Virtual Reality & Intelligent Hardware

• Article •

October 2023 Vol. 5 No. 5: 422–438
10.1016/j.vrih.2023.04.001

Novel learning framework for optimal multi-object video trajectory tracking

Siyuan CHEN, Xiaowu HU, Wenying JIANG, Wen ZHOU*, Xintao DING

School of Computer and Information, Anhui Normal University, Anhui 241002*, China*

Received 28 November 2022; **Revised** 22 February 2023; **Accepted** 17 April 2023

Abstract: Background With the rapid development of Web3D, virtual reality, and digital twins, virtual trajectories and decision data considerably rely on the analysis and understanding of real video data, particularly in emergency evacuation scenarios. Correctly and effectively evacuating crowds in virtual emergency scenarios are becoming increasingly urgent. One good solution is to extract pedestrian trajectories from videos of emergency situations using a multi-target tracking algorithm and use them to define evacuation procedures. **Methods** To implement this solution, a trajectory extraction and optimization framework based on multi-target tracking is developed in this study. First, a multi-target tracking algorithm is used to extract and preprocess the trajectory data of the crowd in a video. Then, the trajectory is optimized by combining the trajectory point extraction algorithm and Savitzky–Golay smoothing filtering method. Finally, related experiments are conducted, and the results show that the proposed approach can effectively and accurately extract the trajectories of multiple target objects in real time. **Results** In addition, the proposed approach retains the real characteristics of the trajectories as much as possible while improving the trajectory smoothing index, which can provide data support for the analysis of pedestrian trajectory data and formulation of personnel evacuation schemes in emergency scenarios. **Conclusions** Further comparisons with methods used in related studies confirm the feasibility and superiority of the proposed framework.

Keywords: Web3D; Virtual evacuation; Multi-object tracking; Trajectory extraction; Trajectory optimization

Supported by the National Science Foundation of China (61902003; 61976006).

Citation: Siyuan CHEN, Xiaowu HU, Wenying JIANG, Wen ZHOU, Xintao DING. Novel learning framework for optimal multi-object video trajectory tracking. Virtual Reality & Intelligent Hardware, 2023, 5(5): 422–438

1 Introduction

Web3D, virtual reality (VR), and digital twins (DTs) have become important research topics, and the effective and reliable simulation of real scenarios is gradually becoming more important, particularly for Web3D scenarios. Video data can provide sufficient information to guide the conduct and decisions of a virtual avatar; thus, modeling the vivid and lifelike effects of a virtual scenario is convenient. Moreover, analyzing and understanding video data are becoming increasingly necessary, particularly in emergency evacuation scenarios. In emergency scenarios, such as fire evacuations, implementing measures using Web3D and VR

*Corresponding author, w.zhou@ahnu.edu.cn

technologies is reasonable because these can decrease expenses and costs. These processes markedly enhance multi-availability compared with real fire exercises. In addition, these processes can lower the latent danger and uncertainty of accidents, such as accidental casualties of numerous participants, when exercises are conducted. Thus, obtaining real trajectories from emergency videos is a critical problem. This problem can be solved by using a multi-object tracking (MOT) approach.

MOT has been a long-term goal in computer vision for tracking the trajectories of multiple objects of interest when estimating videos[1]. Extracting pedestrian motion trajectories using the MOT method has practical importance in trajectory data analysis[2–4], evacuation path planning, and other issues. Based on the real-time monitoring of videos of large-scale crowds, the MOT method can extract pedestrian motion trajectories during evacuation, helping emergency management departments to obtain increasingly intuitive crowd evacuation information and provide certain references for emergency escape and the formulation of evacuation strategies.

Surveillance technology[5] is widely used in daily life, and surveillance cameras help monitor and record everything that occurs around us, thereby providing a good platform for obtaining sources of emergency data. Previous research on video data was not effective because of the limitations in computing performance; however, with the continuous improvement of computer hardware and introduction of artificial-intelligence-related technologies, computer vision research has entered a new era. Deep learning[6] has become a popular artificial intelligence technology in recent years. Its advantage lies in its ability to learn specific tasks and extract features, accurately grasping the detailed features of images. MOT[7–9] is a classic midstream task in computer vision[10], and it is different from the analysis of image-level data through object detection. Object detection manages spatiotemporal data, such as videos, marks the position of an object in each video frame with a bounding box, and assigns the same ID to the same object. Owing to the powerful learning and feature extraction abilities of deep learning, multi-target tracking algorithms based on deep learning have been qualitatively improved in terms of tracking accuracy and reasoning speed. Therefore, a MOT algorithm based on deep learning can be used to extract the trajectory of pedestrians in an emergency scene under surveillance as a better way to obtain trajectories. Costs are low and restrictions are few, which are the requirements of trajectory extraction in various scenarios. In addition, trajectory data obtained in this way are suitable for relevant research on pedestrians, vehicles, and other objects in a DT. The DT concept[11,12] was first proposed and applied in the aerospace industry to remotely monitor and detect aircraft faults by building a full-life twin system. With the continuous extension of this concept, new DT concepts, such as smart cities, smart transportation, and smart parks, have been proposed by experts and researchers. Specifically, the trajectory data of pedestrians, vehicles, and other objects obtained from surveillance videos through multi-target tracking can provide good data support for updating the location status of the corresponding DT. However, after the real trajectory data are extracted in this manner and visualized by conducting simulation experiments, many redundancies and jitters of trajectory points occur, which do not conform to the realistic trajectory characteristics and cannot be directly applied in subsequent work.

Thus, this study makes the following contributions to the literature:

(1) Pedestrian trajectory data in an emergency scene video are obtained by using the FairMOT[13] multi-target tracking algorithm, and the data are preprocessed to form a trajectory dataset.

(2) Visualization is conducted based on the trajectory dataset, i.e., the extracted trajectory features are visualized, and the trajectory point redundancy and jitter phenomena are analyzed.

(3) A trajectory optimization method combining trajectory point decimation and Savitzky–Golay (S–G) filtering[14,15] is proposed to reduce redundancy and smooth the trajectory data, and an optimized trajectory is drawn to validate the proposed method.

2 Related research

The primary tasks of multi-target tracking (i.e., MOT) can be divided into locating multiple objects, maintaining their identities, and generating their respective tracks in a video. Tracked objects can include people, animals, vehicles, and other non-living objects. In contrast to single-target tracking, multi-target tracking uses bounding boxes to frame object locations and assigns the same ID ordinal to the same target for tracking representation. The number of targets and their appearance are a priori unknown. The challenges in multi-target tracking can be primarily summarized as follows:

(1) Occlusion between targets and between targets and the environment;

(2) High similarity in appearance between targets;

(3) Severe weather (rain, fog, sandstorms, etc.);

(4) Large-scale crowds.

According to the classification of object initialization, existing multi-target tracking algorithms can be categorized into two types: detection-based tracking (DBT) and detection-free tracking (DFT). The DBT method is more popular because it discovers new objects and automatically terminates disappearing objects, whereas the DFT method cannot manage the emergence of new objects, although it has the advantage of not requiring a pretrained object detector. According to the classification of processing modes, multi-target tracking algorithms can also be classified as online and offline. The difference between online and offline methods is that the latter cannot use the information of future frames; thus, the relative accuracy is low, but the processing speed is high, which is suitable for real-time scenarios.

The SORT[16] algorithm is an early classical online-detection-based multi-target tracking algorithm. In the object detection link, the faster region-based convolutional neural network[17] was employed as a detector to replace the detector calculated using aggregate channel features and improved the MOT accuracy[18] value by 18.9% on the MOT15[19] dataset. In addition, this algorithm was the first case of using the Kalman filter[20] to match the Hungarian algorithm[21], which is a classic tracking paradigm. The advantages of this algorithm are its simple structure and fast reasoning; however, because it does not use appearance features and uses only the motion model for prediction, the tracking accuracy is poor.

The DeepSORT[22] algorithm is an improved version of the SORT algorithm. In the matching process of the existing trajectory and current detection, the motion and appearance information are comprehensively considered, and the minimum cosine distance between the contained feature vectors is detected and tracked as the appearance similarity between the two. In addition, a cascade matching strategy is used to assign higher priority to more frequently observed objects, thereby improving the matching accuracy and effectively alleviating the ID switch problem caused by occlusion (experiments proved that it is reduced by approximately 45%).

Compared with the SORT algorithm, the DeepSORT algorithm improves the accuracy on the MOT16[23] dataset; however, the inference speed is reduced by approximately 20 Hz owing to the added calculation for appearance features. Wang et al. proposed the JDE algorithm, which is different from the previous two-stage and post-detection tracking paradigms[24]. Further, they innovatively designed a network by fusing the two parts of the target detection link and appearance feature information extraction link into a network, which markedly improved the inference speed of the multi-target tracking algorithm and achieved near-real-time frame processing. Because most computations can be shared after the two modules are fused into a network, the primary advantage of the JDE algorithm is its speed compared with the state-of-the-art algorithm at that time (2–3 times faster), but its accuracy is 2.6% lower. Moreover, this difference is large when compared with DeepSORT, and the number of ID switches is doubled.

In a recent study, Zhang et al. made improvements based on the JDE algorithm and proposed the FairMOT

algorithm[13]. They summarized three important factors causing the poor performance of previous one-shot methods, such as JDE, and proposed optimization schemes in a targeted manner. First, the anchor-based method is not suitable for the reidentification (Re-ID) process; thus, an anchor-free method should be used. Features extracted at anchor points are likely to be out of alignment with the center of the object, and two anchor points will also appear to be responsible for the center of an object, leading to ambiguity. Concurrently, the feature map is often downsampled eight times to balance accuracy and speed, which is marginally rough for the Re-ID process; the center of the object will likely be inconsistent with the rough anchor position. The location of the target center should be estimated using pixel keys in high-resolution feature maps, and ID classification should be performed. Second, for one-shot methods, the Re-ID process should be performed by applying multilayer feature fusion because high- and low-level features are required to match for large and small targets, respectively, which can further reduce the ID switch. Finally, existing Re-ID methods usually learn high-dimensional features; however, learning low-dimensional features is more beneficial to the MOT process.

Learning low-dimensional features involves fewer training images than Re-ID, helps reduce the risk of overfitting to small data, decreases the calculation time, and improves robustness. Because the algorithm accounts for the calculation speed and tracking accuracy, this method is selected in this study for the extraction of pedestrian trajectory data information from surveillance videos.

In addition, a method[25] combining trajectory point decimation with the S–G filter[26] is proposed for smooth trajectory optimization. The decimation process can remove some redundant trajectory point data and obtain trajectory key points. S–G filtering is a commonly used filtering method for the smooth denoising of data streams; its core idea is to perform K-order polynomial fitting on data points within a certain length of window to obtain the fitted results. After discretization, it essentially becomes a special sliding window weighted averaging algorithm. The key feature of the S–G filter is that it filters out noise data while ensuring that the shape and width of the signal remain unchanged. We apply the S–G filter to smooth the pedestrian trajectories, thereby increasing the smoothness of the pedestrian trajectories while retaining the real trajectory motion characteristics after optimization, which meets the real needs.

3 Proposed framework

The proposed framework primarily consists of two parts, as shown in Figure 1. Specifically, the first part is based on the FairMOT algorithm for extracting pedestrian trajectory information from an emergency scene video and form a trajectory dataset after data preprocessing. The second part optimizes the trajectories in the trajectory dataset and draws them. The detailed process of this method involves inputting an emergency scene video into the FairMOT model.

Then, the FairMOT algorithm is used to extract and preprocess the trajectory data of the pedestrians in the input video. Finally, the method forms a trajectory dataset after preprocessing. Aimed at the large number of redundancies and trajectory point jitter phenomena in the trajectory point data extracted in this way in real extracted trajectory data, a trajectory optimization method[27] that combines trajectory point extraction and S–G filtering is used to effectively alleviate the phenomenon of trajectory point redundancy and trajectory point jitter and retain the real trajectory characteristics while reducing the amount of data. This can provide more data support for the formulation of pedestrian trajectory data analysis and personnel evacuation in emergency scenarios, use the trajectory data extracted and optimized by trajectory, and establish a trajectory case of virtual and real integration.

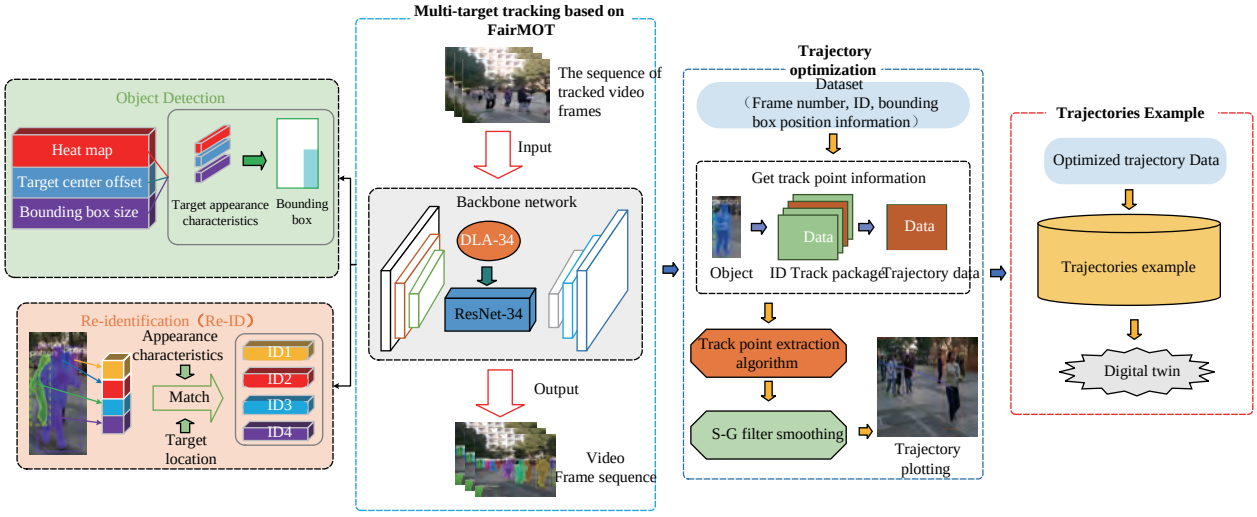


Figure 1 Overview of trajectory extraction and optimization framework based on multi-target tracking.

3.1 Trajectory extraction and data preprocessing

Given the accuracy and inference speed of tracking, the FairMOT algorithm was used to acquire the trajectory information of pedestrians in the video. The algorithm is an end-to-end one-shot multi-target tracking method based on deep learning that consists of three important components: a backbone network, object detection branch, and Re-ID branch. Object detection and Re-ID are two completely isomorphic branches that are used to detect target objects and extract Re-ID features, respectively. The object detection branch uses an un-anchored object detection method to estimate the center of a target object, and its size is represented by a position-aware measurement map. In addition, the Re-ID branch estimates the Re-ID characteristics of each pixel and is used to describe the pixel-centric target object. Because these two branches are completely isomorphic, compared with the method of cascading object detection and Re-ID, the FairMOT algorithm eliminates the unfair advantage of a single branch, effectively learns high-quality Re-ID features, and achieves a good balance between the object detection and Re-ID tasks, resulting in better tracking results. To achieve a good balance between speed and accuracy, the FairMOT algorithm uses the ResNet-34 network as a backbone, alongside an improved deep layer aggregation[28] (DLA) method to fuse multi-layer features (DLA-34), as shown in Figure 1.

Compared with DLA, the improved DLA-34 has more jump connections between low- and high-level features and can combine multilayer features. In addition, DLA-34 replaces the convolutional layers of all upsampled modules with variable convolutional layers, enabling it to dynamically adjust the receiving domain according to the proportion and attitude of the target object. The backbone network takes a video frame image as the input and outputs a $c \times w \times h$ feature map, where w and h are the width and height of the input image, respectively; $w = W/4$ and $h = H/4$.

The object detection branch of the FairMOT algorithm is based on CenterNet[29]. In addition, three parallel heads were added to DLA-34 to estimate the heatmap, target center offset, and bounding box size. The head of each parallel head is a regression operation on the bounding box by 3×3 convolution of the output features of DLA-34 (number of channels = 256), and the output is a 1×1 convolutional layer to generate the final target. In the FairMOT algorithm, a heatmap is used to estimate the center point position of the target object, which is of size $1 \times W \times H$.

If a location in the heatmap coincides with the center of the real target object, the response for that location is expected to be 1. As the distance between the location and center of the target object in the heatmap increases,

the response decays exponentially. The center offset section is responsible for more precise positioning of the object, optimizing center misalignment, and improving detection performance. The bounding box dimensions section sets the height and width of the bounding box.

The goal of the Re-ID branch[30] is to generate features that distinguish the target object, which can extract the features of the center point of the target object while associating it with its ID. Ideally, the affinity between different objects should be lower than that between the same objects. To achieve this goal, this branch extracts the Re-ID information of each target object using a 128-kernel convolutional layer on the feature map, $M \in {}^{128\times H\times W}$, of the backbone network, generates a feature map that distinguishes different targets, and extracts a Re-ID feature, $M_{x,y} \in \mathrm{R}^{128}$, which is the center point of the (x, y) target object from the feature map. The introduction of appearance features also improves the tracking accuracy. As a result, the inference time is markedly reduced.

The output information obtained by the FairMOT algorithm has two main parts: (1) the tracking video generated with the same frame rate as the original video, in which each line is labeled with an ID through the bounding box and (2) a record of the text information of the trajectory data. Each record contains the frame number (frame), ID, horizontal coordinate of the upper-left corner of the bounding box (x), ordinate of the upper-left corner of the bounding box (y), width of the bounding box (w), and height of the bounding box (h). This information is not the ideal form of trajectory data; thus, it is preprocessed. First, center point conversion of the trajectory is performed using the center point coordinate of the bounding box as the trajectory point coordinate of the pedestrian. Because this is an image coordinate system, the conversion formula is as follows:

$$\begin{cases} x_c = x + \dfrac{1}{2}w \\ y_c = y + \dfrac{1}{2}h \end{cases} \tag{1}$$

We then arrange each record based on the frame number into continuous track point data based on each ID to form a track dataset. For an object with t track points and ID n, the trajectory data are $I_n\{(frame_1, x_{c1}, y_{c1}), (frame_2, x_{c2}, y_{c2}), ..., (frame_t, x_{ct}, y_{ct})\}$. We use the converted trajectory dataset to simulate the real performance of the obtained trajectory data using a dot operation to simulate pedestrians in the original video background.

3.2 Track data extraction and smoothing

Because the crowd in an evacuation scene may be standing, moving less, or moving slowly, there are many redundant track points, which markedly increases the amount of data. In addition, there are differences in the calculation of the bounding box size by the multi-target tracking algorithm, which easily leads to the occurrence of bounding box jitter; thus, the final generated trajectory is too rough and does not conform to the real trajectory point characteristics. To solve these problems, we propose a method suitable for such scenarios to optimize the trajectory by combining trajectory point extraction and S–G filtering.

The trajectory point redundancy considered in this study is primarily a large number of repeated or approximately repeated trajectory points. Thus, we propose a simple and efficient way to remove redundant trajectory points based on the Euclidean distance of neighboring points by setting thresholds, as shown in Algorithm 1:

For different thresholds, ω,the degree of extraction of trajectory points is different. In this case, we select the appropriate threshold through experiments, retaining key trajectory points while reducing the amount of data. Then, the S–G filter, which is a low-pass filter proposed by Savitzky and Golay in 1964 and widely used to smooth noise reduction in data streams, is used to smooth the key trajectory. Therefore, as a filtering method based on local polynomial least squares fitting in the time domain, this filter can remove noise while keeping

Algorithm 1 Tracking point extraction

Input: Trajectory datasets Ψ, Tracing track set Γ
Output: collection of trajectories $\ddot{\Psi}$
Initialization: $\ddot{\Psi} \leftarrow$ $|\ddot{\Psi}| \rightarrow 0$ $|\Gamma| \leftarrow$ $|\Gamma| \leftarrow 0$;
// Traversion of all video frames to obtain related target trajectories
foreach *video frame* $\in$ *video* **do**
foreach *target object* $o_j \in$ O **do**

end
if $o_j \subset f_i$ then
if $o_j \notin \forall \tau_k$ $(\tau_k \subset \Gamma)$ **then**
create a new ID trace track $\hat{\tau}$
$\hat{\tau} \rightarrow \Gamma$
$|Gamma| \leftarrow |Gamma| + 1$
end
else
update $\tau_k, o_j \rightarrow \tau_k$
$|\tau_k| \leftarrow |\tau_k| + 1$
end
end
end
// traverse trajectory points in the ID trace track Γ
foreach $\tau_i \subset \Gamma$ **do**
$\psi_i \leftarrow \varnothing$
foreach *trajectory point* $\epsilon_j \in \tau_i \left(0 \leqslant j \left\langle \left| \epsilon^i{}_j \right| - 1 \right)\right.$
do
calculate the Euclidean distance $\epsilon_j^i \Delta$ $\Delta_j$between ϵ_j^i and its adjacent point ϵ_{j+1}^i
if $\Delta_j \geq$ **then**
$\varepsilon_j^i \rightarrow \psi_i$
end
else
remove the adjacent point ϵ_{j+1}^i ;
end
end
$\psi_i \rightarrow \ddot{\Psi}$
end

the shape and width of the signal unchanged. Thus, the S–G filter is available to smooth and optimize the preprocessed trajectory key points. Specifically, a filter window is set with a width of $2m$+1 by constructing an n^{th}-order polynomial, ($n \leq 2m+1$). Alternatively, using this polynomial to fit a set of data under $x(i)$ ($-m \leq i \leq m$) yields the fitting equation:

$$f(\theta) = a_0 + a_1\theta + \ldots + a_n\theta^n = \sum_{k=0}^{n} a_k\theta^k \tag{2}$$

To determine the polynomial of Eq. (2), its coefficients must first be addressed. If $i = 0$, then $f(0) = a_0$, and we can obtain the center point of the window $x(0)$ by $f(0)$ and then fit all the data by sliding the window. In addition, the S–G filter can be used to manage the related data in the sliding window. Specifically, a weighted average calculation is performed to obtain the center point (0). The sum of squared residuals, Θ_i, is used to fit the final result:

$$\Theta_n = \sum_{\theta=-m}^{m} [f(\theta) - x(\theta)]^2 = \sum_{\theta=-m}^{m} \left[\sum_{k=0}^{n} a_k\theta^k - x(\theta) \right]^2 \tag{3}$$

In addition, to minimize the sum of squares of the residuals, Θ_n, the partial derivative of Θ_n for each parameter should be 0, i.e., the derivative equation is

$$\frac{\partial \Theta_n}{\partial a_i} = \sum_{\theta=-m}^{m} \theta^i \left[\sum_{k=0}^{n} a_k\theta^k - x(\theta) \right] = 0 \tag{4}$$

When $k \in [0,n]$, based on the above equation, we have

$$\sum_{k=0}^{n} a_k \sum_{\theta=-m}^{m} \theta^{i+k} = \sum_{\theta=-m}^{m} \theta^{i} x(\theta) \tag{5}$$

To solve Eq. (5), we assume matrix $\mathbf{A}=(b^{i}_{\theta})_{(2m+1)\times(n+1)}$, where $b^{i}_{\theta}=\theta^{i}$, and we set matrix $\mathbf{B}=\boldsymbol{A^{T}A}$. Then, the following can be obtained:

$$a_{\theta}^{k} = \sum_{j=-m}^{m} b_{\theta}^{j} b_{j}^{k} = \sum_{j=-m}^{m} \theta^{j+k} = a_{k}^{\theta} \tag{6}$$

where $k \in [-m, m]$ and $\theta \in [0,n]$. We assume that $\mathbf{X} = [x(-m)...x(m)]^{\mathrm{T}}$ and $a = [a_0...a_n]^{\mathrm{T}}$. Then, $\boldsymbol{Ba = A^{T}Aa = A^{T}}$, Therefore, the following can be obtained:

$$a = B^{-1}A^{T}x = Hx \tag{7}$$

where the first row vector of matrix H is the convolution coefficient determined only by n and is independent of x.

Based on the aforementioned principle of the S–G filter, the preprocessed trajectory key points are smoothed, and the larger the window size, the stronger the smoothing effect but makes the more different it is from the real trajectory. Therefore, to improve the realism and smoothing effect of the trajectory, a third-order polynomial with a window size of 21 is used for fitting in the proposed approach, and an overview of the operation is shown in Figure 2.

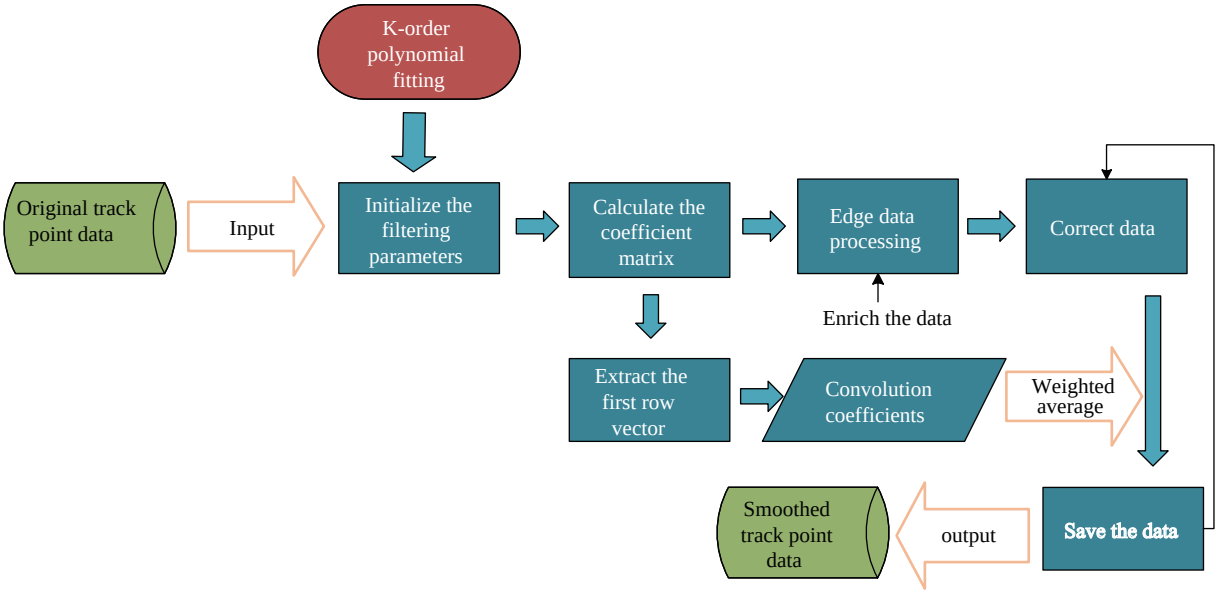


Figure 2 Overview of S–G filter processing.

First, we initialize the relevant filtering parameters and then calculate the coefficient matrix by using Eq. (7), where the first row vector is the convolution coefficient. In edge data processing, the data that are truncated at the beginning and end are added, and the convolution coefficient is calculated as per the aforementioned steps; thus, the value of the center position can be weighted and averaged to the data in the window and saved. Finally, we can draw the target's travel trajectories according to the optimized trajectory points.

4 Experiments

In this section, we describe an experiment conducted on the proposed framework to validate its superiority and feasibility. The proposed framework consists of two parts. The first is validating the FairMOT-based model, and the motivation was to demonstrate the tracking effect in several different scenarios. The second is

demonstrating the superiority of the optimized trajectory through experiments. The optimized trajectory data are drawn in the corresponding video.

4.1 Settings of experimental environment

In this study, all programming was performed in the Python language. The related configurations are listed in Table 1. Several open-access datasets from the MOT field were used to validate the feasibility of the proposed method. Specifically, the MOT17-01, MOT17-03, MOT17-07, MOT17-08, and MOT17-12 video sequences from the MOT17 dataset and six videos of different emergency scenarios were available to analyze the related performance of the multi-scene tracking effect and trajectory optimization. Scene # 1 was an air defense evacuation surveillance video from a century square in H city, with nearly 250 personnel participating. Scene # 2 was an earthquake emergency evacuation video of a primary school in G city, with approximately 270 teachers and students. Scene # 3 was a fire escape video of a middle school, with approximately 310 students. Scene # 4 was a self-shot dormitory fire drill video simulating a sudden fire through red fog with approximately 280 teachers and students. Scene # 5 was a vicious injury incident in a prison, with approximately 244 offenders. The CUSK-SYSU dataset was used to train the MOT algorithm; it was divided into training and test sets. The training set contained 11206 images and 5532 query objects, and the test set contained 6978 images and 2900 query objects. Finally, Table 2 lists the specific training parameters.

Table 1 Configuration of the computation environment

Name	Value
Operation	Windows 10
CPU	AMD R7-5800H
GPU	Nvidia GeForce RTX 3080Ti
Memory	DDR4 16GB

Table 2 Training parameter settings

Parameter	Value
Size of samples	15
Learning rate η	0.001
Number of training epochs	16
Dimensions of Re-ID	128
Size of input images	1088 × 608
Size of feature map	272 × 152

4.2 Performance evaluation

To verify the effectiveness of the trajectory optimization method used in this study, the smoothing index and average moving speed of the crowd were employed to comprehensively evaluate the true degree of the smoothing effect of the trajectory. The evaluation criteria can be summarized as follows:

(1) Smoothness Index (SI): This quantitatively evaluates the smoothness of the trajectory. The $\forall \tau_j^i \in \Gamma = i \in N \mid \tau^i$trajectory is calculated by the inner angle with the two line segments composed of three adjacent trajectory points. Clearly, the larger the mean inner angle is, the higher the smoothness, and the aforementioned process can be represented by the following equation:

$$SI(\tau) = \frac{1}{n-1} \sum_{j}^{n-2} \Theta(\tau_j \tau_{j+1}, \tau_{j+1} \tau_{j+2}) \tag{8}$$

(2) Average moving speed (AMS) of a trajectory: This assesses the degree to which the original trajectory velocity characteristics are preserved. Specifically,

$$\forall \tau j \in \Gamma = \{0 \le i \le n-1 | \tau \quad _j\},$$

the Euclidean distance of adjacent trajectory points is calculated, and the average moving speed Δ of the trajectory is obtained to measure the trajectories. AMS(τ) can be represented by the following equation:

$$AMS(\tau) = \frac{1}{n-1} \sum_{j=1}^{n-1} \quad (\tau_j, \tau_{j+1}) \tag{9}$$

To validate the robustness of the FairMOT method in a real emergency environment and its advantages over other algorithms, we selected a video from HiEve[31] in a severe fighting scenario involving several people to

perform our experiment. First, the objects in the video were labeled to obtain the ground truth data. Specifically, there were ten labeled objects, and the video frame rate was 25. Then, the classic CLEAR metric was used for performance evaluation and compared with the existing SORT algorithm[16] and DeepSORT algorithm[22]. The detailed results are presented in Table 3.

Table 3 Comparison between our method and other existing methods in terms of several indicators

Methods	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	IDs ↓	MOTP ↑
SORT [16]	30.232	31.178	25	88	100	72.843
DeepSORT[22]	36.642	38.212	34	79	86	77.006
Ours	43.086	45.245	48	52	69	78.071

The results show that the proposed method achieved better tracking performance than the classical SORT and DeepSORT algorithms. The numbers of ID switches and missed detections within the key time frame will have a strong impact on the authenticity of the obtained track data, the value of the data itself, and the related reasoning research based on the data; simply counting the numbers of global ID switches and missed detections cannot fully describe the algorithm performance in an emergency environment. In this case, the average Re-ID accuracy and weighted average Re-ID accuracy indicators are proposed to highlight the superiority of the proposed method. Specifically, 50 continuous key frames of 10 target objects were counted, and the average Re-ID accuracy rate was calculated as the ratio of the number of frames in the key range that the target object kept its ID unchanged to the total number of frames, and the average value was calculated. The weighted average Re-ID accuracy rate was used to assign different weights to different objects, and the average was then calculated. In many experiments, setting the weight of the two primary people involved in the fight to 0.2 and the remainder to 0.075 produced better results. Detailed information is presented in Table 4.

In most complex emergent cases, such as crowd-fighting scenarios, the frequent switching of object IDs causes poor object tracking accuracy. As shown in Table 4, the proposed method still achieves superior results over other state-of-the-art approaches, with good robustness and stability.

Table 4 Comparison between our method and other existing methods in terms of the two proposed indicators

Methods	Average Re-ID accuracy	Weighted average Re-ID accuracy
SORT [16]	82.2%	82.4%
DeepSORT[22]	85.4%	84.3%
Ours	90.8%	90.1%

4.3 Results and comparison

We trained the FairMOT model on the CUSK-SYSU dataset, and the total trained loss function is given by

$$L_{\text{total}} = \frac{1}{2}\left[\frac{1}{e^{w1}}L_{dec} + \frac{1}{e^{w2}}L_{ide} + w1 + w2\right] \tag{10}$$

where $w1$ and $w2$ are hyperparameters, and L_{dec} and L_{iec} are the losses of object detection and rerecognition, respectively, to balance the two tasks of object detection and RE-ID. We also conducted training on the FairMOT model to track the MOT17-01, MOT17-03, MOT17-07, MOT17-08, and MOT17-12 video sequences from the MOT17 dataset and the pedestrian objects in 5 videos in different emergency scenarios; the results show that the proposed method can accurately capture the objects in the video, and the effect is shown in Figures 3 and 4.

Because the position of every pedestrian in the trajectory data generated by multi-target tracking is determined by the coordinate information of the bounding box, there are some errors when calculating this information, which can easily lead to problems such as redundant trajectory points and coarse-grained tracking trajectories. Specifically, a two-dimensional simulation of crowd movement trajectory data can visually

Figure 3 Examples of trace results on the MOT17 test set.

represent the above problems by observing crowd movement in a scene and describing the direction for subsequent trajectory optimization, which is important for the reproduction of a three-dimensional scene. Therefore, a crowd motion simulation experiment was conducted in an emergency scenario prior to trajectory optimization.

For ease of observation, we selected related videos of earthquake emergency evacuation drills for the two-dimensional simulation. Intuitively, the same background image with the same frame number as the original video created by the matplotlib package can replace the pedestrians with solid dots, which are drawn on the corresponding figures according to the obtained trajectory data. Then, the ffmpeg tool can be used to compress these pictures into a two-dimensional simulation video with the same frame rate as the original video; a simulation result example is shown in Figure 5.

The results show that salient trajectory jitter exists in the obtained trajectory data, and a large amount of redundancy exists in the data itself.

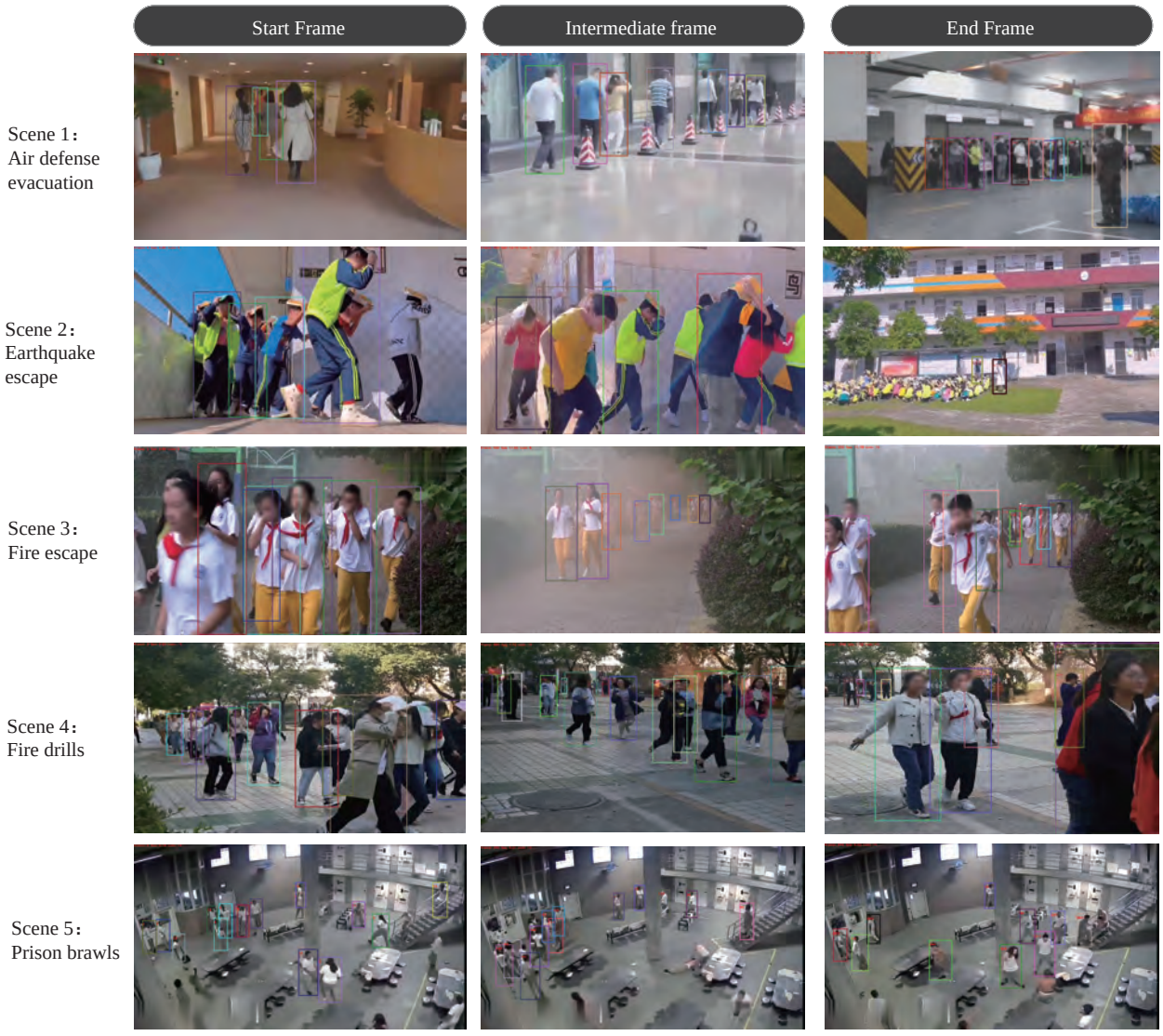


Figure 4 Frame tracking results of emergency scene videos.

For the redundancy of trajectory data, to reduce the amount of data, the trajectory point extraction algorithm obtains only the key trajectory points; thus, redundant trajectory points are removed based on the Euclidean distance between two consecutive points by setting the threshold ω. In particular, to find the appropriate and reasonable threshold, an object trajectory in Scene #1 was selected to set four different thresholds (i.e., ω=1, 2, 5, 10) and analyze it.

As depicted in Figure 6, many redundant track points still exist. Some key points are shown at the beginning, where the lack of track points is more apparent.

The results show that salient trajectory jitter exists in the obtained trajectory data, and a significant amount of redundancy exists in the data.

To equalize the true characteristics and redundancy of the trajectories, ω=2 was selected as the best threshold. There were 798 trajectory points before extraction, whereas the number of key trajectory points obtained after extraction was 335, which is marked reduction in redundant data. In addition, we conducted a related experiment on a video sequence in MOT17 and all objects in five different emergency video scenarios and calculated the total numbers of trajectory points before and after extraction. The details are listed in Table 5. Table 3 describes the effectiveness of this method for removing redundant trajectory points. In particular,

Figure 5 Example of crowd evacuation.

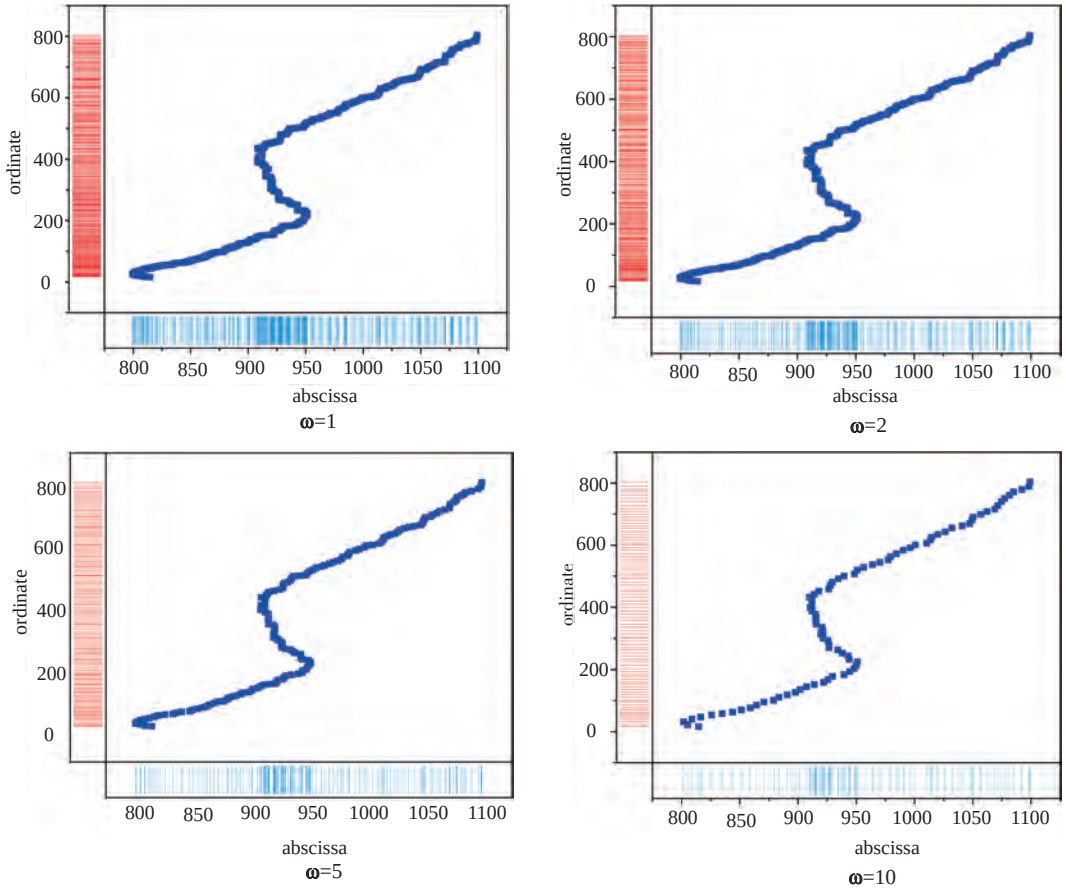


Figure 6 Track point overview for several different thresholds, ω.

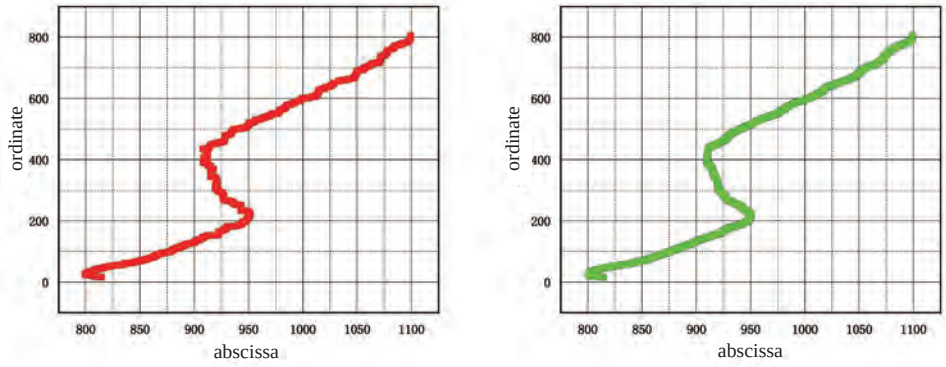
this method can decrease the amount of data in long-term surveillance video trajectory extraction of large crowds. The removal of redundant track points also contributes to smoothing optimization.

After trajectory point extraction, the trajectory key points of the object can be acquired. Undoubtedly, using the previous analysis of the trajectory point performance in the simulation video, the trajectories are found to

often occur during jitter, which is potentially caused by an error in the position and size of the bounding box of the same object in the multi-target tracking algorithm and short identity switching. In this case, the S–G filtering algorithm was used to smooth and optimize the trajectory composed of the key points of the trajectories after decimation. To address this issue, a comparison between before and after smoothing of the key trajectory points of the object in Scene #1 is shown in Figure 7.

Table 5 Comparison of numbers of trajectory points in different scenarios before and after extraction

Scenarios	Number of before	Number of after
MOT17-01	4310	2076 ↓
MOT17-03	94994	38998 ↓
MOT17-07	10369	7651 ↓
MOT17-08	8029	5257 ↓
MOT17-12	5073	4521 ↓
scene #1	6931	4649 ↓
scene #2	9583	8406 ↓
scene #3	5641	4824 ↓
scene #4	14836	11817 ↓
scene #5	7936	6075 ↓



(a) Before smoothing (b) After smoothing

Figure 7 Comparison with trajectory smoothing processing.

To evaluate the smoothness of the trajectories quantitatively, the SI and AMS were used to evaluate the degree of retention of the original trajectory speed features. Ablation experiments were performed to compare the proposed trajectory optimization method, which combines trajectory point decimation and S–G filtering. Specifically, owing to the filter window size, object trajectories with fewer than 21 track points were filtered, and the average smoothing index was calculated by extracting and filtering the filtered object trajectory in the five emergency scene videos.

Then, the trajectories of an object from Scene # 4 were selected, and the trajectories before and after optimization were drawn using OpenCV. This was compared with the manual trajectory (i.e., the trajectory characteristics observed by the human eye). Specifically, the blue curves in Figure 8a–c represent the original trajectory curves of the target object, the trajectory curve processed by the trajectory point extraction approach, and the trajectory optimized by the method in this study, respectively. The red curve in Figure 8d represents the trajectory curve drawn manually by observing the movement trajectories of the target object in the video.

The results show that the proposed framework can effectively optimize the trajectory data redundancy and jitter phenomena of the multi-target tracking method in emergency video scenarios. In addition, the jitter becomes smoother and more consistent with the trajectory characteristics observed by the human eye in real life.

The experimental results are shown in Figure 9. In the ablation experiment with the optimized trajectories of the proposed framework, the highest smoothing index was compared with the original trajectory, trajectory after only trajectory point extraction processing, and trajectory after only smoothing by S–G filtering. The results of the fusion method used in this study were closer to the average velocity of the original trajectory.

(a) Original trajectory

(b) Only the trajectory after the trajectory point is diluted

(c) The trajectory after optimization

(d) Manual tracks

Figure 8 Comparison of trajectory curves.

5 Conclusions

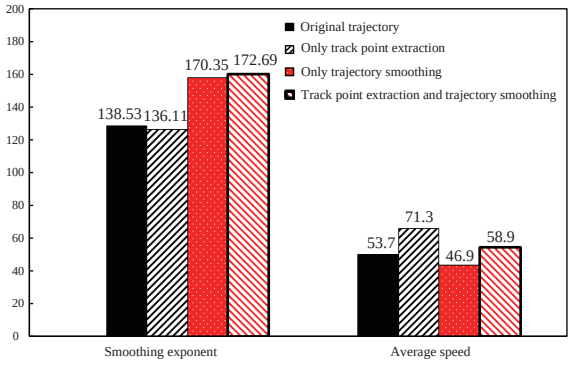


Figure 9 Comparisons of SI and AMS.

To obtain the real trajectory data of crowd evacuation and construct a real trajectory dataset under large-scale emergency scenarios, a novel learning framework for trajectory extraction based on multi-target tracking was proposed. In particular, we focused on the current situation in which the suddenness and unpredictability of emergency events lead to difficulty in obtaining track point data in these types of scenarios. Compared with the traditional method of extracting trajectory data, there are many advantages of the proposed framework, including ease of acquisition, low cost, and few restrictions. Additionally, considering the roughness of trajectories, the redundancy of trajectory points and jitter phenomena often result from the extracted trajectory data; therefore, a method combining a trajectory point extraction algorithm and S–G smoothing filtering can be used to optimize the trajectory.

Based on the proposed framework, the amount of track point data was reduced, and the smoothness of the trajectory could be improved. The trajectory characteristics observed by the human eye were fitted, and to a certain extent, they could retain the true characteristics of the trajectory. In addition, follow-up pedestrian trajectory data analyses, such as behavioral abnormality analyses, must be performed in emergency scenarios.

However, there are still many limitations of the proposed framework:

(1) The raw data of the upstream trajectory depend heavily on the performance of the multi-target tracking algorithm itself, and its accuracy and speed still have significant room for improvement.

(2) In fact, the image coordinates of the trajectories are 2D, while 3D coordinate information is required in

the virtual scenario. Therefore, it is necessary to develop an efficient method for mapping the coordinates in a 3D scene.

In future work, we plan to optimize the performance of the tracking algorithm. In addition, based on the transformation of 2D to 3D space, we can achieve multi-agent path-planning targets in the virtual Web3D scenario by acquiring 3D pedestrian trajectory point data through video multi-target tracking.

Declaration of competing interest

We declare that we have no conflict of interest.

References

1 Chen L, Ai H Z, Zhuang Z J, Shang C. 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, CA, USA, IEEE, 2018, 1–6
DOI: 10.1109/icme.2018.8486597

2 Yan Z, Spaccapietra S. Towards semantic trajectory data analysis: a conceptual and computational approach. In: Proceedings of the VLDB 2009 PhD Workshop Co-located with the 35th International Conference on Very Large Data Bases (VLDB 2009) Lyon, France, 2009

3 Ferrero C A, Alvares L O, Bogorny V. Multiple aspect trajectory data analysis: research challenges and opportunities. In: XVII Brazilian Symposium on Geoinformatics. 2016

4 Schreck T, Bernard J, von Landesberger T, Kohlhammer J. Visual cluster analysis of trajectory data with interactive kohonen maps. Information Visualization, 2009, 8(1): 14–29
DOI: 10.1057/ivs.2008.29

5 Masli A, Peters G F, Richardson V J, Sanchez J M. Examining the potential benefits of internal control monitoring technology. The Accounting Review, 2010, 85(3): 1001–1034
DOI: 10.2308/accr.2010.85.3.1001

6 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436–444
DOI: 10.1038/nature14539

7 Luo W, Xing J, Milan A, Zhang X, Liu W, Kim T K. Multiple object tracking: A literature review. Artificial Intelligence, 2021, 293: 103448
DOI: 10.1016/j.artint.2020.103448

8 Leal-Taixé L, Milan A, Schindler K, Cremers D, Reid I, Roth S. Tracking the trackers: an analysis of the state of the art in multiple object tracking. 2017

9 Ciaparrone G, Luque Sánchez F, Tabik S, Troiano L, Tagliaferri R, Herrera F. Deep learning in video multi-object tracking: A survey. Neurocomputing, 2020, 381: 61–88
DOI: 10.1016/j.neucom.2019.11.023

10 Shapiro L G, Stockman G C. Computer Vision, 2001

11 Tao F, Zhang H, Liu A, Nee A Y C. Digital twin in industry: state-of-the-art. IEEE Transactions on Industrial Informatics, 2019, 15(4): 2405–2415
DOI: 10.1109/tii.2018.2873186

12 Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterising the digital twin: a systematic literature review. CIRP Journal of Manufacturing Science and Technology, 2020, 29: 36–52
DOI: 10.1016/j.cirpj.2020.02.002

13 Zhang Y, Wang C, Wang X, Zeng W, Liu W. FairMOT: on the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision, 2021, 129(11): 3069–3087
DOI: 10.1007/s11263-021-01513-4

14 Ovaska S J, Valiviita S. Angular acceleration measurement: a review. In: IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No.98CH36222). St. Paul, MN, USA, IEEE, 2002, 875–880
DOI: 10.1109/imtc.1998.676850

15 Cai T J, Tang H. An overview of the principle of least squares fitting of smoothing filters. Digital Communication, 2011, 38(1), 63–68
DOI: 10.3969/j.issn.1001-3824.2011.01.017

16 Bewley A, Ge Z Y, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). Phoenix, AZ, USA, IEEE, 2016, 3464–3468
DOI: 10.1109/icip.2016.7533003

17 Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149
DOI: 10.1109/tpami.2016.2577031

18 Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, 2008, 1–10
DOI: 10.1155/2008/246309

19 Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K. MOTChallenge 2015: towards a benchmark for multi-target tracking. 2015

20 Kalman R E. A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 1960, 82(1): 35–45
DOI: 10.1115/1.3662552

21 Kuhn H W. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 1955, 2(1–2): 83–97
DOI: 10.1002/nav.3800020109

22 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). Beijing, China, IEEE, 2018, 3645–3649
DOI: 10.1109/icip.2017.8296962

23 Milan A, Leal-Taixe L, Reid I, Roth S, Schindler K. MOT16: A benchmark for multi-object tracking. 2016
DOI: https://arxiv.org/abs/1603.00831

24 Wang Z D, Zheng L, Liu Y X, Li Y L, Wang S J. Towards real-time multi-object tracking. In: Vedaldi A, Bischof H, Brox T, Frahm JM. European Conference on Computer Vision. Cham: Springer, 2020,107–122
DOI:10.1007/978-3-030-58621-8_7

25 Pang J M, Qiu L L, Li X, Chen H F, Li Q, Darrell T, Yu F. Quasi-dense similarity learning for multiple object tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA, IEEE, 2021,164–173
DOI: 10.1109/cvpr46437.2021.00023

26 John A, Sadasivan J, Seelamantula C S. Adaptive savitzky-golay filtering in non-gaussian noise. IEEE Transactions on Signal Processing, 2021, 69: 5021–5036
DOI: 10.1109/tsp.2021.3106450

27 Liu Q, Chen D, Chu Q, Yuan L, Liu B, Zhang L, Yu N. Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. Neurocomputing, 2022, 483: 333–347
DOI: 10.1016/j.neucom.2022.01.008

28 Zheng L, Yang Y, Hauptmann AG: Person Re-identification: Past, Present and Future. 2016
DOI: https://arxiv.org/abs/1610.02984

29 Duan K W, Bai S, Xie L X, Qi H G, Huang Q M, Tian Q. CenterNet: keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2020, 6568–6577
DOI: 10.1109/iccv.2019.00667

30 Yu F, Wang D Q, Shelhamer E, Darrell T. Deep layer aggregation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 2403–2412
DOI: 10.1109/cvpr.2018.00255

31 Lin W Y, Liu H B, Liu S Z, Li Y X, Qian R, Wang T, Xu N, Xiong H K, Qi G J, Sebe N. Human in events: a large-scale benchmark for human-centric video analysis in complex events. 2020