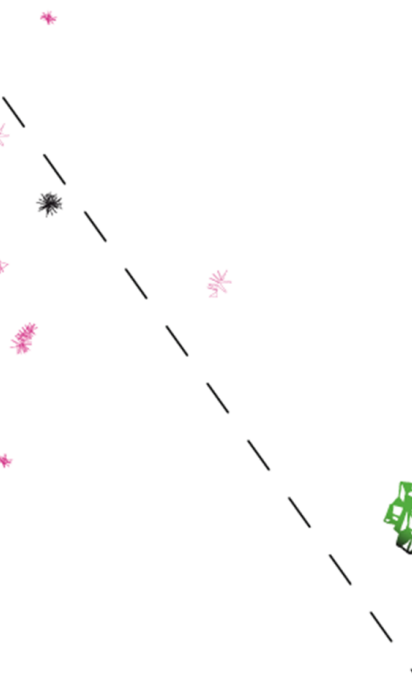




UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Multi-Camera Tracking of Soccer Players Through Severe Occlusions



W.G. Oude Elferink
M.Sc. Thesis
22 August 2017

Supervisors:

dr. ir. L.J. Spreeuwers
prof. dr. ir. R.N.J. Veldhuis
dr. C. Brune

Services, Cybersecurity and Safety Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Multi-Camera Tracking of Soccer Players Through Severe Occlusions

Wout Oude Elferink*

Abstract—Accurate visual tracking of soccer players is in high demand in the industry. Existing solutions are however unable to track players during severe occlusions, making manual supervision necessary. In order to get to a fully autonomous system we propose a setup based on the fusion of 14 cameras. An algorithm with at its core visual hull calculations combined with an HOG and HSV based appearance model is employed to track the players. It is shown that the proposed system is able to reduce the number of identity switches while maintaining tracking accuracy during severe occlusions.

I. INTRODUCTION

Using data is becoming more and more important in the sports industry with applications ranging from real time coaching advisory to scouting, media and betting. One of the first success stories dates back to the late 1990s when the Oakland Athletics baseball team started using data in their organization. Later the Boston Red Sox became champions in the world series due to the influence of data analyst Bill James. The recent successes in soccer of FC Midtjylland, who won the 2014-2015 Danish Superliga with the help of data statistics, led to an increase of interest in data by the soccer industry. Leading clubs like Arsenal and Manchester United now have their own data-analytics department.

In response to the increased interest several companies started generating positional data on soccer players and the ball. Opta [1] collects statistics by manually annotating the goals, passes, possession information and so on from every major league soccer match. Lower level information on player positions is gathered using computer vision by companies like Sentiosports [2], Prozone [3] and Chyron Hego with their TRACAB tracking system [4]. These companies do however still need manual interaction to correct tracking errors and generally have difficulties tracking players in cluttered situations.

The most interesting situations in soccer - corners, free-kicks and scoring opportunities - are generally also the most difficult to track due to many similar looking players occluding each other. Manually correcting faulty tracks is labor intensive, possibly not real-time and costly. Therefore a solution for automatically tracking the players to generate accurate and reliable data even in cluttered scenes is proposed here.

In this paper we will first look at related work. Then the hardware setup is discussed. Next the basic version of the algorithm employed in this paper is explained. The

additions to this basic framework for improving the tracking performance are then given. It is then shown what impact these additions have on the tracking performance. Finally a conclusion and recommendation is given.

II. LITERATURE

Tracking soccer players is essentially a multiple object tracking (MOT) problem. This problem has been studied extensively, mostly for crowd monitoring using security cameras [5]. The MOT problem has as main difficulties that there are an unknown number of persons to track which do possibly occlude each other, get occluded by the surroundings and wear an unknown type and color of clothing [5]. Similarly the soccer tracking problem also needs to track persons which get occluded. But different from the MOT problem, there is a known number of soccer players. Furthermore these players cannot be occluded by the surroundings - except for occlusions by the goal posts - only by other players. Although these occlusions tend to be more difficult than in the MOT problem due to a similar appearance of players (i.e. wearing the same shirt and shorts), sudden movement and direction change, physical contact between players and large groups in a small area (e.g. corner situations). A typical example of occlusion with views from multiple cameras can be found in figure 1.

Solutions to the MOT problem can be categorized into several groups. First of all you have "online trackers" which give an immediate answer to where the persons are after a frame is given. Next to this there are "delayed trackers" which can combine the hypotheses of the position of persons from multiple frames to calculate a globally optimal track, this however means that positions are generated with some delay [5].

Another aspect the trackers can be grouped on is the use of a single or multiple cameras [6]. Using multiple cameras has the advantage that there are observations from different perspectives, reducing the number of occlusions and possibly having a better view on the target [7]. This however gives rise to more computation power and needs an algorithm which can combine the data from the different cameras. This data fusion can happen before tracking, e.g. calculating tracks after a homography computation [8], or after tracking, e.g. combining the tracks calculated on separate videos [9].

Soccer Player Tracking Literature

Tracking soccer players using an automatic camera based system has been subject to extensive research over the years. Early attempts in the year 2000 were already able to track the ball and players in simple situations using low resolution

*W.G. Oude Elferink is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands w.g.oudeelferink@alumnus.utwente.nl

Special thanks to my committee: Luuk Spreeuwiers, Raymond Veldhuis and Christoph Brune

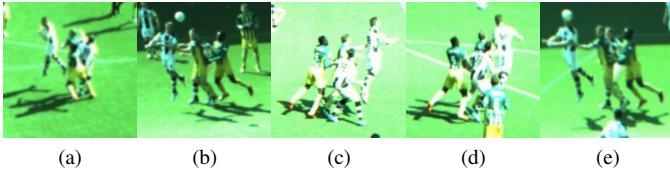


Fig. 1: (a-e) Multiple views of the same situation showing severe occlusion.

cameras [10]. The promise of a fully autonomous system which can track the players in every situation does however not exist, with many authors not even considering the hard situations (e.g. corners, free kicks) where many players are occluded. Although tracking in the more simple situations has been solved in multiple papers [11]–[32].

Multiple approaches have been taken to solve the tracking problem. Some authors make use of video streams from the dynamic broadcast camera [11], [13], [16], [19], [27], [33]–[37], multiple static cameras are used by [14], [17], [25], [29], [31], [32], [38]–[41] and multiple static cameras with overlapping field of view are employed in [10], [12], [18], [20], [23], [24], [26], [28], [30], [42]–[45]. Where broadcast cameras have the benefit of having no need for extra installations and their wide availability, however tracking can only be done on the players which are currently visible in the camera view and a sophisticated algorithm which can compensate for camera motion (zoom, pan, tilt) and which can map to a global coordinate view is needed. Static cameras with no overlapping field of view need only a couple of cameras - usually 2 or 3 - to track players, this reduces the cost of the system and makes it more easy to install and possibly move between locations. Furthermore using static cameras makes it possible to employ sophisticated background subtraction algorithms. Static cameras with an overlapping field of view is the most costly system set-up with up to 15 cameras [42] and the need for a more permanent placement in the soccer stadium. However it has the benefit of solving occlusions by tracking in multiple cameras and combining the tracks in a global manner.

Since we are interested mainly in tracking in difficult situations using multiple static cameras with overlapping field of view, only the related work with the best occlusion handling methods, appearance models and global data fusion shall be presented here.

Particle filters are a popular method for tracking single targets, however in multi-target tracking particles tend to end up quickly at a single mode losing track of the target. In [11] a mixture particle filter is made which can maintain multiple modes and has a better performance in multi object tracking tasks. As appearance model the authors make use of normalized color histograms.

Tracking by blob-detection after background segmentation using the Multiple Hypothesis Tracker (MHT) framework [46] is done in [16], [47]. To track multiple players which form one blob, players are segmented using multiple heuris-

tics. Teams are segmented based on a color template with height bins, furthermore a compactness constraint - players have a certain size - and a height constraint are introduced to split multi-player blobs.

Collaborative particle filters with multiple overlapping static cameras are described in [18]. The principal axis from particles of all cameras is mapped to a ground plane using homographies, tracking can then be done in the ground plane after which particles in each camera view are re-sampled based on the new ground plane position. In this way the individual particle filters are guided through occlusions if the target remains visible in other cameras. As appearance model a color histogram is used.

Where the features used for weighting the particles in a particle filter approach are usually weighted using a preset constant in [21] the weights of the features in the feature vector are learned in order to get the best performance. This also allows for a combination of many features which are combined in an optimal fashion. The features used for tracking are based on constant acceleration, motion direction, RGB color histogram, blob area, non-overlapping regions and proximity in state space. By incorporating features generated by other single player particle filters, the particle filters become aware of one another while they don't need to share particles - which is normally computationally expensive - the authors call these pseudo-independent particle filters.

Morais et al [30] use multiple static cameras with overlapping field of view. The authors fuse the data from particle filters from different views with likelihood based on a per camera appearance model. This is done by projecting particles on an overall view and presenting them as a Gaussian likelihood distribution with covariance estimated from ground truth measurements. This allows for tracking on the ground plane using a likelihood map. As appearance model for the particle filter both HOG and HSV histograms with three height bins are used.

Instead of having dynamically changing positions of particles, in [32] model field particles are formed. These particles have a static location on the field and are sampled as a grid on the field. Tracking is performed by combining an HOG player detector with an HSV color histogram with height bins and motion from a Kalman filter for each particle around a player. Occlusions are handled by allowing particles to be used by multiple tracks and relying on the motion model.

For a more direct comparison of the papers an overview of all literature with respect to soccer tracking is given in appendix VII-A. For every paper the type of camera system, the number of cameras, tracking method, appearance model and occlusion handling method is stated. As well as an indication on the performance of the algorithm.

III. HARDWARE

There are several considerations concerning the camera setup. First of all the number of cameras, too many cameras are costly while too few cameras make the tracking more difficult. Since players tend to occlude each other far more often near the goal area, it is beneficial to have more cameras

covering this area. This maximizes the probability that at least one camera has an unobstructed view of the target. Next to this one can choose the field of view of the cameras and how high the cameras should be placed. A smaller field of view gives a higher resolution but covers a smaller area. The higher a camera is placed the less occlusions will be seen and the larger the coverage, although it does lead to a lower resolution.

Ultimately the design of the stadium lays constraints on the possible camera positions. Considering all the variables it is chosen to place 14 cameras with a 2332x1752 resolution, of which two are placed high (16m) above the field behind the goals - the overview cameras, these cover almost the whole field - and the remaining cameras are placed in the first ring at about 8m high and 16m away from the field. This leads to the camera coverage as seen in figure 2.

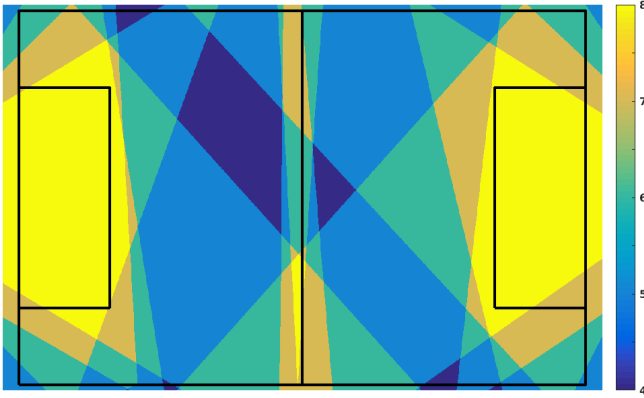


Fig. 2: Camera coverage, in color the number of cameras which see that part of the field. The camera coverage is especially high in the penalty area.

IV. ALGORITHM

In this section the basic algorithm without additions is given. This is the algorithm available at the start of this work and which is used as basis for further developments.

Figure 3 is a flowchart showing the main steps of the algorithm. As input to the algorithm the 14 RGB and background subtracted (BW) images are given. Using these images the players are found for initializing the tracks, furthermore the BW images are used for calculating the visual hull. The visual hull is converted into the Z-Map which is in turn used for tracking the players. As a final step the players can be automatically classified in teams. As output of the algorithm we get the player tracks and voxels every $1/25^{th}$ second.

A. RGB

As input to the system we get 14 camera images with a resolution of 2332x1752 at a framerate of 25 fps. See appendix VII-B for views from each camera.

B. BW

One of the most reliable methods for extracting moving targets from an image sequence is background subtraction.

Specifically we make use of the K-Nearest Neighbor algorithm from [48]. This algorithm is chosen because of its efficiency - especially on a GPU - its good segmentation and its ability to remove shadows. As output of the algorithm we get 14 binary images showing the moving targets. An example output foreground image (BW) can be seen in figure 4.



Fig. 4: Example of a foreground image after applying background subtraction (cropped image)

C. Find Players

To be able to initialize the tracking and for applying team classification, the individual segmented players need to be found. This is done using blob detection on the foreground images. The detected blobs are filtered using the constraints that the blob needs to be on the field, it does not touch the borders of the image, the height is within 1.4 and 2.1 meters and the width is within 0.6 meter. Using these constraint both the public, multi player blobs and small artifacts are filtered out. The result of the algorithm are the segmented players as seen in figure 5.

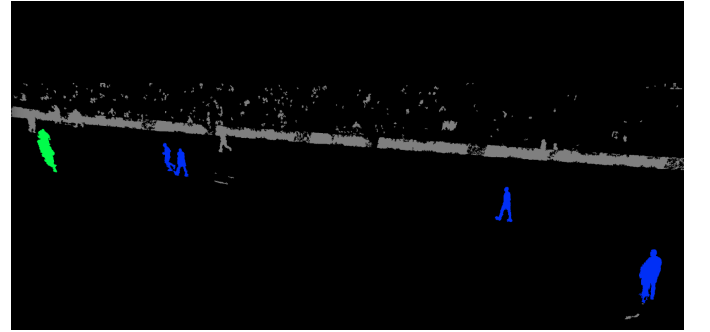


Fig. 5: Segmented single player blobs (blue) and multi player blobs (green) from the find player algorithm (cropped image, best viewed in color)

D. Visual Hull

There is one view in which players normally never get occluded, the top view, because it is very unlikely that players will be standing on top of one another. It is however impossible to install multiple static cameras hanging in the air above the field. Another method to get this view is calculating the 3D space from the different camera views and calculating the top view from this 3D space.

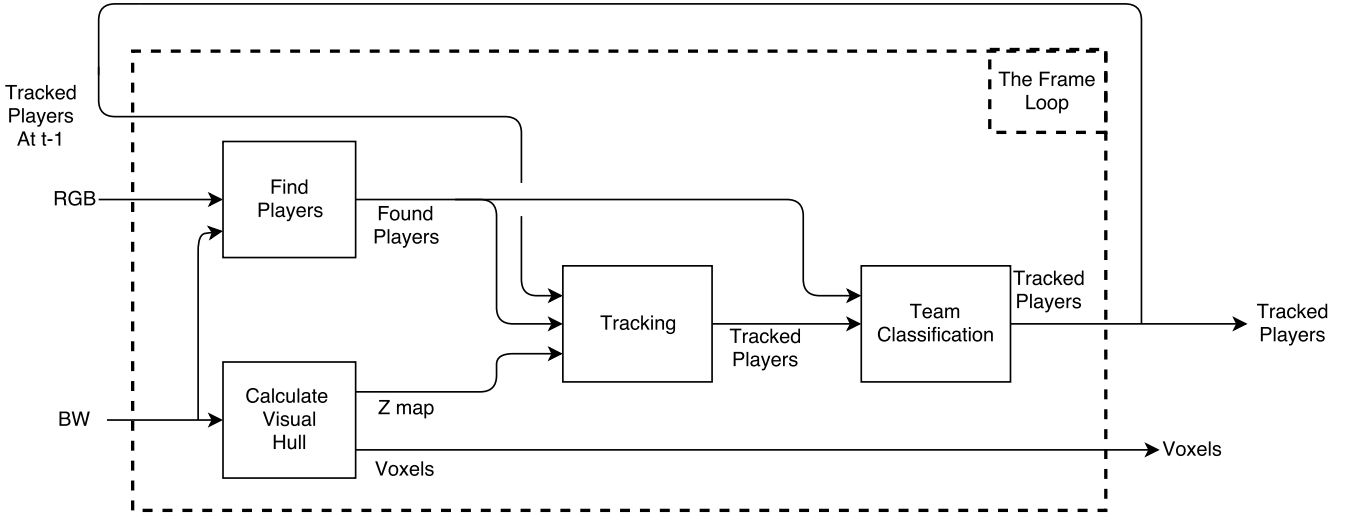


Fig. 3: Overview of the player tracking algorithm without additions

With the given camera setup there are two possibilities for calculating this 3D voxel space. The first being wide baseline stereo matching, although this method relies on matching points which is extremely difficult during occlusions of multiple players wearing the same color jersey and which are captured in low resolution. The other method is calculating the visual hull, which can efficiently be calculated using only the foreground images. The visual hull creates 3D objects which completely envelop the real 3D objects but are not necessarily the same shape and is an upper-bound on the actual shape of the object [49].

As output of the visual hull algorithm we get a 3D voxel space discretized in blocks of $5 \times 5 \times 5$ cm in a total space of $109 \times 70 \times 1.8$ m, where each block is either 1 (there is something there) or 0 (the space is empty). In figure 6 a visualization of the 3D voxel space is given.

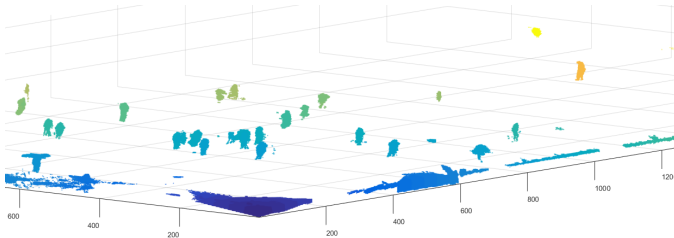


Fig. 6: 3D Voxel space, output of the visual hull algorithm (best viewed in color)

E. Z-Map

As said before the top view is the best view for tracking soccer players. No occlusions occur when looking down on the field. Furthermore we assume that players are standing upright. Using these assumptions it is found that a good clue for tracking is reducing the voxel space by summing the values in the z-direction. After summing in the z direction, a 2 dimensional map with values between 0 and 36 (i.e.

$1.8m/0.05m$) is created, the Z-Map (figure 7). Before tracking these values are normalized to values between 0 and 1 by dividing by 36.

Using the assumptions, it can be argued that the Z-Map holds the following properties. First the values are the highest above the center of a player because players are assumed to be standing upright. Secondly every grid cell can at most be assigned to one player since we are looking from above. And finally the values are evenly distributed around the center of a player and decreasing the further away from the center.

F. Tracking Players

The tracking part of the algorithm consists of three steps. First the tracks need to be initialized. Secondly the newly initialized tracks and the tracks from the previous time step are updated to get the player location at the current time step. And finally the tracks which do not track a player anymore can be deleted.

Initialize Tracks: Initializing tracks depends on combining the found players in the foreground images with players found in the Z-Map. In this way false positives (i.e. values at places where no players are) in the Z-Map can be filtered out. An overview of the initialization can be found in figure 8. In the first step the Z-Map is binarized by thresholding the Z-Map for values larger than 0.05 which means at least 2 voxels in the z-direction at that location. This binarized map is then dilated with size "voxels per meter (vpm)" / $4 = 20 / 4 = 5$ to get a new map which we will call D_5 . In D_5 the blobs which correspond to tracked players are selected using the position of the players from the previous frame which are possibly updated using a motion model. The non selected blobs are deleted, while the selected blobs are dilated once again with size $vpm \cdot 1.2 = 24$, we call this new map D_{24} . These two maps are then combined with the logical statement $D_5 \& !D_{24}$. The result of this combination is a map with blobs which do not correspond to or are close to an already tracked player. In this map the

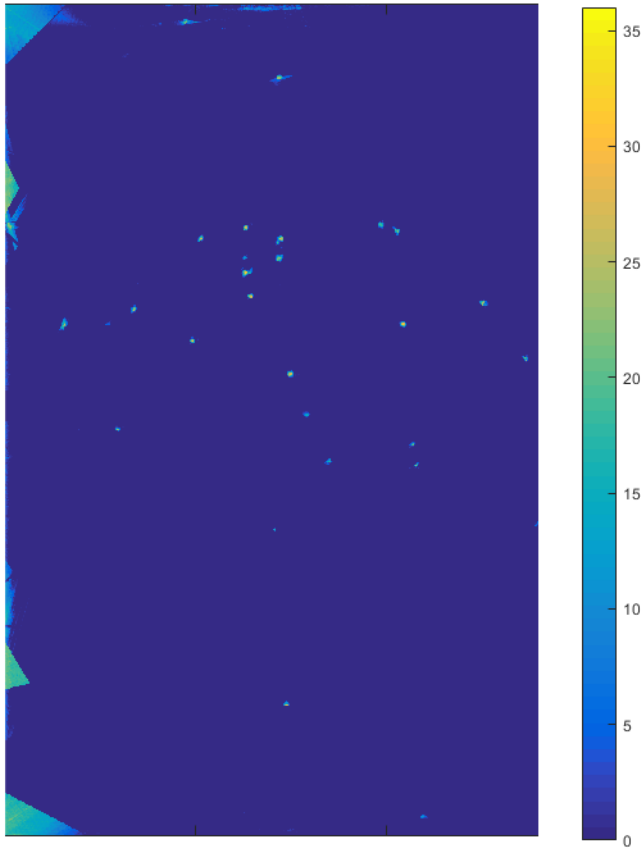


Fig. 7: Z-Map created from the 3D voxel space from figure 6 (best viewed in color)

center of the blobs are selected as new player locations if the blob area is between $(vpm/2)^2 = 100$ and $vpm^2 = 400$, the find player algorithm has found a player within 0.5 meter and other already tracked players are more than 2 meters away. If however there was a deleted track within 10 seconds and within 4 meter of this location, the deleted track is reinitialized and the positions in the missing frames are interpolated.

Update Tracks: For every new frame the player positions should be updated to their new locations based on the given data. The algorithm for updating the player positions can be found as pseudo code in 1 and as flowchart in figure 9 and is based on these assumptions:

- 1) Players cannot occupy the same space when looking downwards on the field.
- 2) Players are standing upright which means that the largest value of a blob in the Z-Map corresponds to the center of that player.
- 3) Players do not move faster than 40 km/h. The current world record for the 100m sprint shows a top speed of about 36 km/h.

Using 2 we know it is possible to make a box around a player in real world coordinates for the top down view and use mean shift tracking to track the players location. Using

1 we can remove other players from the Z-Map such that the mean shift tracker will not start to track other players. Using 3 we know that the initial size of the tracking box can be set to 90x90 cm and the player can always be found in that box given the last known position of the player.

```

foreach  $i \leftarrow \text{tracked player}$  do
  maskImage  $\leftarrow$  Z-Map
  foreach  $j \leftarrow \text{tracked player}$  do
    if  $j \neq i$  then
      set maskImage to 0 using a circular mask
      with diameter 11 (55cm) around the center
      of player j
    end
  end
  newPosition  $\leftarrow$  do a mean shift tracking on the
  maskImage with an initial circular box of 18x18
  (90x90cm) centered at previous position of player i
end

```

Algorithm 1: Tracking players with mean shift

Mean shift is chosen as tracking algorithm because it is able to find the highest mode in the area without using much computational power, furthermore it has been well studied in the tracking literature. For the mean shift algorithm itself, camshift is used as first implemented by Bradski et al. [50]. The algorithm essentially shifts the window towards the mean in that window and checks if the shift was more than a certain threshold (in this case 7.5cm). If the shift is lower, the algorithm converged and the position is returned, if the shift is higher the algorithm does a new iteration using the newly calculated mean position. The initial window size used is an ellipse with both diameters 90cm. However inside the mean shift algorithm these diameters might change per iteration depending on the square root of the covariance matrix of the image moments as in [50]. The diameters are then calculated as being two times the square root of the eigenvalues of the aforementioned matrix, unless the change is more than 10% in which case the previous window size is used.

Delete Lost Tracks: Since players can be substituted, walk out of the camera view or are lost due to other reasons, we need an algorithm that deletes lost player tracks. This is a simple algorithm which sums the number of on pixels in a square of 30x30 cm around the center of a track in the thresholded player likelihood image (image after thresholding the Z-Map in figure 8). If less than 50 percent of the pixels is "on" in 10 consecutive frames the track is marked as lost and will not be tracked anymore in the next frames.

G. Team Classification

After the tracking step the algorithm is able to classify the players in teams. But since this step does not influence the tracking it won't be described in this paper.

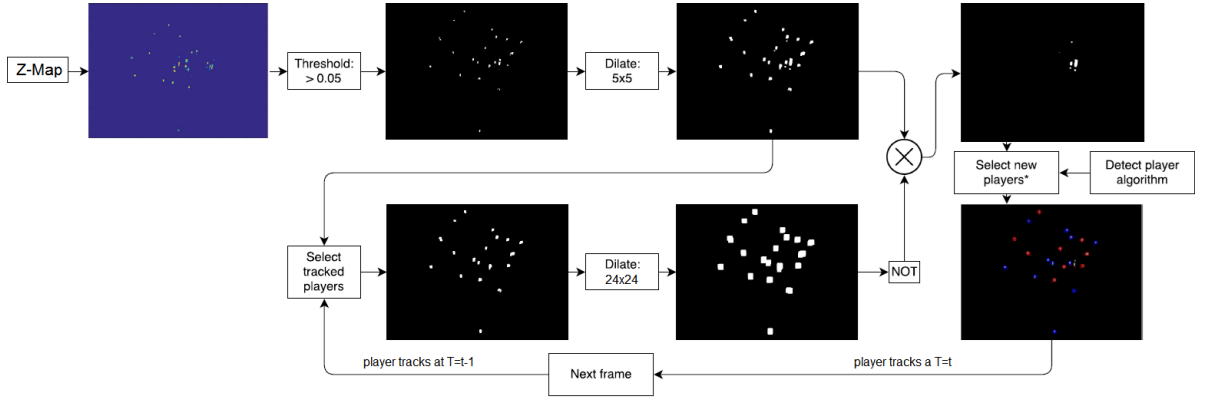


Fig. 8: Initializing new player tracks. *Blobs are selected as new players if $100 < \text{area} < 400$, detect player algorithm has found a player within 0.5m and other already tracked players are more than 2m away

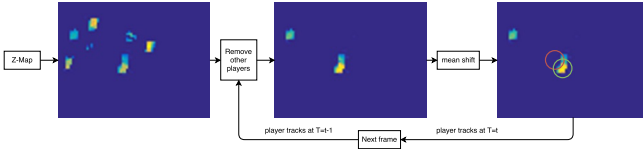


Fig. 9: Tracking players using the mean shift algorithm. First the other players are removed such that mean shift will not end up on those positions. Then the mean shift algorithm finds the new player location by shifting a box to the highest mode in the area (shift from the red start position to the green end position)

V. ADDITIONS TO THE ALGORITHM

The described algorithm is taken as the basis for improvements with several new additions. The main focus of these additions is on reducing the number of identity (ID) switches, since these type of errors have the most impact on the correctness of the data.

Where the basic algorithm only makes use of the foreground images converted to Z-Map, the proposed algorithm also incorporates features from RGB images. This is done as in figure 10. By calculating an HSV and HOG based appearance model of a player and comparing this to extracted crops at possible player positions both color and texture/shape information are taken into account. Furthermore a motion model has been added to get more accurate start locations for the tracking step in the next frame.

In this section the addition of the appearance model, its update policy and the calculation of likelihood maps is described. Furthermore it is given how the likelihoods from the different cameras are combined and how these combined likelihood maps are used for tracking. Next to this there are several separate additions, the forward-backward search for the best player configuration, the addition of a motion model, an improved background model update policy, annealing mean-shift instead of normal mean-shift, incorporating the motion model in the likelihood calculation and finally a check if two tracks have switched identity if the resulting configuration from tracking is unlikely.

A. Appearance model

Since we want to get a per camera per player likelihood of a player standing at a certain position we also need a per player per camera appearance model. How and when the appearance model is generated and updated can be found in the "update policy" section below.

For now let's assume that a player image - extracted using a bounding box of 180x40cm - is given in a certain camera as in figure 11. From this image we can calculate the appearance model feature vector for both HSV and HOG as described below.

When tracking we would like to compare the feature vector of the appearance model to the feature vector of the possible player positions. Therefore a sliding window is applied with a 1 pixel stepsize and a range of 1m around the expected player position. For each of these windows the feature vector f_t is calculated as in figure 11.

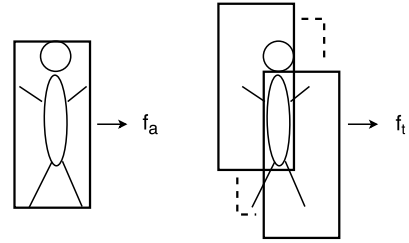


Fig. 11: Appearance Model, left an extracted player image on which the appearance model is based. Right, during test time a sliding window is applied to generate feature vectors used for the similarity calculation

HSV features: From the extracted images as seen in figure 11 we can calculate the HSV feature vector f_a as a 60 bin independent HSV histogram - 20 bins per H, S and V channel - with 3 height bins resulting in a vector length of 180 as seen in table I. Three height bins are chosen since this gives a good separation of players shirts and trousers whereas it is still robust for changes in player stance.

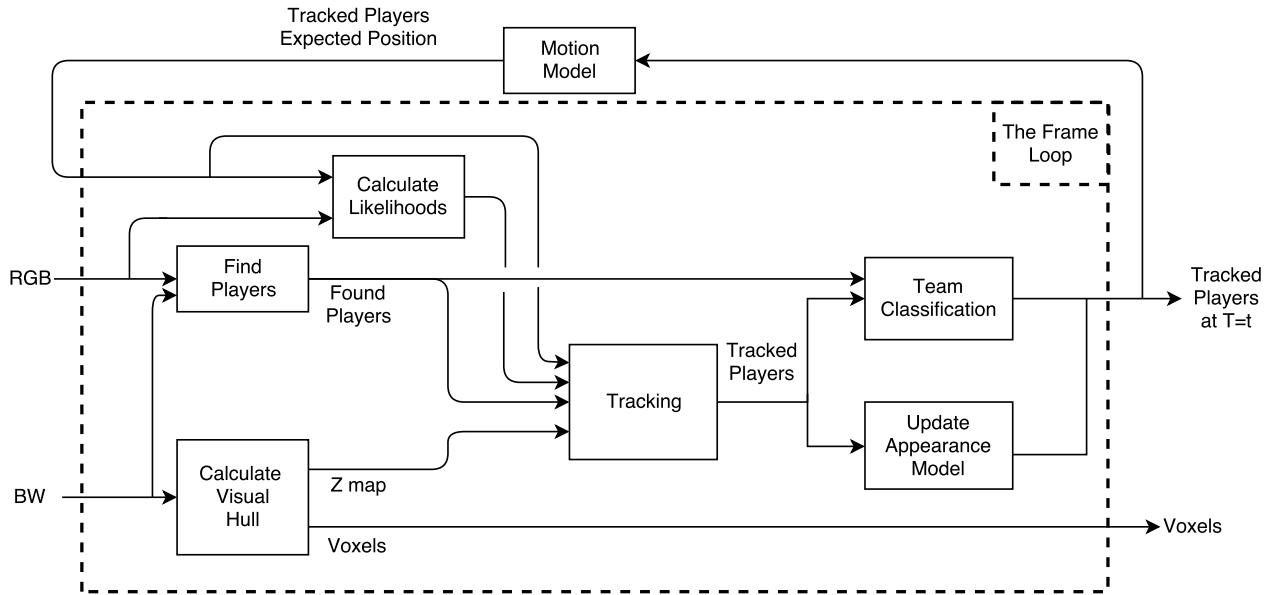


Fig. 10: Overview of the player tracking algorithm with additions (appearance model, motion model and likelihood calculation). Further additions not visible in this flowchart include a forward-backward search, annealing mean-shift and motion model likelihood in the tracking part of the algorithm. Also not visible is the new update policy for background subtraction and the check if two tracks switched identity.

Height Bin 0:60 cm			Height Bin 61:120 cm			Height Bin 121:180 cm		
H	S	V	H	S	V	H	S	V

TABLE I: HSV + height bins feature vector, each H, S and V histogram is independently calculated and consists of 20 bins per channel. Thus resulting in each height bin consisting of 60 values.

HOG features: The HOG features are calculated over the same images as used for the HSV calculation. Calculating the HOG features is done by aggregating HOG features over 6 horizontal and 30 vertical blocks. Where each block consists of 9 signed orientation HOG features. Giving a total feature length of $30 \cdot 6 \cdot 9 = 1620$. The number of horizontal and vertical blocks is chosen such that the feature is the most sensitive in both vertical and horizontal sliding direction while still being able to calculate the feature vector - players can be as small as 6 pixels in width - and having a reasonable length of the feature vector such that the computation is fast enough.

B. Update Policy

To initialize and update the per player per camera appearance model some rules are set to make it more likely that the appearance model is correct. For a good view of the player we need to know that the player is visible in that camera and not occluded by other players.

Determining if a player is visible in a camera can be done using the player position and the camera coverage. Occlusions from other players can be determined by calculating the overlap of bounding boxes between players using the camera parameters and expected player positions. For this

an average player bounding box with a width of 40 cm and height of 180 cm is taken. To make sure we do not miss players which might be standing in front of the player the appearance model can only be updated if at least 16 players are being tracked (this check normally fails only during the initialization phase of the algorithm).

If all those conditions are satisfied the player image can be extracted and from this the feature vector will be calculated. A summary of the appearance model update policy can be seen in algorithm 2.

```

if number of players tracked  $\geq 16$  then
  for All tracked players do
    for All cameras do
      if Visible in that camera then
        if last update  $> 100$  frames ago then
          if  $\leq 15$  percent of the player
            occluded then
            Extract player image;
             $f_a \leftarrow$  Calculate feature vector;
            Appearance Model  $\leftarrow f_a$ ;
          end
        end
      end
    end
  end
end

```

Algorithm 2: Appearance model update policy

C. Likelihood Generation

For every camera in which the player is visible the sliding window method from figure 11 is applied to generate feature vectors at the possible player positions. From these feature vectors the likelihood that a player is standing on position (x,y) is determined by calculating the similarity of the feature vectors using the Bhattacharyya coefficient:

$$\tilde{\mathbf{f}} = \frac{\mathbf{f}}{\sum_{k=1}^n f_k}, \mathbf{f} = \langle f_1, f_2, \dots, f_{n-1}, f_n \rangle \quad (1)$$

$$S = \sum_{k=1}^n \sqrt{\tilde{f}_k^a \cdot \tilde{f}_k^t}, S \in \mathbb{R} : 0 \leq S \leq 1 \quad (2)$$

In the first step the feature vectors \mathbf{f} are normalized such that a different number of pixels (e.g. player is standing closer or further away from the camera with respect to when the appearance model was made) has no influence on the similarity score and such that the similarity score will always be between 0 and 1. In the second step the similarity score S is calculated between the appearance model feature vector $\tilde{\mathbf{f}}^a$ and the test feature vector $\tilde{\mathbf{f}}^t$.

The similarity scores calculated in this way can then be non-linearly scaled to provide the best information. We would like to have high values and more variance the more certain we are that a player is at a certain location. Because in this way the cameras with the most information contribute the most to the total likelihood. Next to this we set three points where the similarity score should map to:

- The similarity score should map to 0 at the location where the true values occurrence is close to zero. This is at 0.3 for HOG and at 0.6 for HSV.
- The similarity score should map to 0.5 at the location where the true values occurrence is at its top. This is at about 0.65 for HOG and at 0.9 for HSV.
- The similarity score should map to 1 if the similarity score is 1.

A mapping which satisfies these requirements is given in figure 12. Because HOG features show a peak in occurrence at about 0.6 already it is chosen to use a straight line for the mapping in order to already get a high variance in the most occurring region without sacrificing variance at higher similarity scores. Since HSV likelihoods show a much more distinct peak in occurrence it is chosen to have a higher slope in the upper region giving even more significance to these values.

D. Combining Likelihoods

Ultimately we want to know $\mathcal{L}_{x,y}(A)$, the likelihood that player A is at location (x,y) . To get this value the likelihoods from the Z-Map and those from the HSV and HOG likelihoods of different cameras need to be combined. This is done as in equation 3, where $\mathcal{L}_{x,y}(A|Z_{map_{x,y}})$ is the normalized Z-Map, c_e is the number of cameras for which player A is visible and for which the player is expected to be less than 80% occluded, $\mathcal{L}_{x,y,c}(A|S_{HSV})$ is the likelihood of player A being at location (x,y) given the HSV similarity

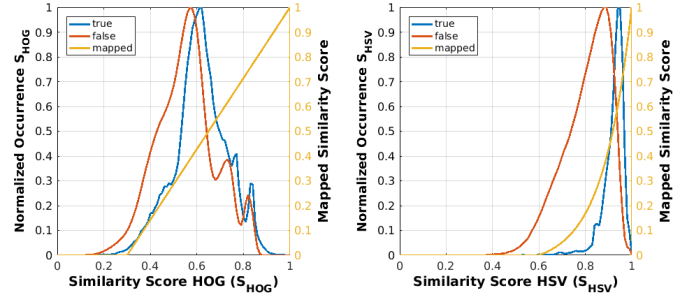


Fig. 12: Normalized (divided by maximum value) occurrence similarity score values vs the similarity score for, blue line) good localizations (i.e. within 5 cm of the target), red line) non player (i.e. more than 15 cm away from the target) and the similarity score mapping based on these values

score from camera c and $\mathcal{L}_{x,y,c}(A|S_{HOG})$ is the same given the HOG similarity score. The value $\frac{1}{2 \cdot c_e}$ is to ensure that the likelihood values stay between 0 and 1, independent of the number of cameras.

$$\mathcal{L}_{x,y}(A) = \mathcal{L}_{x,y}(A|Z_{map_{x,y}}) \cdot \frac{1}{2 \cdot c_e} \sum_{c=1}^{c_e} (\mathcal{L}_{x,y,c}(A|S_{HSV}) + \mathcal{L}_{x,y,c}(A|S_{HOG})) \quad (3)$$

The likelihoods are combined in this fashion for several reasons. First multiplying with the Z-Map ensures locations where the player is definitely not standing have a likelihood of 0 assuming the Z-Map always has values higher than 0 at player locations. Secondly adding the HOG and HSV likelihood instead of multiplying automatically ensures that the best appearance models have the most impact on the likelihood score. Scores of both an "old" appearance model and those given by cameras in which the player is (partly) occluded will be lower than their counterparts, while the variance might still be high. Multiplying these scores would thus not show the desired behavior.

E. Tracking

Tracking is similar as before since the mean-shift algorithm works equally well on likelihood maps as Z-Maps. To reduce the computational load the likelihood maps are only generated for players which are within 1.5 meter of another player, otherwise only the Z-Map is used for that player.

F. Forward-Backward Search

Because the implemented tracking algorithm is inherently sequential, players which are tracked first are more likely to get the best position and steal the track of another player. To reduce this effect a forward-backward search is implemented. This is done by both doing a forward and reverse ordered tracking step and choosing the locally optimal combination. The locally optimal solution is set to be the solution for which the combined likelihood for players which are standing within 0.8 meter from each other is the

highest. A player likelihood is defined as the sum of the likelihood values ($\mathcal{L}_{x,y}$) within a radius of 0.25 meter of the tracked (forward or backward) player position times the motion likelihood (\mathcal{L}_m , see section V-G). In formulas the forward search likelihood is calculated using:

$$\begin{aligned} \mathcal{L}_{f,i} = & \mathcal{L}_{m_i} \left(\sum_{x,y} \mathcal{L}_{x,y}(i) \cdot \mathcal{L}_{m_i} + \right. \\ & \sum_j (D(x_i, x_j) \cdot \mathcal{L}_{m_j} \cdot \sum_{xx,yy} \mathcal{L}_{xx,yy}(j)) \\ & \{x, y : (x - x_i)^2 + (y - y_i)^2 \leq 0.25^2\} \\ & \{xx, yy : (xx - x_j)^2 + (yy - y_j)^2 \leq 0.25^2\} \\ & \{j \in N : j \leq p, j \neq i\} \\ & \left. D(x_i, x_j) \begin{cases} 1 & \text{if } (x_j - x_i)^2 + (y_j - y_i)^2 < 0.8^2 \\ 0 & \text{otherwise} \end{cases} \right) \end{aligned} \quad (4)$$

Where $\mathcal{L}_{f,i}$ is the forward likelihood of player i ; x_i, y_i is the forward search player position of player i and p is the number of players. The calculation if a player is standing in the vicinity is for both the forward and backward search determined by the backward search location, because otherwise it could be possible that there are more players close by in the forward search with respect to the backward search or the other way around.

For the backwards likelihood the calculation is similar only changing the motion likelihood and x_i, y_i and x_j, y_j to the backward search location. A players position is then determined by

$$x, y = \begin{cases} x_f, y_f & \text{if } \mathcal{L}_{f,i} \geq \mathcal{L}_{b,i} \\ x_b, y_b & \text{if } \mathcal{L}_{f,i} < \mathcal{L}_{b,i} \end{cases} \quad (5)$$

Where x_f, y_f is the forward search player location and x_b, y_b is the backward search player location.

This implementation improves the player tracking algorithm by making a large jump in player position far less likely.

G. Motion Model

It is better to start the tracking given the expected location than using the old location. Since the tracking history for a player is known this can be calculated. While most tracking algorithms make use of a Kalman filter, the irregular movements of soccer players in both speed, acceleration and direction degrade the performance of the Kalman filter too much to be useful. It is therefor chosen to implement a simple constant velocity motion model smoothed over the last 15 frames as in equation 6. Where $\mathbf{xy}(t)$ is the position of a player at time t .

$$\mathbf{xy}(T+1) = \frac{1}{14} \cdot \sum_{t=T-13}^T (\mathbf{xy}(t) - \mathbf{xy}(t-1)) \quad (6)$$

To also incorporate the fact that players cannot walk through each other and collide often, a constraint is added

that the expected location of a player cannot be within 0.4 meter of another player. If this is the case using the motion model, the expected player locations of the colliding players will be gradually returned in small steps to their position at $t=T$ until the expected locations are more than 0.4 meter away from each other.

Using the motion model a motion model likelihood can be set as well, penalizing locations which are further away from the expected location. The motion model likelihood is calculated as in equation 7, where d is the distance between the tracked and expected location. This likelihood is used in the forward-backward calculation to calculate the optimal configuration.

$$\mathcal{L}_m = \begin{cases} 1 & \text{if } d < 0.4 \\ 1.5 - (d + 0.6)^2 & \text{if } 0.4 \leq d \leq 0.62 \\ 0 & \text{if } d > 0.62 \end{cases} \quad (7)$$

H. Background Model Update Policy

Because players standing still will over time be incorporated in the background model and therefor disappear as foreground - and thus disappear in the Z-Map - the background subtraction algorithm is adjusted. By not updating the background model around a player - defined as a bounding box of 180x40cm plus 20 pixels at the last player location - players will no longer disappear in the background, subsequently leading to fewer misses.

I. Annealing Mean-Shift

Annealing mean-shift is a form of mean-shift where the mean-shift algorithm is repeated with a smaller window size (i.e. 70% smaller each step) after the previous mean-shift has converged. This is repeated for four times. This form of mean-shift is both robust and accurate. Robust because it first uses a large search window to find the global mode and precise by subsequently centering a smaller window exactly atop the highest mode.

J. Check for ID Switch

If the motion likelihood is lower than 0.75 there is the probability that two players have switched tracks. To check if this is indeed the case, the likelihood of the configuration where the two possibly switched players are located is compared to the configuration where they have switched locations. The likelihood of the configuration (\mathcal{L}_c) is calculated as in equation 8. Where $\mathcal{L}_{x,y}(i)$ and $\mathcal{L}_{xx,yy}(j)$ are the likelihood maps for player i and j respectively, (x_i, y_i) is the location of player i and (x_j, y_j) is the location of player j . For the switched position calculation (i) is replaced with (j) and vice versa.

$$\begin{aligned} \mathcal{L}_c = & \sum_{x,y} \mathcal{L}_{x,y}(i) + \sum_{xx,yy} \mathcal{L}_{xx,yy}(j) \\ & \{x, y : (x - x_i)^2 + (y - y_i)^2 \leq 0.25^2\} \\ & \{xx, yy : (xx - x_j)^2 + (yy - y_j)^2 \leq 0.25^2\} \end{aligned} \quad (8)$$

If the switched position is more likely than the non switched position, the locations of the players are switched. This is done for 12 subsequent frames - about 0.5 seconds - after the first ID switch check to enlarge the possibility that the players are better separated and that the cause which leaded to the switch in the first case is not there anymore. During this time interval the appearance model update step for the involved players is postponed.

VI. EXPERIMENTS AND RESULTS

Since the algorithm is specifically tailored on the camera setup given in section III, it can only be used on data from this setup. Furthermore the player tracking algorithms given in the literature are made for other specific setups and are not publicly available. It is therefor chosen to only report the results on the individual additions of the algorithm such that the value of each component becomes clear.

A. Dataset

The performance of the algorithm is assessed on three datasets of which the first two have also been used during development for parameter estimation. The first is 40 seconds (1000 frames) footage of the match Heracles-Utrecht, the second is 30 seconds footage (750 frames) of the match Heracles-Twente and the third is 15 seconds footage (375 frames) of the match Heracles-Utrecht. All datasets are chosen based on the numerous occlusions that occur.

Each dataset has been manually annotated every 5 frames using the overview camera with the best view. Between annotations, tracks are interpolated to get a per frame performance. Players which are outside the field of view of the annotation camera are not considered in the results. The first 125 frames are needed for initializing the algorithm and are therefor also not considered.

B. Results

The performance of the algorithm is assessed using the clear MOT metrics, which are the standard in multi person tracking [51]. For this paper the track association distance is set to 1 meter for tracks which are within 20 meter in the y-direction of the goal and to 2.5 meter further away. The performance metrics consist of the multiple object tracking precision (MOTP), calculated as:

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \quad (9)$$

Where d_t^i is the distance between hypothesis i and its corresponding annotated track and c_t is the number of matches found for time t, where only the tracks within 20 meter in the y-direction of the goal are counted. This measure shows the accuracy of the tracker independent of the errors the tracker makes.

Furthermore the multiple object tracking accuracy (MOTA) is used, which accounts for all object configuration

errors - i.e. false positives (fp_t), identity switches (id_t) and misses (m_t) - except global mismatches and is calculated as:

$$MOTA = 1 - \frac{\sum_t (fp_t + id_t + m_t)}{\sum_t g_t} \quad (10)$$

Where g_t is the number of ground truth tracks at time t.

As last a new measure is introduced, called the expected error per frame (EPPF). This measure shows the number of errors one can expect to see when viewing a single frame from the radar view. An EPPF of 1 thus means that in every frame one thing is expected to be wrong. Which can be a wrong identity, a miss, a false positive or a global mismatch. This also means that for a normal soccer match with 23 tracks (11 per team and one referee) and an EPPF of 1, 22 out of 23 tracks are expected to be correct at any time instance. The EPPF is calculated as:

$$EPPF = \frac{\sum_t (400 \cdot id_t + fp_t + gmm_t + m_t) + 125 \cdot gmmT}{\sum_t 1} \quad (11)$$

Where $gmmT$ is the number of tracks with more than 125 consecutive global mismatches and gmm_t is the number of global mismatches at time t only counting tracks with less than 126 consecutive global mismatches. The weights are set to the above values on the assumptions that 1) an ID switch can be detected and corrected after approximately 400 frames (20 sec) with a shirt number recognition algorithm 2) global mismatches can be detected after approximately 125 frames (5 sec) by checking that there are more tracks than possible and two tracks are continuously at about the same position. This error rate is therefor the error rate for an online tracker, using the same algorithm offline or with a delay of at least 20 seconds makes it possible to correct tracks after detecting the ID switch or GMM error.

TABLE II: The additions which a certain version of the algorithm makes use off. Version A is the original algorithm at the start of the thesis.

Additions \ Version	A	B	C	D	E	F	G	H	I	J	K
HSV Features		x		x	x	x	x	x	x	x	x
HOG Features			x	x	x	x	x	x	x	x	x
Forward-Backward Search					x	x	x	x	x	x	x
Motion Model						x	x	x	x	x	x
Background Model Update Policy							x	x	x	x	x
Annealing Mean-Shift								x	x	x	
Motion Model Likelihood									x	x	x
ID switch check										x	x

In table III the results on the three different datasets are given, as well as the average performance over the three datasets. In figure 13 the EPPF versus MOTA is displayed for the averaged datasets. It can immediately be found that the algorithm with all the additions performs the best with an EPPF of just 0.15. Furthermore using both the HOG and HSV features improves the performance over just using one of the two type of features, showing that the information is complementary. The only addition which degrades the performance on all three datasets is annealing mean-shift,

using all additions except for this one slightly improves the performance of the algorithm (version K). Although it should be noted that it reduces the performance on the validation dataset.

TABLE III: Results for different versions of the algorithm, 1-3 are the different datasets, C is the average of the three datasets, MOTA is displayed as MOTA·100 for readability

	A	B	C	D	E	F	G	H	I	J	K
1 MOTP	0.08	0.08	0.09	0.08	0.08	0.08	0.08	0.08	0.09	0.09	0.08
1 MOTA	99.93	99.92	99.89	99.86	99.96	99.89	99.90	99.80	99.93	99.95	99.96
1 EEPF	0.94	1.55	1.41	1.46	0.02	0.94	0.94	1.88	0.16	0.16	0.01
2 MOTP	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
2 MOTA	98.06	99.79	98.98	99.90	99.90	99.90	99.90	99.81	99.79	99.89	99.90
2 EEPF	4.83	0.05	4.10	0.02	0.02	0.02	0.02	1.32	0.05	0.03	0.02
3 MOTP	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
3 MOTA	97.38	97.15	97.10	97.36	96.95	98.87	97.96	97.71	97.32	98.87	98.47
3 EEPF	3.79	3.84	4.06	3.79	5.60	0.26	3.66	5.48	2.38	0.26	0.35
C MOTA	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
C EEPF	3.19	1.81	3.19	1.76	1.88	0.41	1.54	2.89	0.86	0.15	0.13

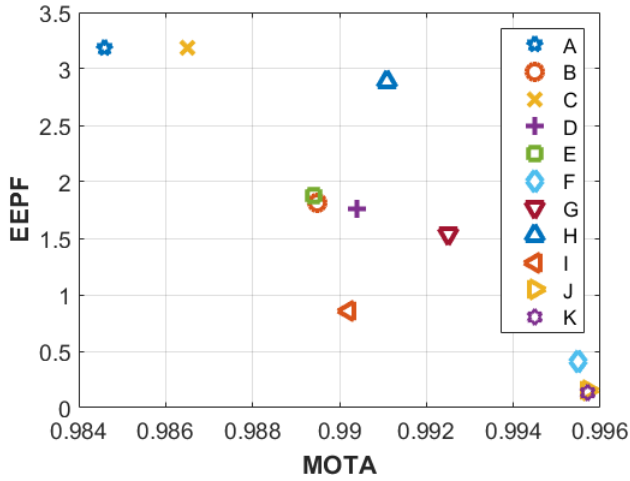


Fig. 13: MOTA against EEPF, values are the average of the three datasets

VII. CONCLUSION

Accurate tracking of soccer players can greatly improve statistics used for coaching advisory, scouting, media and betting. Several companies have attempted to tackle the tracking problem with varying success. The most difficult problem - reducing the number of identity switches in crowded situations - had yet to be solved. The main cause for the identity switch is similar looking players occluding each other while their movement changes suddenly and thus cannot be predicted with the common Kalman filter.

In this paper the use of multiple cameras - 14 in total - increases the probability of a camera having a non occluded view of the player. Switching to the top view after calculating the 3D space further reduces the occlusion problem, since it is highly unlikely that players stand on top of one another. The introduction of a per camera HOG and HSV based appearance model enables the tracker to use both color and texture/shape. While a smart combination of the features

from the different cameras gives a higher weight to cameras with a good view and less degraded appearance model. This combination is based on a mapping of the bhattacharyya similarity coefficient between the appearance model and sliding window features to more informative values. Cameras are combined by adding their respective mapped HOG and HSV similarity scores. This score is then multiplied with the 3D map - which has been summed in the z-direction - to get a more accurate localization. This approach increases the MOTA from 0.9846 to 0.9904 and reduces the expected error per frame (EEPF) from 3.19 to 1.76 with respect to the basic algorithm. Which means a performance increase of one error (false positive, miss or ID switch) per 65 ground truth annotations to one per 104.

Several other additions including forward-backward search, a motion model, a new background model update policy and a check if players have switched ID further increases the MOTA to 0.9944 and reduces the EEPF to just 0.13, a performance increase to one error per 179 ground truth annotations. If the tracker is deployed as an offline tracker or a $\geq 20s$ delayed tracker these values will improve even more.

With the addition of a shirt number recognition engine, the proposed algorithm is able to satisfy most of the use case requirements and can be deployed in real world applications.

REFERENCES

- [1] "Opta," <http://www.optasports.com/>, accessed: 2016-11-20.
- [2] "Sentiosports," <https://sentiosports.com/>, accessed: 2016-11-20.
- [3] "prozone," <http://www.stats.com/football>, accessed: 2016-11-20.
- [4] "Tracab," <http://chyronego.com/sports-data/tracab>, accessed: 2016-11-20.
- [5] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Mottchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.
- [6] "2d mot challenge 2015," https://motchallenge.net/results/2D_MOT_2015/, accessed: 2016-11-20.
- [7] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 952–957.
- [8] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*. Springer, 2006, pp. 133–146.
- [9] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.
- [10] Y. Ohno, J. Miura, and Y. Shirai, "Tracking players and estimation of the 3d position of a ball in soccer games," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1. IEEE, 2000, pp. 145–148.
- [11] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multimodality through mixture tracking," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1110–1116.
- [12] M. Xu, J. Orwell, L. Lowey, and D. Thirde, "Architecture and algorithms for tracking football players with multiple cameras," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 232–241, 2005.
- [13] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European Conference on Computer Vision*. Springer, 2004, pp. 28–39.
- [14] T. Yu and Y. Wu, "Collaborative tracking of multiple targets," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. 1–834.

- [15] J. Sullivan and S. Carlsson, "Tracking and labelling of interacting multiple targets," in *European Conference on Computer Vision*. Springer, 2006, pp. 619–632.
- [16] M. Beetz, S. Gedikli, J. Bandouch, B. Kirchlechner, N. von Hoyningen-Huene, and A. C. Perzylo, "Visually tracking football games based on tv broadcasts," in *IJCAI*, 2007, pp. 2066–2071.
- [17] P. Figueroa, N. Leite, R. M. Barros, I. Cohen, and G. Medioni, "Tracking soccer players using the graph representation," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 787–790.
- [18] W. Du and J. Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Asian Conference on Computer Vision*. Springer, 2007, pp. 365–374.
- [19] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 103–113, 2009.
- [20] E. Morais, S. Goldenstein, A. Ferreira, and A. Rocha, "Automatic tracking of indoor soccer players using videos from multiple cameras," in *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2012, pp. 174–181.
- [21] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 240–247.
- [22] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Visual tracking of numerous targets via multi-bernoulli filtering of image data," *Pattern Recognition*, vol. 45, no. 10, pp. 3625–3635, 2012.
- [23] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 3, pp. 505–519, 2009.
- [24] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 137–144.
- [25] T. Misu, M. Naemura, W. Zheng, Y. Izumi, and K. Fukui, "Robust tracking of soccer players based on data fusion," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1. IEEE, 2002, pp. 556–561.
- [26] H. Li and M. Flierl, "Sift-based multi-view cooperative tracking for soccer video," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 1001–1004.
- [27] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.
- [28] R. Hamid, R. Kumar, J. Hodgins, and I. Essa, "A visualization framework for team sports captured using multiple static cameras," *Computer Vision and Image Understanding*, vol. 118, pp. 171–183, 2014.
- [29] J. Liu, P. Carr, R. T. Collins, and Y. Liu, "Tracking sports players with context-conditioned motion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1830–1837.
- [30] E. Morais, A. Ferreira, S. A. Cunha, R. M. Barros, A. Rocha, and S. Goldenstein, "A multiple camera methodology for automatic localization and tracking of futsal players," *Pattern Recognition Letters*, vol. 39, pp. 21–30, 2014.
- [31] I. Kazuya, S. Takahashi, T. Ogawa, and M. Haseyama, "Player tracking in far-view soccer videos based on composite energy function," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 7, pp. 1885–1892, 2014.
- [32] S. Baysal, "Model field particles with positional appearance learning for sports player tracking," Ph.D. dissertation, bilkent university, 2016.
- [33] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1. IEEE, 2002, pp. 303–306.
- [34] G. Zhu, C. Xu, Q. Huang, and W. Gao, "Automatic multi-player detection and tracking in broadcast sports video using support vector machine and particle filter," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1629–1632.
- [35] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Image and Vision Computing*, vol. 25, no. 8, pp. 1271–1281, 2007.
- [36] A. Dearden, Y. Demiris, and O. Grau, "Tracking football player movement from a single moving camera using particle filters," in *Proceedings of the 3rd European Conference on Visual Media Production (CVMP)*, London, 2006, pp. 29–37.
- [37] 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," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 2442–2446.
- [38] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-target tracking-linking identities using bayesian network inference," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 2187–2194.
- [39] P. J. Figueroa, N. J. Leite, and R. M. Barros, "Tracking soccer players aiming their kinematical motion analysis," *Computer Vision and Image Understanding*, vol. 101, no. 2, pp. 122–135, 2006.
- [40] R. Martín and J. M. Martínez, "A semi-supervised system for players detection and tracking in multi-camera soccer videos," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 1617–1642, 2014.
- [41] M. Schlippsing, J. Salmen, M. Tschentscher, and C. Igel, "Adaptive pattern recognition in real-time video-based soccer analysis," *Journal of Real-Time Image Processing*, pp. 1–17, 2014.
- [42] S. Iwase and H. Saito, "Parallel tracking of all soccer players by integrating detected positions in multiple view images," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 751–754.
- [43] —, "Tracking soccer players based on homography among multiple views," in *Visual Communications and Image Processing 2003*. International Society for Optics and Photonics, 2003, pp. 283–292.
- [44] M. Leo, N. Mosca, P. Spagnolo, P. L. Mazzeo, T. D'Orazio, and A. Distant, "Real-time multiview analysis of soccer matches for understanding interactions between ball and players," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*. ACM, 2008, pp. 525–534.
- [45] J. Ren, M. Xu, J. Orwell, and G. A. Jones, "Multi-camera video surveillance for real-time analysis and reconstruction of soccer games," *Machine Vision and Applications*, vol. 21, no. 6, pp. 855–863, 2010.
- [46] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [47] S. Gedikli, J. Bandouch, N. von Hoyningen-Huene, B. Kirchlechner, and M. Beetz, "An adaptive vision system for tracking soccer players from variable camera settings," in *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS)*, 2007.
- [48] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.
- [49] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 2, pp. 150–162, 1994.
- [50] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.
- [51] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.
- [52] J. Liu and P. Carr, "Detecting and tracking sports players with random forests and context-conditioned motion models," in *Computer Vision in Sports*. Springer, 2014, pp. 113–132.
- [53] S. Baysal and P. Duygulu, "Sentioscope: a soccer player tracking system using model field particles," 2015.
- [54] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, and M. Nitti, "A visual tracking algorithm for real time people detection," in *Image Analysis for Multimedia Interactive Services, 2007. WIAMIS'07. Eighth International Workshop on*. IEEE, 2007, pp. 34–34.
- [55] P. L. Mazzeo, P. Spagnolo, M. Leo, and T. D'Orazio, "Visual players detection and tracking in soccer matches," in *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*. IEEE, 2008, pp. 326–333.
- [56] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, M. Nitti, and A. Distant, "An investigation into the feasibility of real-time soccer offside detection from a multiple camera system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 12, pp. 1804–1818, 2009.

APPENDIX

A. Soccer player tracking literature overview

In this appendix an overview of the literature on soccer tracking is given. It is specified for every paper what type of cameras are used: static cameras, static cameras with an overlapping field of view or broadcast cameras. Also the number of used cameras is given. Next to this the tracking method is described shortly, as well as the appearance model and constraints used for tracking. Since handling occlusions is especially important for soccer tracking it is stated what kind of occlusion handling method is used. And finally a performance score is given to each paper indicating how well the algorithm would likely perform based on the results and conclusion published in the paper and the description of the algorithm. See tables V, VI and VII.

Performance: To give an idea about the performance of the algorithms and the scenarios which need to be solved in order to get a perfect tracking algorithm the papers with the best tracking algorithms are reviewed for handling different scenarios. The scenarios are compiled from possibly occurring situations in soccer and specified on the main difficulties the trackers may need to handle.

The scenarios include several aspects: the number of players involved, where more players in the same area are generally harder to handle. The team to which the players belong, players from different teams can normally be separated using an appearance model based on color while players from the same team cannot be that easily separated. The type of occlusion, full or partial where at least 25 percent of the occluded player is always visible. If there is interaction between the players, e.g. the players bump into each other or touch each other. And finally the type of motion, similar or distinct. Where distinct motion is a type of motion which can be solved by a motion model (i.e. Kalman filter or similar) and similar motion cannot due to the players walking in the same direction with the same speed or having almost no speed at all.

The fifteen scenarios common in soccer are:

- S1: Single player, not occluded
- S2: Two players, different team, partial occlusion, interacting, similar motion
- S3: Two players, different team, full occlusion, interacting, similar motion
- S4: Two players, same team, partial occlusion, not interacting, distinct motion
- S5: Two players, same team, partial occlusion, not interacting, similar motion
- S6: Two players, same team, partial occlusion, interacting, similar motion
- S7: Two players, same team, full occlusion, not interacting, distinct motion
- S8: Two players, same team, full occlusion, not interacting, similar motion
- S9: Two players, same team, full occlusion, interacting, similar motion

- S10: Three players, same team, partial occlusion, non interacting, distinct motion
- S11: Three players, same team, partial occlusion, non interacting, similar motion
- S12: Three players, same team, partial occlusion, interacting, similar motion
- S13: Four players, partial occlusion, non interacting, similar motion
- S14: Four players, partial occlusion, interacting, similar motion
- S15: Five or more players interacting

TABLE IV: Performance of soccer tracking algorithms per scenario

	Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Vermaak et al. [11]	x	x	x	x	x										
Beetz et al. [16], [47]	x	x	x	x	x		x			x					
Du et al. [18]	x	x	x	x	x										
Morais et al. [20]	x	x	x	x	x	x	x			x					
Hess et al. [21]	x	x	x	x	x	x	x			x					
Misu et al. [25]	x	x		x	x					x					
Liu et al. [29], [52]	x	x	x	x	x		x			x					
Morais et al. [30]	x	x	x	x	x	x	x			x					
Kazuya et al. [31]	x	x	x	x	x		x			x					
Baysal [32], [53]	x	x	x	x	x	x	x			x					

As can be seen in table IV the most easy and common cases for soccer tracking have been solved in multiple papers. Tracking similar appearing players through occlusions who have a distinct motion pattern can easily be solved by a motion model. While tracking players of different teams through full occlusion can be handled by an appearance model. However when two or more players from the same team enter an occlusion with similar motion (same speed and direction or standing still, a common scenario in free-kicks and corners) most algorithms will have difficulties tracking the correct player.

To get a fully automatic player tracking system it is important that also these scenarios can be solved by the tracker. By making use of a multi-camera setup these situations can be solved more easily because occlusions in one camera can be solved from another camera where the object is less or not occluded. The algorithm presented in this paper is therefore able to solve all scenarios except S15.

TABLE V: Soccer tracking papers overview part 1

Paper	Type camera	#cameras	Tracking Method	Appearance Model / Constraints	Occlusion handling method	Performance
[10]	Static overlapping	8	Tracking-by-detection by searching for vertical pairs of shirt and pants regions	Shirt / Pants match	Estimation during occlusions using constant velocity, color, vertical position on the field	Low
[42]	Static overlapping	15	Map bottom of blobs to global view, cluster feet in global view	Background Subtraction (BGS)	None	Low
[11]	Dynamic	Broadcast	Mixture particle filter	Normalized color histograms	Multi modality of mixture particle filter	Medium
[12]	Static overlapping	8	Bounding box regression using blobs, kalman filter in both single as total view for estimations	Constant velocity, BGS	Check blobs using color	Low / Medium
[33]	Dynamic	Broadcast	See [10] + estimation of 3D position			
[13]	Dynamic	Broadcast	Boosted Particle Filter (based on [11])	Cascaded Adaboost on a 2 spatial bin, 110 HSV bin histogram	Multi modality of mixture particle filter	Medium
[14]	Static	1	Mean Field Monte Carlo	PCA based	Dynamics and constraint that one observation can only be from one target	Medium
[17]	Static	4	Track graph of blobs with merge and split	BGS	Size, shape, vertical intensity distribution	Low
[43]	Static overlapping	8	Matching blobs	BGS, blob size, color	Matching positions from multiple cameras	Low
[15], [38]	Static	4	Track graph of blobs with merge and split	BGS, position relative to team mates, color, velocity, relative depth	Track graph merge split	Medium
[16], [47]	Dynamic	Broadcast	Blob segmentation and Multiple Hypotheses Tracker (MHT)	BGS based on non-field color, color template with height bins, compactness constraint, height constraint	Blob segmentation and MHT	Medium
[39]	Static	4	Track graph of blobs with merge and split	BGS	Merge and split in track graph; using: blob area, blob perimeter, blob width and height, vertical intensity distribution, direction, velocity	Medium
[34]	Dynamic	Broadcast	Support Vector Regression based particle filter	SVM model	Implicitly done by particle filter	Low
[35]	Dynamic	Broadcast	Adaptive color based particle filter	Color histograms	Implicitly done by particle filter	Low
[36]	Dynamic	Broadcast	Particle filter	chromacity values, velocity	None	Low

TABLE VI: Soccer tracking papers overview part 2

Paper	Type camera	#cameras	Tracking Method	Appearance Model / Constraints	Occlusion handling method	Performance
[18]	Static overlapping	3	Collaborative particle filters with messages between ground plane and image plane	Color histogram, homography	Homography constraint from multiple cameras	Medium
[19]	Dynamic	Broadcast	MCMC data association	Haar filter player detector, velocity, no spatial overlap	Handled with MCMC	Medium
[44], [54]	Static overlapping	6	Hidden Markov Model	BGS, Velocity, Acceleration and Size	Handled with HMM	Low
[55], [56]	Static	1	Solving as a MAP problem	BGS, Velocity, Acceleration and Size	Handled with MAP	Low
[20]	Static overlapping	4	Particle filter with shared observation function and homography mapping with importance based on likelihood	HOG and HSV histograms with three height bins	Implicitly done by particle filter and homography	Medium
[21]	Static	1	Pseudo-independent log-linear particle filters. Single trackers for each object but with knowledge of previous state estimates of other trackers. Weights of feature vector for weighing particles are learned for an optimal combination.	Constant acceleration, motion direction, RGB histogram, blob area, non-overlapping regions, proximity in state space	Done by the features of the particle filter	Medium / High
[22]	Static	1	multi-Bernoulli filter	Learned HSV color histogram with two height bins	Implicitly handled	Medium
[23]	Static overlapping	Multiple	Minimizing an energy function on a multi-plane homography map similar to [12]	Multi plane homography, foreground likelihood	Implicitly handled	Medium / High
[45]	Static overlapping	8				
[24]	Static overlapping	6	Linear Programming approach on a probability occupancy map	Non-overlapping regions, CIE-LAB color histogram	Match in LP using color and motion	Medium
[25]	Static	1	Pattern matching with occlusion weights	Color, texture, texture of head, local motion vectors, chromakey based patch matching	Only use those features which give also info during partial occlusions	Medium
[26]	Static overlapping	3	Finding corresponding SIFT features between views and frames	SIFT on foreground areas	Hope on non-occluded view from other cameras	Medium

TABLE VII: Soccer tracking papers overview part 3

Paper	Type camera	#cameras	Tracking Method	Appearance Model / Constraints	Occlusion handling method	Performance
[27]	Dynamic	Broadcast	Tracking by detection using MCMC on a frame by frame basis (no global optimization)	Deformable part model; Smooth change on appearance (color histogram), location and size; mutual exclusion	Implicitly handled	Medium
[28]	Static overlapping	3	Particle filter based blob tracker and graph matching between views	BGS, motion, color	Implicitly handled	Medium
[37]	Dynamic	Broadcast	Detecting single players, group in tracklets, group tracklets using number recognition	Player outline (BGS), number recognition	No tracking of groups	Low
[29], [52]	Static	1	Tracking by detection and hierarchical clustering of low/mid/high level trajectories	Color, player detections, motion, game context	Clustering algorithm	Medium
[30]	Static overlapping	4	Fusion of data from particle filter from different views with likelihood based on a per camera appearance model projected on an overall view and presented as a Gaussian likelihood distribution with covariance estimated from ground truth measurements	Haar filter based player detection, HOG and HSV histograms with three height bins	Implicitly handled	Medium / High
[40]	Static	1+	Matching blobs	BGS, blob size, color	Matching positions from multiple cameras	Low
[31]	Static	1	Minimizing a Composite Energy Function	Potential Energy: color, hog; Elastic Energy: based on maintaining formation; Movement Direction Based Energy	Implicitly handled	Medium
[32], [53]	Static	2	Model Field Particles	BGS, HoG player recognition, motion using Kalman filter, HSV histogram with height bins	Implicitly handled + occlusion reasoning based on motion and color likelihood + particles can be used by multiple tracks	Medium / High
[41]	Static	2	Tracking by blob detection and Kalman filter	BGS	Human operator	Low

B. Camera Views

The camera views used for the algorithm in this paper are given in figure 14, notice the overview cameras in 14b and 14m.

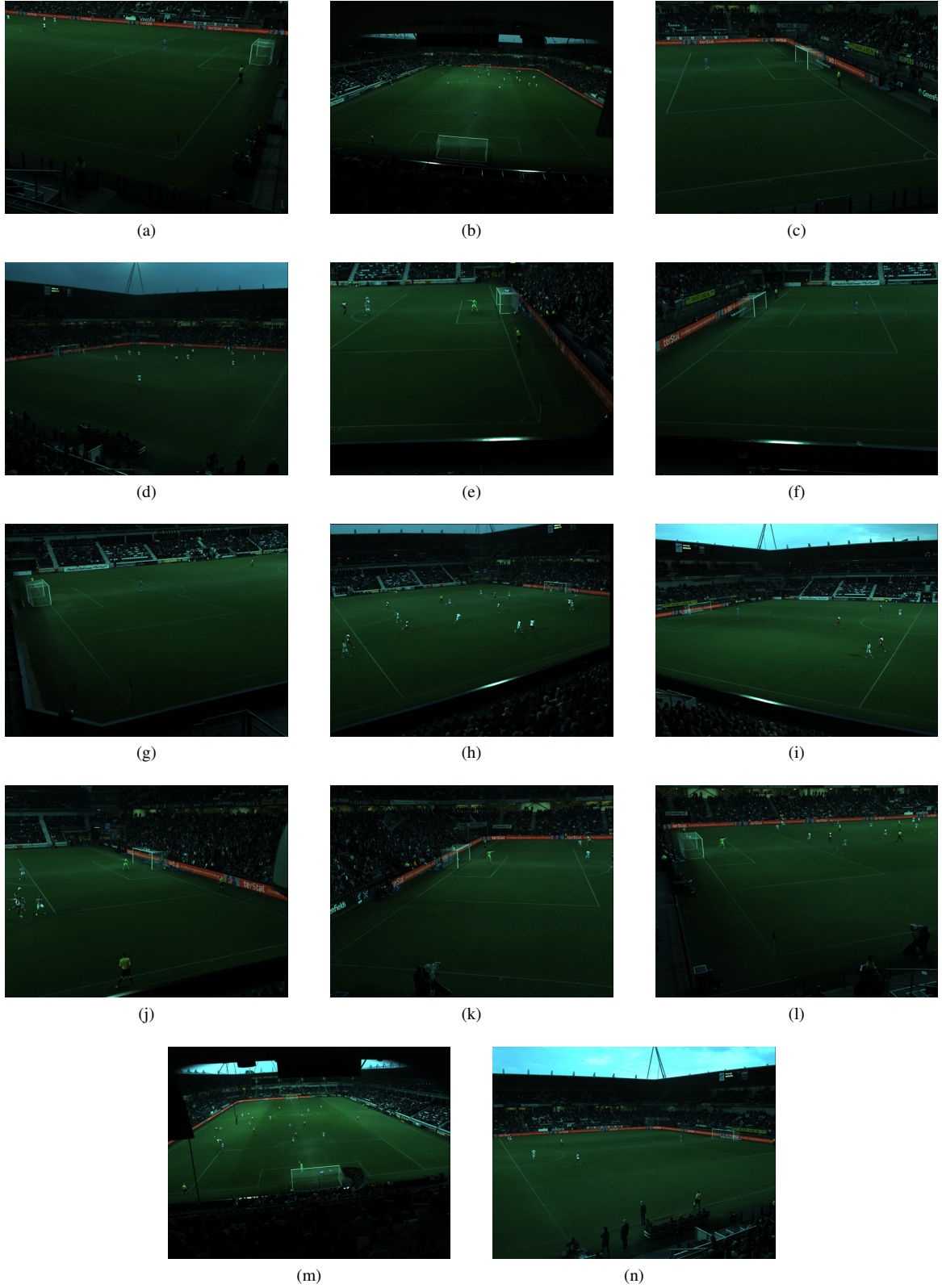


Fig. 14: The different camera views

C. The clear MOT metrics

The clear MOT metrics are the standard for measuring the performance of multi person tracking, the different measures are based on:

- False Positives, the number of hypotheses which do not correspond to a ground truth track
- True Positives, the number of hypotheses which do correspond to a ground truth track
- Misses, the number of ground truth tracks to which no hypothesis can be assigned
- ID switches, the number of hypotheses which track a different person than it was created on. Only counted once for each switch, after which the identity of the hypothesis is set to that of the new target
- Global mismatches, the number of hypotheses which are tracking an already tracked track

In figure 15 it can be found how the data association is done.

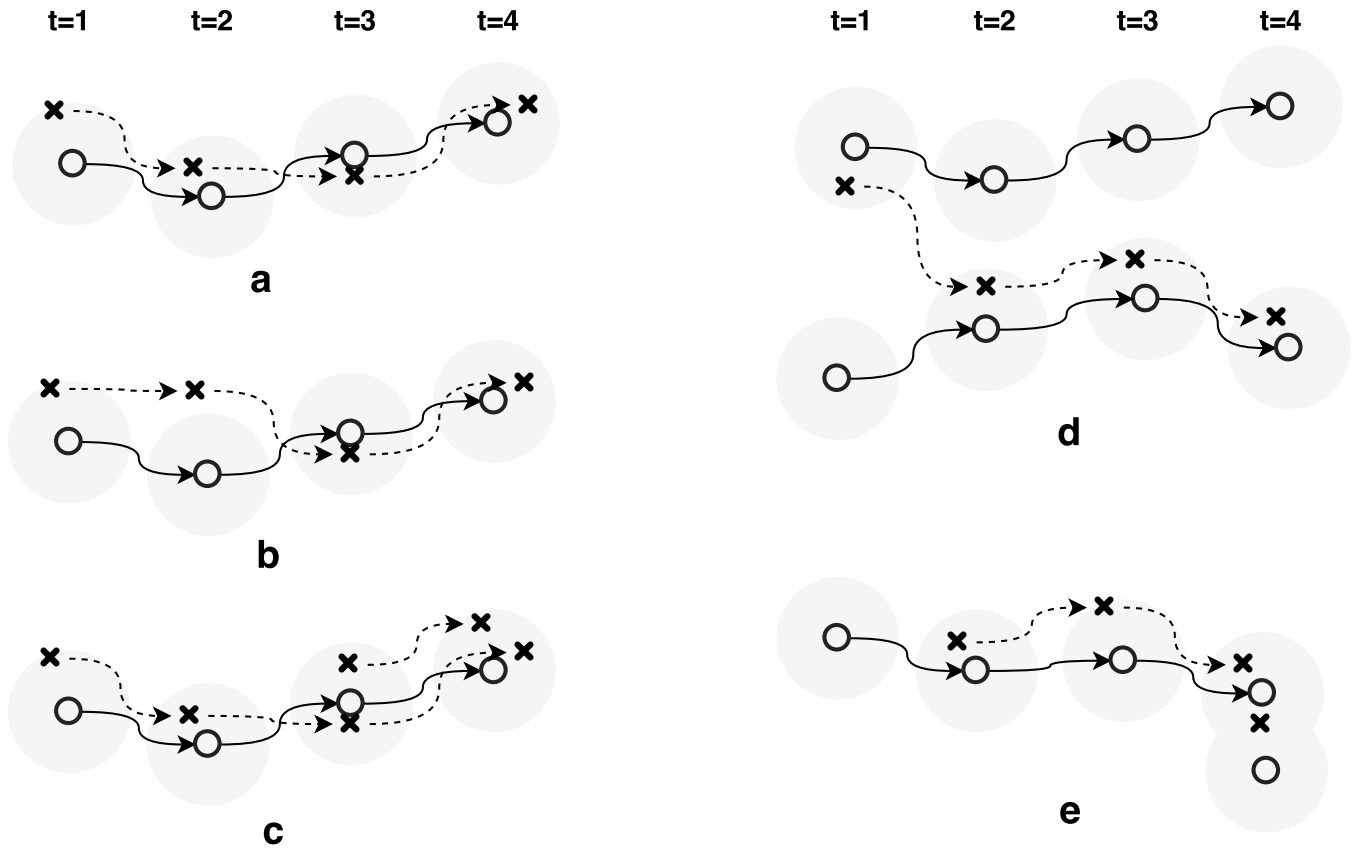


Fig. 15: Possible tracking errors. Circles are ground truth, the gray area around the circles marks the distance wherein a hypotheses (cross) can be such that it is considered a true positive. The distance for which a track is considered a true positive is set to 1 meter if the target is within 20 meter in the vertical direction of the annotation camera and to 2.5 meter if further away. (a) Correct tracking, ground truth corresponds to the hypotheses, (b) false positive (cross not within certain distance from ground truth) and miss (no hypotheses which corresponds with the ground truth) at $t=2$, (c) global mismatch, hypotheses on an already tracked track, (d) ID switch, (e) assigning a new track to the ground truth at $t=2$, assigning the new hypotheses at $t=4$ to the lower ground truth track although it is closer to the upper ground truth, but that track is already assigned.

D. Detailed Results

The detailed results for each dataset are given in this appendix (tables VIII, IX and X). For every dataset it is shown:

- how many frames are used to calculate the results
- the number of false positives
- the number of true positives
- the number of misses
- the number of identity switches
- the number of global mismatches
- MOTP
- the standard deviation of the MOTP
- MOTA (displayed as $\text{MOTA} \cdot 100$)
- the number of tracks generated by the algorithm
- the number of tracks which are not 100% correct (i.e. at least one FP, ID switch or GMM)
- the number of tracks which have more than 125 consecutive global mismatches
- the number of global mismatches except those which belong to tracks which have more than 125 consecutive global mismatches
- the number of tracks which have at least one false positive
- the number of annotated tracks
- the number of tracks which are missed for at least 1 frame
- the expected error per frame (EPPF)

TABLE VIII: Results for dataset 1

Version	A	B	C	D	E	F	G	H	I	J	K
Frames	875	875	875	875	875	875	875	875	875	875	875
False Positives	5	5	7	11	3	6	5	16	6	3	3
True Positives	19191	19190	19186	19185	19193	19185	19186	19180	19191	19191	19193
Misses	7	8	12	13	5	13	12	18	7	7	5
ID switches	2	3	3	3	0	2	2	4	0	0	0
Global mismatch	13	614	17	56	12	5	5	9	832	835	0
MOTP	0.08	0.08	0.09	0.08	0.08	0.08	0.08	0.08	0.09	0.09	0.08
std MOTP	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.11	0.11	0.11	0.1
MOTA	99.93	99.92	99.89	99.86	99.96	99.89	99.90	99.80	99.93	99.95	99.96
# tracks T	23	24	23	24	23	23	23	24	24	24	23
# incorrect tracks T	3	4	5	5	3	5	5	4	4	4	3
# GMM tracks T	1	3	3	3	1	1	1	2	2	2	0
# GMM tracks T > 125 frames	0	1	0	0	0	0	0	0	1	1	0
GMM for tracks T ≤ 125 frames	13	15	17	56	12	5	5	9	2	2	0
# FP tracks T	3	3	4	5	3	5	5	5	4	3	3
# tracks A	23	23	23	23	23	23	23	23	23	23	23
# miss tracks A	4	4	5	6	4	6	6	5	4	4	4
EPPF	0.94	1.55	1.41	1.46	0.02	0.94	0.94	1.88	0.16	0.16	0.01

TABLE IX: Results for dataset 2

Version	A	B	C	D	E	F	G	H	I	J	K
Frames	625	625	625	625	625	625	625	625	625	625	625
False Positives	171	21	55	7	7	7	7	10	21	8	7
True Positives	13881	13967	13894	13968	13968	13968	13968	13961	13967	13967	13968
Misses	94	8	81	7	7	7	7	14	8	8	7
ID switches	6	0	6	0	0	0	0	2	0	0	0
Global mismatch	397	0	26	0	0	0	0	4	0	0	0
MOTP	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
std MOTP	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
MOTA	98.06	99.79	98.98	99.90	99.90	99.90	99.90	99.81	99.79	99.89	99.90
# tracks T	25	24	23	23	23	23	23	23	24	23	23
# incorrect tracks T	7	1	7	1	1	1	1	4	2	2	1
# GMM tracks T	5	0	3	0	0	0	0	1	0	0	0
# GMM tracks T > 125 frames	2	0	0	0	0	0	0	0	0	0	0
GMM for tracks T ≤ 125 frames	105	0	26	0	0	0	0	4	0	0	0
# FP tracks T	8	2	7	1	1	1	1	4	3	2	1
# tracks A	23	23	23	23	23	23	23	23	23	23	23
# miss tracks A	7	1	6	1	1	1	1	3	2	2	1
EEPF	4.83	0.05	4.10	0.02	0.02	0.02	0.02	1.32	0.05	0.03	0.02

TABLE X: Results for dataset 3

Version	A	B	C	D	E	F	G	H	I	J	K
Frames	250	250	250	250	250	250	250	250	250	250	250
False Positives	26	2	25	2	31	0	6	39	2	0	0
True Positives	5560	5523	5543	5535	5542	5617	5573	5593	5532	5617	5594
Misses	121	158	138	146	139	64	108	88	149	64	87
ID switches	2	2	2	2	3	0	2	3	1	0	0
Global mismatch	1	0	51	0	31	0	0	43	43	0	0
MOTP	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
std MOTP	0.11	0.11	0.11	0.11	0.11	0.1	0.11	0.11	0.1	0.1	0.1
MOTA	97.38	97.15	97.10	97.36	96.95	98.87	97.96	97.71	97.32	98.87	98.47
# tracks T	25	24	26	23	26	23	24	25	24	23	24
# incorrect tracks T	3	2	4	2	5	0	3	5	2	0	0
# GMM tracks T	1	0	1	0	1	0	0	1	1	0	0
# GMM tracks T > 125 frames	0	0	0	0	0	0	0	0	0	0	0
GMM for tracks T ≤ 125 frames	1	0	51	0	31	0	0	43	43	0	0
# FP tracks T	4	2	5	2	6	0	3	6	2	0	0
# tracks A	23	23	23	23	23	23	23	23	23	23	23
# miss tracks A	4	4	5	4	6	1	3	4	3	1	2
EEPF	3.79	3.84	4.06	3.79	5.60	0.26	3.66	5.48	2.38	0.26	0.35