



# MF-SORT: Simple Online and Realtime Tracking with Motion Features

Heng Fu<sup>1</sup>, Lifang Wu<sup>1</sup>, Meng Jian<sup>1(✉)</sup>, Yuchen Yang<sup>1</sup>,  
and Xiangdong Wang<sup>2</sup>

<sup>1</sup> Beijing University of Technology, Beijing 100124, China  
mjian@bjut.edu.cn

<sup>2</sup> Sports Science Research Institute of the State Sports General Administration,  
Beijing, China

**Abstract.** Multiple object tracking (MOT) plays a key role in video analysis. On MOT, DeepSORT (Simple Online and Realtime Tracking with a deep association metric) performs effectively by combining features of appearance and motion for estimating data association. However, computing with multiple features are time consuming. In certain applications, cameras are static, such as pedestrian surveillance, sports video analysis and so on. Here, without camera movement the motion trajectories of objects are generally possible to estimate. The introduction of more features cannot improve the performance of object tracking discriminatively. Furthermore, the time cost rises evidently. To address this problem, we propose a novel Simple Online and Realtime Tracking with motion features (MF-SORT). By focusing on the motion features of the objects during data association, the proposed scheme is able to take a trade-off between performance and efficiency. The experimental results on the MOT Challenge benchmark and MOT-SOCCER (newly established in this work) demonstrate that the proposed method is much faster than DeepSORT with the comparable accuracy.

**Keywords:** Multiple Object Tracking · Online tracking · Static camera video · Benchmark · Video analysis

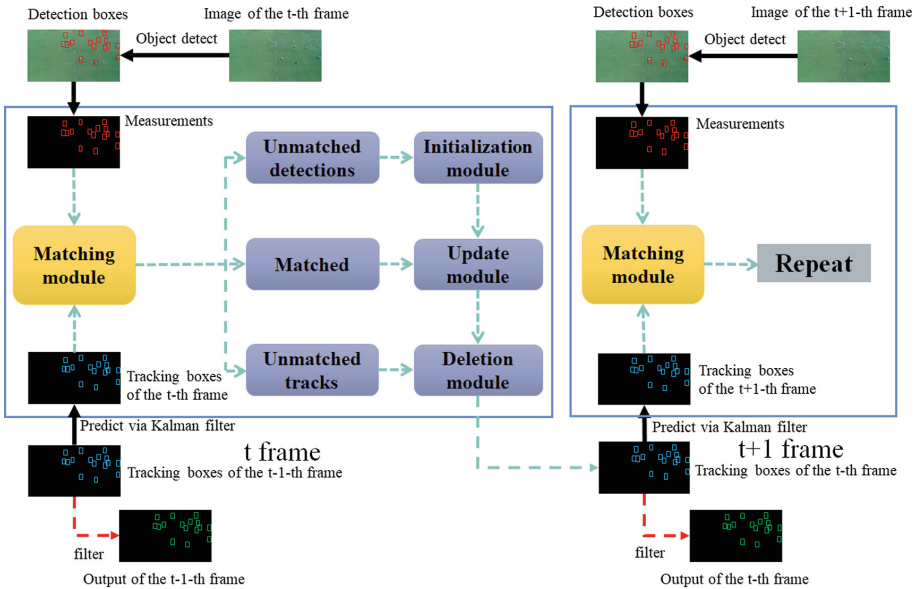
## 1 Introduction

Multiple object tracking (MOT) is an essential task in video analysis, such as video pedestrian surveillance [1, 2], sport players analysis [3, 4], autopilot [5], etc. Currently, the state-of-the-art methods of MOT are primarily based on a tracking-by-detection paradigm [6–11], taking advantage of progress in object detection. The key challenge in this framework is data association, which aims to accurately associate existing object trajectories, according to the detection results in each frame.

The existed MOT schemes can be categorized into three classes: online tracking [12–15], near-online tracking [15] and offline tracking [16, 17]. DeepSORT [9] is one of representative online tracking algorithms with high tracking accuracy but slow processing speed, due to introducing the objects' appearance features.

In real application scenarios such as sports video analysis, pedestrian surveillance and so on, the videos are captured in view of the static cameras. The object trajectory is generally predictable and appearance features are not necessary.

Motivated by the above, we propose a scheme of Simple Online and Realtime Tracking with motion features (MF-SORT). The framework of the proposed scheme is as illustrated in Fig. 1. First, the location of tracking boxes is estimated based on Kalman filter. Then, the data from the object detections (measurements) and the predicted estimations (tracking boxes) are matched based on motion features. Finally, according to the matching results, initialization, update and deletion modules are determined and implemented to produce tracking results. The experimental results demonstrate that the proposed scheme is more adaptable to the static camera video scene.

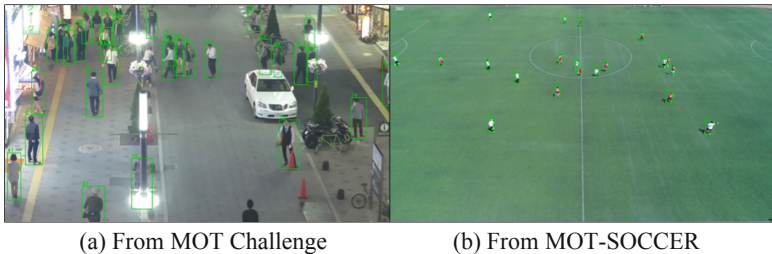


**Fig. 1.** The framework of the proposed MF-SORT.

The popular benchmark database for evaluating MOT algorithms is MOT Challenge. It focuses on video surveillance and provides numerous false positive (false detection) and false negative (missed detection) detection results. It is one of the bottlenecks that influences the effectiveness of the MOT algorithms. In addition, in this paper, we establish a supplementary database referred as MOT-SOCCER. It consists of 10 clips of static camera sports videos with annotations. This benchmark provides high-quality public detection whose F1-score is over 90%. An exemplary frame from MOT Challenge and MOT-SOCCER are shown in Fig. 2.

The main contributions of this work are as follows:

1. We propose a novel simple online and realtime object tracking algorithm MF-SORT. Simply with motion features in data association, it is able to track the objects in the static cameras effectively and efficiently. The comparative experimental results demonstrate that the proposed scheme can achieve competitive results with less computation complexity in MOT Challenge and MOT-SOCCER benchmark.



**Fig. 2.** Example frames in the MOT Challenge and MOT-SOCCER.

2. We establish a benchmark MOT-SOCCER which provides a high-quality detection. The benchmark consists of 10 clips of sports videos with static camera. It helps to enrich the performance assessments of MOT researches.

## 2 The Proposed Scheme

### 2.1 The Framework of the Proposed Scheme

The scheme is proposed by modifying DeepSORT in the initialization and matching stages. The framework is shown in Fig. 1. Assume that there are  $M$  detection boxes in the  $(t)$ -th frame. And there are  $N$  tracking boxes from the Kalman filter based on the results in the  $(t - 1)$ -th frame. The model of Kalman filter is defined on the eight-dimensional state space  $(u; v; a; h; \dot{u}; \dot{v}; \dot{a}; \dot{h})$ , which contains the center of the bounding box  $(u; v)$ , the aspect ratio  $a$  and height  $h$  of the bounding box. It is intuitive to employ the output of the Kalman filter as the tracking boxes. The  $M$  detection boxes and  $N$  tracking boxes are fed into the matching modules for association matching. The similarity between detection boxes and the tracking boxes are computed in matching module, based on their motion features.

There are three possible cases in matching results: (1) Matched: It means that some detection boxes and tracking box are successfully matched. Suppose that  $M1$  boxes are matched. (2) Unmatched detections: It means that some detection boxes have not been matched to the tracking boxes. These boxes possibly are the new objects in the  $(t)$ -th frame. The number should be  $M - M1$ . (3) Unmatched tracks: It means that some tracking boxes have not been matched with the detection boxes. The number of boxes should be  $N - M1$ . Following each case, the corresponding operation is then elaborately designed. For case “matched”, the bounding boxes of the objects are updated from the tracking box to the corresponding detection boxes. For case “unmatched detections”, these detection boxes are initialized as the bounding boxes of the new objects. For case “unmatched tracks”, the objects of these tracking boxes may not stay in this frame, they are deleted. The remaining of this section would introduce the corresponding details of matching, initialization, update and deletion module respectively.

### 2.2 Matching Module

In order to improve the matching efficiency, the priority of all the tracking boxes are estimated based on the *time\_since\_update*. Sequentially, cascade matching [9] is

implemented based on the priorities. For the tracking boxes which have not been matched in the cascade stage, global matching is further employed, in which the similarity between all the unmatched tracking boxes and unmatched the detection boxes are computed by appropriate metrics.

Because the videos are collected with static cameras, the trajectory of objects is predictable and motion features are robust and sufficient for data association. Mahalanobis distance has the characteristic of scale independence. Therefore, we introduce the squared Mahalanobis distance of motion features instead of the cosine distance of appearance features in DeepSORT to measure the similarity between the tracking box and detection box:

$$d(i, j) = (x_j - y_i)^T C_i^{-1} (x_j - y_i) \quad (1)$$

where the projection distribution of the ( $i$ )-th tracking box is represented as  $(y_i, C_i)$ , which can be obtained from the Kalman filter directly. And the ( $j$ )-th detection bounding box is represented as  $x_j$ . The metric computation is faster than appearance feature based in DeepSORT, and it is more reliable than the IoU (Intersection-over-Union) metric in SORT [8]. The detailed algorithm is summarized in Algorithm1.

---

**Algorithm 1** Matching Module
 

---

**Input:** Trackers  $T = \{t_1, t_2, \dots, t_n\}$ , Detections  $D = \{d_1, d_2, \dots, d_m\}$ , Maximum age  $A_{max}$

**Output:** Matched  $M$ , Unmatched detections  $U_d$ , Unmatched tracks  $U_t$

---

Select confirmed tracks  $T_c$  and tentative tracks  $T_t$  from  $T$

Initialize set of unmatched detections  $U_d \leftarrow D$

Initialize set of matched  $M \leftarrow \Phi$

**for**  $l \in \{1, 2, \dots, A_{max}\}$  **do**

Select tracks by age  $T_l \leftarrow \{t_i \in T_c \mid t_i'age = l\}$

$C \leftarrow$  compute\_cost\_matrix( $T_l, U_d$ ) using eq.1

$M_l \leftarrow$  hungarian\_assignment( $C$ )

**for**  $(x_i, y_j) \in M_l$  **do**

**if**  $C[(x_i, y_j)] > th_{ca}$  (9.488) **then**  $M_l \leftarrow$  remove( $x_i, y_j$ ) from  $M_l$  **end**

**else**  $U_d \leftarrow$  remove  $d_{y_j}$  from  $U_d$  **end**

**end for**

$M \leftarrow M \cup M_l$

**end for**

Initialize set of unmatched tracks  $U_t \leftarrow \Phi$

**for**  $t_i \in T$  **do**

**if**  $i \notin M$  **then**  $U_t \leftarrow U_t \cup t_i$  **end**

**end for**

$U_t \leftarrow U_t \cup T_t$

$C \leftarrow$  compute\_cost\_matrix( $U_t, U_d$ ) using eq.1

$M_g \leftarrow$  hungarian\_assignment( $C$ )

**for**  $(x_i, y_j) \in M_g$  **do**

**if**  $C[(x_i, y_j)] > th_{go}$  (13.277) **then**  $M_g \leftarrow$  remove( $x_i, y_j$ ) from  $M_g$  **end**

**else**  $U_d \leftarrow$  remove  $d_{y_j}$  from  $U_d$   $U_t \leftarrow$  remove  $t_{x_j}$  from  $U_t$  **end**

**end for**

$M \leftarrow M \cup M_g$

**return**  $M, U_d, U_t$

---

Further, it is necessary to delete the impossible associations by setting a threshold of the Mahalanobis distance. In cascade matching, the threshold  $th_{ca}$  for Mahalanobis distance is set as 9.488 (this threshold corresponds to a confidence value 0.95 in four-dimensional chi-square distribution). While in global matching stage, the threshold  $th_{go}$  is set as 13.277 (this threshold corresponds to a confidence value 0.99 in four-dimensional chi-square distribution), to obtain broader range of matching result.

### 2.3 Initialization, Update and Deletion Module

As shown in Sect. 2.1, there are three cases for matching results: matched, unmatched detections and unmatched tracks. For each case, one of the corresponding operations (initialization, update and deletion) are then conducted respectively.

The update and deletion module in DeepSORT [9] are remained in the proposed MF-SORT method. When the defined Kalman filter estimates the tracking boxes in each frame [21], the time interval (*time\_since\_update*) will be increased by 1. This value is reset to 0 in the update module after each successful match. When a tracker has not been successfully matched for a long time, this variable will be accumulated with each frame of Kalman filter estimation until it exceeds the maximum age we set ( $max\_age = 5$ ), and then the tracker will be deleted. More details in the update module and the deletion module are preserved for tentative tracker. In the update module, trackers with more than 3 successful matches hits ( $hits = 3$ ) can be set to a confirmed state. In the deletion module, the tentative tracker will be deleted immediately when it does not successfully match in matching module.

In the initialization module, an additional gating method is introduced into the initialization module. The aim is to reduce the false trackers initialized by erroneous detection and avoid subsequent adverse impacts on tracking. In this work, IoU between each unmatched detection box and all tracking boxes are evaluated. In case that the IoU is higher than the given threshold ( $th_{gating} = 0.7$ ), it means that the detection box is a false positive detection. It is initialized as the bounding box of a new object. The detailed initialization algorithm is shown in Algorithm 2.

---

#### Algorithm 2 Initialization Module

---

**Input:** Trackers  $T = \{t_1, t_2, \dots, t_n\}$ , Unmatched Detections  $U_d = \{d_{y_1}, d_{y_2}, \dots, d_{y_j}\}$

**Output:** Trackers  $T$

---

```

for  $d_{y_j} \in U_d$  do
  if  $IoU(d_{y_j}, T) > th_{gating} (0.7)$  then  $U_d \leftarrow$  remove  $d_{y_j}$  from  $U_d$  end
end for
for  $d_{y_j} \in U_d$  do
  Initialize Kalman filter tracker  $t_{new}$  with  $d_{y_j}$ 
  Set  $t_{new}$ 's state = tentative
  Set  $t_{new}$ 's time_since_update = 0
  Set  $t_{new}$ 's hits = 0
  Set  $t_{new}$ 's id = the_next_id
   $T \leftarrow T \cup t_{new}$ 
end for
return  $T$ 

```

---

### 3 Benchmark<sup>1</sup>

#### 3.1 Overview

In most tracking-by-detection algorithms, the results are influenced greatly by the performance of object detection. In other words, the quality of detection boxes seriously impacts the performance of these methods. The MOT Challenge benchmark [18] are usually used for evaluating MOT algorithms, while the quality of public detection in MOT16 or MOT17 is not proper due to its complicated background. This directly results in that some of the estimated detection boxes are false. To alleviate the problem, MOT-SOCCER benchmark is established.

The dataset consists of 10 clips of amateur soccer videos that are collected with a static camera installed in a straight view of high position. It provides the detection boxes with F1-score over 90%. Some example frames in MOT-SOCCER are shown in Fig. 3.

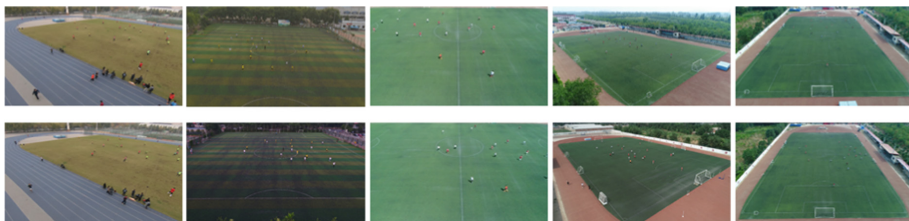
Different from other tracking tasks, the objects in MOT-SOCCER display smaller scale changes as well as relatively similar appearance features. Although MOT-SOCCER is collected from soccer matches, it includes many specific cases in MOT Challenge such as inter-target occlusion, target disappearing and complex movement. Therefore, the MOT-SOCCER can also make sense of realistic MOT task.

We have compiled total 10 clips, half of which are applied to training and the rest to testing. An overview of this benchmark is shown in Table 1.

**Table 1.** Overview of the sequences currently included in the MOT-Soccer benchmark

Name	FPS	Resolution	Length	Tracks	Boxes	Density
MOT-S-01	30	2704 × 1520	920 (00:30)	20	11733	12.7
MOT-S-03	25	4096 × 2160	644 (00:25)	23	14398	22.3
MOT-S-05	30	2720 × 1530	900 (00:30)	27	16136	17.9
MOT-S-07	30	4096 × 2160	899 (00:30)	22	19800	22
MOT-S-09	30	4096 × 2160	735 (00:24)	22	16192	22
Total training			4098 (02:19)	114	78259	19.1
MOT-S-02	30	2704 × 1520	867 (00:28)	21	11733	16.1
MOT-S-04	25	4096 × 2160	644 (00:25)	23	14398	21.8
MOT-S-06	30	2720 × 1530	900 (00:30)	27	16136	19.2
MOT-S-08	30	4096 × 2160	899 (00:30)	22	19800	22
MOT-S-10	30	4096 × 2160	659 (00:21)	22	16192	22
Total testing			3969 (02:14)	114	79679	20.1

<sup>1</sup> MOT-SOCCER benchmark can be downloaded at <https://github.com/jozeeandfish/motsoccer>.



**Fig. 3.** An overview of the MOT-Soccer dataset. Top: training sequences; bottom: test sequences.

### 3.2 Detection

In order to support multiple object tracking methods, we provide a high-quality public detection results on MOT-SOCCER database, which is generated by LFFD object detection [20]. Its F1-score reaches 93.62%. It is much higher than that in MOT Challenge benchmark. The detailed performance is shown in Table 2.

**Table 2.** Public detection performance provided in each benchmark. The IoU threshold used in the evaluation is set to 0.5.

	Detection method	Precision	Recall	F1-score
MOT-SOCCER	LFFD [20]	<b>94.42</b>	<b>92.84</b>	<b>93.62</b>
MOT Challenge	MOT16-DPM [18]	60.34	42.89	50.14
	MOT17-FRCNN [19]	68.82	45.18	54.55
	MOT17-SDP [19]	<b>72.82</b>	<b>53.47</b>	<b>61.66</b>
	POI [7]	69.83	51.20	59.08

### 3.3 Data Format

The data format in MOT-SOCCER are definitely consistent with the MOT Challenge benchmark [18]. All images are converted into JPEG format and named sequentially to a 5-digit file name (e.g. 00001.jpg). Detection and annotation files are comma-separated text files. Each line represents one object instance. It contains 9 properties including frame number, tracking id, coordinates of the bounding box (x, y, w, h), confidence score, and category. In case of any property absent, 1 or -1 is used to fill this vacancy according to the criterion in MOT Challenge [18].

## 4 Experiments

### 4.1 Implement Details

The parameters of the proposed method referred in Sect. 2 have been determined on training sequences, which are provided by MOT-SOCCER. In the reproduced source code, we conduct experiments with the default parameters set in the corresponding paper. Moreover, multiple object tracking performance is evaluated through the MOT Challenge Development Kit [19] provided by A. Milan. The computing device hardware for the experimental application is i7 7700HQ (2.80 GHz), Nvidia GTX 1060.

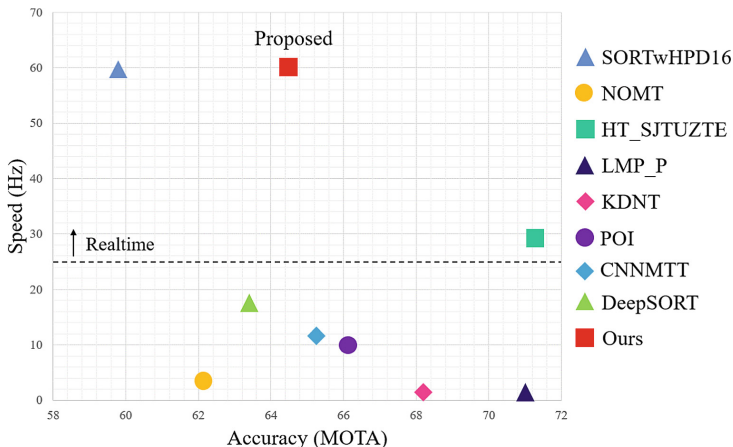
### 4.2 Evaluation on MOT Benchmarks

Many existing methods used POI [7] public detection as inputs in their work, they did not try the SDP public detection or others updated in MOT17 [19] to evaluate tracking performance. Therefore, the best-performance public detection in the benchmark (See Table 2) MOT17-SDP is applied as inputs, and the annotation of MOT17 acts as a ground truth. In this case, the performance of the proposed MF-SORT scheme is compared to that of DeepSORT. The results are shown in Table 3. In addition, we also compared the performance and efficiency of MF-SORT with several state-of-the-art methods as shown in Fig. 4.

**Table 3.** Tracking results on the MOT Challenge training sequences with SDP detection input.

Sequence	Camera	Method	FP↓	FN↓	IDS↓	FM↓	MOTA↑	Hz↑
<b>MOT16-02</b>	<b>Static</b>	MF-SORT	1015	<b>9109</b>	215	<b>271</b>	<b>44.4</b>	<b>46</b>
		DeepSORT	<b>342</b>	9987	<b>128</b>	298	43.7	16
<b>MOT16-04</b>	<b>Static</b>	MF-SORT	226	<b>10890</b>	143	<b>175</b>	<b>76.3</b>	<b>34</b>
		DeepSORT	<b>56</b>	12626	<b>131</b>	352	73.1	12
MOT16-05	Moving	MF-SORT	674	<b>2341</b>	94	<b>107</b>	55.1	<b>173</b>
		DeepSORT	<b>83</b>	2733	<b>56</b>	112	<b>58.5</b>	42
<b>MOT16-09</b>	<b>Static</b>	MF-SORT	71	<b>1702</b>	48	<b>46</b>	<b>65.8</b>	<b>129</b>
		DeepSORT	<b>9</b>	1904	<b>29</b>	55	63.5	20
MOT16-10	Moving	MF-SORT	1036	<b>3241</b>	169	<b>243</b>	65.4	<b>49</b>
		DeepSORT	<b>379</b>	3715	<b>92</b>	244	<b>67.4</b>	8
<b>MOT16-11</b>	<b>Moving</b>	MF-SORT	517	<b>2308</b>	65	<b>80</b>	<b>69.4</b>	<b>123</b>
		DeepSORT	<b>174</b>	2710	<b>46</b>	100	68.9	44
MOT16-13	Moving	MF-SORT	1114	<b>4908</b>	74	120	47.6	<b>50</b>
		DeepSORT	<b>390</b>	5528	<b>31</b>	<b>89</b>	<b>48.9</b>	23
Total		MF-SORT	4653	<b>34499</b>	808	<b>1042</b>	<b>64.4</b>	<b>60</b>
		DeepSORT	<b>1433</b>	39203	<b>513</b>	1250	63.4	18





**Fig. 4.** Benchmark performance of the proposed scheme (MF-SORT) in relation to several state-of-the-art trackers.

The results show that the proposed MF-SORT has obtained higher MOTA (multiple object tracking accuracy) scores than that of DeepSORT in the MOT Challenge training sequences. It is shown that MF-SORT achieves the best performance in videos from static cameras (MOT 16-02, MOT 16-04 and MOT 16-09). Most importantly, the improved scheme is capable to produce a satisfying trade-off between tracking performance and efficiency. The results in Fig. 4 demonstrate that the proposed MF-SORT achieves competitive results with less computational complexity than existing SOTA methods.

### 4.3 Comparison of Tracking Performance with Different Detections

In order to investigate how the quality of detection boxes influences the tracking performance of our proposed scheme, we utilize the detection boxes from POI and MOT17-SDP (The detection performance is shown in Table 2.) and the ground truth (GTP) as inputs respectively. In the videos from static cameras (MOT 16-02, MOT 16-04 and MOT 16-09), the tracking performance of the proposed MF-SORT is compared with that of DeepSORT. The results are shown in Table 4.

**Table 4.** Tracking results in the videos from static camera with different detection quality.

Detection input	Method	FP↓	FN↓	IDS↓	FM↓	MOTA↑
POI	MF-SORT	6558	<b>23298</b>	354	441	<b>57.7</b>
	DeepSORT	<b>1651</b>	30033	<b>122</b>	<b>395</b>	55.5
SDP	MF-SORT	1312	<b>21701</b>	406	<b>492</b>	<b>67.2</b>
	DeepSORT	407	24517	<b>288</b>	705	64.7
GTP	MF-SORT	378	<b>189</b>	<b>0</b>	<b>0</b>	99.2
	DeepSORT	19	351	25	13	<b>99.4</b>

From Fig. 4 we can see that both DeepSORT and MF-SORT achieve performance improvement with the quality of detection results increasing. Moreover, the proposed scheme achieves better performance under high-quality detection and also has higher processing speed.

#### 4.4 Evaluation on MOT-SOCCER Benchmarks

Aiming at comprehensively evaluating multiple object tracking performance of the proposed MF-SORT in static camera videos, a comparative experiment is carried out on the MOT-SOCCER benchmark we established. The performance of the MF-SORT compared to DeepSORT methods in the test sequences of MOT-SOCCER is shown in Table 5.

The result shows that MF-SORT achieves a slightly increasing MOTA score in MOT-SOCCER compared to DeepSORT, and made a balance between performance and processing speed, which is similar to those in the MOT Challenge benchmark. Since the detection quality in MOT-SOCCER is better than that in the MOT Challenge, we could conclude that the proposed scheme is more effective and efficient than DeepSORT in the condition of good detection quality.

**Table 5.** Tracking results on the test sequences of MOT-SOCCER benchmark.

Sequence	Method	FP↓	FN↓	IDS↓	FM↓	MOTA↑	Hz↑
MOT-S-02	MF-SORT	192	<b>647</b>	28	<b>50</b>	<b>93.8</b>	<b>18</b>
	DeepSORT	<b>163</b>	760	<b>15</b>	62	93.3	8
MOT-S-04	MF-SORT	145	<b>221</b>	19	<b>39</b>	<b>97.3</b>	<b>32</b>
	DeepSORT	<b>102</b>	277	<b>6</b>	44	<b>97.3</b>	14
MOT-S-06	MF-SORT	234	<b>248</b>	<b>18</b>	<b>46</b>	<b>97.1</b>	<b>95</b>
	DeepSORT	<b>210</b>	311	24	48	96.8	27
MOT-S-08	MF-SORT	46	<b>417</b>	28	<b>49</b>	<b>97.5</b>	<b>44</b>
	DeepSORT	<b>33</b>	516	<b>27</b>	63	97.1	8
MOT16-10	MF-SORT	30	<b>51</b>	3	<b>19</b>	<b>99.4</b>	<b>28</b>
	DeepSORT	<b>23</b>	95	<b>2</b>	<b>19</b>	99.2	6
Total	MF-SORT	647	<b>1584</b>	96	<b>203</b>	<b>97.1</b>	<b>44</b>
	DeepSORT	<b>531</b>	1959	<b>74</b>	236	96.8	12

## 5 Conclusion

In this paper, we propose a novel simple online and realtime tracking with motion features (MF-SORT). It utilizes the motion features instead of appearance features in data association in the tracking-by-detection paradigm, which helps improve efficiency of data association. The experimental results demonstrate that the proposed MF-SORT achieves competitive results with less computational costs compared with state-of-the-art methods. It produces a satisfactory trade-off between performance and efficiency, which is more competent for realtime application scenarios. We also establish an

open-download MOT benchmark MOT-SOCCER, which provides a high-quality detection. It comes to enrich the assessments of MOT methods.

**Acknowledgement.** This work was supported by the National Natural Science Foundation of China under Grant 61702022 and 61802011, in part by the Beijing Municipal Education Committee Science Foundation under Grant KM201910005024, in part by “Ri Xin” Training Programme Foundation for the Talents by Beijing University of Technology.

## References

1. Yang, B., Huang, C., Nevatia, R.: Learning affinities and dependencies for multi-target tracking using a CRF model. In: CVPR 2011, pp. 1233–1240. IEEE (2011)
2. Pellegrini, S., et al.: You’ll never walk alone: modeling social behavior for multi-target tracking. In: ICCV 2009, pp. 261–268. IEEE (2009)
3. Lu, W., et al.: Learning to track and identify players from broadcast sports videos. IEEE Trans. Pattern Anal. Mach. Intell. **35**(7), 1704–1716 (2013)
4. Xing, J., Ai, H., Liu, L., et al.: Multiple player tracking in sports video: a dual-mode two-way bayesian inference approach with progressive observation modeling. IEEE Trans. Image Process. **20**(6), 1652–1667 (2011)
5. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 189–196. Springer, Heidelberg (1994). [https://doi.org/10.1007/3-540-57956-7\\_22](https://doi.org/10.1007/3-540-57956-7_22)
6. Feng, W., et al.: Multiple object tracking with multiple cues and switcher-aware classification. arXiv preprint [arXiv:1901.06129](https://arxiv.org/abs/1901.06129) (2019)
7. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: POI: multiple object tracking with high performance detection and appearance feature. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 36–42. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-48881-3\\_3](https://doi.org/10.1007/978-3-319-48881-3_3)
8. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468. IEEE (2016)
9. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649. IEEE (2017)
10. Long, C., Haizhou, A., Zijie, Z., Chong, S.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, pp. 1–6 (2018)
11. Yoon, Y., et al.: Online multiple object tracking with historical appearance matching and scene adaptive detection filtering. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE (2018)
12. Milan, A., Rezatofighi, S.H., Dick, A.R., Reid, I.D., Schindler, K.: Online multi-target tracking using recurrent neural networks. In: AAAI, vol. 2, p. 4 (2017)
13. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.-H.: Online multi-object tracking with dual matching attention networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 379–396. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01228-1\\_23](https://doi.org/10.1007/978-3-030-01228-1_23)

14. Fang, K., Xiang, Y., Li, X., et al.: Recurrent autoregressive networks for online multiple object tracking. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 466–475. IEEE (2018)
15. Choi, W.: Near-online multi-target tracking with aggregated local flow descriptor. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE (2015)
16. Henschel, R., Leal-Taixe, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multiple object tracking. In: Computer Vision and Pattern Recognition Workshops (CVPRW) (2018)
17. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person reidentification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3539–3548. IEEE (2017)
18. Milan, A., et al.: MOT16: a benchmark for multiple object tracking. arXiv preprint [arXiv: 1603.00831](https://arxiv.org/abs/1603.00831) (2016)
19. Multiple object tracking benchmark. <https://motchallenge.net>. Accessed 26 Apr 2019
20. Xu, D., et al.: LFFD: a light and fast face detector for edge devices. arXiv preprint [arXiv: 1904.10633](https://arxiv.org/abs/1904.10633) (2019)
21. Kalman, R.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(Series D), 35–45 (1960)