INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2013, VOL. 59, NO. 1, PP. 75–84
Manuscript received February 13, 2013; revised March, 2013. DOI: 10.2478/eletel-2013-0009

Segmentation of Football Video Broadcast

Sławomir Maćkowiak

***Abstract*—In this paper a novel segmentation system for football player detection in broadcasted video is presented. Proposed detection system is a complex solution incorporating a dominant color based segmentation technique of a football playfield, a 3D playfield modeling algorithm based on Hough transform and a dedicated algorithm for player tracking, player detection system based on the combination of Histogram of Oriented Gradients (HOG) descriptors with Principal Component Analysis (PCA) and linear Support Vector Machine (SVM) classification. For the shot classification the several classification technique SVM, artificial neural network and Linear Discriminant Analysis (LDA) are used.**

Evaluation of the system is carried out using HD (1280×720) resolution test material. Additionally, performance of the proposed system is tested with different lighting conditions (including non-uniform pith lightning and multiple player shadows) and various camera positions.

Experimental results presented in this paper show that combination of these techniques seems to be a promising solution for locating and segmenting objects in a broadcasted video.

***Keywords*—segmentation, video surveillance, stereoscopic video, sport video sequences.**

I. Introduction

SEGMENTATION plays an important role in digital media processing, pattern recognition, and computer vision. The task of image/video segmentation emerges in many application areas, such as image interpretation, video analysis and understanding, and video summarization and indexing. Over the last two decades, the problem of segmenting image/video data has become a fundamental one and had significant impact on both new pattern recognition and applications.

Although detection and tracking of objects in video is commonly known in literature, most of the existing approaches assume specific conditions such as fixed cameras, single moving object, and relatively static background. In sports video broadcasts, such strict conditions are not applicable. Firstly, the cameras that are used to capture sports games are not static and they are in almost permanent motion. A broadcasted video is the one selected according to the broadcast director's instruction from frequent switches among multiple cameras. Thirdly, there are numerous players moving in various directions in the broadcasted video. Finally, the background in sports video changes rapidly. Those conditions make detection and tracking of objects in broadcasted video difficult.

The main goal of the paper is to present the system dedicated to sports application where many cameras, many different shots, many different lightning conditions and fast moving objects exists in a sequence together.

S. Maćkowiak is with the the Faculty of Electronics and Telecommunications at Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland (e-mail: smack@et.put.poznan.pl).

Based on observed characteristics of various broadcasted football games and analyses on difficulties of existed algorithms, the author proposes a novel approach which uses a dominant color based segmentation for football playfield detection, line detection algorithm based on the Hough transform to model the playfield and a combination of Histogram of Oriented Gradients (HOG) descriptors [1] with Principal Component Analysis (PCA) and Support Vector Machine (SVM) as a classifier [2] to detect players and player tracking system. For the shot classification the several classification technique SVM, artificial neural network and Linear Discriminant Analysis (LDA) are compared and present in the paper.

In order to create a complex football video segmentation system several types of techniques need to be incorporated. One of techniques used for dominant color detection in the playfield detection is MPEG-7 dominant color descriptor (DCD), however, it operates on three dimensional color representation and its results are not illumination independent [2]. Approach [3] is based on Euclidean distance to trained dominant color in IHS color space. Ren et al. [4] presented an image block classification method based on color hue variance followed by hue value classification by trained Gaussian mixture model.

Most of line detection algorithms used in the playfield line detection are based on Hough transform of binary line image [5] which can detect presence of a straight line structure and estimate its orientation and position. Some other approaches use modified Hough transforms like probabilistic Hough transform [6] or Block Hough transform [7] for computation speed improvements. Thuy et al. [8] proposed Hough transform modification which allows line segment detection instead of straight line presence. On the other hand, random searching methods might be also used. Such methods [5] incorporate a random searching algorithm which selects two points and checks whether there is a line between them. Another issue is line image generation. Here, edge detection approaches and other gradient based techniques perform best [5].

Object detection is always based on extraction of some characteristic object features. Dalal et al. [9] introduced a HOG descriptor for the purpose of pedestrian detection and achieved good results.

Another important issue in object detection is object classification which separates objects belonging to different classes to distinguish requested objects from the others. One of the most commonly used object classifiers is SVM classifier which has been successfully applied to a wide range of pattern recognition and classification problems. The advantages of SVM compared to other methods are:

1) better prediction on unseen test data,
2) a unique optimal solution for training problem,
3) fewer parameters.

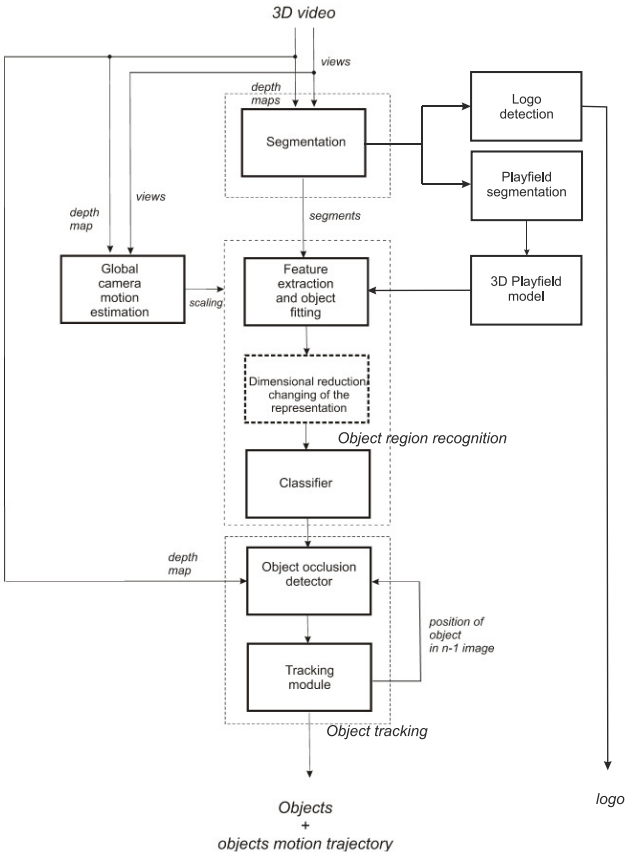

Fig. 1. The proposed object extraction algorithm when we deal with multiple cameras, changing the camera position and focal length.

Other classification systems: Artificial Neural Network and Linear Discriminant Analysis are also used in the paper.

II. Overview of the System

Specific conditions in segmentation of football video broadcast require an adequate approach therefore a dedicated system for football player detection was proposed. After analyzing various segmentation and tracking techniques, the authors proposed a solution that combines a segmentation method and a method of tracking of segmented regions using nonlinear classifiers and detector overrides. The proposed system is shown in Fig. 1. The main components of the system are: playfield detector, playfield model fitter and object region recognition and object tracking module.

The camera global motion estimation algorithm is used to improve scene objects segmentation and tracking algorithms and to detect a zoom in the analyzed sequence. This can also help to better adjust the size of detection windows used for object detection. The proposed algorithm was presented in the previous papers [10], [11].

The remaining blocks are detailed in several next sections.

III. Logo Segmentation

Static logo detection (one of the first step in video broadcast background analysis) is used to preserve global motion estimation and field detection algorithms from errors and also can be incorporated for the purpose of semantic scene description.

The procedure of estimating static logo position is based on [5]. Subsample frames in time at a rate eg. of one frame per second (when subsample frames is too high, is more substance than just a logo after the logo detection). For each frame Canny edge detector is used (weak and strong edge threshold values are equal 0.1 and 0.3 respectively). Next the edges are detected in time according to a formula:

$$S_i = \alpha S_{i-1} + (1-\alpha)E_i, \begin{cases} \alpha = \frac{i-1}{i}, & i \le n \\ \alpha = \frac{n-1}{n}, & i > n \end{cases}$$

where i is the frame index, E_i is edge field detected in step 2, S_i is time averaged edge field and n is a logo refresh parameter. In the experiment $n = 10$ was assumed, which enables refreshing of the logo in less than 30 [s]. Next, edge pixels time-consistency is checked – edge pixels with S_i value exceeding predefined time consistency threshold are classified as potential logo. When the consistency is checked, also the edge size must be checked – edges longer than predefined threshold value are classified as logo. Edge pixels smaller than predefined threshold are also classified as logo, but only if they are located close to large edges. On the edge image, the morphological operations – closing, hole filling and opening morphological operations are applied.

Each separated logo area is represented as a separated segment. Next, logo segments close to each other are merged. Finally, logo segments smaller than predefined value are discarded. Each logo segment is represented as a rectangle which covers the whole area containing this logo segment.

The algorithm requires short learning procedure at the beginning of analyzed sequence. However, the number of frames needed for this purpose is small and learning period of 2 [s] should be sufficient.

IV. Playfield Segmentation

Accurate detection of a playfield area is very important for further segmentation process. In order to do accurate detection, some assumptions are done. First, a playfield is a homogenous region with relatively uniform hue. Because of possible shadows and highlights, from the segmentation point of view, the playfield area may appear as a set of smaller areas. Nevertheless, those areas are expected to exhibit the relatively uniform hue. Another assumption is related to the size of the playfield. The playfield is supposed to be the largest homogenous area in the whole image. In close-up views, playfield covers the whole image, therefore can be considered as background.

For each video frame, playfield detection is performed at first. The proposed flow diagram of algorithm is shown in Fig. 2.

The first step is creation of 2D chrominance histogram of each frame. Histogram is then smoothed using Gaussian kernel 2D FIR filter to remove chrominance noise effects. After that, color quantization is performed. Colors are quantized using 2D vector quantization algorithm, where vectors consist of chrominance values (U and V) of colors. Number of quantization bins (Voronoy cells) is fixed during processing time. This number is chosen according to experimental results. Quantization independently performed on subsequent frames

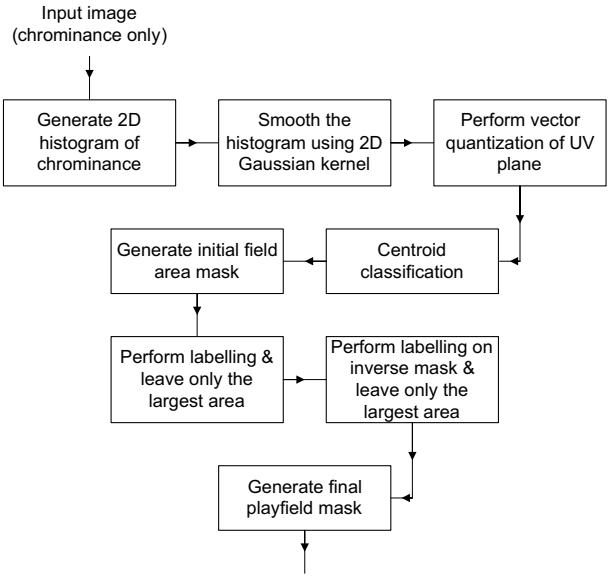

Fig. 2. Playfield detection algorithm flow diagram.

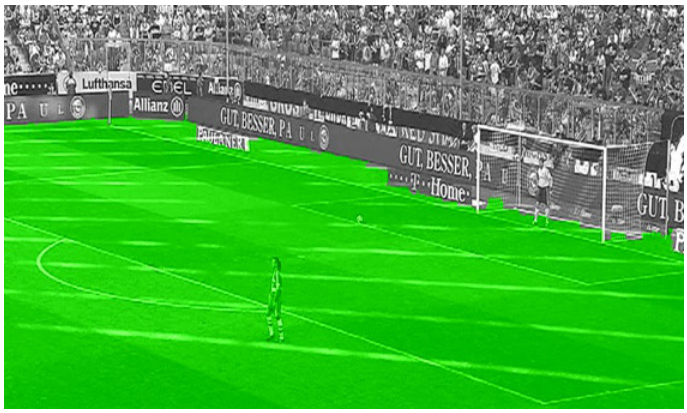

Fig. 3. A result of the playfield detection.

would may lead to unstable results (rapid changes of resulting colors values). In the software, temporal consistency of quantized colors (centroids of Voronoy cells) is provided by iterative LBG vector quantization algorithm that is used. The one iteration of LBG is performed on one of the consecutive frames. This leads to smooth evolution of centroid positions calculated for consecutive frames. Any color change caused either by camera adaptation (white balance & exposure) or camera motion will not lead to abrupt changes of quantized colors.Vector quantization produces information of presence of dominant colors in an image. In order to distinguish between playfield an non-playfield colors further classification is needed. Centroids are classified basing on their representing vector's angle which is similar in interpretation to color hue. Centroid vector's angle is then compared to the two previously defined, fixed values which define green color range. Centroids which fall into that range are classified as playfield area, others are classified as non-playfield area. On this basis, initial playfield mask is generated. The result of the playfield detection is presented in Fig. 3.

The next step in the playfield segmentation is line detection in the area of the playfield. Algorithm is divided into two main stages: line detection and line parameter extraction and tracking.

In order to detect lines in a single frame a modified approach from [5] is used. As the first step, the four directional gradient images are created using a set of derivative of Gaussian masks. The gradient directions are $0°$, $45°$, $90°$, $135°$. Then for each gradient image a centerline response is computed. The centerline feature is defined as presence of rising and falling gradient along image's gradient direction. As a final centerline response maximum value of all four responses is taken. As the next step centerline response image is thresholded using adaptive threshold computed for each pixel independently using its neighborhood. In order to remove small artifacts image is filtered with morphological closing filter (to connect possibly shattered larger areas) and then subjected to labeling procedure. During labeling of all disconnected regions, their' sizes are computed and at the end only the largest area is left. The final step of line detection is morphological thinning. The thinning procedure thins every line to 1 pixel thick which allows further parameter extraction to be more accurate.

The main tool used in line parameter extraction is a Hough transform. The author research showed, that applying Hough transform directly to line image may result in many false detections. To overcome that problem, method used in [6]–[8] was chosen. Line image from previous stage is divided into rectangular blocks (which may overlap) called linelets. Then, for each linelet, line parameters are extracted using a linear regression. If the regression error is too high, block is rejected as it does not contain valid line fragment. After processing blocks, their parameters are used in a voting procedure of Hough transform. Detection of Hough transform peaks is done via adaptive thresholding of Hough transform accumulator. Each peak represents single line by providing its angle and distance to an origin point.

Because of finite Hough transform accumulator resolution, it is necessary to perform further line parameter refinement. For each line a linear regression is computed using pixels that lies closer to line than predefined distance threshold. If the regression error is too high, line is rejected as a false detection. There is a possibility that after parameter refinement two or more lines may end with the same or very similar parameters. These lines are aggregated by averaging their parameters. Finally a set of detected lines is subjected to final stage which is line tracking.

For each new frame, existing lines are compared with newly detected ones. If their parameters are similar, then lines are joined into single line with parameters of newly detected one. Each tracked line has two counters: lifetime counter and timeout counter. Line is considered valid if its lifetime reaches predefined threshold (line must exist for some time). If a tracked line cannot be joined with any newly detected line then its timeout is increased. If line's timeout reaches its threshold value, line is removed. Finally, set of tracked lines is outputted as final line detection result (Fig. 4).

Fitting of playfield model (Fig. 5) allows to position fragment of playfield on video frame. Information of real playfield

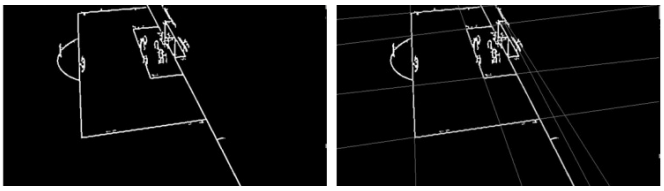

Fig. 4. A result of the line detection algorithm.

relation to video frame might be used to obtain precise positions of players.

Playfield model is defined by set of line sections with starting and ending point 2D coordinates. The author assumes that the model is flat and constructed on Z-plane. Model fitting algorithm is based on [12]. In order to find exact transformation between model space and video frame space we need 4-point to 4-point correspondence. Both model lines and detected video frame lines are classified into subsets of most parallel lines (meaning lines of nearly the same direction). Then for each two lines of each two classes four crossing points are calculated and a set of point quadruples for both model and video frame is constructed.

For each 4-point to 4-point correspondence a transformation matrix is created according to [12]. Then quality of each solution is measured by comparing position of transformed playfield model lines with video frame line pixels provided by line detection algorithm. Solution that places model closest to the real lines is chosen as final.

The detail of the calculation of camera parameters from homography matrix, which is used to 3D model fitting is described by the author in the previous work [11].

Playfield model fitting is an experimental algorithm, some parts of it still needs some improvements. Currently there is no tracking of playfield model, solutions for subsequent frames are completely independent.

V. Player Detection

In the classical segmentation algorithms, a major problem appears to be low quality video as well as problems resulting from the dynamically changing content of the images. Object segmentation algorithms do not calculate the position of the

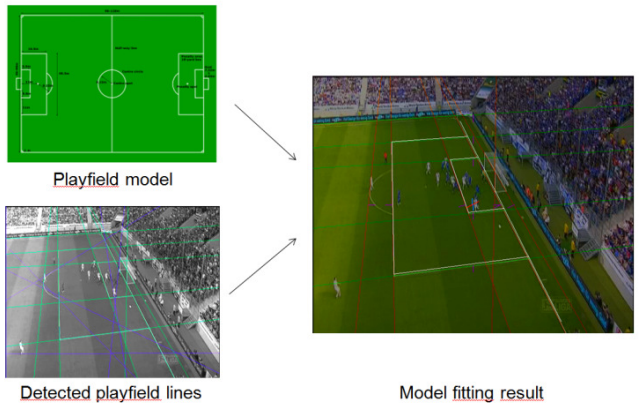

Fig. 5. Fitting of playfield model.

Fig. 6. A result of playfield model fitting.

objects correctly, therefore, they are complemented by the object detection method basing on characteristic features search. For this purpose, a feature descriptor operating in an adaptively selected, predefined window around the position of the object (the window is a potential candidate to detect the object) is constructed. The best results are obtained using locally one of the methods: SIFT, SURF or HOG [1], [9], [13], [14]. Generally, the idea of operation of these methods is similar and bases on finding stable local features within a defined search window. Features detected by the algorithms are chosen in such a way that they are not sensitive to changes in scale and orientation, as well as minor changes in illumination, noise, and shifting points of view. An important feature of these methods is resistance to partial covering of the objects. Therefore, these descriptors have become extensively used in the segmentation process improvement issue.

As main player and non-player distinguishing feature HoG detector has been chosen. A non-modified version described in [15] is used. The window size of 16×32 pixels divided into 8×8 blocks constructed of 2×2 cells is used. Blocks overlap by half of its' sizes (4 pixels). L2Hys [9] block normalization scheme is used. The idea of the normalization of the HOG blocks in this way is drawn from the original work of the HOG. The gradient histogram consists of 9 bins and is computed using maximum gradient values of all three RGB channels. With all these parameters, single HoG descriptor contains 756 numbers.

In contrast to previous work [10], [11] additionally PCA (Principal Component Analysis) is used. PCA is used here to reduce the dimensionality of the HOG descriptor. In this way more distinctive representation is received. A smaller size of descriptor than 756: 150, 175 and 225 PCA-HOG features for profile was selected, frontal and vertical pose respectively. Evaluation results demonstrate that PCA-HOG detector performs better than pure HOG detector with respect to the precision metric.

At the beginning pure HOG descriptor is determined separately for the positive and negative images. The data are

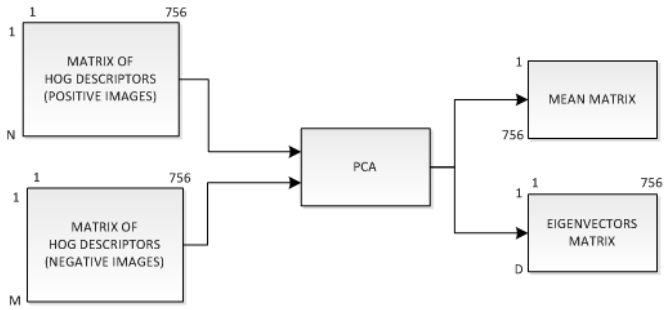

Fig. 7. Principal Component Analysis of input images.

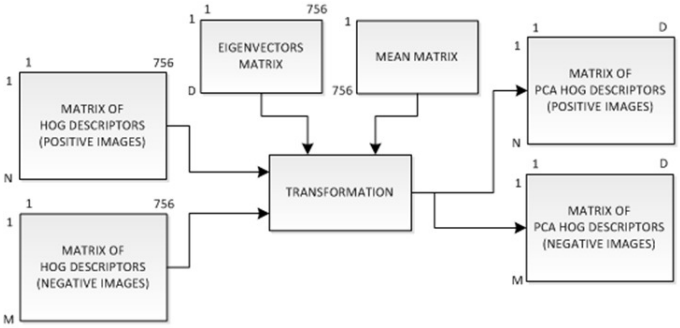

Fig. 8. PCA-HOG descriptors computation.

grouped into two matrices as presented in Fig. 7 (N and M are the total numbers of positive and negative images respectively). Then, Principal Component Analysis that results in the dimensionality reduction (D features) is performed. The result is the mean and eigenvectors matrices that are further used to project HOG descriptors of the positive and negative images to linear subspace. In this way the PCA-HOG descriptors are generated. The whole process is illustrated in Fig. 8.

In the similar way PCA-HOG descriptor is computed for any location of the window detection in an analyzed frame: pure HOG descriptor computation, transformation to linear subspace with the eigenvectors and mean matrices.

As a basic classification method SVM (Support Vector Machine) classifier has been chosen. Because of large variety of possible player postures a single classifier would not be enough. Three SVM classifiers are used. The classifiers working in parallel in order to detect different poses of players: first one was trained on images with vertical frontal poses, second on vertical profile poses and the last on joint set of all vertical poses. All SVM classifiers were using the same negative sample set.

The player template database contains over 600 vertical frontal and vertical profile poses as positive examples and over 3000 negative vertical, non-player images. The positive templates were manually generated, the negative examples were obtained manually and by bootstrapping procedure.

Box aggregation is an important step which decreased the number of false detections, as some of resultant boxes from HoG+SVM detector module usually contain only parts of the

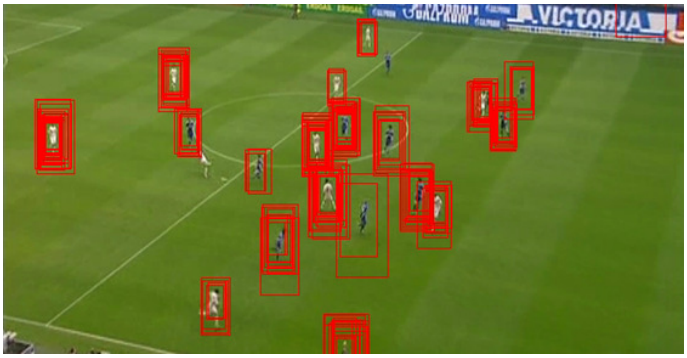

Fig. 9. Result of a classification of the players.

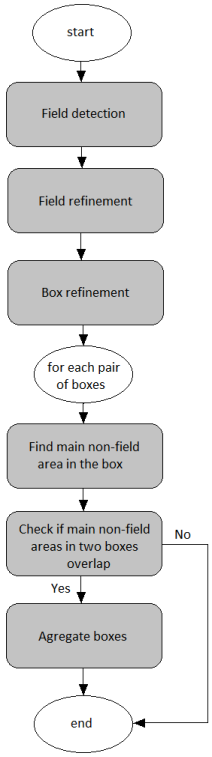

Fig. 10. Box aggregation algorithm.

players body (ex. leg or arm) (Fig. 9). To overcome this problem an additional merging operation was proposed in order to improve performance of the segmentation algorithm. The merging algorithm is presented in Fig. 10. In presented approach aggregation of boxes is independent for each processed frame.

At each frame, currently tracked bounding boxes are compared with boxes obtained by HoG detector for that frame. Comparison is based on size and overlap area. For each tracked box a cost of similarity between a current box and a candidate box is evaluated. The cost function incorporates the overlap area and the size and it is defined as follows:

$$cost = \left[1 - \left(\frac{overlap\_area}{min(size1, size2)}\right)\right] + \\ + \left[1 - \frac{min(size1, size2)}{max(size1, size2)}\right]$$

If overlap area is smaller than predefined threshold the candidate box is rejected. After cost evaluation, the candidate box with minimal cost is considered to be the bounding box of the same player as the currently processed box. If a new box is not matched with an existing one then it's added to tracked boxes list.

During sequence's time flow each tracked box has assigned a motion vector. The motion vector calculation is based on position of the box in previous frames. It is an average of all motion vectors computed between a position in the current frame and each of the memorized positions in the previous frames. If a tracked box cannot be matched with any box from detection results, its position is predicted using the motion vector. At the same time a timeout counter of the box is increased. If timeout counter reaches its threshold value, the tracked box is removed.

It is possible to appear only one box for one frame during a detection error. In such case motion vector cannot be calculated. Such boxes are rejected as false detections.

A. Detection Results

Three measures are used to perform the detection evaluation: precision, recall and missed ratio. Precision and recall are defined as follows:

$$precision = TP/(TP + FP),$$
$$recall = TP/(TP + FN),$$

where TP is the set of true positives, FP is the set of false positives (false detections) and FN is the set of false negatives (missed objects). The set of true positive, false positive and negative are defined as:

$$\boldsymbol{TP} = \{\boldsymbol{r}|\boldsymbol{r} \in \boldsymbol{D} : \exists \boldsymbol{g} \in \boldsymbol{G} : \boldsymbol{s_0}(\boldsymbol{r}, \boldsymbol{g}) \geq \boldsymbol{T}\},$$
$$\boldsymbol{FP} = \{\boldsymbol{r}|\boldsymbol{r} \in \boldsymbol{D} : \forall \boldsymbol{g} \in \boldsymbol{G} : \boldsymbol{s_0}(\boldsymbol{r}, \boldsymbol{g}) < \boldsymbol{T}\},$$
$$\boldsymbol{FN} = \{\boldsymbol{r}|\boldsymbol{r} \in \boldsymbol{G} : \forall \boldsymbol{g} \in \boldsymbol{D} : \boldsymbol{s_0}(\boldsymbol{r}, \boldsymbol{g}) < \boldsymbol{T}\},$$

$s_0(a, b)$ is called a degree of overlap between two regions a and b (i.e. bounding boxes of detected objects):

$$s_0(a, b) = \frac{|a \cap b|}{|a \cup b|}.$$

T is a threshold defining the degree of overlap required to determine two regions as overlapping. The set of ground truth regions G and detected regions D for a given frame are defined as: $G = \{g_1, \ldots, g_n\}$ and $D = \{d_1, \ldots, d_m\}$, with n – the number of ground truth regions and m – the number of detected regions in analyzed frame.

The last metric (missed ratio) represents the percentage of the undetected players for the given overlap degree (T).

The system performance is evaluated with threshold values T equal 0.4 to 0.9 using three different Support Vector Machines (SVM) namely: vertical, amface and profile for different stages of detection.

The quality of the system depends mainly on the pose, type of HOG descriptor, number of features used, kind of data (no occlusion set, whole set). The quality of the system is measured using the precision and missed ratio parameters

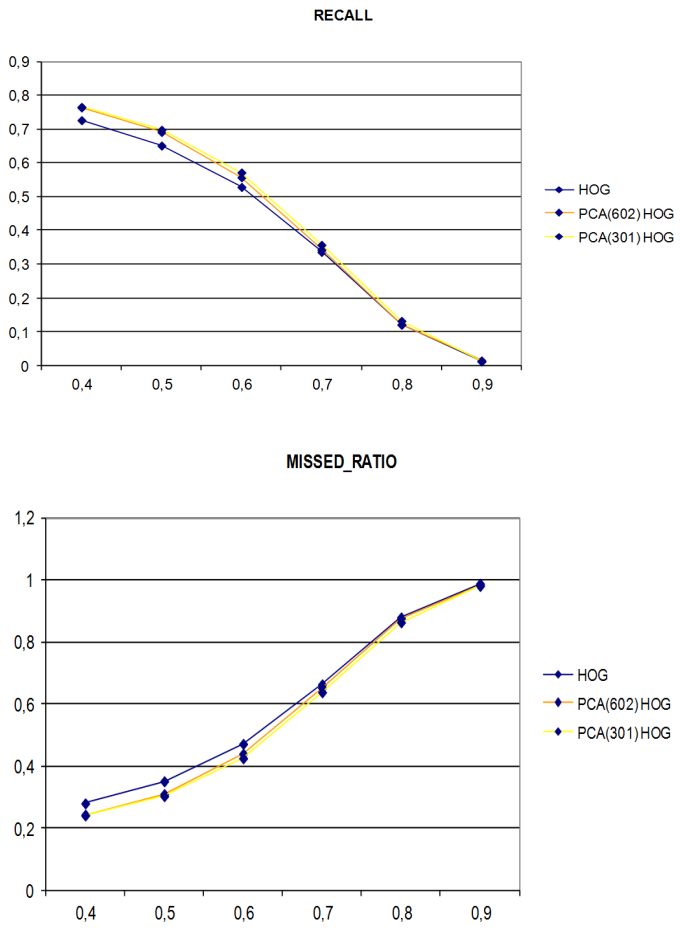

Fig. 11. The results of the PCA-HOG classification in comparison to pure HOG classification (percentage of the detected and undetected players vs. threshold).

together. The highest system performance does not depend on the maximum values of evaluation measures used (obtained for $T = 0.4$). To point the best combination of the input data that results in the highest system quality, outcomes for different values of T parameter should be taken into account. Therefore, area under curves of precision and missed ratio functions is computed. The performance of the whole system (with box aggregation and tracking enabled) can be described by the below expression (the higher value the better):

$$\delta = \sum_{n=4}^{9} \left(\alpha \cdot \mathrm{P}\left(\frac{n}{10}\right) + \beta \cdot \left(1 - \mathrm{M}\left(\frac{n}{10}\right)\right)\right),$$

where P and M are precision and missed ratio functions respectively, α and β are weights. Three scenarios are considered here:

I. $\alpha = \beta = \frac{1}{2}$ (both precision and missed ratio parameters are taken into account with the same weights),
II. $\alpha = 1$, $\beta = 0$ (only precision evaluation measure),
III. $\alpha = 0$, $\beta = 1$ (only missed ratio parameter).

The results are presented in the Fig. 11. PCA-HOG detector performs better than pure HOG detector with respect to the measure parameters.

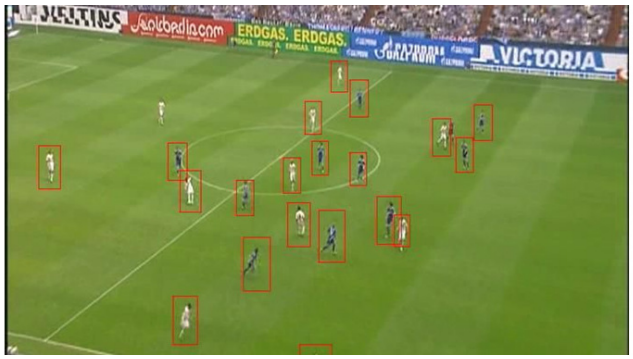

Fig. 12. A result of a player detection (HOG-PCA classification).

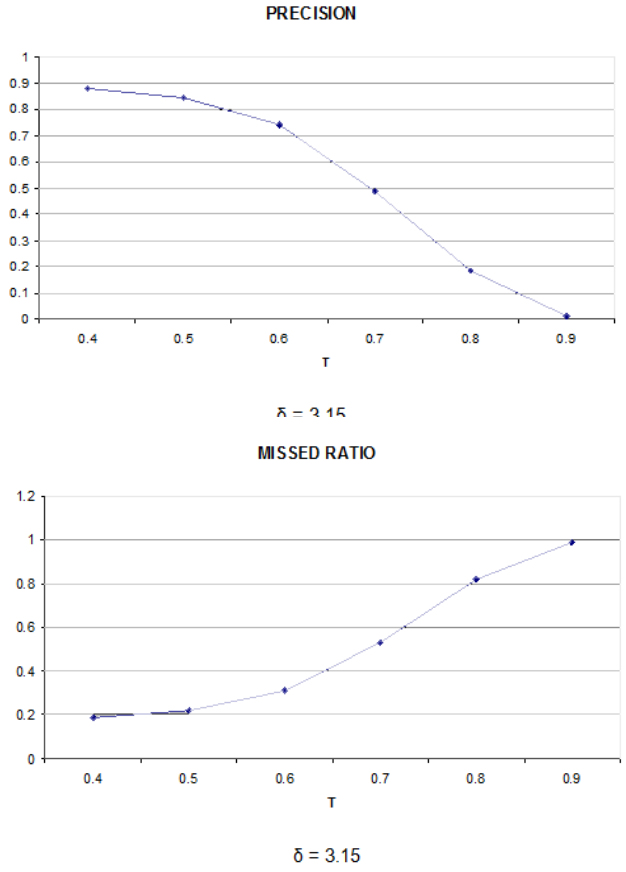

Fig. 13. The result of the best performance of the system.

The best performance of the system is received for the following conditions: a vertical pose, PCA-HOG, no of features: 225 and group threshold equals 2. Figure 13 demonstrates associated precision and missed ratio functions.

The results for PCA-HOG together with box aggregation and box tracking are very promising. The near 90% efficiency of the precission parameter is achieved. Of course, these results are achieved under the assuming that we have 40% of an overlap boxes. It can be notice that an algorithm of the aggregation has more impact on the result. The closer boxes to the players cause that the overall outcome is a little worse now because it is harder to fit the boxes, a small error results

Fig. 14. The four different type of shots.

in a shifting of good results toward lower T. If the better pre-segmenation algorithm will be used, the results with new aggregation of the boxes will raise the overall outcome.

If you look at the entire system globally and compared to the previous works, the results are much better. Curve of precision is raised for a much wide range of T. When analyzing sequences with and without occlusions the overall result for the sequence without occlusion is better, the δ parameter is higher for the sequences with occlusions.

VI. Shot Classification

A football video broadcast contains different types of shots that may be classified as: long (far), medium, close-up and out-of-field view. The first presents the global view of the field, the second usually displays the whole body of a player, the third one shows the above-waist view of a person and the last is associated with audience (Fig. 14) [16]. The author decided to classify all shots into the following four categories: close-up shot – e.g. player close-up view, there is no full player's body on screen. Medium shot – e.g. action close-up, we can see players of about a half of the screen height. Long shot – overview of playfield, camera is placed significantly above ground. Audience view – a view of audience, no part of playfield is visible and Unknown shot – every shot that does not fit any previous category or is not related to football (e.g. commercial block). Shots labeled as "Unknown" are omitted in any further processing.

The shot classification is a challenging problem in football sequences. The reason is high correlation of colors between different shot types that may result in insignificant histogram differences. Therefore several classification techniques was used to classify the shots: SVM – Support Vector Machine, Artificial Neural Network and LDA with k-nearest neighbor classifier.

A. Support Vector Machine

For each shot a feature vector is computed. Major part of that vector consists of features computed on shot frames. Each shot frame is divided by regular grid into $m \times n$ blocks. For each block, the features are computed independently. In the system, three types of block features: grass pixel ratio, edge

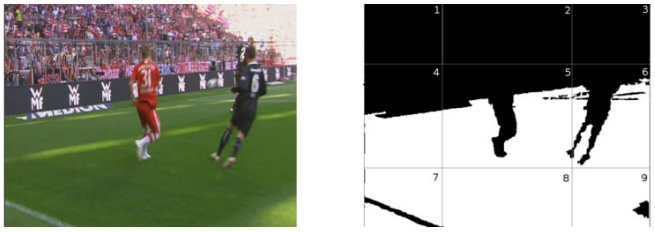

Fig. 15. The input frame and the mask as an output of the grass pixel ratio feature calculation.

pixel ratio and skin color pixel ratio were chosen and used. Each ratio feature value is defined as a number of feature pixels divided by the total number of pixels in block. The grass pixel classification (Fig. 15) is based on playfield detection algorithm described in previous section.

$$GR_n = \frac{\sum_{w=0}^{W_n} \sum_{h=0}^{H_n} B(w,h)}{H_n W_n},$$

where:

$$B(w,h) = \begin{cases} 1, & \text{if pixel } (w,h) \text{ belongs to a playfield} \\ 0, & \text{otherwise} \end{cases}$$

where subscript n denotes the number of block, (w,h) is a pair of pixel coordinates, H_n and W_n are height and width of n-th block.

Edge pixels are generated by applying Canny edge detector on luminance component of input frame (Fig. 16). Edge features are less sensitive to whether conditions, lighting, field color than color features. Moreover, edge distribution of a frame is associated with shot type. Therefore this information is used to improve shot classification performance. *Edge distribution* (ED) feature which is defined as the ratio of edge pixels in n-th block to the bock size:

$$ED_n = \frac{\sum_{w=0}^{W_n} \sum_{h=0}^{H_n} E(w,h)}{H_n \cdot W_n},$$

where:

$$E(w,h) = \begin{cases} 1, & \text{if pixel } (w,h) \text{ is an edge point} \\ 0, & \text{otherwise} \end{cases}$$

where subscript n denotes the number of blocks, (w,h) is a pair of pixel coordinates, H_n and W_n are height and width of n-th block.

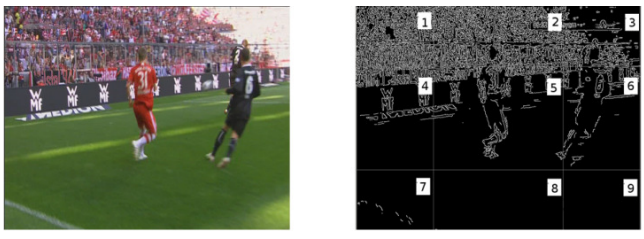

Fig. 16. The input frame and the mask as an output of the grass pixel ratio feature calculation.

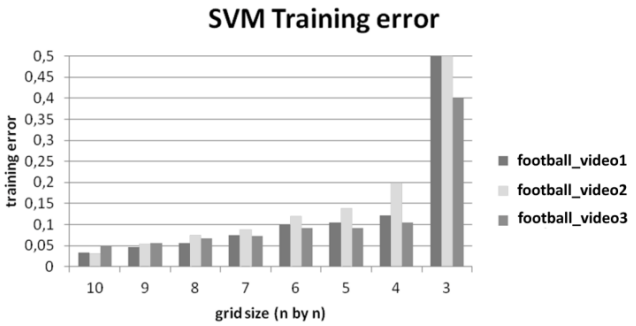

Fig. 17. The results for the SVM training.

Skin color detection is done in RGB color space according to [16]. It may not be very accurate but sufficient for shot type classification.

Features described above are computed for each frame of a single shot. The shot feature vector consists of the average of features computed for all shot frames. To eliminate the problems caused by different transitions at the beginning and end of a shot, frames which contain transitions are skipped. There is one additional feature – shot length expressed in seconds. To summarize, for a grid of $m \times n$ blocks we have $(m * n) * 3$ block features plus shot length.

In case of unacceptably low classification precision it is possible to add some additional block features. Football broadcast video has a number of colors the meaning of which is significant for the content. They are called semantic colors [17]. It is possible to compute additional pixel ratio features based on these semantic colors for each block. After adding semantic color ratio features, the length of feature vector can be increased by $(m * n)$ features multiplied by the number of semantic colors used. Unfortunately, a problem with the semantic color may occur. It may be a specific color for a single sequence; therefore a classifier trained in using colors and other data from one sequence may not work properly on another.

In the experiments four SVM classifiers was trained to detect single shot class. Each SVM is trained to distinguish between a particular shot class and the rest of shot classes. Then during classification each feature vector is subjected to all SVMs. Sometimes more than one SVM can give positive response. In such case a shot class is determined by the SVM which gives maximum response meaning that feature vector has the farthest location from the SVM's hyperplane.

In the shot classification experiment, three shot sets: football_video1, football_video2 and football_video3 were used. Each shot in set was labeled manually as one of four possible types. Shots that do not fall in any of these categories were labeled as unknown type. For each shot a number of feature sets was generated by software using the original video sequence. Each feature set was generated using different grid setting to find its optimal size. The grid sizes vary from 3×3 to 10×10 blocks.

The goal of the first experiment was to find optimal grid size that minimizes the classifier training error. For each grid size

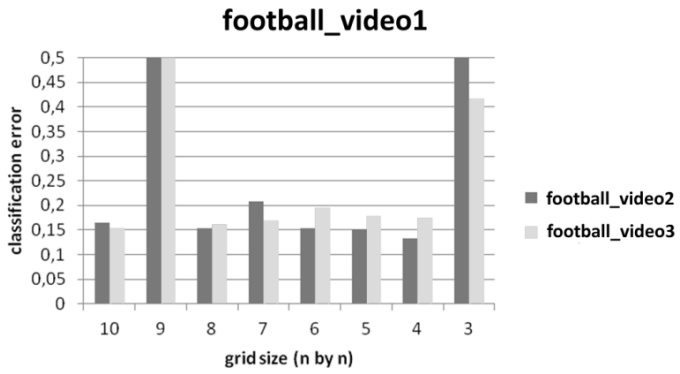

Fig. 18. The results for the SVM classification.

a set of SVM classifiers was trained using feature set generated for that size and ground-truth shot type classification. Next, the same sequence used in training was classified to provide necessary data. Results show (Fig. 17), that a grid larger than 4×4 provides sufficient feature set to keep classification error reasonably low. Further, grid size increase does not provide any meaningful gain.

The second experiment goal was to test SVMs performance on a sequence different than the training sequence (real classification situation). In this experiment, single SVM set trained on each one of three sequences and used it to classify shots from the other two sequences. Results (Fig. 18) show, that a grid larger than 4×4 is sufficient. Increasing grid size allows classifiers to fit better into training data, but when it comes to classify other data set, classification error increases.

The parameter *classification_error* is defined as follows: *classification_error = the_number_of_false_detected_shots / all_defined_shots_in_the_sequence*.

B. Artificial Neural Network

The support vector machine classifier may solve only a linear separable classification problem. In order to classify more complex data sets we need a different solution. As the second classification method a multi-layer perceptron neural network was chosen. The tested network has N inputs and M outputs where N equals the number of shot features and M the number of possible shot types (4 in the tests). The number of hidden

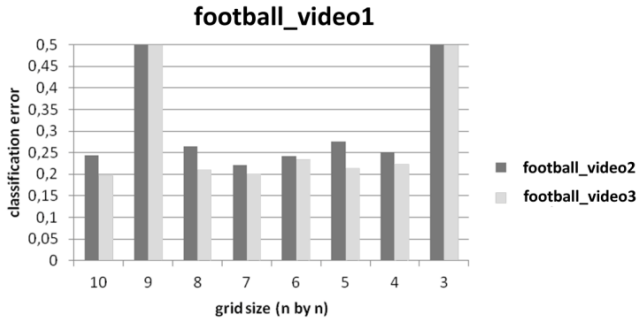

Fig. 19. The results for the Artificial Neural Network.

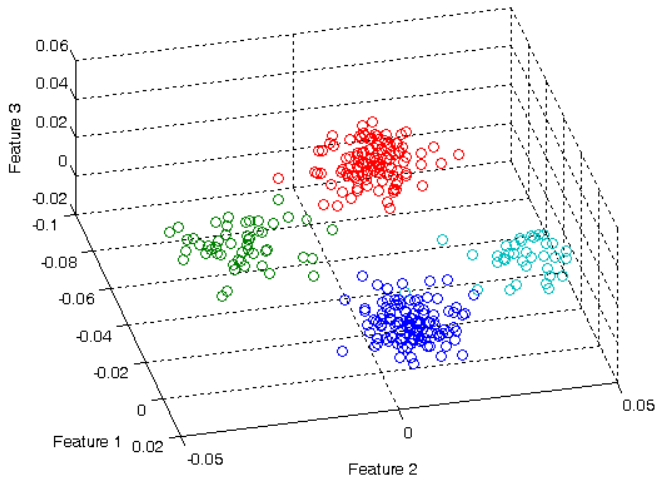

Fig. 20. A projection of an original feature set onto three dimensional space (a different colour for the different type of the shot).

layers may be adjusted. In experiment was set it to one. The major problem with artificial neural network is its training. Error back propagation algorithm is used. This algorithm is known for being stuck sometimes in a local minimum of error function. As the experiments showed, this happens quite often. Despite that flaw, neural network performance was not significantly worse than SVM (Fig. 19).

C. LDA with k-nearest Neighbor Classifier

In the third experiment Linear Discriminant Analysis (LDA) was used. LDA reduces the dimensionality of feature vector and to find some subspace where data are easily separated. LDA that belongs to the supervised techniques projects data onto lower dimensional space maximizing the distance between the means of classes and minimizing the variance within each class. An analyze of pre experiment projection of an original feature set onto three dimensional space performs well (Fig. 20). It is clear that LDA with respect to classes discrimination and the projected data can be easily separated.

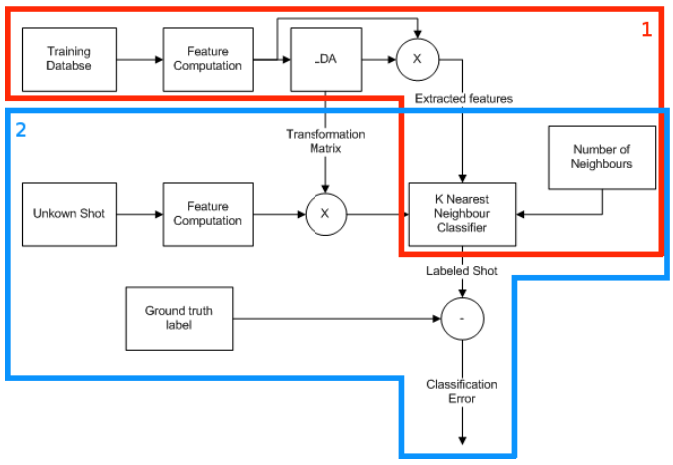

Fig. 21. LDA-based shot training and classification system.

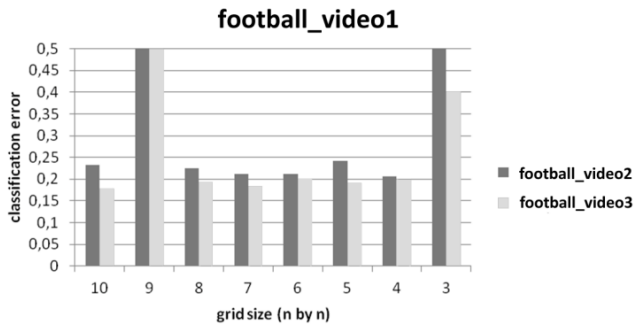

Fig. 22. The results for the LDA classification.

The whole framework is presented in Fig. 21. The training stage is marked with the red rectangle and contains feature computation of training data, dimensionality reduction with LDA and kNN classifier training. Finally, classification of unknown shots is performed (the blue rectangle). The result of the classification is presented in Fig. 22.

Experiments in every classification system were done for each shot sequences. Results are very closed to the result of the presented set of the shots. The obtained results demonstrate that the average classification error is higher than 15% in each of the classification system and may vary depending on settings of the classifier. The classification error can be lower in the case of used of more training data. The experiments show that the Artificial Neural Network is the most worse technique in the classification process.

VII. Conclusion

In the paper, a novel segmentation system for football video broadcast is proposed. The proposed system is a complex solution which incorporates several techniques which are used to detect players, playfield and shots. These methods were selected based on their potential robustness in case of great inconstancy of weather, lighting and quality of the input video sequences. Results show that proposed solution seems to achieve high objective and subjective notes in terms of precise location of the detected objects, however, the number of missed objects still needs to be decreased. Consequently, there are some works deserving further research in the proposed approach.

References

[1] M. Muja and D. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Classification," in *International Conference on Computer Vision Theory and Application (VISAPP 2009)*, Lizbona, 2009.

[2] S. Hong, W. Yueshu, C. Wencheng, and Z. Jinxia, "Image Retrieval Based on MPEG-7 Dominant Color Descriptor," in *The 9th International Conference for Young Computer Scientists (ICYCS 2008)*, 2008, pp. 753–757.

[3] L. Ying, L. Guizhong, and Q. Xueming, "Ball and Field Line Detection for Placed Kick Refinement," in *WRI Global Congress on Intelligent Systems (GCIS '09)*, 2009, pp. 404–407, vol. 4.

[4] R. Ren and J. M. Jose, "Football Video Segmentation Based on Video Production Strategy," *Lecture Notes in Computer Science*, vol. 3408, pp. 433–446, 2005.

[5] Q. Li, L. Zhang, J. You, D. Zhang, and P. Bhattacharya, "Dark line detection with line width extraction," in *International Conference on Image Processing (ICIP 2008)*, 2008, pp. 621–624.

[6] X. Yu, H. C. Lai, S. X. F. Liu, and H. W. Leong, "A gridding Hough transform for detecting the straight lines in sports video," in *International Conference on Multimedia and Expo (ICME 2005)*, 2005, pp. 518–521.

[7] T. T. Nguyen, X. D. Pham, and J. W. Jeon, "An improvement of the Standard Hough Transform to detect line segments," in *IEEE International Conference on Industrial Technology (ICIT 2008)*, 2008, pp. 573–585.

[8] G. Jiang, X. Ke, S. Du, and J. Chen, "A straight line detection based on randomized method," in *9th International Conference on Signal Processing (ICSP 2008)*, 2008, pp. 1149–1152.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005, pp. 886–893, vol. 1.

[10] S. Maćkowiak, J. Konieczny, M. Kurc, and P. Maćkowiak, "A complex system for football player detection in broadcasted video," in *International Conference on Signals and Electronic Systems (ICSES 2010)*, 7–10 September 2010, pp. 119–122.

[11] S. Makowiak and J. Konieczny, "Player Extraction in Sports Video Sequences," in *International Conference on Systems, Signals and Image Processing (IWSSIP 2012)*, Vienna, Austria, 11–13 April 2012, pp. 423–426.

[12] D. Farin, S. Krabbe, P. de With, and W. Effelsberg, "Robust Camera Calibration for Sport Videos using Court Models," *Proceedings of SPIE*, vol. 5307, pp. 80–91, 2004.

[13] M. Grabner, H. Grabner, and H. Bischof, "Fast Aproximated SIFT," in *Asian Conference on Computer Vision*, Washington, 1999.

[14] D. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, 2004.

[15] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[16] K. Nallaperumal, S. Ravi, C. N. K. Babu, R. K. Selvakumar, A. L. Fred, C. Seldev, and S. S. Vinsley, "Skin Detection Using Color Pixel Classification with Application to Face Detection: A Comparative Study," *Proceedings of International Conference on Computational Intelligence and Multimedia Applications, 2007*, vol. 3, pp. 436–441, 13-15 December 2007.

[17] Z. Niu, X. Gao, D. Tao, and X. Li, "Semantic Video Shot Segmentation Based on Color Ratio Feature and SVM," in *2008 International Conference on Cyberworlds*, 22–24 September 2008, pp. 157–162.

[18] N. Oezay and B. Sankur, "Automatic TV Logo Detection and Classification in Broadcast Videos," in *The 2009 European Signal Processing Conference (EUSIPCO 2009)*, Glasgow, Scotland, August 2009, pp. 839–843.