Sampling-Resilient Multi-Object Tracking for Efficient Video Query Processing [Scalable Data Science]

Zepeng Li
Zhejiang University, China
lizepeng@zju.edu.cn

Dongxiang Zhang
Zhejiang University, China
zhangdongxiang@zju.edu.cn

Sai Wu
Zhejiang University, China
wusai@zju.edu.cn

Mingli Song
Zhejiang University, China
brooksong@zju.edu.cn

Kian-Lee Tan
National University of Singapore
tankl@comp.nus.edu.sg

Gang Chen
Zhejiang University, China
cg@zju.edu.cn

ABSTRACT

The success of OTIF, a video database with cutting-edge efficiency, is contingent upon the utilization of multi-object tracking (MOT) as a pre-processing step to extract trajectories of moving objects from videos and construct offline indexes to facilitate online query processing. By leveraging temporal redundancy in video data, OTIF adopts down-sampling to reduce frame processing cost. However, we observe that OTIF, as well as other MOT models proposed within the computer vision community, exhibit a substantial decline in accuracy when confronted with a low sampling rate. This observation serves as a catalyst for our investigation into sampling-resilient multi-object tracking, aiming to achieve a superior balance between efficiency and accuracy for video query processing.

In this paper, we devise a sampling-resilient tracker called SR-Track with more accurate motion estimation and robust data association. Its key components include a Kalman filter tailored for sparse observations and a comprehensive similarity metric that systematically integrates multiple spatial matching signals. We conduct extensive experiments to compare SR-Track with OTIF and 8 alternative MOT strategies on 5 benchmark datasets and evaluate performance in terms of object tracking and video query processing, including selection, aggregation and top-k queries. The results clearly establish the superiority of SR-Track over existing MOT methods when supporting video query processing.

PVLDB Reference Format:
Zepeng Li, Dongxiang Zhang, Sai Wu, Mingli Song, Kian-Lee Tan, and Gang Chen. Sampling-Resilient Multi-Object Tracking for Efficient Video Query Processing [Scalable Data Science]. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/lzzppp/SR-Track.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

1 INTRODUCTION

The proliferation of smart cities has resulted in a remarkable upsurge in the deployment of video cameras, which play a pivotal role in fortifying public safety and optimizing public transportation. These cameras produce continuous flows of live video streams that constitute a massive video database for analytical queries and insight extraction through the utilization of increasingly accurate machine learning models. To address the efficiency issue, scalable video query optimization has re-emerged as an attractive research topic in recent years and a noticeable number of video database systems has been proposed.

Since the primary performance bottleneck originates from the computationally intensive inference overhead incurred by deep learning models, the prevailing approach is to construct fast proxy models to replace the the time-consuming oracle model, while accepting a tolerable level of accuracy degradation. The idea has been widely embraced and implemented within systems like NoScope [20], FOCUS [15], TAHOMA [2], ABAE [21], Everest [24] and FiGO [8]. Alternatively, we can leverage the inherent temporal redundancy within successive video frames to reduce the number of processed frames, either by down-sampling [4, 6] or constructing a lightweight binary classifier to skip irrelevant frames [20, 33]. In FOCUS [15] and Video-zilla [16], frame clustering and inverted index are utilized to improve efficiency. The video frames are first ingested with cheap convolutional neural networks to extract class labels. Subsequently, semantically similar frames are clustered and inverted index is built to reduce the search space of video frame retrieval queries.

Among these systems, OTIF has introduced a new paradigm in the realm of video query processing. It harnesses multi-object tracking (MOT) as a pre-processing step to extract the trajectories of all moving objects portrayed in video footage. With the extracted trajectories, offline indexes are constructed to facilitate online query processing with sub-second latency. For instance, the retrieval of video frames containing both an ambulance and a firetruck can be effortlessly achieved through a simple intersection operation between the inverted lists corresponding to these two labels. Similarly, estimating traffic flow within a specific time period can be accomplished by leveraging the spatial-temporal index to identify relevant video frames and aggregating the count of distinct objects contained within those frames. These features render OTIF as the sole system capable of handling a diverse range of queries and exhibits cutting-edge efficiency, as illustrated in Table 1, while the

Table 1: Experimental results collected from [6] to demonstrate the superior efficiency of OTIF over exsiting video databases. Readers can refer to [6] for more details of query setup and experimental analysis.

Methods	Conference	Query-1	Query-2	Query-3
NoScope [20]	PVLDB	990s	666s	-
CaTDet [28]	SysML	601s	564s	-
Chameleon [17]	SIGCOMM	209s	44s	-
MIRIS [5]	SIGMOD	2665s	320s	-
BlazeIt [19]	CIDR	-	-	665s
TASTI [22]	SIGMOD	-	-	929s
OTIF [6]	SIGMOD	**40**s	**25**s	**101**s

remaining systems are tailored to optimize a specific category of video queries.

In this paper, we adopt the video query processing paradigm proposed by OTIF, i.e., leveraging an MOT model for offline video ingestion and index construction. Our contributions stem from the following two observations.

Observation 1: Multi-object tracking is a crowded research topic in the computer vision community. State-of-the-art MOT models have not been explored by the database community to support video query processing.

OTIF introduces a recurrent reduced-rate tracking strategy. Initially, a segmentation proxy model is proposed to avoid applying the object detection model on the whole video frame. Instead, only local regions that potentially intersect with the bounding boxes of moving objects are extracted to perform object detection. With the detected objects from each sampled video frame, a Recurrent Neural Network (RNN) is employed to generate track-level features and derive matching scores. Although OTIF has been compared to CenterTrack [47], it is crucial to note that several other MOT models, which exhibit superior performance over CenterTrack, have not been examined. Our first contribution in this study is to address this gap by incorporating a comprehensive comparison with 8 other popular MOT models recently proposed. By doing so, we aim to bridge the consensus gap between the database community and the computer vision community. Surprisingly, our findings reveal that replacing the inherent MOT model in OTIF with state-of-the-art MOT models can significantly enhance its efficiency without any degradation in accuracy.

Observation 2: Substantial accuracy decline of OTIF and existing MOT models under low sampling rate.

This observation serves as a catalyst for our investigation into sampling-resilient multi-object tracking. Our aim is to address the scenario of down-sampled multi-object tracking by achieving comparable levels of accuracy while processing significantly fewer video frames. This task poses a significant challenge due to the increasing difficulty in capturing motion patterns and the diminished accuracy of position estimation in subsequent frames. Moreover, the data association strategy employed by existing MOT models, which performs well in dense frame scenarios, becomes inadequate when confronted with sparse frames.

To devise a more sampling-resilient MOT model, we propose SR-Track with more accurate motion estimation and robust data association mechanisms. First, to estimate the next position of an object more accurately with sparse observations, we augment Kalman filter (KF) with more informative state representation to capture the evolution of the object motion state. By measuring the divergence between observation and internal state estimation, we can dynamically estimate the noise scale of the motion process under sparse observations. Furthermore, we propose an aligned state update mechanism that enhances the adaptability of the motion model to new observations. Second, to robustly associate detected bounding boxes under widening intervals, we reveal interesting findings from an experimental analysis on existing metrics. We propose a comprehensive similarity metric that integrates multiple spatial matching clues, including overlap, center point distance and aspect ratio of the bounding boxes.

We also notice alternative efforts dedicated to improving KF for more accurate motion estimation. For example, StrongSORT [12] and GiaoTracker [13] adaptively modulate the observation noise scale according to the quality of object detection. In OC-SORT [9], the authors claim that the assumption of consistent velocity direction does not hold due to the non-linear motion of objects and state noise. They also argue that non-linear variants such as Extended KF [18] and Unscented KF [35] are very difficult to implement for online tracking due to the lack of prior knowledge on the complex motion pattern. Therefore, they design an update strategy under occlusion to reduce noise and add the velocity consistency (momentum) term into the cost matrix for better matching between tracklets and observations. *However, these approaches focus on reducing noise with frequent observations. We will experimentally show that they fail to work well in down-sampled MOT with sparse observations.*

To comprehensively evaluate SR-Track, we conduct extensive experiments by comparing it with OTIF and 8 alternative MOT strategies across 5 video datasets. Our investigation encompasses various aspects of object tracking and video query processing, including selection, aggregation, and top-k queries. The findings of our study highlight ByteTrack [43] as the most efficient MOT tracker. Notably, when replacing the MOT module of OTIF with ByteTrack, we observe a remarkable improvement in efficiency. Our proposed SR-Track emerges as the superior approach, as it achieves a speedup of around 2x compared to ByteTrack while maintaining an equivalent level of accuracy.

2 RELATED WORK

In this section, we review video query optimization systems from the database community and multi-object tracking models from the computer vision community.

2.1 On-the-fly Video Query Optimization

NoScope [20] is a pioneering work that leverages smaller yet faster specialized networks to enhance model inference speed. Moreover, it takes advantage of redundancy in neighboring frames and trains a difference detector for irrelevant frame filtering. The idea of proxy model design to achieve better tradeoff between efficiency and accuracy has been widely adopted by subsequent works, including FOCUS [15], TAHOMA [2], ABAE [21], Everest [24] and FiGO [8].

For example, TAHOMA replaces the accurate-but-expensive CNN with cascades of fast image classifiers. It constructs numerous specialized candidate binary-classification CNN models and identifies a set of Pareto-optimal cascades with varying trade-offs between accuracy and throughput. Everest [24] trains a lightweight convolutional mixture density network (CMDN) to generate an approximate score distribution for each frame. Subsequently, it utilizes uncertain query processing to accelerate top-K analytics while providing probabilistic guarantees. Another approach to enhance performance speed is the adoption of downsampling techniques to reduce the number of processed frames. MIRIS [4] and OTIF [6] adopt uniform downsampling to efficient object tracking. ExSampler [31] extends the sampling strategy from uniform downsampling to importance sampling.

To reduce the manual efforts of tuning configuration parameters, such as video resolution, sampling rate and the choice of deep learning model instances, automatic configuration of the video query processing has garnered considerable attention. These works are similar to automatic database tuning in traditional RDBMS, such as CDBTune [42], QTune [25]. Zerus [10] adopts reinforcement learning to indicate the relevant video segments and the sampling rate to control the tradeoff between efficiency and accuracy. SMOL [23] considers input image size as a key factor. With smaller input image size, the image processing time and model inference cost can be reduced.

2.2 Index-Assisted Video Query Optimization

To support video frame selection queries for target objects, FOCUS [15] and Video-zilla [16] first ingest video frames with cheap convolutional neural networks to extract class labels. Subsequently, semantically similar frames are clustered and inverted index is built to improve efficiency. When supporting aggregation queries, BlazeIt [19] consists of an ingestion step to randomly sample a subset of video frames and annotate them using the expensive CNN models. The images as well as their labels will be used to train a specialized NN to estimate the statistic, whose variance is further reduced via the technique of control variates [3]. In OTIF [6], the goal of the offline ingestion is to extract all the object tracks in an efficient and accurate fashion. Two optimization techniques, including segmentation proxy models and recurrent reduced-rate tracking, are developed for speed acceleration. With the extracted tracks, common selection and aggregation queries can be answered with sub-second latency.

2.3 Multi-Objet Tracking Models

SORT [7], DeepSORT [37], OC-SORT [9], StrongSORT [12], BoT-SORT [1] and ByteTrack [43] are representative tracking-by-detection methods, which 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 [32] as the default detector, which is replaced by YOLOX [14] 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, they adopt vanilla 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 [37], StrongSORT and BoT-SORT integrate appearance features into the tracker, which requires additional computation cost to derive visual embedding.

JDE [36] allows object detection and appearance embedding to be learned in a single network. However, its shared network is biased towards the detector task and unfair to the ReID task. To resolve the competition issue, CSTrack [27] devises a cross-correlation network to learn task-dependent representations. RelationTrack [40] presents global context disentangling (GCD) to decouple th learned features in the two tasks. FairMOT [44] 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 [26] 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, CenterTrack [46] and TransCenter [39] attempt to further incorporate the estimation of inter-frame object motion in the training framework. The models are trained to minimize the regression loss of the object offset between adjacent frames. Recently, TrackFormer [29] adopts the concept of track queries and employs the attention mechanism to track the objects in an autoregressive fashion. In the current stage, these jointly trained trackers are computation expensive to achieve high accuracy and not suitable for real-time tracking.

3 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 estimation 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 estimation 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 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 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 P'_k is also updated according to the Kalman gain K_k.

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

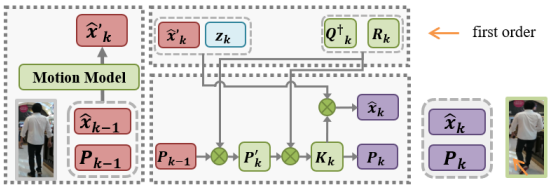
$$\hat{x}_k = \hat{x}'_k + K_k \left(z_k - H\hat{x}'_k\right) \quad (4)$$

$$P_k = (I - K_k H) P'_k \quad (5)$$

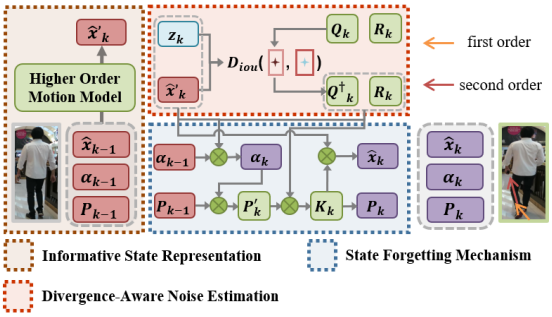
In the scenario of down-sampled MOT, the observations become sparse and each object appears in fewer number of video frames. Consequently, the uncertainty is amplified and it becomes more challenging to capture the model pattern. The traditional KF as well as its improved variants in StrongSORT and OC-SORT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

3.1 Sparse-Observation Kalman Filter

The pipeline of our proposed Sparse-Observation Kalman Filter (SOKF) is illustrated in Figure 1, with the following three key components.



(a) Original Kalman filter



(b) Sparse-Observation Kalman filter

Figure 1: Pipelines of KF and SOKF.

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, 37], the shape information s and r in the state information are replaced with height h and width w. When observations are sparse, even minor changes in an object's position can result in significant and complex alterations over time. The original state representation becomes inadequate and 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 principles of the state vector. First, we use the center point (x_d, y_d) of the *bottom edge*, rather than the center of the whole rectangle. This is a dataset-oriented trick to leverage the fact that the target persons are often moving on the flat ground. Take DanceTrack as an example, the dancers are moving on the stage and the center point of the *bottom edge* can provide more reliable position information, which is less sensitive to the variance of height. When the status of a dancer changes from stand to squat, the height of the bounding box shrinks, but our definition of center point remains the same.

Secondly, existing solutions utilized a first-order change rate for motion tracking, which we think is not sufficient to capture the rapid position update and complex motion pattern in down-sampled MOT. Our strategy is to incorporate higher order change rates to acquire more informative motion representation. Since more dimensions of state will trigger higher computational and tuning difficulties, we only augment the state with 4-dimensional second-order change rate, in order to strike a balance between performance and cost.

Divergence-Aware Noise Estimation. In Kalman filter, the process noise covariance matrix Q_k and the observation noise covariance matrix R_k play very important role because they are used to regulate the impact of prediction and observation on the system state estimation, with the goal of maximizing a posterior estimation. If the noise parameters are incorrect, the accuracy of KF may reduce dramatically.

When the video frames are down-sampled, 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, Q_k captures motion process noise that represent the estimation noise scale and increases due to the amplified uncertainty in the scenario of sparse observations. Thus, we focus on the update mechanism of Q_k and design an divergence-aware mechanism (DAM) to estimate the process noise covariance matrix 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 change. Our goal is explicitly enlarge the values in matrix Q_k when the uncertainty increases. In other words, when the divergence between estimation and observation becomes large, we use it as a signal to reactively adjust the matrix Q_k. Formally, we utilize the IoU of between the estimated bounding box ($H\hat{x}_k$ in Eq. (4)) and the observed bounding box z_k to measure the divergence between estimation and observation.

$$Q_k^\dagger = (1 + D_{iou}(H\hat{x}_k, z_k))Q_k \quad (6)$$

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

State Forgetting Mechanism. KF captures the noise distribution of the system via updating the state estimation covariance matrix $\boldsymbol{P}_{k-1}$, which determines the influence of the historical state. However, in down-sampled MOT with sparse observations, the historical state estimation becomes less reliable and we should pay more attention to recent observations. Inspired by fading KF [38], we incorporate a state forgetting mechanism into the updating process of $\boldsymbol{P}_{k-1}$. As shown in Eq. (7), we add a fading factor α_k in the transition process of the estimated noise covariance matrix $\boldsymbol{P}'_k$.

$$\boldsymbol{P}'_k = \boldsymbol{\alpha}_k \boldsymbol{F}_k \boldsymbol{P}_{k-1} \boldsymbol{F}_k^\top + \boldsymbol{Q}_k^\dagger \tag{7}$$

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

3.2 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, 7, 9, 43] or center point distance between two bounding boxes [12, 37, 44]. 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 [45]), but they are 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 be 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 2. 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.

$$\boldsymbol{D}_{dist} = \frac{\|(x_d, y_d)_1, (x_d, y_d)_2\|^2}{c^2} \tag{8}$$

Table 2: distance metrics analysis on the MOT17 dataset.

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

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

$$\boldsymbol{D}_{scale} = \frac{4}{\pi^2}\left(\arctan\frac{w_1}{h_1} - \arctan\frac{w_2}{h_2}\right)^2 \tag{9}$$

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 2, 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 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 $\boldsymbol{D}_{rda} = \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2}$. 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.

$$\boldsymbol{D}_{rda} = \begin{cases} \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2} & \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}}{2} < \sigma \\ \frac{\boldsymbol{D}_{dist}+\boldsymbol{D}_{iou}+2\boldsymbol{D}_{scale}}{4} & \text{otherwise} \end{cases} \tag{10}$$

3.3 Video Processing Workflow

Given the proposed modules of sparse-observation Kalman filter (SOKF) and robust data association (RDA), we present the complete workflow that transforms input video clips or video streaming data into offline indexes to facilitate video query optimization. As shown in Figure 2, uniform down-sampling is first applied to select a subset of frames for multi-object tracking. Given two neighboring frames F_i and F_{i+1}, an existing object detection model is applied on both frames. For the detected objects, we can obtain their bounding boxes as well as class labels. Then, we apply our proposed SOKF on the detected objects in frame F_i to estimate their future positions in frame F_{i+1}. Subsequently, RDA is estimate the similarity between a detected bounding box in F_{i+1} and the predicted box from the preceding frame F_i by SOKF. With the similarity scores, we can apply Hungarian algorithm to generate matching trajectories. Therefore, there is no training required for our SR-Track.

4 EXPERIMENT

4.1 Experimental Setup

Benchmark Datasets. We use 5 real video datasets for performance evaluation. Among them, MOT17 [30] and MOT20 [11] are two popular benchmark datasets for multi-object tracking. 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

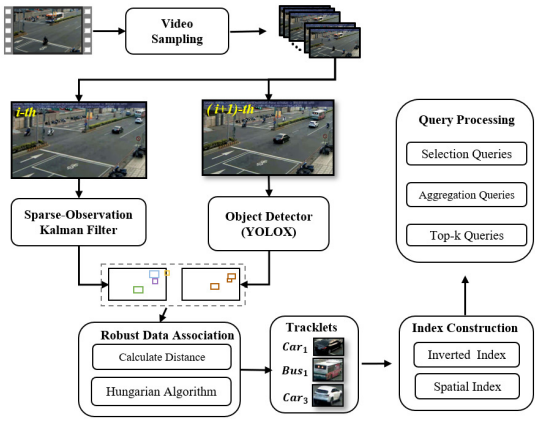


Figure 2: The pipeline of SR-Track for video ingestion.
-5mm

train stations, town squares and a sports stadium. For video query evaluation, we use three datasets used in previous VDBMS with multifaceted traffic scenarios and diverse weather conditions, including Jackson Town [6], Square Northeast [20], and Taipei [20]. The time span of each dataset is one hour. Jackson Town and Square Northeast are filmed during nocturnal and snowy conditions, respectively and they offer complex intersection scenarios. Taipei is a dual-lane scene teeming with occlusions of vehicles.

Video Queries. As depicted in Table 3, we craft 5 video queries, with 2 selection queries, 2 aggregation queries and 1 top-k query.

Table 3: Video queries.

Query ID	Query definition
Q_1	Select video frames that contain "car".
Q_2	Select video frames that contain "truck".
Q_3	Count the number of "cars" in every minute.
Q_4	Count the number of "trucks" in every minute.
Q_5	Find 50 most densest frame clips. Each clip contains 100 frames.

Comparison Methods. We conduct performance evaluation on two tasks, including multi-object tracking and video query processing. We compare SR-Track with OTIF and 8 other open-sourced trackers recently proposed in the computer vision community, including ByteTrack [43], SimpleTrack [26], OC-SORT [9], BoT-SORT [1], StrongSORT [12], TransTrack [34], MOTR [41] and TrackFormer [29].

Note that we did not include other VDBMS as comparison approaches because OTIF [6] has extensively demonstrated its performance and efficiency against numerous video query optimization approaches [4, 15, 20]. In this paper, we treat OTIF as the state-of-the-art video database system and focus on the comparison with MOT families.

Performance Metrics. To assess the overall tracking accuracy, we use MOTA, IDF1, and HOTA metrics. Generally speaking, MOTA leans towards evaluating object detection quality, IDF1 highlights accurate association, and HOTA is a recent metric that balances detection, association, and localization. We also introduce a new metric, **Time@HOTA**, to better display processing efficiency. This metric allows us to adjust RR to show a balance between processing time and HOTA. For instance, $Time@62 = 19$ for SR-Track on the MOT17 dataset suggests that it takes 19 seconds for SR-Track to process the test videos with a HOTA accuracy level of 62.

Regarding video query processing, we employ the F_1-score as the performance metric for selection queries, given that the ground truth comprises a set of target frames. In the case of aggregation queries, which involve counting the number of objects within a time window, we adopt the mean absolute error (MAE) as our evaluation criterion. Finally, $R@k$ is used for top-k queries to measure the percentage of correctly retrieved frames.

4.2 Implementation Details

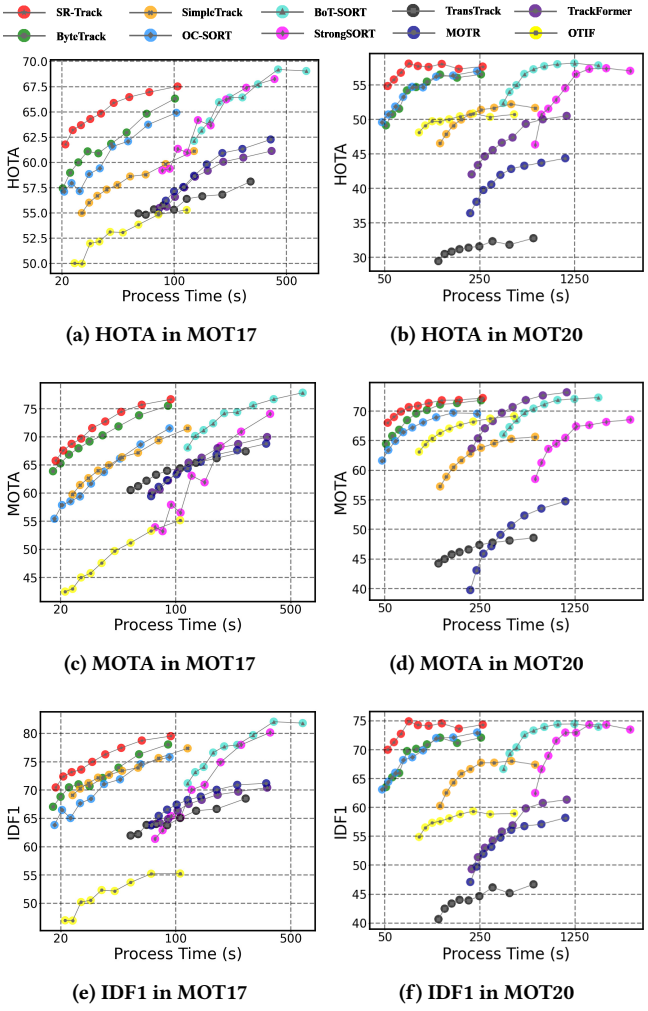
Our SR-Track follows the paradigm of tracking-by-detection. We adopt YOLOX [14] as the object detector. As to our proposed Kalman filter, we set $\boldsymbol{\alpha}_0 = 1.0$ for the adaptive fading factor. Since OTIF is implemented with other optimizations such as a joint parameter tuning module, we adopt the OTIF framework to report the results of video query processing for SR-Track and other MOT models. In other words, we replace the tracking module of OTIF with the selected MOT model to ensure a fair comparison. 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.

4.3 Performance on Multi-Object Tracking

In the first experiment, we compare SR-Track with other trackers including OTIF under varying frame reduction ratios (RR set from 2 to 10). As shown in Figure 3, OC-SORT, ByteTrack, OTIF and SR-Track demonstrate similar inference speed. They use off-the-shelf 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 speed is slightly lower than other real-time trackers. Among these efficiency trackers, SR-Track achieves the highest MOTA, IDF1 and HOTA across all the datasets, owing to its Kalman filter 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 $RR = 9$.

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 speed is 6 times lower than SR-Track. Furthermore, the advantage of SR-Track becomes more obvious when RR increases. In MOT20, with RR greater than 6, our SR-Track can even achieve higher accuracy than BoT-SORT with over 5× speedup,

It's also astonishing to find that the tracking performance of OTIF is not competitive and falls significantly behind multiple existing MOT models. This is mainly because its tracking mechanism is not accurate. With the same frame reduction ratio, OTIF yields



(a) HOTA in MOT17 (b) HOTA in MOT20

(c) MOTA in MOT17 (d) MOTA in MOT20

(e) IDF1 in MOT17 (f) IDF1 in MOT20

Figure 3: Multi-Object Tracking result in MOT17 and MOT20 datasets (better viewed in color).

the lowest scores in terms of HOTA, MOTA, and IDF1 across the MOT17 dataset.

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

	Time@66	Time@65	Time@64	Time@63	Time@62
TrackFormer [29]	> 808.2	> 808.2	> 808.2	> 808.2	> 808.2
MOTR [41]	> 797.5	> 797.5	> 797.5	> 797.5	317.1
OTIF [6]	> 239.2	> 239.2	> 239.2	> 239.2	> 239.2
TransTrack [34]	> 598.1	> 598.1	> 598.1	> 598.1	> 598.1
SimpleTrack [26]	> 265.5	> 265.5	> 265.5	> 265.5	> 265.5
StrongSORT [12]	193.5	167.1	166.2	120.4	112.3
BoT-SORT [1]	195.1	157.3	156.6	136.8	117.2
OC-SORT [9]	183.3	94.6	92.6	58.2	45.1
ByteTrack [43]	91.7	62.3	61.5	48.3	37.3
SR-Track	**45.1**	**31.9**	**31.6**	**22.2**	**19.0**

As a convenient quantitative evaluation metric that captures the tradeoff between tracking efficiency and accuracy, we present

Table 5: Time@HOTA in MOT20 (in seconds).

	Time@55	Time@52	Time@49	Time@46	Time@43
TrackFormer [29]	> 2178.3	> 2178.3	797.7	440.0	284.2
MOTR [41]	> 2126.4	> 2126.4	> 2126.4	> 2126.4	1137.7
StrongSORT [12]	1376.7	1005.2	771.5	714.5	665.3
SimpleTrack [26]	> 1275.9	> 1275.9	195.6	143.3	115.4
TransTrack [34]	> 1240.4	> 1240.4	> 1240.4	> 1240.4	> 1240.4
OTIF [6]	> 893.1	> 893.1	119.3	91.5	73.5
BoT-SORT [1]	546.3	414.1	346.0	297.1	260.4
OC-SORT [9]	132.8	72.2	52.7	41.1	33.6
ByteTrack [43]	119.6	76.4	58.3	48.3	41.3
SR-Track	**61.5**	**45.3**	**35.8**	**29.7**	**25.3**

the Time@HOTA metric for the multi-object tracking methods in Tables 4 and 5. The result with '>' means that the method cannot reach the specified HOTA even when there is no down-sampling. In this case, we report the running time without sampling as the lower bound. ByteTrack outperforms the comparison trackers in MOT17 by achieving the best Time@HOTA. This accomplishment is attributed to its simple yet effective tracking paradigm. Compared to OTIF, ByteTrack demonstrates a significant 7-fold speedup while attaining HOTA=66 in MOT17, and a corresponding 7-fold speedup for HOTA=55 in MOT20.

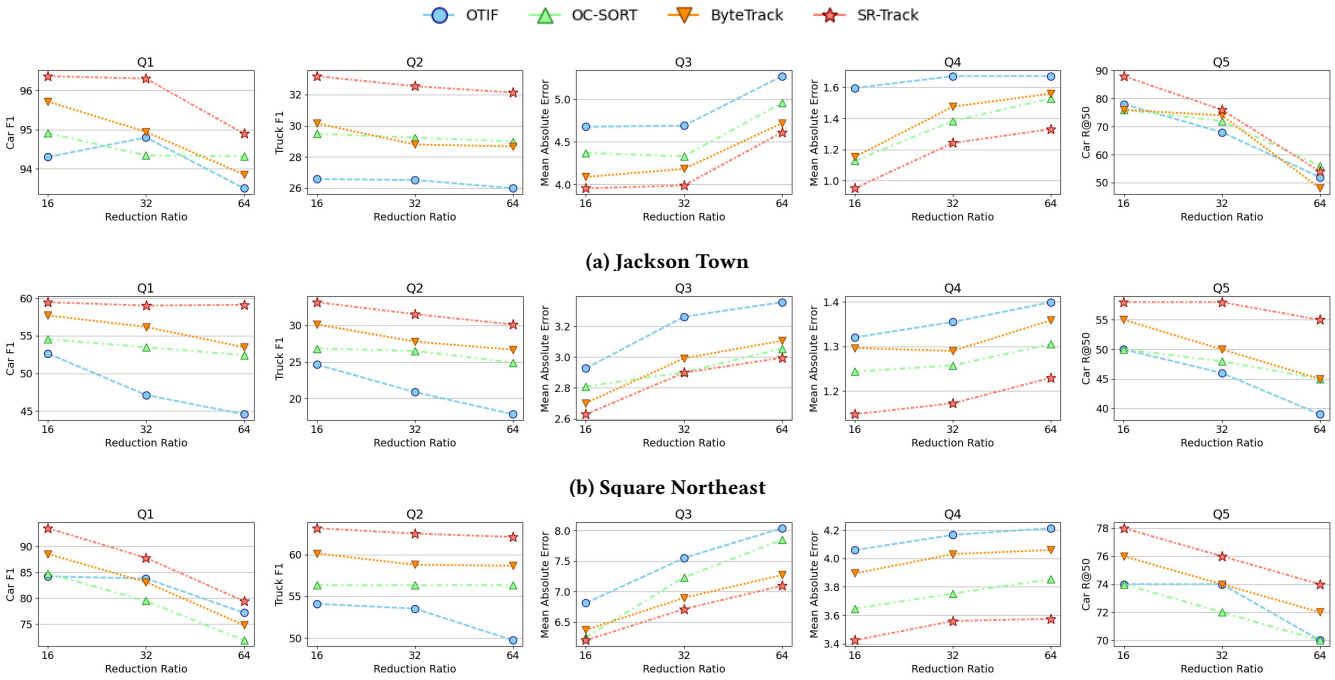
Compared with ByteTrack, our SR-Track can further reduce the processing time by half. For example, it takes SR-Track 45.1s to generate tracking results in MOT17 with $HOTA = 66$, whereas ByteTrack requires 91.7s. In MOT20, the efficiency of SR-Track is also around 2x better than ByteTrack in reaching $HOTA = 55$.

4.4 Performance on Video Query Processing

As listed in Table 3, we examine 5 video queries namely Q_1 to Q_5, with 2 selection queries, 2 aggregation queries and 1 top-k query. We compare OTIF with SR-Track and two trackers (i.e., ByteTrack and OC-SORT) with superior tracking performance in previous experiments. Since OTIF has shown to achieve better efficiency than existing video database systems, as shown in Table 1, we did not incorporate them in this experiment.

Concerning the accuracy of query processing, there appears to be an overall positive correlation with the accuracy of object tracking, since a flawless extraction of the object trajectories can guarantee that the error rate is 0 for video queries Q_1 through Q_5. With the inferior tracking accuracy, we can observe subpar performance of OTIF across all queries. Again, our SR-Track establishes clear superiority in video query processing. With an equivalent reduction ratio, SR-Track demonstrates remarkably higher F_1-scores than its counterparts for selection queries Q_1 and Q_2. We also observe that the F_1-score of "truck" retrieval is not as good as "car" retrieval. This is because "car" is a more general label and trucks can be mistakenly recognized and labelled as "car" by the object detector. For aggregation queries Q_3 and Q_4, SR-Track results in the smallest estimation error. The MAE in Taipei exceeds that of the other two datasets, owing to Taipei's particularly high density of moving objects. For top-k query Q_5, the results become less distinguishable for the metric $R@50$. Nonetheless, SR-Track is still the best performer.

The video processing time in Jackson Town and Square Northeast with $RR \in \{16, 32, 64\}$ is reported in Figure 5. It's interesting to



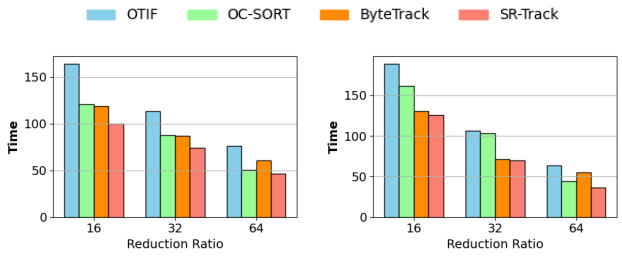
(a) Jackson Town

(b) Square Northeast

(c) Taipei

Figure 4: The video query processing results of Q_1-Q_5 on Jackson Town, Square Northeast and Taipei (best viewed in color).

find that SR-Track is the most lightweight tracker. With the same number of sampled frames to perform object tracking, it incurs the least amount of processing time. In contrast, ByteTrack needs to maintain bounding boxes with low confidence. With these additional bounding boxes, its data association module becomes more expensive. OTIF requires a RNN network to generate track-level features and derive matching scores. Hence, its tracking module incurs the highest inference time.



(a) Jackson Town **(b) Square Northeast**

Figure 5: Video processing time in Jackson Town and Square Northeast.

4.5 Ablation Study

We evaluate the advantage brought by the Sparse-Observation Kalman filter (SOKF) and robust data association (RDA) in Table 6. When SOKF is replaced with traditional KF or RDA or RDA is substituted with the data association module in ByteTrack, we observe performance degradation in all queries. When both components are disabled, the gap is further widened. We also conduct a break-down analysis on the components of SOKF and examine the effect of our proposed Informative State Representation (ISR), Divergence-Aware Mechanism (DAM) and State Forgetting Mechanism (SFM). We can see that they all contribute to the improvement of query accuracy.

Table 6: Ablation study of SR-Track on the Square Northeast dataset with $RR = 64$.

	$Q_1 \uparrow$	$Q_2 \uparrow$	$Q_3 \downarrow$	$Q_4 \downarrow$	$Q_5 \uparrow$
SR-Track	**58.8%**	**30.1%**	**3.0**	**1.2**	**80.0%**
SR-Track w/o SOKF	55.4%	26.9%	3.08	1.37	70.0%
SR-Track w/o RDA	56.4%	28.4%	3.1	1.3	80.0%
SR-Track w/o SOKF+RDA	54.2%	26.7%	3.15	1.37	70.0%
Break-down analysis on SOKF					
SOKF w/o ISR	57.7%	28.3%	3.03	1.2	80.0%
SOKF w/o DAM	57.4%	29.0%	3.05	1.25	70.0%
SOKF w/o SFM	57.7%	29.7%	3.05	1.28	80.0%

5 CONCLUSION

In this paper, we devise a sampling-resilient multi-object tracker called SR-Track to support more efficient video query processing. We propose sparse-observation Kalman filter (SOKF) for more accurate motion estimation and robust data association (RDA) that systematically integrates multiple spatial matching signals. We compare SR-Track with OTIF and 8 alternative MOT strategies on 5 benchmark datasets. The results show that SR-Track provides at least 10× lower video inference time than OTIF in order to achieve sufficiently high tracking accuracy.

REFERENCES

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. 2022. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *CoRR* abs/2206.14651 (2022). https://doi.org/10.48550/arXiv.2206.14651 arXiv:2206.14651

[2] Michael R. Anderson, Michael J. Cafarella, Germán Ros, and Thomas F. Wenisch. 2019. Physical Representation-Based Predicate Optimization for a Visual Analytics Database. In *ICDE*. IEEE, 1466–1477.

[3] Jack Baker, Paul Fearnhead, Emily B. Fox, and Christopher Nemeth. 2019. Control variates for stochastic gradient MCMC. *Stat. Comput.* 29, 3 (2019), 599–615.

[4] Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael J. Cafarella, Tim Kraska, and Sam Madden. 2020. MIRIS: Fast Object Track Queries in Video. In *SIGMOD*. ACM, 1907–1921.

[5] Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael J. Cafarella, Tim Kraska, and Sam Madden. 2020. MIRIS: Fast Object Track Queries in Video. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 1907–1921. https://doi.org/10.1145/3318464.3389692

[6] Favyen Bastani and Samuel Madden. 2022. OTIF: Efficient Tracker Pre-processing over Large Video Datasets. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 2091–2104. https://doi.org/10.1145/3514221.3517835

[7] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. IEEE, 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003

[8] Jiashen Cao, Karan Sarkar, Ramyad Hadidi, Joy Arulraj, and Hyesoon Kim. 2022. FiGO: Fine-Grained Query Optimization in Video Analytics. In *SIGMOD*, Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 559–572.

[9] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. 2022. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. *CoRR* abs/2203.14360 (2022). https://doi.org/10.48550/arXiv.2203.14360 arXiv:2203.14360

[10] Pramod Chunduri, Jaeho Bang, Yao Lu, and Joy Arulraj. 2022. Zeus: Efficiently Localizing Actions in Videos using Reinforcement Learning. In *SIGMOD*, Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 545–558.

[11] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. 2020. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv: Computer Vision and Pattern Recognition* (2020).

[12] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. 2022. StrongSORT: Make DeepSORT Great Again. *CoRR* abs/2202.13514 (2022). arXiv:2202.13514 https://arxiv.org/abs/2202.13514

[13] Yunhao Du, Junfeng Wan, Yanyun Zhao, Binyu Zhang, Zhihang Tong, and Junhao Dong. 2021. 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*. IEEE, 2809–2819. https://doi.org/10.1109/ICCVW54120.2021.00315

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv: Computer Vision and Pattern Recognition* (2021).

[15] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodík, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B. Gibbons, and Onur Mutlu. 2018. Focus: Querying Large Video Datasets with Low Latency and Low Cost. In *OSDI*. 269–286.

[16] Bo Hu, Peizhen Guo, and Wenjun Hu. 2022. Video-zilla: An Indexing Layer for Large-Scale Video Analytics. In *SIGMOD*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1905–1919.

[17] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodík, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018*, Sergey Gorinsky and János Tapolcai (Eds.). ACM, 253–266. https://doi.org/10.1145/3230543.3230574

[18] Simon J. Julier and Jeffrey K. Uhlmann. 1997. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, Ivan Kadar (Ed.), Vol. 3068. International Society for Optics and Photonics, SPIE, 182 – 193. https://doi.org/10.1117/12.280797

[19] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proc. VLDB Endow.* 13, 4 (2019), 533–546.

[20] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Deep CNN-Based Queries over Video Streams at Scale. *Proc. VLDB Endow.* 10, 11 (2017), 1586–1597.

[21] Daniel Kang, John Guibas, Peter D. Bailis, Tatsunori Hashimoto, Yi Sun, and Matei Zaharia. 2021. Accelerating Approximate Aggregation Queries with Expensive Predicates. *Proc. VLDB Endow.* 14, 11 (2021), 2341–2354.

[22] Daniel Kang, John Guibas, Peter D. Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2022. TASTI: Semantic Indexes for Machine Learning-based Queries over Unstructured Data. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1934–1947. https://doi.org/10.1145/3514221.3517897

[23] Daniel Kang, Ankit Mathur, Teja Veeramacheneni, Peter Bailis, and Matei Zaharia. 2020. Jointly Optimizing Preprocessing and Inference for DNN-based Visual Analytics. *Proc. VLDB Endow.* 14, 2 (2020), 87–100.

[24] Ziliang Lai, Chenxia Han, Chris Liu, Pengfei Zhang, Eric Lo, and Ben Kao. 2021. Top-K Deep Video Analytics: A Probabilistic Approach. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM, 1037–1050.

[25] Guoliang Li, Xuanhe Zhou, Shifu Li, and Bo Gao. 2019. QTune: A Query-Aware Database Tuning System with Deep Reinforcement Learning. *Proc. VLDB Endow.* 12, 12 (2019), 2118–2130.

[26] Jiaxin Li, Yan Ding, Hua-Liang Wei, Yutong Zhang, and Wenxiang Lin. 2022. SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking. *Sensors* 22, 15 (2022), 5863. https://doi.org/10.3390/s22155863

[27] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. 2020. Rethinking the competition between detection and ReID in Multi-Object Tracking. *CoRR* abs/2010.12138 (2020). arXiv:2010.12138 https://arxiv.org/abs/2010.12138

[28] Huizi Mao, Taeyoung Kong, and Bill Dally. 2019. CaTDet: Cascaded Tracked Detector for Efficient Object Detection from Video. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*, Ameet Talwalkar, Virginia Smith, and Matei Zaharia (Eds.). mlsys.org. https://proceedings.mlsys.org/book/256.pdf

[29] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. 2022. 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*. IEEE, 8834–8844. https://doi.org/10.1109/CVPR52688.2022.00864

[30] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A Benchmark for Multi-Object Tracking. *arXiv: Computer Vision and Pattern Recognition* (2016).

[31] Oscar Moll, Favyen Bastani, Sam Madden, Mike Stonebraker, Vijay Gadepally, and Tim Kraska. 2020. ExSample: Efficient Searches on Video Repositories through Adaptive Sampling. *CoRR* abs/2005.09141 (2020).

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

[33] Francisco Romero, Johann Hauswald, Aditi Partap, Daniel Kang, Matei Zaharia, and Christos Kozyrakis. 2022. Optimizing Video Analytics with Declarative Model Relationships. *Proc. VLDB Endow.* 16, 3 (2022), 447–460.

[34] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. 2020. TransTrack: Multiple-Object Tracking with Transformer. *CoRR* abs/2012.15460 (2020). arXiv:2012.15460 https://arxiv.org/abs/2012.15460

[35] E.A. Wan and R. Van Der Merwe. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 153–158. https://doi.org/10.1109/ASSPCC.2000.882463

[36] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. 2020. Towards Real-Time Multi-Object Tracking. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI (Lecture Notes in Computer Science)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.), Vol. 12356. Springer, 107–122. https://doi.org/10.1007/978-3-030-58621-8_7

[37] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. 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*. IEEE, 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962

[38] Qijun Xia, Ming Rao, Yiqun Ying, and Xuemin Shen. 1994. Adaptive fading Kalman filter with an application. *Automatica* 30, 8 (1994), 1333–1338.

[39] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. 2021. TransCenter: Transformers with Dense Queries for Multiple-Object Tracking. *CoRR* abs/2103.15145 (2021). arXiv:2103.15145 https://arxiv.org/abs/2103.15145

[40] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. 2021. RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation. *CoRR* abs/2105.04322 (2021). arXiv:2105.04322 https://arxiv.org/abs/2105.04322

[41] Fangao Zeng, Bin Dong, Tiancai Wang, Cheng Chen, Xiangyu Zhang, and Yichen Wei. 2021. MOTR: End-to-End Multiple-Object Tracking with TRansformer. *CoRR* abs/2105.03247 (2021). arXiv:2105.03247 https://arxiv.org/abs/2105.03247

[42] Ji Zhang, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, Minwei Ran, and Zekang Li. 2019. An End-to-End Automatic Cloud Database Tuning System Using Deep Reinforcement Learning. In *SIGMOD*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 415–432.

[43] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2021. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *CoRR* abs/2110.06864 (2021). arXiv:2110.06864 https://arxiv.org/abs/2110.06864

[44] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. 2021. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vis.* 129, 11 (2021), 3069–3087. https://doi.org/10.1007/s11263-021-01513-4

[45] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. 2020. 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*. AAAI Press, 12993–13000. https://ojs.aaai.org/index.php/AAAI/article/view/6999

[46] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2020. Tracking Objects as Points. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV (Lecture Notes in Computer Science)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.), Vol. 12349. Springer, 474–490. https://doi.org/10.1007/978-3-030-58548-8_28

[47] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as Points. In *arXiv preprint arXiv:1904.07850*.