MULTIPLE OBJECT TRACKING USING A TRANSFORM SPACE

Minglei Li[1*], Jiasong Li[1], Alexis Tamayo[1], Liangliang Nan[2]

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China;
2 Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, the Netherlands.

Commission IV, WG5

KEY WORDS: Multiple Object Tracking, Tracking-by-Detection, Data Association, Deep Features, Transform Space.

ABSTRACT:

This paper presents a method for multiple object tracking (MOT) in video streams. The method incorporates the prediction of physical locations of people into a tracking-by-detection paradigm. We predict the trajectories of people on an estimated ground plane and apply a learning-based network to extract the appearance features across frames. The method transforms the detected object locations from image space to an estimated ground space to refine the tracking trajectories. This transform space allows the objects detected from multi-view images to be associated under one coordinate system. Besides, the occluded pedestrians in image space can be well separated in a rectified ground plane where the motion models of the pedestrians are estimated. The effectiveness of this method is evaluated on different datasets by extensive comparisons with state-of-the-art techniques. Experimental results show that the proposed method improves MOT tasks in terms of the number of identity switches (IDSW) and the fragmentations (Frag).

1. INTRODUCTION

Multiple object tracking (MOT) aims to associate the detected objects across frames in a video stream. It is a fundamental problem in a wide range of applications, such as security management, autonomous driving, and traffic monitoring. Tracking-by-detection is a leading paradigm to solve the MOT problems (Ross et al., 2008; Avidan, 2007), which consists of two main steps, namely 1) detecting the potential locations of multiple objects frame-by-frame, and 2) associating the detected objects to the estimated position of existing tracks. The first step can be addressed with some recent successful learning-based detectors (Bochkovskiy et al., 2020; Ren et al., 2016). However, data association is still unresolved as it suffers from issues such as a large number of false-positive tracks.

Data association requires some clues, among which the most important information are the spatial positions and the appearance features. We observe that the previous works study the spatial positions mostly in image space using pixel-level positions (Kim et al., 2021; Dicle et al., 2013; Rezatofighi et al., 2015; Kim et al.,2015). However, the pixel changes of human motion in an image are not proportional to the actual human motion in the real world due to perspective projection. This work aims to improve the quality of pedestrian tracking by converting the detection results from image space to a predicted ground plane (i.e. the transform space). The main advantage of this strategy is that the motion model in the ground plane is more in line with the actual situation and the trajectories with intervals could be associated to reduce the fragmentation caused by occlusion.

In this work, we use a Kalman filter to predict the motion of objects in a transform space and introduce a dynamic appearance feature into the tracker. In the transform space, the physical ellipse intersection-over-union (IOU) instead of pixel-level bounding box IOU is used to generate the association measurements. The appearance features model trained with an adaptive weighted triplet loss is used for the re-identification of the tracker. Compared with the model with cross-entropy loss, the proposed model can better distinguish the pedestrians with similar appearances. After computing a matching score, the cost matrix is solved by satisfying the one-to-one association constraint. Our extensive experiments have revealed that using transform space instead of image space can avoid losing tracks of true positives with low confidence. Extensive experiments demonstrate that using the locations in the transform space can avoid losing tracks of true positives with few detections of low confidence caused by occlusions and noise. Our implementation is publicly available at
https://github.com/Jeasonlee313/MOT_predict_by_physical.

The main contributions of this paper are the following:
(1) A tracking filter based on predicted physical location, which extends the tracking algorithm to predict the locations of the pedestrians in an estimated ground plane;
(2) Use physical ellipse intersection-over-union (IOU) instead of pixel-level bounding box IOU to generate the association measurements;
(3) A cosine metric learning model trained with the triplet loss, which further improves the robustness of MOT. Compared with the model trained using cross-entropy loss, the new model can better distinguish the pedestrians with similar appearances.

2. RELATED WORK

Using spatial information. Most online multiple object trackers include a Bayesian tracking process, which predicts the state of each track using previously assigned observations. In this way, the likelihood between the track and the observation is calculated to form a cost matrix for data association. During tracking, the targets are captured by matching the data distribution in the incoming frame. In the SORT method (Bewley et al., 2016), the Kalman filter estimates the current states of the trajectories from detections and their previous states, which can significantly improve the association efficiency. Yoon et al. (2021) extracted the motion context of multiple objects in the assignment problems. Milan et al. (2017) present a novel recurrent neural

* The corresponding author: minglei_li@nuaa.edu.cn.

network-based (RNN) multi-target tracker using bounding-box features to define the cost matrix. The trackers using only the motion models have short-term memories, so they are sensitive to occlusion.

Using appearance features. Previous works have revealed that using appearance features can improve the performance of a tracker (Kim et al., 2015; Yoon et al., 2021; Milan et al., 2017). Wojke et al. (2017) integrate appearance information to improve SORT performance, which can track objects through longer periods of occlusions, effectively reducing the number of identity switches (IDSW). Kim et al. (2018) extract appearance features through the Siamese network and associate those features using a deep long short-term memory (LSTM). Then, among all the features, the most probable one is used for tracking objects. Ristani et al. (2018) used the residual network with an adaptive weighted triplet loss for appearance modeling. Bae et al. (2018) use the Siamese network with a triplet loss for appearance modeling and adaptive training of the network during tracking. In addition to pedestrian tracking, appearance features are also used in vehicle tracking and re-identification. Lin et al. (2019) propose a joint representation of these pyramidal features used for learning discriminative features for vehicle Re-ID. Yang et al. (2021) proposed to train the deep feature network in an end-to-end manner. Bergmann et al. (2019) exploited the bounding box regression to predict the position of an object as a straightforward re-identification. The deep learning-based methods have significantly improved traditional tracking performance (Mutiple Object Tracking Benchmark, 2020) and have become an indispensable part of the tracking pipeline.

Besides, the different variations of the Hungarian algorithm (Kuhn, 1955) have been used in some trackers for data association, showing competitive performances. The Hungarian algorithm treats the assignment problem as a special case of the transportation problem. The assignment problem is solved by finding the min-cost flow, so it can be operated with the cost matrix (Bewley et al., 2016; Bae et al., 2018; Sadeghian et al., 2017). We adopt the Hungarian algorithm because of its simplicity and competitive performance.

Multi-view object tracking. Object detected in multi-view images can use cross validation to reduce the error correlation of trajectories. The key for multi-view object tracking is to find cross-view correspondences. Jiang et al. (2007) used integer programming for data association. Some networks are developed for multi-view images, such as network flow (Wu et al., 2009) and multi-commodity network (Shitrit et al., 2014). In this paper, we propose an association strategy, which is suitable not only for single camera video, but also for multi-view videos. By incorporating the motion models in the ground space and deep appearance attributes, the algorithm has a competing performance.

3. METHODOLOGY

Figure 1 shows the pipeline of the proposed tracking-by-detection MOT method. The first is to detect the pedestrians in the image space and convert the positions of the objects to the estimated ground plane. Meanwhile, the trackers predict the locations of the previous trajectories using the motion models. The association matrix is calculated by combining the two kinds of costs based on physical locations and the deep features. Based on the association matrix, the Hungarian algorithm associates the detections with the trackers. Finally, the routine updates every tracker for the next frame.

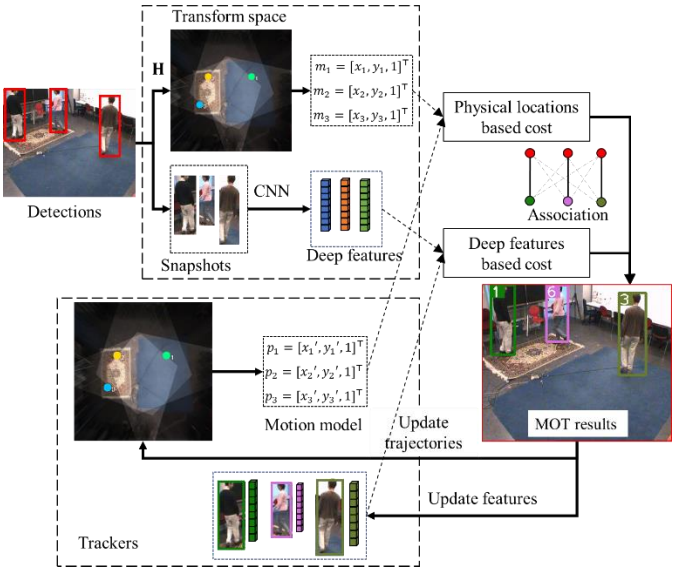

Figure 1. The proposed MOT diagram.

3.1 Tracking problem statement

Given a set of detected objects $\mathbf{O} = \{\boldsymbol{o}_1, \dots, \boldsymbol{o}_n\}$ in a frame, each object is calculated with a structure $\mathbf{o}_i = (t_i, u_i, v_i, w_i, h_i, class_i, conf_i)$, where t_i denotes the timestamp; (u_i, v_i) denotes the left-top image coordinates of the object's bounding box and (w_i, h_i) is its width and height; $class_i$ represents the class of the detection; and $conf_i$ indicates the likelihood of the detection.

The goal of MOT is to obtain a set of trajectories $\{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_m\}$ that best explains the motion observations, where $\mathbf{T}_i = \{\mathbf{o}_i^1, \dots, \mathbf{o}_i^k\}$ is a single trajectory composed of a set of frame-ordered detections for object $\boldsymbol{o}_i$; k denotes the frame index. The data association task is modeled by calculating a cost matrix $\mathbf{C} = \{c_{ij} \,|\mathbf{o}_i^t, \mathbf{o}_j^{t-1} \in \mathbf{O}\}$, where c_{ij} denotes the association cost between the i-th object $\mathbf{o}_i^t$ in the t-th frame and the j-th object $\mathbf{o}_j^{t-1}$ in the $(t-1)$-th frame.

3.2 Prediction of physical locations in transform space

We need a homography matrix $\mathbf{H}$ to convert the detected objects from the image space to an estimated ground space. The physical coordinates of objects are calculated as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{H}$ can be calculated if the camera matrix has been calibrated under a world coordinate framework. In case the camera is not calibrated, $\mathbf{H}$ can be calculated by applying the direct linear transform algorithm based on a set of extracted corresponding points on the image plane and the ground plane. In our study, we extract vertices of some rectangles on the ground (tiles and lane lines), and these vertices and their image coordinates are used to calculate $\mathbf{H}$. In practical applications, engineers can easily calibrate the camera when they install the devices. Therefore, the homography matrix relationship between the images and the ground can be obtained at the beginning.

On the rectified ground space, we define a linear motion model to approximate the inter-frame displacement of each object. It is assumed that the movement of each object is independent of others. The state space of moving objects is modeled as:

$$\mathbf{s} = [x, y, \dot{x}, \dot{y}]^{\top}, \quad (2)$$

where (x, y) represents an object's location state in transform space, and $(\dot{x}, \dot{y})$ denotes the corresponding velocity state of the object. We assume that the walking speed of a person is approximately constant, so the step size can be estimated using two interval frames. An object position $[u, v]^{\top}$ in image space can be directly observed after detection, then its position on the ground is calculated using equation (1).

After the homography transformation, we obtain the physical location of each object in a rectified ground space. Figure 2 shows such an example, in which the ground plane can be seen as a rectified map of the scene in a top view.

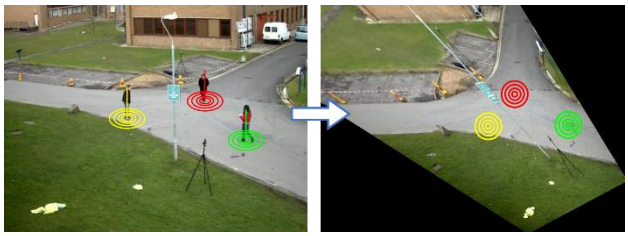

Figure 2. Predicted locations of people in the image space (left) and the transform space (right).

When a detection is associated with a tracking object, it updates the velocities of the object state via a Kalman filter. If an object is lost in a frame, its state will maintain the linear velocity model.

3.3 Deep appearance Feature

When an object is occluded for a period of time, subsequent Kalman filter might fail to associate the object location. We add the appearance features into the data associate model for the task of Re-Identification (Re-ID). To get the appearance feature of the detected object, we trained a CNN model offline for the deep metric learning. Specifically, the feature network for Re-ID is trained on two public datasets, namely Market-1501 (Zheng et al., 2015) and Duke MTMC-reID (Ristani et al., 2016).

The CNN for deep feature extraction uses a ResNet50 (He et al., 2016) and follows its pool5 layer by a dense layer with 1024 units, batch normalization, and ReLU. The dense layer with normalization yields a 512-dimensional appearance descriptor. The initial weights of this model are pre-trained on the ImageNet (Deng et al., 2009). To improve generalization ability, we use the triple loss function for iterative convergence. Given an anchor sample x_a, positive samples $x_p \in \mathbf{P}(a)$, and negative samples $x_n \in \mathbf{N}(a)$, the triplet loss is written as:

$$L_{\text{triple}} = [m + \sum_{x_p \in \mathbf{P}(a)} w_p d(x_a, x_p) - \sum_{x_n \in \mathbf{N}(a)} w_n d(x_a, x_n)]_+, \quad (3)$$

where m is the inter-person separation margin; $d(\cdot,\cdot)$ denotes the cosine distance of appearance, and $[\cdot]_+ = max(0,\cdot)$. The main architecture is shown in Table 1.

Name	Patch Size/Stride	Output Size
Conv 1	3*3/1	32*128*64
Conv 2	3*3/1	32*128*64
Max Pooling 3	3*3/2	32*64*32
Residual 4	3*3/1	32*64*32
Residual 5	3*3/1	32*64*32
Residual 6	3*3/2	64*32*16
Residual 7	3*3/1	64*32*16
Residual 8	3*3/2	128*16*8
Dense 9		1024
Batch and l_2 normalization		1024
Dense 10 and l_2 normalization		512

Table 1. Overview of the CNN network architecture

To construct the deep feature model, we use the idea of PK batches introduced by Hermans et al. (2017), in which K sample images for each of P identities are used in each batch. During a training epoch, each identity is selected in its batch in turn, and the remaining P-1 batch identities are sampled randomly. K samples are also selected randomly.

For the training, we set $P = 16$, $K = 4$, and $m = 1$. The learning rate is set to 10^{-4} for the first 25000 iterations, and it decays to 10^{-5} at iteration 35000. The weights w_p and w_n are reformulated as:

$$w_p = {}^{1}/_{N_p}, w_n = {}^{1}/_{N_n}, \quad (4)$$

where N_p and N_n is the total number of positive and negative samples.

3.4 Data association

The association matrix $\boldsymbol{C}$ aims to associate the detected objects $\{\mathbf{o}_i\}$ with the existing trackers $\{\mathbf{T}_j\}$. The association cost is defined as:

$$c_{i,j} = \lambda d_p(i,j) + (1-\lambda) d_a(i,j), \quad (5)$$

where $d_p(i,j)$ denotes the location distance in the transform space and $d_a(i,j)$ denotes the appearance feature distance. The parameter lambda is used to balance the impact of the location term and the appearance term. A simple setting for the parameter lambda is 0.5. If we have a more accurate location information, we can set lambda larger than 0.5, and the upper limit of lambda is 1, and vice versa.

$d_p(i,j)$ is computed as the physical ellipse IOU distance between the current detected objects and all predicted locations from existing trajectories. To incorporate motion information, we use the Euclidean distance between the predicted Kalman states and the newly arrived measurements:

$$d_p(i,j) = \left\| [x,y]_{\mathbf{o}_i} - [x,y]_{\mathbf{T}_j} \right\|_2, \quad (6)$$

where $[x,y]_{\mathbf{o}_i}$ and $[x,y]_{\mathbf{T}_j}$ represent the physical location of the current detection $\mathbf{o}_i$ and the last physical location of the confirmed trajectory $\mathbf{T}_j$ respectively.

Meanwhile, for each detected object we compute an deep feature descriptor $\boldsymbol{f}_i$ subject to $||\boldsymbol{f}_i||_2 = 1$, using the trained CNN model. The trajectory $\mathbf{T}_j$ holds a feature gallery $\mathbf{R}_j = \{\boldsymbol{f}_j^{(k)}\}_{k=1}^{100}$, each of which is an associated feature from previous 100 detections. The distance $d_a(i,j)$ is calculated between every appearance feature in the gallery and that of the detected object:

$$d_a(i,j) = \min\left\{1 - \boldsymbol{f}_i^{\top} \boldsymbol{f}_j^{(k)} \mid \boldsymbol{f}_j^{(k)} \in \mathbf{R}_j\right\}. \quad (7)$$

where $\boldsymbol{f}_i$ denotes the appearance descriptor of detection $\mathbf{o}_i$, and $\boldsymbol{f}_j^{(k)}$ denotes the k-th appearance vector in the gallery of a confirmed tracked object $\mathbf{o}_j$. By sorting the appearance distances of all detection and object pairs, the most similar detection and object pair can be selected as the one with the smallest appearance distance. Using appearance features is particularly useful to recover the blocked objects after long-term occlusions.

By plugging the motion distance defined in Equation (6) and the appearance distance defined in Equation (7) into Equation (5), we obtain the combined association cost matrix.

So far, we obtain the association cost matrix $\boldsymbol{C}$, where each element is calculated by Equation (5). To remove unreasonable pairs, once the location distance is larger than the threshold $gate_p$ or the feature distance is larger than 0.2, the element $c_{i,j}$ will be set to infinite. $gate_p$ is empirically set to the 3 times step size. Finally, the Hungarian algorithm is used to find the minimum loss assignment between the potential tracks and the current objects. The matching steps are given in Algorithm 1.

Algorithm 1. Cascading matching

Input: Trajectories $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n\}$ and detections $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_m\}$

1: Compute the cost matrix of physical location $\mathbf{D}_p = [d_p(i,j)]$ using Equation (6)

2: Compute the cost matrix of appearance feature $\mathbf{D}_a = [d_a(i,j)]$ using Equation (7)

3: Initialize the set of matches $\mathbf{M} = \emptyset$

4: Initialize the set of unmatched detections $\mho = \mathbf{O}$

5: **for** $d_p(i,j)$, $d_a(i,j)$ **in** $\mathbf{D}_p$, $\mathbf{D}_a$:

6: **if** $d_p(i,j) > gate_p$ || $d_a(i,j) > 0.2$:

7: $d_a(i,j) = \infty$

8: **end if**

9: **end for**

10: $\mathbf{M} = \{(\mathbf{T}_i, \mathbf{o}_j)\} \leftarrow$ Hungarian_assignment$(\mathbf{T}, \mho, \mathbf{D}_a)$

11: $\mho = \mho \setminus \{\mathbf{o}_j | \mathbf{o}_j \text{ in } \mathbf{M}\}$

12: $\mathbf{T} = \mathbf{T} \cup \mho$

13: return: **M** and new **T**

4. EXPERIMENTS

4.1 Dataset

The method has been tested on MOTChallenge benchmarks, including MOT15, MOT16, and MOT17 (Leal-Taix éet al., 2015; Milan et al., 2016). The dataset consists of several challenging pedestrian tracking sequences, with frequent occlusions, varying perspectives, crowded scenes, and camera movements. The homography matrix of each dataset is calculated by manually selecting the image points of rectangle patterns on the ground. These benchmarks provide the ground truth of trajectories. The datasets contain image sequences with varying viewing angles, sizes, numbers of objects, and frame rates.

We also carried out a tracking evaluation on the EPFL dataset (Xu et al., 2017), which contains two multi-view videos and the homography matrices between the cameras and the grounds.

4.2 Results

We exploited the trained YOLOv5 model to provide pedestrian detection results for tracking, and some other detectors are also applicable. With an Nvidia GeForce GTX 1660 Ti GPU, one forward pass of 16 bounding box region detection takes approximately 25 ms, which makes it suitable for online tracking. The networks are then trained with the Market-1501 and the DukeMTMC-reID datasets. Statistics of the performances are given in Table 2.

Dataset	Rank-1	Rank-5	Rank-10	mAP
Market-1501	0.8625	0.9426	0.9608	0.7026
DukeMTMC-reID	0.7639	0.8712	0.9062	0.5937

Table 2. The accuracy of the network re-ID

In Figure 3, we show an example frame from the EPFL dataset. The images are acquired in a laboratory with fixed cameras. The people in the images block each other frequently. We can easily distinguish different people taking advantage of the transform ground space. In the top view, every position in state space is presented by the circle centre. The different radii denote the covariance of the states.

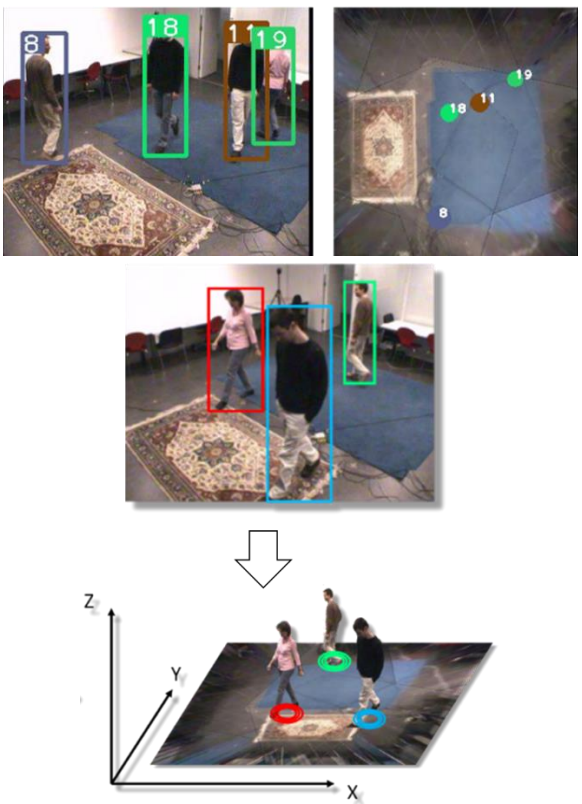

Figure 3. MOT results using the transform space in a laboratory.

Figure 4 shows two open street scenes with more people than that in the indoor scenes. The proposed method produced stable identities. After the pedestrians appeared to block each other, the proposed method preserved the identity after the people separated in most cases. It is worth noting that there are fewer identity switches in the outdoor scene than in the indoor scenes, which is because the appearance CNN model is trained with outdoor images.

In street scenes, the identities of far pedestrians switch more frequently. It can also be observed that when the pitch angle of the camera is small, the closer the person is to the camera, the more accurate the predicted position is. Our method has relatively stable tracking results for pedestrians whose physical locations

can be accurately estimated through the homography transformation. In all these tests, our method demonstrated promising performance for the images taken from a looking-down view.

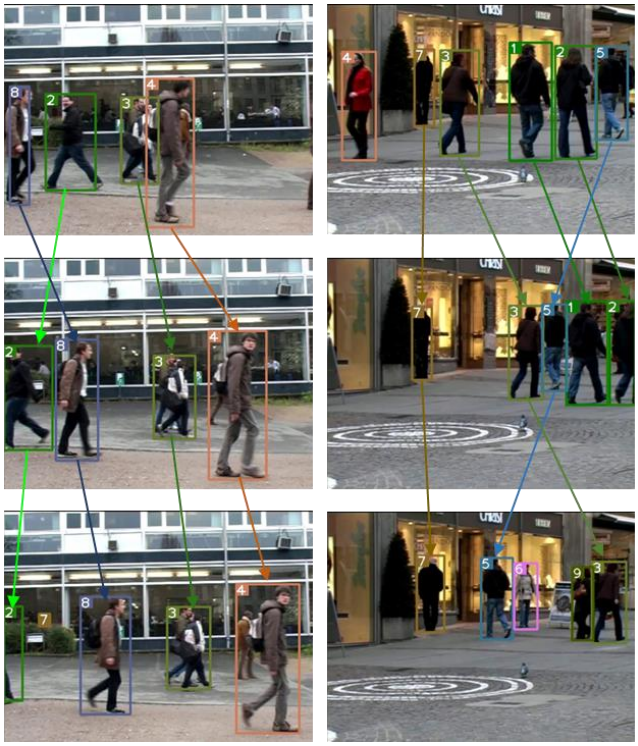

Figure 4. The tracking results on street video sequences

In MOTChallenge videos, there are cases where the cameras are moving and their view directions are kept approximately parallel to the ground. There are two street views captured by moving cameras in Figure 5. In these cases, the space transformation process could enlarge the estimation errors of object locations, hence it lowers the performance of the proposed method. This is a limitation of the proposed algorithm.

The proposed method prefers an overhead camera angle, which will provide a more accurate transformation space. Although our method may not be applicable in such scenarios of moving cameras, fortunately, video based security and traffic monitoring in most cities can meet the conditions of overhead camera angles.

Figure 5. Two street scenes with moving cameras

4.3 Comparison

The comparison is conducted with some most relevant trackers, i.e. the state-of-the-art methods, namely SORT (Bewley et al., 2016), mfi_tst (Yang et al., 2021), SLA_Track (Bergmann et al., 2019), and Tracker++ (Mutiple Object Tracking Benchmark, 2020) on the MOT16 and MOT17 datasets. The performance of our tracker and the comparison with other state-of-the-art trackers are shown in Table 3.

Data	Name	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	IDSW↓	Frag↓
MOT15	**Ours**	47.9	52.3	**54.9**	**264**	**63**	**244**	**738**
	SORT (Bewley et al., 2016)	33.4	40.4	21.1	84	223	1001	1764
	mfi_tst (Yang et al., 2021)	**49.2**	**58.7**	41.5	210	176	912	1397
	SLA_Track (Bergmann et al., 2019)	47.0	57.9	43.0	163	196	558	1580
	Tracker++ (Mutiple Object Tracking Benchmark, 2020)	46.6	47.6	37.6	131	201	1290	1702
MOT16	**Ours**	57.7	42.9	46.3	167	222	**321**	1095
	mfi_tst (Yang et al., 2021)	59.9	58.7	**46.9**	183	234	616	**1050**
	SLA_Track (Bergmann et al., 2019)	**60.6**	**59.5**	46.8	**184**	**221**	643	1171
	Tracker++ (Mutiple Object Tracking Benchmark, 2020)	56.2	54.9	44.6	157	272	617	1069
MOT17	**Ours**	57.8	42.7	46.8	543	726	**933**	**3198**
	SORT (Bewley et al., 2016)	43.1	39.8	34.0	295	997	4852	7127
	mfi_tst (Yang et al., 2021)	**60.1**	58.8	47.2	**612**	**699**	2065	3829
	SLA_Track (Bergmann et al., 2019)	59.7	**63.4**	**49.1**	566	732	1647	3819
	Tracker++ (Mutiple Object Tracking Benchmark, 2020)	56.3	55.1	44.8	498	831	1987	3763

Table 3. Comparison of our experiment results with other method results on MOTChallenge benchmarks

The evaluation metrics include the multi-object tracking accuracy (MOTA), the ratio of correctly identified detection over the average number of ground-truth and computed detection (IDF1), higher order tracking accuracy (HOTA), the ratio of mostly tracked targets (MT), the ratio of mostly lost targets (ML), the number of ID switches (IDSW) and the number of fragments (Frag). The upper arrow ↑ means that larger value of this metric shows better performance. The down arrow ↓ means that smaller value of this metric shows better performance. We explain these metrics in the **APPENDIX** at the end of the paper.

As we can see in Table 3 that the proposed method decreases the value of IDSW in most cases. It demonstrated that our method has an advantage in trajectory preservation. The proposed method can recover occluded trajectories of targets with varying scales, as it uses a physical ellipse IOU distance instead of pixel bounding box IOU distance. Specifically, when the target is occluded in the image, the position in the ground space remains within a reasonable range. This allows recovery of the target after occlusion.

We observe that the proposed method also reduces the number of trajectory fragments (Frag). It revealed that physical locations are more suitable for tracking objects when they are partially occluded. Considering all factors on the MOTA score, using a larger confidence threshold to the detections can potentially increase MOTA values. A larger number of false positives can decrease tracking accuracy. In sum, the proposed method is a strong competitor to other online tracking frameworks in the aspect of the other metrics.

5. CONCLUSION

In this paper, we introduce an idea to predict object trajectories in a transform physical space for multi-object tracking tasks. The proposed algorithm has two core techniques: (1) a Kalman filter for predicting locations of pedestrians by transforming their positions from the image space to the ground space, which provides a reference plane for object association in multi-view tracking; and (2) an appearance feature deep network for re-identification of blocked pedestrians. The proposed method is able better distinguish people in the crowd and track through longer periods of occlusion. Extensive experiments have shown that the predicted locations are effective for multi-object tracking, in particular in reducing metrics of the number of identity switches and the fragments. As a future work, we plan to integrate the proposed method into an embedded hardware system and apply it to solve the problem of traffic monitoring.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant: 41801342, and supported by the Fundamental Research Funds for the Central Universities，NO. NZ2020008.

REFERENCES

[1] Avidan, S., 2007. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2), 261-271.

[2] Bae, S.H., Yoon, K.J., 2018. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans Pattern Anal Mach Intell*, 40(3): 595-610.

[3] Bergmann, P., Meinhardt, T., Leal-Taixé, L., 2019. Tracking without bells and whistles. *In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 941-951.

[4] Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016. Simple online and realtime tracking. *In: 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464-3468.

[5] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

[6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F., 2009. ImageNet: A large-scale hierarchical image database. *In: 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

[7] Dicle, C., Camps, O.I., Sznaier, M., 2013. The way they move: Tracking multiple targets with similar appearance. *In: 2013 IEEE International Conference on Computer Vision*, pp. 2304-2311.

[8] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *In: 2016 IEEE Conference on Computer Vision and Pattern*, pp. 770-778.

[9] Hermans, A., Beyer, L., Leibe, B., 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

[10] Jiang, H., Fels, S., Little, J., 2007. A linear programming approach for multiple object tracking. *In: 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

[11] Kim, C., Li, F., Alotaibi, M., Rehg, J., 2021. Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking. *In: IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9553-9562.

[12] Kim, C., Li, F., Ciptadi, A., Rehg, J.M., 2015. Multiple hypothesis tracking revisited. *In: 2015 IEEE International Conference on Computer Vision*, pp. 4696-4704.

[13] Kim, C., Li, F., Rehg, J.M., 2018. Multi-object tracking with neural gating using bilinear LSTM. *In: 15th European Conference on Computer Vision*, pp. 200-215.

[14] Kuhn, H.W., 1955. The Hungarian method for the assignment problem. *Nav Res Logist Quarterly*, 2(1-2): 83-97.

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

[16] Lin, X., Zeng, H., Hou, J., Zhu, J., Chen, J., Ma, K.K., 2019. Vehicle re-identification using joint pyramid feature representation network. *In: International Conference on Internet of Things as a Service*, pp. 527-536.

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

[18] Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K., 2017. Online multi-target tracking using recurrent neural networks. *In: Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4225-4232.

[19] Mutiple Object Tracking Benchmark, 2020. SLA_public: Spatial-Location Aware Tracker [Internet]. 2020 [updated 2020 Dec 13; sited 2021 Nov 24]. Available: https://motchallenge.net/method/MOT=3748&chl=2.

[20] Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell*, 39(6), 1137-1149.

[21] Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I., 2015. Joint probabilistic data association revisited. *In: 2015 IEEE International Conference on Computer Vision*, pp. 3047-3055.

[22] Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. *In: 14th European Conference on Computer Vision (ECCV)*, pp. 17-35.

[23] Ristani, E., Tomasi, C., 2018. Features for multi-target multi-camera tracking and re-identification. *In: 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6036-6046.

[24] Ross, D.A., Lim, J., Lin, R.S., Yang, M.H., 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1), 125-141.

[25] Sadeghian, A., Alahi, A., Savarese, S., 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *In: 16th IEEE International Conference on Computer Vision (ICCV)*, pp. 300-311.

[26] Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P., 2014. Multicommodity network flow for tracking multiple people. *IEEE Trans Pattern Anal Mach Intell*, 36(8), 1614-1627.

[27] Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. *In: 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645-3649.

[28] Wu, Z., Hristov, N.I., Hedrick, T.L., Kunz, T.H., Betke, M., 2009. Tracking a large number of objects from multiple views. *In: 12th IEEE International Conference on Computer Vision*, pp. 1546-1553.

[29] Xu, Y., Liu, X., Qin, L., Zhu, S.C., 2017. Cross-view people tracking by scene-centered spatio-temporal parsing. *In: 31st AAAI Conference on Artificial Intelligence*, pp. 4299-4305.

[30] Yang, J., Ge, H., Yang, J., Tong, Y., Su, S., 2021. Online multi-object tracking using multi-function integration and tracking simulation training. *Applied Intelligence*, 19(1): 1-21.

[31] Yoon, Y.C., Kim, D.Y., Song, Y., Yoon, K., Jeon, M., 2021. Online multiple pedestrians tracking using deep temporal appearance matching association. *Information Sciences*, 561(1): 326-351.

[32] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q., 2015. Scalable person re-identification: A benchmark. *In: 2015 IEEE International Conference on Computer Vision*, pp. 1116-1124.

APPENDIX

The performance of our tracker and the comparison with other state-of-the-art trackers are shown in Table 3. In the following, we explain the metrics used in our experiments.

- MOTA is used to statistic the error accumulation in tracking. It is combined with false-negative detections, false-positive detections, and identity switch. Given the number of false-negative detection FN_t in the frame t, the number of false-positive detection FP_t, and the number of identity switch IDSW_t, the mathematical form is written as:
$$\mathrm{MOTA} = 1 - \frac{\sum_t(\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{IDSW}_t)}{\sum_t g_t},$$
where g_t is the number of ground truth detection in the frame t.

- IDF1 is the ratio of the number of correct identities to the average number of detection in each frame. The form of IDF1 is:
$$\mathrm{IDF1} = 2\frac{\mathrm{IDP} \cdot \mathrm{IDR}}{\mathrm{IDP} + \mathrm{IDR}}.$$
IDP is the precision of the object identities, and IDR is the recall of the object identities. The IDP is formulated as:
$$\mathrm{IDP} = \frac{\mathrm{ID}_{\mathrm{TP}}}{\mathrm{ID}_{\mathrm{TP}} + \mathrm{ID}_{\mathrm{FP}}},$$
where $\mathrm{ID}_{\mathrm{TP}}$ is the number of the correct identities and $\mathrm{ID}_{\mathrm{FP}}$ is the number of the wrong identities:
$$\mathrm{IDR} = \frac{\mathrm{ID}_{\mathrm{TP}}}{\mathrm{ID}_{\mathrm{TP}} + \mathrm{ID}_{\mathrm{FN}}},$$
where $\mathrm{ID}_{\mathrm{FN}}$ is the number of the identities which are not detected and identified.

- HOTA is the average of detection accuracy and association accuracy. It can be formulated as:
$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\substack{\alpha=0.05 \\ \alpha+=0.05}}^{0.95} \mathrm{HOTA}_\alpha$$
$$= \frac{1}{19} \sum_{\substack{\alpha=0.05 \\ \alpha+=0.05}}^{0.95} \sqrt{\mathrm{DetA}_\alpha \cdot \mathrm{AssA}_\alpha},$$
where DetA is the detection accuracy, and AssA is the association accuracy. The form of DetA is written as:
$$\mathrm{DetA} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}},$$
where TP is the number of correct detection, FP is the number of wrong detection and FN is the number of objects which are not detected.

- The mathematical form of AssA is written as:
$$\mathrm{AssA} = \frac{1}{C}\sum_C \frac{\mathrm{TPA}}{\mathrm{TPA} + \mathrm{FNA} + \mathrm{FPA}},$$
where TPA is the number of correct associations in a trajectory; FPA is the number of wrong associations; FNA is the number of fail detection in one trajectory; and C is the total number of trajectories. In the HOTA, there is an angle mark α, which is used as a threshold to determine if one detection is a TP or FP (TPA or FPA). Given a detection bounding-box D and a ground-truth bounding-box G, the α is formulated as:
$$\alpha = \frac{\mathrm{D} \cap \mathrm{G}}{\mathrm{D} + \mathrm{G} - \mathrm{D} \cap \mathrm{G}}.$$
In DetA, if the α is large than the threshold, the detection is a TP. In AssA, calculated with the object of the previous frame in the same trajectory, if the α is large than the threshold, the association is a TPA. The reverse held as well.

- MT represents the number of the trajectories which overlap more than 80% compared with ground-truth; ML represents the number of the trajectories which overlap less than 20% compared with ground-truth.

- IDSW is the summary of the number of transitions from one identity to another on a trajectory in the tracking result.

- Frag is the summary of the number of fragments in one trajectory in the tracking result.