 0.8, respectively. Finally, we train a linear SVM classifier using CNN features (i.e., the output of the last layer of CNN) instead of diffusion features for classification. For testing 11,713 players, BB locations are manually labeled from three camera views, as we explained in Sect. 8. Testing images are also re-sized to 227×227 according to AlexNet image requirements for classification.

In the selective search method [33], images are segmented to produce image regions. Then, a hierarchical grouping algorithm is recursively used to group smaller regions into larger regions until the whole image becomes a single region. During hierarchical grouping, they combine multiple grouping criteria such as similarities in color, texture, brightness, size and shape compatibility, thus able to deal with many image conditions as possible. After determining the image regions, they classify the object present in that region. For the classification, SVM with HOG is used by utilizing the bag-of-words for object recognition. We use the original MATLAB codes implemented by authors for comparison. We use the original MATLAB codes implemented by authors [37] for comparison.

On the other hand, some of the methods (such as the DPM + LSVM [30]) perform better when the hard negative mining (HNM) technique is applied. In HNM, negative examples (false positives) are feed into the classifier so that the classifier learns from the negative examples. Generally, few rounds of negative examples are applied. After few rounds, adding more negative examples does not improve the classification accuracy significantly and that there may be an inbalance between number of positive and negative samples (HNM may produce more negative examples). Our aim is to evaluate the effect of HNM on average precision values. Therefore, in the evaluations we included HNM in the training of the proposed method and in the training of all compared methods. For each method, we retrain the SVM with negative examples that are incorrectly classified. We repeat the process 2 rounds (2 cycles of hard negative mining) for each method.

8.1 Quantitative evaluation

Performance evaluation is based on comparing the detected BB locations with the manually labeled ground truth BB locations for test sequences. A detected BB and a ground truth BB form a potential match if they overlap sufficiently. Each detected BB and ground truth BB may be matched at most once. If a detected BB matches multiple ground truth BBs, the match with highest overlap is used. The overlap is measured with Eq. 7, and a correct detection is achieved if $\Gamma_{ij} > 0.25$. Note that Enzweiler et al. [38] also used $\Gamma_{ij} > 0.25$ to evaluate the pedestrian detection algorithms using Eq. 4. We measure the performance based on two different acceptable measurement methods. For the first measurement, we present the precision–recall curves and evaluate the average precision value for each method as in PASCAL VOC challenges [39]. For the second measurement, we evaluate the precision, recall and F-Score values at a single threshold.

8.1.1 Precision–recall curves and average precision value

We provide precision–recall curves and also report average precision over the fixed recall levels $[0, 0.1, 0.2, \ldots, 1]$. Here, the precision is defined as $P = P_c/P_t$, where P_c is the number of BB locations correctly predicted and P_t is the total number of BB locations predicted as belonging to player class. The recall (i.e., detection rate) is defined as $R = R_c/R_t$, where R_c is the number of BB locations correctly predicted and R_t is the total number of BB locations that actually belong to the player class. The precision at each recall level is interpolated; this also reduces the impact of wiggles in the precision–recall curves. The average precision (AP) summarizes the shape of the curve. This notation has been used in PASCAL VOC challenges [39]. Figure 6a–c shows the precision–recall curves of the methods for Camera #1 , Camera #2 and Camera #3 datasets, respectively. Since we apply HNM, we also illustrate average precision values with and without the HNM technique for all of the compared methods. Table 1 shows the average precision of the methods for each camera view with and without the HNM.

According to the average precision values in Camera #1 dataset, when the HNM is not applied, CNN [32], the proposed method, HOG + SVM [31] and PSHOG outperform the DPM + LSVM [30], BS [7], selective search [33], SIGD, Gauss and Morph methods. Without the HNM, CNN achieves the best accuracy and PSHOG is slightly behind the CNN. The proposed method performs slightly better than the HOG + SVM [31]. Although DPM + LSVM [30] achieves good results in pedestrian/person detection, this method is not good at player detection in field sports. Because the players appear at small scale, low resolution as well as different orientation because of the distance to camera and direction of their movement. It is difficult to distinguish and describe the human body parts under these conditions, and therefore the DPM + LSVM [30] method fails to detect players. In HOG+SVM [31], the HOG features describe the whole body shape and do not include the body part features separately in the description. This is the reason it performs better than the DPM + LSVM [30]. Describing the whole body shape alone is more effective when the object appears at small scale and low resolution. The selective search and DPM + LSVM methods achieve similar performances comparing to other methods. Although the selective search is good at detecting various objects, it is not good for detecting players at small scale, which is also shown by their results on VOC 2010 dataset [33]. The BS method [7] also fails to detect players because of variability of lighting and weather conditions as

Table 1 Average precision values (P) for each camera dataset (with and without hard negative mining (HNM))

Method	Camera 1 without HNM	Camera 1 with HNM	Camera 2 without HNM	Camera 2 with HNM	Camera 3 without HNM	Camera 3 with HNM
Proposed Features + SVM	0.9181	0.9364	0.9049	0.9459	0.9196	0.9308
HOG + SVM [31]	0.9171	0.9289	0.8514	0.9155	0.9204	0.9290
PSHOG	0.9239	0.9297	0.8957	0.9108	0.9115	0.9235
CNN	0.9259	0.9432	0.9028	0.9314	0.9223	0.9329
DPM + LSVM [30]	0.8055	0.8984	0.7980	0.8920	0.7859	0.8835
Selective search	0.7690	0.8103	0.8462	0.8790	0.7676	0.8062
BS [7]	0.6292	0.6675	0.4771	0.5538	0.5962	0.6258
SIGD	0.8865	0.9208	0.8662	0.9005	0.9146	0.9237
Gauss	0.8620	0.8902	0.8521	0.8804	0.9012	0.9127
Morph	0.8535	0.8893	0.8329	0.8523	0.8952	0.9062

well as low resolution. Overall, the results show that when the HNM is not applied, CNN, PSHOG and the proposed method achieve good performances. Using CNN as a feature extractor is the state-of-the-art deep learning method that we applied for player detection. Even though we have used a pre-computed AlexNet network, feature extraction using CNN is computationally expensive. In addition, with high-resolution images, generally CNN as a feature extractor achieves results close to 100%. However, as a requirement of the AlexNet, we re-size low-resolution images of very small-sized players that are captured from a distance to 227 × 227 image dimension requirements. As a result, this affected the performance of the CNN. On the other hand, without the HNM, the proposed method generally achieves good results since, in general, it can handle the distance (i.e., players' scales), low resolution, as well as the occlusion problems well.

When the HNM is applied, in Camera #1 dataset, the proposed method improves the performance considerably with 0.9364 average precision and it is slightly behind the performance of CNN (0.9432). Although the PSHOG and the HOG + SVM improve their performances with the HNM technique, they stay behind the performances of CNN and the proposed method. In addition, when the HNM is applied, we observed that the DPM + LSVM considerably improve the average precision values compared to other methods. This is because the DPM + LSVM method produces many bounding boxes and designed to learn from negative examples using HNM. Overall, the results show that when the HNM is applied, the CNN and the proposed method achieve the best results.

The resolution of the images captured by Camera #2 is a bit lower than the resolution of images captured by Camera #1. This difference appears because of some technical problems in Camera #2. From the average precision values in Camera #2 dataset, it can be observed that all of the methods are affected by the lower resolution problem except the proposed method (including with and without the HNM). Without the HNM, the average precision value of the proposed method is 0.9049, which is better than the other methods. The average precision values of HOG + SVM [31], BS method [7], DPM + LSVM [30], PSHOG, CNN [32], the selective search are 0.8514, 0.4771, 0.7980, 0.8957, 0.9028 and 0.8462, respectively (without the HNM). We observe that CNN is affected by low-resolution images. In this dataset, proposed method, CNN and HOG + SVM are the best performing techniques. The closest performance to these three methods is achieved by PSHOG. In particular, CNN and PSHOG handled the low-resolution images slightly better than the rest of the methods. When we look at the performance of the selective search method on different datasets, it is also shown that this method can handle low-resolution images better (i.e., performance on Camera #2) since it combines various similarity metrics during hierarchical grouping

Table 2 Comparison of the precision (P%), recall (R%) and F-score (F%) of the methods when the overlap measure is greater than 0.25

Cam.	# of players	BS [7]			DPM + LSVM [30]			HOG + SVM [31]			Proposed Feat. + SVM		
		P%	R%	F%	P%	R%	F%	P%	R%	F%	P%	R%	F%
1.	4526	84.25	68.45	75.53	91.48	80.21	85.48	98.51	89.24	93.65	99.21	89.11	93.89
2.	4780	78.47	64.96	71.08	90.45	74.53	81.72	98.91	83.54	90.58	97.90	87.57	92.45
3.	2407	79.78	81.64	80.70	84.92	86.83	85.86	93.13	95.18	94.14	94.93	97.30	96.10
Tot.	11,713	80.90	69.73	74.90	87.14	79.98	83.41	97.41	88.13	92.53	97.71	90.16	93.78

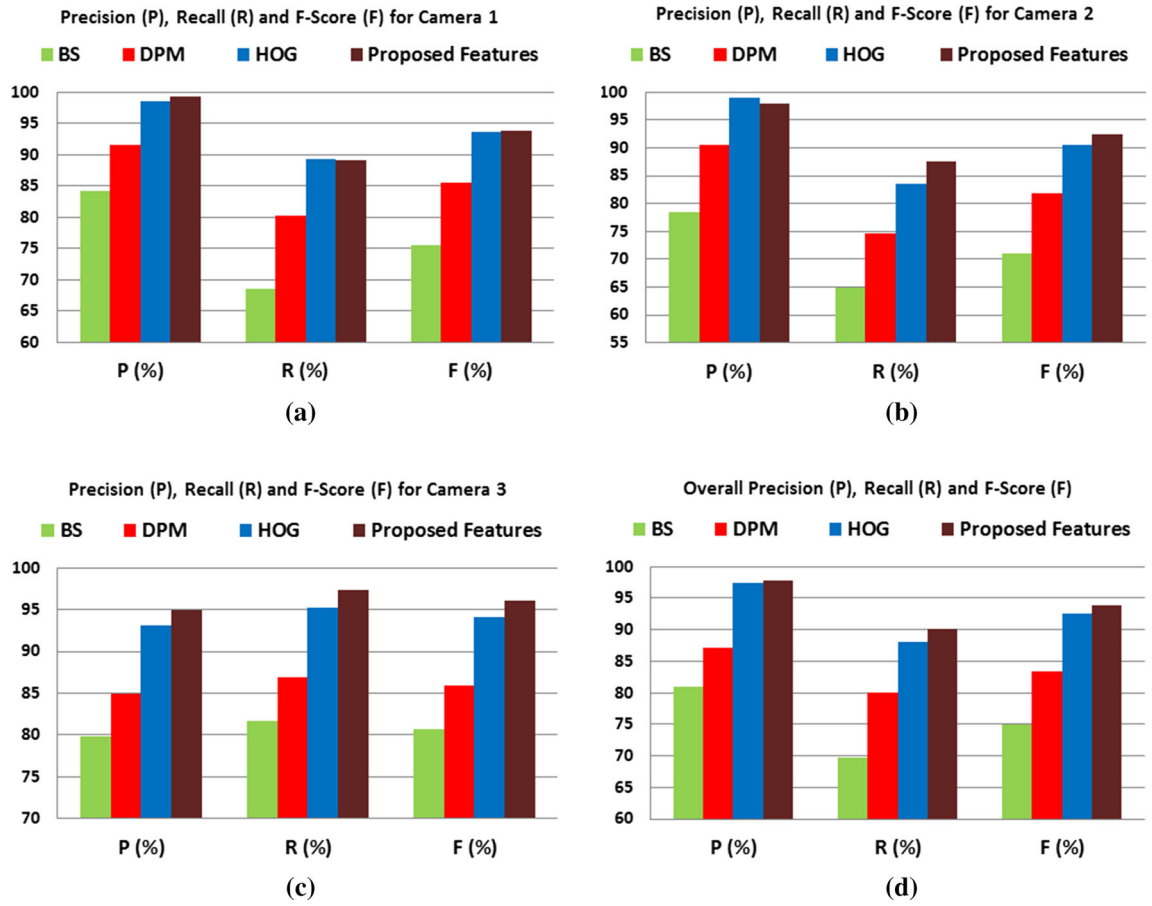


Fig. 8 Graphical illustration of the P%, R% and F% of the methods for the overlap measure is greater than 0.25. **a** Camera 1 dataset. **b** Camera 2 dataset. **c** Camera 3 dataset. **d** Overall performances

of image regions. When the HNM is applied, in Camera #2 dataset, the proposed method achieves the best results with 0.9449 average precision. This result shows that the proposed method both handles low-resolution images better and improves more with the HNM comparing to other methods. We also observed that again DPM + LSVM significantly improves the average precision with the HNM and also improvement in the PSHOG with the HNM stay limited comparing to other techniques (the PSHOG improve slightly).

The Camera #3 is a side view camera, and the dataset includes frames captured during the fast movement of players, when they were running. This causes higher variation of the human body shape with respect to time. According to the average precision results, the DPM + LSVM [30] method cannot handle the large variations of the body parts and perform worse comparing to other methods. In this dataset, with and without the HNM, the proposed method, CNN and HOG + SVM [31], outperforms other methods, and their average precision values are very close. In this dataset, performances of the rest of the methods are correlating with their performances on Camera #1 and Camera #2 datasets.

The novelty of the proposed method is mainly the use of the heat diffusion equation for shape representation. Therefore, we also perform comparison with the different shape

Table 3 Comparison of the precision (P%), recall (R%) and F-score (F%) of the methods when the overlap measure is greater than 0.5

Cam.	# of players	BS [7]			DPM + LSVM [30]			HOG + SVM [31]			Proposed Feat.+SVM		
		P%	R%	F%	P%	R%	F%	P%	R%	F%	P%	R%	F%
1.	4526	51.18	41.58	45.89	66.81	55.33	60.53	76.56	69.35	72.78	79.75	71.63	75.47
2.	4780	39.12	32.38	35.44	52.42	41.18	46.13	76.20	64.35	69.77	74.86	66.97	70.69
3.	2407	55.83	57.13	56.47	44.27	43.91	44.10	63.05	64.44	63.74	66.92	68.59	67.75
Tot.	11,713	47.58	41.02	44.05	53.83	46.03	49.63	73.28	66.30	69.61	74.89	69.10	71.87

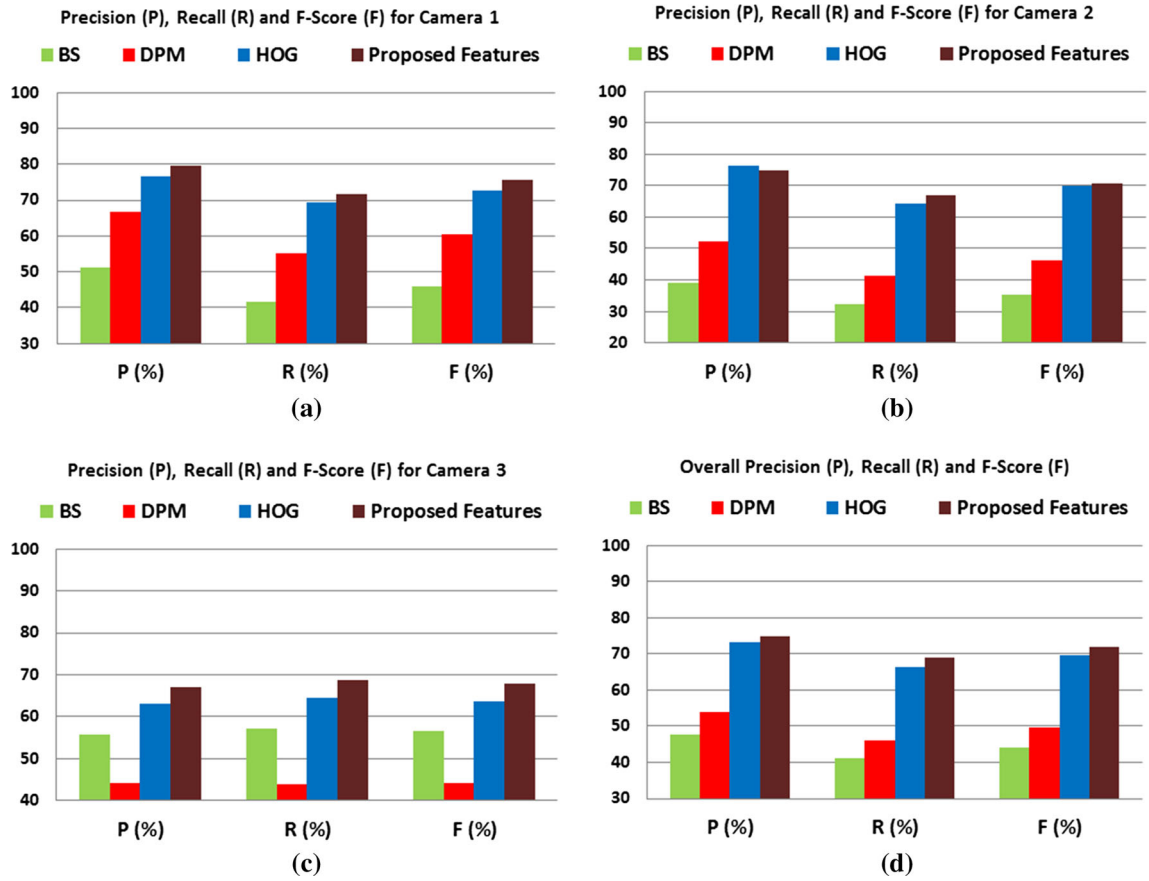


Fig. 9 Graphical illustration of the P%, R% and F% of the methods for the overlap measure is greater than 0.5. **a** Camera 1 dataset. **b** Camera 2 dataset. **c** Camera 3 dataset. **d** Overall performances

representations. In our algorithm, instead of using the proposed shape proposal we use other shape proposals for comparison. For example, we use binary foreground mask of an object appear after morphological operations to the binary edge image (abbreviated with MORPH). We also compare with the Gaussian-smoothed version of the binary foreground mask (abbreviated with Gauss). In addition, we compare with the shape that appears after passing the binary foreground mask through sigmoid function to get values between 0 and 1. This method is abbreviated with SIGD in evaluations. These evaluations are shown in Fig. 7 for all of the datasets. Again, we report the results with and without the HNM technique. Evaluations show that the proposed shape proposal performs consistently better than MORPH, Gauss and SIGD. In particular, in Camera #1 and Camera #2 datasets, the proposed method outperforms the other methods. On Camera #3 dataset, we observe very close performances, but the proposed method still achieves slightly better accuracy. With the HNM, performances of MORPH, Gauss and SIGD improve consistently, but they stay behind the performance of the proposed method with the HNM on all datasets.

In Table 1, average precision values of all methods are presented. The proposed method achieves the best accuracy on Camera #2 dataset and performs well in Camera #1 and #3 datasets (slightly behind the CNN), while HOG + SVM, PSHOG and CNN methods provide competitive accuracies

Table 4 Number of correct detections and detection rates (R%) of the methods in occlusion cases when the overlap measure is greater than 0.25

Occlusion cases	# of players	BS [7]		DPM + LSVM [30]		HOG + SVM [31]		Proposed Feat. + SVM	
		# of Det.	R%	# of Det.	R%	# of Det.	R%	# of Det.	R%
No Occ.	10,705	7858	73.40	8323	77.75	9601	89.68	9849	92.00
Partial Occ.	246	63	25.60	142	57.72	191	77.64	180	73.17
Heavy Occ.	762	247	32.41	305	40.03	531	69.68	532	69.98
Total	11,713	8168	69.73	8770	74.87	10323	88.13	10,561	90.16

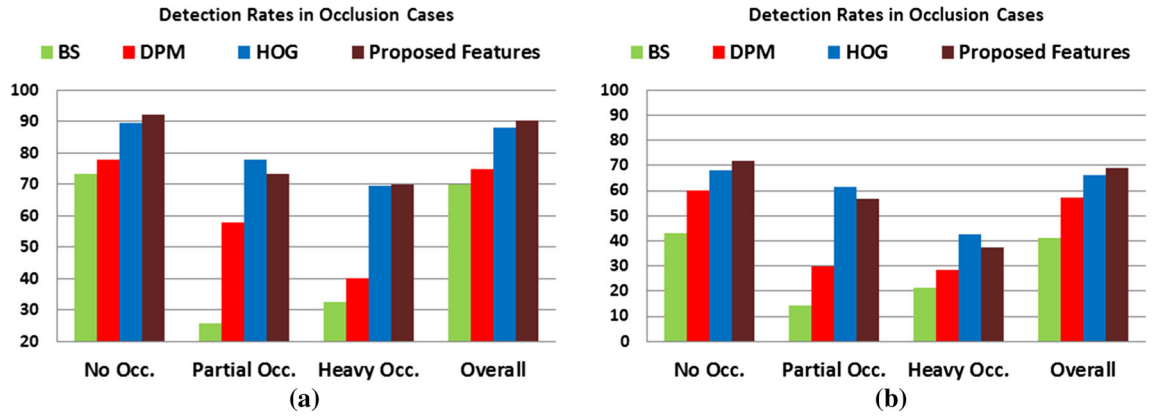


Fig. 10 Illustration of the detection rates in occlusion cases for the overlap measure is **a** greater than 0.25, and **b** greater than 0.5

comparing to the proposed method. Among the rest of the methods, BS performs the worst performance.

8.1.2 Precision, recall and F-score values at a single threshold

Since we compare performances of nine different methods, it is not feasible to assess performances of all these methods on various thresholds. Among these methods, we selected three of them for further analysis and comparison with the proposed model. In particular, HOG + SVM, PSHOG, CNN and the proposed method achieve good results and results are similar to each other. We selected HOG + SVM for comparison. The other method is DPM + LSVM, which provides similar results comparing to the selective search method. Finally, the worst performing method, BS is selected for further analysis. We present the precision, recall and F-score results at a single threshold for the overlap area is greater than 25 and 50% separately (i.e., for overlap measure is greater than 0.25 and 0.5). The threshold value for each method is determined experimentally on a different validation set. In this experiment, results are computed without the HNM. Here, the precision is defined as $P\% = (P_c/P_t) \times 100$, where P_c is the number of BB locations correctly predicted and P_t is the total number of BB locations predicted as belonging to player class. The recall (i.e., detection rate) is defined as $R\% = (R_c/R_t) \times 100$, where R_c is the number of BB locations correctly predicted and R_t is the total number of BB locations that actually belong to the player class. The F-score is a measure of accuracy that combines precision and recall results as follows: $F\% = 2{\cdot}((P\%{\cdot}R\%)/(P\%{+}R\%))$. In this evaluation, all of the measures must be high for a method to show that it can provide sufficient discrimination and detection. Table 2 and Fig. 8 show the precision, recall and F-score results, obtained using each method for each camera view, when the overlap measure is greater than 0.25. It is observed that the proposed features with SVM performs better than the other methods in each camera dataset. In total, 11,713 players are annotated for testing using these three camera views. In overall, the proposed method has the highest precision, recall (i.e., detection rate) and F-score (i.e., accuracy) as shown at the bottom of Table 2 and in Fig. 8d. The overall accuracy of the proposed method, HOG + SVM [31], the BS [7], and the DPM + LSVM [30] is 93.78, 92.53, 74.90 and 83.41%, respectively. The proposed method achieves better than other methods because, in general, it can handle the distance (i.e., players' scales), low resolution, as well as the occlusion problems better than the other methods.

Table 3 and Fig. 9 show the precision, recall and F-score results, obtained using each method for each camera view, when the overlap measure is greater than 0.5. It is again observed that the proposed features+SVM performs better than the other methods in each camera dataset. In total, the proposed method has the highest precision, recall and F-score (i.e., accuracy) as shown at the bottom of Table 3 and in Fig. 9d. The overall accuracy of the proposed method,

Table 5 Number of correct detections and detection rates (R%) of the methods in occlusion cases when the overlap measure is greater than 0.5

Occlusion cases	# of players	BS [7]		DPM + LSVM [30]		HOG + SVM [31]		Proposed Feat. + SVM	
		# of Det.	R%	# of Det.	R%	# of Det.	R%	# of Det.	R%
No Occ.	10,705	4606	43.03	6401	59.79	7292	68.12	7670	71.65
Partial Occ.	246	35	14.22	73	29.67	151	61.38	140	56.91
Heavy Occ.	762	164	21.52	218	28.61	323	42.38	284	37.27
Total	11,713	48.05	41.02	6692	57.13	7766	66.30	8094	69.10

Table 6 Average time required for player detection per frame

Camera	BS (s) [7]	DPM + LSVM (s) [30]	HOG + SVM (s) [31]	Proposed Feat. + SVM (s)
1.	16.78	34.80	30.91	44.79
2.	18.50	35.71	31.79	42.89
3.	17.09	32.67	27.29	33.55

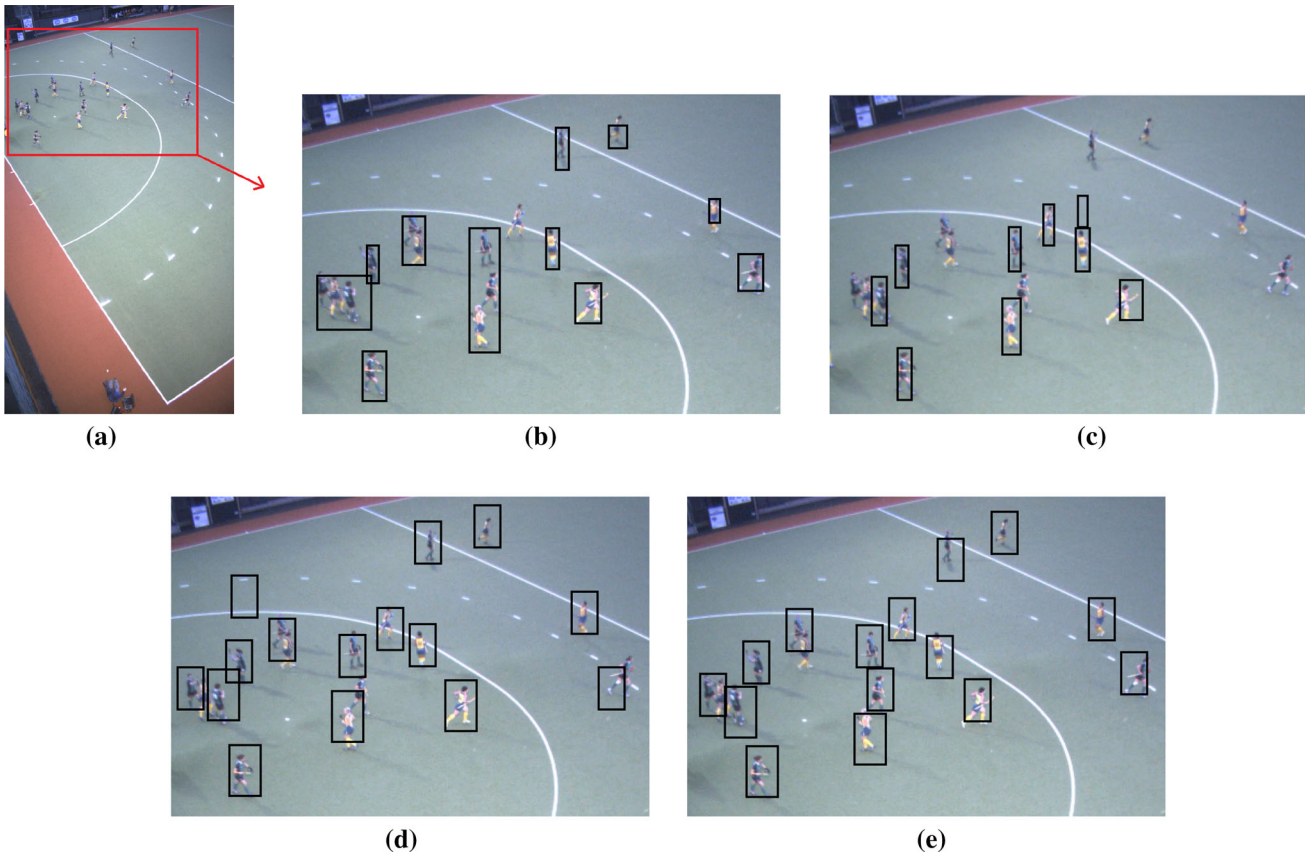


Fig. 11 Detection results in a frame from Camera 1 dataset (without Hard Negative Mining). **a** Sample Frame, **b** BS [7], **c** DPM + LSVM [30], **d** HOG + SVM [31], **e** Proposed Features + SVM

HOG + SVM [31], the BS [7] and the DPM + LSVM [30] is 71.87, 69.61, 44.05 and 49.63%, respectively. Therefore, the accuracy of the methods decreases if we restrict the overlap area, between the detection bounding box and the ground truth bounding box, to be greater than 50%. The reason is that the players' scales appear to be small in the datasets and this makes the detection bounding boxes rather imprecise.

8.2 Occlusion statistics and evaluation

We also annotated the occluded players in our datasets with two bounding boxes, where one of the BB denotes the visible and the other BB denotes the full player region. For each occluded player, we compute the fraction of the occlusion (i.e., one minus the visible player area divided by total

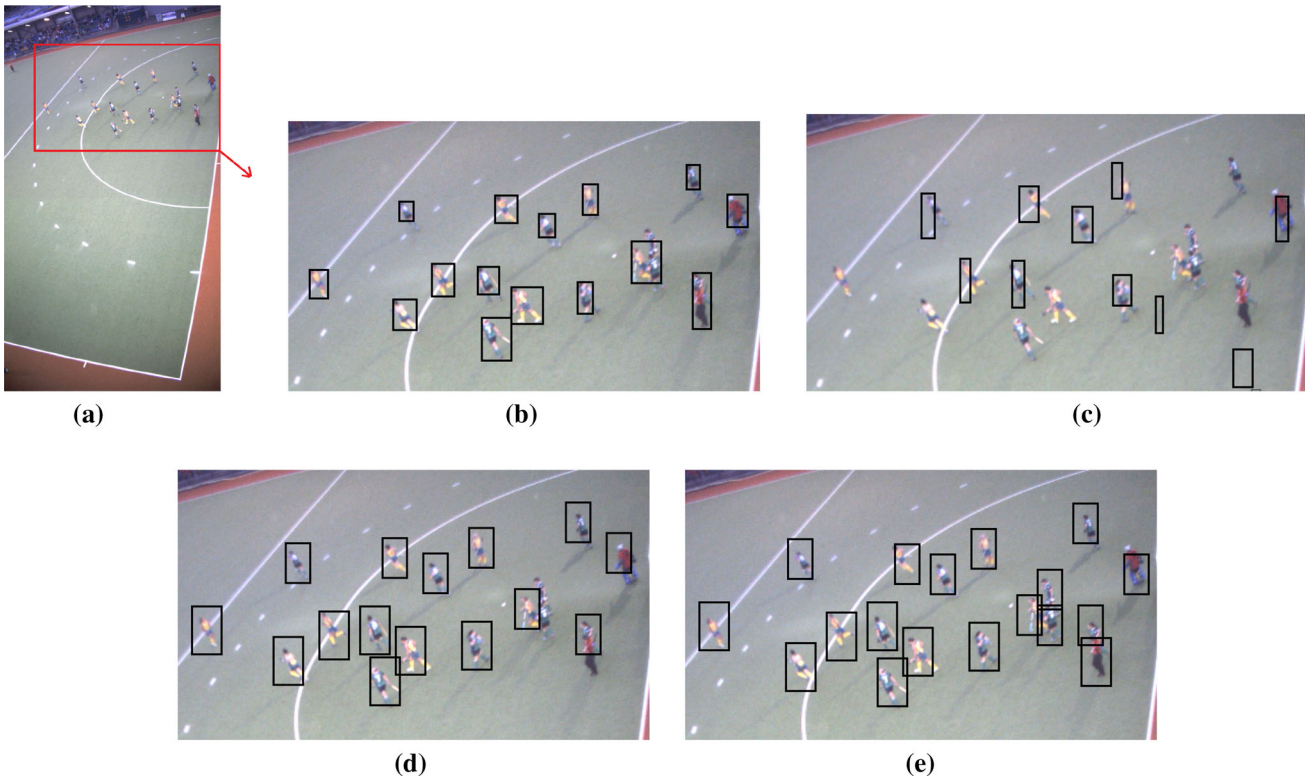


Fig. 12 Detection results in a frame from Camera 2 dataset (without Hard Negative Mining). **a** Sample Frame, **b** BS [7], **c** DPM + LSVM [30], **d** HOG + SVM [31], **e** Proposed Features + SVM

player area). Our dataset is divided into three occlusion cases: no occlusion (0% area occluded), partial occlusion (0–50% area occluded) and heavy occlusion (over 50% area occluded). Overall, there are 11,713 players in our dataset, where 10,705 players are not occluded, 246 players are partially occluded and 762 players are heavily occluded. We measure the performances of each method in different occlusion cases for the overlap area is greater than 25 and 50% separately (i.e., for overlap measure is greater than 0.25 and 0.5).

Table 4 shows the number of correct detections and the detection rates (i.e., recall) for each method in different occlusion cases and in total, when the overlap area is greater than 25%. Figure 10a also shows the detection rate (i.e., recall) for each method in different cases. It is observed that the proposed method and the HOG + SVM [31] perform significantly better than the BS method [7] and the DPM + LSVM [30] in all cases. In no occlusion case, the proposed method detects 9849 players out of 10,705, while the HOG + SVM [31] detects 9601 players. This means that our method can detect 248 (2.32%) more players than the HOG + SVM [31]. In partial occlusion case, the HOG + SVM [31] detects 191 players out of 246 and the proposed method detects 180 players. In partial occlusion case, the difference between the proposed method and the HOG + SVM [31] is 11 players (i.e., 4.47%). In heavy occlusion case, our method and the HOG + SVM [31] performs similarly, where our method detects 532 players out of 762 and the HOG + SVM [31] detects 531. In total, the proposed method can find 10561 players out of 11,713, while the HOG + SVM [31] can find 10323 players. Therefore, our method detects 238 more players (2.03%) than the HOG+SVM [31]. Overall, the proposed method performs better than the other methods.

Table 5 shows the number of correct detections and the detection rates for each method in different occlusion cases, when the overlap area is greater than 50%. Figure 10b also shows the detection rate (i.e., recall) for each method in different cases. It is observed that the proposed method and the HOG + SVM [31] perform significantly better than the other methods in all cases. In no occlusion case, the proposed method detects 7670 players out of 10,705, while the HOG + SVM [31] detects 7292 players. This means that our method can detect 378 (3.53%) more players than the HOG + SVM [31]. In partial occlusion case, the HOG + SVM [31] detects 151 players out of 246 and the proposed method detects 140 players. In heavy occlusion case, the HOG + SVM [31] detects 323 players out of 762 and the proposed method detects 284 players. In total, the proposed method can find 8094 players out of 11,713, while the HOG + SVM [31] can find

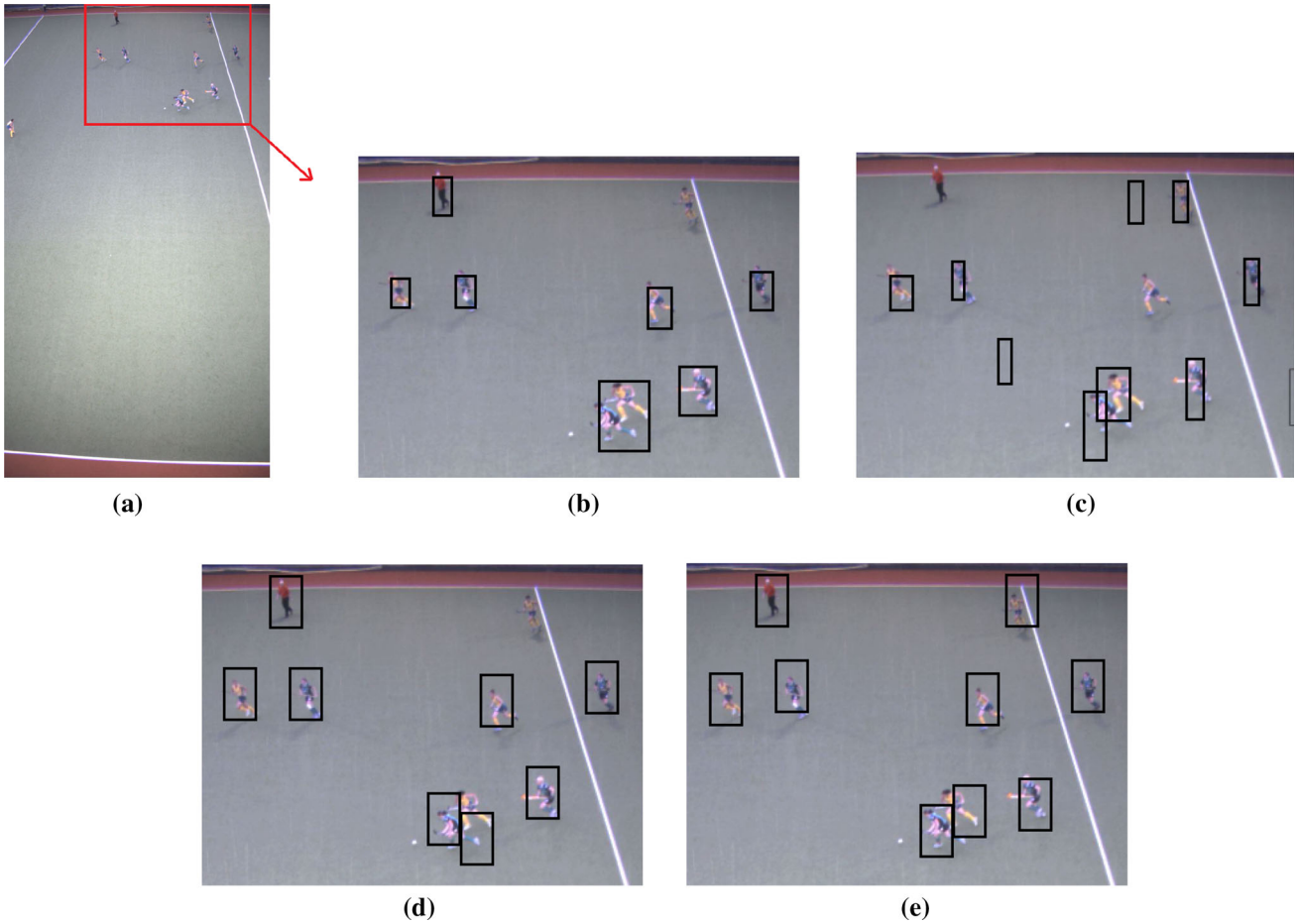


Fig. 13 Detection results in a frame from Camera 3 dataset (without Hard Negative Mining). **a** Sample Frame, **b** BS [7], **c** DPM + LSVM [30], **d** HOG + SVM [31], **e** Proposed Features + SVM

7766 players. Therefore, our method detects 328 more players (2.8%) than the HOG + SVM [31]. In total, the proposed method performs better than the other methods, when the overlap measure is greater than 0.5. The accuracy of the methods decreases if we restrict the overlap area, between the detection bounding box and the ground truth bounding box, to be greater than 50%. As we explained before, the reason is that players' scales appear to be small in the datasets and this makes the detection bounding boxes rather imprecise.

8.3 Computational efficiency

Table 6 shows the average time required for player detection per frame for each camera dataset. Results are obtained using MATLAB 7 on a Windows 7 Operating System with Intel Core i7-2670, 2.2 GHz and 8 GB RAM. It is observed that the BS [7] is more efficient than the proposed method, the HOG + SVM [31] and the DPM + LSVM [30]. Although the proposed approach is computationally less efficient, it has better accuracy than the other methods.

8.4 Discussions

The proposed method achieves better player detection because, in general, it can handle the distance (i.e., players' scales), low resolution, as well as the occlusion problems better than the other methods. For example, in low-resolution cases there are missing edges of an object in the image. Despite the missing edges, the solution of the proposed diffusion equation in the detector window can fill inside the object and preserve the shape information. Therefore, the extracted shape features become effective. On the other hand, the HOG features [31] are sensitive to low resolution and cannot detect players well in this case. The DPM + LSVM [30] method fails to detect players because it is difficult to distinguish and describe the player body parts when the player has small scale, lower resolution as well as large variations of body parts. Describing the whole body shape alone is more effective when the object appears at small scale and low resolution. The BS method [7] also fails to detect players because of variability of lighting, weather conditions, low resolution as well as when the players are very close or occluding each other.

We also present a visual comparison of the proposed method, the BS [7], the DPM+LSVM [30] and the HOG+SVM [31] on frames. The comparison is done for each camera view in Figs. 11, 12 and 13. In general, it can be seen that the proposed method performs better than the other methods in these frames.

9 Conclusions

We have presented an approach for player detection with a fixed camera based on a new feature extraction technique. We compute a binary edge image of a given frame, and then the detector window scans the edge regions. In each window, we solve a particular diffusion equation to generate a shape information image. This is the key stage and the main contribution in this new algorithm. Then, the shape information image is processed to extract scale- and rotation-invariant features. A SVM classifier is used to label the player regions. Our approach is evaluated on three different field hockey datasets. Results show that the proposed feature extraction is effective and performs competitive results compared to the state-of-the-art methods.

Acknowledgements This work is supported by Science Foundation Ireland under Grant 07/CE/I114. The authors would like to acknowledge Disney Research Pittsburgh for their help in constructing the camera network around the field hockey playground in Ireland. We also would like to thank Irish Hockey Association for their collaboration to collect the field hockey datasets.

References

1. Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H.: Automatic player detection, labeling and tracking in broadcast soccer video. Pattern Recognit. Lett. **30**(2), 103–113 (2009)
2. Khatoonabadi, S.H., Rahmati, M.: Automatic soccer players tracking in goal scenes by camera motion elimination. Image Vis. Comput. **27**(4), 469–479 (2009)
3. Beetz, M., Gedikli, S., Bandouch, J., Kirchlechner, B., Hoyningen-Huene, N., Perzylo, A.: Visually tracking football games based on TV broadcasts. In: International Joint Conference on Artificial Intelligence, pp. 2066–2071 (2007)
4. D'Orazio, T., Leo, M.: A review of vision-based systems for soccer video analysis. Pattern Recognit. **43**(8), 2911–2926 (2010)
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
6. Xu, M., Orwell, J., Lowey, L., Thirde, D.: Architecture and algorithms for tracking football players with multiple cameras. IEE Vis. Image Signal Process. **152**(2), 232–241 (2005)
7. Figueroa, P.J., Leite, N.J., Barros, R.M.L.: Tracking soccer players aiming their kinematical motion analysis. Comput. Vis. Image Underst. **101**(2), 122–135 (2006)
8. Hamid, R., Kumar, R.K., Grundmann, M., Kihwan, K., Essa, I., Hodgins, J.: Player localization using multiple static cameras for sports visualization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 731–738 (2010)
9. Kim, K., Grundmann, M., Shamir, A., Matthews, I., Hodgins, J., Essa, I.: Motion fields to predict play evolution in dynamic sport scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 840–847 (2010)
10. Carr, P., Sheikh, Y., Matthews, I.: Monocular object detection using 3D geometric primitives. In: European Conference on Computer Vision, vol. 7572, pp. 864–878 (2012)
11. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 246–252 (1999)
12. Li, L., Huang, W., Gu, I., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Process. **13**(11), 1459–1472 (2004)
13. Lee, D.S.: Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 827–832 (2005)
14. Han, B., Davis, L.S.: Density-based multifeature background subtraction with support vector machine. IEEE Trans. Pattern Anal. Mach. Intell. 34(5), 1017-1023 (2012)
15. Vandenbroucke, N., Macaire, L., Postaire, J.G.: Color image segmentation by pixel classification in an adapted hybrid color space: application to soccer image analysis. Comput. Vis. Image Underst. **90**(2), 190–216 (2003)
16. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 629–639 (1990)
17. Manay, S., Yezzi, A.: Anti-geometric diffusion for adaptive thresholding and fast segmentation. IEEE Trans. Image Process. **12**(11), 1310–1323 (2003)
18. Direkoglu, C., Dahyot, R., Manzke, M.: On using anisotropic diffusion for skeleton extraction. Int. J. Comput. Vis. **100**(2), 170–189 (2012)
19. Direkoglu, C., Nixon, M.S.: Image-based multiscale shape description using Gaussian filter. In: Sixth Indian Conference on Computer Vision, Graphics, Image Processing (2008)
20. Makrogiannis, S.K., Bourbakis, N.G.: Motion analysis with application to assistive vision technology. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 344–352 (2004)
21. Direkoglu, C., Nixon, M.S.: Moving-edge detection via heat flow analogy. Pattern Recognit. Lett. **32**(2), 270–279 (2011)
22. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
23. Holman, J.P.: Heat Transfer, 9th edn. McGraw-Hill, New York (2002)
24. Ursell, T.: The Diffusion Equation—A Multi-Dimensional Tutorial. http://www.rpgroup.caltech.edu/natsirt/aph162/diffusion.pdf (2007)
25. Direkoglu, C.: Feature extraction via heat flow analogy. Ph.D. Thesis, University of Southampton, UK (2009)
26. Trottenberg, U., Oosterlee, C., Schuller, A.: Multigrid. Academic Press, London (2001)
27. Direkoglu, C., Nixon, M.: Shape classification via image-based multiscale description. Pattern Recognit. **44**(9), 2134–2146 (2011)
28. Zhang, D.S., Lu, G.: Generic Fourier descriptor for shape-based image retrieval. In: IEEE International Conference on Multimedia and Expo, vol. 1, pp. 425–428 (2002)
29. Wood, J.: Invariant pattern recognition: a review. Pattern Recognit. **29**(1), 1–17 (1996)
30. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
31. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)

32. Image Category Classification using Deep Learning. https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html
33. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
34. Object detection with discriminatively trained part based models. Matlab code: http://cs.brown.edu/pff/latent-release3/
35. Hertel, L., Barth, E., Kaster, T., Martinetz, T.: Deep convolutional neural networks as generic feature extractors. In: International Joint Conference on Neural Networks (2015)
36. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems Conference, pp. 1097–1105 (2012)
37. Selective search for object recognition. Matlab code: http://koen.me/research/selectivesearch/
38. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: survey and experiments. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12), 2179–2195 (2009)
39. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)