IEICE TRANS. INF. & SYST., VOL.E98–D, NO.8 AUGUST 2015

PAPER

Automatic Soccer Player Tracking in Single Camera with Robust Occlusion Handling Using Attribute Matching

Houari SABIRIN[†a)], ***Nonmember***, **Hiroshi SANKOH**[†], ***Member***, ***and*** **Sei NAITO**[†], ***Senior Member***

SUMMARY This paper presents an automatic method to track soccer players in soccer video recorded from a single camera where the occurrence of pan-tilt-zoom can take place. The automatic object tracking is intended to support texture extraction in a free viewpoint video authoring application for soccer video. To ensure that the identity of the tracked object can be correctly obtained, background segmentation is performed and automatically removes commercial billboards whenever it overlaps with the soccer player. Next, object tracking is performed by an attribute matching algorithm for all objects in the temporal domain to find and maintain the correlation of the detected objects. The attribute matching process finds the best match between two objects in different frames according to their pre-determined attributes: position, size, dominant color and motion information. Utilizing these attributes, the experimental results show that the tracking process can handle occlusion problems such as occlusion involving more than three objects and occluded objects with similar color and moving direction, as well as correctly identify objects in the presence of camera movements.

key words: *free viewpoint, attribute matching, automatic object tracking, soccer video*

1. Introduction

Free viewpoint applications provide an innovative user experience for enjoying a soccer match. The applications enable users to freely select which viewpoint has the most suitable angle for a certain event in a soccer match (e.g. soccer spectators tend to see a goal from as many different angles as possible) [1]. To create a video that enables such experience, a free viewpoint authoring application has been developed to render visual texture of the object of interest (i.e. the players, goalkeepers, and referee within the soccer field area) using a billboard model [2]–[4]. The authoring application extracts the texture of the soccer players from the original video and then merges the extracted pixels into a synthesized 3D environment that can be freely navigated by the users.

The authoring application utilizes the method proposed in [5]–[7] to perform texture extraction in order to reconstruct the billboard models. The texture extraction method comprises several steps, in which the most important parts are: projection of borders of the soccer field to determine precise positions of the objects and the area of the field; background segmentation to segment the objects from the field and acquire only the region of the objects; and object tracking to approximate the location and recognize the movement of the objects during the sequence.

Dynamic camera calibration [6] is utilized in the application for projection of the soccer field and object position estimation from multiple cameras to extract the object textures. Background segmentation is performed by observing temporal color similarity to remove the field area and retain the pixels of the objects. And finally, automatic object tracking is performed by observing the correlations among the extracted silhouette regions in multiple cameras. As a result, robust texture extraction including occlusion problems, which frequently occur in soccer video, can be performed.

However, the automatic object tracking method proposed in [5] requires specific conditions for the input video. Firstly, the input shall be acquired from multiple cameras to ensure accurate object tracking. Secondly, those cameras must be static in order to achieve high performance in automatic object segmentation, which in turn significantly affects the object tracking result. In most cases, however, these two conditions cannot be met. For example, when the free viewpoint video needs to be generated from a soccer match acquired from TV broadcasting (such as game analysis of an Olympic or World Cup match), only one stream of video is available and it is most likely to have pan-tilt-zoom occurrences. In this case, the method in [5] may not be sufficient, especially when handling dynamic background and changes in objects' texture due to camera movements. Therefore, we attempted to improve the method in [5] to be able to extract the texture of objects from a single, moving camera. Background segmentation from a dynamic scene is improved by further refinement using a watershed algorithm and Hough transform. Automatic object tracking is improved by evaluating the spatial and temporal correlation among detected moving objects in two frames. The evaluation is conducted by finding the similarities between two sets of object attributes.

This paper is structured as follows: we first review related work in Sect. 2; Sect. 3 provides details of the proposed method; the experimental results are presented in Sect. 4 and Sect. 5 concludes this paper.

2. Related Work

There are some works similar to our method in automatic object detection and tracking for soccer video that has the capability to correctly and uniquely identify all objects of interest and to correctly track those objects when they are involved in occlusion. In [8] a rather simple method to iden-

Manuscript received September 19, 2014.
Manuscript revised March 18, 2015.
Manuscript publicized May 14, 2015.

[†]The authors are with KDDI R&D Laboratories, Inc., Fujimino-shi, 356–8502 Japan.
a) E-mail: ho-sabirin@kddilabs.jp

DOI: 10.1587/transinf.2014EDP7313

tify soccer players and the ball is proposed. In the proposed method, the Sobel gradient method is utilized to extract the players and ball from the field, followed by a line detection algorithm to remove the lines in the field. In [12], Euclidean distance transform is utilized to extract players from the soccer field and refine the segmentation using skeleton pruning of the segmented objects. The localization method proposed in [13] can reduce the complexity of object detection and tracking by providing an individual tracking module specific to the character of each of the detected objects. While providing satisfactory segmentation and object detection, the methods proposed in [8], [12] and [13] all lack unique identification for the detected objects.

Graph-based theory is also one technique that has been utilized in tracking soccer players as in [10], [11] and [14]. In [10] a k-means clustering method based on the dominant color is utilized to segment the soccer players from the field and other regions. The segmentation process is then refined using a multilayer perceptron (MLP) neural network classifier and then the tracking is performed using the method proposed in [11]. The method proposed in [14] uses graph structure for a 3D particle filter likelihood detection that enabled unique identification of the detected object. However, the method has a problem with complex occlusion such as when there are two objects from the same team (i.e. similar color), or in crowded occlusion in the penalty area that involves more than two players.

Furthermore, the detection and tracking method based on the particle filter technique is also proposed in [15], [16] and [17]. Using a particle filter, unique identification can also be produced as in [14]. However, not all player objects are detected and identified by those methods. The method in [17] utilizes a particle filter from a synthesized template image to provide more precise likelihood evaluation of the particles. Nevertheless, the identification in those proposed methods is only to distinguish different teams instead of individual players.

Considering the shape of a soccer field and the field lines, Hough transform is also noticeably utilized in soccer player detection and tracking. The methods proposed in [9], [20], [23] and [24] utilize Hough transform for field area detection and subtraction from the foreground (i.e. soccer players, the referee, and the ball) followed by different tracking approaches. A histogram of oriented gradient (HOG) descriptors and linear support vector machine (SVM) classification has been proposed in [9] to train the football player template to determine player and non-player objects. The tracking of the recognized player objects in [9] is then performed by evaluating each object's cost computed based on their position and size. A localized temporal spatio-velocity (VCT) transform based on Hough transform is proposed in [20] utilizing gray-scale and color features of the object's mass by computing velocity and position of the detected object. In [23] and [24] the data training method is utilized to enhance the accuracy of the segmentation and tracking. The methods focused on the soccer video taken from broadcast video, especially for a wide camera angle. The tracking algorithm is performed by observing the motion of the players during a specified time interval. To achieve optimal computation with less complexity, the observation time interval is limited. Therefore, a long term occlusion cannot be handled. Moreover, as mentioned by the authors, the method still leads to failure in the case of camera motion and video blur.

We also review soccer player detection and tracking methods that utilize the Bayesian classifier and Kalman filter. A template-based approach is proposed in [18] for localizing soccer players from multiple cameras. By acquiring more information from different camera positions, the high accuracy of player tracking is ensured in the method. Moreover, the utilization of hue and saturation information in classifying different teams provides a good clue in our method. A multiple object tracking (MOT) algorithm is proposed in [19] by using the dual-mode two-way Bayesian inference approach to handle individual object tracking as well as multiple occlusion events. However, only the identification to distinguish different teams is produced.

In [21], a so-called *tracklets* constructed using a Kalman filter is defined as the representation of the trajectory of the detected object and is used to recognize each of the objects by utilizing player number recognition. Those aforementioned methods perform well in detection and uniquely identify objects. A sophisticated method has been developed in [22] for basketball video by utilizing MSER, SIFT, and RGB color histograms altogether that is more robust compared to the other reviewed methods. However, the detection of objects during occlusion is not fully obtained, especially in [20] possibly when the occlusion occurred for objects with the same color; and in [21] due to incorrect player number recognition.

3. Proposed Method

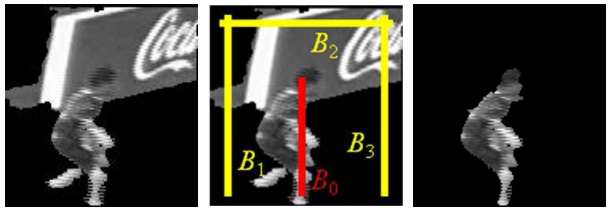
We proposed a novel method to enable automatic object tracking from a single, moving camera. The method assigns a set of attributes and adaptively updates its values to maintain the integrity of the attributes in the temporal domain. In addition, unsupervised region marking for commercial board removal is also introduced. By doing so, we can achieve satisfactory performance in object tracking and handle the issues that arose in the related work shown in Sect. 2.

3.1 Segmentation and Feature Extraction

To extract the objects of interest from a soccer video, a semi-automatic segmentation process is performed. The method was proposed in [6] and can be described briefly as follows. In the first stage, the green region of the soccer field needs to be manually marked at the first frame. Here, the level of intensity and saturation that matches the field and the field lines can be adjusted to ensure that the segmentation process can distinguish the color of the field and field lines by the color of the jersey (nevertheless, if their jersey color is

(a) (b) (c)

Fig. 1 Segmenting (a) original image data into (b) object masks. (c) The initial ROI of the object mask produced by the feature extraction process. Images are cropped from 4K resolution for clear presentation.



(a) (b) (c)

Fig. 2 Board removal step: (a) original object mask, (b) generated markings are drawn, and (c) the final segmented object

also green, the segmentation method may fail). Next, an automatic process based on a graph-cut algorithm is performed to remove the field and lines area from all frames in the video sequences.

In the second stage, the area outside the field (the "outer area") needs to be removed, so that the remaining segmentation would result only in the pixels of the objects of interest. To determine the outer area, the points in a soccer field must be marked manually in the first frame. Employing the standardized measurement of a soccer field and utilizing a homographic matrix, at least only six points in the soccer field need to be marked to determine the area of the field [6]. Based on these predetermined points, the corresponding points in the next frames in the sequence can be acquired using SURF descriptors and the outer area to be removed can be automatically determined for all frames.

After removing the green area of the field, the white lines and the outer area, the remaining pixels would be the pixels of the objects of interest. Throughout this paper, such pixels are called "object masks." Figure 1 shows an example of the segmentation result of original image data into object masks.

Following background subtraction, feature extraction is performed to determine the initial ROI of each object mask that represents their initial positions and sizes as shown in Fig. 1 (c). The ROI is determined as a rectangle that tightly encapsulated an object mask. Throughout this paper, an object mask is associated with an initial ROI. Note that when occlusion occurs, as shown by the ROI labeled as "12" in Fig. 1 (c), an object mask may contain the pixels of the combined objects. The separation of ROIs for those occluded objects is performed by the occlusion handling process and is described in Sect. 3.4.

In the feature extraction process, each object mask is assigned a set of information called object attributes that are determined from the mask's pixels. The assigned attributes are: identity, position, size, dominant color, and motion information. Let $\mathbf{C}^f = \left\{C_0^f, \ldots, C_N^f\right\}$ be the set of all object masks resulted from the segmentation in frame f, where N is the number of object masks, for the i-th object we define the attributed object as

$$C_i^f = [X \quad \mathbf{L} \quad \mathbf{S} \quad \mathbf{Y} \quad \mathbf{M}]^T \tag{1}$$

where the attributes are determined as follows. The identity of the object $X = \{0, \ldots, n | n < N\}$ denotes unique identification, therefore duplicated identity is not allowed. The position of the object, relative to the top-left position of the frame, is denoted by $\boldsymbol{L} = \{x, y | 0 \le x \le W, 0 \le y \le H\}$ where W and H are the width and the height of the frame, respectively. The size of the object is denoted by $\boldsymbol{S} = \{w, h | 0 \le w \le W, 0 \le h \le H\}$ that represents the width and height of a rectangle that encapsulates the non-zero pixels of the object mask, hence its ROI is defined.

The color attribute of the object is denoted by $\boldsymbol{Y} = \{hue, sat | 0 \le hue \le 1, 0 \le sat \le 1\}$ where it contains the values of hue and saturation components of the dominant color of the object. We acquired the dominant color of an object by constructing 30 bins and 32 bins color histograms for the hue and saturation channel, respectively. The usage of HSV color space was considered due to its advantage in distinguishing different colors as verified in [25].

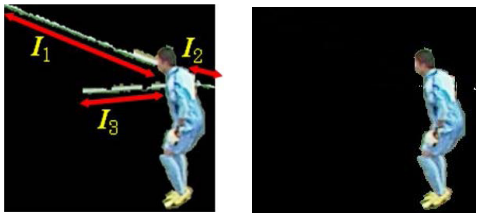
Finally, the motion attribute $\boldsymbol{M} = \{mv_x, mv_y\}$ denotes the motion vector of the object in the horizontal and vertical direction, respectively. The motion vector is obtained by calculating the optical flow of the hue channel of the centroid of the object mask using the KL method [26].

3.2 Segmentation Refinement

The aforementioned segmentation method needs to be followed by manual operation to produce perfect object segmentation if the input data is produced by a single, moving camera. Due to the dynamic background produced by such camera, performing only the semi-automatic process may cause imperfect object segmentation from the field area and the field lines, especially for the objects that are located at the side of the field.

An example of imperfect segmentation for objects located at the side of the field is shown in Fig. 2 (a). In this example, an object mask of the goalkeeper is overlapped with a commercial billboard and the billboard cannot be perfectly removed. This problem occurred due to the segmentation process removing only the pre-determined green color of the field area. Therefore, when the object masks are overlapped with the outer area, some parts of the outer area having color other than green cannot be subtracted.

We utilize the watershed algorithm [27] to remove the commercial board due to its good performance given that the regions to be segmented are already known. Here, two regions are determined: the object region that will be retained and the commercial billboard region that will be removed.



(a) (b)

Fig. 3 Removing lines from (a) imperfect segmented ROI and (b) the removal result.

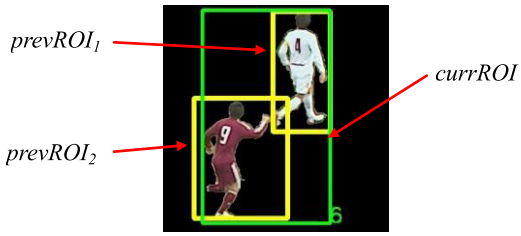


Fig. 4 Occlusion event is confirmed when the ROI of the current object mask is overlapped with the ROIs from the previous frame's object masks.

To avoid time-consuming manual regions marking for all object masks, the marking should be performed automatically.

The initial marking for the object region is drawn as a vertical line located in the approximate position of the object shown as the B_0 line in Fig. 2 (b). To approximate the position of an object within the object mask, the value of the sum of the square difference of RGB pixel values between an object template and all non-zero pixels of the object mask is calculated. An object template is the object mask of another object selected from the middle of the frame, where its possibility of overlapping with any commercial billboards is low. Let x_{min}, y_{min} be the position of the pixel in an object mask where the difference of pixel values between the object mask and the object template yields the smallest value, a vertical line is then drawn from position $(x_{min} + w_{temp}/2, y_{min})$ to position $(x_{min}+w_{temp}/2, y_{min}+h_{temp})$. Here w_{temp} and h_{temp} denote the width and the height of the template object mask, respectively.

With an assumption that a commercial billboard is most likely to be located around the border of an object mask, setting the initial markings for the board can be made simpler. The markings can be made as straight lines drawn five pixels away from the left, top, and right borders of the object mask to be refined, shown as B_1, B_2, and B_3 lines, respectively in Fig. 2 (b). The watershed algorithm segmentation is then performed for the marked object mask and the object region is saved as the segmented object mask. Figure 2 (c) shows the segmentation results.

An example of imperfect segmentation that cannot remove the field lines is shown in Fig. 3 (a) where three lines remain from the segmentation. The removal of those lines is performed automatically and the procedure is relatively simple. To remove a line, a simple line detection method based on Hough line transform is utilized. By observing neighboring pixels of the detected lines for non-zero pixels, the correct location of a true line can be found within an object mask. Let $\boldsymbol{I}_d$ be the set of pixels representing the d-th line detected by Hough transform within the object mask, the line is removed by simply setting all pixels in $\boldsymbol{I}_d$ as equal to zero. In Fig. 3 (b) lines $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ and $\boldsymbol{I}_3$ are removed by setting their pixel values to zero.

3.3 Temporal Correlation via Attribute Matching

After object attributes are set, initial object tracking is performed by finding temporal correlation between two objects in two consecutive frames. An object in the current frame is said to be correlated to an object in the previous frame if the values of their attributes are similar. Hence we need to find

$$j_{best} = \arg\min_i \left(D_{LY}\left(C_j^{f_t}, C_i^f\right)\right) \tag{2}$$

where C_i^f and $C_j^{f_t}$ are the i-th object in frame f and the j-th object in frame $f_t < f$, respectively. D_{LY} denotes the similarity measurement of $\boldsymbol{L}$ and $\boldsymbol{Y}$ attributes between the j-th object in frame f_t and the i-th object in frame f, calculated as

$$D_{LY} = \alpha \|D_L\| + \beta \|D_Y\| \tag{3}$$

where $\|\cdot\|$ denotes the Euclidean distance between the attributes of C_i^f and $C_j^{f_t}$. The weighting coefficients α and β are defined due to different scales between position and color attribute values. From experiments, setting $\alpha = 2$ and $\beta = 0.5$ yielded the best results. If (2) is satisfied, then the identity attribute of C_i^f is assigned to be the same as the identity of $C_j^{f_t}$ as indicated by j_{best}.

In cases when any objects are missing or appearing in a frame, their temporal correlations are also calculated. When an object is missing from a frame (i.e. its object mask is available in the previous frame but cannot be found in the current frame), its attribute information will be retained in the memory. If the last known position of the missing object is just near the border of the frame, we assume that the object is leaving a frame and its information will be removed. To recognize whether a new object had just entered a frame, its position attribute most likely will not be correlated to another already matched object. In this case, the object will be assigned a new identity.

3.4 Occlusion Handling

When an occlusion occurs, an object mask contains actual pixels from the merged objects as illustrated in Fig. 4. To validate the occlusion, the ROI of the object mask in the current frame (*currROI*) is correlated against the ROIs (*prevROIs*) of the object masks at the same position in the previous frame. To confirm that two objects or more are involved in an occlusion, we determine that the overlapping area between the *currROI* and *prevROIs* should be more than 60% of the total area of each of the *prevROIs*. If the threshold

is satisfied, then an occlusion between the objects is confirmed.

When overlaps are observed in frame f-t, new objects are temporarily constructed in the current frame. These temporary objects are projected from the objects in frame f-t that overlap with each other. The set of temporary objects at a period of occlusion is defined as $\dot{\mathbf{C}}^{f_{occ}} = \left\{\dot{C}_0, \ldots, \dot{C}_k\right\}$, where k is the number of overlapping objects in frame f-t and $f_{occ} > f$-t is a frame within a period of occlusion such that $\dot{\mathbf{C}}^{f_{occ}} \subset \mathbf{C}^{f-t}$ where $\mathbf{C}^{f-t}$ is the set of all objects in frame f-t. To ensure accurate approximation of the ROIs of the overlapping object, the position of each object is updated with its motion information. Thus the position attribute of the i-th overlapping object at frame f_{occ} is updated as

$$\dot{\mathbf{L}}_i^{f_{occ}} \leftarrow \dot{\mathbf{L}}_i^{f_{occ}} + \mathbf{M}_i^{f-t}. \tag{4}$$

To approximate the position and size of the ROI to match the actual object position during occlusion, the X and $\boldsymbol{S}$ attributes of the temporary objects are updated based on the corresponding X and $\boldsymbol{S}$ attributes of the objects in reference frame f-t. This can be accomplished by matching the identity of the temporary objects with the identity of the objects in frame f-t. In a similar approach to (2), finding the identity of the i-th object during occlusion based on the identity of the j-th object in frame f-t is to satisfy

$$j_{best} = \arg\min_i \left(D_{LY}\left(C_j^{f-t}, \dot{C}_i^{f_{occ}}\right)\right). \tag{5}$$

At the end of the occlusion period, indicated when an object has no overlapping ROI with other objects, the similarity matching is performed once again between frame f-t and the current frame using (2) to find the corresponding match of the object after occlusion.

After all occlusions in a frame, if any, have been managed, the last step is to assign final ROI to the object masks. This means the position and size of initial ROI of an object mask are replaced with the position and size attribute of the object according to the identity of the object mask that were updated via attribute matching and occlusion handling. The information from final ROI is then used for temporal correlation and occlusion handling in the subsequent frames.

Finally, the actual texture of objects can be extracted from the original frame based on the position determined by the final identification process. These object textures will then be utilized for the synthesized free viewpoint video.

4. Experimental Results

We tested our method in several test sequences of soccer matches in 4K and HD resolutions with frame rates of 30fps. The test sequences are focused on the scenes near the goal, which are commonly used in the highlights of a soccer match video. Since such highlight videos usually have a short time period, our test sequences mostly last for around 170 to 200 frames. Thus, to evaluate the robustness of the proposed method, we use seven test sequences. For clear presentations, all screenshots from the test sequences are cropped from their original image to show only the area of the point of interest. Note that the results from test sequences 3 to 7 are shown in silhouette due to copyright issue with the original video.

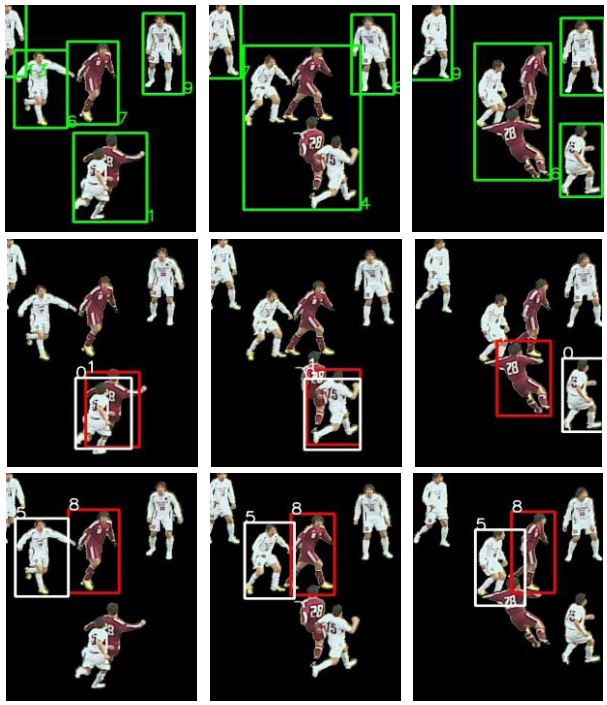


Fig. 5 Handling complex occlusion in test sequence 1. The first row shows the original ROI of frame 134, 140, and 147, respectively. The second row shows the resulting ROI for object 0 and object 1 from the same frames. The third row shows the corresponding results for object 5 and object 8.

Figure 5 presents the result of the proposed method in handling complex occlusion that occurred in test sequence 1. As depicted by the initial ROIs shown in the first row of Fig. 5, there are at least five to six players from both teams involved in more than two occlusion events. The final ROIs generated for the occluded objects are shown separately in the middle row and bottom row for clear presentation. While the ROIs of object 5 and object 8 are relatively easy to separate due to their prior information in frame 134, separating the ROIs of object 0 and object 1 is a challenge due to the closely located position of both objects. However, since the dominant color and the moving direction of both objects can be distinguished, their ROIs during occlusion can also be approximated separately.

The results shown in Fig. 6 present the capability of the proposed method to handle a blurred object mask due to very fast camera movement in test sequence 3. Figure 6 shows screen captures from two spatial observation windows that represent two main occurrences in the test sequence 3. Observation window A represents the occurrence when the camera moves fast toward the right side of the field and at the same time performs zooming in. The results in every two frames from frame 90 to frame 96 are shown at

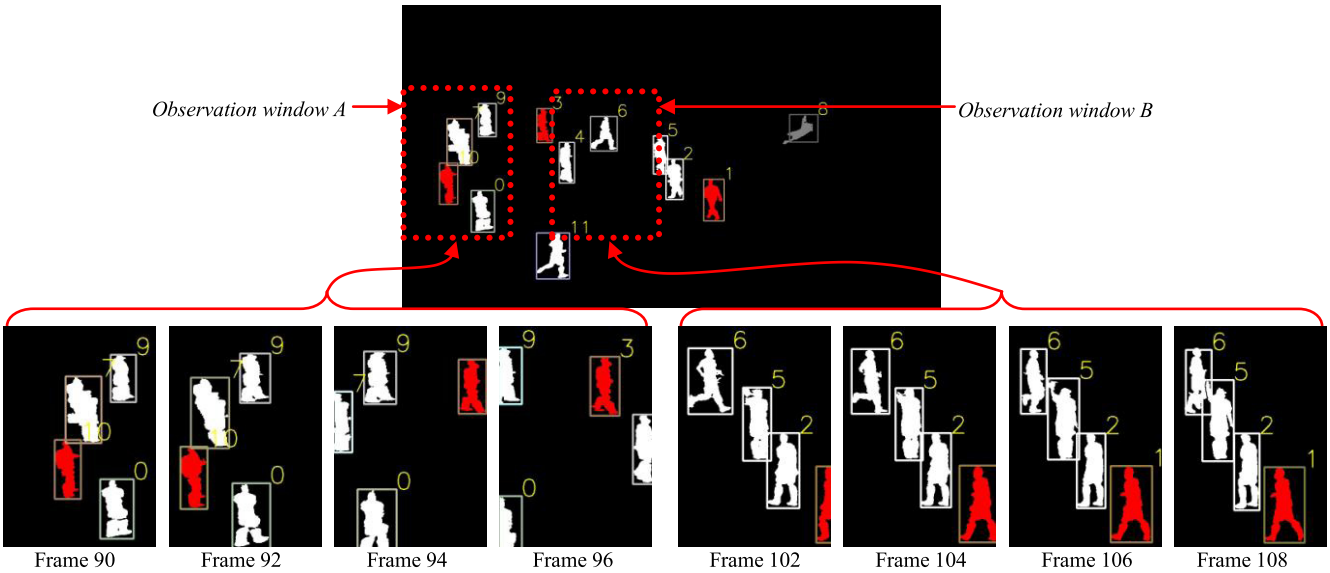


Fig. 6 The performance of the proposed method to track objects in the event of fast camera movement that causes the object masks to become blurry in test sequence 3. The results from *observation window A* are shown at the bottom left side for frame 90 to frame 96. The results from *observation window B* are shown at the bottom right side for frame 102 to frame 108.

the bottom left side of Fig. 6. Observation window *B* represents the occurrence just after the event that was shown in observation window *A*. At this point, the camera movement and zooming in are not as fast as before whilst enlarging the size of the ROIs of the objects. As shown at the bottom right side of Fig. 6 in every two frames from frame 102 to frame 108, despite the increment of ROI sizes, the objects can be correctly identified even in the presence of occlusion. In the events of camera pan, tilt and/or zoom, it is shown that the proposed method can reliably handle the camera movements. The main issue which usually occurs in intense camera movement is when the objects become blurred in horizontal direction (the camera would commonly moves in horizontal direction towards the location of goal whilst zoom-in into the area where the players with ball are present, when the attacking team is about to score a goal, for example). While it may change the size of the objects (especially the width of the objects), the dominant colors for different soccer teams and their relative positions from the previous frames can still be comprehended. Since the proposed method is taking into account the relations of objects attribute in temporal domain, the current tracking results can be achieved.

We compare the performance of the proposed occlusion handling method with the tracking method in [5] when only one camera is utilized and camera movement is presented. We also compare the proposed method with the Camshift algorithm [28] that sufficiently and efficiently tracks objects based on a color histogram even in the event of camera movement and occlusion. However, it does not perfectly track objects when the occlusion occurs for objects with a similar color. Figure 7 shows a case of simple occlusion in sequence 3 involving three objects. Figure 8 shows the results from the existing method and Camshift algorithm

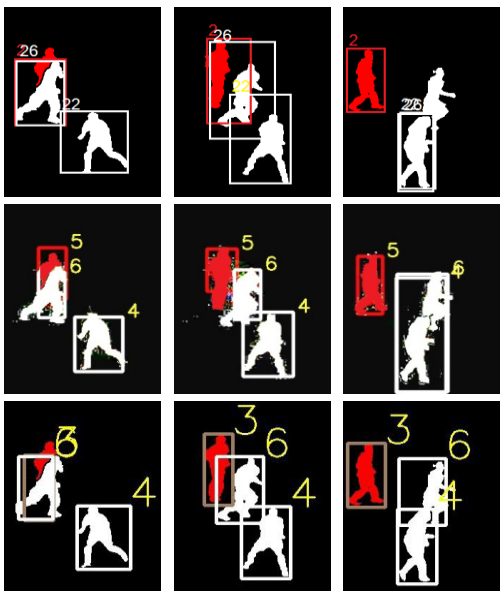


Fig. 7 Comparison of occlusion handling for test sequence 3 between the existing method (above row), Camshift method (middle row), and the proposed method (bottom row) is shown for frame 154, 160 and 168, respectively.

for the same sequence in Fig. 5. In this case, Camshift algorithm performs relatively better compared to the existing method to generate accurate approximation of the ROIs of the occluded objects.

Figure 9 shows a case of occlusion of two objects with similar color and moving direction. The figures show that the proposed method more accurately approximates the ROIs of the occluded objects compared to the existing method and Camshift method. The proposed occlusion

handling method shows sufficient results to cope with the case when the occluded objects have similar color and similar moving direction in test sequence 2. Here, the occlusion handling takes advantage of the different motion speed of the two objects, which affects the calculation of Eq. (3). Therefore, the proposed method can correctly separate the ROIs of two similar objects better than the existing method.

Figure 10 shows a further comparison of the proposed method with the methods in [5] and [27] for test sequences 4 to 7, in which each of them has an HD resolution. The captured area from test sequences 4 to 7 is shown from row 1 to row 4, respectively. Three sample frames of each test sequence for the existing method, Camshift method and the proposed method are shown in the left, middle, and right column, respectively. We select occlusion cases from the sequences to point out the main issues with the methods in [5] and [27]. While the accuracy of those methods is relatively high for detecting different objects with different colors, tracking two or more objects with similar color would fail.

The performance of the proposed method and the method in [5] is assessed using the F-measure score, calculated as

$$\text{F} - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{where Precision} = \frac{TP}{TP + FP} \text{ and Recall} = \frac{TP}{TP + FN}.$$

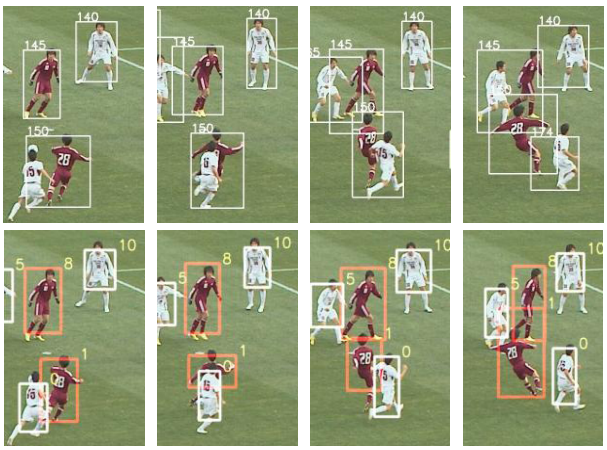


Fig. 8 Comparison of occlusion handling for test sequence 1 between the existing method (above row) and Camshift method (bottom row) is shown for frame 130, 135, 140, and 145, respectively.

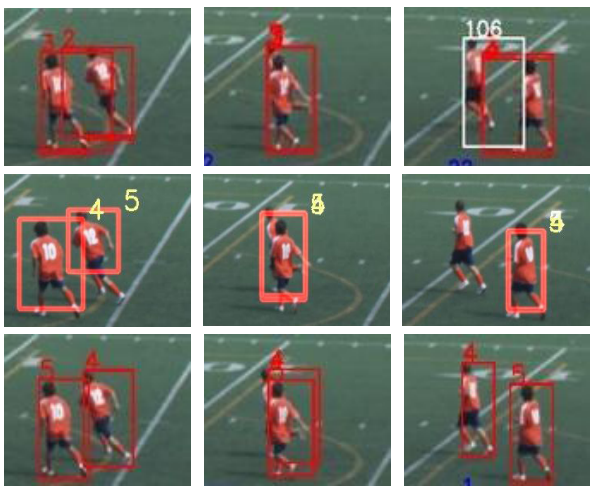


Fig. 9 Comparison of managing occlusion of two objects with the same color and direction in test sequence 2 for frame 250, 264, and 278 between the existing method (above row), Camshift method (middle row), and the proposed method (bottom row) shows the robustness of the proposed method.

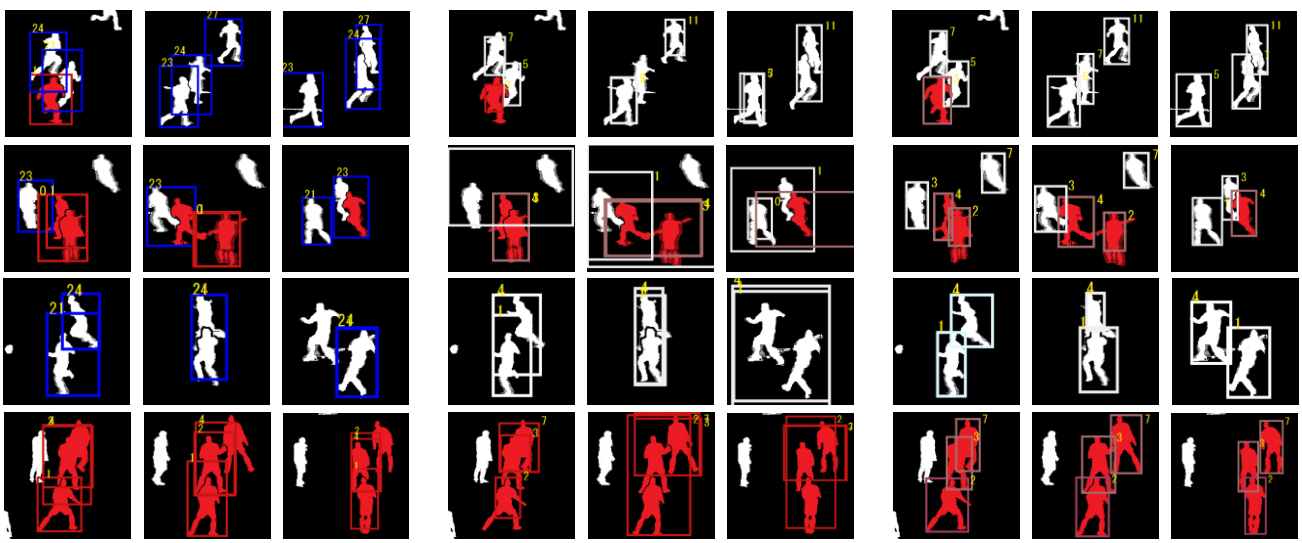


Fig. 10 The screen captures from frames 47, 67, and 87 of test sequence 4 (1st row), frames 70, 80, and 90 of test sequence 5 (2nd row), frames 16, 26, and 36 of test sequence 6 (3rd row) and frames 3, 23, and 43 of test sequence 7 (4th row) showing the tracking results for the existing method (left columns), Camshift method (middle columns) and the proposed method (right columns).

Table 1 Performance evaluation

Seq.	Recall			Precision			F-measure		
	Prop.	[5]	CamS.	Prop.	[5]	CamS.	Prop.	[5]	CamS.
Test 1	0.992	0.916	0.945	0.952	0.864	0.865	0.971	0.886	0.903
Test 2	1	1	0.979	0.995	0.971	0.934	0.997	0.985	0.955
Test 3	0.929	0.922	0.897	0.983	0.943	0.896	0.955	0.931	0.896
Test 4	0.994	0.989	0.879	0.834	0.827	0.802	0.902	0.900	0.900
Test 5	0.958	0.844	0.882	0.946	0.798	0.789	0.951	0.820	0.832
Test 6	0.999	0.821	0.824	0.885	0.792	0.803	0.934	0.805	0.812
Test 7	0.963	0.913	0.918	0.964	0.809	0.778	0.963	0.858	0.842

Here, a true positive (*TP*) denotes the number of ROIs that match with the actual object (in terms of location and size), a false positive (*FP*) denotes the number of ROIs that do not match with the actual object (including misplaced ROIs), and the number of objects having no ROIs assigned as a false negative (*FN*), respectively.

The comparison of performance between the proposed method with the existing method in [5] and Camshift method in [27] is shown in Table 1. The proposed method performs comparably better than the other methods with up to 10% accuracy. The proposed method performs relatively well when the sizes of objects are small (relative to the frame size), such as in test sequences 1 and 2 where the sizes of the detected objects are roughly 10% of the height of the frame. The proposed method achieves better performance because it can correctly handle the occlusion occurrences that involve two objects with similar color as shown in Fig. 9. For the sequence where camera fast movement and zooming are present, the proposed method performs better than the existing method and the Camshift method. In fact, the method in [5] can almost track the objects better than the Camshift method as shown by the recall values. However, the identification accuracy of both methods decreases when the camera movement is intense.

The drawback of the proposed method is that when two or more objects are occluded from the beginning of the frame, the objects cannot be separated automatically. Therefore, the objects will be assigned the same ROI until the objects are separated. At this separation point, each object can be successfully identified as individual objects.

5. Conclusions and Future Works

We have shown in this paper the performance of automatic object tracking in a single, moving camera based on an attribute matching algorithm that measures the similarity of the values of the attributes of the objects in the temporal domain. By computing the correlation of those attributes, the tracking process can be performed to correctly identify each object even during occlusion as presented in the experimental results. In addition, some improvements in the segmentation process have also been provided to ensure the accuracy of the tracking process for a dynamic background.

Some limitations are present in the proposed method. Firstly, the segmentation of the object from the commercial board solely depends on the watershed algorithm, which may fail when the regions to be segmented are similar in color. Secondly, the automatic tracking highly depends on the segmentation process, thus tracking performance will suffer from imperfect segmentation. Lastly, the occlusion handling cannot separate the pixels of two or more occluded objects. Therefore, when the objects are occluded at the first observed frame, the proposed method can only separate their ROIs after the objects are actually separated.

Our future work includes enhancing the segmentation process, especially to provide automatic segmentation for dynamic background (camera movement with pan-tilt-zoom occurrences), as well as a method to separate object textures during occlusion, to ensure more accurate and more reliable object tracking to produce better texture extraction for free viewpoint application.

Acknowledgements

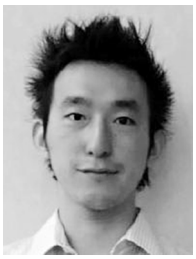
This work was supported by "R&D on Ultra-Realistic Communication Technology with Innovative 3D Video Technology," the Commissioned Research of the National Institute of Information and Communications Technology (NICT), Japan.

References

[1] T. Kanade, P.W. Rander, and P.J. Narayanan, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes," IEEE Multimedia, vol.4, no.1, pp.34–47, 1997.
[2] T. Koyama, I. Kitahara, and Y. Ohta, "Live Mixed-reality 3d Video in Soccer Stadium," Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality, pp.178–187, 2003.
[3] K. Hayashi and H. Saito, "Synthesizing Free-viewpoint Images from Multiple View Videos in Soccer StadiumADIUM," Proc. IEEE International Conf. on Computer Graphics, Imaging and Visualization, pp.220–225, 2006.
[4] M. Germann, A. Hornung, R. Keiser, R. Ziegler, S. Würmlin, and M. Gross, "Articulated Billboards for Video-based Rendering," in Computer Graphics Forum, vol.29, no.2, pp.585–594, 2010.
[5] H. Sankoh and S. Naito, "Free-viewpoint Video Rendering in Large Outdoor Space such as Soccer Stadium based on Object Extraction and Tracking Technology," The Journal of The Institute of Image Information and Television Engineers (ITE), vol.68, no.3, pp.J125–J134, 2014.
[6] H. Sankoh, M. Sugano, and S. Naito, "Dynamic Camera Calibration Method for Free-viewpoint Experience in Sport Videos," Proc. 20th ACM International Conference on Multimedia, p.1125–1128, 2012.
[7] H. Sankoh, A. Ishikawa, S. Naito, and S. Sakazawa, "Robust Background Subtraction Method based on 3D Model Projections with Likelihood," Proc. Multimedia Signal Processing, pp.171–176, 2010.
[8] M.M.N. Ali, M. Abdullah-alWadud, and S.-L. Lee, "An Efficient Algorithm for Detection of Soccer Ball and Players," Proc. 16th ASTL Control and Networking, vol.16, 2012.
[9] S. Mackowiak, J. Konieczny, M. Kurc, and P. Mackowiak, "A Complex System for Football Player Detection in Broadcasted Video," Proc. International Conference on Signals and Electronic Systems, pp.119–122, 2010.
[10] M. Heydari and A.M.E. Moghadam, "An MLP-based Player Detection and Tracking in Broadcast Soccer Video," Proc. International Conference on Robotics and Artificial Intelligence, pp.195–199, 2012.
[11] V. Pallavi, J. Mukherjee, A.K. Majumdar, and S. Sural, "Graph-Based Multiplayer Detection and Tracking in Broadcast Soccer Videos," IEEE Trans. Multimedia, vol.10, no.5, pp.794–805, 2008.

[12] Y. Huang, J. Llach, and S. Bhagavathy, "Players and Ball Detection in Soccer Videos Based on Color Segmentation and Shape Analysis," Proc. MCAM International Workshop, pp.416–425, 2007.
[13] B. Müller, Jr. and R.O. Anido, "Distributed Real-Time Soccer Tracking," Proc. ACM 2nd International workshop on Video Surveillance & Sensor Networks, pp.97–103, 2004.
[14] H. Itoh, T. Takiguchi, and Y. Ariki, "3D Tracking of Soccer Players Using Time-Situation Graph in Monocular Image Sequence," Proc. 21st International Conf. on Pattern Recognition, pp.2532–2536, 2012.
[15] H. Kataoka, K. Hashimoto, and Y. Aoki, "Player Position Estimation by Monocular Camera for Soccer Video Analysis," Proc. SICE Annual Conference, pp.1985–1990, 2011.
[16] H. Kataoka and Y. Aoki, "Football Players and Ball Trajectories Projection from Single Camera's Image," in Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp.1–4, 2011.
[17] K. Choi and Y. Seo, "Tracking Soccer Ball in TV Broadcast Video," Proc. 13th International Conf. on Image Analysis and Processing, pp.661–668, 2005.
[18] R. Hamid, R.K. Kimar, M. Grundmann, K. Kim, I. Essa, and J. Hodgins, "Player Localization Using Multiple Static Cameras for Sports Visualization," in IEEE Conf. on Computer Vision and Pattern Recognition, pp.731–738, 2010.
[19] J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple Player Tracking in Sports Video, A Dual-Mode Two-Way Bayesian Inference Approach With Progressive Observation Modeling," IEEE Transaction on Image Processing, vol.20, no.6, pp.1652–1667, 2011.
[20] K. Sato and J.K. Aggarwal, "Tracking Soccer Players Using Broadcast TV Images," Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, pp.546–551, 2005.
[21] T. Yamamoto, H. Kataoka, M. Hayashi, Y. Aoki, K. Oshima, and M. Tanabiki, "Multiple Players Tracking and Identification Using Group Detection and Player Number Recognition in Sports Video," Proc. 39th Annual Conf. of the IEEE Industrial Electronics Society, pp.2442–2446, 2013.
[22] W.-L. Lu, J.-A. Ting, J.J. Little, and K.P. Murphy, "Learning to Track and Identify Players from Broadcast Sports Videos," IEEE Trans. Pattern Anal. Mach. Intell., vol.35, no.7, pp.1704–1716, 2013.
[23] J. Liu, X. Tong, W. Li, and T. Wang, "Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video," Pattern Recognition Letters, vol.30, no.2, pp.103–113, 2009.
[24] X. Tong, J. Liu, T. Wang, and Y. Zhang, "Automatic Player Labeling, Tracking and Field Registration and Trajectory Mapping in Broadcast Soccer Video," ACM Trans. Intelligent Systems and Technology, vol.2, no.2, p.15, 2011.
[25] J.R. Smith and S. Chang, "VisualSEEk a Fully Automated Content-Based Image Query System," ACM Multimedia, pp.87–98, 1996.
[26] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker, Description of the Algorithm," Intel Corp. Microprocessor Research Labs., 2000.
[27] F. Meyer, "Color Image Segmentation," in Image Processing and its Applications, IET, pp.303–306, 1992.
[28] Z. Wang, X. Yang, Y. Xu, and S. Yu, "Camshift Guided Particle Filter for Visual Tracking," IEEE Workshop on Signal Processing Systems, pp.301–306, 2007.

Houari Sabirin received his Ph.D. in Information and Communications Engineering from Korea Advanced Institute of Technology, Daejeon, Korea in 2012. From 2012-2013 he was a postdoctoral researcher with the Information & Electronics Research Institute, in KAIST, Korea. He is now with KDDI R&D Laboratories, Inc.



Hiroshi Sankoh received his B.E. in Information Science and M.E. in Intelligence Science Technology, both from Kyoto University in 2006 and 2008, respectively. Since 2008 he has been with Ultra-Realistic Communications Laboratory in KDDI R&D Laboratories, Inc.

Sei Naito received his B.E., M.E., and Ph.D. degrees from Waseda University in 1994, 1996, and 2006, respectively. He joined Kokusai Densin Denwa Corporation (currently KDDI) in 1996. He is currently a senior manager of Ultra-Realistic Communications Laboratory in KDDI R&D Laboratories, Inc.