Sampling-Resilient Multi-Object Tracking

Anonymous CVPR submission

Paper ID 4830

Abstract

Multi-object tracking (MOT) is a cornerstone operator to support video surveillance or analytical tasks. To process large-scale live video streams in real time, there is critical demand for an efficient and accurate tracking mechanism. In this paper, we study a new scenario called down-sampled multi-object tracking, which performs object detection and association only upon a subset of video frames, to improve efficiency with tolerable accuracy decline.

The problem is challenging — directly applying state-of-the-art MOT methods on the down-sampled video frames results in significant performance degradation. To devise a sampling-resilient tracker, we make the following contributions. First, we propose an enhanced Kalman filter with more informative state representation and dynamic parameter matrix update mechanism to reduce the divergence between observations and internal state prediction. Second, to associate the detected bounding boxes robustly, we propose a comprehensive similarity metric that systematically integrates multiple spatial matching signals.

Experiments on three benchmark datasets show that our proposed tracker is the most suitable for real-time tracking and achieves the best trade-off between efficiency and accuracy. Compared with ByteTrack, we can further reduce the processing time by 2× in MOT17 and 3× in DanceTrack and reach the same level of tracking accuracy.

1. Introduction

Multi-object tracking (MOT) aims at detecting and tracking all the moving objects from video clips or live streams, while maintaining a unique identifier for each object. Massive research efforts has been devoted into this domain with fruitful progress. The proposed trackers have witnessed great success in numerous applications, such as smart video surveillance [27], traffic monitoring [23], customer behavior analysis [15] and sports analytics [12].

In this paper, we focus on real-time MOT and study a new scenario called down-sampled multi-object tracking. It performs object detection and association only upon a

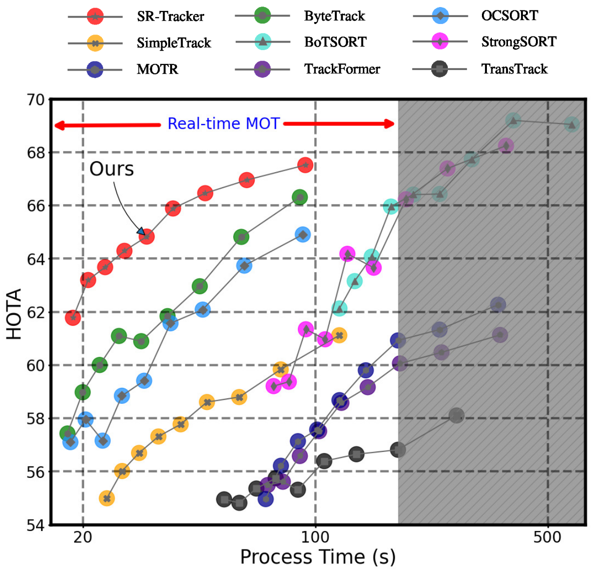

Figure 1. We plot the trade-off between tracking efficiency and accuracy for state-of-the-art MOT methods, by adjusting sampling rate in the test set of MOT17. With fewer video frames sampled, the total process time is reduced, but at the cost of lower tracking accuracy. Since the total length of the test data is 177.2 seconds, the non-shaded area implies real-time tracking which requires online inference to catch up with the speed of video streaming. The results on HOTAshow that our proposed SR-Tracker establishes clear superiority to support real-time tracking — it is the most efficient at the same accuracy level. Note that the process time in the x-axis is in log scale.

sampled set of video frames to further reduce inference time, with tolerable accuracy decline. The task has the potential to achieve very good trade-off between tracking efficiency and accuracy. It has not been well exploited in the past literature and is useful for scalable video analytic applications. In practice, the computation overhead of deep neural networks remains prohibitive for many surveillance cameras or low-end IoT devices. Their captured video streams have to be collected and transmitted to an edge server or cloud server for real-time processing [11]. Given a smart surveillance system with thousands of cameras, there is a critical demand for an efficient and accurate tracking mechanism so that the video streams can be processed with affordable computation devices.

The new problem of down-sampled MOT is challenging because the motion pattern becomes more difficult to capture and the position prediction in the next frame becomes less accurate. 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 tradeoff between efficiency and accuracy in terms of MOTA, IDF1 and HOTA, by adjusting different sampling rates. When the video frames are sampled at a very low rate, it's indeed that the processing time can be significantly reduced. However, the tracking accuracy also declines to an impractical level. Detailed performance analysis of these trackers will be presented in Section 4.5. These findings lead to the conclusion that existing MOT solutions are not sampling resilient.

To devise a more sampling-resilient MOT model, we are motivated to devise more accurate motion prediction and robust data association. We follow the tracking-by-detection paradigm as used in ByteTrack and propose a novel Sampling-Resilient Tracker (SR-Tracker) with the following technical contributions. First, to predict the next position of an object more accurately, we propose an enhanced Kalman filter with more informative state representation and parameter matrix update mechanism. More specially, we augment the state representation with acceleration speed to capture non-linear motion pattern. The estimated and process covariance matrix can be dynamically updated according to the divergence between observation and internal state prediction. Second, to associate the detected bounding boxes more robustly, we reveal interesting findings from an experimental analysis on existing metrics. We propose a comprehensive similarity metrics that integrates multiple spatial 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-Tracker exhibit clearly higher accuracy accuracy. In addition, compared with the state-of-the-art real-time tracker, which is known as ByteTrack, we can further reduce the processing time by 2× in MOT17 and 3× in DanceTrack and reach the same level of tracking accuracy.

In summary, we make the following contributions:

1. We study a new and challenging scenario called down-sampled multi-object tracking.
2. We propose SR-Tracker, with an enhanced Kalman filter for more accurate motion prediction and a comprehensive data association metric for more robust inter-frame matching.
3. Extensive experiments verify that SR-Tracker establishes new SOTA performance for real-time tracking.

2. Related Work

In this section, we review state-of-the-art multi-object trackers and divide them 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.

2.1. Tracking-by-Detection Methods

SORT [3], DeepSORT [26], OCSORT [4], StrongSORT [6], BoTSORT [1] and ByteTrack [31] 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 [18] as the default detector, which is replaced by YOLOX [8] 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. Almost all the tracking-by-detection methods adopt Kalman filter to predict the location of the tracklets in the next frame. 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. In StrongSORT [6], Kalman filter is further improved to adaptively modulate the noise scale according to the quality of object detection. It also adopts enhanced correlation coefficient (ECC) for camera motion compensation. As to visual similarity, DeepSORT [26], StrongSORT and BoTSORT 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 [31] achieves the best tradeoff between efficiency and accuracy. It does not rely on visual similarity to save computation cost. As a compensation, it presents a robust association strategy to take into account detected objects with low confidence.

2.2. Joint-Detection-and-Tracking Methods

JDE [25] is a pioneering work that allows target 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 task. To resolve the competition issue, CSTrack [10] devises a cross-correlation network to learn task-dependent representations. RelationTrack [29] presents global context disentangling (GCD) to decouple th learned features in the two tasks. FairMOT [32] 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 [9] is designed to migitate 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 prediction of inter-frame object motion in the training framework. In other words, Kalman filter is discarded. CenterTrack [34] and TransCenter [28] 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 [28] 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 [24] and FUEFT [20], motion and appearance features are fed into a graph neural network (GNN) to predict the association matrix of tracklets and detected bounding boxes. TransTrack [22] utilizes the attention mechanism to model the detection and tracking, and outputs the predicted bounding box of tracked objects. Recently, TrackFormer [14] 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 for real-time tracking.

3. Method

In this section, we first present the preliminary knowledge on Kalman filter (KF). Afterwards, we address the limitations of KF when applied in existing trackers and propose a more robust Kalman filter with informative state representation and divergence-aware parameter update mechanism. Finally, we present an improved data association strategy with a more comprehensive similarity metric.

3.1. Principles of Kalman Filter

Kalman filter (KF) has been widely adopted in object tracking to predict 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 prediction, rendering it well-suited for real-time analysis.

As preliminary knowledge, we first explain the workflow of KF when applied in object tracking. 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 process and is an important parameter matrix in KF. 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.

$$\hat{\boldsymbol{x}}'_k = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \tag{1}$$

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

In the update step, KF blends the new measurement 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 measurement matrix and $\boldsymbol{R}_k$ is the measurement noise covariance matrix. In Eq. (4), the actual measurement $\boldsymbol{z}_k$ 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 $\boldsymbol{K}_k$.

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

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

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

3.2. Enhanced Kalman Filter (EKF)

Traditional KF fails in the scenario of down-sampled MOT due to its constant velocity assumption and simple noise scale estimation. We also notice that there have been efforts devoted to improve KF for more accurate motion prediction. For examples, StrongSORT [6] and GiaoTracker [7] adaptively modulate the measurement noise scale according to the quality of object detection, i.e., the detection confidence is taken into account in the update of $\boldsymbol{R}_k$. In OCSORT, the authors notice that the assumption of consistent velocity direction does not hold due to the non-linear motion of objects and state noise. They still approximate the motion as linear but propose a way to reduce the noise and add the velocity consistency (momentum) term into the cost matrix for better matching between tracklets and observations.

Nonetheless, these improved variants of KF described above cannot provide robust object motion prediction across down-sampled frames. When the video frames become sparse, the underlying assumptions of these trackers may not hold because the detected objects exhibit rapid position movement or complex motion pattern between two adjacent frames. These challenges call for systematic improvement of KF and our goal is to re-design a Kalman filter with new state representation and parameter update mechanism.

Informative State Representation. The object state is normally represented by its position, shape and motion information. As a common practice, the state vector adopted by most MOT trackers is in the form of $(x_c, y_c, s, r, \dot{x}_c, \dot{y}_c, \dot{s})$, where (x_c, y_c) is the center point; s and r capture the area and aspect ratio, respectively; $(\dot{x}_c, \dot{y}_c, \dot{s})$ are the first-order change rate of (x_c, y_c, s) and capture the velocity information. It is worth noting that in some trackers [1, 26], the shape information s and r in the state information are replaced with height h and width w.

The motion pattern in the down-sampled video frames is more challenging to capture and predict because the temporal gap between two adjacent frames is enlarged, resulting in complex moving pattern between two frames. To mitigate the issue, we devise a more informative object state as follows:

$$(x_d, y_d, w, h, \dot{x}_d, \dot{y}_d, \dot{w}, \dot{h}, \ddot{x}_d, \ddot{y}_d, \ddot{w}, \ddot{h})$$

where (x_d, y_d) is the center point of *bottom edge* in the bounding box; (w, h) refer to width and height; $(\dot{x}_d, \dot{y}_d, \dot{w}, \dot{h})$ and $(\ddot{x}_d, \ddot{y}_d, \ddot{w}, \ddot{h})$ store the first-order and second-order change rate of (x_d, y_d, w, h), respectively.

Here, we explain the design principle of the state vector. Compared with existing state representation in MOT methods, it has two unique properties. First, we use the center point (x_d, y_d) of the *bottom edge*, rather than the center of the whole rectangle, to capture the location information The reason is that in the benchmark of MOT datasets such as MOT17 and MOT20, the target persons are moving on the flat ground. In DanceTrack, the dancers are standing on the stage. The center point of the *bottom edge* can provide more reliable position information, which is less sensitive to the variance of height. For example, if the status of a dancer changes from stand to squat, the height of the bounding box shrinks. In this case, his/her position on the stage remains the same, but the center point of the person changes.

Second, linear motion assumption tends to fail in the down-sampled MOT and we have to resort to non-linear motion model. Based on Taylor's Formula, any nonlinear function can be approximated by its Taylor expansion polynomial $f(t) = \sum_{n=0}^{\infty} \frac{f^{(n)}(t)}{n!} (\Delta t)^n$. In the scenario of down-sampled MOT, Δt in fact increases to a non-negligible value, which renders it inaccurate for previous MOT methods to represent the motion pattern only with the first-order linear approximation. Therefore, we augment the state vector with the second-order linear factor to capture more complex motion patterns.

Divergence-Aware Parameter Update. In Kalman filter, the process noise covariance matrix $\boldsymbol{Q}_k$ and the measurement noise covariance matrix $\boldsymbol{R}_k$ play very important role because they are used to regulate the impact of prediction values and measurements on the system state estimation, with the goal of minimizing their divergence. If the noise parameters are incorrect, the accuracy of Kalman filtering may be reduced dramatically.

When the video frames are down-sampled, $\boldsymbol{R}_k$ is not affected because it is mainly determined by the inherent performance of the object detector, which is YOLOX in our implementation. However, $\boldsymbol{Q}_k$ captures process noise that represent the prediction noise scale and increases due to the uncertainty of the object motion under the down-sampling. Thus, we focus on the update mechanism of $\boldsymbol{Q}_k$ and design an divergence-aware mechanism (DAM) to estimate the process noise covariance matrix $\boldsymbol{Q}_k$. We observe that abrupt change or significant variation on the motion pattern becomes more frequent in the scenario of down-sampled MOT. The traditional Kalman filter cannot immediately update the parameters to keep in line with the dramatic motion update. Our goal is explicitly enlarge the values in matrix $\boldsymbol{Q}_k$ when the uncertainty increases. In other words, when the divergence between prediction and measurement becomes large, we use it as a signal to reactively adjust the matrix $\boldsymbol{Q}_k$. Formally, we utilize the IoU of between the predicted bounding box ($\boldsymbol{H}\hat{\boldsymbol{x}}_k$ in Eq. (4)) and the observed bounding box $\boldsymbol{z}_k$ to measure the divergence between prediction and measurement.

$$\boldsymbol{Q}_k^{\dagger} = (1 + \boldsymbol{D}_{iou}(\boldsymbol{H}\hat{\boldsymbol{x}}_k, \boldsymbol{z}_k))\boldsymbol{Q}_k$$

$\boldsymbol{D}_{iou}$ is the IoU distance between two bounding boxes and defined as $\boldsymbol{D}_{iou} = 1 - IoU$, where IoU is the overlap area between two bounding boxes divided by their union area.

We also notice that fading Kalman filter (FKF), which introduces a fading factor to control the memory length and reduce the influence of the old measurement, has been shown to be a useful to improve convergence. However, to the best of our knowledge, FKF has not been adopted by state-of-the-art MOT methods. The reason is that the assumption of linear motion pattern fits well with historical data. However, in the down-sampled scenario, the temporal gap between two frames increases, the change of motion tends to be more abrupt and intense, rendering the old measurement over long time intervals less reliable. Therefore, we adopt the idea of FKF and forget the past data via fading factors. As shown in Eq. (2), we add a fading factor α_k in the transition process of the estimated noise covariance matrix $\boldsymbol{P}'_k$.

$$\boldsymbol{P}'_k = \boldsymbol{\alpha}_k \boldsymbol{F}_k \boldsymbol{P}_{k-1} \boldsymbol{F}_k^{\top} + \boldsymbol{Q}_k^{\dagger}$$

Initially, we set $\boldsymbol{\alpha} = 1.0$ and update it according to the divergence of prediction and measurement $\|\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}'_k\|$ in the subsequent iterations. When the divergence between prediction and measurement at $(k-1)^{th}$ step is large, the fading factor α_k increases, so as to reduce the influence of the old measurement.

3.3. 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) [1, 3, 4, 31] or center point distance between two bounding boxes [6, 26, 32]. 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 [33]), but not leveraged by object tracking.

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 and for each setting, we randomly collect $10,000$ bounding box association cases that can successfully solved by at least one of the following metrics, including the overlap, center point distance and aspect ratio of the bounding boxes. We denote them by IoU, DIST, and SCALE, respectively.

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. When there is no down-sampling (with $RR = 1$), it's indeed that IoU or distance-based metric demonstrate very good performance as they 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 incorporate all these three factors.

Let $\boldsymbol{D}_{iou}$ denote the overlap distance between two bounding boxes and $\boldsymbol{D}_{dist}$ denote the normalized distance between two center points of the bounding boxes

$$\boldsymbol{D}_{dist} = \frac{\|(x_d, y_d)_1, (x_d, y_d)_2\|^2}{c^2}$$

where c is is the diagonal length of the smallest enclosing box covering the bounding boxes. For the factor of aspect ratio, we define $\boldsymbol{D}_{scale}$ as

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

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 $\frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2}$ to reversely approximate for the confidence. This is a reasonable estimation because it implies that the predicted 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 $\boldsymbol{D}_{rda} = \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2}$. Otherwise, we need to incorporate $\boldsymbol{D}_{scale}$ as a complementary factor and set $\boldsymbol{D}_{rda}$ as a linear combination of the three factors.

	$RR=1$	$RR=3$	$RR=5$	$RR=7$	$RR=9$
$\lvert S_{IoU}\rvert$	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.

$$\boldsymbol{D}_{rda} = \begin{cases} \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2} & \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2} < \sigma \\ \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}+2\boldsymbol{D}_{scale}}{4} & \text{otherwise} \end{cases} \quad (6)$$

4. Experiment

In this section, we compare SR-Tracker with state-of-the-art MOT methods on three benchmark datasets. Due to space limit, selective results are presented and analyzed. Readers can refer to the supplementary materials for more comprehensive experimental evaluation.

4.1. Experimental Setup

Benchmark Datasets. We use three benchmark datasets for performance evaluation, including MOT17 [16], MOT20 [5] and DanceTrack [21]. 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, uniform appearance and diverse body gestures of dancers bring particular challenges. It provides 100 videos and the split ratio for training, validation and test dataset is $40:24:35$.

Note that the testing videos of these datasets are not annotated and their performance evaluation requires the submission of the generated tracklets to the official website. Since 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 with equal length and use them for training and testing, respectively. For DanceTrack, we use the training set for model training and report the performance on its validation set.

Performance Metrics. To evaluate the overall tracking accuracy, we adopt MOTA [2], IDF1 [19] and HOTA [13]. Generally speaking, MOTA score 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, association and localization.

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 underlying motivation is that we can adjust RR to generate a tradeoff curve between processing time and HOTA, as shown in Figure 1. It can be expected that with 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-Tracker at dataset MOT17 implies that it takes 19 seconds for SR-Tracker to process the testing videos in MOT17 with an accuracy level of $HOTA = 62$.

Comparison Methods. We compare SR-Tracker with representative and open-sourced trackers in both categories of tracking-by-detection and joint-detection-and-tracking. Among these competitors, we consider ByteTrack [31], OCSORT [4] and SimpleTrack [9] as **real-time trackers** because they can achieve as high FPS as our SR-Tracker. The remaining approaches, including TransTrack [22], TrackFormer [14], MOTR [30], StrongSORT [6] and BoT-SORT [1], are called **expensive trackers** as they exchange processing time for higher tracking accuracy.

4.2. Implementation Details

Our SR-Tracker follows the paradigm of tracking-by-detection. For object detector, we directly adopt the trained YOLOX provided by previous trackers (e.g., ByteTrack for MOT17 and MOT20, OCSORT for DanceTrack). The future positions of the detected objects are predicted by the enhanced Kalman filter (EKF) proposed in this paper. The objects between different frames are associated via the robust data association (RDA). Therefore, there is no training required for our SR-Tracker.

As to our proposed Kalman filter, we set $\alpha_0 = 1.0$ for the adaptive fading factor and adopt the previous work [1] to estimate process noise covariance matrix $\boldsymbol{Q}_k$ and measurement noise covariance matrix $\boldsymbol{R}_k$ for each frame. 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.

4.3. Comparison with Real-time Trackers

In the first experiment, we compare our SR-Tracker 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. OCSORT, ByteTrack and SR-Tracker use YOLOX as the object detector and their association ignores 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-Tracker achieves the highest MOTA, IDF1 and HOTA across all the datasets, owing to its enhanced Kalman filter for more accurate motion prediction and improved data association mechanism for more robust object inter-frame matching. The performance gap between ByteTrack and our SR-Tracker is widened when RR increases. In MOT20, the HOTA of SR-Tracker 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 pattern 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. OCSORT outperforms ByteTrack in this dataset because it is specially designed for DanceTrack and occlusion with excessive nonlinear motion. Nevertheless, they are both significantly inferior to our SR-Tracker. When $RR = 9$, we boost the HOTA from 33.4 in OCSORT to 38.9, with 16.5% improvement.

4.4. Comparison with Expensive Trackers

In Table 3, we compare SR-Tracker with the expensive trackers under $RR = 5$ and $RR = 9$. For TransTrack, TrackFormer, MOTR, StrongSORT, their performance is clearly inferior to our SR-Tracker in terms of both tracking efficiency and accuracy. BoTSORT is the only method whose tracking accuracy can be slightly better than our SR-Tracker in MOT17. However, its tracking speed is very slow and the FPS is 6 times lower than SR-Tracker. Furthermore, similar to previous findings in Table 2, the advantage of our SR-Tracker becomes more obvious when RR increases. In MOT20 with $RR = 9$, our SR-Tracker can even achieve higher accuracy than BoTSORT, with 7 times faster tracking speed. The results on DanceTrack are

	$RR = 3$			$RR = 5$			$RR = 7$			$RR = 9$			FPS
	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	
	Dataset MOT17												
SimpleTrack [9]	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
OCSORT [4]	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 [31]	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-Tracker (ours)	**67.0**	**75.7**	**78.8**	**65.9**	**72.7**	**76.3**	**64.2**	**69.6**	**73.6**	**63.1**	**67.5**	**72.4**	28.9
	Dataset MOT20												
SimpleTrack [9]	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
OCSORT [4]	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 [31]	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-Tracker (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**	17.0
	Dataset DanceTrack												
OCSORT [4]	49.2	84.6	48.7	40.8	77.8	41.6	36.1	69.7	37.9	33.4	63.1	34.8	29.0
ByteTrack [31]	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-Tracker (ours)	**53.8**	**88.1**	**53.6**	**46.6**	**84.3**	**45.4**	**42.6**	**79.5**	**40.7**	**38.9**	**74.7**	**37.2**	29.0

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

not available because we lack sufficient hardware resources to re-train these models.

	$RR = 5$			$RR = 9$			FPS
	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1	
	MOT17						
TransTrack [22]	56.8	66.1	66.6	54.8	61.2	62.2	10.0
TrackFormer [14]	59.1	66.2	68.2	55.6	60.6	64.2	7.4
MOTR [30]	59.8	65.5	68.8	56.2	61.0	65.4	7.5
StrongSORT [6]	63.6	61.9	70.9	59.3	53.2	62.9	7.1
BoTSORT [1]	**66.4**	**74.3**	**77.9**	**63.1**	**70.1**	**73.1**	4.5
SR-Tracker (ours)	65.9	72.7	76.3	**63.1**	67.5	72.4	**28.9**
	MOT20						
TransTrack [22]	31.6	47.3	44.6	30.5	44.9	42.4	7.2
TrackFormer [14]	47.4	70.6	56.8	43.3	65.3	51.3	4.1
MOTR [30]	42.8	50.6	56.1	38.0	43.0	49.7	4.2
StrongSORT [6]	56.5	67.4	72.8	50.7	61.2	66.6	1.4
BoTSORT [1]	**57.7**	71.1	73.9	54.0	67.2	69.3	2.4
SR-Tracker (ours)	57.6	**71.3**	**74.1**	**55.8**	**68.9**	**71.2**	**17.0**

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

4.5. Trade-off Between Efficiency and Accuracy

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 2. 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. OCSORT outperforms ByteTrack in dataset DanceTrack because OCSORT is better at capturing complex motion patterns. StrongSORT and BoTSORT utilize visual similarity by extracting appearance features and achieve 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, object ReID and motion prediction in the same framework. Its performance is not satisfactory due to insufficient annotated samples for training and its online inference cost is also expensive.

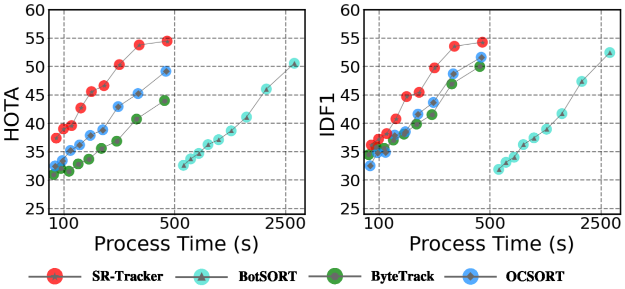

Figure 2. Trade-off analysis in DanceTrack.

We also study the performance of these trackers in terms of our proposed metric Time@HOTA. As reported in Tables 4 and 5, ByteTrack achieves the best Time@HOTA among the comparison trackers in MOT17. Our SR-Tracker can further reduce its processing time by half with a given HOTA requirement. For example, it takes SR-Tracker 19s to generate tracking results in MOT17 with HOTA=62, whereas ByteTrack requires 37.3s. In dataset DanceTrack, the advantage of SR-Tracker in terms of Time@HOTA is enlarged to 3x. If there are a large number of dancing video streams that require real-time tracking, it can estimated that our SR-Track only consumes roughly $1/3$ of GPU devices to achieve the same performance as ByteTrack.

4.6. Ablation Study

We evaluate the advantage brought by the enhanced Kalman filter (EKF) and robust data association (RDA) in Table 6. Overall, the positive effect brought by EKF is more significant than that from RDA. We also conduct a break-

	Time@66	Time@65	Time@64	Time@63	Time@62
TrackFormer [14]	718.3	718.3	718.3	718.3	718.3
MOTR [30]	708.8	708.8	708.8	708.8	317.1
TransTrack [22]	531.6	531.6	531.6	531.6	531.6
SimpleTrack [9]	235.9	235.9	235.9	235.9	235.9
StrongSORT [6]	193.5	167.1	166.2	120.4	112.3
BoTSORT [1]	195.1	157.3	156.6	136.8	117.2
OCSORT [4]	183.3	94.6	92.6	58.2	45.1
ByteTrack [31]	91.7	62.3	61.5	48.3	37.3
SR-Tracker	**45.1**	**31.9**	**31.6**	**22.2**	**19.0**

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

	Time@53	Time@50	Time@47	Time@44	Time@41
BoTSORT [1]	3989.8	2890.2	2036.1	1662.6	1405.1
OCSORT [4]	879.6	577.5	345.5	249.2	196.7
MOTR [30]	621.8	384.9	295.1	248.4	214.4
ByteTrack [31]	861.8	861.8	861.8	432.1	295.6
SR-Tracker	**270.6**	**229.5**	**182.8**	**136.8**	**118.7**

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

down analysis on the components of EKF and examine the effect of our proposed informative state representation (ISR), divergence-aware mechanism (DAM) and fading Kalman filter (FKF) We can see that ISR makes an important contribution as it can better fit the non-linear motion pattern. Both AFF and DAM are also contributing to the improvement of tracking accuracy as they can adaptively update the parameter matrices in Kalman filter.

	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
SR-Tracker	**64.9**	**63.3**	**67.0**	**71.4**	**75.1**
SR-Tracker w/o EKF	62.5	61.1	64.9	69.7	73.6
SR-Tracker w/o RDA	64.0	62.9	65.6	71.0	73.6
Break-down analysis on EKF					
EKF w/o ISR	63.6	62.0	66.1	70.9	74.7
EKF w/o DAM	64.3	62.6	66.7	70.7	74.8
EKF w/o FKF	64.4	63.1	66.2	70.9	74.2

Table 6. Ablation study of our method on the MOT17 dataset.

Table 7 presents an alternative way to validate the effectiveness of our proposed Kalman filter, by applying our proposed Kalman filter on existing trackers. The results are encouraging — we can observe performance improvement on all the metrics in dataset MOT17.

	HOTA+Δ	MOTA+Δ	IDF1+Δ
SimpleTrack [9]	$57.9 + 1.2$	$65.2 + 1.5$	$72.9 + 1.3$
OCSORT [4]	$60.3 + 0.6$	$62.5 + 1.1$	$69.4 + 1.2$
ByteTrack [31]	$61.6 + 2.4$	$69.4 + 1.6$	$72.1 + 1.6$
StrongSORT [6]	$63.4 + 0.8$	$62.2 + 0.4$	$69.9 + 0.8$
BoTSORT [1]	$66.0 + 1.1$	$73.4 + 0.5$	$77.2 + 0.7$

Table 7. Applying our Kalman filter on other trackers in MOT17.

4.7. Experiments Without Down-Sampling

We are also curious to examine the performance of our SR-Tracker on the original dataset without down-sampling. Table 8 shows the results returned by the leadboard 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 [32]	39.7	66.7	23.8	82.2	40.8
QDTrack [17]	45.7	72.1	29.2	83.0	44.8
TransTrack [22]	45.5	75.9	27.5	88.4	45.2
MOTR [30]	48.4	71.8	32.7	79.2	46.1
ByteTrack [31]	47.3	71.6	31.4	89.5	52.5
OCSORT [4]	55.1	80.3	38.0	89.4	54.2
SR-Tracker	**59.1**	**81.5**	**42.9**	**92.4**	**59.3**

Table 8. Performance on DanceTrack test dataset.

5. Case Analysis

Finally, we perform a case analysis by comparing SR-Tracker and ByteTrack on MOT17 with $RR = 9$. As shown in Figure 3, we highlight the incorrect association generated by ByteTrack. From frame 4 to frame 5, its Kalman filter makes wrong prediction of the next bounding box, whereas our enhanced Kalman filter delivers accurate prediction. From frame 16 to frame 17, ByteTrack incurs ID switching caused by occlusion, but our SR-Tracker, with more robust association metric, is able to resolve the issue.

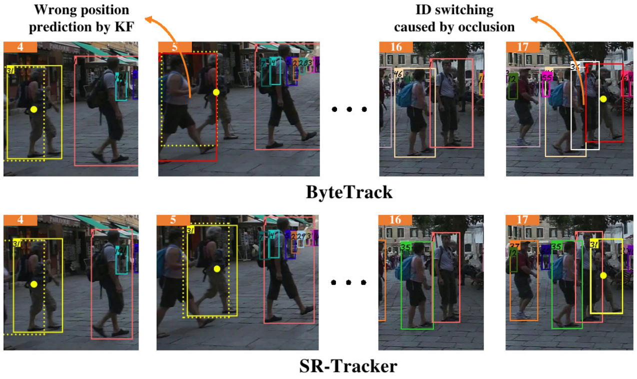
Wrong position prediction by KF
ID switching caused by occlusion
4
5
16
17
ByteTrack
4
5
16
17
SR-Tracker

Figure 3. A case study for ByteTrack and SR-Tracker in MOT17.

6. 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 an enhanced Kalman filter for accurate motion prediction and a comprehensive data association metric for robust inter-frame matching. Experiments on three benchmark datasets show that our proposed SR-Tracker establishes new SOTA performance for real-time tracking.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Botsort: Robust associations multi-pedestrian tracking. *CoRR*, abs/2206.14651, 2022. 2, 4, 5, 6, 7, 8

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008, 2008. 6

[3] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 3464–3468. IEEE, 2016. 2, 5

[4] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric SORT: rethinking SORT for robust multi-object tracking. *CoRR*, abs/2203.14360, 2022. 2, 5, 6, 7, 8

[5] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv: Computer Vision and Pattern Recognition*, 2020. 5

[6] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. Strongsort: Make deepsort great again. *CoRR*, abs/2202.13514, 2022. 2, 3, 5, 6, 7, 8

[7] Yunhao Du, Junfeng Wan, Yanyun Zhao, Binyu Zhang, Zhihang Tong, and Junhao Dong. Giaotracker: A comprehensive framework for MCMOT with global information and optimizing strategies in visdrone 2021. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, pages 2809–2819. IEEE, 2021. 3

[8] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2

[9] Jiaxin Li, Yan Ding, Hua-Liang Wei, Yutong Zhang, and Wenxiang Lin. Simpletrack: Rethinking and improving the JDE approach for multi-object tracking. *Sensors*, 22(15):5863, 2022. 3, 6, 7, 8

[10] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *CoRR*, abs/2010.12138, 2020. 3

[11] Shengzhong Liu, Tianshi Wang, Jinyang Li, Dachun Sun, Mani B. Srivastava, and Tarek F. Abdelzaher. Adamask: Enabling machine-centric video streaming with adaptive frame masking for DNN inference offloading. In João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni, editors, *ACM MM*, pages 3035–3044. ACM, 2022. 1

[12] Wei-Lwun Lu, Jo-Anne Ting, James J. Little, and Kevin P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1704–1716, 2013. 1

[13] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021. 6

[14] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8834–8844. IEEE, 2022. 3, 6, 7, 8

[15] Djamal Merad, Kheir-Eddine Aziz, Rabah Iguernaissi, Bernard Fertil, and Pierre Drap. Tracking multiple persons under partial and global occlusions: Application to customers' behavior analysis. *Pattern Recognit. Lett.*, 81:11–20, 2016. 1

[16] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv: Computer Vision and Pattern Recognition*, 2016. 5

[17] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 164–173. Computer Vision Foundation / IEEE, 2021. 8

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[19] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In Gang Hua and Hervé Jégou, editors, *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, volume 9914 of *Lecture Notes in Computer Science*, pages 17–35, 2016. 6

[20] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Tracklets predicting based adaptive graph tracking. *arXiv preprint arXiv:2010.09015*, 2020. 3

[21] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 20961–20970. IEEE, 2022. 5

[22] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *CoRR*, abs/2012.15460, 2020. 3, 6, 7, 8

[23] Wei Tian, Martin Lauer, and Long Chen. Online multi-object tracking using joint domain information in traffic scenarios. *IEEE Trans. Intell. Transp. Syst.*, 21(1):374–384, 2020. 1

[24] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 13708–13715. IEEE, 2021. 3

[25] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking.

In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2020. 2

[26] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*, pages 3645–3649. IEEE, 2017. 2, 4, 5

[27] Ronghua Xu, Seyed Yahya Nikouei, Yu Chen, Aleksey Polunchenko, Sejun Song, Chengbin Deng, and Timothy R. Faughnan. Real-time human objects tracking for smart surveillance at the edge. In *2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20-24, 2018*, pages 1–6. IEEE, 2018. 1

[28] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *CoRR*, abs/2103.15145, 2021. 3

[29] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *CoRR*, abs/2105.04322, 2021. 3

[30] Fangao Zeng, Bin Dong, Tiancai Wang, Cheng Chen, Xiangyu Zhang, and Yichen Wei. MOTR: end-to-end multiple-object tracking with transformer. *CoRR*, abs/2105.03247, 2021. 6, 7, 8

[31] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *CoRR*, abs/2110.06864, 2021. 2, 5, 6, 7, 8

[32] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129(11):3069–3087, 2021. 3, 5, 8

[33] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 12993–13000. AAAI Press, 2020. 5

[34] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*, volume 12349 of *Lecture Notes in Computer Science*, pages 474–490. Springer, 2020. 3