Hindawi
Scientific Programming
Volume 2023, Article ID 3895703, 15 pages
https://doi.org/10.1155/2023/3895703

Research Article

SKT-MOT and DyTracker: A Multiobject Tracking Dataset and a Dynamic Tracker for Speed Skating Video

Junwu Wang,[1] Zongmin Li ,[1] Yachuan Li,[1] Shaobo Yang,[1] Ben Wang,[1] and Hua Li[2]

[1]School of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[2]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Correspondence should be addressed to Zongmin Li; lizongmin@upc.edu.cn

Received 17 April 2023; Revised 9 September 2023; Accepted 20 September 2023; Published 18 October 2023

Academic Editor: Jianping Gou

Speed skating serves as a significant application domain for multiobject tracking (MOT), presenting unique challenges such as frequent occlusion, highly similar appearances, and motion blur. To address these challenges, this paper constructs an MOT dataset called SKT-MOT for speed skating and analyzes the shortcomings of existing datasets and methods. Accordingly, we propose a dynamic MOT method called DyTracker. The method builds upon the DeepSORT baseline and enhances three key modules. At the global level, we design the track dynamic management (TDM) algorithm. In the motion branch, a novel metric is proposed to evaluate occlusion and Kalman filter dynamic update (KFDU) is implemented. In the appearance branch, we account for the difference in human posture and propose the feature dynamic selection and updating (FDSU) strategy. This makes our DyTracker flexible and efficient to achieve a multiobject tracking accuracy (MOTA) of 93.70% and identification F1 (IDF1) score of 92.39% on SKT-MOT, which is a significant advantage over existing SOTA methods. To validate the generalization of our proposed module, two dynamic update modules are inserted into other methods and validated on the public dataset MOT17, and the accuracy is generally improved by 0.2%–0.6%.

1. Introduction

Speed skating is of significant importance as a prominent winter Olympics event, with substantial influence worldwide. The application of multiobject tracking (MOT) technology to provide supplementary data analysis for speed skaters holds practical significance. Tracking speed skaters presents a distinctive case within MOT, entailing numerous unique challenges. This paper aims to enable MOT to be efficiently completed in speed skating scenarios.

MOT is a classic problem in computer vision that aims to identify and track all objects of a specific category in a video. Early methods [1, 2] relied mainly on handcrafted features to compute the similarity between frames and achieve object association. With the development of deep learning, methods based on deep neural networks have gradually become mainstream. SORT [3] was the first method to apply object detection networks to MOT, completing the association task through the Kalman filter (KF) [4] and the Hungary matching algorithm [5]. This was also the first tracking-by-detection (TBD) paradigm framework. DeepSORT [6] introduced a reidentification module on this basis, which jointly completes tracking using appearance features and motion clues. It is the most widely used method in the industry. With the development of MOT, some methods [7–11, 21, 22] integrated these modules into a unified network to reduce the inference time and attempted to achieve the end-to-end. These methods are called joint detection and tracking (JDT) paradigms. Both paradigms have made significant progress in recent years.

As MOT technology matures, its applications become increasingly widespread. In certain competitive sports fields [12–14, 20], MOT technology is widely used to guide the training and competition of athletes. However, in the speed skating scene, the development of MOT is relatively slow, mainly due to a lack of data and unique challenges in speed skating. To this end, we first construct SKT-MOT, an MOT dataset for short-track speed skating, consisting of 56 video clips with three scenes and a total of 53,178 images. Based on this, we also developed object detection and re-ID datasets for speed skaters. Additionally, we analyze the unique challenges and advantages of speed skating scenarios. These challenges include:

(1) Frequent occlusions between athletes.
(2) Athletes dress similarly or even identically.
(3) Speed skating is fast and prone to motion blur.

These difficulties hinder the efficient completion of MOT tasks. But the speed skating scene has advantages, with its advantages lying in a relatively small and fixed number of athletes and a relatively clean environment.

Aiming at these advantages and challenges, we propose DyTracker, an MOT method that builds on the DeepSORT baseline and improves three modules: (1) track dynamic management (TDM), which employs a dynamic tracks management algorithm to overcome the influence of false detections and maintain tracks number stability; (2) Kalman filter dynamic update (KFDU), which evaluates the degree of occlusion per athlete and implements KF dynamic update, which improves the robustness of KF against occlusion; and (3) feature dynamic selection and updating (FDSU), which analyzes the deficiencies of traditional association methods for highly similar appearance and detection noises issues and proposes a dynamic matching and updating strategy based on the difference in posture and detection quality.

In summary, the main contributions of this paper are as follows:

(1) We constructed an MOT dataset SKT-MOT for speed skating to compensate for the lack of data.
(2) We analyzed the unique advantages and challenges of the speed skating scene and proposed a dynamic MOT method—DyTracker.
(3) We carried out adequate experiments on SKT-MOT and MOT17 dataset [16] to verify the effectiveness and generalization of the proposed method and modules.

The paper is organized as follows: Section 1: introduction, Section 2: related work, Section 3: SKT-MOT dataset, Section 4: DyTracker, and Section 5: experiment and discussion, followed by conclusions. In the appendix, we list the specific meanings of the abbreviations in the article.

2. Related Work

2.1. MOT Datasets. In various scenarios, numerous datasets for MOT have emerged, as shown in Table 1. Alongside the dataset we have proposed, several existing datasets also concentrate on human tracking. For example, PETS2009 [18] and TUD [19] are early pedestrian tracking datasets, albeit with relatively small scales. To form larger-scale datasets, MOT15 [15] integrates these early pedestrian datasets. MOT17 [16] further enriches pedestrian tracking by expanding to new scenes and dynamic perspective. MOT20 [17] increases the difficulty of tracking by increasing pedestrian density. In recent years, human tracking datasets have emerged in complex scenarios, such as DanceTrack [35] in dance scenes and SoccerNet-Tracking [12] in soccer scenes, which significantly contribute to the advancement of MOT in human tracking.

In addition to human tracking, other datasets have been proposed for various object types. In the field of autonomous driving, there exists a dataset called KITTI [36] that specifically focuses on vehicle tracking, representing the earliest large-scale MOT dataset in this domain. Additionally, BDD100K [37] and KITTI360 [38] further expand vehicle tracking data. CTMC [39] is dedicated to tracking biological cells, while TAO [40] focuses on multicategory tracking, annotating 833 target categories, significantly enriching the content of MOT.

2.2. MOT Methods. Most MOT methods can be categorized into TBD and JDT, as shown in Table 2. TBD [3, 6, 23, 24, 41] involves three independent components:

(1) An existing object detector to generate detection boxes for each frame.
(2) A re-ID embedding model used to extract the appearance features of objects.
(3) A tracker to associate objects based on their motion cues or appearance features.

TBD is a flexible framework, with each component able to be replaced, giving it high generalization and suitability for complex scenes. However, it has the drawback of being time-consuming during inference.

Instead, JDT [7–11, 21, 22] incorporates several components into a unified network, reducing the inference time. Typically, JDT builds on detectors, fusing a tracker for them or adding a feature extraction branch. Over the past period, this paradigm has become mainstream. However, due to contradictions between modules, achieving global optimality for the JDT paradigm is problematic.

In recent years, significant advancements have been made in both paradigms. DeepSORT [6] represents the classic method within the TBD paradigm, leveraging motion and appearance as the two primary target features to accomplish the tracking task through Hungarian matching. StrongSORT [41] enhances DeepSORT by incorporating more powerful components. ByteTrack [23], in pursuit of faster speed, utilizes only motion cues for data association. In addition, it incorporates low-scoring detection frames into the association process, significantly reducing missed detections. On this basis, OC-SORT [25] corrects the accumulation of errors in Kalman filtering and introduces the directional consistency metric, which effectively improves robustness to occlusion; BoT-SORT [24] introduces camera motion compensation while adjusting the state parameters of the KF.

In the JDT method, JDE [9]/FairMOT [10] integrates a feature extraction branch into the original detector, unifying the detector and feature extraction models. On the other hand, CTracker [44] proposes a chain tracking framework

Table 1: Overview of MOT dataset.

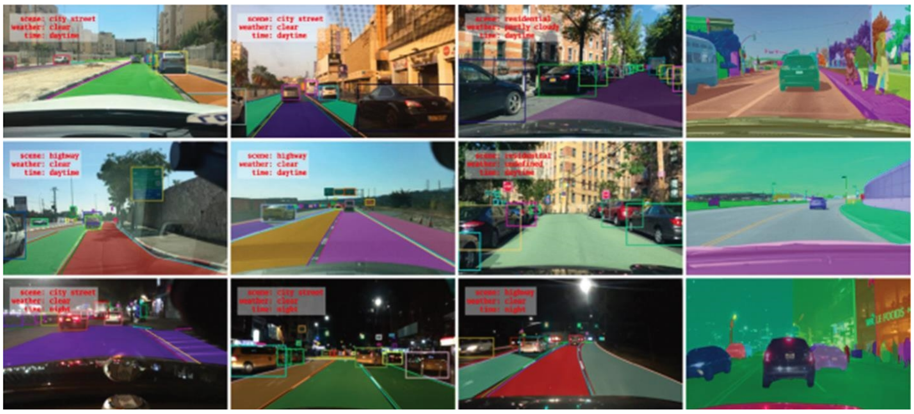
Type	Dataset	Frame rate	Scene graph
Human tracking	MOT17 MOT20 DanceTrack SoccerNet-Tracking	30 25 20 25	MOT17-03 MOT17-08 MOT17-07 MOT17-14
Vehicle tracking	KITTI KITTI360 BDD100	10 10 30	
Others	CMTC TAO	7.5 1	

Table 2: Overview of MOT method.

Paradigm types	Mainstream methods	Characteristic
TBD	SORT, DeepSORT, StrongSORT, ByteTrack, OC-SORT, BoT-SORT	Each of its components is independent of each other and can be replaced, which gives it great flexibility and generalizability, but also leads to time-consuming.
JDT	SST, JDE, FairMOT, CTracker, SCT, CenterTrack, TraDeS, MOTR	It allows several components to be integrated into a unified network, which leads to faster speeds, but makes it difficult to achieve global optimality due to conflicts between components.

based on two frame input, transforming the data association problem into a pairwise bounding boxes regression problem. SCT [45] chains them together using IoU, KF, and binary matching and introduces attention to better extract features. Centertrack [21] follows this two-frame input framework and borrows the idea of using points in CenterNet [42] to represent objects. It directly predicts the offset of the target between frames to achieve the association of data. TraDeS [8] constructs a global similarity matrix to predict this offset while simultaneously correcting the detection and segmentation results of the current target. The transformer-based MOTR [11] approach introduces a novel concept called track query. Each track query models the complete track of a target, enabling its transfer and update from frame-to-frame, thereby achieving end-to-end tracking. Recently, Unicorn [43] and OmniTracker [46] present a unified framework that uses a single network to simultaneously address four tracking tasks: single object tracking (SOT), MOT, video object segmentation (VOS), and multiobject tracking and segmentation (MOTS).

2.3. Motivation. Section 2.1 presents several human tracking datasets; however, most of them [15–19] predominantly focus on urban street scenes and indoor environments, while sports scenes are relatively scarce. Moreover, these datasets have certain limitations:

(1) They exhibit simple motion patterns, primarily slow and linear motion.

(2) The objects in these datasets have significant appearance differences, making them easily distinguishable.

These limitations have somewhat hindered the development of MOT. To address this gap, we proposed SKT-MOT, which provides new data for sports scenes, breaks through existing limitations, and poses new challenges to MOT.

In Section 2.2, we discuss different method types and mainstream approaches. Considering the challenges associated with optimizing the JDT paradigm and the lack of speed skating data, we followed the TBD paradigm. This paradigm allows us to independently train the detector and utilize additional detection data, making it easier to achieve higher accuracy.

The current mainstream framework of TBD is to complete the association using two major cues: motion and appearance. Some methods use only motion cues for tracking to pursue speed, but the smaller number of individuals in the speed skating scene meant less inference time, so we chose DeepSORT [6], which utilizes both cues, as the baseline. DeepSORT is, in fact, not a novel approach. However, it can still perform well when equipped with a powerful detector and an appropriate correlation strategy, as verified by this paper and StrongSORT [41].

However, the accuracy is generally not high when applying mainstream methods [6, 7, 9, 10, 23] such as DeepSORT to speed skating scenes. This could be attributed to the fact that existing methods are constrained by the dataset limitations and struggle to handle frequent occlusions, motion blurring, and clothing proximity between skaters. In addition, we investigated existing MOT methods for speed skating scenes but only found one work, LocalSort [47], which designs a local matching measurement method for occlusion problems, but it doesn't take into account similarities in appearance and motion blurring. To this end, we have performed a comprehensive analysis of the impact of these challenges and designed an efficient dynamic tracker to enhance tracking performance in speed skating scenes.

3. SKT-MOT

3.1. Dataset Construction. SKT-MOT dataset collected 56 short-track speed skater daily training videos with a frame rate of 30 fps and a resolution of 1,920 × 1,080. Thirty-six videos were selected as the training set, 10 as the validation, and 10 as the test. The videos were taken from two speed skating scenes at Beijing Capital Indoor Stadium and Ice and Snow Sports Base of Beijing Sport University, 44,402 and 8,776 images were labeled, respectively, by LabelMe [26] in the two scenes. Labeling information included the athlete's identification and the bounding box. For fully occluded objects, keep the ID consistent before and after occlusion. The basic information of the SKT-MOT is shown in Table 3.

3.2. Dataset Analysis. We quantitatively analyzed the clothing similarity between athletes, as shown in Figure 2. The results indicate a high degree of similarity between athletes' appearance color. In terms of the motion pattern, we analyze the trajectory and speed changes, as shown in Figure 3. The speed skating trajectory showed a unique insole shape, which differs significantly from the general pedestrian trajectory.

Table 3: Comparison of SKT-MOT with other multihuman tracking datasets.

Data type	MOT17 [16]	MOT20 [17]	SKT-MOT
Videos	14	8	56
Average tracks	96	432	6
Total tracks	1,342	3,456	336
FPS	30	25	30
Total images	11,235	13,410	53,178

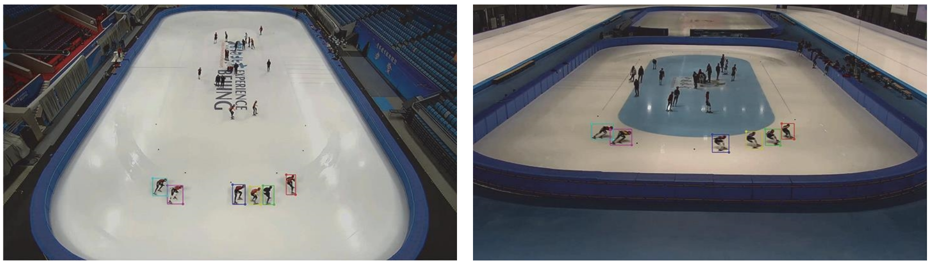


Figure 1: Overview of SKT-MOT scenes.

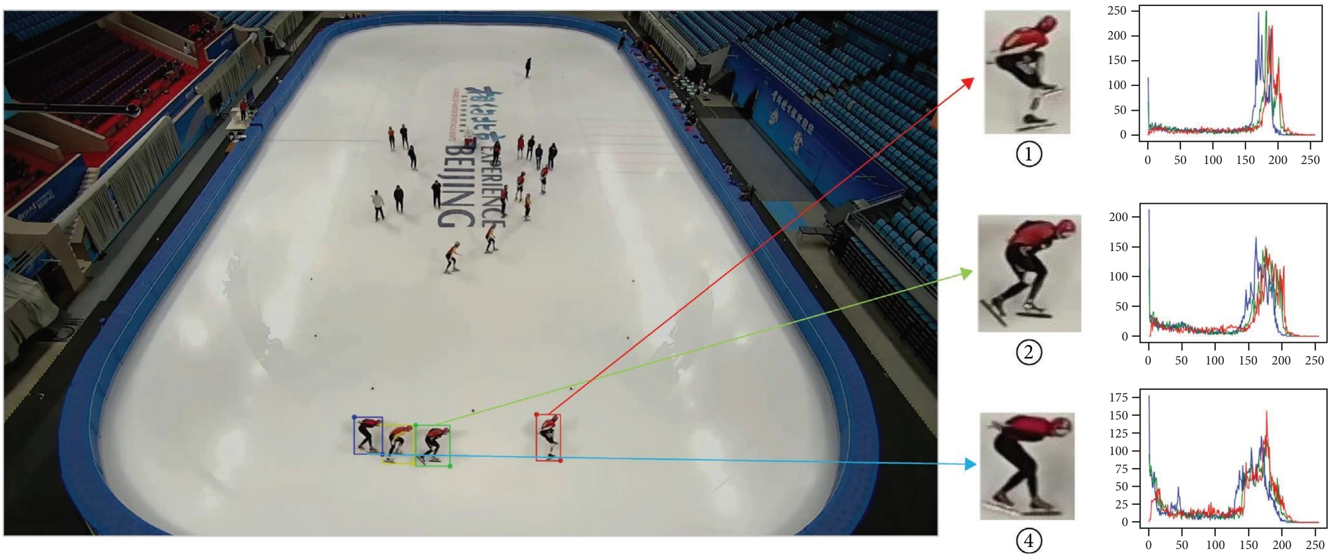


Figure 2: Quantitative analysis of the similarity of athletes' clothing. We cut out athletes with similar clothing and compared them with the help of color histograms. Histograms of athletes 1, 2, and 4 show similarities, with athletes 2 and 4 basically the same.

The speed changes exhibit ups and downs, and the average speed is high at 10 m/s, making it prone to motion blur. Moreover, due to the intense competition in short-track speed skating, athletes frequently exchange positions, resulting in frequent occlusion occurring in a single view. These unique issues pose new challenges for MOT.

4. DyTracker

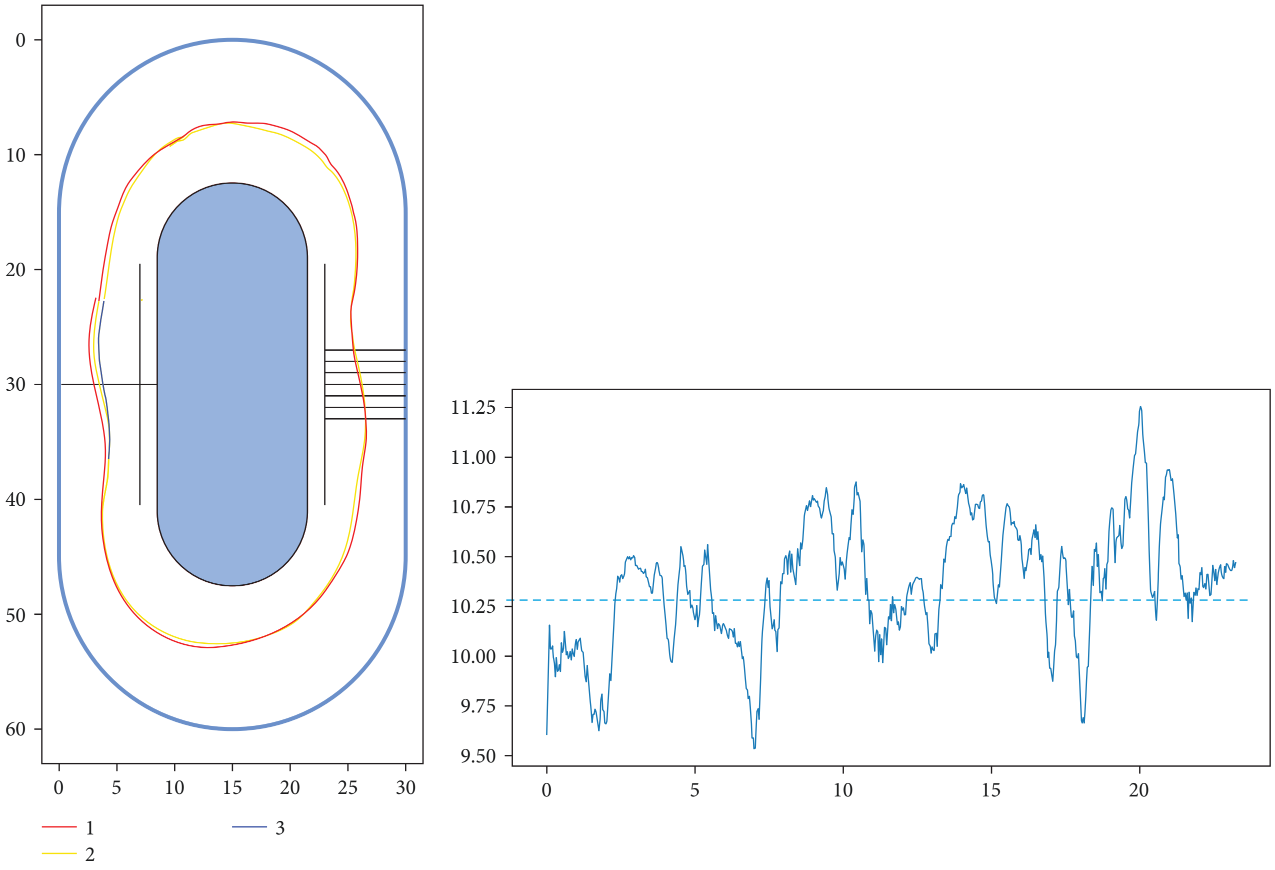
DyTracker (Dynamic Tracker) is an efficient MOT method for speed skating scenes, and Figure 4 illustrates our DyTracker built upon the TBD paradigm. It improved DeepSORT [6] with TDM, KFDU, and FDSU modules.

4.1. Preview DeepSORT. The DeepSORT algorithm is a two-branch framework consisting of a motion branch and a feature branch, where the detection results are fed into both branches frame-by-frame to complete the matching and updating process.

4.1.1. Matching. In the motion branch, the KF [4] predicts the state of the trajectory (box position and scale, etc.) in the current frame. The correlation between the predicted state of trajectory and the newly input detection information is computed using the Mahalanobis distance [28].

$$d^{(1)}(i,j) = \left(m_j - p_i\right)^T S^{-1} \left(m_j - p_i\right), \qquad (1)$$

where m_j is the newly input jth detection information, p_i is the predicted state of the ith trajectory, and S is a covariance matrix.



(a)

(b)

Figure 3: Analysis of motion pattern. With the help of perspective transformation [27], we mapped the GT results, converting from a boundary line view to a plan view. (a) Shows the final mapping result, which can help to observe the shape of the trajectory. At the same time, we calculated the athlete's speed. (b) Shows the athlete's speed change, where the average speed is indicated by a dotted line.

In the feature branch, a reidentification module is used to extract the appearance feature of the newly input detection. Furthermore, it uses a feature gallery to store the latest 100 frame features for each trajectory and integrates them as the trajectory feature in frame k. Then, the feature similarity is measured by the minimum cosine distance.

$$d^{(2)}(i,j) = \min\left\{1 - f_j^T f_i^{(k)} \mid f_i^{(k)} \in G_i\right\}, \tag{2}$$

where f_j is the feature of the newly entered jth detection, $f_i^{(k)}$ is the ith trajectory feature in frame k, and G_i is the feature gallery of the ith trajectory.

The two distances mentioned above are used together to construct a similarity matrix $c_{i,j}$. On the basis of Hungarian matching [5], a cascade matching strategy is proposed for a two-round matching process. The first round depends on the similarity matrix, and the second round uses a simple IoU.

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j). \tag{3}$$

4.1.2. Updating. After the matching is completed, in the motion branch, the KF performs a state update, fuses the detection and prediction values to generate the final correction result, and updates the relevant parameters; in the feature branch, newly matched object features are inserted into the feature gallery to complete the feature update.

4.2. TDM. TDM focusses on the characteristics of speed skating. Once speed skating begins, athletes rarely disappear from the video and join halfway through, resulting in a relatively fixed number of tracks in a video. However, false detections can easily disrupt the stability of the number of tracks, as shown in Figure 5. Based on this, we designed a dynamic management module for the number of tracks, as shown in Algorithm 1, to maintain the stability of the number of tracks and improve the robustness to false detection.

4.3. KFDU. In the motion branch, the KF [4] operation requires inputting detections' position and scale information. However, frequent occlusion will cause this information to be inaccurate, which also affects the accuracy of the KF. KFDU aims at the problem, proposes a metric to evaluate the degree of occlusion, and performs the dynamic update of the KF according to the metric, improving the robustness to occlusion.@

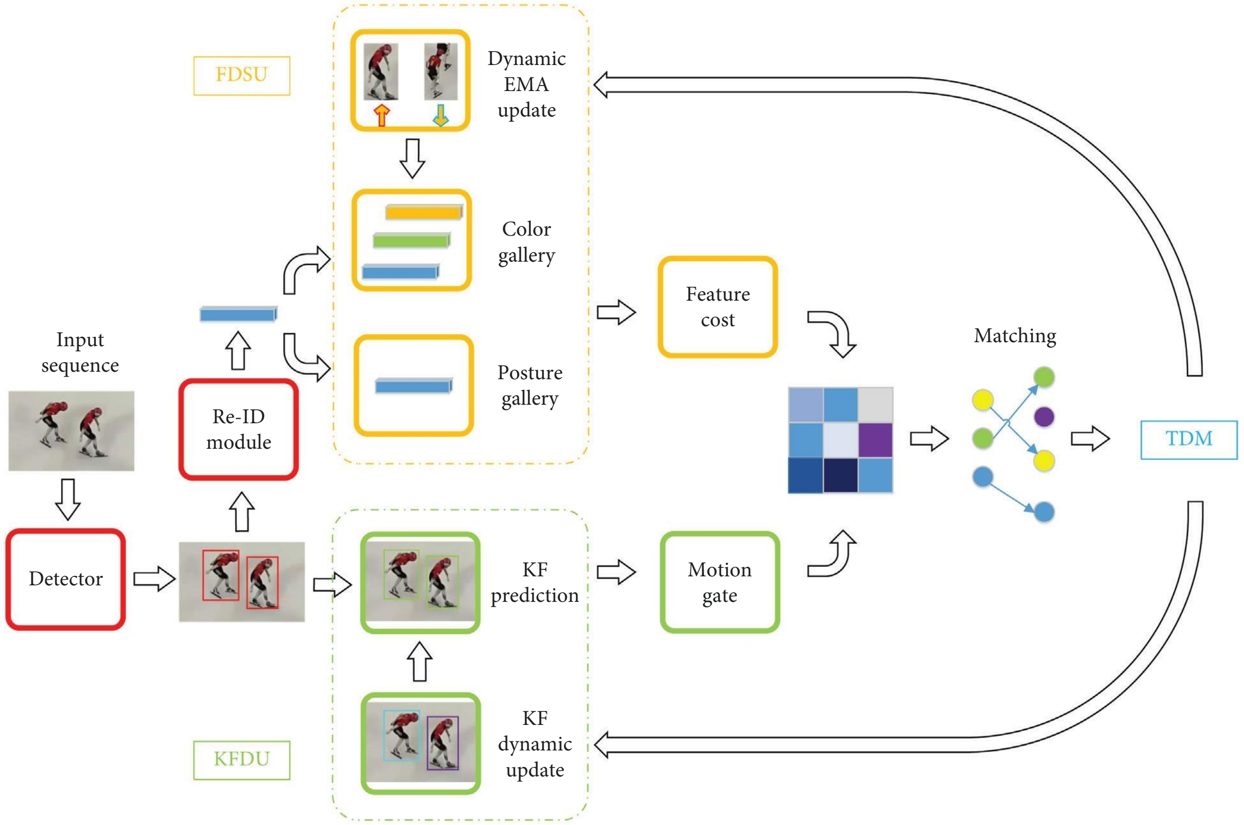


Figure 4: Overview of our proposed DyTracker pipeline.

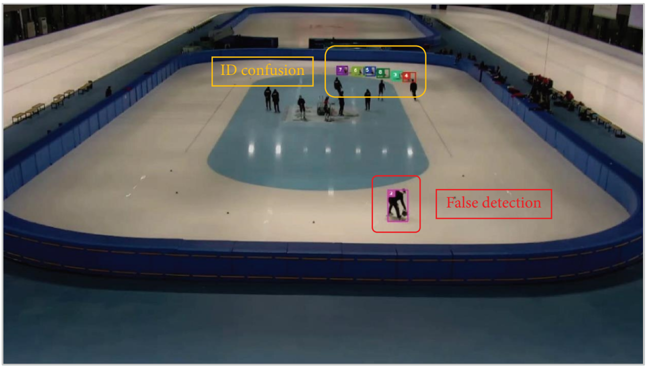


Figure 5: False detection during speed skating. In this scene, only skaters need to be tracked, while coaches, bystanders, and other unrelated people can easily produce false detection (marked by a red box). When misdetection occurs, the number of tracks increases incorrectly, confusing the identification of the athlete (marked by an orange box).

4.3.1. Evaluate Occlusion. The evaluation of an athlete's occlusion is often based on the detection confidence, but this criterion is not specific enough. Confidence is jointly determined by object classification and location accuracy. Since the KF relies on location information as input and motion blur interference, location information should be considered more. Figure 6 shows how motion blur affects detection confidence, making occlusion assessment less reliable. For these issues, we proposed an adjustment factor σ that calculates the IoU and the distance between the central point (similar to the DioU) between the current detection box and other boxes. The maximum σ to adjust the detection confidence to obtain the final occlusion metric o_k, weakens the effect of motion blur and enriches location information.

$$\sigma = \max\left\{ \mathrm{IoU}(i, j) - \frac{d(i, j)^2}{l(i, j)^2} \right\}, \tag{4}$$

$$o_k = c_k - 0.5\sigma, \tag{5}$$

where IoU is the degree of overlap between two boxes, i represents the current detection box, j represents other boxes, d represents the distance between the central points of two boxes, l represents the diagonal distance of the smallest external rectangle, and c_k is the detection confidence.

In this paper, no further distinction is made between the occluder and the occluded, both of whom should receive less trust compared to athletes without occlusion. In addition, detection confidence can distinguish them to some extent.

4.3.2. Kalman Filter Dynamic Update. The KFDU retained the KF state prediction step and improved the state update step. The KF process is shown in Figure 7. In the update step, the observation noise covariance R_* reflects the observation uncertainty, a smaller observation noise means that this observation is more trustworthy. However, in the KF

```
Input: Video length $N$;
Number of tracks $T$;
$T.value \leftarrow$ Initial frame tracks number;
$T.state \leftarrow$ instability;
for frame $k \leftarrow 2$ to $N$ do
    if $T.state$ is instability then
        if presence of $n$ detections not matched to any track and their confidence are enough high then
            Generate $n$ new tracks;
            $T.value \leftarrow T.value + n$;
        end
        else if presence of $n$ tracks not matched to any detection for fifteen consecutive frames then
            Delete these $n$ tracks;
            $T.value \leftarrow T.value - n$;
        else if $T.value$ no change for fifteen consecutive frames then
            $T.state \leftarrow$ stability;
    end
    else if $T.state$ is stability then
        if presence of $n$ detections not matched to any track for fifteen consecutive frames then
            Generate $n$ new tracks;
            $T.value \leftarrow T.value + n$;
        end
        else if presence of $n$ tracks not matched to any detection for fifteen consecutive frames then
            Delete these $n$ tracks;
            $T.value \leftarrow T.value - n$;
        else
            $T$ remain unchanged;
end
```

Algorithm 1: TDM.

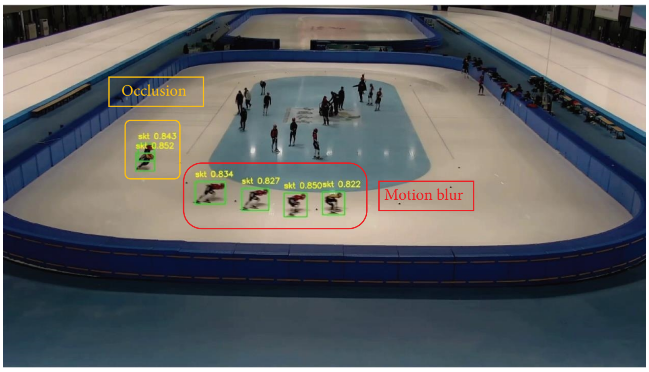


Figure 6: Effect of occlusion and motion blur on detection confidence. The red box marks motion blur, and the orange box marks occlusion. Motion blur produces lower confidence, but its position information is accurate.

algorithm, R_* is a constant matrix, which gives the same trust to observations of different qualities but it should be dynamic. In other words, when the object is heavily occluded, we should weaken the observation and give more trust to the prediction. In comparison, for high-quality observations, we should give more trust. KFDU is shown in Algorithm 2. Specifically, occlusion metric o_k is used to measure the observation quality and achieve dynamic adjustment of the measurement noise covariance R_k. This gives the KF a dynamic trust for different observations.

4.4. FDSU. The appearance branch mainly includes feature similarity matching and feature updating. In the matching step, we considered that the athletes in the speed skating scene are dressed similarly, but they have differences in their postures. Therefore, we proposed a similarity-matching method that dynamically selects these two features (FDS). In the update step, occlusion and motion blurring produce low-quality detections and pollute the feature gallery, for which we proposed a dynamic feature update strategy (FDU).

4.4.1. Feature Dynamic Selection (FDS). The existing human tracking datasets [16–19] have large differences in clothing but similar postures, which results in traditional matching methods ignoring posture information and relying solely on differences in appearance color, using historical features for the association. However, in the speed skating scene, it's just the opposite. This means that traditional matching methods do not apply. To address this, we considered differences and instantaneous invariance of posture and argued that matching using only adjacent frame features can also be efficient for the tracking. Figure 8 illustrates this point.

Specifically, we reduced the weighting of historical features and took more consideration of proximity features to increase the weighting of posture features. Additionally, based on the existing gallery (color gallery), FDS added a posture gallery that only stores adjacent frames' features. For athletes with clear postures, we select a posture gallery for similarity matching. Otherwise, we use the color gallery. It judges whether the athlete's posture is clear based on the occlusion metric o_*, as shown in Section 4.3.1. If o_* exceeds the threshold of 0.9, it is considered clear; otherwise, it is considered blurry. With this strategy, athletes can dynamically select the appropriate feature gallery for matching similarity.

4.4.2. Feature Dynamic Update (FDU). In the feature update, DeepSORT [6] builts a gallery of features for each trajectory and inserted new features into it to achieve the update, which results in a significant waste of spatial and temporal resources. JDE/FairMOT [9, 10] improved this approach by using an exponential moving average (EMA) feature update strategy in which only one feature state is maintained per trajectory, which is a resource saver and the current dominant feature update solution. However, this approach has flaws. Problems such as occlusion and motion blur cause an increase in detection noise, resulting in differences in the quality of detection. The EMA strategy treats detections of different qualities equally. However, this process should be dynamic. High-quality features should be retained with

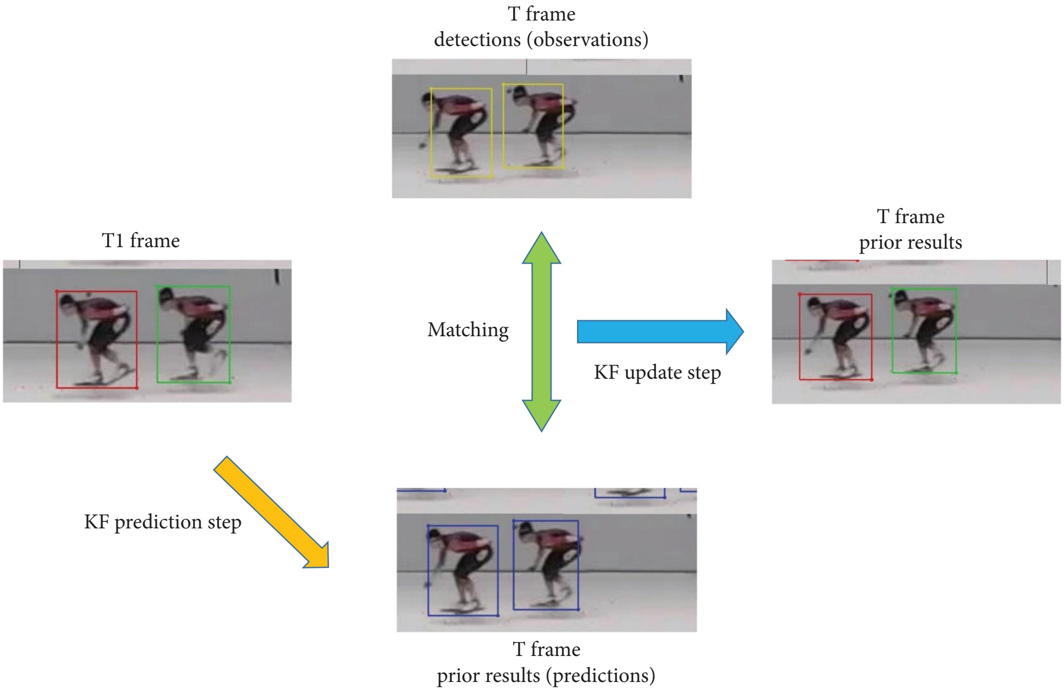


Figure 7: The Kalman filtering process. The KF contains two parts: state prediction and state update. The prediction step uses a constant-velocity model to predict the prior (prediction) state in frame T based on the posterior result in frame T1. The update step fuses new observation with the matched prediction depending on the Kalman gain K calculated, resulting in the posterior state and covariance matrix at frame T.

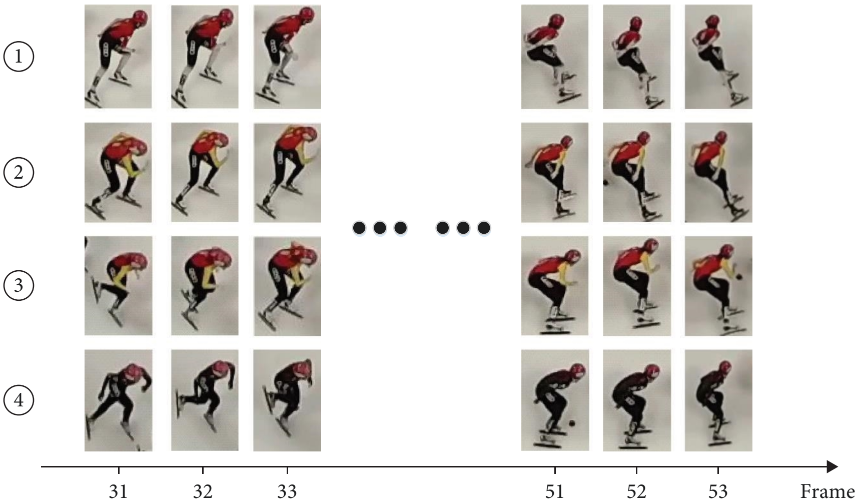


Figure 8: Comparison of athlete posture changes. The horizontal axis is the time, the vertical axis is the athlete's ID, and the four athletes are taken from the same video. Observing the figure can be obtained: in the speed skating scene (1) due to differences in the position and habits of athletes, there are apparent differences in posture among them at the same time; (2) the posture shows instantaneous invariance (i.e., high similarity of the same athlete between adjacent frames) due to motion inertia; (3) due to the clothing worn by athletes being similar, appearance color may not be as reliable.

greater weight, while low-quality detection should be ignored. Specifically, we introduce detection confidence to reflect detection quality and dynamically adjust the momentum term α_k, which achieves dynamic updating of the EMA.

$$\alpha_k = c_k \alpha, \qquad (6)$$

$$e_k^t = \alpha_k f_k^t + (1 - \alpha_k) e_k^{t-1}, \qquad (7)$$

where c_k is the detection confidence, e_k^t represents the feature state of the trajectory k in frame t, f_k^t is the appearance embedding of the new detection, and α is the original static momentum term.

```
Input: Observation $Z_k$
Observation noise covariance $R_k$
Measurement occlusion degree $o_k$
Predicted state $X_{k|k-1}$
Predicted state covariance $P_{k|k-1}$
The observation model $H$
Output: Updated state $X_k$
Updated state covariance $P_k$
Step:
1 $\tilde{R}_k = (1.3 - o_k)R_k$
  //Updating dynamically observation noise covariance
2 $K_k = P_{k|k-1}H^T\left(HP_{k|k-1}H^T + \tilde{R}_k\right)^{-1}$
  //Calculating corrected Kalman gain
3 $X_k = X_{k|k-1} + K_k(Z_k - HX_{k|k-1})$
  //Based on K, fusing observation and predicted state
4 $P_k = (1 - KH)P_{k|k-1}$
  //Updating state covariance
```

Algorithm 2: KFDU (state update step at sate k).

4.5. Complexity Analysis. TDM algorithm controls the variation of trajectory number to effectively address the issues of false detection and missed detection, maintaining the purity of trajectory library. In terms of time complexity, assuming a video length of N frames, the algorithm performs one or two judgments for each frame, which is a single loop problem, so the time complexity is $O(N)$. In terms of space complexity, this algorithm only needs to store two one-dimensional variables, the total number of trajectories and its state, as well as two timers, and dynamical updating without storing overwritten data, so its space complexity is $O(1)$.

The FDSU largely inherits the original KF, with only two-step operation in the update step, so its time complexity is consistent with the KF's time complexity. Assuming a video length of N frames and one iteration per frame, because matrix operations are required, its time complexity is O(N^2). In terms of space complexity, the state vector and error covariance matrix of each moment need to be stored, and these vectors and matrices are squared with the number of observation data T, so the space complexity also is O(T^2).

DyTracker, like DeepSORT, is difficult to analyze specifically due to the overall complexity affected by multiple modules. Therefore, we mainly compared DyTracker with DeepSORT here. TDM and KFDU have been explained in detail in the previous text. For the FDSU module, in terms of time complexity, it mainly adds one judgment and two numerical operations, so the added time can be ignored; in terms of space, we additionally add a storage library for posture information, but it only stores information from adjacent frames, so the added space cost is not significant. Overall, compared with DeepSORT, DyTracker does not add too much time and space consumption, but the efficiency gain is significant.

4.6. Datasets and Metrics

4.6.1. Datasets. We conduct experiments on the SKT-MOT and MOT17 datasets [16]. SKT-MOT is a dataset of the short-track speed skating proposed in this article, and specific details are given in Section 3. For the detection and re-ID module, we transformed, respectively, the data format imitating COCO [32] and MARS [33] datasets, dividing the dataset according to the 7 : 2 : 1. MOT17 is a popular dataset for MOT, which consists of seven sequences, 5,316 frames for training, and seven sequences, 5,919 frames for testing. For ablation studies, we take the first half of each sequence in the MOT17 training set for training and the last half for validation following.

4.6.2. Metrics. The evaluation of tracking performance is mainly based on multiobject tracking accuracy (MOTA), identification F1 (IDF1) score, and multiobject tracking precision (MOTP).

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FN} + \mathrm{FP} + \mathrm{IDSW}}{\mathrm{GT}}. \tag{8}$$

MOTA is an evaluation metric for MOT algorithms that focus on tracking accuracy. It is calculated on the basis of false positive (FP), false negative (FN), and identification switch (IDSW), where FP represents false detection, FN represents missing detection, and IDSW counts the number of identity switches of an object. Despite its limitations and criticisms, it is still the most widely accepted evaluation metric for MOT.

$$\mathrm{IDF1} = \frac{2 \cdot \mathrm{IDTP}}{2 \cdot \mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}}. \tag{9}$$

IDF1 is another important metric in MOT to evaluate the precision of object identification. It responds more to the accuracy of ID matching. Here, identification true positive (IDTP) stands for the correctly identified object, identification false positive (IDFP) stands for the incorrectly identified object, and identification false negative (IDFN) stands for the unidentified identity information. MOTP, which measures the overlap between the resulting bounding box and ground truth, describes the localization precision of the object.

5. Experiments and Discussion

5.1. Experimental Details. For detection, the detector is YOLOv5-x [29] pretrained on the COCO dataset, introduces Diou-NMS, changes the localization loss to CioU-LOSS, and uses the original training schedule. For the embedding of the re-ID feature, the re-ID module [34] of DeepSORT is used, and the initial learning rate is 0.1, using Adam Optimizer [30]. For DyTracker, a threshold of 0.65 is set for nonmaximum suppression (NMS) and a threshold of 0.7 for detection confidence. The minimum feature distance threshold is 0.2, and the momentum term α in the color gallery and the posture gallery is 0.65 and 1, respectively. The weight factor for the appearance cost λ is 0.98.

Table 4: Comparison with state-of-the-art MOT methods on the SKT-MOT dataset.

Method	Ref.	MOTA↑	IDF1↑	MOTP↑	FP↓	FN↓	IDSW↓	FPS↑
SST [7]	TPAMI2019	76.02	56.48	85.87	3,448	3,991	1427	4.2
JDE [9]	ECCV2020	79.87	72.91	78.31	2,983	4,943	812	13.2
LocalSORT [47]	JSS2021	83.14	76.82	82.43	2,365	4,352	624	11.8
FairMOT [10]	IJCV2021	86.53	76.51	85.71	1,359	4,308	557	16.4
ByteTrack [23]	ECCV2022	84.56	73.72	87.32	3,565	**3,219**	397	**17.8**
DeepSORT [6]	ICIP2017	81.78	77.61	81.24	2,955	4,363	735	9.3
DyTracker	This study	**93.70**	**92.39**	**87.79**	**939**	3,670	**154**	8.9

TBD methods use the same detection results. The best results are in bold.

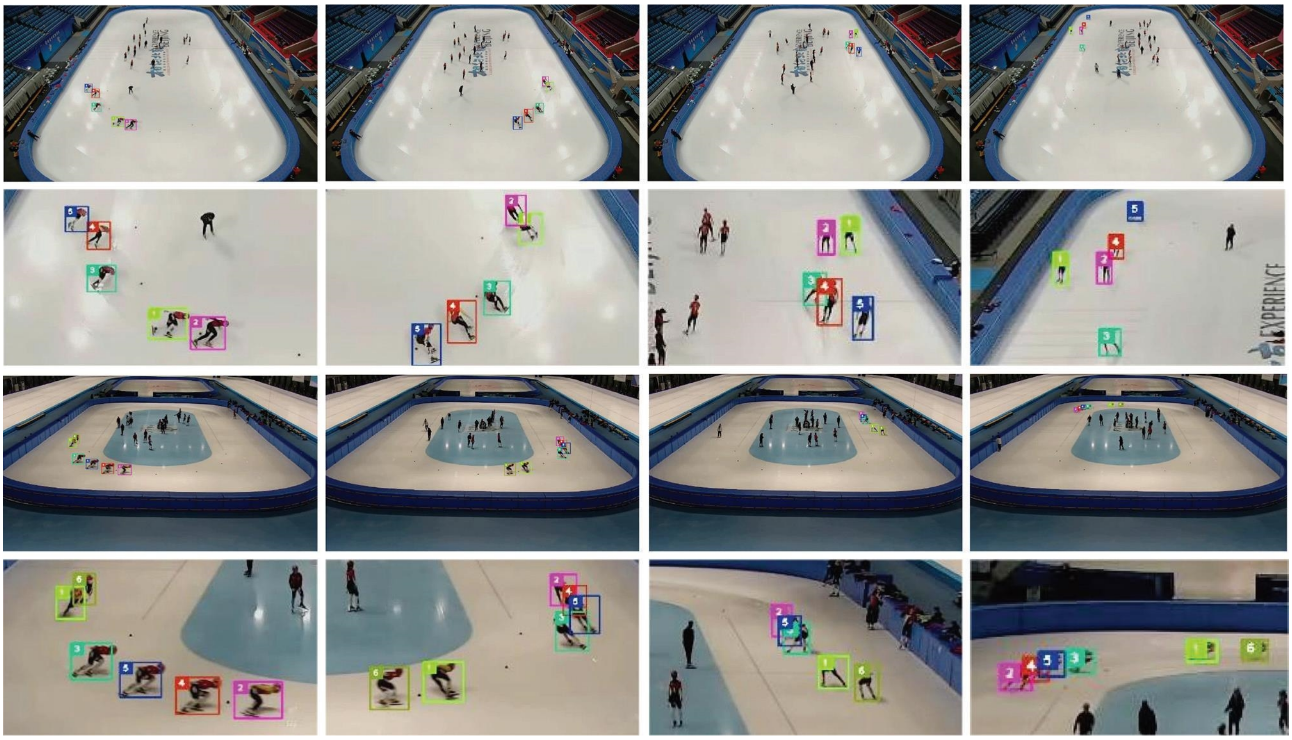
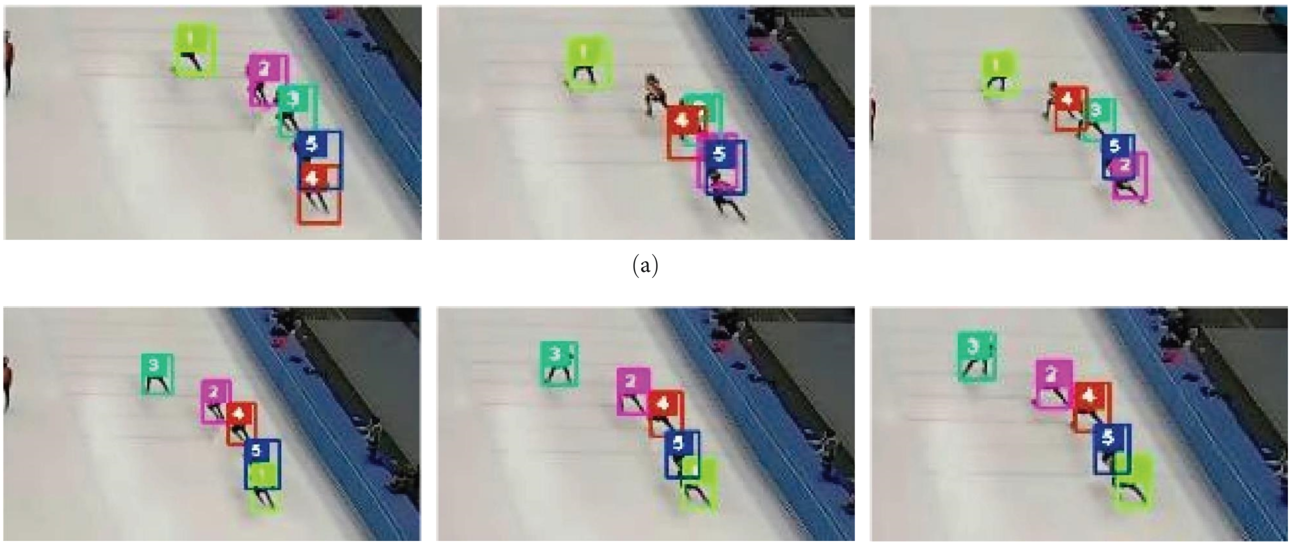


Figure 9: Visualization results of DyTracker on the SKT-MOT.

All experiments are conducted on a server machine with two 12 GB 2080Ti.

5.2. Comparative Experiment. We compared our DyTracker with state-of-the-art methods on the SKT-MOT dataset, and Table 4 lists the detailed performance results. The experimental results show that our DyTracker is much superior to other methods in MOTA, with a highest of 93.70% in all methods. Compared with JDT methods [7, 9, 10], we also have significant advantages. Compared to similar TBD methods [6, 23], we use the same detector and have achieved certain improvements. Second, the performance of DyTracker is also best in MOTP and IDF1, which confirms that our tracker can achieve more accurate object localization, more efficient completion of association tasks, and better tracking of speed skaters. On the other hand, our method compares favorably with LocalSORT, which is also designed for speed skating scenarios, demonstrating significant advantages. Limited by the two-stage framework, the FPS performance of our method is mediocre. Figure 9 shows the visualization of the DyTracker tracking results on the SKT-MOT.

Compared to our baseline DeepSORT, DyTracker improves 11.92% and 14.78% in MOTA and IDF1, respectively, and the rest of metrics are also significantly improved. As can be seen from the FPS, these performance increases result in only minimal time consumption. Figure 10 compares the visualization effects of DeepSORT and DyTracker. It is clear that when occlusion occurs, the position of object boxes in DeepSORT will have a significant deviation from skaters. Moreover, due to similar appearance and other problems, ID matching errors and IDSW also occur frequently. In contrast, the tracks in DyTracker are more precise and stable, which further demonstrate that our proposed method performs better in complex situations such as occlusion and similar dress.



(a)

(b)

Figure 10: Comparison of visualization between DeepSORT and DyTracker: (a) represents the DeepSORT tracking results and (b) is the DyTracker.

Table 5: Ablation study on the SKT-MOT dataset for different modules, that is, track dynamic management (TDM), dynamic selection and update of features (FDSU), dynamic Kalman filter update (KFDU), and evaluation occlusion (Occ).

Method	Conf.	Occ.	MOTA↑	IDF1↑	MOTP↑	FP↓	FN↓	IDSW↓
Baseline			81.78	77.61	82.12	2,955	4,363	735
+TDM			89.38	87.86	82.63	1,878	4,298	577
+TDM + FDSU	√		92.33	91.97	85.51	1,079	3,820	160
+TDM + FDSU		√	92.45	92.14	85.32	1,072	3,812	163
+TDM + FDSU + KFDU	√		93.33	92.32	87.27	959	3,705	146
+TDM + FDSU + KFDU		√	93.70	92.39	87.79	939	3,670	154

5.3. Ablation Study

5.3.1. Ablation Study for DyTracker. Table 5 summarizes the DeepSORT to DyTracker process:

(1) TDM: Overcoming the influence of false detection has significantly reduced FP, thereby improving MOTA. At the same time, the range of ID values is also controlled, reducing the occurrence of ID switches, making ID matching more accurate and improving IDF1.

(2) FDSU: Improved matching accuracy, reduced the influence of detection noise, and significantly increased IDF1, while also improving MOTA to a certain extent.

(3) KFDU: Improved object location accuracy, resulting in a significant increase in MOTP and improved the robustness of KF, leading to improvement in MOTA.

(4) Occlusion: Using the occlusion metric instead of the confidence to evaluate the degree of occlusion has reduced the impact of motion blur, leading to improvements in all metrics.

5.3.2. Extended Experiments for KFDU and FDU. To solve the occlusion problem, this article proposed two modules, KFDU and FDU. We argue that occlusion occurs to some extent in current datasets. These two modules should have universal adaptation. Table 6 shows that we have inserted the two update modules into the existing method, and the verification results on mot 17val, both MOTA and IDF1 have been improved.

5.3.3. Ablation Study for Threshold. We conducted an ablation experiment on the threshold to evaluate whether the pose is clear. As shown in Figure 11, the trend of MOTA and IDF1 is basically the same. We chose the 0.9 corresponding to the peak as the final threshold. When the threshold is small enough, that is, entirely relying on the posture gallery, which only correlates adjacent frames, can also achieve good results; when the threshold is large enough to rely entirely on the color gallery, which takes more into account historical features, no better than the former. This further suggests that proximity features should be more considered in speed

Table 6: Results of applying KFDU and FDU to various MOT methods.

Method	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDSW↓
DeepSORT + KFDU + FDU	76.80 (+0.1)	77.9 (+0.6)	56.7	11.8	3,165	9,910	231
FairMOT + KFDU + FDU	69.1	73.4 (+0.6)	41.3	15.6	2,011	14,480	283
ByteTrack + KFDU	76.6 (+0.1)	79.7 (+0.2)	59.6	11.8	3,431	9,680	243

The TBD methods use the YOLOX detector [31]. All experiments are performed on the MOT17 validation set.

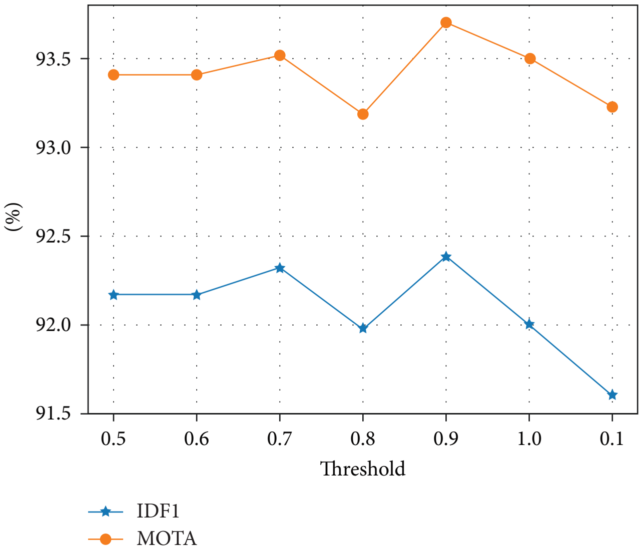


Figure 11: Threshold to evaluate whether the posture is clear.

skating scenes and appropriately reduce the weighting of historical features.

5.4. Limitations and Future Work

5.4.1. Limitations. The speed skating scene is different from general scenes. Taking the appearance similarity problem as an example, the appearance difference between targets is relatively large and their poses are close in general scenes, while the opposite is true in speed skating scenes (similar appearance, large pose differences). This explains why existing algorithms are not suitable for speed skating and also pose a challenge for the performance of our proposed modules in general scenes.

To better demonstrate the generalization of our modules, we conducted extensive ablation studies, as shown in Section 5.3, which show that the two proposed update modules are generally applicable. However, TDM algorithm is limited to scenarios with a relatively fixed number of trajectories, while FDS is limited to cases with large pose differences.

Additionally, due to the relatively small amount of speed skating data and the difficulty in optimizing end-to-end methods, we opted for a relatively basic two-stage framework. Although good accuracy is achieved, it ran slowly.

5.4.2. Future Work. Although FDS in this paper has certain limitations, we believe that the idea of dynamic selection has great potential for further research. For example, when objects are occluded, it is difficult to distinguish them by appearance alone, so motion direction and displacement should be considered more. In cases where the appearance difference between objects is large, appearance can be used as the main factor for matching. For scenarios with large pose differences, pose information can be taken into account. Based on different situations, different cues can be determined as the leading factors to achieve an adaptive matching process.

In addition, we believe that pose information can be greatly valuable in certain MOT scenarios, such as DanceTrack and skating. We will also continue to research in this direction, such as using pose information to guide feature extraction and achieving the unity of pose recognition and tracking tasks (sharing a common network). On the other hand, we will also collect more speed skating data for expansion and try more end-to-end methods to seek faster speeds and greater accuracy.

6. Conclusions

This study explores the potential development space of MOT from the perspective of short-track speed skating. First, we constructed a short-track speed skating MOT dataset and analyzed its unique challenges, revealing the limitations of existing datasets and the inadequacies of existing methods. Accordingly, we proposed a dynamic tracker specifically designed for speed skating scenarios, which improves three modules based on DeepSORT: the TDM module mainly addresses the issues of FP and missed detections, KFDU enhances the robustness of KF against occlusions, and FDSU considers the posture differences to address the clothing similarity problem and proposed a dynamic update strategy to mitigate the impacts of occlusions and motion blur. Compared to existing methods, our method achieved the highest MOTA of 93.7 and IDF1 of 92.39 in the SKT-MOT dataset. Furthermore, we conducted extensive ablation experiments to analyze the generalization and potential values of all modules. We believe that there are differences and similarities between speed skating scenarios and general scenarios, which provide new insights to solve existing MOT problems and have great research value.

Appendix

In this article, we used many abbreviations. To facilitate better understanding for readers, we provided specific explanations for these abbreviations, as shown in Table 7.

Table 7: Abbreviations and their full names.

Abbreviations	Full names
CTracker	Chained tracker
DeepSORT	Deep simple online and real-time tracking
DyTracker	Dynamic tracker
EMA	Exponential moving average
FDSU	Feature dynamic selection and ppdating
FDS	Feature dynamic selection
FDU	Feature dynamic updating
FP	False positive/false detection
FN	False negative/missed detection
IoU	Intersection over union
IDF1	Identification F1 score
IDSW	Identification switch
IDTP	Identification true positive
IDFP	Identification false positive
IDFN	Identification false negative
JDE	Joint detection and embeding
JDT	Joint detection and tracking
KF	Kalman filter
KFDU	Kalman filter dynamic update
TDM	Track dynamic management algorithm
MOT	Multiobject tracking
MOTA	Multiobject tracking accuracy
MOTP	Multiobject tracking precision
MOTS	Multiobject tracking and segmentation
NMS	Nonmaximum suppression
OC-SORT	Observation-centric sort
Occ	Occlusion
Re-ID	Reidentification
SORT	Simple online and real-time tracking
SST	Single shot-multibox tracker
SCT	Superchained tracker
SOT	Single object tracking
TBD	Tracking-by-detection
TraDeS	Track to detect and segment
VOS	Video object segmentation

Data Availability

The data used to support the findings of this study are available from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Key R&D Program (grant no. 2019YFF0301800), National Natural Science Foundation of China (grant no. 61379106), and the Shandong Provincial Natural Science Foundation (grant nos. ZR2013FM036, ZR2015FM011).

References

[1] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1815–1821, IEEE, Providence, RI, USA, 2012.

[2] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *CVPR 2011*, pp. 1345–1352, IEEE, Colorado Springs, CO, USA, 2011.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, IEEE, Phoenix, AZ, USA, 2016.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[5] B. Yaw and H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, Beijing, China, 2017.

[7] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 104–119, 2021.

[8] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: an online multi-object tracker," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual*, pp. 12352–12361, Computer Vision Foundation/IEEE, 2021.

[9] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Computer Vision–ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm, Eds., vol. 12356 of *Lecture Notes in Computer Science*, pp. 107–122, Springer, Cham, 2020.

[10] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: on the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, pp. 3069–3087, 2021.

[11] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: end-to-end multiple-object tracking with transformer," in *Computer Vision–ECCV 2022*, Proceedings, Part XXVII, pp. 659–675, Springer, Tel Aviv, Israel, 2022.

[12] A. Cioppa, S. Giancola, A. Deliège et al., "Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3491–3502, IEEE Computer Society, Los Alamitos, CA, USA, 2022.

[13] Y. Gong and G. Srivastava, "Multi-target trajectory tracking in multi-frame video images of basketball sports based on deep learning," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 10, no. 2, Article ID e9, 2023.

[14] B. T. Naik and M. F. Hashmi, "YOLOv3-Sort: detection and tracking player/ball in soccer sport," *Journal of Electronic Imaging*, vol. 32, Article ID 011003, 2023.

[15] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: towards a benchmark for multi-target tracking," arXiv preprint arXiv: 1504.01942, 2015.

[16] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: a benchmark for multi-object tracking," arXiv preprint arXiv: 1603.00831, 2016.

[17] P. Dendorfer, H. Rezatofighi, A. Milan et al., “MOT20: a benchmark for multi object tracking in crowded scenes,” arXiv preprint arXiv: 2003.09003, 2020.

[18] J. Ferryman and A. Shahrokni, “PETS2009: dataset and challenge,” in *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1–6, IEEE, Snowbird, UT, USA, 2009.

[19] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3D pose estimation and tracking by detection,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 623–630, IEEE, San Francisco, CA, USA, 2010.

[20] J. Wang, Y. Peng, X. Yang, T. Wang, and Y. Zhang, “Sportstrack: an innovative method for tracking athletes in sports scenes,” arXiv preprint arXiv: 2211.07173, 2022.

[21] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” European Conference on Computer Vision (ECCV), 2020.

[22] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 941–951, IEEE, 2019.

[23] Y. Zhang, P. Sun, Y. Jiang et al., “Bytetrack: multi-object tracking by associating every detection box,” in *Computer Vision–ECCV 2022*, vol. 13682 of *Lecture Notes in Computer Science*, pp. 1–21, Springer, Israel, 2022.

[24] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, “BoT-SORT: robust associations multi-pedestrian tracking,” arXiv preprint arXiv: 2206.14651, 2022.

[25] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, “Observation-centric sort: rethinking sort for robust multi-object tracking,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9686–9696, IEEE, Vancouver, BC, Canada, 2023.

[26] A. Torralba, B. C. Russell, and J. Yuen, “Labelme: online image annotation and applications,” *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1467–1484, 2010.

[27] R. M. Haralick, “Using perspective transformations in scene analysis,” *Computer Graphics and Image Processing*, vol. 13, no. 3, pp. 191–221, 1980.

[28] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, “The mahalanobis distance,” *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.

[29] G. Jocher, ““YOLOv5 by Ultralytics,” 2020, [Online]. Available: https://github.com/ultralytics/yolov5.

[30] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” arXiv preprint arXiv: 1412.6980, 2014.

[31] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: exceeding yolo series in 2021,” arXiv preprint arXiv: 2107.08430, 2021.

[32] T.-Y. Lin, M. Maire, S. Belongie et al., “Microsoft COCO: common objects in context,” in *Computer Vision–ECCV 2014*, vol. 8693 of *Lecture Notes in Computer Science*, pp. 740–755, Springer, Cham, 2014.

[33] L. Zheng, Z. Bie, Y. Sun et al., “MARS: a video benchmark for large-scale person re-identification,” in *Computer Vision–ECCV 2016*, vol. 9910 of *Lecture Notes in Computer Science*, pp. 868–884, Springer, Cham, 2016.

[34] Z. Pei, ““Deepsort pytorch,” 2019, [Online]. Available: https://github.com/ZQPei/deep_sort_pytorch.

[35] P. Sun, J. Cao, Y. Jiang et al., “Dancetrack: multi-object tracking in uniform appearance and diverse motion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20993–21002, IEEE, 2022.

[36] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, Providence, RI, USA, 2012.

[37] F. Yu, H. Chen, X. Wang et al., “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, IEEE, 2020.

[38] Y. Liao, J. Xie, and A. Geiger, “KITTI-360: a novel dataset and benchmarks for urban scene understanding in 2D and 3D,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2023.

[39] S. Anjum and D. Gurari, “CTMC: cell tracking with mitosis detection dataset challenge,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4228–4237, IEEE, Seattle, WA, USA, 2020.

[40] A. Dave, T. Khurana, P. Tokmakov, C. Schmid, and D. Ramanan, “TAO: a large-scale benchmark for tracking any object,” in *Computer Vision–ECCV 2020*, pp. 436–454, Springer, Glasgow, UK, 2020.

[41] Y. Du, Z. Zhao, Y. Song et al., “StrongSORT: make deepSORT great again,” *IEEE Transactions on Multimedia*, 2023.

[42] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” arXiv preprint arXiv: 1904.07850, 2019.

[43] B. Yan, Y. Jiang, P. Sun et al., “Towards grand unification of object tracking,” in *Computer Vision–ECCV 2022*, vol. 13681 of *Lecture Notes in Computer Science*, pp. 733–751, Springer, Cham, 2022.

[44] J. Peng, C. Wang, F. Wan et al., “Chained-tracker: chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” in *Computer Vision–ECCV 2020*, vol. 12349 of *Lecture Notes in Computer Science*, pp. 145–161, Springer, Cham, 2020.

[45] S. A. Qureshi, L. Hussain, Q. ul-ain-Chaudhary et al., “Kalman filtering and bipartite matching based super-chained tracker model for online multi object tracking in video sequences,” *Applied Sciences*, vol. 12, no. 19, Article ID 9538, 2022.

[46] J. Wang, D. Chen, Z. Wu et al., “OmniTracker: Unifying object tracking by tracking-with-detection,” arXiv preprint arXiv: 2303.12079, 2023.

[47] Q. Li, H. Mo, X. Wang, and H. Li, “Multiple object tracking and kinematic simulation for short track speed skating,” *Journal of System Simulation*, vol. 33, no. 5, pp. 1039–1050, 2021.