ADAPTTRACK: AN ENHANCED BYTETRACK APPROACH USING A VELOCITY-ADAPTIVE KALMAN FILTER

Hai Hoang Hong[1,*], Long Nguyen Huu[1], Dung Nguyen Tien[1], Nhat Minh Phung[1], Long Hoang Duc[1]

DOI: https://doi.org/10.57001/huih5804.2026.114

ABSTRACT

Multiple Object Tracking (MOT) remains a challenging task in computer vision, particularly under occlusion and detection uncertainty. While ByteTrack improves tracking performance by leveraging both high- and low-confidence detections, it still relies on a Kalman Filter with fixed noise parameters, limiting adaptability to diverse motion patterns. In this paper, we propose an enhanced ByteTrack framework by integrating an Adaptive Kalman Filter to dynamically adjust noise parameters during prediction. This approach improves state estimation and tracking robustness across varying conditions. Additionally, lightweight object detection models are employed to enable deployment on cost-effective hardware. Experimental results demonstrate that the proposed method achieves improved tracking performance while maintaining computational efficiency, making it suitable for real-time applications.

Keywords: *Multiple Object Tracking (MOT), ByteTrack, Computer Vision, Adaptive Kalman Filter*

[1]School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam

[*]Email: hai.hoanghong@hust.edu.vn

Received: 22/3/2026

Revised: 07/5/2026

Accepted: 25/5/2026

1. INTRODUCTION

The problem of Multiple Object Tracking (MOT) has attracted significant attention in the field of Computer Vision due to its wide range of practical applications. The objective of MOT is not only to detect objects accurately but also to maintain consistent identities for each object throughout the tracking process. Depending on the target objects, MOT can be applied in various domains such as traffic monitoring, industrial automation, and surveillance systems.

Among different applications, pedestrian tracking has become one of the most extensively studied problems. In recent years, numerous approaches have been proposed to improve both tracking accuracy and computational efficiency. Benchmark datasets such as MOT17 and MOT20 have played an important role in evaluating and comparing these methods using standardized metrics, including MOTA, HOTA, MOTP, and IDF1.

A major breakthrough in MOT came from tracking-by-detection approaches, pioneered by the SORT [1] algorithm. Following this direction, several methods such as DeepSORT [2], ByteTrack [3], and BoT-SORT [4] have significantly improved tracking performance by enhancing data association strategies and incorporating appearance features. In particular, ByteTrack achieves a strong balance between simplicity and effectiveness by utilizing both high-confidence and low-confidence detections.

However, despite these improvements, the performance of such methods still heavily depends on the motion modeling stage, where the Kalman Filter plays a crucial role. In many scenarios with complex motion or occlusion, the conventional Kalman Filter may not provide sufficiently adaptive predictions, which can lead to identity switches or tracking inaccuracies.

In this work, we build upon the ByteTrack framework and propose a modification to its Kalman Filter component to improve tracking robustness. The proposed method maintains the simplicity of the original pipeline while enhancing adaptability in dynamic environments. The remainder of this paper is organized as follows: first, we review MOT algorithms and related characteristics; next, we present the proposed methodology along with a discussion of existing approaches; then, we describe the evaluation metrics and

experimental setup; finally, we present the results and conclude with insights and future directions.

2. RELATED WORKS

Tracking by detection Methodology: The close family of ByteTrack is SORT [6] and DeepSORT [7]. The SORT approach focuses strongly on reducing the consumption time, regardless of improving the robust accuracy. In an attempt to reach the real-time multi-object tracking, SORT simply uses the overlap of the IoU indicator to achieve the object association stage, which define a unique id for each object of interest. To be more specific, this step matches the object detected by the object detection party with the predicted object outputted from the Kalman Filter [9].

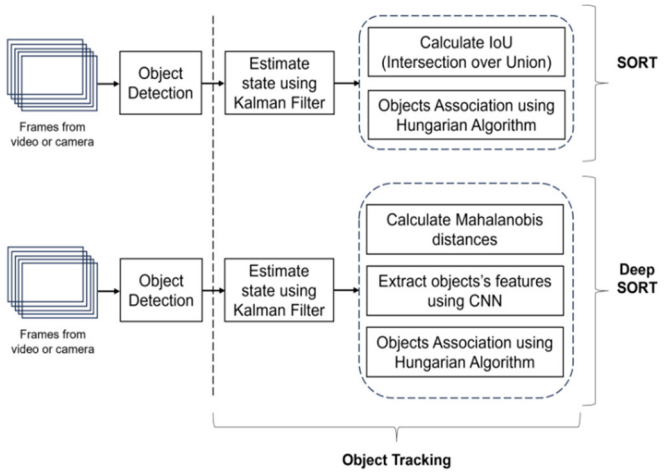


Figure 1. The methodology of SORT and DeepSORT in MOT algorithm

In SORT and deepSORT architectures, they share the same object detector part but differ each other the object association stage as briefly presented in Figure 1. For the detector stage, both retain the simplicity of the object detection procedure and then refines the result with the Kalman Filter. For performing tracking tasks, however, SORT leverages the IoU indicator achieved by the detection stage as the input of the matching level to assign the object ID. In particular, the object is defined with a unique ID using the Hungarian Algorithm [10]. While SORT primarily prioritizes the speed term of tracking, DeepSORT [7] aims to complete the performance by improve the remaining terms, accuracy rate. The Fig. 1 emphasizes the main difference, pointing at the object ID association stage. In DeepSORT, both the Mahalanobis distance of positions and the cosine distance of predicted bounding box are initialized to allow evaluating the Hungarian factor. The Mahalanobis distance is established between predicted and detected objects, whereas the cosine distance is calculated in the feature proximity space. Similarly, empirical experiments conducted by Wojke et al. [7] also show that the appearance similarity is sufficient for the association step. Those such techniques allocate DeepSORT to overcome the challenging cases where object occlusions are present in a long period or when the linear Kalman Filter fail to produce a good prediction.

The difference between SORT and DeepSORT (Figure 1) is that SORT conducts the association step using the IoU calculation, while DeepSORT uses the Mahalanobis distance and cosine distance of objects' features.

3. PROPOSAL METHODOLOGY

3.1. Structure Diagram

Figure 2 illustrates the overall pipeline of the proposed tracking framework. In general, our method follows the track-by-detection paradigm and inherits the two-stage association strategy from ByteTrack. However, we introduce a key modification in the motion modeling stage to enhance tracking robustness.

Given sequential frames, object detections are first obtained and divided into high-score and low-score groups. For high-confidence detections, the first association is performed using Intersection over Union (IoU), optionally combined with appearance embeddings extracted by a ReID model. This extension is inspired by BoT-SORT [8], where fusing motion and appearance cues can improve association accuracy compared to using IoU alone.

Before the association stage, tracklet states are predicted using a Kalman Filter. Unlike the conventional approach, we propose an Adaptive Kalman Filter mechanism that dynamically adjusts its parameters based on object motion. Specifically, a Velocity Monitor module is introduced to estimate the motion characteristics of each tracked object. Based on the observed velocity, the process noise covariance matrix Q is adaptively updated, allowing the filter to better handle both slow and fast-moving objects.

This modification addresses a key limitation of the standard Kalman Filter, where fixed noise parameters may lead to inaccurate predictions under varying motion conditions, especially in the presence of noise or abrupt changes. By adapting Q, the predicted bounding boxes become more reliable, which directly improves the quality of the subsequent data association.

After prediction, the second association stage is applied to match remaining tracklets with low-score

detections, following the original ByteTrack strategy. Finally, tracklets are updated, new tracks are initialized, and lost tracks are removed.

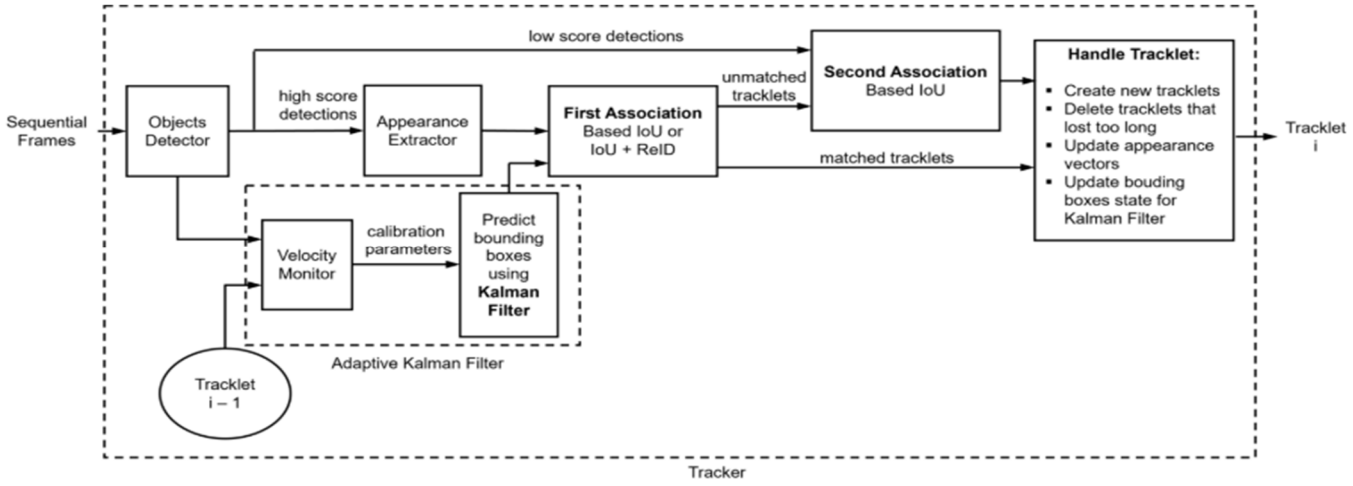


Figure 2. Diagram of our proposal method in the algorithm architecture

In summary, while maintaining the simplicity of ByteTrack, our method enhances the motion model through an adaptive mechanism, which improves prediction accuracy and reduces identity switches in challenging scenarios.

3.2. Proposal Adaptive Kalman Filter

We consider that the process noise matrix Q and the measurement noise matrix R of the Linear Kalman Filter in the ByteTrack algorithm are fixed during all tracking steps, while these fixed values could be inadequate in case of sudden high speed motion or non-constant velocity moving… It affects the results of prediction and updating steps, which can reduce matching performance and tracking efficiency. The classical Kalman Filter for objects with constant velocity is presented briefly here to focus on our proposal.

The estimation state and measurement of objects are provided by the following equations:

$$x_k = F_k x_{k-1} + w_k \quad (1)$$

$$z_k = H_k x_k + v_k \quad (2)$$

Where

F_k is the state transition matrix of the motion model, which calculates the next state from the previous state x_{k-1}

w_k is the process noise, which is the factor of a zero-mean multivariate normal distribution, $\mathcal{N}$, with covariance, Q_k: $w_k \sim \mathcal{N}(0, Q_k)$

H_k is the observation matrix which presents the real state into the observed space

v_k is the measurement noise, which is considered to be a zero mean Gaussian white noise with covariance R_k: $v_k \sim \mathcal{N}(0, R_k)$

The process noise covariance matrix and measurement noise covariance matrix are difficult to estimate because they might change at each time step. The two steps of the entire Kalman Filter are the prediction step and the update step, which can be shown in the following recursive equations:

Predict Step

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (3)$$

Update Step

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k\right)^{-1} \quad (4)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1}\right) \quad (5)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (6)$$

The original Kalman Filter used in the ByteTrack algorithm assumes that the matrix Q and matrix R are invariant. This assumption rarely holds in practice, especially in the case of Multiple Object Tracking, which consists of various objects with different moving characteristics. As a result, we proposed a method to monitor the velocity of each object and decide to change Q according to the moving velocity threshold. Our method will modify the Q value slightly when updating objects' state if their velocities exceed a specific threshold. The proposal method allows the Kalman Filter fit the motion model better than original one by adapting object's moving state. Our Adaptive Kalman Filter can be formulated as follows:

$$Q_{scale} = \begin{cases} 1.0\,, \left(0 < v_{obj} < v_{thresh}\right) \\ \alpha\,, \left(v_{obj} \geq v_{thresh}, 1 < \alpha \leq 1.5\right) \end{cases} \quad (7)$$

$$Q_k = Q_k \cdot Q_{scale} \quad (8)$$

Where

Q_{scale} is the factor that changes the Q_k matrix according to the observation result of the object's velocity

v_{obj} is the current velocity of the object

v_{thresh} is the specific value of the threshold of velocity that makes a decision to change the Q matrix, this value is set to 5.0 in our experiments.

α is the Q_{scale} value in the range of $(1 ; 1.5]$

3.3. Evaluation metrtics

MOTA: Multiple Object Tracking Accuracy [12] score is a metric to evaluate tracking algorithm performances. It calculates the precision of object and trajectory prediction but discards the precision on temporal order space.

MOTP: Multiple Object Tracking Precision proposed in [13] defines the distance of predicted positions of matched object pairs through all frames. It was measured in the Euclidean distance space in [13], which would cause some ambiguities in the image space. As a variant, we instead use the MOTP concept [14] defined as the average overlapping level between all correctly matched hypotheses and their corresponding objects. These indicators reflect the tracking performance from a better perspective.

IDF1: Identification Metric [10] maps the predicted trajectories to the ground truth trajectories. This metric highly focuses on evaluating the association stage where the tracker assigns the object with an unique ID [12, 15].

HOTA: Higher Order Tracking Accuracy is thoroughly analyzed in [12, 16], it covers a wide range of evaluation metrics for different tracking components. It can compensate for the drawbacks of other commonly used metrics.

3.4. Experimental setup

In this article, we implemented experiments on the popular hardware, a laptop with GPU card support. Particularly, the laptop is equipped with the CPU Intel Core i5-8265U 1.60GHz, RAM 12GB DDR4 run on Ubuntu 20.04 LTS. The goal is to investigate the performance of the tracker under the application of a regular computer. With this hardware, we deployed AdaptTrack on MOT17 train datasets, including MOT17-02, MOT17-04, MOT17-05, MOT17-09, MOT17-10, MOT17-11, MOT17-13 [14]. Regarding the object detection module used in the tracker pipeline, we use the YOLOX-nano instead of YOLOX as original ByteTrack because of the limited configuration of a popular laptop with an ordinary GPU to make a trade-off between tracking speed and accuracy.

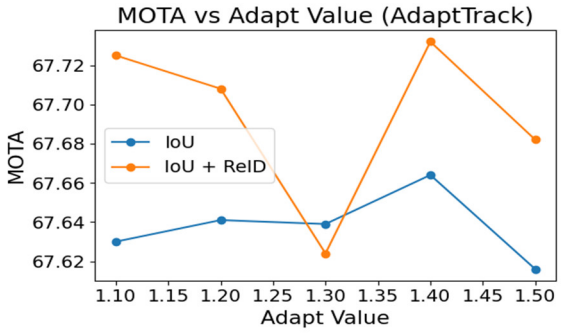
4. RESULTS AND DISCUSSION

4.1. Results

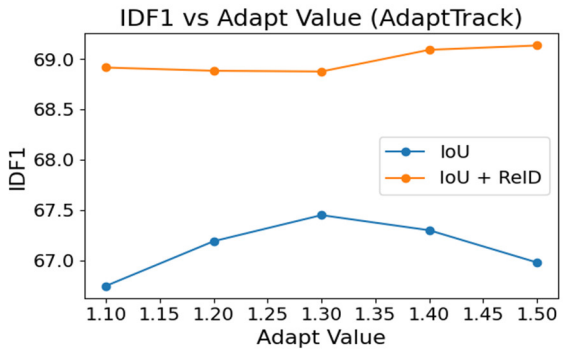
Table 1 shows the tracking result of our AdaptTrack algorithm compared to the original ByteTrack and BoT-SORT pipeline. BoT-SORT is still outstanding with the rank first of almost all scores. However, having a complicated process makes it slow significantly, 11 FPS with IoU and CMC only, even 2 FPS when using a combination of IoU, CMC, and ReID model for embedding apperance. Our method with the Adaptive Kalman Filter by monitoring the velocity of objects improve tracking scores in general compared to the ByteTrack baseline. AdaptTrack reaches the best performance with the adapt value $\alpha = 1.3$ when using IoU only. The higher tracking result is also presented with the same FPS on the baseline method (21 FPS). Our method has better achievement when applying the ReID model to extract the embedded appearance of objects to match in the next step, but it reduces the speed (down to 2 FPS). The best adapt value in this case is $\alpha = 1.4$.

Table 1. AdaptTrack Performance

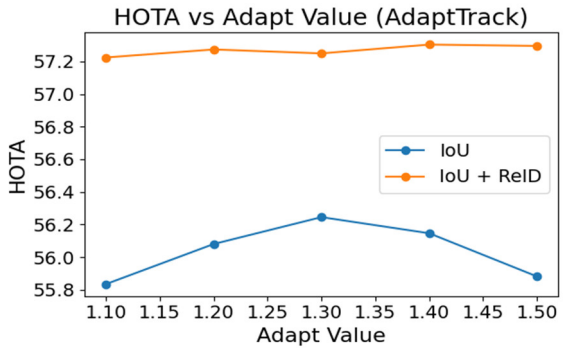
Method	w/ Re-ID	MOTA ↑	IDF1 ↑	HOTA ↑	IDSW ↓	FPS
ByteTrack (Original, GitHub)	✗	69.0	66.3		531	21
ByteTrack (Ours Rerun)	✗	67.47	67.05	56.103	451	21
BoT-SORT	✗	68.83	68.91	57.254	367	11
BoT-SORT-ReID	✓	68.53	70.72	58.504	356	2
AdaptTrack (Ours, IoU, adapt = 1.3)	✗	67.64	67.45	56.245	449	21
AdaptTrack (Ours, IoU+ReID, adapt = 1.4)	✓	67.73	69.09	57.304	410	2



a) MOTA score



b) IDF1 score



c) HOTA score

Figure 3. Comparison of the performances of AdaptTrack using IoU only or the combination of IoU and ReID

We conducted many experiments to find out which adapt value is the best. This process is shown on the two following line charts. IoU and ReID under different "adapt value". The results are from the train dataset of MOT17.

4.2. Discussion

Our proposal method takes advantage of the simple architecture of the original ByteTrack while it integrates our version of the Adaptive Kalman Filter and even the ReID model of BoT-SORT. This combination improved the tracking performance of the ByteTrack baseline, but less complicated compared to BoT-SORT. The drawback of our method is that it depends on choosing an "adapt value", but does not generate this factor automatically. Our experiments are the series of trial and error before finding out the best "adapt value" for each situation. That makes the implementation slow in some cases.

5. CONCLUSION

In summary, our research introduced a novel method of predicting objects' states using an Adaptive Kalman Filter that is useful to apply for MOT. Our AdaptTrack performed better than the original ByteTrack on the MOT17 train dataset. Also, our tracker still has several limitations that need to be solved in future work. However, with our simple proposal method, the AdaptTrack could be used to improve some previous trackers that are based on the tracking-by-detection method. Future work will focus on deepening our understanding of the algorithm and its methodology to identify potential improvements.

ACKNOWLEDGEMENTS

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2025-PC-028

REFERENCES

[1]. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B., "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 3464-3468, IEEE, 2016. https://doi.org/10.1109/ICIP.2016.7533003

[2]. Wojke N., Bewley A., Paulus D., "Simple online and realtime tracking with a deep association metric," *arXiv:1703.07402*, 2017. https://doi.org/10.48550/arXiv.1703.07402

[3]. Zhang Y., Wang C., Pang J., Shi B., Duan C., Luo P., Wang X., Wang W., Loy C. C. "ByteTrack: Multi-object tracking by associating every detection box," *arXiv:2110.06864*, 2021. https://doi.org/10.48550/arXiv.2110.06864

[4]. Aharon N., Orfaig R., Bobrovsky B. Z., "BoT-SORT: Robust associations multi-pedestrian tracking," *arXiv:2206.14651*, 2022. https://doi.org/10.48550/arXiv.2206.14651

[5]. Kálmán R. E., "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 82(1), 35-45, 1960. https://doi.org/10.1115/1.3662552

[6]. Ge Z., Liu S., Wang F., Li Z., Sun J., "YOLOX: Exceeding YOLO series in 2021," *arXiv:2107.08430*, 2021. https://doi.org/10.48550/arXiv.2107.08430

[7]. Kuhn H. W., "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 2(1-2), 83-97, 1955. https://doi.org/10.1002/nav.3800020109

[8]. Bashar M., Islam S., Hussain K. K., Hasan M. B., Rahman A. B. M. A., Kabir M. H., "Multiple object tracking in recent times: A literature review," *arXiv:2209.04796,* 2022. https://doi.org/10.48550/arXiv.2209.04796

[9]. Bernardin K., Stiefelhagen R., "Evaluating multiple object tracking performance," *EURASIP Journal on Image and Video Processing*, 2008, Article 246309, 2008. https://doi.org/10.1155/2008/246309

[10]. Milan A., Leal-Taixé L., Reid I., Roth S., Schindler K., "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831*, 2016. https://doi.org/10.48550/arXiv.1603.00831

[11]. Ristani E., Solera F., Zou R. S., Cucchiara R., Tomasi C., "Performance measures and a data set for multi-target, multi-camera tracking," *arXiv:1609.01775*, 2016. https://doi.org/10.48550/arXiv.1609.01775

[12]. Luiten J., Osep A., Dendorfer P., Torr P., Geiger A., Leal-Taixé L., Leibe B., "HOTA: A higher order metric for evaluating multi-object tracking," *arXiv.2009.07736*, 2020. https://doi.org/10.48550/arXiv.2009.07736