2024 IEEE Intelligent Vehicles Symposium (IV)
June 2-5, 2024. Jeju Island, Korea

AETrack: An Efficient Approach for Online Multi-Object Tracking

Xurui Wang, Yuxuan Han, Qingxiao Liu, Ji Li, Boyang Wang, Haiou Liu, Huiyan Chen

2024 IEEE Intelligent Vehicle Symposium (IV) | 979-8-3503-4881-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/IV55156.2024.10588483

Abstract— **Tracking by detection(TBD) method has achieved great improvements for its high efficiency, extensibility and portability, but it still struggles on computational efficiency. Many recently proposed methods improve performance by integrating appearance similarity and simply extract appearance feature for all the targets. This results redundant calculations as some targets can already be easily tracked without feature extraction, such as targets walking alone. In this work, we tackle the efficiency problem from a new perspective and propose AETrack, an efficient approach for online multi-object tracking(MOT), which integrates three association metrics through a novel cascaded matching strategy. Instead of simply computing all the association metrics for all tracklets, our matching strategy dynamically chooses and fuses the metrics for each tracklet considering both effectiveness and efficiency. Inference speed is boosted greatly and accuracy is still competitive. AETrack achieves 64.7 HOTA on MOT17 test set while running at 58 FPS and 62.8 HOTA on MOT20 at 52 FPS. Our code and models will be public soon.[1].**

I. Introduction

The goal of multi-object tracking(MOT) is to detect interested targets in every frame of a video and associate them across frames to keep track of their movements. In recent years, many trackers follow the pipeline of tracking-by-detection(TBD) and have made great progress[1], [2], [3]. However, as the tracking accuracy is improved, complexity increases and calculation time is sacrificed[4], [2], [5]. It is still a challenge to achieve real-time inference speed, especially when there are a large number of targets in the image. Real-time application faces two main challenges. First, The time complexity of the Hungarian algorithm[6] is $O(N^3)$, where N is the number of vertices in the bipartite graph. Thus the complexity of the binary matching algorithm increases cubically with the number of objects. Second, although deep appearance descriptor brings conductive clues for lost track recovery, maintaining the feature bank takes both time and computation resource.

Association metric determines how the tracklets and detections are matched, and is the focus of the research. Many recent trackers associate tracklets and detections based on intersection over union(IoU)[7], [5], which may lead to incorrect matches. We found that IoU is misleading in crowded scenes for its orientation sensitivity, and replace it with Mahalanobis distance[8] .

In this work, we propose **A**n **E**fficient approach for online multi-object **Track**ing(**AETrack**) that runs faster than main leading trackers with comparable tracking performance. Three association metrics are designed, including one fundamental metric for basic matching and two other metrics for track competitions and occlusions in complex scenes. Instead of fusing these metrics at once, metrics are hierarchically computed by a novel matching method, which enhances the running speed significantly. Our motivation is to keep our tracker high computational efficiency in various scenarios and we observe the convention that targets easy to track should be associated by fundamental metric and separated from the whole target set, then the rest targets are associated through higher-order metric that takes time to compute like fused cost. Thus, we track all the targets just like reeling silk from cocoons. Comparable performance is achieved and inference speed is improved. Our main contributions are as follows:

1) We design a novel cascaded matching method, which decomposes matching candidates according to complexity of target context, speeding up matching while maintaining accuracy. We also apply this method on some other trackers, which shows its general effectiveness and portability.
2) For association metrics, we abandon IoU-based metric and propose three metrics based on motion clue, pseudo-depth and appearance similarity, which significantly reduces incorrect matches.

II. Related work

A. Association Metric

Association metric is used to guide the track to match with the correct detection in the current frame. In most recent trackers, association metric is the combination of motion cost and appearance cost. For motion cost, many trackers[4], [3], [9], [10], [11] use IoU-like metrics for its easy computation and extensibility. Mahalanobis distance is firstly only used in [12] as a gate to avoid infeasible assignments, later it is directly used as motion metric for data association in [13], [5]. OC-SORT[7] proposed Observation-Centric Momentum(OCM) based on velocity consistency which helps to distinguish confused candidates during association. Appearance cost is usually obtained by computing the cosine similarity of the appearance features. Many trackers[14], [4], [15] crop the detection boxes and extract deep appearance feature using a stand-alone CNN model trained on person re-identification datasets. Appearance cost helps with long-term occlusions but introduces computational cost and falls short when target are covered or represented coarsely.

This work was supported by the National Natural Science Foundation of China under Grant 52172378, and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars.(Corresponding author: Haiou Liu)

All the authors are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China. (e-mail: bit_lho@bit.edu.cn)

[1] https://github.com/wxr66777/AETrack

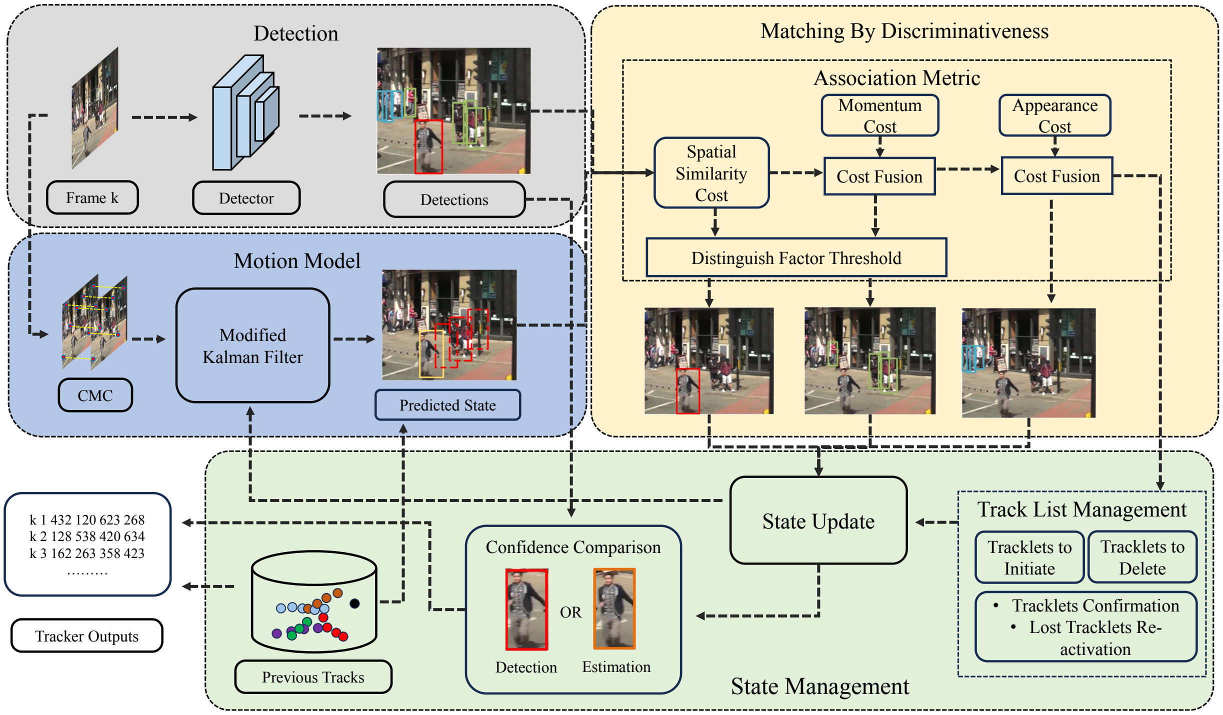

Fig. 1. Pipeline of our method. AETrack consists of four main parts: Detection, Motion Model, Matching By Discriminativeness(MBD) and State Management. See Section. III-A for details.

In this paper, we make full use of multi-modal information obtained from the image stream, based on which three effective metrics are designed to produce reliable matching results.

B. Matching Strategy

Matching strategy is the core of multi-object tracking. Most TBD methods build a linear assignment problem to match tracks and detections and solve it by Hungarian algorithm[16], [17], [4], [15], [5], [7]. As solving a global assignment problem is complex and time-consuming, especially when the target set is large, many researches focus on decomposing the target set and performing cascaded matching. DeepSORT[12] decomposes the target set according to the length of lost time, and gives priority to matching the latest tracked targets, which simultaneously helps to solve the undesirable favour of the Mahalanobis distance to missing targets. ByteTrack[1] divides the detection results into high confidence and low confidence set according to detection confidence, first matches the detection results with high confidence detections, and then matches the unmatched tracks with low confidence detections. SparseTrack[18] divides the whole target set into several subsets according to the pseudo-depth level and perform cascaded matching from near to far.

Our new matching method hierarchically computes the association metrics for each target according to the context complexity, thus avoids redundant computation and improves running speed.

III. Method

In this section, we introduce main modules of AETrack and elaborate our contributions for online multi-object tracking.

A. Overview

The pipeline of our tracker is shown in Fig. 1. When a new frame comes, detections are obtained by the detector and send to matching module as the observation. Meanwhile, motion model predict the location of target through our Kalman filter[19]. In our tracker, the state vector was chosen to be a eight-tuple, $X = [x,y,w,h,\dot{x},\dot{y},\dot{w},\dot{h}]^T$, where (x,y) are the 2D coordinates of the object center in the image plane. w is the bounding box width and h is the bounding box height. Camera motion compensation(CMC) compute an affine matrix to compensate prediction error caused by camera motion. With predictions and observations, our matching module(MBD) computes association metrics and produce matched, lost and new targets by matching cascade. Then matching results are used to update track states. Lastly, due to that observation usually gives more accurate position but is more susceptible to occlusion, confidence comparison is used to determine whether estimation or observation should be used as the tracker output. Observations that are not assigned to any tracklets are initialized as new tracklets but will be deleted if they disappear in next three frames. Tracklets matched with no observation are deactivated and their states will not be updated. We terminate tracklets with no update in the nearest 30 frames.

B. Association Metrics

In this section, we introduce three similarity calculation methods, which make full use of existing prediction and observation information.

1) Spatial Similarity Cost(SSC): Although most recent trackers use IoU between prediction and measurement as spatial similarity cost, but we argue that IoU-based similarity cost can lead to unfair matching results due to the orientation sensitivity of IoU. To tackle this problem, we simply replace IoU metric with squared Mahalanobis distance as our spatial similarity cost, removing orientation sensitivity while both position and shape are considered.

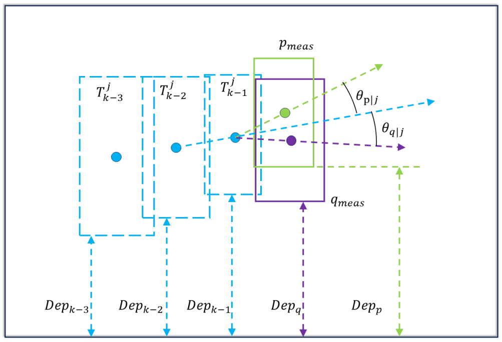

Fig. 2. 3D Momentum Cost(3DM). The outer box indicates the border of the frame, A_{k-3}, A_{k-2} and A_{k-1} are previous tracking results of a target denoted by dashed blue bounding boxes with a circle at the center point, p_{meas} and q_{meas} are two detections near the last tracking results, $\theta_{p|j}$ and $\theta_{q|j}$ are two angles denoting the bias between the ideal orientation and that of two detections.

2) 3D Momentum Cost(3DM): Before we introduce appearance comparison which brings more time cost, detection candidates are judged under the principle of 3D momentum consistency. 3D denotes the two dimensional position of the center point in the image and one dimensional pseudo-depth[18] calculated as Formula. 1, where H denote the height of the image and y, h mean the same as the state variables. Pseudo-depth is one simple but effective way to measure relative depth between different targets with two prior hypotheses that all the targets are above the ground and on the same plane. By integrating 2d direction cost and pseudo-depth into the Formula. 2, we design a momentum cost which guides the track to match with the correct detection when occlusion happens among targets.

$$Dep = H - y - 0.5 * h \tag{1}$$

$$C_{3DM}(i,j) = \frac{\Delta Dep_{i|j}}{\max_i \Delta Dep_{i|j}} + \gamma \frac{\theta_{i|j}}{\pi} \tag{2}$$

where $\Delta Dep_{i|j}$ is the absolute value of the difference between the pseudo-depth of i-th detection and the prediction of j-th track. As shown in Fig. 2, $\theta_{i|j}$ is formed by two lines: the line linking last two observations of the j-th track and the line linking the last one observation of the j-th track and the i-th detection.

3) Appearance Similarity(AS): Deep appearance feature plays an important role in target matching, especially when position clue is of great uncertainty. We adopted a strong baseline(BoT[20]) from the FastReID library, and fine-tune it on the market1501[21]. To relieve the influence of image degradation like occlusion and motion blur, we reduce the weight of appearance similarity according to the detection confidence. When detection confidence(c_j) is less than the threshold(c_0), appearance cost is formulated as:

$$C_{Appr}(i,j) = (c_j - c_0) * (1 - \frac{\alpha_i \beta_j}{\|\alpha_i\| \|\beta_j\|}) \tag{3}$$

where α_i and β_j is the appearance feature vector of the i-th track and the j-th detection respectively. When c_j is less than c_0, appearance feature would not be computed due to unreliable detection quality.

C. Matching by Discriminativeness(MBD)

We design a novel matching strategy that allows us to call the computation of association costs hierarchically. For above introduced three association metrics, the later is only applied on tracklets that the former does not work well. We arrange them in order considering computation complexity and fusion effects. Discriminativeness, which has been used for cost fusion, feature evaluation in some researches[17], [22], is utilized in matching in our method. We compute a distinguish factor(DF) that determines whether the next cost function should be called and how the costs should be fused. The value of DF measures how easy or hard for the tracker to distinguish the target from other candidates. For p-th tracklet and q-th detection, we have:

$$DF(p,q) = DF(p) + DF(q) \tag{4}$$

$$DF(p) = -\min_i C(i,p) + \min_{j \neq i} C(j,p) \tag{5}$$

$$DF(q) = -\min_m C(q,m) + \min_{n \neq m} C(q,n) \tag{6}$$

where $C(i,j)$ denotes the element at i-th row and j-th column of the cost matrix. The fusion of the last used cost C_{last} and the next using cost C_{next} are formulated as follows:

$$C_{fused}(i,j) = \lambda * C_{last}(i,j) + (1-\lambda) * C_{next}(i,j) \tag{7}$$

$$\lambda = \frac{DF_{last}(i,j)}{DF_{last}(i,j) + DF_{next}(i,j)} \tag{8}$$

Here λ is actually an adaptive weight factor that are dynamically set according to the DF value. It makes cost with higher DF value takes a larger part in the fused cost, thus the fused cost can be more conducive to matching.

Our algorithm flow is shown in Algorithm 1. Firstly, we compute the SSC for all targets to fill the cost matrix used for the first association, then $DF(p,q)_1$ is calculated and if $DF(p,q)_1 \geq \theta_1$, p-th tracklet and q-th detection are added into the candidates and we perform Hungarian algorithm to match the candidates. If $DF(p,q)_1 < \theta_1$, we leave the tracklet and detection for the next association. Meanwhile,

Algorithm 1 Matching Cascade of AETrack

```
Input: Tracklets $T_{in}$, Detections $D_{in}$;
    First and second DF threshold $\theta_1, \theta_2$;
    Spatial similarity cost $SSC$, 3D momentum cost $3DM$,
    Appearance similarity cost $AS$, Cost fusion $CF$;
Output: Matched tracklets $T_{tracked}$, deactivated tracklets
    $T_{deactive}$, new tracklets $T_{new}$
First association:
$C_{SSC} \leftarrow SSC(T_{in}, D_{in})$
for $T_i$ in $T_{in}$ do
    for $D_j$ in $D_{in}$ do
        Compute the first DF: $DF_1(i,j)$ using $C_{SSC}$;
        if $DF_1(i,j) > \theta_1$ then
            $T_{target} \leftarrow T_i$, $D_{target} \leftarrow D_j$;
        else
            $T_{remain} \leftarrow T_i$, $D_{remain} \leftarrow D_j$;
        end if
    end for
end for
Associate $T_{target}$ and $D_{target}$ using $C_{SSC}$.
$T_{tracked} \leftarrow$ matched tracklets.
$T_{remain} \leftarrow$ unmatched tracklets, $D_{remain} \leftarrow$ unmatched
    detections.
Second association:
$C_{3DM} \leftarrow 3DM(T_{remain}, D_{remain})$;
$C_{fused} \leftarrow CF(C_{SSC}, C_{3DM})$
for $T_i$ in $T_{remain}$ do
    for $D_j$ in $D_{remain}$ do
        Compute the second DF: $DF_2(i,j)$ using $C_{fused}$;
        if $DF_2(i,j) > \theta_2$ then
            $T_{target} \leftarrow T_i$, $D_{target} \leftarrow D_j$;
        else
            $T_{re-remain} \leftarrow T_i$, $D_{re-remain} \leftarrow D_j$;
        end if
    end for
end for
Associate $T_{target}$ and $D_{target}$ using $C_{fused}$.
$T_{tracked} \leftarrow$ matched tracklets.
$T_{re-remain} \leftarrow$ unmatched tracklets, $D_{re-remain} \leftarrow$ un-
    matched detections.
Third association:
$C_{Appr} \leftarrow AS(T_{re-remain}, D_{re-remain})$;
$C_{fused} \leftarrow CF(C_{fused}, C_{Appr})$;
Associate $T_{re-remain}$ and $D_{re-remain}$ using $C_{fused}$.
$T_{tracked} \leftarrow$ matched track indices
$T_{deactive} \leftarrow$ unmatched tracklets, $T_{new} \leftarrow$ unmatched
    detections
```

unmatched tracklets and detections are also added to candidates for the next association. The next two stages follow the similar steps, but the main difference is that candidates comes from the tracklets and detections left for this stage or unmatched in last association. Thus, we have matched all tracklets and candidate boxes step by step from simple to complex.

IV. EXPERIMENTS

A. Datasets and Metrics

We conduct experiments on MOT17[23] and MOT20[24] datasets to evaluate our method. MOT17 consists of 7 sequences for training and 7 sequences for testing where targets move linearly with camera motion happens while scenes in MOT20 is much more crowded, with 4 sequences for training and 4 sequences for testing. CLEAR[25], acronym for "Classification of Events, Activities and Relationships", is most commonly used metrics which includes MOTA, FP, FN, IDs, etc. MOTA, FP, FN and IDs denote "Multi-Objective Tracking Accuracy", "False Positive", "False Negative" and "Number of Identities Switches" respectively. MOTA is calculated based on FP, FN and IDs and shows more about detection performance. HOTA[26], abbreviation for "Higher Order Tracking Accuracy", is another metric that measures overall performance of both detection and tracking, which combines detection score DetA and association score AssA. Matching accuracy and consistency is measured by IDF1[27], which is the ratio of correctly identified detections over the average number of ground-truth and computed detections. FPS, namely "Frames Per Seconds", indicates inference speed and algorithm complexity. In our experiments, MOTA, HOTA, AssA, IDF1 and FPS are used to evaluate tracking performance.

B. Implementation Details

Considering that our tracker is modified based on the framework of ByteTrack[1], We choose ByteTrack as the baseline method and the public detector YOLOX[28] is set as the external detector. We simply use weight and training configuration same as the baseline method. For MOT17 test set, we train YOLOX-X on the combination of MOT17 training set, CrowdHuman[29] and CityPersons[30]. For MOT20 test set, we train the detector on CrowdHuman and MOT20 train set. We set the DF threshold θ_1 and θ_2 as 0.35 and 0.4 respectively for experiments on MOT17, 0.3 and 0.35 for MOT20. All the experiments were run on a desktop with 12-th Gen Intel Core i7-12700K and NVIDIA GeForce RTX 3080 GPU.

C. Evaluation

We evaluate AETrack on the test set of MOT17 and MOT20 and results are shown in Table. I. To avoid the possible influence caused by the detector, we put more emphasis on the results of trackers that use the same detector(YOLOX). Our tracker achieves comparable performance with huge increase in inference speed on MOT17 test set. Compared with the baseline, AETrack achieves an increase of MOTA by 0.4, IDF1 by 1.8 and HOTA by 1.6 with much less time consumption, which verifies our improvements and demonstrates that our tracker is simple and efficient. Evaluation results on MOT20 also show great performance. AETrack obtains a growth of IDF1 by 1.6 and HOTA by 1.5. Most scenes in MOT20 is much more crowded than that in MOT17, meaning much more targets to track, thus the complexity increases and association takes more time.

TABLE I

EVALUATION RESULTS ON MOT17 AND MOT20 TEST SETS. METHODS MARKED WITH * SHARE THE SAME DETECTOR. THE TWO BEST RESULTS FOR EACH METRIC ARE BOLDED AND HIGHLIGHTED IN RED AND BLUE.

	MOT17				MOT20			
Method	HOTA	IDF1	MOTA	FPS	HOTA	IDF1	MOTA	FPS
FairMOT	59.3	72.3	73.7	37.6	54.6	67.3	61.8	**25.9**
CSTrack	59.3	72.6	74.9	30.8	54.0	68.6	66.6	15.8
TransTrack	54.1	63.5	75.2	20.6	48.5	59.4	65.0	7.2
OC-SORT*	63.2	77.5	78.0	29.0	62.4	76.3	75.7	18.7
StrongSORT*	64.4	79.5	79.6	7.5	62.6	77.0	73.8	1.5
BoT-SORT*	64.6	79.5	80.6	8.4	62.6	76.3	77.7	6.6
SparseTrack*	**65.1**	**80.1**	**80.9**	19.9	**63.4**	**77.3**	**78.2**	12.5
UCMCTrack*	**65.7**	**81.0**	80.6	**49.6**	**62.8**	**77.4**	75.6	24.8
ByteTrack(Baseline)*	63.1	77.5	80.3	29.6	61.3	75.2	**77.8**	17.5
AETrack(ours)*	**64.7**	**79.3**	**80.7**	**58.2**	**62.8**	**76.8**	**77.6**	**52.3**

TABLE II

ABLATION ANALYSIS OF THREE COMPONENTS ON MOT17 VALIDATION SET.

Method	Motion Model	Association Metrics	MBD	HOTA	IDF1	MOTA	FPS
Baseline(ByteTrack)				68.1	79.4	76.4	29.3
Baseline+column1	✓			68.3	79.7	76.8	32.5
Baseline+column1-2	✓	✓		69.0	80.9	76.9	18.9
Baseline+column1-3(AETrack)	✓	✓	✓	**69.5**	**81.3**	**77.1**	**60.4**

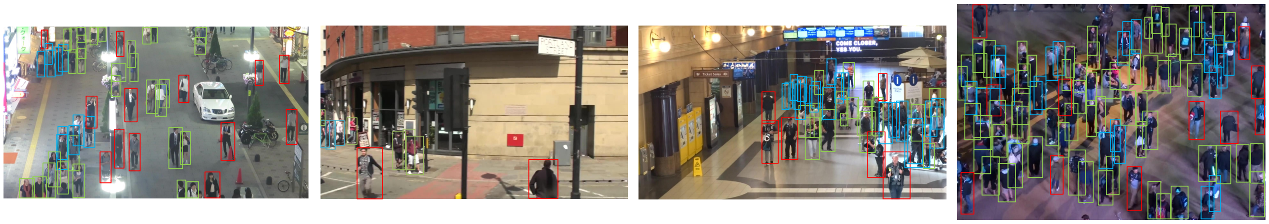

Fig. 3. Visualization of target set decomposition in MOT17 test set. Four images are selected from MOT17-05, MOT17-10, MOT20-02, MOT20-04 respectively. Targets are shown in bounding boxes, while different color denotes in different matching stage the target is matched. Red means the target is relatively easy to track and matched in the first association, green denotes targets matched in the second association and blue boxes means targets are matched in the last association.

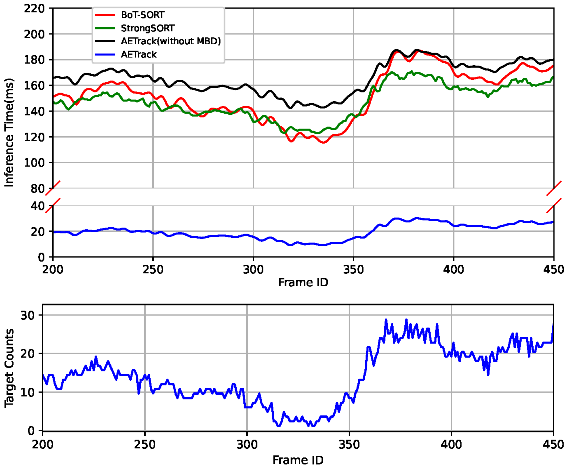

Fig. 4. Results of inference speed test on MOT17-14 sequence.

This can be seen from the evident drop of FPS in most trackers. Nevertheless, our tracker still keeps relatively high processing speed, which indicates that our tracker tackles the efficiency problem thoroughly and we give elaborate study in Section. IV-D.2.

D. Ablation Study

1) Components Analysis: The importance of each component is verified by ablation experiments. Our ablation experiments are conducted on MOT17 validation set. Results are summarized in Table. II. ByteTrack is set as the baseline with detection score threshold of 0.6. As the results shows, our three components all have contributions on the performance improvement. And specifically, our motion model and confidence comparison mainly contributes to higher MOTA and HOTA while association metrics bring more effects on improving IDF1. Our new cascaded matching boosts the speed significantly.

Several examples in MOT17 and MOT20 test sets are selected and listed here to help to show how our MBD works under some typical cases. As illustrated in Fig. 3, MBD separates targets that are easy to track, such as targets moving alone, to accelerate matching process and leaves targets close to each other in second and third matching stage to ensure accuracy.

2) Inference Speed: We choose one specific sequence(MOT17-14) to perform inference speed test, and compare our tracker with two other trackers(BoT-SORT and StrongSORT). To see the effect of MBD module, we make a control group by replacing our MBD with vanilla matching. Fig. 4 demonstrates how inference time of each tracker varies with the target counts from frame 200 to 450. The inference time encompasses detection and tracking for each frame, but excludes video decoding and image acquisition. As the figure shows, these three trackers experience similar trending: the inference time extends as the number grows and shortens as it drops, which is quite explainable as growing target count brings more times of cost computation and increases the complexity of data association. But thanks to our MBD, this variance on our tracker is much less dramatic in comparison with other trackers. AETrack can maintain fast running speed when the more targets show up, which proves that our tracker successfully settles the issue of low inference speed under crowded scenarios and our improvement is robust to the fluctuation of target counts.

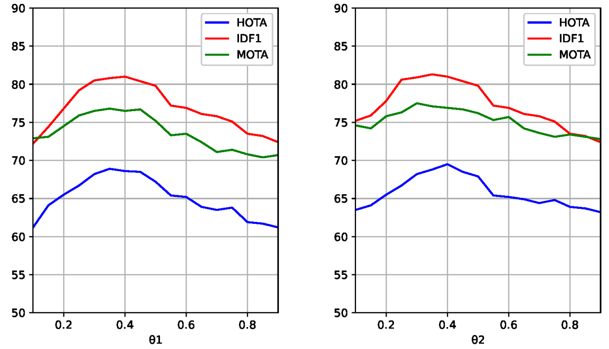

Fig. 5. Results of adjusting the DF thresholds on MOT17 validation set.

3) Distinguish Factor(DF): Threshold of DF is a sensitive hyper-parameter, which determines how confident the tracker is to assume the target is easy or hard to track and should to be carefully tuned. First, we tune the first threshold(θ_1) from 0.2 to 0.8 without using appearance cost and the second threshold while comparing the HOTA, IDF1 and MOTA evaluated on MOT17 validation set. Results are shown in Fig. 5. We give HOTA the highest priority and choose 0.35 as the setting value. Then we hold 0.35 as the value of θ_1, and see how the benchmarks varies with the second threshold(θ_2) from 0.2 to 0.8. Comparing the results, we notice that when the threshold is set lower, more targets are decomposed from original set and matched without further cost computation, thus association takes less time. However, low threshold leads to incorrect matching, which can be seen from the drop of IDF1, IDs and HOTA. In our settings, we give HOTA the highest priority and choose 0.35 and 0.4 as the final implementation settings for MOT17 dataset. Similar experiment is conducted on MOT20 dataset and two thresholds are set to 0.3 and 0.35.

4) Integration to Different Trackers: Our MBD can be applied to multiple trackers that use vanilla matching or BYTE[1] with two or more association metrics. To further explore the effect of our matching cascade, we integrate MBD into four different trackers, including Bot-SORT[4], SMILEtrack[15], CSTrack[3] and StrongSORT[5]. According to the different designs of these trackers, two modes are designed for better compatibility. For trackers that using vanilla matching, we use the motion cost in the first association and use the fused cost for the second association. Taking StrongSORT as an example, the IOU-based motion cost is used for first association and cost for second association is the fusion of motion cost and appearance cost. For trackers that has integrated BYTE, we only integrate cost fusion following the Formula. 7 and perform BYTE using the fused cost.

TABLE III

Effects of integrating MBD into four different trackers on MOT17 validation set.

Trakcer	w/MBD	mode	HOTA	IDF1	MOTA	FPS
StrongSORT			69.5	82.1	78.4	7.5
StrongSORT	✓	1	69.2	81.8	78.2	25.0
CSTrack			60.8	72.3	68.4	16.7
CSTrack	✓	2	60.5	72.1	68.2	30.6
Bot-SORT			69.1	81.8	78.4	6.5
Bot-SORT	✓	2	68.9	81.5	78.5	21.1
SMILEtrack			69.8	79.8	77.5	7.5
SMILEtrack	✓	2	69.5	79.2	77.2	19.7

Evaluation results are illustrated in Table. III, and all the confidence thresholds are set using principle introduced in Section. IV-D.3. We can see that MBD brings stable and evident enhancements over inference speed while maintaining comparable performance on other metrics, exploring the real-time applications of these trackers.

V. Conclusions

In this paper, we propose an efficient tracker called AETrack for online multi-object tracking. Three robust metrics leveraging multi-modal information are proposed and used for data association. For computation efficiency, we present a new cascaded matching method(MBD) that optimizes the complexity by target set decomposition. Experiments are conducted on MOT datasets which proves that our tracker achieves great improvement on running speed with comparable performance. It is worth to mention that our MBD is a portable component and can be applied to other trackers. By integrating MBD into four other trackers, we reduce their runtime and explore their potential for real-time applications.

References

[1] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” in *European Conference on Computer Vision*. Springer, 2022, pp. 1–21.

[2] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, “Dancetrack: Multi-object tracking in uniform appearance and diverse motion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 993–21 002.

[3] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, “Rethinking the competition between detection and reid in multiobject tracking,” *IEEE Transactions on Image Processing*, vol. 31, pp. 3182–3196, 2022.

[4] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, “Bot-sort: Robust associations multi-pedestrian tracking,” *arXiv preprint arXiv:2206.14651*, 2022.

[5] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, “Strongsort: Make deepsort great again,” *IEEE Transactions on Multimedia*, 2023.

[6] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[7] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9686–9696.

[8] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.

[9] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[10] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 352–12 361.

[11] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2018, pp. 1–6.

[12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[13] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *European Conference on Computer Vision*. Springer, 2020, pp. 107–122.

[14] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 2809–2819.

[15] Y.-H. Wang, "Smiletrack: Similarity learning for multiple object tracking," *arXiv preprint arXiv:2211.08824*, 2022.

[16] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[17] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, pp. 3069–3087, 2021.

[18] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth," *arXiv preprint arXiv:2306.05238*, 2023.

[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[20] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[21] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1116–1124.

[22] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 3025–3029.

[23] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[24] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv preprint arXiv:2003.09003*, 2020.

[25] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[26] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[27] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*. Springer, 2016, pp. 17–35.

[28] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[29] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[30] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3213–3221.