ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box

Hyeonchul Jung, Seokjun Kang, Takgen Kim, HyeongKi Kim
HL Klemove, Republic of Korea
hyeonchul.jung@hlcompany.com

Abstract

Kalman filter-based tracking-by-detection (KFTBD) trackers are effective methods for solving multi-person tracking tasks. However, in crowd circumstances, noisy detection results (bounding boxes with low-confidence scores) can cause ID switch and tracking failure of trackers since these trackers utilize the detector's output directly. In this paper, to solve the problem, we suggest a novel tracker called ConfTrack based on a KFTBD tracker. Compared with conventional KFTBD trackers, ConfTrack consists of novel algorithms, including low-confidence object penalization and cascading algorithms for effectively dealing with noisy detector outputs. ConfTrack is tested on diverse domains of datasets such as the MOT17, MOT20, DanceTrack, and HiEve datasets. ConfTrack has proved its robustness in crowd circumstances by achieving the highest score at HOTA and IDF1 metrics in the MOT20 dataset.

1. Introduction

Multi-person tracking (MPT) aims to detect and track multiple persons by allocating identification values and estimating the position of each person. Therefore, MPT has been utilized in various applications, such as autonomous driving or surveillance systems. Among diverse MPT methods, the tracking-by-detection (TBD) method is a predominant method in recent years since the TBD method makes it easy to make tracking systems by only designing a tracker with a well-made detector [34].

Among TBD methods, for a real-time operating tracker system, the Kalman filter-based TBD (KFTBD) method has been introduced [19]. The process of KFTBD-based trackers consists of matching a predicted track box with a detection box, updating the matched track box using the matched detection box, and keeping the unmatched track box without updating [1, 4, 7, 12, 24, 39, 44].

Over decades, KFTBD-based tracking systems' performance has improved since deep-learning backbone models, utilized as detectors, have shown tremendous performance improvement [7, 44]. However, in crowded places, the detection outputs contain lots of noise when detected objects are exposed to an occlusion, intersection, or motion blur. The noise affects the generation of a detection box with a low-confidence score.

Low-confidence detection boxes can cause the Kalman filter to make incorrect predictions, destabilize the state of the matched track, and eventually lead to track failure. We checked the correlation between the intersection over union (IoU) and the confidence score, applying YOLOX [16] to three benchmark validation sets [10, 26, 32]. As shown in Fig. 1, the matched detection boxes show lower IoU values in the case of detection boxes with lower confidence scores. In other words, detection boxes of lower confidence scores affect the tracker's tracking performance since the boxes obtain low IoU values.

To avoid the issues of low-confidence detection boxes, lots of KFTBD-based trackers set a threshold and use only

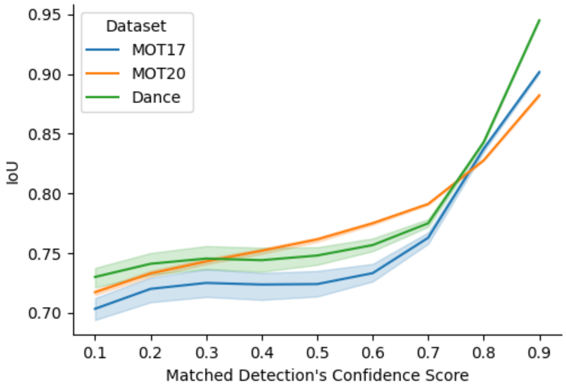


Figure 1. Correlation graph between IoU and confidence score of matched detection boxes. The bold line is the average value of IoU that occurs when the YOLOX detection result is matched with GT. The band represents the distribution of that IoU.

detection boxes whose confidence is higher than the threshold [4, 12, 39]. Therefore, since the strategy discards the remaining lower confidence detection boxes than the threshold for generating tracks, several studies tried to utilize low-confidence detection boxes in the ID matching process [1, 7, 24, 40, 44]. However, low-confidence detection boxes are still not considered in the overall tracking process including track initialization.

In this paper, we suggest a novel KFTBD-based tracker named ConfTrack. According to Fig. 1, we confirmed that the predicted track box from the Kalman filter is closer to the ground truth than the low-confidence detection box from the detector. Inspired by this, we assumed the predicted track box is more reliable than the noisy detection box. Therefore, to mitigate the noise effect, the proposed method is designed to penalize detection boxes with low-confidence in the Kalman prediction, update, and matching stages. Furthermore, compared with the existing KFTBD-based methods that discard low-confidence detection boxes in track initialization, ConfTrack initializes tentative tracks from low-confidence detection boxes and treats them partitively with confirmed tracks in a novel cascade matching strategy.

To demonstrate the reliability of the proposed method, diverse domains of datasets are utilized for verification, including the MOT17, MOT20, DanceTrack, and HiEve dataset [10, 22, 26, 32]. For objective evaluation, we adopt the test metrics including Higher Order Tracking Accuracy (HOTA), Multiple Object Tracking Accuracy (MOTA), and Identification F1 score (IDF1) [3, 23, 29]. ConfTrack has achieved the highest HOTA and IDF1 on the MOT20 dataset.

The main contributions of our work can be summarized as follows.

- A novel KFTBD-based tracker is suggested, called ConfTrack. Several novel penalization methods are introduced for low-confidence objects for robust tracking operations in crowd circumstances containing noisy detection results.

- A novel cascading method of track matching is suggested by initializing tracks, including low-confidence detection results, and handling tentative tracks and confirmed tracks differently. This method prevents the ID switch of confirmed tracks in various tracking scenarios, such as long-term occlusion, truncation, and motion blur.

- ConfTrack is evaluated with various datasets. Among them, the best scores have been recorded at the HOTA and IDF1 metrics on the MOT20 dataset.

2. Related Work

Approaches for multi-person tracking MPT algorithms with neural network backbones can be divided into three types according to their structure: TBD, joint detection and tracking (JDT), and transformer-based.

TBD-based trackers generate object tracks by using the outputs of attached pre-trained deep-learning-based detector backbones. Even though TBD trackers do not consider utilizing detectors' extracted feature information during the tracking process, they are still extensively adopted as a tracking system, such as an embedded system for real-time operation.

JDT-based trackers are end-to-end neural network-based methods that conduct feature extraction and track association at once [2, 15, 38, 45, 46].

Transformer-based trackers can utilize global contextual features that were difficult to obtain in JDT-based trackers by adopting a transformer-based encoder-decoder structure [6, 9, 25, 33, 42, 47].

Kalman filter based tracking-by-detection. KFTBD-based trackers [4, 12, 39] have shown powerful tracking performance by applying classical methods such as the Kalman filter [19] and the Hungarian algorithm [20]. Due to the emergence of superior deep-learning-based detectors [16], KFTBD-based trackers have been the most considered as an MPT system.

To expand the availability of detectors' deep feature, KFTBD-based trackers with deep feature extractors for recovery of long-time occluded tracks have been introduced [1, 24, 39]. In addition, KFTBD-based trackers that calculate the motion of the camera to compensate for the prediction errors of the Kalman filter have been researched [1, 14, 30].

Handling detection noise. Several KFTBD-based trackers with noise-handling functions have been designed to reduce the ratio of measurement values in the Kalman update process. In the case of GIAOTracker [11], It increases the reflection ratio for detections with high confidence by multiplying confidence by the measurement space noise covariance used to calculate the Kalman gain. However, since the confidence score ranges from 0 to 1, it does not reduce the rate of low-confidence detections. In MAATrack [30] research, since the most recently matched detection box of the lost track contains lots of noise, the variation of bounding box height is set to 0 when the lost track predicts the next state.

3. Proposed Method

As shown in Fig. 2, ConfTrack is based on BoT-SORT [1] as a baseline model. Therefore, most of the operation process is similar to the KFTBD-based BoT-SORT. However, for effective utilization of low-confidence de-

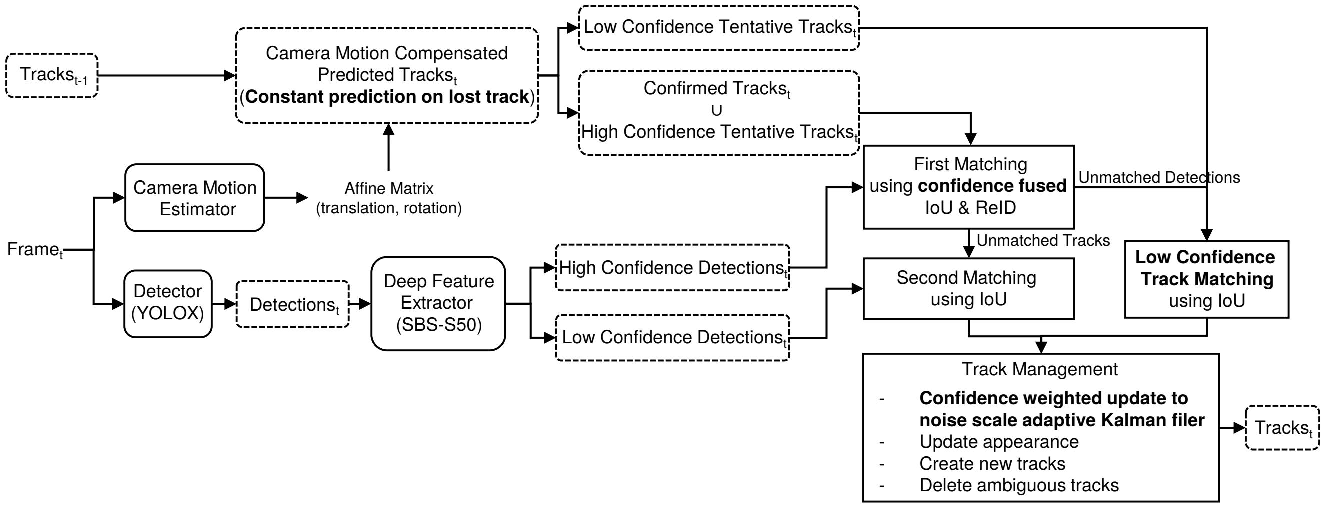

Tracks$_{t-1}$
Camera Motion Compensated Predicted Tracks$_t$
(Constant prediction on lost track)
Low Confidence Tentative Tracks$_t$
Confirmed Tracks$_t$
U
High Confidence Tentative Tracks$_t$
Camera Motion Estimator
Affine Matrix (translation, rotation)
Frame$_t$
Detector (YOLOX)
Detections$_t$
Deep Feature Extractor (SBS-S50)
High Confidence Detections$_t$
Low Confidence Detections$_t$
First Matching using confidence fused IoU & ReID
Unmatched Detections
Unmatched Tracks
Second Matching using IoU
Low Confidence Track Matching using IoU
Track Management
- Confidence weighted update to noise scale adaptive Kalman filer
- Update appearance
- Create new tracks
- Delete ambiguous tracks
Tracks$_t$

Figure 2. Framework of ConfTrack. Proposed methods are marked in bold.

tection outputs, ConfTrack is designed with several novel methods as follows.

3.1. Confidence Weighted Kalman-Update (CW)

In update stage of Kalman filter, an estimation of kth frames $\hat{x}_k$ is updated using prediction $\hat{x}_k^-$, target measurement $\tilde{z}_k$, projection matrix H and Kalman gain K_k as in Eq. (1).

$$\hat{x}_k = \hat{x}_k^- + K_k(\tilde{z}_k - H\hat{x}_k^-) \tag{1}$$

For a detection box whose confidence score c_z is lower than threshold c_{thr}, we set the target measurement by replacing the original detection box z_k with a box closer to the prediction box, weighted by the confidence of the detection box as in Eq. (2).

$$\tilde{z}_k = \begin{cases} z_k, & c_z \geq c_{thr} \\ z_k + (H\hat{x}_k^- - z_k) * (1 - c_z), & \text{otherwise} \end{cases} \tag{2}$$

It directly modifies a target measurement of the Kalman filter. As the confidence score of a detection box is higher, the target remains as the original detection box. In the opposite case, the target resembles the prediction box.

3.2. Noise Scale Adaptive Kalman Filter (NK)

To penalize a low-confidence detection box in a matching stage, the noise scale adaptive Kalman filter (NSAK) [11] is utilized in ConfTrack additionally. However, since the original NSAK multiplies raw confidence c_z to a measurement space noise covariance R between 0 and 1, it cannot amplify the noise. Therefore, we modify the original NSAK to multiply an amplifying factor α to make the range of the value multiplied to R greater than one as in Eq. (3).

$$\tilde{R}_k = R * (1 - c_z) * \alpha \tag{3}$$

The proposed amplified measurement space noise covariance $\tilde{R}_k$ is more suitable for making the Kalman gain value smaller than prior work [11] when a confidence score of a detection box is low. The amplifying factor is a sensitive number. If it is too large, the tracker can over-rely on the predictions of the Kalman filter, making it unable to track the nonlinear motion of a person. Therefore, currently, we maintain this value as a hyperparameter that can vary depending on the dataset.

3.3. Constant Box Prediction on Lost Track (CP)

To prevent the effect of unstable box size variation, we adjust the box size variation of the lost track to zero inspired [30]. When a confirmed track becomes a lost track, the matched detection box used for a lost track contains noise (low-confidence). Since the Kalman filter predicted the wrong box at the noisy detection box, we assumed that the mispredicted box's width and height contributed more to tracking failure than the mispredicted box's position.

The state of the Kalman filter we use is defined as $[x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]$, where x, y are the center point of the bounding box and w, h are width and height of that box. Therefore, we set $\dot{w}, \dot{h}$ of a lost track to zero before the Kalman prediction stage. It allows the Kalman filter to keep the width and height of the predicted box.

3.4. Confidence Fused Cost Matrix (CF)

For robust tracking operation in the ID matching process, the confidence-fused cost matrix (CF) is applied inspired

BYTETrack [44]. The sequences for obtaining the confidence fused cost matrix C_{fused} from original cost matrix C_{org} are described as from Eq. (4) to Eq. (6).

$$S_{org} = 1 - C_{org} \tag{4}$$

$$S_{fused} = S_{org} * C_z \tag{5}$$

$$C_{fused} = 1 - S_{fused} \tag{6}$$

First, calculate the similarity matrix S_{org} as in Eq. (4). Second, multiply a confidence matrix C_z composed of the confidence scores of the detection boxes as in Eq. (5). Finally, turn the confidence fused similarity matrix S_{fused} back into confidence fused cost matrix C_{fused} as in Eq. (6). Unlike previous studies [1, 7, 24, 44] that applied the above fusing method only to the IoU-based cost matrix, we also apply it to the ReID feature-based cost matrix.

3.5. Cascade Matching with Low-Confidence Track (LM)

In a location where an FP box is detected in the current frame, the FP box is able to be detected again in the vicinity of that location in the next frame. Therefore, there is a high possibility that the track initialized from the FP box of the current frame will be immediately matched with the FP box of the next frame and will become a wrong confirmed track. Due to the FP issue, previous studies have not adopted the track initialization from the low-confidence detection box [44]. However, we solve this problem with a novel cascade matching (LM), which separates matching candidates so that the low-confidence track does not match the low-confidence detection. The proposed matching method is described in Algorithm 1.

ConfTrack selects only tracks and detections with high confidence as candidates in the first matching. Therefore, ConfTrack forms the track candidates by joining confirmed tracks and high-confidence tentative tracks and comprising the detection candidates only using high-confidence detections preferentially. In the second matching step, ConfTrack matches tracks that were not matched in the first stage and low-confidence detections like the BYTETrack matching method [44]

However, in the case of LM, low-confidence tentative tracks and high-confidence detections that were not matched in the first stage are selected as matching candidates. Among the low-confidence tentative tracks, we confirmed that there are overlapping objects with unmatched tracks in the second matching step. Therefore, ConfTrack regards these tracks as duplicates and deletes them through non-maximum suppression (NMS) by IoU threshold $\triangle_d$. After that, ConfTrack matches the remaining tracks with unmatched detection candidates from the first stage.

```
Algorithm 1: Proposed cascade matching
Input: confirmed tracks $\mathcal{T}_{confirm}$,
       high confidence tentative tracks $\mathcal{T}_{high-tent}$,
       low-confidence tentative tracks $\mathcal{T}_{low-tent}$,
       high confidence detections $\mathcal{D}_{high}$,
       low-confidence detections $\mathcal{D}_{low}$,
       first matching threshold $\triangle_f$,
       second matching threshold $\triangle_s$
       duplicate track threshold $\triangle_d$,
       low-confidence matching threshold $\triangle_l$,
       track max age $\sigma$
Output: updated set of tracks $\mathcal{T}_u$
$\mathcal{T}_{confirm} \leftarrow \mathcal{T}_{confirm} \cup \mathcal{T}_{high-tent}$
$\mathcal{T}_u \leftarrow \mathcal{T}_{confirm} \cup \mathcal{T}_{low-tent}$
/* First matching */
$C_{1st} = Cost_{1st}(\mathcal{T}_{confirm}, \mathcal{D}_{high})$
Associate $\mathcal{T}_{confirm}$ and $\mathcal{D}_{high}$ using $C_{1st}$, $\triangle_f$
$\mathcal{T}_{conf-remain} \leftarrow$ remaining tracks from $\mathcal{T}_{confirm}$
$\mathcal{D}_{high-remain} \leftarrow$ remaining detections from $\mathcal{D}_{high}$
/* Second matching (BYTE) */
$C_{2nd} = Cost_{2nd}(\mathcal{T}_{conf-remain}, \mathcal{D}_{low})$
Associate $\mathcal{T}_{conf-remain}$ and $\mathcal{D}_{low}$ using $C_{2nd}$, $\triangle_s$
$\mathcal{T}_{conf-remain} \leftarrow$ remaining tracks from
  $\mathcal{T}_{conf-remain}$
$\mathcal{D}_{low-remain} \leftarrow$ remaining detections from $\mathcal{D}_{low}$
/* Low-confidence track matching */
$C_{dup} = Cost_{dup}(\mathcal{T}_{low-tent}, \mathcal{T}_{conf-remain})$
check duplicate tracks using $C_{dup}$, $\triangle_d$
$\mathcal{T}_{tent-valid} \leftarrow$ not duplicate tracks from $\mathcal{T}_{tentative}$
$C_{low} = Cost_{low}(\mathcal{T}_{tent-valid}, \mathcal{D}_{high-remain})$
Associate $\mathcal{T}_{tent-valid}$ and $\mathcal{D}_{high-remain}$ using
  $C_{low}$, $\triangle_l$
$\mathcal{T}_{tent-remain} \leftarrow$ remaining tracks from $\mathcal{T}_{tent-valid}$
$\mathcal{D}_{high-remain} \leftarrow$ remaining detections from
  $\mathcal{D}_{high-remain}$
/* Unmatched track management */
$\mathcal{T}_u \leftarrow \mathcal{T}_u \setminus \mathcal{T}_{tent-remain}$
for t in $\mathcal{T}_{conf-remain}$ do
    if t.age $\geq \sigma$ then
        $\mathcal{T}_u \leftarrow \mathcal{T}_p \setminus t$
/* initialize new tracks */
$\mathcal{D}_{remain} \leftarrow \mathcal{D}_{high-remain} \cup \mathcal{D}_{low-remain}$
for d in $\mathcal{D}_{remain}$ do
    $t \leftarrow track(d)$
    $\mathcal{T}_u \leftarrow \mathcal{T}_u \cup t$
return $T_u$
```

4. Experiments

4.1. Datasets

The MOT17 is the most popular dataset for evaluating MPT algorithms with a stationary or moving camera [26]. While the MOT17 dataset had been collected in general and various environments, the MOT20 dataset focuses on more complicated data for crowded environments [10]. MOT17 and MOT20 datasets are provided as 'train' and 'test', so we construct a validation set by dividing the 'train' data in half as [44] did. The DanceTrack dataset consists of videos of a few people dancing in a static space and is characterized by a lot of non-linear motion [32]. We also use the HiEve dataset to consider more large-scale datasets than MOT datasets and variations in resolution, the angle at which people were photographed, and the actions people took [13].

4.2. Metrics

Three major metrics are utilized to evaluate the proposed method: HOTA, CLEAR, and Identity [3, 23, 29]. HOTA consists of detection accuracy (DetA), association accuracy (AssA), and localization accuracy (LocA), and we use this as a primary metric for its merit for considering both the detection level and the trajectory level. CLEAR is the most commonly used metric and consists of multi-object tracking accuracy (MOTA), multi-object prediction (MOTP), etc. CLEAR focuses more on the detection level than the trajectory level. Identity includes the IDF1 score and can evaluate how well the tracker tracks without an ID switch. We obtain the above metrics using the framework of TrackEval [18].

4.3. Implementation Details

Detector. ConfTrack adopts YOLOX [16] as a detector with reference to previous studies [1, 7, 24, 40, 44]. In addition, for experiments about ConfTrack's compatibility for various detectors, YOLOv7 [35] and Transformer-based DINO [43] are used along with YOLOx, and all detectors use weights learned on COCO dataset [21]. Furthermore, the weights made by Zhang et al. [44] are used for the MOT17, MOT20, and HiEve datasets, using the weights made by Jinkun et al. [7] for the DanceTrack. Detection boxes from every detector are filtered by NMS with a confidence threshold of 0.1 and an IoU threshold of 0.7.

Feature extractor. We use the SBS-S50 model from FastReID as our feature extractor [17]. The weights made by Nir et al. [1] are used for all used datasets. ConfTrack extracts feature only for the high-confidence detection boxes, shortening the time compared with applying them to the entire detection boxes.

Camera motion compensation. For camera motion compensation (CMC), we adopt the OpenCV implementation of the Video Stabilization module as previous studies did [1, 5, 24]. For convenience, we utilize files created in advance by [24] for MOT17, MOT20, and DanceTrack datasets.

Hyper parameters. For track management, track initialization threshold c_i is 0.1, high confidence tentative track threshold c_c is 0.7, and high confidence detection threshold c_d is 0.6. For a tentative track to be confirmed, it has to match 3 frames in a row. A confirmed track is deleted if it cannot match for 30 frames. For the threshold used in CW, we set c_{thr} to 0.7 for DanceTrack and 0.6 for other datasets. For NK, amplifying factor α is set to 10.0 for DanceTrack in Tab. 1 and MOT datasets where CMC is not used in Tab. 2. For other cases, α is set to 100.0. CF is applied only in the first matching stage. The threshold used in the first matching $\triangle_f$ is 0.8, and 0.6 for IoU distance and 0.25 for cosine distance are applied. In the second matching, the matching threshold $\triangle_s$ is 0.5. For LM, the threshold $\triangle_d$ used in duplicate track check is set to 0.7, and the matching threshold $\triangle_l$ is 0.3. We evaluated the generalization performance of the proposed methods by using the above values without adjusting according to the dataset. We implement all the experiments using PyTorch and use a desktop with Intel Core i9-10900K @ 3.70GHz and NVIDIA GeForce RTX 3090.

Post processing. We only apply a linear interpolation proposed by [44] to the MOT17 and MOT20 test sets for fair competition with the existing studies on the benchmark as [1, 24, 44] did.

4.4. Experimental Results

Ablation study. We perform an ablation study on the MOT17-val, MOT20-val, and DanceTrack-val datasets to analyze the performance contribution of each novel component of ConfTrack and combinations of that as in Tab. 1.

As shown in Tab. 1, CW and NK show a consistent contribution. In particular, NK raises HOTA and IDF1 the most among all methods. It proves that reducing the noise of the detection box in the updating process of the Kalman filter makes a KFTBD tracker more stable and less interrupted. CP alone increases all metrics in MOT17 but slightly decreases MOTA in DanceTrack and HOTA and IDF1 in MOT20. The reason is that CP is a method of maintaining the Kalman filter prediction of the lost track, so in MOT20 and DanceTrack, which have objects containing more noise than MOT17, the prediction to be maintained is easily affected by noise. However, if CP applies with CW and NK that penalizes the low confidence detection box, CP improves almost all metrics.

CF makes matching more difficult, reducing the ID switching, which generally lowers MOTA slightly and raises HOTA and IDF1. The effect was prominent in the DanceTrack dataset with a small number of persons, whereas it was negligible in MOT20 collected in a dense environment. LM increases matching candidates, resulting

Table 1. Ablation study on MOT17-val, MOT20-val, and DanceTrack-val datasets.

					MOT17-val			MOT20-val			DanceTrack-val		
CW	NK	CP	CF	LM	HOTA	IDF1	MOTA	HOTA	IDF1	MOTA	HOTA	IDF1	MOTA
					68.99	81.26	77.79	64.07	81.85	82.27	56.04	56.48	89.65
✓					70.00	82.81	**78.25**	64.61	82.73	82.60	56.51	57.03	89.76
	✓				**70.03**	**83.37**	78.08	**65.40**	**83.55**	82.73	**57.30**	**57.61**	89.75
		✓			69.43	81.83	77.89	63.97	81.58	82.42	56.54	56.73	89.58
			✓		69.08	81.70	77.68	64.07	81.87	82.24	56.67	57.21	**89.81**
				✓	69.15	81.67	77.86	64.42	81.94	**83.32**	56.10	56.27	89.66
✓	✓				70.23	84.34	78.54	64.76	82.74	82.10	56.52	56.25	89.60
✓	✓	✓			70.39	84.63	78.45	64.83	82.92	82.18	57.53	57.63	89.59
✓	✓	✓	✓		70.57	84.65	78.41	64.79	82.87	82.17	**57.58**	**57.72**	**89.77**
✓	✓	✓	✓	✓	**70.68**	**84.70**	**78.95**	**65.24**	**83.20**	**83.15**	57.17	57.05	89.70

Table 2. Experiment on the dependence between the proposed methods about modified Kalman filter and the CMC and ReID modules.

Method				MOT17-val			MOT20-val		
	CW	NK	CP	HOTA ↑	IDF1 ↑	MOTA ↑	HOTA ↑	IDF1 ↑	MOTA ↑
ByteTrack				66.05	76.81	76.52	62.88	79.95	82.19
	✓			**66.95**	**78.22**	**76.83**	**64.05**	**81.86**	**82.54**
		✓		66.59	77.88	76.59	63.79	81.25	82.47
			✓	66.21	76.63	76.47	62.87	79.94	82.28
ByteTrack + CMC				68.05	79.79	77.67	63.05	80.22	82.17
	✓			68.70	81.19	78.02	64.11	82.32	82.56
		✓		**68.96**	**81.65**	**78.28**	**65.00**	**83.04**	**82.67**
			✓	68.02	79.88	77.65	63.08	80.35	82.29
ByteTrack + ReID				67.59	79.12	76.88	63.86	81.54	82.34
	✓			**68.56**	**80.35**	**77.31**	64.52	82.57	82.51
		✓		68.02	80.01	77.17	**64.61**	**82.61**	**82.53**
			✓	67.55	78.92	76.99	63.89	81.56	82.42
ByteTrack + CMC + ReID				68.99	81.26	77.79	64.07	81.85	82.27
	✓			70.00	82.81	**78.25**	64.61	82.73	82.56
		✓		**70.03**	**83.37**	78.08	**65.40**	**83.55**	**82.73**
			✓	69.08	81.70	77.68	63.97	81.58	82.42

in more matches. Therefore, contrary to CF, it improves the performance of MOT20 but causes ID switching in DanceTrack to decrease IDF1.

Tab. 1 shows that ConfTrack can track people well in environments where noisy detection can occur often and in various general situations.

Modified Kalman filter. The three novel methods (CW, NK, CP) for the Kalman filter allow the tracker to focus more on the predicted box rather than the low confidence detection box compared to the original Kalman filter. Therefore, we verify whether the proposed methods depend on the CMC and ReID modules used by BoTSORT, our baseline, to complement the linearity of the Kalman filter. As shown in Tab. 2, we test the dependence based on ByteTrack by excluding CMC and ReID from the baseline.

We confirm that CW and NK are methods that increase the performance of the tracker the most, regardless of CMC and ReID modules. When NK is combined with CMC, the performance increase is remarkable compared to other methods. It is difficult to expect consistent performance improvement regardless of CMC and ReID modules when CP is applied alone without the other two techniques.

Compatibilty to other detectors. The results of testing whether the proposed ConfTrack is compatible with detections extracted from other detectors are shown in Tab. 3. To use detection results with lots of noise, the weights of all detectors used only the COCO dataset without using the MOT17 train set. In Tab. 3, all metrics' values are increased with a large margin when ConfTrack is applied rather than the baseline for all detectors.

Table 3. Comparison between ConfTrack and baseline with 3 different detectors.

Method	MOT17-val		
	HOTA	IDF1	MOTA
YOLOx+Baseline	42.91	47.44	34.03
YOLOx+ConfTrack	**45.17**	**52.89**	**36.90**
YOLOv7+Baseline	42.76	47.92	34.06
YOLOv7+ConfTrack	**45.85**	**53.50**	**37.08**
Transformer [43]+Baseline	33.58	34.03	22.79
Transformer [43]+ConfTrack	**36.98**	**40.05**	**26.51**

We notice that the performance gain of metrics in the current experiment is much greater than that of Tab. 1. This shows that ConfTrack is more effective as the noise contained in the detection increases.

Comparison with existing KFTBDs. We compare ConfTrack with other KFTBD trackers for the MOT17 and HiEve datasets. For fairness, we all used the same detection and left the parameter settings of the existing KFTBDs as they were implemented. Among them, ConfTrack achieves the highest score for all metrics except FPS. According to Tab. 4, ConfTrack can track persons better than prior works in a general situation where people exist. In the case of FPS, we confirmed that it is competitive among trackers that adopted the ReID function [1, 24, 39].

Additionally, we visually compare the track trajectories to see how less sensitive ConfTrack is to the noise of the detection box compared to other celebrated KFTBDs [7, 24, 44]. A visualized trajectories can be seen in Fig. 3.

The examples of long-term occlusion are in the first and second columns. People are initialized from the high confidence detection box and pass through the densely populated area in the MOT17 validation set. In the case of the existing KFTBDs, when the person passes through a dense area, trackers are affected by a long-term occlusion with a shrunken detection box containing much noise, and the trajectories end with a much smaller box than the initial box. Baseline continues tracking more than other KFTBD-based trackers, but as in the second column, an ID switch occurs, creating an incorrect trajectory. On the other hand, the proposed ConfTrack is less affected by long-term occlusion, so it was possible to create a trajectory close to GT.

The third column is an example of the trajectory of several people in a crowded environment of the MOT20 validation set. All algorithms generate trajectories similar to GT, except that [7] shows an interruption of the person on the right side. However, we can see that the results from ConfTrack are the most stable, with the least amount of visual jittering in the crowd scene.

Analysis of limitations. Since KFTBD-based trackers are designed based on a linear model such as the Kalman filter, there are still limitations on non-linear object tracking. Especially, It is quite challenging to track the person taking a non-linear motion like the Fig. 4. The person on the Fig. 4a moves to the left and then stops. At this time, occlusion occurs, and tracking fails. Also, in the case of Fig. 4b, the matching fails since the ratio and size of the box change

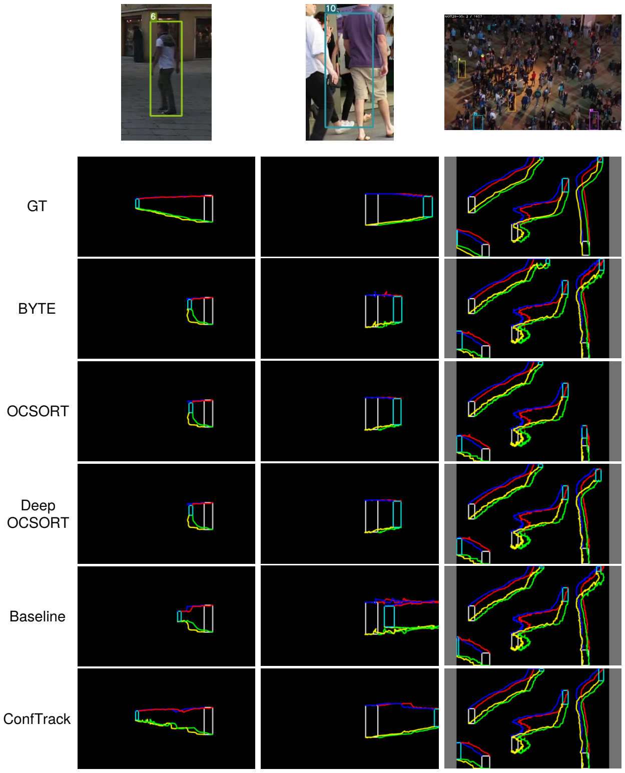
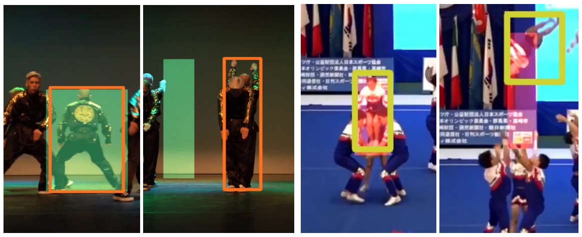


Figure 3. Comparative results between ConfTrack and other KFTBD-based trackers. The top row corresponds to the start track box of the persons selected as an example, and each column represents a trajectory created from the first box. The white box is the first track box, and the sky blue box is the track box at the end of tracking. The red, blue, yellow, and light green lines are the trajectories of the top-right, top-left, bottom-left, and bottom-right of the box, respectively.



(a) DancTrack0034 (b) DancTrack0081

Figure 4. Failure cases of ConfTrack in DanceTrack-val. The box with a bold line is the GT, and the transparent box is the prediction of the tracker.

Table 4. Comparison between ConfTrack and other KFTBD methods using the same detection results.

Method	with	MOT17-val				HiEve-train		
	ReID	HOTA ↑	IDF1 ↑	MOTA ↑	FPS ↑	IDF1 ↑	MOTA ↑	MOTP ↑
SORT [4]		66.27	77.49	74.71	27.70	55.5	57.0	20.2
DeepSORT [39]	✓	66.28	78.00	76.39	9.73	52.8	59.1	20.8
ByteTrack [44]		67.88	79.94	77.84	**28.06**	57.1	62.3	21.3
OC-SORT [7]		66.38	77.78	74.70	26.78	55.1	57.0	20.1
DeepOC-SORT [24]	✓	68.26	81.15	75.20	11.17	57.9	57.1	20.1
BoTSORT [1]	✓	69.19	82.01	78.47	12.43	59.6	62.6	20.9
ConfTrack(proposed)	✓	**70.68**	**84.70**	**78.95**	12.37	**62.4**	**63.9**	**21.5**

Table 5. Comparison with published tracking methods on MOT20 test set of private detections.

Tracker	online	HOTA	IDF1	MOTA
UTM [41]	✓	62.5	76.9	78.2
SelfAT [37]	✓	62.6	76.6	75.0
StrongSORT [12]		62.6	77.0	73.8
RTU-P2 [36]		62.8	76.8	76.5
MotionTrack [27]	✓	62.8	76.5	78.0
BoT-SORT [1]	✓	63.3	77.5	77.8
FineTrack [28]	✓	63.6	79.0	77.9
DeepOC-SORT [24]	✓	63.9	79.2	75.6
SUSHI [8]		64.3	79.8	74.3
ImprAsso [31]	✓	64.6	78.8	**78.6**
ConfTrack(proposed)	✓	**64.8**	**80.2**	77.2

Table 6. Comparison with published tracking methods on MOT17 test set of private detections.

Tracker	online	HOTA	IDF1	MOTA
UTM [41]	✓	64.0	78.7	81.8
CBIOU [40]	✓	64.1	79.7	81.1
FineTrack [28]	✓	64.3	79.5	80.0
StrongSORT [12]		64.4	79.5	79.6
SelfAT [37]	✓	64.4	79.8	80.0
DeepOC-SORT [24]	✓	64.9	80.6	79.4
BoT-SORT [1]	✓	65.0	80.2	80.5
MotionTrack [27]	✓	65.1	80.1	81.1
ImprAsso [31]	✓	66.4	82.1	**82.2**
SUSHI [8]		**66.5**	**83.1**	81.1
ConfTrack(proposed)	✓	65.4	81.2	80.0

while a person is crouching and jumping.

4.5. Results on Benchmark

We participated in MOT17 and MOT20 competitions using private detection with ConfTrack. In Tabs. 5 and 6, the results of comparing the scores with existing published state-of-the-art methods are described. For the MOT20 test set, ConfTrack achieves the highest score in HOTA and IDF1, even including offline algorithms. In particular, by acquiring the highest IDF1, ConfTrack proves that it is robust to the noise of the detection box in a crowded environment and can stably track with less missing track than others. The highest HOTA shows that ConfTrack can generate a stable trajectory that does not easily break when occlusion occurs frequently.

For the MOT17 test set, ConfTrack Although MOTA fell below the baseline [1], ConfTrack ranks third for HOTA and IDF1. If only online algorithms are considered, it is in second place. This shows that ConfTrack performs well in general situations such as alleys and road environments.

5. Conclusion

In this paper, we propose a novel KFTBD-based tracker named ConfTrack. ConfTrack shows state-of-the-art tracking performance by using a detector's low-confidence outputs. Since ConfTrack adopts a novel cascade matching method to utilize the low-confidence outputs, it does not cause tracking failure of confirmed tracks even when tracks are initialized from low-confidence detection boxes.

According to in-depth analysis, the performance robustness of ConfTrack is proven in the diverse domain of datasets containing noise caused by a long-term occlusion or crowd circumstance. Also, according to detector compatibility and ablation studies, ConfTrack proves that it can be considered in various tracking development environments.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 1, 2, 4, 5, 7, 8

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 2

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP J. Image and Video Process.*, 2008:1–10, 2008. 2, 5

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016. 1, 2, 8

[5] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 5

[6] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022. 2

[7] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-Centric SORT: Rethinking sort for robust multi-object tracking. In *CVPR*, pages 9686–9696, 2023. 1, 2, 4, 5, 7, 8

[8] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *CVPR*, pages 22877–22887, 2023. 8

[9] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking. In *WACV*, pages 4870–4880, 2023. 2

[10] 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 preprint arXiv:2003.09003*, 2020. 1, 2, 5

[11] 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 *ICCVW*, pages 2809–2819, 2021. 2, 3

[12] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make deepsort great again. *IEEE Trans. Multimedia*, 2023. 1, 2, 8

[13] Weiyao Lin et al. HiEve: A Large-Scale Benchmark for Human-Centric Video Analysis in Complex Events. *Int. J. Comput. Vis.*, pages 1–25, 07 2023. 5

[14] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1858–1865, 2008. 2

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, pages 3038–3046, 2017. 2

[16] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1, 2, 5

[17] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. FastReID: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 5

[18] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020. 5

[19] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. 1, 2

[20] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5

[22] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Rui Qian, Tao Wang, Ning Xu, Hongkai Xiong, Guo-Jun Qi, and Nicu Sebe. Human in events: A large-scale benchmark for human-centric video analysis in complex events. *arXiv preprint arXiv:2005.04490*, 2020. 2

[23] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129:548–578, 2021. 2, 5

[24] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023. 1, 2, 4, 5, 7, 8

[25] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, pages 8844–8854, 2022. 2

[26] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 1, 2, 5

[27] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. MotionTrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, pages 17939–17948, 2023. 8

[28] Hao Ren, Shoudong Han, Huilin Ding, Ziwen Zhang, Hongwei Wang, and Faquan Wang. Focus On Details: Online multi-object tracking with diverse fine-grained representation. In *CVPR*, pages 11289–11298, 2023. 8

[29] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016. 2, 5

[30] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *WACV*, pages 133–142, 2022. 2, 3

[31] Daniel Stadler and Jürgen Beyerer. An improved association pipeline for multi-person tracking. In *CVPR*, pages 3169–3178, 2023. 8

[32] 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 *CVPR*, pages 20993–21002, 2022. 1, 2, 5

[33] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack:

Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 2

[34] Zhihong Sun, Jun Chen, Liang Chao, Weijian Ruan, and Mithun Mukherjee. A survey of multiple pedestrian tracking based on tracking-by-detection framework. *IEEE Trans. Circuit Syst. Video Technol.*, 31(5):1819–1833, 2020. 1

[35] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *CVPR*, pages 7464–7475, 2023. 5

[36] Shuai Wang, Hao Sheng, Da Yang, Yang Zhang, Yubin Wu, and Sizhe Wang. Extendable multiple nodes recurrent tracking framework with RTU++. *IEEE Trans. Image Process.*, 31:5257–5271, 2022. 8

[37] Shuai Wang, Da Yang, Yubin Wu, Yang Liu, and Hao Sheng. Tracking Game: Self-adaptative agent based multi-object tracking. In *MM*, pages 1964–1972, 2022. 8

[38] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, pages 107–122. Springer, 2020. 2

[39] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 1, 2, 7, 8

[40] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *CVPR*, pages 4799–4808, 2023. 2, 5, 8

[41] Sisi You, Hantao Yao, Bing-Kun Bao, and Changsheng Xu. UTM: A unified multiple object tracking model with identity-aware feature enhancement. In *CVPR*, pages 21876–21886, 2023. 8

[42] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-End Multiple-Object Tracking with TRansformer. In *ECCV*, 2022. 2

[43] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 5, 7

[44] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21. Springer, 2022. 1, 2, 4, 5, 7, 8

[45] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 2

[46] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490. Springer, 2020. 2

[47] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8771–8780, 2022. 2