



Article

Multi-Player Tracking for Multi-View Sports Videos with Improved K-Shortest Path Algorithm

Qiaokang Liang ^{1,2} , Wanneng Wu ^{1,2,3,*}, Yukun Yang ³, Ruiheng Zhang ³ , Yu Peng ^{3,4} and Min Xu ^{3,*}

¹ College of Electrical and Information Engineering, Hunan University, Changsha 410082, China; qiaokang@hnu.edu.cn

² National Engineering Laboratory for Robot Vision Perception and Control Technology, Hunan University, Changsha 410082, China

³ Faculty of Engineering and IT, University of Technology Sydney, Sydney NSW 2007, Australia; yukun.yang@student.uts.edu.au (Y.Y.); ruiheng.zhang@student.uts.edu.au (R.Z.); pengyu32777@gmail.com (Y.P.)

⁴ Tong Xing Technology, Guangzhou 510000, China

* Correspondence: wuwanneng@hnu.edu.cn (W.W.); min.xu@uts.edu.au (M.X.)

Received: 26 November 2019; Accepted: 19 January 2020; Published: 27 January 2020



Abstract: Sports analysis has recently attracted increasing research efforts in computer vision. Among them, basketball video analysis is very challenging due to severe occlusions and fast motions. As a typical tracking-by-detection method, k-shortest paths (KSP) tracking framework has been well used for multiple-person tracking. While effective and fast, the neglect of the appearance model would easily lead to identity switches, especially when two or more players are intertwined with each other. This paper addresses this problem by taking the appearance features into account based on the KSP framework. Furthermore, we also introduce a similarity measurement method that can fuse multiple appearance features together. In this paper, we select jersey color and jersey number as two example features. Experiments indicate that about 70% of jersey color and 50% of jersey number over a whole sequence would ensure our proposed method preserve the player identity better than the existing KSP tracking method.

Keywords: multi-player tracking; object tracking; k-shortest paths; similarity measurement

1. Introduction

Multiple object tracking (MOT) is part of computer vision interests which is of great significance in terms of both commercial and academic potential. Most of its applications focus on driver assistance and visual surveillance, especially for pedestrian tracking and vehicle tracking. Recently, multiple-player tracking in team sports [1] has been another popular research field for MOT with the increasing passion for sports in the world, especially for the basketball and soccer games. The popularity of sports deservedly results in huge market demands for sports analysis using computer vision methods, and the potential applications include player or team performance analysis, technical-tactics analysis, motion capture, and novel applications in sports broadcasting [2,3]. To those ends, multiple-player tracking, serving as the necessary foundation of the above applications, has been widely studied recently.

Compared with multiple pedestrian tracking, basketball player tracking is very challenging, due to frequent occlusions and abrupt movements. Thus, many existing tracking methods fail in this field. Moreover, players in the same team wearing the same uniforms causes them to be more difficult to distinguish. In addition, players' body-shape variations, motion blur, spectator interferences, and the illumination variations make the players hard to be tracked reliably.

Most existing approaches in multiple-player tracking are based on the tracking-by-detection framework [4–7]. With this paradigm, players are first detected and then linked into trajectories. In contrast, another common tracking framework that is free of object detector is called detection-free-tracking [8]. Although the detection-free-tracking methods are convenient for real-time tracking, the tracking-by-detection approaches can gain more accurate results, since a batch of frames are jointly considered based on the detection results. The latter approach can get a global optimum, while we have to accept a delay in the output as a trade-off for better accuracy. However, the tracking-by-detection methods usually accrue very few detections.

Among the tracking approaches based on tracking-by-detection paradigm, k-shortest path (KSP) optimization is one of the most widely used approaches, proposed by J. Berclaz et al. [9]. With this method, the data association step is formulated as a constrained flow optimization and results in a convex problem, which can then be solved using the KSP algorithm. This approach relies on very few parameters and its convexity ensures that a global optimum can be found. What is more, its feasibility of application to sports tracking has been verified [10]. While effective and fast enough, the neglect of the appearance model in the original KSP tracking process easily leads to identity switches when two or more players are intertwined with each other, as shown in Figure 1. To avoid identity switches, the authors have also explored the appearance features in the tracking process [11]. The method formulates the tracking problem as a multi-commodity network flow problem which can be illustrated as a duplication of the graph for each appearance-group. It uses the tracklets as graph nodes instead of individual locations. However, the improved method needs to run the KSP algorithm twice and has a little difficulty implementing it.

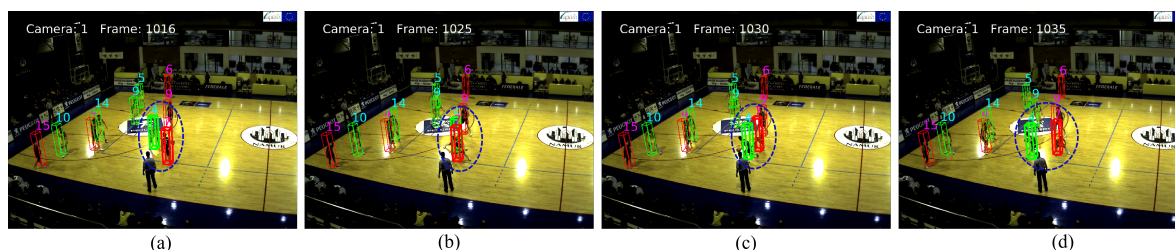


Figure 1. An example of the identity switch problem. It easily occurs when two players annotated with thicker lines are close to each other. (a,b) Blue player number 7 (annotated by red bounding box) and yellow player 4 (labeled with green bounding box) are approaching from frame 1016 to 1025. (c) The two players separate and then swap their identities in frame 1030. (d) The wrong identities remain in frame 1035 and will not change until the next meeting.

Being inspired by KSP-based tracking, we further address the problem of identity switches by computing the similarity of players in consecutive frames. We use the similarity to represent the linking weight between two players. To calculate the similarity, appropriate features need to be carefully selected to distinguish the players. Generally, robust tracking algorithms rely on more than one feature to obtain accurate tracking results [12]. In this paper, both the jersey color and jersey number of each player from a multi-view camera are explored for tracking. Generally speaking, working together, these two features are enough to accurately determine a player’s identity. However, due to occlusions, the two features mentioned above can only be precisely extracted in limited frames, especially for the jersey number. In this regard, our proposed method is also designed to allow more features to be considered.

Overall, our major contribution are as follows. We improve the existing KSP multiple player tracking method by taking the appearance features and similarity measurement into consideration. Although in this paper, only two features are applied for similarity computing, our method can take multiple features into account while assigning different weights on them. Moreover, we carried out extensive experiments to quantify the requirements of appearance features in our proposed approach.

2. Related Work

As a detailed review stated [4], the objective of MOT is to find optimal sequential states of all the objects, which can be generally modeled by performing MAP (maximal a posteriori) estimation from the conditional distribution of the sequential states given all the observations:

$$\hat{\mathbf{S}}_{1:t} = \arg \max_{\mathbf{S}_{1:t}} P(\mathbf{S}_{1:t} | \mathbf{O}_{1:t}), \quad (1)$$

where $\mathbf{S}_{1:t} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_t\}$ denotes all the sequential states of all the objects from the first frame to the t -th frame, and $\mathbf{O}_{1:t} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_t\}$ represents all the collected sequential observations of all the objects throughout the same frames.

Different MOT algorithms from previous works can be thought as designing different approaches to solving the above MAP problem, either from a probabilistic inference perspective or a deterministic optimization perspective. The former approach usually solves the MAP problem using a two-step iterative procedure: step one is a prediction process with a dynamic model and the other step is an update process using an observation model, such as some traditional tracking methods; e.g., mean-shift [13], Kalman filter [14,15], particle filter [16,17], and their variations [18–20]. Once the up-to-time observations are given, these kinds of dynamic approaches can predict the current observations with gradually extend existing trajectories, which make them being very suitable for the online tracking. However, they would easily miss a tracking object when occluded. Not to mention the terrible occlusions in the sports matches.

As for the MOT methods based on deterministic optimization, they solve the above problem by maximizing a likelihood function or conversely minimizing an energy function, which is very popular nowadays. This is because they can obtain a global optimal solution, and thus are usually more robust at dealing with false positives and occlusions. One main disadvantage of these approaches is that there are some delays in outputting final results, as they need to process a batch of consecutive frames. Also, they must depend on the detecting results for each frame. Fortunately, the state-of-the-art detectors, such as Faster RCNN, YOLO et al. [21], can be used in real time. So the tracking output only needs to wait for a short time for processing a batch of frames, the process being able to be controlled within 5 or 10 s, which is acceptable in most applications, even a sports live broadcast.

In the team sports analysis field, the global optimization methods are obviously more popular. Earlier in 2007, people formulated the MOT problem as an integer linear program to obtain a nearly optimal solution [22,23]. But the number of targets in the scene needs to be fixed a-priori, which is a bit confusing. Also, there are some studies that consider multiple-player tracking as a network flow problem, which can be solved in polynomial time [24,25]. Other global optimal methods [26] applied in MOT include graph-based approaches [27,28], and quadratic and linear objective optimization [29]. Even though most of the above global approaches improve a lot compared with the earlier dynamic methods, they are still difficult to optimize and sometimes become trapped in local minima. In practice, issues manifest in tracking errors such as fragmented trajectories and identity switches. Compared with the above methods, the MOT based on KSP algorithm [9] is very fast and easily obtains the global optimal solution. However, identities are also easily be switched when two players are close to each other.

To address this problem of identity switches, people have explored some appearance models [30] or motion models [31]. For example, J. Liu et al. [32] defined a set of game context features to describe the current state of a match. The context-conditioned motion models implicitly incorporates complex inter-object correlations while remaining tractable using cost flow networks. H. B. Shitrit et al. [33] also took image-appearances into account based on KSP tracking. These improved methods can preserve identities better than previous approaches over long sequences, but they perform badly when most of the appearance cues or motion cues cannot be available. This is very common in team sports because of the serious frequent occlusions.

In order to accurately track multiple people with sporadic appearance features, K. C. Amit Kumar et al. [34] firstly proposed a graph-based framework to connect the detections across time based on their position and partial appearance estimates. It proceeds with hypothesis testing in an iterative framework which considers the input data at different time scales. Its main disadvantage is that it is greedy, and consequently there is no guarantee of a global optimal solution. Afterwards, they adopted graph-based label propagation framework [35], in which a number of distinct graphs are constructed in order to capture the spatio-temporal exclusivity constraint, and the appearance information. The effectiveness of the proposed approach has been demonstrated with several challenging datasets, but one limitation is the scalability of the method with respect to the number of nodes.

Other relative tracking approaches towards multiple players in team sports, as reported by a recent review [12], include using bipartite matching [36] or the attribute matching algorithm [37] to associate detections in consecutive frames, extracting dense clusters [38] of tracked players hypotheses to track, and combining multiple local trackers to a global tracker [39]. Recently, MOT problems addressed by correlation filters [40] and deep network flow [41] were also reported. For example, C. Wu et al. [40] have exploited the Markov decision process to integrate the discriminative correlation filters-based (DCFB) tracking method into the MOT framework. Although the proposed method outperforms the state-of-the-art algorithms in many surveillance scenarios, it is yet to be explored in our team sports field.

In this paper we focus on how to further improve KSP tracking by taking appearance features into consideration. Moreover, we introduce a similarity metric between two players to measure their matched probability. The similarity can be computed based on appearance features extracted from multiple views. However, as the occlusions are very serious in basketball videos, some identity features can not be recognized in many frames. In this regard, we carried out extensive experiments to quantify the requirements of appearance features in our KSP-based tracking.

3. Methodology

3.1. An Overview of the Proposed Multiple Players Tracking Framework

Multiple players tracking system generally consists of two steps: detecting players for each frame and then linking the detections into trajectories. The former part is our previous work that has been published as a conference paper [42]. As shown in Figure 2, given a basketball video, we first detect players in 2D images with Mask RCNN [43] frame by frame and then convert the input frames into binary images with foreground-background separation. Afterwards, a probability occupancy map (POM) [44] algorithm is implemented to generate a map representing the existence probability of a player at each discrete location in the basketball court. The input of the POM consists of the above 2D player detections together with the multi-camera calibration data [45]. In the existing KSP tracking approach [9], the output of POM is the only input of KSP tracking. In our proposed method, however, we also consider player appearance features (the jersey color, jersey number, etc.) as other inputs of the tracking model. The same as the output of the POM, the available appearance features also exert important influences on the weights of linking edges. After that, k player trajectories can be output based on KSP algorithm. Since the main difference between our proposed method and the existing KSP tracking approach is that we consider the appearance features while the original KSP does not, we thus name our method KSP-AF.

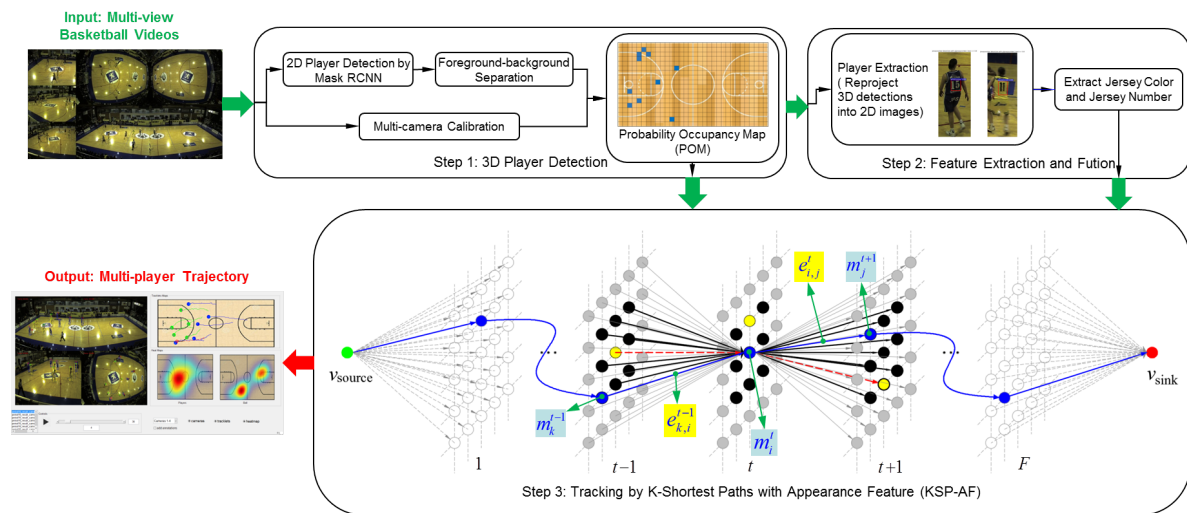


Figure 2. The pipeline of the multiple-player tracking system based on the KSP-AF tracking method mainly consists of three parts: (1) 3D player detection [42], (2) feature extraction, and (3) tracking by k-shortest paths with appearance feature (KSP-AF). The input of the whole system is multi-view basketball videos, and the output is multiple player trajectories. For the KSP-AF tracking, the inputs are both of the 3D detections represented as POM data and the extracted features for each detection.

3.2. Existing KSP Tracking Method

In fact, KSP (k-shortest path) was originally an algorithm for searching the k shortest path among all paths from the start point to the end point of a graph [9]. It can be applied to multiple object tracking which is formulated as a graph optimization problem for finding the shortest path in the graph determined by time and coordinates. In multiple-player tracking using KSP, it is assumed that each person moves one or more grid cells between successive frames in the tracking space of the world coordinate system which is divided into grids. Also, there is a restriction to only one person in each grid cell at most. The minimum cost flow problem that is originally regarded as an NP-hard integer programming problem can be relaxed to linear program problem by placing this constraint and it then becomes possible to solve in polynomial time.

As shown in Figure 2, the network flow graph illustrates the principle of the tracking by KSP-based tracking methods. A node in location i at time t is represented as m_i^t , where i denotes the location index and t is the time stamp. The linking edge $e_{i,j}^t$ connects from node m_i^t to m_j^{t+1} , which means a possible traveling route that a player can move between consecutive frames. The thick black lines in Figure 2 are possible links when the depth of the graph is assumed as 1, while the thin gray lines assume the depth of 2. It is clear to see that when the depth increases, the computing complexity will be multiplied as the possible links increase a lot. Similar to the granularity of the grid, the depth of the KSPs are determined in consideration of the moving speed of players and the frame rate. It is also noted that there are two virtual nodes, v_{source} and v_{sink} , in the graph. These two special nodes are connected to all the possible nodes that player may enter and leave, which makes the KSP-based tracking system allow the players to enter and exit the tracking area.

In the original KSP tracking method, the weight of the linking edge is determined by player occupancy probability at each position, while the linking weights outgoing from the virtual source node are set to zero. However, as the linking weights outgoing from node m_i^t only depend on the player occupancy probability at this node, the weights from m_i^t to any m_j^{t+1} are set as the same, even they belongs to different teams. In this case, when players come close to each other at time t , their identities would be easily switched. In this paper, we rethink the linking weights by considering the identity information of the nodes. First, we need to construct an appearance model based on the existing player occupancy map. Then, we introduce similarity measurement between two nodes to calculate new linking weights. Ideally, when all of the identity cues for each node that might be occupied by a player

can be available, the obtained trajectories would be perfect without any identity switches. However, due to the serious occlusions in team sports, only a few pieces of identity information can be accessed. In this paper we further carry out extensive experiments to quantify the requirements of appearance features in KSP-based tracking.

3.3. Proposed KSP-AF Formulation

According to the original KSP, we first need to discretize the basketball court into G grid cells. If the frame number of one batch is represented as F , the total number of nodes except the two virtual nodes is $n = GF$. The approach tactfully formulates the multiple objects tracking problem as a k -shortest node-disjoint paths problem on a directed, acyclic graph. The nodes represent the human locations in the court and edges denote the possible move for players in consecutive frames, while the weights between nodes pairs are determined by the human occupied probability of every node. After the construction of the graph, k paths between these nodes can be found, as the total costs of the paths are minimum. In this part, we analyze how to take two or more appearances into consideration based on the KSP tracking algorithm.

Assume that we have access to a person detector that estimates the probability of presence of someone at every location i . Then the player occupied probability for each grid can be formula as:

$$\rho_i^t = P(X_i^t = 1 | I^t), \tag{2}$$

where ρ_i^t is the estimated probability of a location i to be occupied by a player at time t , X_i^t is a random variable standing the true occupancy of location i at time t , and I^t represents the original image at time t .

Moreover, what we need to consider in our tracking algorithm is not only the occupied probability of each location, but also more informative features for each grid cell occupied by a player. Thus, we further assume that we can compute an appearance feature model Y_i^t and that we use it to estimate the identity similarity between two node pairs in consecutive frames:

$$s_{i,j \in \mathbb{N}}^t = P(Y_i^t = Y_{j \in \mathbb{N}}^{t+1} | I_t, X_i^t = 1), \tag{3}$$

where $s_{i,j}^t$ is the similarity that the identity information in location i at time t is the same as the location $j \in \mathbb{N}(i)$ at time $t + 1$; $\mathbb{N}(i)$ is the neighborhood of location i ; and Y_i^t and Y_j^{t+1} are appearance features of location i and location $j \in \mathbb{N}(i)$ at time t and $t + 1$, respectively.

As our goal is to find a set of physically possible trajectories that each trajectory contains only one player, on one hand we need to link the detections as many as possible, and, on the other hand, the similarity between the connected nodes should be as high as possible. We can formulate it as:

$$\hat{m} = \arg \max_{m \in \Omega} P(X_i^t = x_i^t, Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I_t), \tag{4}$$

where \hat{m} is a set of occupancy maps and Ω is the space of occupancy maps satisfying several constraints in (7) to (10).

Assume the conditional independence of the occupancy maps given I^t is true and our objective function can be rewritten as follows based on log function (the detailed deduction process can be seen in the Appendix A of this paper):

$$\begin{aligned} \hat{m} &= \arg \max_{x \in \Omega} \log \prod_{t,i} P(X_i^t = x_i^t, Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I_t) \\ &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \log P(X_i^t = x_i^t, Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I_t) \\ &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \left(\log \frac{\rho_i^t \cdot s_{i,j \in \mathbb{N}(i)}^t}{1 - \rho_i^t} \right) \cdot x_i^t. \end{aligned} \tag{5}$$

The x_i^t represents the actual occupancy of location i at time t , and it is equal to the sum of flows leaving from the same location ($\sum f_{i,j}^t$). Therefore, together with the constraints, our goal is to solve the following linear program optimization problem:

$$\max \sum_{t=1}^T \sum_{i=1}^n \left(\log \frac{\rho_i^t \cdot s_{i,j \in \mathbb{N}(i)}^t}{1 - \rho_i^t} \right) \cdot \sum_{j \in \mathbb{N}(i)} f_{i,j}^t \tag{6}$$

s.t.

$$\forall t, i, f_{i,j \in \mathbb{N}(i)}^t \geq 0 \tag{7}$$

$$\forall t, i, \sum_{j \in \mathbb{N}(i)} f_{i,j}^t \leq 1 \tag{8}$$

$$\forall t, i, \sum_{j \in \mathbb{N}(i)} f_{i,j}^t - \sum_{k \in \mathbb{N}(j)} f_{j,k}^t \leq 0 \tag{9}$$

$$\sum_{i \in \mathbb{N}(v_{\text{source}})} f_{v_{\text{source}},i} - \sum_{k, v_{\text{sink}} \in \mathbb{N}(k)} f_{k,v_{\text{sink}}} \leq 0. \tag{10}$$

The imposed constraints (7) and (8) are reasonable because the flows have to be nonnegative and a location cannot be occupied by more than one player at a time. Also, as a player cannot leave from one location if there is no player having entered into this location, so we have the constraint (9). Similarly, we impose the constraint (10) because all of the flows departing from v_{source} will end up in v_{sink} .

In the next step, we will analyze how to solve the above linear program optimization problem. Among various methods, the computing complexity can be drastically reduced by reformulating the problem as a k-shortest node-disjoint paths problem on a directed acyclic graph, as shown in Figure 2. In order to apply the KSP to solve it, we convert the above maximization problem into an equivalent minimization problem by negating the objective function. Therefore, a directed linking edge $e_{i,j}^t$ from location i at time t to location j at time $t + 1$ is assigned the weight value as:

$$w(e_{i,j \in \mathbb{N}(i)}^t) = -\log \left(\frac{\rho_i^t \cdot s_{i,j}^t \cdot c}{1 - \rho_i^t} \right), \tag{11}$$

where the c is a constant coefficient that it can eliminate the interference of $s_{i,j}^t$, when one of the connected node pairs' identity feature cannot be available. It is very important that it makes our proposed algorithm suitable for the cases when the features cannot be available in some frames. The cost value of the edges departing from the source node is set to zero to allow players to appear at any entrance position and at any time instant at no cost.

3.4. Graph Construction

Based on the above theoretical analysis, we can convert the multi-player tracking problem into a graph model to solve; the graph is a directed acyclic one (a path can visit every node once at most); and the paths are node-disjoint (different paths cannot share the same node), as shown in Figure 2. For a graph, nodes and edges are two key elements. Thus, constructing a graph means defining these two elements and assigning the linking weights of the edges.

3.4.1. Definition of the Nodes and Edges

In our experiments, we define the node m_i^t as the location i of the court at time t , with a total of G nodes for one frame. In order to reduce the complexity of the graph, we assume the node m_i^t can only move to its neighbor region in the following frame. The size of the neighbor region, also being named the depth of graph, can be determined in consideration to the speed of the players as well as the frame rate. The thick black lines in Figure 2 are possible links when the depth of the graph is assumed as 1,

while the thin gray lines assume the depth of 2. Afterwards, given a pair of virtual nodes, v_{source} and v_{sink} , any path between them is considered as a player’s trajectory, such as the blue curve in Figure 2.

On the whole, the nodes of the graph are defined as the discrete grids, each of which corresponds the location in the court. And the graph’s edges consist of three types: (1) incoming edges: from the source node to the nodes in the first frame or the access nodes in other frames; (2) outgoing edges: from the nodes in last frame or the access nodes in other frames to the terminal node; (3) middle edges: from the node i to its neighboring nodes $j \in \mathbb{N}(i)$.

3.4.2. Weights Assignment for Linking Edges

Before we solve the graph with KSP algorithm, we need to assign the weight value for each linking edge according to our previous theoretical analysis, which is the most key point of the graph construction. As shown in Figure 2, given a depth of 1, a player occupying the location i at time t can arrive at one of the nine neighboring locations at time $t + 1$. In the original KSP tracking approach, they are assigned the same weights, only depending on the player occupancy probability of node m_i^t regardless of any appearance features. In our proposed method stated above and according to Formula (11), the weights of edges would depend on not only the player occupied probability, but also the similarity between the node pairs to be connected. The more similar the appearance between them, the smaller the weights imposed. According to this, m_i^t labeled as blue in Figure 2 will definitely move to the blue node instead of yellow node. How to measure the similarity between two nodes is described in Section 3.5.

Therefore, once we built a directed acyclic and node-disjoint graph, we could then implement Dijkstra’s algorithm to find k paths between the v_{source} and v_{sink} such that the total cost of the paths is minimal. This issue is well studied in the network optimization literature [46].

3.5. Similarity Measurement

Generally, a combination of more than one feature could make the similarity measurement more accurate, and thus the tracker more robust to occlusions and missed detections. Once the features are extracted, we can first compute the similarity for each feature respectively and then fuse several similarities into one. This can be done simply by a multiplication if they are mutually independent, or by assigning different weights according to the confidence level or important factor for each feature. In this paper, we select two common features—the jersey color and jersey number—to measure the similarity of two players. These two features are enough to help us distinguish most of players even they cannot always be available due to the occlusions. Even so, more features would be appreciated. In the following part, we will take the feature of jersey color as an example to illustrate how to calculate the similarity between two nodes independently.

Figure 3 shows a simple flow model that people in location i at time t can travel to one of their neighboring locations $j \in \mathbb{N}(i)$ at time $t + 1$. The blue nodes (m_i^t and $m_{j_1}^{t+1}$) represent players wearing blue jerseys; a yellow node ($m_{j_2}^{t+1}$) means a player who belongs to yellow team, while the jersey color of other nodes are unavailable. It is easy to know that the node m_i^t is more likely to arrive at the node $m_{j_1}^{t+1}$ with the same jersey color, rather than the yellow node $m_{j_2}^{t+1}$. And matching probabilities between other nodes whose features cannot be available would be the same as 50%. We call the link along the green arrow a strong match; the red arrow with dashed line means an impossible match; the other gray arrows represent possible matches. Thus, given the feature labels and their probabilities, we can define the similarity of two nodes as:

$$s_{i,j}^t = \begin{cases} (p_i^t + p^{t+1}) / 2, & L_i^t = L_j^{t+1} \\ 1 - (p_i^t + p_j^{t+1}) / 2, & L_i^t \neq L_j^{t+1} \\ 0.5, & L_i^t \text{ or } L_j^{t+1} \text{ is unknown,} \end{cases} \quad (12)$$

where L_i^t, L_j^{t+1} are the available feature labels for the two nodes (m_i^t, m_j^{t+1}) respectively and p_i^t, p_j^{t+1} are their corresponding probabilities.

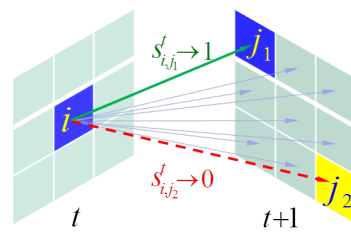


Figure 3. A simple flow model: people in location i at time t can travel to location $j \in \mathbb{N}(i)$ at time $t + 1$. The green arrow means a strong match with a similarity near to 1; the red dotted arrow represents an impossible match with a similarity close to 0; the gray arrows are possible matches whose similarities are set the same as 0.5.

However, due to occlusions, some features (especially the jersey number) can be available only in very few frames. Sometimes, when the appearance feature at current time t is unavailable, the similarity between node m_i^t and node m_j^{t+1} would be set as the same. It means that the probabilities of the player in location i at current time t arriving at anyone of its neighbor location $j \in \mathbb{N}(i)$ at the following time $t + 1$ are the same, like the original KSP. In our proposed method, in order to make full use of the available features as much as possible, we also consider the neighboring nodes in the previous frame to help us discriminate more players when the feature can not be extracted in the current frame due to occlusion or low resolution. For example, when the jersey color in location i at time t is unavailable but can be known for one of its neighboring nodes in the preceding time $t - 1$, as shown in Figure 4a,b), we can compute the similarity between the preceding and following frame instead. However, if more than one node with its feature available could arrive at location i (see Figure 4c), or none of the neighboring nodes can be access to their feature (see Figure 4d) at time $t - 1$, we cannot match the node i and $j \in \mathbb{N}(i)$ because the existing identity information is ambiguous or not enough to judge whether they should be connected.

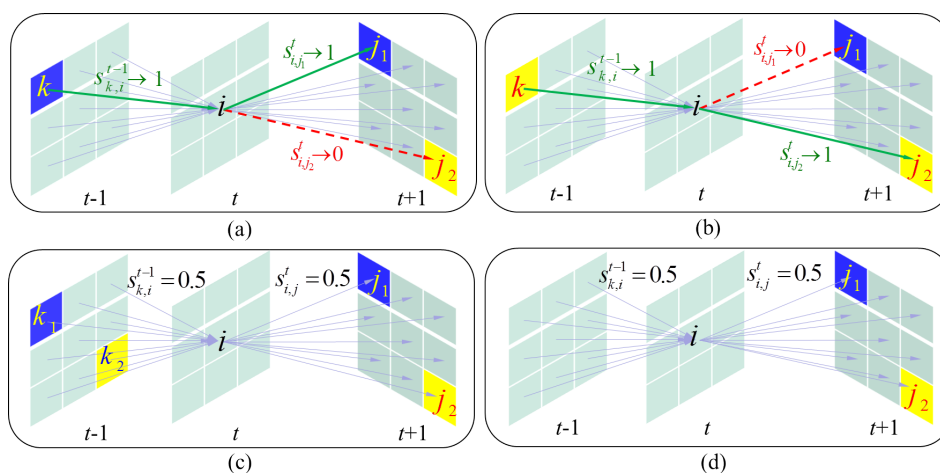


Figure 4. Illustration of similarity measurement when jersey color at the current frame (t) is unavailable: (a,b) one of its neighboring nodes (k) in previous frame can access the jersey color; (c) jersey color can be available in more than one node (k_1 and k_2) at $t - 1$; (d) jersey color cannot be obtained for all of the neighboring nodes at $t - 1$. A green arrow here is called a relative strong match ($s \rightarrow 1$); a red dotted arrow is an impossible match ($s \rightarrow 0$); other gray arrows are possible matches ($s = 0.5$).

3.6. Feature Extraction and Fusion

In our tracking framework, we firstly detect the players in different views frame by frame, which can be done by Faster RCNN [47]. Then we send the 2D detections into POM algorithm [44], and it outputs the player occupied probability for each discrete grid of the court, as shown in Figure 2. Afterwards, the locations where the probability of being occupied by a player surpasses 80% are selected out and mapped into different views as rectangle boxes, as shown in Figure 5.

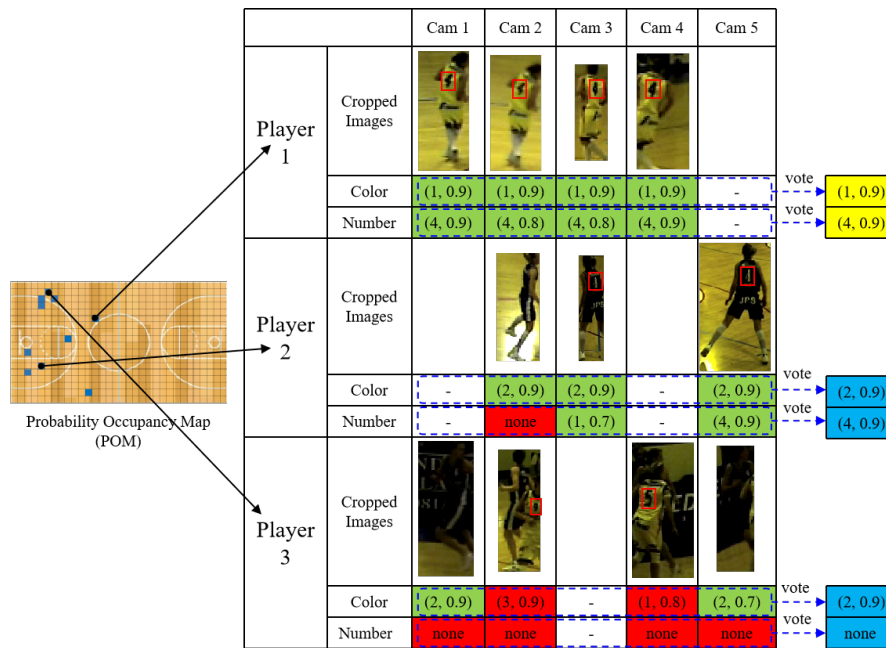


Figure 5. The illustration of feature extraction and fusion: players 1, 2 and 3 are taken as three examples that correspond three locations in the probability occupancy map, respectively. For each player, it can be mapped into several views to crop its image patches. The jersey color and jersey number can be jointly determined by all the cropped images, which can be done by Boyer–Moore voting algorithm. The output of color feature can be represented as (L_c, p) where $L_c \in \{1 \text{ (yellow)}, 2 \text{ (blue)}, 3 \text{ (yellow and blue)}, 4 \text{ (none)}\}$ and p means its probability. The output of number feature can be denoted as (L_n, q) where L_n and q is the jersey number and its probability, respectively.

For one location (i.e., one player), it might correspond to several bounding boxes cropped from different cameras. Thus, the appearance feature can be jointly determined based on these cropped images, which can avoid some interferences caused by occlusions to some extent. The appearance feature used in this paper is the jersey color and jersey number. For the jersey color, assume the output of color feature for one cropped image can be represented as (L_c, p) where $L_c \in \{1 \text{ (yellow)}, 2 \text{ (blue)}, 3 \text{ (yellow and blue)}, \text{and } 4 \text{ (none)}\}$, and p means its probability. Then, we need to fuse multiple color feature candidates into one, which can be considered as a vote issue that needs to determine which team the player should belong to. This problem can be easily solved by the classic Boyer–Moore voting algorithm [47]. Similarly, the jersey number as well as its probability can be obtained in the same way. However, it is worth mentioning that before the jersey number extraction for each cropped image, we need to ensure the player’s color label is the same as the winning color label after voting. If not, the jersey number might be wrong due to occlusion (see the player 3 in Camera 2 and 4 in Figure 5).

As to how to extract the jersey color and jersey number for each cropped image, many existing methods can be available. In this paper, we adopted a pre-trained person re-identification (re-ID) network [48] to obtain a descriptor for the bounding box (a 128-dimensional vector) and then classified the players as four classes using SVM algorithm [49]: 1 (yellow player), 2 (blue player), 3 (yellow and blue player), and 4 (none). For the jersey number, we first train a detection model to locate the jersey number and then classify the jersey number images with a six-layer convolutional neural

network which is borrowed from an jersey number recognition approach crafted for soccer players [50]. Other relative works related to jersey number recognition can be seen in [51,52]. An accurate feature extraction is very important for our tracking framework, and how to further improve its accuracy is our future work while this paper mainly focus on how to improve the problem of identity switches by taking full advantage of the appearance features. Thus, before the implement of our tracking method, two prerequisite knowledge should be given: (1) player occupied probability for each location in the court (provided by POM algorithm [44]); (2) the label of jersey color and jersey number and its corresponding probabilities (provided by re-ID [48] + SVM [49] and jersey number recognition based on CNN [50]).

4. Experiments

4.1. Experiment Data

To verify our method, two datasets were prepared for testing. One is a publicly available sequence named APIDIS dataset [53]. The other was captured by our own at Shantou University in China and we called it STU dataset. The basic information of the two datasets is described as follows.

APIDIS dataset: The 1500-frame sequence was acquired by seven cameras, with five ground cameras around the court and two fish-eye cameras overhead of the court, as shown in Figure 6. Each camera captures the videos at a rate of 25 *fps* in 800×600 resolution. There are 12 people on the court (two referees and two five-player teams). In this dataset, the overall brightness is relatively low and thus the identity of the player is a bit challenging to be recognized.

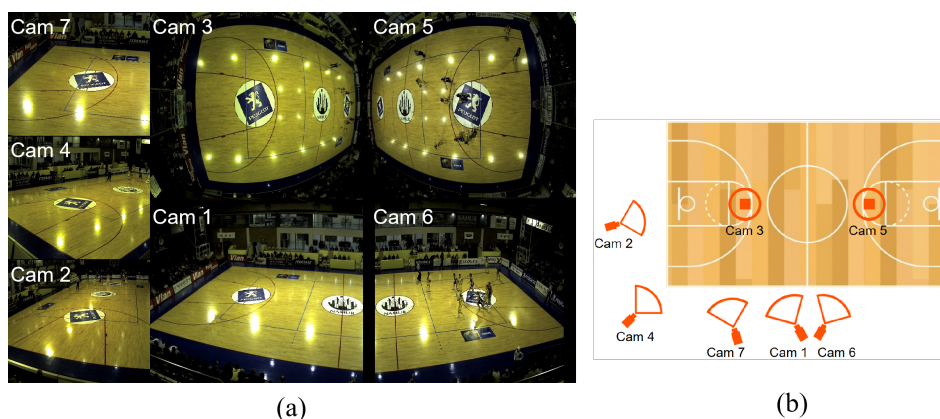


Figure 6. APIDIS basketball dataset is captured by five ground cameras around the court and two fish-eye cameras overhead of the court: (a) images captured from seven views; (b) the schematic diagram of camera installation.

STU dataset: The dataset was collected by eight synchronized cameras with a size of 1280×720 and the frame rate is also 25 *fps*, as shown in Figure 7. The cameras were evenly distributed around the basketball court. The dataset has a better illumination condition compared with the APIDIS dataset. However, the occlusion is more serious since our cameras were installed at a lower position, although this kind of installation can help us recognize more jersey numbers. Thus, this dataset is also challenging for tracking. The STU dataset was collected by our own in a basketball match and we divided the data into 10 periods of sequences. There is not much difference between them except the frame length which varies from 600 to 4400.

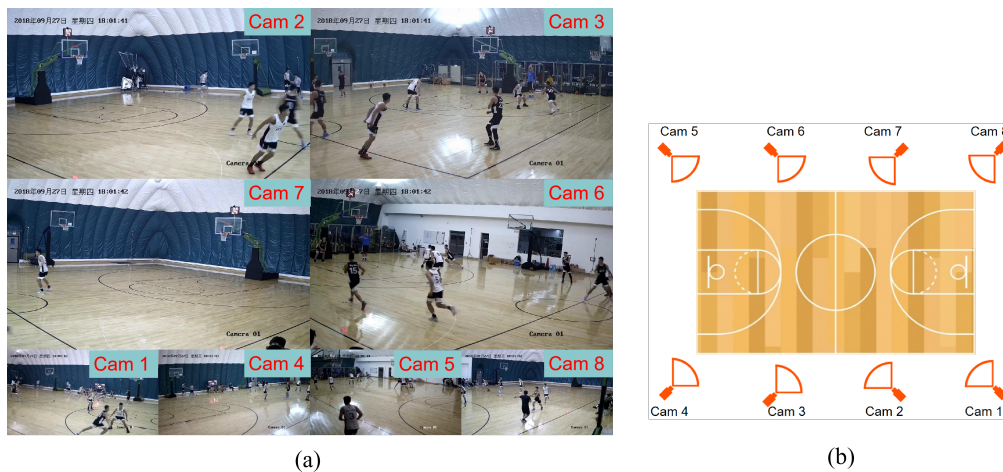


Figure 7. Our own STU basketball dataset is collected with eight ground cameras evenly distributed around the court: (a) images captured from eight views; (b) the schematic diagram of camera installation.

4.2. Setup and Parameter Settings

Our tracking framework can be implemented in Windows or Linux operating system. The only thing worth noting is that one needs to install the Boost library first for running the k shortest paths optimization. The source code of the KSP library developed in C++ can be found at [54].

For any computer configuration, processing a whole video sequence is possible in theory, but impractical, as its computing complexity would increase exponentially compared with a short sequence. Thus, we commonly split the sequence into batches first and then connect them according to some matching cues. The matching process adopts a pattern of slide window, described in detail in Figure 8. Such a batch processing mode can guarantee a global optimization solution within a batch of frames but it also must accept a delay as a tradeoff.

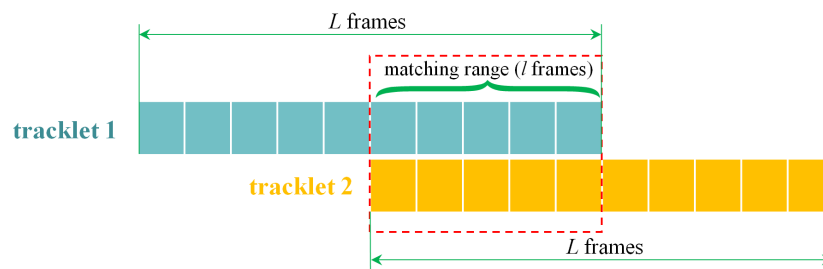


Figure 8. Connect two tracking segments according to players’ positions within the matching range of l frames. For example, suppose the batch size is set as L : both tracklet 1 and tracklet 2 contain L frames with l frames being overlapped. We call the overlapped l frames matching range frames. Then we calculate the distance between the two positions frame by frame within the matching range. If the distance is less than a given threshold, we call it a match. If not, a mismatch. Only if the matching number of times overpasses 80%, we then connect the two tracklets. Otherwise, we consider the current tracklet segment a new trajectory.

Overall, compared with other relative approaches, our proposed method depends on very few parameters: only the granularity of the grid cells, the depth of KSP network, and the batch size. The batch size can be given according to the computing capability, while the other two parameters are determined by the consideration of the moving speed of players and the frame rate. In our experiments, we divide the court into discreted grids with a size of $128 \times 72 \times 1$ as well as the depth of 1. In view of the running speed, the batch size is set as 100 frames and the number of matching range frames is defined as 50 in our experiments. Thus, in addition of the virtual source and sink nodes, 921,602 nodes

were the total for one batch. After the weights of all possible links are calculated by Formulas (11) and (12), a graph network is then constructed, as shown in Figure 2. It is not hard to see that the linking edges count of the graph mainly depends on the three parameters we have mentioned above: KSP network depth, grid number, and the batch size. More edges would result in more time consumption in the tracking process.

4.3. Results

4.3.1. Evaluation Metrics

A well-known metric of MOT is the multiple object tracking accuracy (MOTA) [55], which shows its expressiveness as it combines three sources of errors (false negatives, false positives, and identity switches):

$$\text{MOTA} = 1 - \frac{\sum_t (c_1 \cdot fn_t + c_2 \cdot fp_t + c_3 \cdot idsw_t)}{\sum_t g_t} \quad (13)$$

where fn_t , fp_t , $idsw_t$, and g_t represent the numbers of false negatives, false positives, instantaneous identity switches, and ground truth, respectively. In order to highlight the influence of identity switches which is the primary problem this paper aims to solve, the weighting factors are specially set as $c_1 = c_2 = 1$, and $c_3 = 10$ in this paper, while many other related works set $c_3 = \log_{10}$ [11,56,57]. Anyway, experiments indicate that our setting can better discriminate the MOTA score under different methods.

However, in order to better evaluate the influence of identity switches, H. B. Shitrit [11] introduced a new term $gidsw$ instead of $idsw$ for measuring the proportion of identity switches in a global way. In fact, the $idsw$ only counts the number of instantaneous identity switches, but does not account for the proportion of a trajectory that is correctly labeled over a whole sequence. In contrast, the $gidsw$ would count all the mismatches when the identity is changed. Thus, a trajectory with an identity switch in the middle will be counted wrong for half of its length, instead of just once for the $idsw$. Also, to balance the influence of each count, the weighting coefficients can be adjusted accordingly, as can the global multiple object tracking accuracy (GMOTA), by replacing the multiple object tracking accuracy (MOTA) formula:

$$\text{GMOTA} = 1 - \frac{\sum_t (c_1 \cdot fn_t + c_2 \cdot fp_t + c_3 \cdot gidsw_t)}{\sum_t g_t} \quad (14)$$

where the weighting factors are set to $c_1 = c_2 = c_3 = 1$ since the $gidsw$ is much larger compared with $idsw$; hence, no need for extra penalty.

4.3.2. Players Tracking Performance

The tracking performance of our proposed KSP-AF method seriously depends on three components of the whole tracking system: players' 3D detection, feature extraction and fusion, and trajectory generation. In the following part, we focus on the latter two influence factors while the detection performance is beyond the scope of this paper.

In particular, the KSP-AF tracking is mainly influenced by the proportion of available features. The more features we can extract, the better performance KSP-AF can reach. Ideally, once given the identity cues for each player in all frames, a trajectory for one player is just the detections with the same identity over a long sequence. However, it is impossible for us to extract all the features especially when the player are frequently occluded by each other. In order to analyze how much proportion of the identity features can ensure a satisfied tracking effect, we manually label all the identity information for each detected person in the court and gradually select a proportion of features as the input to test our algorithm.

We have done three groups of experiments to verify the KSP-AF's performance. First, we only consider the color feature and test its improving performance when more color information is used. Second, we gradually increase both of the color feature and number cues to explore the variation

trend of the tracking performance. These two experiments are done on the publicly APIDIS dataset. Finally, to further demonstrate the effectiveness of our method, we use our own STU dataset, totaling 10 periods of sequences whose lengths vary from 600 to 4400 frames.

(1) Tracking experiments with jersey color only (APIDIS dataset).

We first apply the original KSP tracking approach without considering any appearance feature in the 1500-frames APIDIS dataset, and we can obtain as many as 14 trajectories with the MOTA and GMOTA scores being 0.83 and 0.71 respectively, as shown in Figure 9 (0%Color). Then, as we gradually exert more jersey color information into our KSP-AP tracking framework, the tracking performance improves very little or even degrades, as shown in Figure 9. Both the MOTA and GMOTA scores do not increase a lot until more than 70% of the color information can be available. This is because players often wear the same uniforms when they belong to the same team, and thus integrating only the jersey color into the tracking framework cannot completely distinguish players. Anyway, when more than 70% of color feature can be available over the sequence, the tracking performance of KSP-AF would be obviously improved compared with the original KSP method (0%Color).

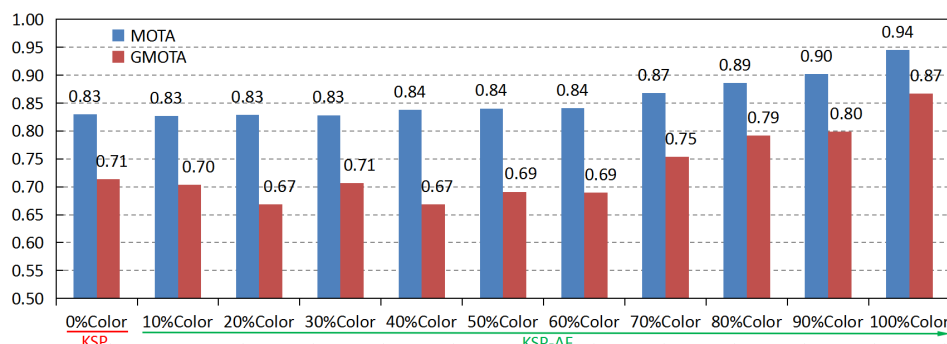


Figure 9. Using only jersey color information cannot largely improve the tracking performance for KSP-AF: (1) 0%Color corresponds the original KSP; (2) both MOTA and GMOTA scores are not largely increased with the escalation of jersey color cues; (3) this experiment is implemented in APIDIS dataset.

(2) Tracking experiments with both jersey color and jersey number (APIDIS dataset).

Generally, we can have access to more jersey color than jersey number. In both APIDIS and STU datasets, we can commonly extract around 50% to 80% of the jersey color and 40% to 60% of the jersey number based on some advanced extracting methods [51,52]. The available proportions of these two features depend on several factors, including the image resolution, light conditions, and extracting approach. More features in the tracking process would result an obviously better tracking performance, as shown in Figure 10. More detailed comparison results can be seen in Table 1.

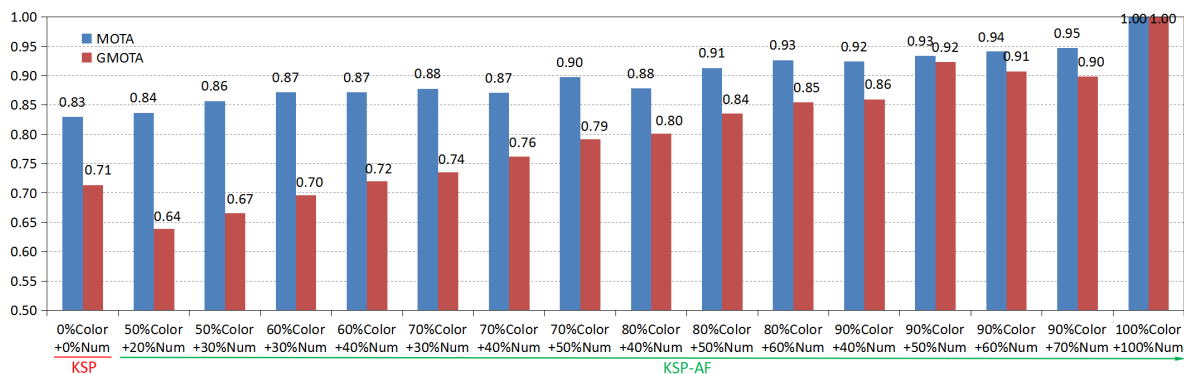


Figure 10. The tracking performance of KSP-AF is gradually improved when more and more jersey colors and jersey numbers are given: (1) 0%Color+%0Num corresponds the original KSP; (2) under KSP-AF, both the MOTA and GMOTA scores gradually increase when more appearance information (e.g., jersey color and jersey number) can be extracted; (3) this experiment is done with APIDIS dataset.

In Figure 10 and Table 1, 0%Color+0%Num represents that none of appearance features are considered in KSP-AF, which exactly corresponds the original KSP. 100%Color+100%Num means that the appearance feature of jersey color and jersey number can be available for all the players in all frames, which can lead to a deserved result (MOTA = 1.0, GMOTA = 1.0). Our KSP-AF tracking performance does not improve a lot, or even degrades to some extent at the beginning when the available features are very limited. Until the available proportions of jersey color and jersey number reach to 60% and 40% respectively, both the MOTA and GMOTA scores begin to outperform original KSP. After several repeated experiments, we can finally report that more than 70% color plus more than 50% jersey number over the whole sequence can ensure our KSP-AF tracking performance against original KSP.

Table 1. The list of tracking performance under different available proportion of the jersey color and jersey number based on KSP-AF in APIDIS dataset.

Index	Available Feature Proportion	Trajectories Count	TP	FN	FP	IDSW	GIDSW	MOTA	GMOTA
1	0%Color+0%Num(KSP)	14	14,982	18	18	252	4260	0.83	0.71
2	50%Color+20%Num	11	14,928	72	72	231	5270	0.84	0.64
3	50%Color+30%Num	13	14,932	68	68	201	4881	0.86	0.67
4	60%Color+30%Num	13	14,933	67	67	180	4425	0.87	0.70
5	60%Color+40%Num	12	14,930	70	70	179	4057	0.87	0.72
6	70%Color+30%Num	11	14,929	71	71	169	3828	0.88	0.74
7	70%Color+40%Num	12	14,926	74	74	179	3418	0.87	0.76
8	70%Color+50%Num	12	14,940	60	60	142	3009	0.90	0.79
9	80%Color+40%Num	12	14,933	67	67	169	2853	0.88	0.80
10	80%Color+50%Num	10	14,953	47	47	121	2377	0.91	0.84
11	80%Color+60%Num	10	14,955	45	45	101	2090	0.93	0.85
12	90%Color+40%Num	10	14,956	44	44	106	2021	0.92	0.86
13	90%Color+50%Num	10	14,961	39	39	92	1075	0.93	0.92
14	90%Color+60%Num	10	14,966	34	34	81	1319	0.94	0.91
15	90%Color+70%Num	10	14,964	36	36	73	1453	0.95	0.90
16	100%Color+100%Num	10	15,000	0	0	0	0	1.00	1.00

Parameters explanation: (1) For the ground truth: Tracked: the ground truth point is tracked in one trajectory (i.e., the distance between ground truth point and tracking point is less than a defined threshold.) False negative (FN): the point is not tracked in any trajectory. (2) For the trajectories: True positive (TP): the hypothesis in the trajectory is within a defined scope of the ground truth point. False Positive (FP): otherwise, it is a false positive. (3) For each trajectory: Identity switches (IDSW): if the true ID at time t is different from the ID at time $t - 1$, we call it an IDSW. Global identity switches (GIDSW): The ID at time t is different from the whole ID label of the trajectory. Thus, it means that a trajectory with an identity switch in the middle will be counted as GIDSW for half of its length, instead of just once for the IDSW. (4) MOTA and GMOTA are explained in Section 4.3.1.

(3) Tracking experiments with more sequences (STU dataset).

To further demonstrate the effectiveness of our method, the STU dataset collected by us is used; it includes 10 periods of sequences. We first use the original KSP to test, and then apply KSP-AF into the same sequences (with 70% jersey color and 50% jersey number). Figure 11 illustrates the comparison of MOTA and GMOTA scores among the ten periods of sequences, respectively. Numbers in brackets represent the frames length of each period, respectively. It is clear to see that, compared with the original KSP approach, both MOTA and GMOTA scores under KSP-AF obviously increase in each period. This means that our KSP-AF method with 70% of the jersey color and 50% of the jersey number can maintain the identity more steadily than the original KSP method.

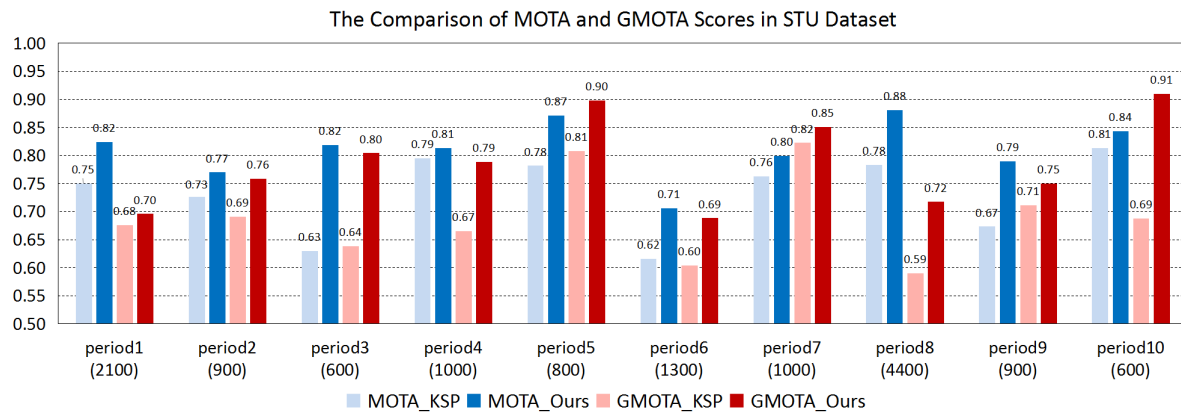


Figure 11. The comparison of MOTA and GMOTA scores between original KSP and our proposed KSP-AF in STU dataset: (1) 70%Color+50%Number is provided for KSP-AF tracking while original KSP considers neither of them; (2) both MOTA and GMOTA scores are clearly improved in KSP-AF compared to original KSP; (3) numbers in brackets are the frame lengths of each period respectively.

4.3.3. Baselines and Comparison

To compare our proposed method with other related works in multiple player tracking, here we list four baselines which have reported their MOTA or GMOTA scores based on the same 1500-frame APIDIS basketball dataset. In order to a fair comparison, we reset weight factors ($c_1 = c_2 = 1$, $c_3 = \log_{10}$) in MOTA and GMOTA according to [11]. The baseline methods are briefly described as follows and their comparison results are shown in Table 2.

KSP tracker [9]: The original KSP tracking framework was firstly proposed by Berclaz etc. It can track multiple objects efficiently but is prone to identity switches due to the neglect of appearance features.

T-MCNF tracker [11]: It is a tracklet-based multi-commodity network that it uses the tracklets as graph nodes instead of individual locations. It needs to run the KSP algorithm two times: the first time is used to generate tracklets being composed of obviously connected nodes and the second time is used to solve the multi-commodity network built for multiple appearance-groups.

MOT-SA tracker [58]: The method uses geometric information regarding 3D scene structure rather than appearance information, and it is very robust for 3D tracking of multiple objects with similar appearances.

MOT-CE traker [59]: The algorithm is designed to track rigid objects in 3D scene and it also uses the geometric information to track.

Table 2. The comparison of MOTA and GMOTA scores in APIDIS dataset.

Trackers	MOTA	GMOTA
MOT-CE [59]	0.799	-
MOT-SA [58]	0.855	-
KSP [9]	0.959	0.714
KSP-AF (60%Color+40%Num)	0.963	0.720
KSP-AF (70%Color+50%Num)	0.970	0.791
T-MCNF [11]	0.92	0.86
KSP-AF (90%Color+50%Num)	0.981	0.923

4.3.4. Visualization of the Improved Tracking Effects

Suppose we can extract about 70% of the jersey color as well as 50% of the jersey number over the sequence; the proposed KSP-AF can better maintain the players' identities compared with the original KSP method. Figure 12 illustrates the improved tracking effects using APIDIS and STU datasets based on KSP-AF. In Figure 12a–d), although two players annotated with thicker line (yellow player 4 and blue player 7) come very close to each other from frame 350 to 380, their identities remain unchanged in frame 393, which is impossible when the original KSP approach is applied in the same sequence. The same improvement can be seen in Figure 12e–h) when the KSP-AF is used in our STU dataset.

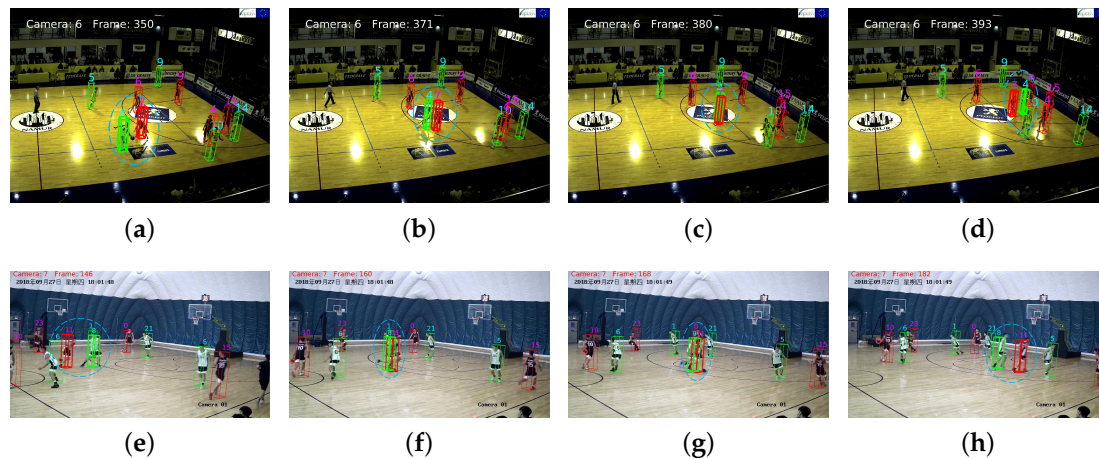


Figure 12. Identity switches can be successfully avoided based on our KSP-AF tracking method in APIDIS and STU datasets. (a–d) Part of the tracking effects in APIDIS dataset. Two players annotated with thicker line (yellow player with number 4 and blue one with number 7) come close to each other from frame 350 to 379 and then separate apart at frame 380. After that, they can still remain their own identities at frame 393, which is impossible when the original KSP approach is applied; (e–h) the same effectiveness of our KSP-AF method when it is applied in STU dataset.

4.3.5. Discussion

In order to test the performance of our proposed method, we did three groups of experiments. The first two experiments were working toward the public 1500-frames APIDIS dataset while the third one was implemented on our own STU dataset (10 periods of long sequences). Based on these experiments, we can have the following discussions:

- (1) Compared with the original KSP tracking which ignores appearance features, the identity switches problem in multiple-player tracking can be obviously improved by considering appearance cues (see Figures 9 and 10). It also can be seen that the combination of two features is better for KSP-AF tracking than the case when only color feature is included. Although the MOTA and GMOTA scores can not be largely improved or even drop at the beginning when very few features are available, they can surpass the original KSP method when more than 60% jersey color and 40% jersey number are extracted. Ideally, both of the MOTA and GMOTA scores of our KSP-AF can reach 1 if we can extract jersey color and jersey number for each player in all frames, as shown in Figure 10 (100%Color+100%Num).
- (2) It also can be seen from Table 2 that KSP-based methods are better adaptable to the APIDIS dataset than other related methods (e.g., MOT-CE [59] (MOTA = 0.799) and MOT-SA [58] (MOTA = 0.855)). Among them, KSP-AF (70%Color+50%Num) can gain an obviously large improvement compared to original KSP. When it reaches to 90%Color+50%Num, it can even defeat the state-of-the-art method (T-MCNF [11]).
- (3) Finally, we use longer sequences (STU dataset collected by our own) to further verify the effectiveness of our proposed KSP-AF. The STU dataset contains 10 periods of sequences

varying from 600 to 4400 frames, and more occlusions, but higher image resolution compared with the public APIDIS dataset. The original KSP method and our KSP-AF with 70% color and 50% number are respectively applied in these sequence. Experimental results show that our method (70% color + 50% number) can maintain the player identity more steadily than the original KSP method.

To sum up, our proposed KSP-AF method can avoid more identity switches if given more features. And the basic requirements of extracting 70% color and 50% number information are not very challenging considering that many advanced feature extraction methods are emerging [50–52]. In addition, our method is also flexible for taking more features into consideration. More features would make our method more robust to occlusions and miss-detections.

5. Conclusions and Future Work

As a typical tracking-by-detection method, the existing KSP algorithm has been widely used for multi-view basketball player tracking. While effective and fast enough, the neglect of the appearance model in original KSP tracking process easily leads to identity switches when two or more players are intertwined with each other. This paper addresses this problem by taking the appearance features into account based on KSP tracking framework. Although in this paper, only two features are applied for experiments, our proposed KSP-AF method can take multiple features into consideration with assigning different weights. Furthermore, we carried out extensive experiments to quantify the requirements of appearance features in KSP-AF tracking. Experiments demonstrate that if given 70% of the jersey colors and 50% of the jersey numbers, the number of false negatives, false positives and IDSW or GIDSW are significantly reduced. From the aspect of a standard evaluation metric, both MOTA and GMOTA scores (especially the GMOTA) increase a lot. Moreover, our method is very flexible in that it allows more appearance features to be considered to result in a more robust tracking. Thus, compared with the existing KSP tracking algorithm, our proposed KSP-AF method can better remain players' identities when it is used for multi-player tracking in multi-view sports videos. However, since more features can make our proposed approach more robust, our future work will focus on how to accurately extract them. In addition, we will also explore some potential applications based on the generated multi-player trajectories for sports broadcasting, such as visualizing a player's running trajectory, tallying how far a player has ran, automatic detecting a goal or passing event, and evaluating player or team performance.

Author Contributions: Conceptualization, Q.L. and Y.P.; formal analysis, W.W.; funding acquisition, M.X.; investigation, Q.L.; methodology, W.W. and Y.P.; software, W.W., Y.Y., and R.Z.; project administration, Y.P.; supervision, M.X.; validation, W.W., Y.Y., and R.Z.; writing—original draft, Q.L. and W.W.; writing—review and editing, Q.L., W.W., and M.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China (NSFC 61673163), Chang-Zhu-Tan National Indigenous Innovation Demonstration Zone Project (2017XK2102), Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing (IRT2018003), and the Chinese Scholarship Council (CSC Student ID 201706130020).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The deduction process of Formula (5) is detailed described as follows:

$$\begin{aligned}
 \hat{m} &= \arg \max_{x \in \Omega} \log \prod_{t,i} P(X_i^t = x_i^t, Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t) \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \log P(X_i^t = x_i^t, Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t) \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \log \left[P(Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t, X_i^t = x_i^t) \cdot \right. \\
 &\quad \left. P(X_i^t = x_i^t | I^t) \right] \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \left[\log P(Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t, X_i^t = x_i^t) \right. \\
 &\quad \left. + \log P(X_i^t = x_i^t | I^t) \right] \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \left[(1 - x_i^t) \log P(Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t, X_i^t = 0) \right. \\
 &\quad \left. + x_i^t \log P(Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t, X_i^t = 1) \right. \\
 &\quad \left. + (1 - x_i^t) \log P(X_i^t = 0 | I^t) \right. \\
 &\quad \left. + x_i^t \log P(X_i^t = 1 | I^t) \right] \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \left[x_i^t \log P(Y_i^t = Y_{j \in \mathbb{N}(i)}^{t+1} | I^t, X_i^t = 1) \right. \\
 &\quad \left. + x_i^t \log \frac{P(X_i^t = 1 | I^t)}{P(X_i^t = 0 | I^t)} \right] \\
 &= \arg \max_{x \in \Omega} \sum_{t=1}^T \sum_{i=1}^n \left[\log \frac{\rho_i^t \cdot s_{i,j \in \mathbb{N}(i)}^t}{1 - \rho_i^t} \right] \cdot x_i^t.
 \end{aligned}$$

References

1. Thomas, G.; Gade, R.; Moeslund, T.B.; Carr, P.; Hilton, A. Computer vision for sports: Current applications and research topics. *Comput. Vis. Image Underst.* **2017**, *159*, 3–18. [[CrossRef](#)]
2. Chen, H.T.; Chou, C.L.; Fu, T.S.; Lee, S.Y.; Lin, B.S.P. Recognizing tactic patterns in broadcast basketball video using player trajectory. *J. Vis. Commun. Image Represent.* **2012**, *23*, 932–947. [[CrossRef](#)]
3. Manafifard, M.; Ebadi, H.; Moghaddam, H.A. Appearance-based multiple hypothesis tracking: Application to soccer broadcast videos analysis. *Signal Process. Image Commun.* **2017**, *55*, 157–170. [[CrossRef](#)]
4. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Zhao, X.; Kim, T.K. Multiple object tracking: A literature review. *arXiv* **2014**, arXiv:1409.7618.
5. Zhang, L.; Van Der Maaten, L. Preserving structure in model-free tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 756–769. [[CrossRef](#)] [[PubMed](#)]
6. Li, M.; He, X.; Wei, Z.; Wang, J.; Mu, Z.; Kuijper, A. Enhanced Multiple-Object Tracking Using Delay Processing and Binary-Channel Verification. *Appl. Sci.* **2019**, *9*, 4771. [[CrossRef](#)]
7. Liu, P.; Li, X.; Liu, H.; Fu, Z. Online Learned Siamese Network with Auto-Encoding Constraints for Robust Multi-Object Tracking. *Electronics* **2019**, *8*, 595. [[CrossRef](#)]
8. Yang, B.; Nevatia, R. Online learned discriminative part-based appearance models for multi-human tracking. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 484–498.
9. Berclaz, J.; Fleuret, F.; Turetken, E.; Fua, P. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1806–1819. [[CrossRef](#)]

10. Nishikawa, Y.; Sato, H.; Ozawa, J. Performance evaluation of multiple sports player tracking system based on graph optimization. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2903–2910.
11. Shitrit, H.B.; Berclaz, J.; Fleuret, F.; Fua, P. Multi-commodity network flow for tracking multiple people. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1614–1627. [[CrossRef](#)]
12. Al-Ali, A.; Almaadeed, S. A review on soccer player tracking techniques based on extracted features. In Proceedings of the 2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA), Muscat, Oman, 19–21 December 2017; pp. 1–6.
13. Cai, Y.; de Freitas, N.; Little, J.J. Robust visual tracking for multiple targets. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 107–118.
14. Fu, T.S.; Chen, H.T.; Chou, C.L.; Tsai, W.J.; Lee, S.Y. Screen-strategy analysis in broadcast basketball video using player tracking. In Proceedings of the Visual Communications and Image Processing (VCIP), Tainan, Taiwan, 6–9 November 2011; pp. 1–4.
15. He, M.; Luo, H.; Hui, B.; Chang, Z. Pedestrian Flow Tracking and Statistics of Monocular Camera Based on Convolutional Neural Network and Kalman Filter. *Appl. Sci.* **2019**, *9*, 1624. [[CrossRef](#)]
16. Breitenstein, M.D.; Reichlin, F.; Leibe, B.; Koller-Meier, E.; Van Gool, L. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1820–1833. [[CrossRef](#)] [[PubMed](#)]
17. Yang, Y.; Li, D. Robust player detection and tracking in broadcast soccer video based on enhanced particle filter. *J. Vis. Commun. Image Represent.* **2017**, *46*, 81–94. [[CrossRef](#)]
18. Itoh, H.; Takiguchi, T.; Ariki, Y. 3D tracking of soccer players using time-situation graph in monocular image sequence. In Proceedings of the 2012 21st International Conference on Pattern Recognition (ICPR), Tsukuba, Japan, 11–15 November 2012; pp. 2532–2536.
19. Najafzadeh, N.; Fotouhi, M.; Kasaei, S. Multiple soccer players tracking. In Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing (AISP), Mashhad, Iran, 3–5 March 2015; pp. 310–315.
20. Baysal, S.; Duygulu, P. Sentioscope: a soccer player tracking system using model field particles. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1350–1362. [[CrossRef](#)]
21. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *arXiv* **2018**, arXiv:1809.02165.
22. Jiang, H.; Fels, S.; Little, J.J. A linear programming approach for multiple object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 18–23 June 2007; pp. 1–8.
23. Berclaz, J.; Fleuret, F.; Fua, P. Multiple object tracking using flow linear programming. In Proceedings of the 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), Snowbird, UT, USA, 7–9 December 2009; pp. 1–8.
24. Zhang, L.; Li, Y.; Nevatia, R. Global data association for multi-object tracking using network flows. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
25. Butt, A.A.; Collins, R.T. Multi-target tracking by lagrangian relaxation to min-cost network flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1846–1853.
26. Henriques, J.F.; Caseiro, R.; Batista, J. Globally optimal solution to multi-object tracking with merged measurements. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2470–2477.
27. Wu, Z.; Kunz, T.H.; Betke, M. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1185–1192.
28. Zamir, A.R.; Dehghan, A.; Shah, M. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 343–356.

29. Wu, Z.; Thangali, A.; Sclaroff, S.; Betke, M. Coupling detection and data association for multiple object tracking. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1948–1955.
30. Kang, T.; Mo, Y.; Pae, D.; Ahn, C.; Lim, M. Robust visual tracking framework in the presence of blurring by arbitrating appearance-and feature-based detection. *Measurement* **2017**, *95*, 50–69. [[CrossRef](#)]
31. Li, Z.; Gao, S.; Nai, K. Robust object tracking based on adaptive templates matching via the fusion of multiple features. *J. Vis. Commun. Image Represent.* **2017**, *44*, 1–20. [[CrossRef](#)]
32. Liu, J.; Carr, P.; Collins, R.T.; Liu, Y. Tracking sports players with context-conditioned motion models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1830–1837.
33. Shitrit, H.B.; Berclaz, J.; Fleuret, F.; Fua, P. Tracking multiple people under global appearance constraints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
34. KC, A.K.; Delannay, D.; Jacques, L.; De Vleeschouwer, C. Iterative hypothesis testing for multi-object tracking with noisy/missing appearance features. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 412–426.
35. Amit Kumar, K.; De Vleeschouwer, C. Discriminative label propagation for multi-object tracking with sporadic appearance features. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2000–2007.
36. Lu, W.L.; Ting, J.A.; Little, J.J.; Murphy, K.P. Learning to track and identify players from broadcast sports videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1704–1716.
37. Sabirin, H.; Sankoh, H.; Naito, S. Automatic Soccer Player Tracking in Single Camera with Robust Occlusion Handling Using Attribute Matching. *IEICE Trans. Inf. Syst.* **2015**, *98*, 1580–1588. [[CrossRef](#)]
38. Bozorgtabar, B.; Goecke, R. Efficient multi-target tracking via discovering dense subgraphs. *Comput. Vis. Image Underst.* **2016**, *144*, 205–216. [[CrossRef](#)]
39. Milan, A.; Gade, R.; Dick, A.; Moeslund, T.B.; Reid, I. Improving global multi-target tracking with local updates. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 174–190.
40. Wu, C.; Sun, H.; Wang, H.; Fu, K.; Xu, G.; Zhang, W.; Sun, X. Online Multi-Object Tracking via Combining Discriminative Correlation Filters With Making Decision. *IEEE Access* **2018**, *6*, 43499–43512. [[CrossRef](#)]
41. Schulter, S.; Vernaza, P.; Choi, W.; Chandraker, M. Deep network flow for multi-object tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2730–2739.
42. Yang, Y.; Xu, M.; Wu, W.; Zhang, R.; Peng, Y. 3D Multiview Basketball Players Detection and Localization Based on Probabilistic Occupancy. In Proceedings of the 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, Australia, 10–13 December 2018; pp. 2730–2739.
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
44. Fleuret, F.; Berclaz, J.; Lengagne, R.; Fua, P. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 267–282. [[CrossRef](#)] [[PubMed](#)]
45. Usamentiaga, R.; Garcia, D. Multi-camera calibration for accurate geometric measurements in industrial environments. *Measurement* **2019**, *134*, 345–358. [[CrossRef](#)]
46. Eppstein, D. Finding the k shortest paths. *SIAM J. Comput.* **1998**, *28*, 652–673. [[CrossRef](#)]
47. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
48. Boyer, R.S.; Moore, J.S. MJRTY—A fast majority vote algorithm. In *Automated Reasoning*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 105–117.
49. Gunn, S.R. Support vector machines for classification and regression. *ISIS Tech. Rep.* **1998**, *14*, 5–16.
50. Gerke, S.; Muller, K.; Schafer, R. Soccer jersey number recognition using convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 17–24.

51. Li, G.; Xu, S.; Liu, X.; Li, L.; Wang, C. Jersey Number Recognition with Semi-Supervised Spatial Transformer Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1783–1790.
52. Gerke, S.; Linnemann, A.; Müller, K. Soccer player recognition using spatial constellation features and jersey number recognition. *Comput. Vis. Image Underst.* **2017**, *159*, 105–115. [[CrossRef](#)]
53. Jean-François Prior; Philippe Delmulle. APIDIS Dataset. Available online: <https://sites.uclouvain.be/ispgroup/Softwares/APIDIS> (accessed on 29 July 2016).
54. Osvald, L. K-th Shortest Path C++ Library. Available online: <https://github.com/losvald/ksp> (accessed on 21 March 2013).
55. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *J. Image Video Process.* **2008**, *2008*, 1. [[CrossRef](#)]
56. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
57. Kasturi, R.; Goldgof, D.; Soundararajan, P.; Manohar, V.; Garofolo, J.; Bowers, R.; Boonstra, M.; Korzhova, V.; Zhang, J. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 319–336. [[CrossRef](#)]
58. Sekii, T. Robust, real-time 3d tracking of multiple objects with similar appearances. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4275–4283.
59. Ghedia, N.S.; Vithalani, C.; Kothari, A. A Novel Approach for Monocular 3D Object Tracking in Cluttered Environment. *Int. J. Comput. Intell. Res.* **2017**, *13*, 851–864.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).