Recognition of Soccer Players after Occlusions using Temporal Color Signatures

IPCV 2011

Thomas Müller, Daniel Manger, and Jürgen Metzler (contact author)

***Abstract*— This paper deals with the automatic assignment of players' identities in images of sport scenes. The task to correctly assign a player id to one unique player over many frames remains a challenging task as the players usually wear nearly identical jerseys and therefore generate visually similar appearances. If low image resolutions prohibit a direct jersey number recognition, tiny visual differences between the players like shoe, hair or skin colors can still support the recognition process. We propose a novel system which gathers such visual differences of players over multiple frames and allows for a successful identification of players. Results are presented in the context of a merge and split handling module of a player tracking system where the recognition can significantly improve the correct assignment of players after ambiguous merge and split situations.**

***Keywords*—soccer, tracking, recognition, color signature.**

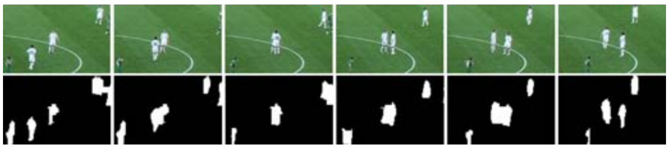

Fig. 1. A challenging occlusion situation in soccer player tracking (top row). A tracker based on the foreground segmentation (bottom row) needs additional appearance information to successfully recognize and reassign the two players of the same team after the merge situation.

I. Introduction

The automatic camera-based tracking of soccer players is an important component of modern sport with many applications. The trajectories of players offer interesting information to both media for online reporting and coaches who can analyze and optimize the players' behavior after a match. However, today, the acquisition of the relevant data still relies on a notable amount of human interaction. In current player tracking systems, human operators have to correct the output of the tracking system especially after situations when players come very close to each other and therefore cause occlusions in the respective camera images. Furthermore, some game situations like corner kicks or free kicks often lead to scenarios where tracking systems are likely to fail due to visual ambiguities when multiple players of one team are standing close together. Reducing the required amount of human interaction in such systems is therefore an active research area.

Some current systems make use of many high resolution cameras to ensure an appropriate resolution of the players' images all over the pitch in order to solve occlusions [1]. Another approach is to identify players after the occlusion dispersed using optical character recognition (OCR) methods for jersey number recognition [2], [3] or face identification approaches [4]. Due to their requested resolution and the claimed number of cameras such approaches are not applicable in lightweight systems. The resolution is not sufficient in order to recognize faces and in most cases the jersey numbers are not visible in the image. Figure 1 illustrates such a difficult occlusion situation. The recognition of players in critical situations, therefore, has to be accomplished by different algorithms. For instance, players can be recognized by color histogram back projection as proposed in [5]. This approach is appropriate if there are only players of different teams in the analysis region but it fails if there are multiple players from the same team as they yield similar color histograms. Another disadvantage is the vulnerability against changes of lightning conditions. In [6], soccer player recognition is mainly done by color histogram correlation. Although this difference measurement has established as a standard in color image processing, its application has strong limitations with respect to distinguishing tiny player color differences due to their weak weighting in comparison to the contribution of the other colors.

In this article, we present a system which deals with low resolution images of players and processes tiny visual differences in order to still discriminate players of one team with similar visual appearances.

This paper is organized as follows. Section II describes the details of the recognition system starting with the application context and its available input data to the system. Subsequently, Section III presents results in the context of a tracking system which successfully makes use of the recognition system to solve merge & split conflicts in ambiguous situations. Section IV concludes summarizing the main benefits and limits of the system.

The authors are with the Fraunhofer Institute for Optronics, System Technologies and Image Exploitation IOSB, Fraunhoferstrasse 1, 76131 Karlsruhe, Germany (email: {Thomas.Mueller | Daniel.Manger | Juergen.Metzler}@iosb.fraunhofer.de).

II. System Details

A. Application context

The recognition system presented in this paper is embedded in an overall player tracking system. Please note, that an exhaustive description of the tracking system is not the objective of this paper. Nevertheless, we give a brief outline of the system and depict its interfaces and its interactions with the recognition system.

The acquisition of images is performed by a single static camera which covers the whole pitch. The camera provides Full HD resolution images which, however, do not contain enough details to successfully recognize the players' jersey numbers. Due to real-time constraints, feature-based detections which often involve a sliding windows approach cannot be used. Thus, the tracking system works with a detection stage based on a fast foreground / background segmentation [7]. In this step, temporal static background like the pitch and game infrastructure are segmented as background, whereas moving players generate changing appearance and therefore lead to foreground segmentation.

Subsequently, all foreground regions are collected and extracted as blobs. The tracking of the players is performed by a multi-object-tracking which updates the tracks with the new blob information in every frame. As soccer players don't stick to a specific dynamic model, the tracking does not implement a state estimation with a dynamic model but performs a nearest-neighbor-assignment of blobs to tracks in each frame. When two or more players come close to each other, e.g. during a tackle, occlusion effects arise. In this case, the player detection step only extracts one joint blob for the players. The tracking module therefore has to explicitly deal with merge & split effects of the underlying blobs. This is solved through a merge & split detection process prior to the update of the tracks. Whenever two blobs have been registered as merged, the tracking module keeps track of the merged blob maintaining the number and identities of the players inside the joint track. If, afterwards, a split is detected, the players have to be reassigned to the original corresponding single tracks.

Thus, continuously tracking of players across merge & split situations requires a recognition module which solves the player assignment issue after a split. In advantageous situations where e.g. two players of different teams merge and split again, the assignment can be carried out with the assistance of color classification methods. However, in situations where multiple players of the same team are involved in a merge & split case, the emerging ambiguity can only be solved by an approach which tries to figure out tiny still existing differences in the visual appearance of the players, e.g. different shoe, hair or skin colors.

As the slight differences of two players can depend on both the current lighting conditions and the size of the current image section of the players (varying with the position on the pitch), an adaptive online-approach is essential. This means, that the appropriate visual information has to be collected and updated from the most recent frames prior to a merge. After a split, this history can then be compared against the new visual information coming up from the single tracks.

The next subsection characterizes the exact type and amount of information which is the main contribution of our work.

B. Recognition algorithm

The basic idea to distinguish soccer players is to build up a data structure for each player representing a kind of color finger print. As long as a player is tracked without merging with others in the image, the data structure is updated in order to cover the color range the player exhibits over time. When a merge of players in the image is detected, their color data structures are frozen as long as their tracking regions remain merged. When they split, the color information is extracted for the related players and compared to the color data structures in memory in order to determine who is who.

The color data structure and the comparison operations are constructed in a way that they can handle not only major color differences of jerseys and shorts, but also, to a large extend, some minor color differences of socks, shoes, hair or skin as well as differences caused by existing or non existing wristbands or sweatbands.

C. Color data structure

Let

$$p^{(n)} : D \mapsto \{0, \dots, 255\}^3$$

with $D := \{0, \dots, w-1\} \times \{0, \dots, h-1\}$ be the n-th color image in an image sequence of size $w \times h$ pixels with $p^{(n)}(x, y) = (c_1^{(n)}, c_2^{(n)}, c_3^{(n)})^T$ representing the three color channels at image position $(x, y)^T$ with respect to a chosen color space model like RGB, HSI or Lab.

Furthermore, let

$$T^{(n)} := \{l^{(n)}, \dots, r^{(n)}\} \times \{u^{(n)}, \dots, d^{(n)}\} \subseteq D$$

be the rectangular *tracking region* around a soccer player in image $p^{(n)}$ with the horizontal, i.e. left and right borders $l^{(n)}$, $r^{(n)}$, and the vertical borders $u^{(n)}$ and $d^{(n)}$.

The *foreground segmentation* (*blob extraction*) which is mentioned in the introduction of this section is a function

$$s^{(n)} : T^{(n)} \mapsto \{0, 1\} \quad (1)$$

which has the value 0 for the background pixels of $T^{(n)}$ and 1 for the foreground pixels, i.e. the calculated blob of the player in the image plane. See Figure 2 for illustration.

As a first step to obtain our proposed color data structure, we apply a *color reduction function*

$$f : \{0, \dots, 255\}^3 \mapsto \{0, \dots, 2^k - 1\}^3 \quad (2)$$

on image $p^{(n)}$, where $k < 8$ denotes the number of target color bits. On the one hand, this color reduction leads directly to advantages concerning grouping of similar colors and differentiation between relevant and irrelevant color differences. This is necessary due to color noise in the images and other color effects like shading, illumination changes, pixel aliasing, motion blurring and so forth. On the other

Fig. 2. Samples of foreground segmentation (blob extraction) $s^{(n)}$ (lower image row) for the image content in $T^{(n)}$ (upper row). $s^{(n)}(x,y)=0$ is depicted in black color and $s^{(n)}(x,y)=1$ in white color.

hand, the needed color information is compressed to a useful data amount which also leads to small evaluation times. Our optimization experiments led us to the value of $k=4$ as an appropriate value. This performs a reduction from the $256^3 \approx 16.8$ million colors (8 bits per color channel) of the used camera to $(2^k)^3 = 2^{12} = 4096$ colors (4 bits per channel). This step can be done very easily and efficiently with an arithmetic logical shift of the color bytes 4 bit positions to the right, which is equivalent to a division by 16 and truncating the decimal digits, i.e. $f(c_1,c_2,c_3) = (\lfloor\frac{c_1}{16}\rfloor, \lfloor\frac{c_2}{16}\rfloor, \lfloor\frac{c_3}{16}\rfloor)^T$, $c_1,c_2,c_3 \in \{0,\ldots,255\}$.[1]

For a tracking region $T^{(n)} \subseteq D$, the function

$$H^{(n)}_{T^{(n)}} : \{0,\ldots,2^{3k}-1\} \mapsto I\!N_0 \ , \tag{3}$$
$$H^{(n)}_{T^{(n)}}(i) := |\{(x,y) \in T^{(n)} \mid s^{(n)}(x,y) = 1 \ \wedge \ h(f(p^{(n)}(x,y))) = i\}|$$

defines the *foreground color histogram* for $T^{(n)}$ in image $p^{(n)}$. The argument i is a color index and the function h calculates this *linearized index* out of a (reduced) color $(c_1,c_2,c_3)^T$:

$$h : \{0,\ldots,2^k-1\}^3 \mapsto \{0,\ldots,2^{3k}-1\} \ , \tag{4}$$
$$h(c_1,c_2,c_3) := 2^{2k}c_1 + 2^k c_2 + c_3 \ .$$

Since the color histogram of each soccer player varies over time due to a variety of different reasons like player pose, shadows or color pixel aliasing, the color histograms have to be accumulated over a period of time in order to capture such variations and to obtain a more constant color finger print.

Multiple experiments have led to different attempts to do that. Best results were obtained by defining the color finger print structure by calculating the two *accumulated color histograms*

$$H^{(n)}_{\min}(i) := \min_{j=n-m+1,\ldots,n} H^{(j)}_{T^{(j)}}(i) \quad \text{and} \tag{5}$$
$$H^{(n)}_{\max}(i) := \max_{j=n-m+1,\ldots,n} H^{(j)}_{T^{(j)}}(i) \tag{6}$$

for each soccer player over an image history of m images with respect to the current n-th image frame $p^{(n)}$ ($i \in \{0,\ldots,2^{3k}-1\}$). That means, the color finger print $(H^{(n)}_{\min}, H^{(n)}_{\max})$ of each soccer player is represented by $2 \cdot 2^{3k} = 2 \cdot 4096 = 8192$ integer values.

$H^{(n)}_{\min}$ corresponds to an AND-operation and, therefore, represents the minimal color emission of a soccer player, i.e. the colors and numbers of colored pixels the player has always shown in history m. Thus, if the same player appears after a blob merge phase, his color histogram $H^{(n)}_{T^{(n)}}$ can be assumed to be equal or greater than $H^{(n)}_{\min}$ (with respect to all or at least most entries).

$H^{(n)}_{\max}$ realizes an OR-operation and, therefore, comprises all colors and numbers of colored pixels the soccer player has shown in history sometime. After a blob merge phase, the same player will not (or rarely) show more colors or higher numbers of colored pixels in his color histogram $H^{(n)}_{T^{(n)}}$ than in $H^{(n)}_{\max}$.

Paramter m has to be chosen reasonably, of course. It should be chosen big enough to capture the players' color appearance variations over time. If it is chosen too big, the described color model gets overfitted and may accumulate accidental and noise effects. In the examples presented here, we used $m=10$. The approach presented here has shown as being robust against the exact choice of this parameter. With values of m up to 25 or 50 we obtained similar results compared to $m=10$.

D. Soccer player recognition

As stated above, a soccer player can be recognized by calculating his current color histogram $H^{(n)}_{T^{(n)}}$ and comparing it to all relevant players' color finger prints $(H^{(n)}_{\min}, H^{(n)}_{\max})$ in order to determine the best match. So, by using the abbreviations $H(i) := H^{(n)}_{T^{(n)}}(i)$, $H_{\min} := H^{(n)}_{\min}$, $H_{\max} := H^{(n)}_{\max}$, the best match conforms to

$$H_{\min}(i) \leq H(i) \leq H_{\max}(i) \tag{7}$$

for a high number of indices i (i.e. colors). However, simply counting the number of matching histogram bins i is not discriminative enough. Therefore, a quantitative error measurement function is constructed. This is done by penalizing the contribution of an index i if $H(i) < H_{\min}(i)$ or $H(i) > H_{\max}(i)$ incorporating ratings $g(x)$ of the differences $x = H_{\min}(i) - H(i)$ or $x = H(i) - H_{\max}(i)$, respectively. Additionally, distances $d(H,i)$ in color space are taken into account in order to increase the contribution of exceptional colors of a player.

x is rated with values from 1 to 7 according to

$$g : I\!N \mapsto \{1,\ldots,7\} \ ,$$

[1] Alternatively, f could be calculated with a more powerful color reduction function, if a higher processing time is uncrucial. Multiple promising approaches can be found in the literature, for example, working with tree clustering for an adaptive color reduction [8], self-organizing maps and growing self-organizing neural networks [9], Kohonen self-organizing feature maps combined with the Gustafson-Kessel fuzzy algorithm [10], or an ant colony based approach [11]. An optimization concerning this matter would be interesting, but is not a subject in this paper.

$$g(x) := \begin{cases} 1, & \text{if } x < 3 \\ 2, & \text{if } \ 3 \le x < \ 7 \\ 3, & \text{if } \ 7 \le x < 15 \\ 4, & \text{if } 15 \le x < 25 \\ 5, & \text{if } 25 \le x < 40 \\ 6, & \text{if } 40 \le x < 60 \\ 7, & \text{if } x \ge 60 \end{cases} . \tag{8}$$

In this way, on the one hand, small variations of histogram bin differences are neglected since they are mainly produced by random effects. On the other hand, bigger variations indicate structural effects which have to be taken into account according to their magnitude. The concrete values chosen in Formula (8) were determined empirically in order to optimize the soccer player recognition.

$d(H,i)$ calculates for a color histogram or accumulated color histogram H the euclidean distance in color space from the color with index i to the nearest color with $H(*) > 0$:

$$d(H,i) := \min_{\substack{c_1,c_2,c_3 \in \{0,\ldots,2^k-1\}, \\ d_1,d_2,d_3 \in \{0,\ldots,2^k-1\}, \\ i = h(c_1,c_2,c_3), \\ H(h(d_1,d_2,d_3)) > 0}} \sqrt{(c_1-d_1)^2 + (c_2-d_2)^2 + (c_3-d_3)^2} . \tag{9}$$

Discrepancies are assessed with the function r which is defined as

$$r(i) := \begin{cases} 0, & \text{if } H_{\min}(i) \le H(i) \le H_{\max}(i) \\ 10\sqrt{d^2(H,i)-1} \cdot g(H_{\min}(i)), & \text{if } H_{\max}(i)>0 \wedge \\ & \quad H(i)=0 \wedge H_{\min}(i)>0 \wedge \\ & \quad d(H,i)>1 \\ g(H_{\min}(i)), & \text{if } H_{\max}(i)>0 \wedge \\ & \quad H(i)=0 \wedge H_{\min}(i)>0 \wedge \\ & \quad d(H,i)=1 \\ 10\sqrt{d^2(H_{\max},i)-1} \cdot g(H(i)), & \text{if } H_{\max}(i)=0 \wedge \\ & \quad H(i)>0 \wedge d(H_{\max},i)>1 \\ g(H(i)), & \text{if } H_{\max}(i)=0 \wedge \\ & \quad H(i)>0 \wedge d(H_{\max},i)=1 \\ 0, & \text{else} \end{cases} \tag{10}$$

in order to rate the contribution of a color with index i.

Intuitively, $g(H_{\min}(i) - H(i))$ along with the condition $H(i) < H_{\min}(i)$ and $g(H(i) - H_{\max}(i))$ along with $H(i) > H_{\max}(i)$ should be used in Formula (10) instead of $g(H_{\min}(i))$ if $H(i) = 0$ and $g(H(i))$ if $H_{\max}(i) = 0$. This would be a more obvious formulation of the contribution of differences between the involved histogram bins. In fact, we started with that. But, finally, our result optimization experiments led to the variant stated in Formula (10). In other words, only histogram colors are taken into account that appear without being in the color finger print or colors that disappear after having been in the color finger print. So, obviously, it is advantageous to stress the presence or absence of colors and to neglect the concrete, less important pixel color numbers.

Furthermore, it should be commented why the cases $d(*,i) > 1$ and $d(*,i) = 1$ are distinguished and handled separately in the calculation of $r(i)$ in Formula (10). On the one side, $d(*,i) = 1$ often occurs according to color noise and aliasing effects. Experiments have shown that the high frequent occurence of those cases often overrules the more important (but less frequent) cases $d(*,i) > 1$ when calculating the sum of the $r(i)$ (cf. Formula (11) below) and therefore degrades the measurement function significantly. On the other side, the cases $d(*,i) = 1$ are needed sometimes to distinguish soccer players of the same team when most colors are similar. In order to lower the influence of the cases $d(*,i) = 1$ with simultaneous consideration of taking them into account when they are needed, the influence of the cases $d(*,i) > 1$ is weighted tenfold the cases with $d(*,i) = 1$.

The error measurement function e just sums up the colors' contributions

$$e(H,(H_{\min},H_{\max})) := \sum_{i \in \{0,\ldots,2^{3k}-1\}} r(i) \tag{11}$$

to evaluate the distance between a histogram H and a player's color finger print $(H_{\min}, H_{\max})$.

Eventually, the recognition of a soccer player can be done in two ways. The first one is to calculate his current histogram H and to search for the player's color finger print $(H_{\min}, H_{\max})$ with minimal error $e(H,(H_{\min},H_{\max}))$.

Alternatively, since, in general, all N players have to be recognized after a blob merge of N soccer players ($N \ge 2$), the optimization process should preferably be done as follows instead of the optimization strategy stated before. First, each player's histogram H_j is calculated ($j = 1,\ldots,N$). Then, the minimum of e is calculated for each (fixed) relevant color finger print $(H_{\min}, H_{\max})$ for the H_j in order to determine which histogram matches best to a fixed chosen color finger print. This approach has two advantages in comparison to the method described above. First, the optimization is done over the players' histograms for constant color finger prints at a time (and not vice versa as above). This has to be preferred due to construction of the color finger print and e. Second, the N calculated minima can be compared in order to do a consistency check. If all results are consistent, this is a good indication that this coincides with reality. Otherwise, the found ambiguities can be solved by comparing the N calculated minima (and, maybe, all calculated N^2 evaluations of e, too).

The latter optimization strategy is done in Section III in two examples for $N = 2$. Consider them for illustration.

The discussed formulas were applied to the color spaces RGB, HSI and Lab (cf. definition of $p^{(n)}$ at the beginning of Subsection II-C). Experiments have shown slightly better results when using HSI instead of RGB and slightly worse results when using Lab instead of RGB. The HSI color model was used to process the results presented in the following section.

III. Results

Figure 3 shows a typical merge & split situation of two soccer players of different teams.

Fig. 3. Example of a merge & split situation of two soccer players labeled P5 and P7 by the tracker. The three images show the situation before blob merge (left; image number 10131), the merge phase (middle; image number 10145) and the situation directly after the split (right; image number 10163). The image colors were transformed using the color reduction function f of Formula (2). The tracking regions are overlayed with yellow rectangles.

The players are labeled P5 and P7 by the tracking procedure. After the split both detected tracking candidates C1 and C2 are matched against the color finger prints $(H^{\mathrm{P5}}_{\min}, H^{\mathrm{P5}}_{\max})$ and $(H^{\mathrm{P7}}_{\min}, H^{\mathrm{P7}}_{\max})$ which were calculated until the blob merge occured, i.e. over time before image 10131. The evaluation results after the split at image 10163 are

$$\begin{aligned} e(H^{\mathrm{C1}}, (H^{\mathrm{P5}}_{\min}, H^{\mathrm{P5}}_{\max})) &= 11 \ , \\ e(H^{\mathrm{C2}}, (H^{\mathrm{P5}}_{\min}, H^{\mathrm{P5}}_{\max})) &= 1965 \ , \\[1ex] e(H^{\mathrm{C1}}, (H^{\mathrm{P7}}_{\min}, H^{\mathrm{P7}}_{\max})) &= 1712 \ \text{ and} \\ e(H^{\mathrm{C2}}, (H^{\mathrm{P7}}_{\min}, H^{\mathrm{P7}}_{\max})) &= 77 \ . \end{aligned}$$

The minimum of the upper two values as well as the minimum of the lower two values indicate clearly that C1 = P5 and C2 = P7 which coincides with reality.

A second example is shown in Figure 4, this time with two players of the same team (labeled P15 and P35 by the tracking procedure).

Fig. 4. Example of a merge & split situation of two players labeled P15 and P35 who play in the same team. The images show the situation before blob merge (left; image number 10224), the merge phase (middle; image number 10233) and the situation directly after the split (right; image number 10242). The image colors were transformed using the color reduction function f of Formula (2). The tracking regions are overlayed with yellow rectangles.

With an analogous naming of the variables as in the example before, the evaluation results after the split (i.e. at image 10242) are

$$\begin{aligned} e(H^{\mathrm{C2}}, (H^{\mathrm{P15}}_{\min}, H^{\mathrm{P15}}_{\max})) &= 9 \ , \\ e(H^{\mathrm{C1}}, (H^{\mathrm{P15}}_{\min}, H^{\mathrm{P15}}_{\max})) &= 46 \ , \\[1ex] e(H^{\mathrm{C2}}, (H^{\mathrm{P35}}_{\min}, H^{\mathrm{P35}}_{\max})) &= 9 \ \text{ and} \\ e(H^{\mathrm{C1}}, (H^{\mathrm{P35}}_{\min}, H^{\mathrm{P35}}_{\max})) &= 3 \ . \end{aligned}$$

TABLE I

Performance of the presented recognition approach.

K	correct	wrong	unclear	#events
1	98.58 %	1.30 %	0.12 %	90861
2	97.89 %	1.98 %	0.13 %	90733
3	97.30 %	2.54 %	0.16 %	90605
4	96.92 %	2.92 %	0.16 %	90477
5	96.64 %	3.20 %	0.16 %	90350
10	95.81 %	4.03 %	0.16 %	89694
20	94.88 %	4.94 %	0.18 %	86217
50	92.86 %	6.93 %	0.21 %	76851
100	90.62 %	9.18 %	0.20 %	63356
200	87.45 %	12.34 %	0.21 %	40276
400	88.14 %	11.63 %	0.23 %	12102
500	90.88 %	8.83 %	0.29 %	3060

The minimum of the upper two values as well as the minimum of the lower two values mean that C2 = P15 and C1 = P35 which coincides with reality.

In order to quantify the performance of the proposed recognition approach we evaluated each player's histogram calculated in the current image against each other player's color finger print calculated at up to K images before, i.e. calculated over m images before the gap of the recent K images. So, assuming every player in every image of our image sequence would be in a split situation after a virtual merge phase of K images in length with each other player, we calculated the probability of recognizing him and not to confuse him with the second hypothetical merge & split candidate. In this way, we calculated the performance values presented in Table I. Depending on K, we determined how many player recognitions are correct, wrong, or unclear (that means the two players had equal evaluations e). The last column shows how many hypothetical merge & split events have taken place, leading to the values in the first three columns. As one can see, the recognition performance slightly decreases with increasing parameter K. Altogether, the values prove that it is possible to recognize soccer players to a large extent with the presented method by using the described color finger print.

Additionally, please note that in real sequences the duration of most merge situations is less than 200 images (i.e. 8 seconds assuming 25 images per second). Furthermore, due to segmentation issues, many merge situations last only one or a few images.

Table II shows the calculation time on an Intel Pentium D PC with 3.2 GHz. The value for the histogram calculation depends on the size of the tracking regions $T^{(n)}$. In our image sequence this size varies from 20×42 to 62×70 pixels for the soccer players. The stated calculation time is the average over all histogram calculations.

The calculation time shows the real-time capability of the algorithm, even if every player was involved in a split event in every image.

TABLE II

CALCULATION TIMES PER FRAME AND PER PLAYER.

Histogram calculation (plus conversion from RGB to HSI, if needed	0.09 ms 0.38 ms)
Color finger print update	0.035 ms
Evaluation e	1.60 ms

IV. CONCLUSION AND FUTURE WORK

In this work, we presented an algorithm which can support a soccer player tracking framework with respect to solving critical merge & split situations caused by occlusions. Our method enables correct recognitions and reassignments of multiple players after occlusions even in low resolution images or if the players are from the same team. The presented performance evaluation shows that our method can be a powerful and valuable component for a real-time image-based soccer player tracking system.

In the future, the proposed method will be applied on a multitude of soccer games for evaluation purposes. The focus will be to further optimize the recognition performance using mass data in order to minimize human interaction in the overall tracking system.

REFERENCES

[1] S. Iwase and H. Saito: "Tracking soccer players based on homography among multiple views". *Proc. of SPIE 2003* vol. 5150, 2003, pp. 283-292.

[2] M. Šaric, H. Dujmic, V. Papic, N. Rožic, and J. Radic: "Player number recognition in soccer video using internal contours and temporal redundancy". *Proc. of the 10th WSEAS International Conference on Automation & Information*, Prague, Czech Republic, 2009. Published at World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, ISBN 978-960-474-064-2, pp. 175–180.

[3] M. Bertini, A. Del Bimbo, and W. Nunziati: "Player identification in soccer videos". *Proc. of the 7th ACM SIGMM International Workshop on Multime Dia Information Retrieval MIR '05*, Hilton, Singapore, 2005. Published at ACM, New York, NY, USA, ISBN 1-59593-244-5, pp. 25–32.

[4] L. Ballan, M. Bertini, A. Del Bimbo, and W. Nunziati: "Automatic detection and recognition of players in soccer videos". *Proc. of the 9th International Conference on Advances in Visual Information Systems* VISUAL '07, Shanghai, China, 2007. Published at Springer Verlag, Berlin, Heidelberg, Germany, ISBN 3-540-76413-5, 978-3-540-76413-7, pp. 105–116.

[5] S. C. Yongduek, S. Choi, Y. Seo, H. Kim, and K.-S. Hong: "Where are the ball and players? : Soccer game analysis with color-based tracking and image mosaick". *Proc. ICIAP*, 1997, pp. 196–203.

[6] O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka: "An object detection method for describing soccer games from video". *Proc. of the International Conference on Multimedia and Expo ICME*, vol. 1, 2002, pp. 26–29.

[7] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis: "Real-time foreground-background segmentation using codebook model". *Real-Time Imaging* 11, 2006, pp. 167-256.

[8] N. Papamarkos, A. E. Atsalakis, and C. P. Strouthopoulos: "Adaptive color reduction". *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 32, no. 1, 2002, pp. 44-56.

[9] G. Cheng, J. Yang, K. Wang, and X. Wang: "Image color reduction based on self-organizing maps and growing self-organizing neural networks". *Sixth International Conference on Hybrid Intelligent Systems HIS 2006*, Rio de Janeiro, Brazil, 2006, pp. 24–24.

[10] K. Zagoris, N. Papamarkos, and I. Koustoudis: "Color reduction using the combination of the Kohonen self-organized feature map and the Gustafson-Kessel fuzzy algorithm". *Proc. of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition MLDM '07*. Published at Springer-Verlag Berlin, Heidelberg, Germany, 2007, ISBN 978-3-540-73498-7, pp. 703–715.

[11] A. T. Ghanbarian, E. Kabir, and N. M. Charkari: "Color reduction based on ant colony". *Pattern Recognition Letters*, vol. 28, issue 12, 2007, pp. 1383–1390.