Sampling-Resilient Multi-Object Tracking

Zepeng Li[1], Dongxiang Zhang[2*], Sai Wu[2], Mingli Song[2], Gang Chen[2]

[1] The State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2] College of Computer Science and Technology, Zhejiang University
{lizepeng,zhangdongxiang,wusai,brooksong,cg}@zju.edu.cn

Abstract

Multi-Object Tracking (MOT) is a cornerstone operator for video surveillance applications. To enable real-time processing of large-scale live video streams, we study an interesting scenario called down-sampled MOT, which performs object tracking only on a small subset of video frames. The problem is challenging for state-of-the-art MOT methods, which exhibit significant performance degradation under high frame reduction ratios. In this paper, we devise a sampling-resilient tracker with a novel sparse-observation Kalman filter (SOKF). It integrates an LSTM network to capture non-linear and dynamic motion patterns caused by sparse observations. Since the LSTM-based state transition is not compatible with the original noise estimation mechanism, we propose new estimation strategies based on Bayesian neural networks and derive the optimal Kalman gain for SOKF. To associate the detected bounding boxes robustly, we also propose a comprehensive similarity metric that systematically integrates multiple spatial matching signals. Experiments on three benchmark datasets show that our proposed tracker achieves the best trade-off between efficiency and accuracy. With the same tracking accuracy, we reduce the total processing time of ByteTrack by $2\times$ in MOT17 and $3\times$ in DanceTrack.

Introduction

Multi-object tracking (MOT) aims at detecting and tracking moving objects from video clips or live streams, while maintaining a unique identifier for each object. Massive research efforts have been devoted into this domain with fruitful progress. The proposed trackers have witnessed great success in numerous applications, such as smart video surveillance (Xu et al. 2018; Xiao et al. 2023), traffic monitoring (Tian, Lauer, and Chen 2020; Zhang et al. 2023), customer behavior analysis (Merad et al. 2016) and sports analytics (Lu et al. 2013).

In this paper, we study an interesting scenario called down-sampled MOT, which performs object tracking *only upon a small subset of video frames*. Since the processing time of MOT is positively correlated with the number of sampled frames, the task has the potential to achieve an ideal trade-off between tracking efficiency and accuracy, and thus is particularly useful in handling *large-scale video streams with limited computing resources*. In other words, with the same amount of GPU cards, a tracker that works well on a higher frame reduction ratio can support MOT on many more video streams simultaneously.

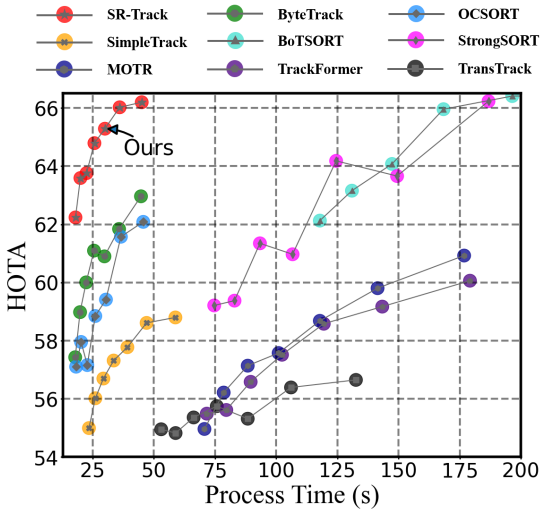


Figure 1: We plot the trade-off between tracking efficiency and accuracy for mainstream MOT methods, by adjusting the sampling rate in MOT17. When the video frame reduction ratio is high, the tracking accuracy of existing methods declines to an impractical level. Our SR-Track is the only approach to achieve promising HOTA with very small processing time (e.g., around 30 seconds to reach HOTA$=66$).

Down-sampled MOT is challenging because the motion dynamics increase and the patterns become non-linear and more difficult to capture. In addition, the data association strategy such as IoU that works well in dense frames fails in the scenario of sparse frames. Therefore, directly applying state-of-the-art MOT methods on the down-sampled frames would result in significant performance degradation. As shown in Figure 1, we report the trade-off between efficiency and accuracy in terms of HOTA, by adjusting different sampling rates. When the video frame reduction ratio is high, it's indeed that the processing time can be significantly reduced. However, the tracking accuracy also declines to an impractical level. Detailed performance analysis

*Corresponding author

of these trackers will be presented in the experiment. These findings lead to the conclusion that existing MOT solutions are not sampling-resilient.

To devise a more sampling-resilient MOT model, we propose SR-Track with a novel variant of Kalman filter (KF) for accurate motion prediction under sparse observations. Specifically, we replace the linear motion assumption in conventional KF with a LSTM network to capture the nonlinear motion patterns with high dynamics. Since Gaussian covariance matrix of traditional KF is not compatible with the LSTM-based state transition, we introduce new noise estimation mechanisms based on the Bayesian neural networks and derive the optimal Kalman gain to minimize the discrepancy between the true state and our estimated state. Furthermore, to robustly associate detected bounding boxes under enlarged temporal gaps, we propose a comprehensive similarity metric that integrates multiple matching clues, including overlap, center point distance and aspect ratio of the bounding boxes.

Experiments are conducted on three benchmark datasets, among which DanceTrack is the most challenging due to frequent crossover and diverse body gestures. The results show that our proposed tracker outperforms most trackers in terms of both efficiency and accuracy. For the real-time trackers that can achieve similar FPS, our SR-Track exhibits clearly higher accuracy. Compared with ByteTrack, the state-of-the-art real-time tracker, we can further reduce the total processing time by $2\times$ in MOT17 and $3\times$ in DanceTrack, with the same level of tracking accuracy.

Related Work

We divide existing multi-object trackers into two categories, namely *tracking-by-detection* and *joint-detection-and-tracking*, according to whether its object detection network is a separate module or requires joint training.

Tracking-by-Detection Methods

SORT (Bewley et al. 2016), DeepSORT (Wojke, Bewley, and Paulus 2017), OC-SORT (Cao et al. 2023), StrongSORT (Du et al. 2022), BoT-SORT (Aharon et al. 2022) and ByteTrack (Zhang et al. 2022) are representative tracking-by-detection methods. They treat MOT as a pipeline of object detection and association, and optimize each module separately. Firstly, an existing object detector is adopted to locate objects in each video frame. Early trackers (e.g., SORT and DeepSORT) use Faster RCNN (Ren et al. 2015) as the default detector, which is replaced by YOLOX (Ge et al. 2021) in recent trackers. Secondly, an object association mechanism is designed to connect these detected objects into tracklets. Coherence in motion pattern and similarity in visual appearance are two important factors in object association. As to motion pattern, almost all the tracking-by-detection methods adopt Kalman filter for future position estimation. A detected object is assigned to an existing tracklet if its spatial matching distance (e.g., IoU distance) between the two bounding boxes is small. As to visual similarity, DeepSORT (Wojke, Bewley, and Paulus 2017), StrongSORT and BoT-SORT integrate appearance features into the tracker, which requires additional computation cost to derive visual embedding. The spatial matching score and appearance similarity are combined as the final association metric.

Among these trackers, ByteTrack (Zhang et al. 2022) achieves the best trade-off between efficiency and accuracy. It discards visual similarity and only relies on spatial matching to save computation cost. As a compensation, it introduces a robust association strategy to take into account the detected objects with low confidence.

Joint-Detection-and-Tracking Methods

JDE (Wang et al. 2020) is a pioneering work that allows object detection and appearance embedding to be learned in a single network. Compared with DeepSORT, its low-level visual features can be shared by the detector and embedding model to avoid re-computation cost. However, the shared network in JDE is biased towards the detector task and unfair to the ReID (Ye et al. 2022; Li et al. 2023) task. To resolve the competition issue, CSTrack (Liang et al. 2020) devises a cross-correlation network to learn task-dependent representations. RelationTrack (Yu et al. 2023) presents global context disentangling (GCD) to decouple the learned features in the two tasks. FairMOT (Zhang et al. 2021) adopts another way by implementing two homogeneous branches for the detection and ReID tasks, rather than performing them in a two-stage cascaded style. SimpleTrack (Li et al. 2022) is designed to mitigate the issue of object occlusion and presents a new association matrix that combines embedding cosine distance and Giou distance of objects. Note that these works still rely on an online data association strategy based on Kalman filter and appearance similarity to connect the detected boxes.

To push forward the idea of joint training, the following trackers attempt to further incorporate the estimation of inter-frame object motion in the training framework. In other words, Kalman filter is discarded. CenterTrack (Zhou, Koltun, and Krähenbühl 2020) and TransCenter (Xu et al. 2021) predict the object offset between adjacent frames to facilitate object tracking. The models are trained to minimize the regression loss of the object offset between adjacent frames. TransCenter (Xu et al. 2021) proposes a Transformer-based architecture, together with dense but non-overlapping representations for detection, to globally and robustly infer the offset of objects' centers. For GSDT (Wang, Kitani, and Weng 2021) and FUEFT (Shan et al. 2020), motion and appearance features are fed into a graph neural network (GNN) to predict the association matrix of tracklets and detected bounding boxes. TransTrack (Sun et al. 2020) utilizes the attention mechanism to model the detection and tracking, and outputs the predicted bounding box of tracked objects. Recently, TrackFormer (Meinhardt et al. 2022) adopts the concept of track queries and employs the attention mechanism to track the objects in an autoregressive fashion. In the current stage, these trackers are computation expensive to achieve high accuracy and not suitable to support large-scale video streams.

Methodology of SR-Track

Before we present our SR-Track, we first briefly review Kalman filter (KF), which has been widely adopted in object tracking to estimate object location in the subsequent frame. It works as an efficient recursive filter with the stages of prediction and update. KF requires small computational power and provides satisfactory estimation, rendering it well-suited for real-time analysis.

Let $\hat{\boldsymbol{x}}_{k-1}$ be the object state at the $(k-1)^{th}$ frame and $\boldsymbol{F}$ be the state transition matrix. In the prediction step, the state at the k^{th} frame $\hat{\boldsymbol{x}}'_k$ and state estimated covariance matrix $\boldsymbol{P}'_k$ are predicted via the following equations, where $\boldsymbol{Q}_k$ is the process noise covariance matrix. $\boldsymbol{Q}_k$ consists of the errors caused in the motion process. For example, if the velocity of the detected object changes rapidly, KF can determine an appropriate $\boldsymbol{Q}_k$ matrix to reflect the unreliability of the system at this moment.

$$\hat{\boldsymbol{x}}'_k = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \quad (1)$$

$$\boldsymbol{P}'_k = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^\top + \boldsymbol{Q}_k \quad (2)$$

In the update step, KF blends the new observation with the old information from prior state with the Kalman gain matrix $\boldsymbol{K}_k$. The estimation of $\boldsymbol{K}_k$ is shown in Eq. (3), where $\boldsymbol{H}$ is the observation matrix and $\boldsymbol{R}_k$ is the observation noise covariance matrix. In Eq. (4), the actual observation $\boldsymbol{z}_k$ is obtained to generate a posterior state estimate of $\hat{\boldsymbol{x}}'_k$. The residual $\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}'_k$ reflects the divergence between the predicted state and the observed state. Finally, in Eq. (5), the estimation state covariance matrix $\boldsymbol{P}'_k$ is also updated according to the Kalman gain $\boldsymbol{K}_k$.

$$\boldsymbol{K}_k = \boldsymbol{P}'_k\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{P}'_k\boldsymbol{H}^\top + \boldsymbol{R}_k\right)^{-1} \quad (3)$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}'_k + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}'_k\right) \quad (4)$$

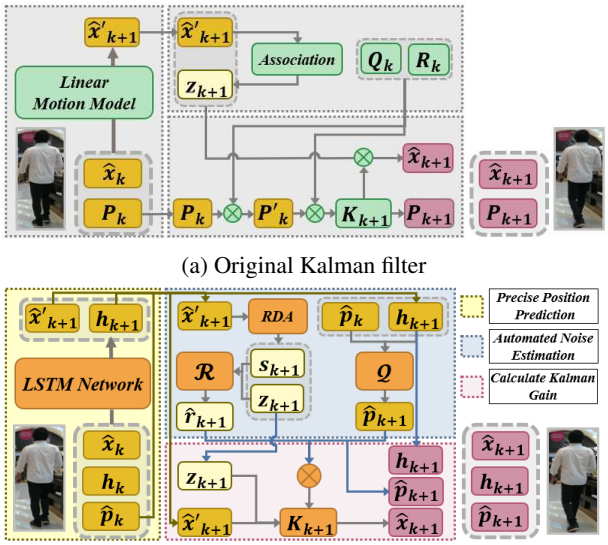
$$\boldsymbol{P}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}'_k \quad (5)$$

In the scenario of down-sampled MOT, the observations become sparse and each object appears in fewer number of video frames. Consequently, the uncertainty is amplified and it becomes more challenging to capture the model pattern. The traditional KF as well as its improved variants in StrongSORT and OC-SORT fail to address these unique challenges. Therefore, we are motivated to devise a new variant KF for sparse observations.

Sparse-Observation Kalman Filter

The pipeline of our proposed Sparse-Observation Kalman Filter (SOKF) is illustrated in Figure 2, with the following three key components.

LSTM-Based Position Prediction. Linear motion assumption has been commonly adopted by existing KF-based MOT models and yields satisfactory results even in datasets with obviously non-linear motion patterns (e.g., DanceTrack (Sun et al. 2022) with dancers performing on the stage). The reason is that cameras typically possess high frame rates and the motion between two neighboring frames can still be approximated as linear. Nevertheless, in down-sampled scenarios, the enlarged temporal gap between neighboring frames introduces more intricate motion dynamics, rendering the linear motion assumption untenable.



(a) Original Kalman filter

(b) Sparse-Observation Kalman filter

Figure 2: Pipelines of KF and SOKF.

There are some works (Li et al. 2008; Wei et al. 2019; Zhang et al. 2015) studying object tracking for cameras inherently with low frame rate. Their algorithm pipelines are focused on robust tracking, and often incur higher computation overhead. For example, (Zhang et al. 2015) adopts a complex matching mechanism based on particle swarm optimization. There are also several studies within the MOT domain have focused on the influence of non-linear motion. (Yang and Nevatia 2012; Lu et al. 2019) use visual factors for collecting nonlinear motion patterns to predict object positions. Since the goal of this paper is to achieve real-time tracking by purposely reducing the number of frames, the above solutions cannot be applied to down-sampled MOT.

We employ LSTM network to support more accurate position prediction with sparse observations and substitute Eq. (1) with Eq (6) for non-linear state transition. Besides $\hat{\boldsymbol{x}}_k$, the input of LSTM includes the observation outcome $\boldsymbol{z}_k$ and temporal gap $\boldsymbol{t}_s$ between neighboring frames so that the prediction can be adaptive to different sampling rates.

$$\hat{\boldsymbol{x}}'_{k+1} = \mathcal{F}_{lstm}([\hat{\boldsymbol{x}}_k, \boldsymbol{z}_k, \boldsymbol{t}_s]) \quad (6)$$

BNN-Based Noise Estimation. In Eq. (1) and (2), the state and covariance matrix are recursively updated via a shared linear transition matrix $\boldsymbol{F}$. Since we have discarded the linear motion assumption and replaced $\boldsymbol{F}$ with a LSTM network for state transition, we also need to devise a new mechanism for noise estimation update. Specifically, we discard the Gaussian noise assumption and adopt Bayesian Neural Network (BNN) (Shalileh 2021) to directly estimate the prediction error of LSTM. As depicted in Eq. (7), the input contains the prediction error $\hat{\boldsymbol{p}}_k$ and the current hidden feature $\boldsymbol{h}_{k+1}$ in the LSTM network. The prediction error estimation $\hat{\boldsymbol{p}}_{k+1}$ is derived through a two-layer fully connected

Bayesian neural network $\mathcal{Q}$, which outputs the distribution by treating its weights as a probability distribution, allowing derivation of the error margin of the LSTM.

$$\hat{\boldsymbol{p}}_{k+1} \quad = \quad \mathcal{Q}(\hat{\boldsymbol{p}}_k, \boldsymbol{h}_{k+1}) \tag{7}$$

To estimate the observation noise, we argue that state-of-the-art object detection models have exhibited outstanding performance and we should focus on the error derived from the object association module, which has been neglected by existing KF-based MOT models. Again, we adopt BNN for error prediction and its input consists of the distance vector $\boldsymbol{s}_{k+1}$ and the observation result $\boldsymbol{z}_{k+1}$. $\boldsymbol{s}_{k+1}$ consisting of the distances between the observed track and the five closest detection boxes, measures the impact of nearby tracks on the observed track associated to the correct detection.

$$\hat{\boldsymbol{r}}_{k+1} \quad = \quad \mathcal{R}(\boldsymbol{s}_{k+1}, \boldsymbol{z}_{k+1}) \tag{8}$$

Optimal Kalman Gain. The Kalman gain details the degree to which each measurement is incorporated into the new state estimate. Our objective is to derive the Kalman gain that minimizes the discrepancy $\mathcal{E}$ between $\boldsymbol{x}_k$ and $\hat{\boldsymbol{x}}_k$.

$$\mathcal{E} \quad = \quad ||\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k||^2 \tag{9}$$

$$\quad = \quad ||\boldsymbol{x}_k - \hat{\boldsymbol{x}}'_k + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}'_k)||^2 \tag{10}$$

$\mathcal{E}$ is a convex function and we can show that its hessian matrix $\mathcal{H}_\mathcal{E}$ can be formed by multiplying a non-zero vector with its transpose, and is thus a positive definite matrix.

$$\mathcal{H}_\mathcal{E} = 2(\boldsymbol{H}\hat{\boldsymbol{p}}_k + \hat{\boldsymbol{r}}_k)(\boldsymbol{H}\hat{\boldsymbol{p}}_k + \hat{\boldsymbol{r}}_k)^T = 2\boldsymbol{A}\boldsymbol{A}^T \tag{11}$$

Therefore, we can derive the optimal $\boldsymbol{K}_k$ that minimizes $\mathcal{E}$ by setting the derivative of $\mathcal{E}$ to zero.

$$\frac{\partial}{\partial \boldsymbol{K}_k}||\boldsymbol{x}_k - \hat{\boldsymbol{x}}'_k - \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}'_k)||^2 = 0 \tag{12}$$

$$\Rightarrow \frac{\partial}{\partial \boldsymbol{K}_k}||(I - \boldsymbol{K}_k\boldsymbol{H})\hat{\boldsymbol{p}}_k - \boldsymbol{K}_k\hat{\boldsymbol{r}}_k||^2 = 0 \tag{13}$$

$$\Rightarrow -2(\hat{\boldsymbol{p}}_k - \boldsymbol{K}_k(\boldsymbol{H}\hat{\boldsymbol{p}}_k + \hat{\boldsymbol{r}}_k))(\boldsymbol{H}\hat{\boldsymbol{p}}_k + \hat{\boldsymbol{r}}_k)^T = 0 \tag{14}$$

$$\Rightarrow \boldsymbol{K}_k = \hat{\boldsymbol{p}}_k\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1} \tag{15}$$

Model Training. Our novel KF variant incorporates LSTM and BNN that require training. For LSTM, we construct the ground truth of $\boldsymbol{x}_{k+1}$ with varying sampling rates from the object tracks in the training data of MOT benchmark. The network is trained via the mean square error loss between $\boldsymbol{x}_{k+1}$ and $\hat{\boldsymbol{x}}'_{k+1}$. For BNN training in noise estimation, we utilize the difference between the LSTM predicted state and the ground truth state as the training target of $\mathcal{Q}$. Likewise, for the BNN $\mathcal{R}$, the training loss is set to the difference between the detected bounding box associated with the track and the bounding box represented by the ground truth state.

Robust Data Association (RDA)

Data association is also a key component in the tracking-by-detection paradigm. The mainstream metrics estimate the spatial matching score according to either IoU (Intersection of Union) (Zhang et al. 2022; Cao et al. 2023) or center point distance between two bounding boxes (Wojke, Bewley, and Paulus 2017; Du et al. 2022). On the other hand, there also exist certain factors that have been adopted in the loss of object detection (e.g., aspect ratio in CIoU loss (Zheng et al. 2020)), but they are not leveraged by object tracking.

Inspired by (Zhao et al. 2022), we perform an experimental analysis on these metrics when applied to object tracking across down-sampled video frames. We denote the sample reduction ratio by RR, which implies that $\frac{1}{RR}$ frames are sampled. When $RR = 1$, all the frames are preserved. We vary RR from 1 to 9 for each setting and randomly collect $10,000$ bounding box association cases that can be successfully solved by at least one of the following metrics, including the overlap, center point distance, and aspect ratio of the bounding boxes, denoted by IoU, DIST, and SCALE.

Interesting findings can be derived from the results reported in Table 1. The set S_{metric} includes the cases that can be correctly matched by the associated metric. P_{SCALE} represents the cases that can only be solved by SCALE, i.e., IoU and DIST fail in these cases. With $RR = 1$, the IoU or DIST are able to correctly identify around 99% of the matching cases. The metric SCALE is inferior to the two metrics as it generates many false negatives. Its complementary effect to IoU and DIST can be negligible because only 0.31% of cases can be uniquely solved by SCALE. This may explain why SCALE is not adopted by the state-of-the-art MOT methods. However, when RR increases, IoU and DIST become less reliable as the sizes of $|S_{IoU}|$ and $|S_{DIST}|$ reduce. It is interesting to find that the factor of SCALE plays a more important role and its size of P_{SCALE} increases with RR. This finding motivates us to devise a comprehensive association metric that incorporates all metrics.

RR	1	3	5	7	9
$\lvert S_{IoU}\rvert$	9899	9504	9169	8812	8565
$\lvert S_{DIST}\rvert$	9891	9579	9320	8999	8797
$\lvert S_{SCALE}\rvert$	7886	6928	6444	6191	6010
$\lvert P_{SCALE}\rvert$	31	118	174	234	275

Table 1: distance metrics analysis on the MOT17 dataset.

Let $\boldsymbol{D}_{iou}$ denote the overlap distance between two bounding boxes and $\boldsymbol{D}_{dist}$ denote the distance between two center points of the bounding boxes, which is further normalized by dividing by the diagonal length of the smallest enclosing box covering the bounding boxes. For the factor of aspect ratio, we define $\boldsymbol{D}_{scale}$ as

$$\boldsymbol{D}_{scale} = \frac{4}{\pi^2}\left(\arctan\frac{w_1}{h_1} - \arctan\frac{w_2}{h_2}\right)^2 \tag{16}$$

where w_i and h_i are the width and height of the two bounding boxes, respectively. To integrate these three distances, we define $\boldsymbol{D}_{rda}$ as follows. The idea is to first use IoU and DIST if these two metrics can provide confident matching results. This is because as revealed in Table 1, these two factors normally provide better results than SCALE. We use $\mu(\boldsymbol{D}_{dist}, \boldsymbol{D}_{iou})$ to reversely approximate for the confidence. The function denoted by μ represents the arithmetic mean (average) of a given set of values. This is a reasonable estimation because it implies that the estimated bounding box

is close to the region of the detected object. If this value is smaller than a threshold σ, the tracking confidence is high and we directly set $\boldsymbol{D}_{rda} = \mu(\boldsymbol{D}_{dist}, \boldsymbol{D}_{iou})$. Otherwise, we need to incorporate $\boldsymbol{D}_{scale}$ as a complementary factor and set $\boldsymbol{D}_{rda} = \mu(\boldsymbol{D}_{dist}, \boldsymbol{D}_{iou}, \boldsymbol{D}_{scale})$.

Experiment

Experimental Setup

Benchmark Datasets. We use three benchmark datasets for performance evaluation, including MOT17 (Milan et al. 2016), MOT20 (Dendorfer et al. 2020) and DanceTrack (Sun et al. 2022). MOT17 contains 14 videos (7 for training and 7 for testing) of pedestrians in both indoor and outdoor scenes. MOT20 contains 8 videos (4 for training, 4 for testing) in crowded environments such as train stations, town squares and a sports stadium. DanceTrack is a recent dataset proposed to emphasize the importance of motion analysis. The frequent crossover and diverse body gestures bring particular challenges. It provides 100 videos and the split ratio for training, validation and test dataset is $40 : 25 : 35$.

Since the testing videos of these datasets are not annotated and the focus of this paper is down-sampled MOT, we directly use the annotated videos for performance evaluation. For MOT17 and MOT20, we split the videos into two parts of equal length and used them for training and testing, respectively. For DanceTrack, we use the training set for training and report the performance on its validation set.

Performance Metrics. To evaluate the overall tracking accuracy, we adopt MOTA (Bernardin and Stiefelhagen 2008), IDF1 (Ristani et al. 2016) and HOTA (Luiten et al. 2021). Generally, the MOTA is biased towards measuring the quality of object detection and IDF1 emphasizes the effect of accurate association. HOTA is a recent metric proposed to explicitly balance the effect of detection and association.

As to efficiency, we adopt FPS as a straightforward metric. It refers to the number of video frames that can be processed per second. In addition, we propose a new metric called **Time@HOTA**. The motivation is that we can adjust RR to generate a trade-off curve between processing time and HOTA, as shown in Figure 1. It can be expected that with a larger processing time (i.e., smaller RR), we can obtain higher HOTA. Time@HOTA measures the processing time required to reach a specified HOTA. For example, $Time@62 = 19$ for our SR-Track at dataset MOT17 implies that it takes 19 seconds for SR-Track to process the testing videos in MOT17 with an accuracy level of $HOTA = 62$.

Comparison Methods. We compare SR-Track with representative and open-sourced trackers in all paradigms. Among these competitors, we consider ByteTrack (Zhang et al. 2022), OC-SORT (Cao et al. 2023) and SimpleTrack (Li et al. 2022) as **real-time trackers** because they can achieve as high FPS as our SR-Track. The remaining approaches, including TransTrack (Sun et al. 2020), TrackFormer (Meinhardt et al. 2022), MOTR (Zeng et al. 2022), StrongSORT (Du et al. 2022) and BoT-SORT (Aharon et al. 2022), are called **expensive trackers** as they exchange processing time for higher tracking accuracy.

Implementation Details

Our SR-Track follows the paradigm of tracking-by-detection. For object detector, we directly adopt the trained YOLOX provided by previous trackers . As to our proposed Kalman filter, we set hidden size to 128 for the LSTM network and adopt two-layer Bayesian neural network to implement $\mathcal{Q}$ and $\mathcal{R}$. All models are trained using the Adam optimizer for 100 epochs with a batch size of 32. The initial learning rate is set to 0.01 and linearly decayed to 0.0001. All the experiments are conducted using PyTorch and ran on a desktop with 10th Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz and NVIDIA GeForce RTX 3090Ti GPU.

Comparison with Real-time Trackers

In the first experiment, we compare our SR-Track with the real-time trackers under different reduction ratios (with RR set to 3, 5, 7 and 9, respectively). As shown in Table 2, these trackers demonstrate similar inference speed. OC-SORT, ByteTrack and SR-Track use YOLOX as the object detector and ignore visual similarity. Although SimpleTrack adopts appearance similarity for person ReID, it trains the object detector and visual embedding with a single network to avoid re-computation cost. Its FPS is slightly lower than other real-time trackers. Among these real-time trackers, SR-Track achieves the highest metrics across all the datasets, owing to its KF designed for the observation-sparse scenario. The performance gap between ByteTrack and our SR-Track is widened when RR increases. In MOT20, the HOTA of SR-Track is higher than ByteTrack by 2.3% when $RR = 3$, which is enlarged to 10% when $RR = 9$.

DanceTrack is a challenging dataset with complex motion patterns and frequent crossover of dancers, which are difficult for existing trackers to perform correct association. Thus, their derived IDF1 and HOTA in DanceTrack are generally lower than those in MOT17 and MOT20. OC-SORT outperforms ByteTrack in this dataset because it is specially designed for DanceTrack and occlusion with excessive nonlinear motion. Nevertheless, the HOTA of OC-SORT degrades to be close to ByteTrack when RR increases, implying that its strategy is not robust to the observation-sparse scenario. These two models are both significantly inferior to our SR-Track. When $RR = 9$, we boost the HOTA from 33.4 in OC-SORT to 39.1, with 17.1% improvement.

Comparison with Expensive Trackers

In Table 3, we compare SR-Track with the expensive trackers under $RR = 5$ and $RR = 9$. For TransTrack, TrackFormer, MOTR, StrongSORT, their performance is clearly inferior to our SR-Track in terms of both tracking efficiency and accuracy. BoT-SORT is the only method whose accuracy can be slightly better than our SR-Track in MOT17. However, its tracking speed is very slow and the FPS is 6 times lower than SR-Track. Furthermore, similar to previous findings in Table 2, the advantage of SR-Track becomes more obvious when RR increases. In MOT20 with $RR = 9$, our SR-Track can even achieve higher accuracy than BoT-SORT, with 7 times faster tracking speed. The results on DanceTrack are not available because we lack sufficient hardware resources to re-train these models.

	$RR=3$			$RR=5$			$RR=7$			$RR=9$			FPS
	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	
	Dataset MOT17												
SimpleTrack	59.8	69.3	75.6	58.6	66.3	73.4	57.3	63.9	72.1	56.0	61.4	70.2	22.5
OC-SORT	63.7	68.6	74.5	61.5	63.7	71.0	58.8	59.3	67.6	57.9	57.8	66.4	29.0
ByteTrack	64.8	73.8	76.3	61.8	70.3	72.1	61.0	67.9	71.0	58.9	65.2	68.8	**29.6**
SR-Track (ours)	**67.2**	**76.0**	**78.5**	**66.2**	**73.7**	**76.9**	**65.2**	**70.7**	**75.4**	**63.4**	**68.4**	**73.3**	29.4
	Dataset MOT20												
SimpleTrack	52.2	65.3	68.0	51.4	63.7	67.7	50.0	61.6	65.8	47.8	58.8	62.5	7.0
OC-SORT	56.3	69.6	72.1	54.6	68.0	69.8	53.2	66.4	68.1	50.7	63.3	64.4	**18.7**
ByteTrack	56.0	71.3	71.1	55.5	70.1	70.8	54.2	68.4	69.7	50.7	65.7	65.2	17.5
SR-Track (ours)	**57.3**	**71.8**	**73.6**	**57.6**	**71.3**	**74.1**	**58.1**	**70.6**	**74.8**	**55.8**	**68.9**	**71.2**	16.4
	Dataset DanceTrack												
OC-SORT	45.2	81.6	43.7	38.8	73.8	36.0	36.2	66.8	34.8	33.4	61.1	32.5	29.0
ByteTrack	40.7	82.3	46.9	35.5	74.7	39.8	32.8	68.5	37.0	32.0	63.0	35.8	**29.6**
SR-Track (ours)	**54.1**	**88.2**	**53.2**	**46.6**	**84.3**	**45.4**	**42.7**	**79.6**	**40.8**	**39.1**	**74.8**	**37.1**	29.4

Table 2: Comparison with real-time trackers on three benchmark datasets with varying frame reduction ratio RR.

	$RR=5$			$RR=9$			FPS
	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	
	MOT17						
TransTrack	56.8	66.1	66.6	54.8	61.2	62.2	10.0
TrackFormer	59.1	66.2	68.2	55.6	60.6	64.2	7.4
MOTR	59.8	65.5	68.8	56.2	61.0	65.4	7.5
StrongSORT	63.6	61.9	70.9	59.3	53.2	62.9	7.1
BoT-SORT	**66.4**	**74.3**	**77.9**	63.1	**70.1**	73.1	4.5
SR-Track	66.2	73.7	76.9	**63.4**	68.4	**73.3**	**29.4**
	MOT20						
TransTrack	31.6	47.3	44.6	30.5	44.9	42.4	7.2
TrackFormer	47.4	70.6	56.8	43.3	65.3	51.3	4.1
MOTR	42.8	50.6	56.1	38.0	43.0	49.7	4.2
StrongSORT	56.5	67.4	72.8	50.7	61.2	66.6	1.4
BoT-SORT	**57.7**	71.1	73.9	54.0	67.2	69.3	2.4
SR-Track	57.6	**71.3**	**74.1**	**55.8**	**68.9**	**71.2**	**16.4**

Table 3: Comparison with expensive trackers on the MOT17 and MOT20 under different settings of RR.

Trade-off Between Efficiency and Accuracy

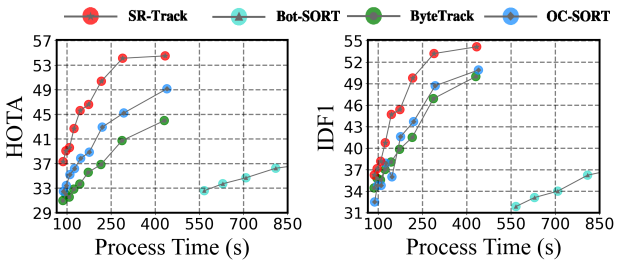


Figure 3: Trade-off analysis in DanceTrack.

At the beginning of the paper, we have reported the trade-off between processing time and tracking accuracy for MOT17. The results on DanceTrack in terms of IDF1 and HOTA are presented in Figure 3. ByteTrack is fast and accurate because it does not rely on visual similarity and improves the association mechanism by taking into account detected objects with low confidence. OC-SORT outperforms ByteTrack in the dataset DanceTrack because OC-SORT is better at capturing complex motion patterns. StrongSORT and BoT-SORT utilize visual similarity by extracting appearance features and achieving high accuracy but at the cost of significantly higher computation overhead. SimpleTrack, the most recent work proposed in the paradigm of joint training of object detection and embedding, achieves modest performance. However, since the joint training is difficult to coordinate, it does not demonstrate superiority in terms of effectiveness. Finally, TransTrack jointly trains object detection, ReID and motion estimation in the same framework. Its performance is not satisfactory due to limited training samples and its online inference is also cost prohibitive.

Time@HOTA	66	65	64	63	62
TrackFormer	>716.8	>716.8	>716.8	>716.8	>716.8
MOTR	>707.2	>707.2	>707.2	>707.2	309.1
TransTrack	>530.4	>530.4	>530.4	>530.4	>530.4
SimpleTrack	>235.4	>235.4	>235.4	>235.4	>235.4
StrongSORT	182.6	166.8	123.3	117.3	111.8
BoT-SORT	170.6	157.0	145.8	128.8	116.4
OC-SORT	148.3	94.4	65.9	53.1	43.9
ByteTrack	81.1	62.3	52.1	45.0	36.9
SR-Track	**33.8**	**25.5**	**22.1**	**19.5**	**17.9**

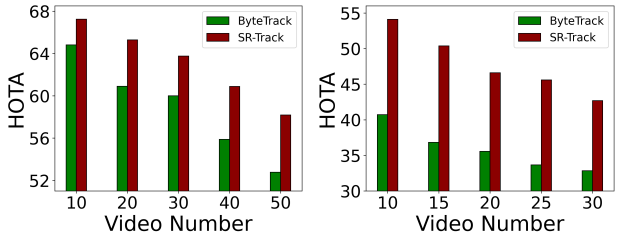
Table 4: Time@HOTA in MOT17 (in seconds).

We also study the performance of these trackers in the metric Time@HOTA. As reported in Tables 4 and 5, ByteTrack implements the best among comparison trackers for MOT17. Our method can further reduce its processing time by half with a given HOTA requirement. For example, it takes SR-Track 17.9s to generate tracking results in MOT17 with HOTA=62, whereas ByteTrack requires 36.9s. In the DanceTrack, the advantage of SR-Track is enlarged to 3×. In Figure 4, we use one 3090Ti GPU to perform real-time tracking on multiple video streams simultaneously. With more video streams, we increase RR to guarantee real-time tracking, but sacrifice HOTA. SR-Track dominates ByteTrack when handling large-scale video streams and can save

a significant amount of GPU resources when deployed in smart city applications with thousands of cameras.

Time@HOTA	53	50	47	44	41
BoT-SORT	3873.1	2671.9	2036.1	1662.6	1405.1
OC-SORT	879.6	504.9	345.5	249.2	196.7
MOTR	610.9	363.6	295.1	248.3	214.3
ByteTrack	>861.8	>861.8	>861.8	432.1	295.6
SR-Track	**263.1**	**211.6**	**177.3**	**132.5**	**115.1**

Table 5: Time@HOTA in DanceTrack (in seconds).



(a) MOT17 (b) DanceTrack

Figure 4: Parallel processing of large-scale video streams.

	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
SR-Track	**63.6**	**62.9**	**64.9**	**68.6**	**73.2**
Ours w/o SOKF	59.6	58.0	62.5	64.1	70.3
Ours w/o RDA	61.4	60.1	61.2	65.7	70.2
ByteTrack	59.0	57.6	61.6	65.3	68.8
Break-down analysis on SOKF					
SOKF w/o LSTM	60.8	61.6	61.0	67.6	70.6
SOKF w/o BNN	62.0	60.0	64.7	66.3	72.6
SOKF w/o OKG	61.0	58.5	64.4	65.0	72.1

Table 6: Ablation study of SR-Track on MOT17 ($RR = 9$).

Ablation Study

We evaluate the advantage brought by the Sparse-Observation Kalman filter (SOKF) and robust data association (RDA) in Table 6. ByteTrack can be viewed as a variant without these two components. It is not surprising to find that when RDA is removed, the performance on the matching-related metrics, such as IDF1, drops significantly. In contrast, SOKF is more important for the remaining metrics. We also conduct a break-down analysis on the components of SOKF and examine the effect of our proposed LSTM-Based Position Prediction (LSTM), BNN-Based Noise Estimation (BNN) and Optimal Kalman Gain (OKG). We can see that they all contribute to the improvement of tracking accuracy.

Experiments Without Down-Sampling

We examine the performance of our SR-Track on the original dataset without down-sampling. Table 7 shows the results returned by the leaderboard of DanceTrack. SR-Track is the best performer and improves the metrics of HOTA, IDF1 and AssA by 7.3%, 9.4% and 12.9%, respectively.

	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
FairMOT	39.7	66.7	23.8	82.2	40.8
QDTrack	45.7	72.1	29.2	83.0	44.8
TransTrack	45.5	75.9	27.5	88.4	45.2
MOTR	48.4	71.8	32.7	79.2	46.1
ByteTrack	47.3	71.6	31.4	89.5	52.5
OC-SORT	55.1	80.3	38.0	89.4	54.2
SR-Track	**59.1**	**81.5**	**42.9**	**92.4**	**59.3**

Table 7: Performance on DanceTrack test dataset.

Case Analysis

Finally, we perform a case analysis by comparing SR-Track and ByteTrack on MOT17 with $RR = 9$. As shown in Figure 5, we highlight the incorrect association generated by ByteTrack. From frame 4 to frame 5, its KF makes the wrong estimation of the next bounding box, whereas our SOKF delivers accurate estimation. From frame 16 to frame 17, ByteTrack incurs ID switching caused by occlusion, but our SR-Track, with a more robust association, can resolve the issue.

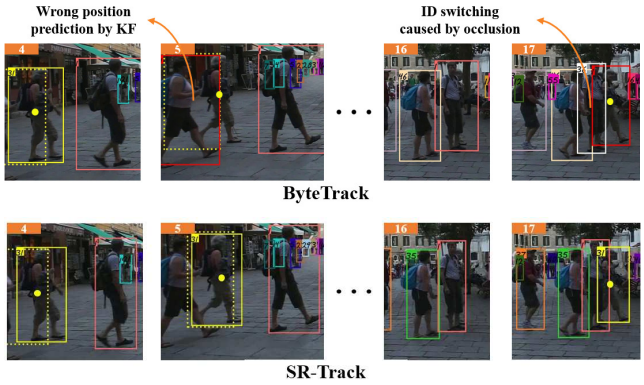


Figure 5: A case study in MOT17 dataset.

Conclusion

In this paper, we studied a new scenario of multi-object tracking on down-sampled video frames and devised a sampling-resilient tracker. In particular, we proposed a novel sparse-observation Kalman filter (SOKF) for accurate motion estimation and a comprehensive data association metric for robust inter-frame matching. Experiments on three datasets show that our proposed SR-Track establishes new SOTA performance for down-sampled object tracking.

Acknowledgments

This work is sponsored by the National Key Research and Development Project of China (2022YFF0902000), the Key Research Program of Zhejiang Province (2023C01037) and CCF-Huawei Populus Grove Fund.

References

Aharon, N.; Orfaig, R.; Bobrovsky, B.; and Bobrovsky, B. 2022. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *CoRR*, abs/2206.14651.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP J. Image Video Process.*, 2008.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F. T.; and Upcroft, B. 2016. Simple online and realtime tracking. In *ICIP 2016*, 3464–3468. IEEE.

Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; and Kitani, K. 2023. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *CVPR 2023*, 9686–9696.

Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I. D.; Roth, S.; Schindler, K.; and Leal-Taixé, L. 2020. MOT20: A benchmark for multi object tracking in crowded scenes. *CoRR*, abs/2003.09003.

Du, Y.; Song, Y.; Yang, B.; and Zhao, Y. 2022. StrongSORT: Make DeepSORT Great Again. *CoRR*, abs/2202.13514.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. YOLOX: Exceeding YOLO Series in 2021. *CoRR*, abs/2107.08430.

Li, J.; Ding, Y.; Wei, H.; Zhang, Y.; and Lin, W. 2022. SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking. *Sensors*, 22(15): 5863.

Li, Y.; Ai, H.; Yamashita, T.; Lao, S.; and Kawade, M. 2008. Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10): 1728–1740.

Li, Z.; Zhang, D.; Shen, Y.; and Chen, G. 2023. Human-in-the-Loop Vehicle ReID. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 6048–6055. AAAI Press.

Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; and Zou, J. 2020. Rethinking the competition between detection and ReID in Multi-Object Tracking. *CoRR*, abs/2010.12138.

Lu, W.; Ting, J.; Little, J. J.; and Murphy, K. P. 2013. Learning to Track and Identify Players from Broadcast Sports Videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7): 1704–1716.

Lu, W.; Zhou, Z.; Zhang, L.; and Zheng, G. 2019. Multi-target tracking by non-linear motion patterns based on hierarchical network flows. *Multim. Syst.*, 25(4): 383–394.

Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P. H. S.; Geiger, A.; Leal-Taixé, L.; and Leibe, B. 2021. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *Int. J. Comput. Vis.*, 129(2): 548–578.

Meinhardt, T.; Kirillov, A.; Leal-Taixé, L.; and Feichtenhofer, C. 2022. TrackFormer: Multi-Object Tracking with Transformers. In *CVPR 2022*, 8834–8844. IEEE.

Merad, D.; Aziz, K.; Iguernaissi, R.; Fertil, B.; and Drap, P. 2016. Tracking multiple persons under partial and global occlusions: Application to customers' behavior analysis. *Pattern Recognit. Lett.*, 81: 11–20.

Milan, A.; Leal-Taixé, L.; Reid, I. D.; Roth, S.; and Schindler, K. 2016. MOT16: A Benchmark for Multi-Object Tracking. *CoRR*, abs/1603.00831.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS 2015*, 91–99.

Ristani, E.; Solera, F.; Zou, R. S.; Cucchiara, R.; and Tomasi, C. 2016. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In *ECCV 2016 Workshops*, volume 9914 of *Lecture Notes in Computer Science*, 17–35.

Shalileh, S. 2021. Improving Maximum Likelihood Estimation Using Marginalization and Black-Box Variational Inference. In *IDEAL 2021*, volume 13113 of *Lecture Notes in Computer Science*, 204–212. Springer.

Shan, C.; Wei, C.; Deng, B.; Huang, J.; Hua, X.-S.; Cheng, X.; and Liang, K. 2020. Tracklets predicting based adaptive graph tracking. *arXiv preprint arXiv:2010.09015*.

Sun, P.; Cao, J.; Jiang, Y.; Yuan, Z.; Bai, S.; Kitani, K.; and Luo, P. 2022. DanceTrack: Multi-Object Tracking in Uniform Appearance and Diverse Motion. In *CVPR 2022*, 20961–20970. IEEE.

Sun, P.; Jiang, Y.; Zhang, R.; Xie, E.; Cao, J.; Hu, X.; Kong, T.; Yuan, Z.; Wang, C.; and Luo, P. 2020. TransTrack: Multiple-Object Tracking with Transformer. *CoRR*, abs/2012.15460.

Tian, W.; Lauer, M.; and Chen, L. 2020. Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios. *IEEE Trans. Intell. Transp. Syst.*, 21(1): 374–384.

Wang, Y.; Kitani, K.; and Weng, X. 2021. Joint Object Detection and Multi-Object Tracking with Graph Neural Networks. In *ICRA 2021*, 13708–13715. IEEE.

Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; and Wang, S. 2020. Towards Real-Time Multi-Object Tracking. In *ECCV 2020*, volume 12356 of *Lecture Notes in Computer Science*, 107–122. Springer.

Wei, P.; Shi, H.; Yang, J.; Qian, J.; Ji, Y.; and Jiang, X. 2019. City-scale vehicle tracking and traffic flow estimation using low frame-rate traffic cameras. In *ISWC 2019*, 602–610. ACM.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP 2017*, 3645–3649. IEEE.

Xiao, Z.; Zhang, D.; Li, Z.; Wu, S.; Tan, K.; and Chen, G. 2023. DoveDB: A Declarative and Low-Latency Video Database. *Proc. VLDB Endow.*, 16(12): 3906–3909.

Xu, R.; Nikouei, S. Y.; Chen, Y.; Polunchenko, A.; Song, S.; Deng, C.; and Faughnan, T. R. 2018. Real-Time Human Objects Tracking for Smart Surveillance at the Edge. In *ICC 2018*, 1–6. IEEE.

Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; and Alameda-Pineda, X. 2021. TransCenter: Transformers with Dense Queries for Multiple-Object Tracking. *CoRR*, abs/2103.15145.

Yang, B.; and Nevatia, R. 2012. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR 2012*, 1918–1925. IEEE Computer Society.

Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; and Hoi, S. C. H. 2022. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(6): 2872–2893.

Yu, E.; Li, Z.; Han, S.; and Wang, H. 2023. RelationTrack: Relation-Aware Multiple Object Tracking With Decoupled Representation. *IEEE Trans. Multim.*, 25: 2686–2697.

Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; and Wei, Y. 2022. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *ECCV 2022*, volume 13687 of *Lecture Notes in Computer Science*, 659–675. Springer.

Zhang, D.; Ma, T.; Hu, J.; Bei, Y.; Tan, K.; and Chen, G. 2023. Co-movement Pattern Mining from Videos. *CoRR*, abs/2308.05370.

Zhang, X.; Hu, W.; Xie, N.; Bao, H.; and Maybank, S. J. 2015. A Robust Tracking System for Low Frame Rate Video. *Int. J. Comput. Vis.*, 115(3): 279–304.

Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; and Wang, X. 2022. ByteTrack: Multi-object Tracking by Associating Every Detection Box. In *ECCV 2022*, volume 13682 of *Lecture Notes in Computer Science*, 1–21. Springer.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2021. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vis.*, 129(11): 3069–3087.

Zhao, X.; Zeng, W.; Tang, J.; Li, X.; Luo, M.; and Zheng, Q. 2022. Toward Entity Alignment in the Open World: An Unsupervised Approach with Confidence Modeling. *Data Sci. Eng.*, 7(1): 16–29.

Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; and Ren, D. 2020. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In *AAAI 2020*, 12993–13000. AAAI Press.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking Objects as Points. In *ECCV 2020*, volume 12349 of *Lecture Notes in Computer Science*, 474–490. Springer.