2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

FocoTrack: Multi Object Tracking by Focusing On Overlap at Low Frame Rate

Jae Hyeok Lee, Jae-Hyeon Park, and Dong Eui Chang*

***Abstract*— Multi-object tracking (MOT) presents a crucial challenge in robotics. Due to limited resources embedded in robots, one time step per processing time for algorithms can be considerably large. This scenario necessitates the operation of MOT at a low frame rate. However, algorithms within the MOT research field have been constructed around datasets functioning at 10–30 frames per second (fps) which can be difficult to operate in the limited resources. In response to it, we introduce a new algorithm, called FocoTrack, which maintains tracking ability in four situations, one of which is when objects are overlapped by each other. Our algorithm exhibits remarkable performance without using any deep appearance descriptor, surpassing existing MOT methods which even use the deep appearance descriptor on a 2.5 fps dataset. We also demonstrate strong results with our algorithm on DanceTrack dataset at 20 fps and provide comprehensive insights through detailed analysis of our tracking model.**

I. INTRODUCTION

Multi-object tracking (MOT) plays an essential role in various domains, including autonomous driving [1], [2], surveillance operations [3], and more. Within a context of autonomous mobile robots, the ability of MOT to identify locations and motions of surrounding objects is vital. By analyzing the historical trajectory of objects through MOT, predictions about their future path can be made [4], offering valuable insights for planning and navigation of robot movement [5]. Using MOT in robotics presents a critical challenge. In commercializing robotic solutions, cost is an inevitable consideration. This often translates to constraints on the computational resources of robots, and one timestep needed for all algorithms functioning can be bigger. For instance, an object detection model YOLOv5s [6] on a Jetson Nano clocks at a speed of 4.64 fps [7]. Incorporating additional modules might further reduce this rate. It is imperative, therefore, that MOT algorithms remain efficient even at a reduced frame rate. However, most existing MOT studies focus on enhancing performance using high frame rate datasets of 10 fps or above, where object motion and occlusion behave differently compared to lower frame rates. Figure 1 illustrates this contrast by comparing occlusion scenarios at high and low frame rate. Row (a) in the figure shows a minimal variation in occlusion between frames t and $t+1$ in a high frame rate dataset. In contrast, row (b) displays significant overlap within a single timestep in a low frame rate dataset. This attribution has the effect of diminishing the efficiency of previous MOT algorithms.

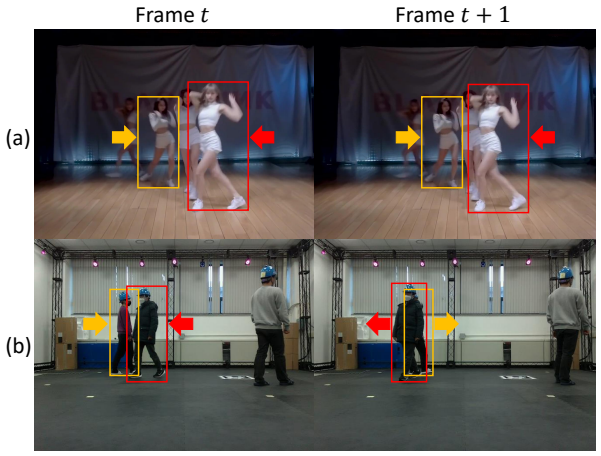

Fig. 1: (a) 20fps DanceTrack dataset, (b) 2.5fps dataset. Two overlapping objects are represented by red and yellow boxes, respectively, and an arrow indicates the direction the objects are facing. In (a), there is almost no change in the positions of the two objects, but in (b), the change in position is quite large.

To address this issue, we introduce a new algorithm, named FocoTrack (FOCusing on Overlap Tracker). FocoTrack employs cascade matching, utilizing four distinct modules designed to address four different scenarios: one in which objects overlap with each other, and the three others where objects do not overlap. FocoTrack especially employs our new FOCO (FOCusing on Overlap) algorithm to match tracks and detections using velocity cues in the first scenario. It utilizes matching techniques by dividing a set of tracks into three groups to cover the other situations. To react to nonlinear motion exhibited in low-rate frames, FocoTrack leverages Interacting Multiple Model (IMM) [8], presenting an alternative to using only constant velocity Kalman filter [9] that is frequently used in other MOT algorithms. Our results show that FocoTrack outperforms existing MOT algorithms that do not use deep appearance descriptor, presenting more than two times better performance in terms of HOTA metric [10] on low frame rate datasets without need for deep appearance descriptor. Furthermore, efficiency of our model surpasses even the descriptor incorporated model.

In summary, our key contributions are three folds. First,

* Corresponding author

This work was in part supported by Korea Research Institute for Defense Technology planning and advancement (KRIT) grant funded by Korea government DAPA (No. KRIT-CT-22-006-002, Development of the situation/environment recognition technology for micro-swarm robot), by the CARAI grant funded by DAPA and ADD (No. UD190031RD), by IITP (No. 2022-0-00469, No. 20210005900012003, No.2021-0-02068), by NRF (No. 2021R1A2C2010585) and by BK21.

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, South Korea. {jaelee, alchemiclove, dechang}@kaist.ac.kr

based on the behavior of occlusions in low frame rate settings, we propose the FOCO algorithm. Second, we divide a set of tracks to three groups, leading to the development of a cascade matching technique which includes FOCO. This method also adopts IMM. Third, we evaluate both existing MOT algorithms and FocoTrack using a custom 2.5 fps low frame rate dataset and perform ablation analyses. To further prove enhanced capabilities of FocoTrack with the public dataset, we benchmark its performance against DanceTrack dataset [11].

II. RELATED WORKS

A. Tracking by Detection

With rapid development of object detection technology, tracking by detection method has become a mainstream of MOT. First, existing tracks are predicted one step forward by motion model. Second, an object detector detects detection boxes surrounding objects. Third, a cost between track and detection boxes is computed and matching algorithm assigns the detection boxes to the tracks with the Hungarian matching algorithm. Then, the matched tracks are corrected by matched detections. SORT [12] sets the baseline for tracking by detection, which employs a Kalman filter-based motion model. The predicted boxes are matched to detection boxes using Intersection over Union (IoU) metrics. DeepSORT [13] advances this technique by incorporating a deep appearance descriptor, which generates appearance features of objects. The process involves initial matching using appearance features and subsequent matching using SORT for remaining tracks. Employing a series of matching algorithms one after the other is termed cascade matching, and DeepSORT employs this approach to compensate for cases when the deep appearance descriptor does not succeed in making a match. ByteTrack [14] proposes a strategy to handle boxes with low confidence scores from object detectors. Detections are divided into two groups based on detection confidence score, which means the possibility that an object is within the detection boxes. Cascade matching is then applied, where high confidence score detections are matched to tracks first, followed by low confidence score ones. OC-SORT [15] introduces matching techniques to track objects exhibiting nonlinear movements without depending on appearance features. The primary criterion for matching depends on a weighted sum of IoU and velocity cues. Subsequently, the cost is computed between earlier matched detection boxes assigned to tracks and freshly detected detections.

B. Low Frame Rate MOT

To address low-frame rate environments, early tracking methodologies leverage appearance cues for associations at reduced frame rate [16][17]. This technique later evolved into the domain of deep appearance descriptor, where deep learning models are employed to utilize distinctive appearance features [18]. Interestingly, even when a frame rate is reduced in DeepSORT that uses a deep appearance descriptor [13], there is not a substantial decline in its performance [19]. Yet, the deep appearance descriptor can face challenges in scenario where objects have similar appearances [11] and utilization of GPU resources makes it less suitable for resource-limited robots. Our method proposed in this paper offers capability to effectively track objects at a lower frame rate without need to rely on any deep appearance descriptor.

C. Motion Models

A prevalent approach in tracking by detection algorithms is incorporation of motion model with Kalman filter [9]. Yet, an assumption of the Kalman filter is a linear relationship between prior and present states. Addressing this limitation, IMM [8] presents an alternative to Kalman filter. IMM uses several motion models depending on the situation, so it is more accurate than the Kalman filter in a non-linear situation [20]. This ability has led to the implementation of IMM in constructing motion models within the domain of 3D object tracking [21], [22]. Our empirical studies further demonstrate the capability of IMM to enhance performance of 2D object tracking at a low frame rate.

III. METHOD

A. Focusing on Overlap(FOCO)

Typically, tracking by detection algorithms uses IoU as a cost metric for matching. IoU measures the overlap between two bounding boxes. Accuracy of IoU is significantly influenced by the location and dimension of the boxes. Although there have been other methods developed to enhance IoU, they still heavily rely on the position and size of the boxes [23], [24]. When track paths overlap, using a cost function that emphasizes location and size can lead to an ID switch. These overlaps usually occur when two objects pass each other in opposite directions. In such occlusion scenarios, it can be more logical to consider displacement and direction of the objects for matching rather than location and size. Hence, we introduce a new matching method that focuses on displacement and direction of the objects for better matching under occlusion and name this method, *FOCO*. Figure 2 illustrates the flow chart with example pictures when FOCO activates.

Let $\mathcal{T}_t$ denote a set of boxes of tracks at time t. Our goal is to estimate $\mathcal{T}_{t+1}$ from $\mathcal{T}_t$ using FOCO. Let us initialize $\mathcal{T}_{t+1} = \emptyset$. First, given a track box $\alpha \in \mathcal{T}_t$, let us define a set $\bar{\mathcal{T}}_{t+1}$ of predicted boxes as

$$\bar{\mathcal{T}}_{t+1} = \{\bar{\alpha} \mid \bar{\alpha} = f(\alpha), \alpha \in \mathcal{T}_t\}. \tag{1}$$

where f is a function that predicts one step forward with IMM, which uses two Kalman filters one with a constant velocity model and the other with a constant acceleration model [25]. Let us define $\mathcal{O}_{t+1}$ as

$$\mathcal{O}_{t+1} = \{(\bar{\alpha}, \bar{\beta}) \mid \mathrm{IoU}(\bar{\alpha}, \bar{\beta}) > \rho_1, \;\; \bar{\alpha}, \bar{\beta} \in \bar{\mathcal{T}}_{t+1}\}.$$

where ρ_1 is a threshold.

Second, apply a deep learning object detection algorithm on the $t+1$ frame image to find detection boxes. Let $\mathcal{D}_{t+1}$ denote a set of the detection boxes at time $t+1$. Now, we

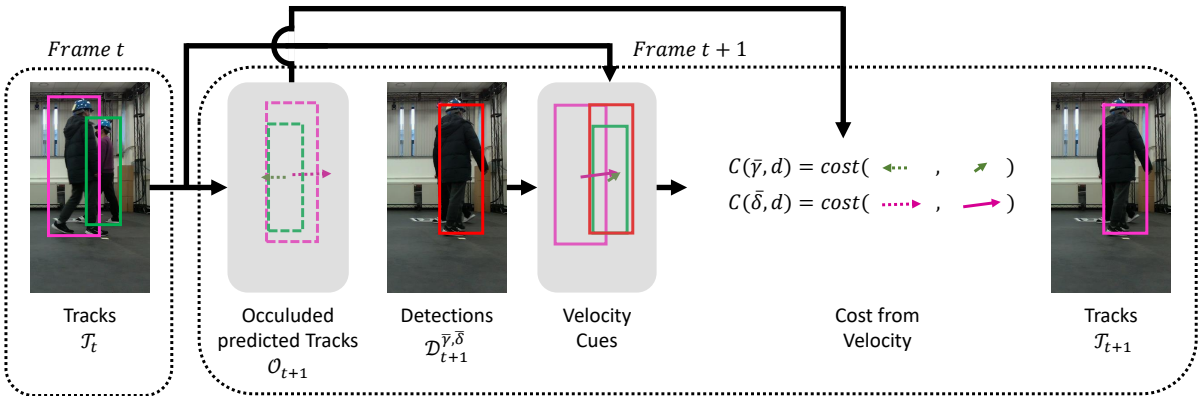

Fig. 2: A flowchart for FOCO: Solid pink and green boxes represent distinct track boxes, while dotted boxes indicate those predicted by IMM. Solid red boxes are for detection boxes. In instances where the predicted track boxes overlap, FOCO algorithm activates. This algorithm forms a cost function based on displacement and direction difference calculated from new detection boxes and track boxes.

want to match the detection boxes to elements of $\mathcal{O}_{t+1}$. Let us define a set $\mathcal{D}_{t+1}^{\bar{\alpha},\bar{\beta}}$ as

$$\mathcal{D}_{t+1}^{\bar{\alpha},\bar{\beta}} = \{d \mid \text{IoU}(\bar{\alpha}, d) > \rho_2 \text{ or } \text{IoU}(\bar{\beta}, d) > \rho_2,\ d \in \mathcal{D}_{t+1}\},$$

for $(\bar{\alpha}, \bar{\beta}) \in \mathcal{O}_{t+1}$, where ρ_2 is a threshold. Let us define $\mu(\text{box})$ as a function that gives the image coordinates of the center of the box. Choose an element in $\mathcal{O}_{t+1}$ with the highest IoU value and denote it as $(\bar{\gamma}, \bar{\delta})$, where $\bar{\gamma} = f(\gamma)$ and $\bar{\delta} = f(\delta)$ for some $\gamma, \delta \in \mathcal{T}_t$ by definition of $\mathcal{O}_{t+1}$. For $d \in \mathcal{D}_{t+1}^{\bar{\gamma},\bar{\delta}}$, compute the following values:

$$s_1^{\bar{\gamma}}(d) = ||\mu(d) - \mu(\gamma)||_2,\ \ s_1^{\bar{\delta}}(d) = ||\mu(d) - \mu(\delta)||_2,$$
$$\theta_1^{\bar{\gamma}}(d) = \frac{\mu(d) - \mu(\gamma)}{s_1^{\bar{\gamma}} + \epsilon},\ \ \theta_1^{\bar{\delta}}(d) = \frac{\mu(d) - \mu(\delta)}{s_1^{\bar{\delta}} + \epsilon},$$

where $||\cdot||_2$ is the Euclidean norm, ϵ is a small constant to avoid division by zero, and γ and δ are elements in $\mathcal{T}_t$ corresponding to $\bar{\gamma}$ and $\bar{\delta}$, respectively. In the above equations, $s_1^{\bar{\gamma}}$ and $s_1^{\bar{\delta}}$ signify current displacements of the boxes, and $\theta_1^{\bar{\gamma}}$ and $\theta_1^{\bar{\delta}}$ signify current directions of the boxes. Also, we define and calculate previous displacements $s_2^{\bar{\gamma}}$ and $s_2^{\bar{\delta}}$, and directions $\theta_2^{\bar{\gamma}}$ and $\theta_2^{\bar{\delta}}$ as follows:

$$s_2^{\bar{\gamma}} = ||\mu(\gamma) - \mu(\gamma_{-1})||_2,\ \ s_2^{\bar{\delta}} = ||\mu(\delta) - \mu(\delta_{-1})||_2,$$
$$\theta_2^{\bar{\gamma}} = \frac{\mu(\gamma) - \mu(\gamma_{-1})}{s_2^{\bar{\gamma}} + \epsilon},\ \ \theta_2^{\bar{\delta}} = \frac{\mu(\delta) - \mu(\delta_{-1})}{s_2^{\bar{\delta}} + \epsilon},$$

where γ_{-1} and δ_{-1} denote the values of γ and δ at the previous time step, respectively. Finally, define and compute the deviations in displacements and directions as below:

$$\Delta s^{\bar{\gamma}} = \mid s_1^{\bar{\gamma}}(d) - s_2^{\bar{\gamma}} \mid,\ \ \Delta s^{\bar{\delta}} = \mid s_1^{\bar{\delta}}(d) - s_2^{\bar{\delta}} \mid,$$
$$\Delta\theta^{\bar{\gamma}} = -\langle\theta_1^{\bar{\gamma}}(d), \theta_2^{\bar{\gamma}}\rangle,\ \ \Delta\theta^{\bar{\delta}} = -\langle\theta_1^{\bar{\delta}}(d), \theta_2^{\bar{\delta}}\rangle,$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product in 2 dimensions. We define cost functions as follows:

$$C^{\bar{\gamma}}(d) = \Delta\theta^{\bar{\gamma}} + \lambda\Delta s^{\bar{\gamma}},\quad C^{\bar{\delta}}(d) \quad = \Delta\theta^{\bar{\delta}} + \lambda\Delta s^{\bar{\delta}},$$

where λ is a threshold. Using the cost functions, we define cost vector of each track in $(\bar{\gamma}, \bar{\delta})$ as below:

$$C^{\bar{\gamma}} = (C^{\bar{\gamma}}(d_1), C^{\bar{\gamma}}(d_2), \cdots, C^{\bar{\gamma}}(d_i), \cdots, C^{\bar{\gamma}}(d_n)) \in \mathbb{R}^n,$$
$$C^{\bar{\delta}} = (C^{\bar{\delta}}(d_1), C^{\bar{\delta}}(d_2), \cdots, C^{\bar{\delta}}(d_i), \cdots, C^{\bar{\delta}}(d_n)) \in \mathbb{R}^n,$$

where $d_i \in \mathcal{D}_{t+1}^{\bar{\gamma},\bar{\delta}}$ and n is the number of elements of $\mathcal{D}_{t+1}^{\bar{\gamma},\bar{\delta}}$. Finally, the cost matrix is formulated as below:

$$C^{\bar{\gamma},\bar{\delta}} = [C^{\bar{\gamma}}, C^{\bar{\delta}}] \in \mathbb{R}^{n\times 2}.$$

The above cost matrix is used in the Hungarian algorithm for matching elements in $\{\bar{\gamma}, \bar{\delta}\}$ with the detection boxes in $\mathcal{D}_{t+1}^{\bar{\gamma},\bar{\delta}}$. The matched track boxes are updated by correction process of IMM with the matched detection boxes. These corrected matched track boxes are added to $\mathcal{T}_{t+1}$ and removed from $\bar{\mathcal{T}}_{t+1}$. Also, the matched detection boxes are removed from $\mathcal{D}_{t+1}$. In $\mathcal{O}_{t+1}$, elements containing the matched track boxes are removed. We repeatedly choose an element in $\mathcal{O}_{t+1}$ with the highest IoU and execute the matching process until $\mathcal{O}_{t+1}$ becomes empty. After the execution of FOCO matching algorithm, the remaining elements in $\bar{\mathcal{T}}_{t+1}$ and $\mathcal{D}_{t+1}$ undergo the next matching modules in Section III-B.

B. Cascade Matching

Our cascade matching system is composed of FOCO which matches occluded tracks between each other and three methods that match the remaining tracks. Figure 3 shows an execution order of the modules in the cascade matching and what situations should arise to activate the certain modules.

First, we use FOCO described in Section III-A to match occluded objects specifically. By using FOCO first, we match occluded objects correctly and remove them from $\bar{\mathcal{T}}_{t+1}$, so that the next matching algorithms have lower possibility of ID switch. A detailed experiment on order of FOCO among the matching algorithms is explained in Section IV-D.1. Given $\mathcal{D}_{t+1}$ with some detection boxes removed by FOCO, let us define the following sets:

$$\mathcal{D}_{t+1}^{H} = \{d \mid \text{score}(d) > \rho_3,\ d \in \mathcal{D}_{t+1}\},$$
$$\mathcal{D}_{t+1}^{L} = \{d \mid \text{score}(d) < \rho_3,\ d \in \mathcal{D}_{t+1}\},$$

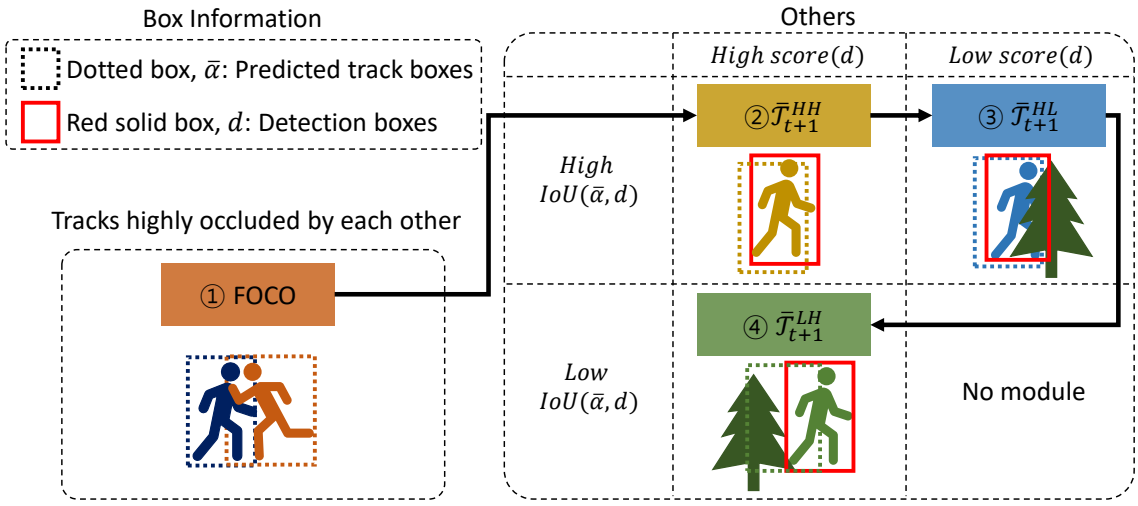

Fig. 3: Overview of our cascade matching. There are four modules that run in sequence to achieve appropriate matching based on track state. Solid red boxes signify detection boxes, while dotted boxes in various colors represent predicted track boxes for distinct objects.

where $\text{score}(d)$ gives the detection confidence score of $d \in \mathcal{D}_{t+1}$ from the deep learning object detection algorithm, and ρ_3 is a threshold. Also, given $\bar{\mathcal{T}}_{t+1}$ with some tracks removed by FOCO, define the following sets:

$$\bar{\mathcal{T}}_{t+1}^{HH} = \{\bar{\alpha} \mid \text{IoU}(\bar{\alpha}, d) > \rho_4,\ d \in \mathcal{D}_{t+1}^{H},\ \bar{\alpha} \in \bar{\mathcal{T}}_{t+1}\},$$
$$\bar{\mathcal{T}}_{t+1}^{HL} = \{\bar{\alpha} \mid \text{IoU}(\bar{\alpha}, d) > \rho_4,\ d \in \mathcal{D}_{t+1}^{L},\ \bar{\alpha} \in \bar{\mathcal{T}}_{t+1}\},$$
$$\bar{\mathcal{T}}_{t+1}^{LH} = \{\bar{\alpha} \mid \text{IoU}(\bar{\alpha}, d) < \rho_4,\ d \in \mathcal{D}_{t+1}^{H},\ \bar{\alpha} \in \bar{\mathcal{T}}_{t+1}\},$$

where ρ_4 is a threshold.

Second, we match detection boxes in $\mathcal{D}_{t+1}^{H}$ with tracks in $\bar{\mathcal{T}}_{t+1}^{HH}$. This approach is the most standard matching method in that it relies on trustworthy detections and positions of tracks predicted by the motion model. Third, we match detection boxes in $\mathcal{D}_{t+1}^{L}$ with tracks in $\bar{\mathcal{T}}_{t+1}^{HL}$. The methodology complements unmatched tracks in $\bar{\mathcal{T}}_{t+1}$ getting matched when the previous module does not match low detection confidence score boxes in $\mathcal{D}_{t+1}^{L}$ which can occur due to obstacle overlap [14].

Finally, we match detection boxes in $\mathcal{D}_{t+1}^{H}$ and tracks in $\bar{\mathcal{T}}_{t+1}^{LH}$. When tracks are obscured by obstacles over a duration, there are no detection boxes available for matching. As a result, only the prediction of the motion model is continually conducted without correction step and results in low $\text{IoU}(\bar{\alpha}, d)$ after detection comes out again. This module plays a crucial role in low frame rate situations. The matching between $\mathcal{D}_{t+1}^{L}$ with tracks with low value of $\text{IoU}(\bar{\alpha}, d)$ is not used because low $\text{IoU}(\bar{\alpha}, d)$ means the motion model is not trustworthy and a low detection confidence score also means the possibility of the object in the detection box is low. In the above three matching algorithms, we use the Hungarian algorithm with the cost, $\text{IoU}(\bar{\alpha}, d)$.

IV. EXPERIMENTS

A. Experimental setup

Datasets. To evaluate our method, we made a new dataset consisting of 290 images, capturing 3 individuals wandering arbitrarily at a frame rate of 2.5 fps, which we termed as LF25. This dataset is designed to simulate occlusion scenarios typically experienced by autonomous robots within a 10-meter proximity to their vision systems. Alongside our dataset, we also employed a public dataset called DanceTrack [11], captured at 20 fps. DanceTrack is distinct due to the presence of individuals with closely similar appearances and their unpredictable movements. This dataset helps us benchmark our technique against widely accepted public dataset.

Metrics. We adopt HOTA [10] as our primary evaluation metric, considering its balance between the accuracy of detection and tracking performance. Further, to spotlight tracking efficiency, we include AssA and IDF1 metrics. In addition, MOTA and DetA, metrics highly linked to detection performance, are employed.

Implementation details. To maintain consistency and fairness in our assessment, we utilize YOLOX detector [26], which is publicly accessible. For LF25, we choose YOLOX-Tiny, a compact detector optimized for real-time deployment on constrained resources. For DanceTrack, we employ YOLOX-s, a model which has better performance than YOLOX-Tiny and the same model weight previously used for evaluations in other studies [15], [14]. Given a small quantity of LF25, YOLOX-Tiny was solely trained on the DanceTrack training data for 50 epochs without relying on additional datasets. In the tests with LF25, we set the FOCO related thresholds ρ_1, ρ_2 to 0.2, 0.3 and cascade related thresholds ρ_3, ρ_4 to 0.7, 0.5 respectively. The cost threshold λ is set to 0.02 and the small scale constant ϵ is set to 10^{-6}. In DanceTrack, ρ_1 is set to 0.85.

B. Evaluation

LF25. Table I illustrates the performance of our algorithm on the low frame rate dataset LF25 against existing methodologies. When assessed by HOTA criterion, our approach notably outperforms, achieving more than two times better

TABLE I: Results on LF25 dataset(left) and DanceTrack test dataset(right).

Tracker	LF25					DanceTrack Test				
	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
SORT [12]	21.4	61.0	7.5	71.0	17.6	47.9	72.0	31.2	89.5	52.5
DeepSORT [13]	63.6	73.1	55.4	92.4	83.7	45.6	71.0	29.7	87.8	47.9
ByteTrack [14]	35.4	72.9	17.3	91.5	35.6	47.3	71.6	31.4	89.5	52.5
OC-SORT [15]	37.1	73.0	18.9	80.1	40.5	55.1	80.4	40.4	**92.2**	54.9
Ours	**82.0**	**82.1**	**81.9**	**94.6**	**97.2**	**57.1**	**80.8**	**40.5**	91.4	**58.2**

score of recent studies that do not use deep appearance descriptor like OC-SORT and ByteTrack. It surpasses the performance of DeepSORT, which relies on the descriptor, by a margin of 18.4 HOTA. With an improvement of 9.0 and 26.5 in DetA and AssA metrics respectively against DeepSORT, our method enhances tracking accuracy. Such results prove that our algorithm is capable of appropriate tracking at low frame rates without using additional GPU resources. The comparisons of tracking with LF25 are visualized in Fig. 4.

DanceTrack. Table I shows the performance of our algorithm when evaluated on DanceTrack dataset. Compared to OC-SORT, our algorithm improves by 2.0 score on HOTA criterion. While there is a negligible enhancement in DetA and AssA, IDF1 metric which evaluates accurate ID assignment and robustness against ID-induced false positives [27], sees a significant improvement by 3.3. This indicates that when our algorithm is applied on scenarios where the use of deep appearance descriptor is restricted due to similar appearances, there is an improvement in ID assignment performance.

C. Ablation Study

We perform ablation studies to assess the impact of individual modules on overall efficacy and IMM. Table II presents contributions of different modules, while Table III details influence of IMM.

TABLE II: Ablation studies of four modules in LF25. FC, HH, HL and LH represent FOCO, second, third and fourth module respectively.

FC	HH	HL	LH	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
✓	✓	✓	✓	82.0	82.1	81.9	94.6	97.2
	✓	✓	✓	43.5	81.9	23.1	93.7	48.1
	✓		✓	41.1	81.3	20.8	92.4	41.9
✓	✓	✓		23.7	73.9	7.7	75.9	16.5

TABLE III: Experiments involving IMM. Within this context, CV, CA refers to Constant Velocity kalman filter model, and Constant Acceleration one respectively.

motion model	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
IMM with CV, CA	82.0	82.1	81.9	94.6	97.2
Only CV	77.8	81.8	73.9	94.1	91.4
Only CA	71.6	81.5	63.0	93.8	84.5

1) Effect of modules: As seen in Table II, omitting FOCO module leads to a decline of 38.5 in HOTA score. While there is not a notable decline in DetA or MOTA, there is a huge decrease in AssA and IDF1 by 58.8 and 49.1 respectively. This indicates that FOCO plays an important role in tracking performance at low-frame rate. Removing the fourth module shows the largest drop in the low frame rate dataset. Even with FOCO module in place, without accounting for the fourth module, results are inferior to both OC-SORT and ByteTrack, yielding a HOTA score of 23.7. This shows that the fourth module is the most important module for low frames and is required to use FOCO. The third module also enhances performance in settings with fewer frames. Removing this module from a model that does not incorporate FOCO leads to 2.4 drop in the HOTA score.

2) Effect of IMM: A comparison of the IMM, which employs two types of Kalman filter, with single types of Kalman filter, is given in Table III. According to the table, the IMM outperforms only constant-velocity model by 4.2 and constant-acceleration model by 10.4 in terms of HOTA. While existing algorithms, equipped with only one Kalman filter, are proficient at capturing these non-linear patterns at high frame rate, as noted by [15], the performance drops in low frame rate environments. The results shown in Table III indicate that IMM approach is more adept at low frame rate.

D. Furthur Study

1) Order of cascade matching: As illustrated in Table IV, we test various sequences of modules and determine that positioning FOCO at the beginning yields the best results. A recent cascade matching investigation emphasizes benefits of first highly reliable matching method followed by the less one [28] [14]. Given that FOCO is integrated due to the unreliability of IoU-based matching stemming from occlusions, one might presume that it would follow a similar sequence as in the prior research. However, when FOCO is not given priority, other matching algorithms tend to create inaccurate matches, which accounts for the enhanced performance observed when FOCO is positioned at the forefront.

TABLE IV: Experiments on order of cascade matching. FC, HH, HL and LH represent FOCO, second, third and fourth module respectively.

matching order	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
FC ⇒ HH ⇒ HL ⇒ LH	82.0	82.1	81.9	94.6	97.2
HH ⇒ FC ⇒ HL ⇒ LH	48.1	81.9	28.3	93.7	53.8
HH ⇒ HL ⇒ FC ⇒ LH	43.4	81.9	23.1	93.7	48.1
HH ⇒ HL ⇒ LH ⇒ FC	43.4	81.9	23.1	93.7	48.1

2) Threshold: The efficacy of the FOCO algorithm is intrinsically linked to its activation threshold ρ_1 and Table V shows the results of the threshold. When the value of ρ_1

Fig. 4: Example results of LF25 dataset. The flow of time goes from left to right. Whereas current algorithms struggle with lost tracks or ID switching during occlusions, FocoTrack effectively assigns IDs even when three individuals are closely clustered in the occlusion scenario.

is set below 0.2, the FOCO mechanism becomes active even when tracks are not occluded sufficiently, which can degrade performance. Conversely, if the threshold is set too high, the FOCO algorithm might fail to engage even when occlusions are present, leading to suboptimal outcomes. We explore more extensively to understand the effects of threshold ρ_4. As Table VI indicates, in cascade matching, a reduction in this threshold tends to adversely impact performance, blurring distinction between modules. Thus, to utilize the full potential of the cascade matching, it is preferable to maintain the threshold higher, which is not recommended when using only the second module. Table VII shows the impact of the cost related threshold λ. If λ is bigger than 0.02 which means displacement part has a quite big influence, performance goes down. However, as can be seen in the last row of the table, even a value of λ as small as 0.001 has a bad impact on performance. This means that it is important to find the right balance between displacements and directions.

V. CONCLUSION

We present FocoTrack which addresses four situations with cascade matching. The FOCO algorithm, which is the first module in our cascade matching system, primarily handles occlusions and enables efficient matching at low frame rate. The remaining situations, we separate tracks into three groups and matching system corresponding to each specific situation operate sequentially. Our tests on a dataset with 2.5 fps reveal that FocoTrack outperforms existing methods. Additionally, its capabilities are further confirmed on DanceTrack. The validity of our method is demonstrated through ablation studies and further studies. These results prove our method can be applied to mobile robots with limited resources.

TABLE V: Effect of FOCO activation threshold ρ_1. Num means the number of FOCO activated in LF25

ρ_1	Num	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
ρ_1 =0.1	85	43.1	80.7	23.0	91.9	46.3
ρ_1 =0.2	62	82.0	82.1	81.9	94.6	97.2
ρ_1 =0.3	51	70.2	81.8	60.2	94.0	82.9
ρ_1 =0.4	30	48.8	81.3	29.3	92.7	56.7

TABLE VI: Effect of the cascaded related threshold ρ_4

ρ_4	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
ρ_4 =0.5	82.0	82.1	81.9	94.6	97.2
ρ_4 =0.4	72.0	81.6	63.6	93.8	85.9
ρ_4 =0.3	65.9	82.5	52.7	94.3	79.3

TABLE VII: Effect of the cost related threshold λ

λ	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
λ =1	39.4	80.9	19.2	89.5	42.1
λ =0.1	42.9	79.8	23.1	91.6	47.8
λ =0.02	82.0	82.1	81.9	94.6	97.2
λ =0.001	48.1	81.0	28.5	93.3	55.4

ACKNOWLEDGEMENT

The authors would like to thank Hee-Deok Jang, Dong Hyun Park, and Xiaowei Xing for their valuable comments.

REFERENCES

[1] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *IEEE International Conference on Robotics and Automation*, 2021.

[2] C. Luo, X. Yang, and A. Yuille, "Exploring simple 3d multi-object tracking for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[3] J. Alikhanov and H. Kim, "Online action detection in surveillance scenarios: A comprehensive review and comparative study of state-of-the-art multi-object tracking methods," *IEEE Access*, 2023.

[4] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *IEEE International Conference on Robotics and Automation*, 2018.

[5] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *IEEE International Conference on Robotics and Automation*, 2023.

[6] G. Jocher, L. Changyu, A. Hogan, L. Y. 于力, changyu98, P. Rai, and T. Sullivan, "ultralytics/yolov5: Initial release," 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3908560

[7] X. Zhao, Z. Huang, L. Ye, and Y. Lv, "Real-time detection method for submarine pipeline leakage based on deep learning and jetson nano," in *OCEANS, Hampton Roads*, 2022.

[8] H. A. P. Blom, "An efficient filter for abruptly changing systems," in *The 23rd IEEE Conference on Decision and Control*, 1984.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

[10] 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*, 2021.

[11] 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.

[12] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *IEEE International Conference on Image Processing*, 2016.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *IEEE International Conference on Image Processing*, 2017.

[14] 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*, 2022.

[15] J. Cao, X. Weng, R. Khirodkar, J. Pang, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," *arXiv preprint arXiv:2203.14360*, 2022.

[16] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[17] X. Zhang, W. Hu, N. Xie, H. Bao, and S. Maybank, "A robust tracking system for low frame rate video," *International Journal of Computer Vision*, 2015.

[18] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[19] T. Zhou, W. Luo, Z. Shi, J. Chen, and Q. Ye, "Apptracker: Improving tracking multiple objects in low-frame-rate videos," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[20] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, 1998.

[21] P. Liu and Z. Duan, "An imm-enabled adaptive 3d multi-object tracker for autonomous driving," in *IEEE 24th International Conference on Information Fusion*, 2021.

[22] S. Gautam, G. P. Meyer, C. Vallespi-Gonzalez, and B. C. Becker, "Sdvtracker: Real-time multi-sensor association and tracking for self-driving vehicles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[23] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[25] A. F. Genovese, "The interacting multiple model algorithm for accurate state estimation of maneuvering targets," *Johns Hopkins APL technical digest*, 2001.

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

[27] R. Henschel, T. von Marcard, and B. Rosenhahn, "Simultaneous identification and tracking of multiple people using video and imus," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[28] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.