2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Multi-target Tracking with Occlusion Resistance for Mobile Robots in Dynamic Environments*

Zhongyan Liu, Biao Lu*, Xinghai Xing, Dun Mao, Yongchun Fang

Abstract— **In the context of tracking multiple targets on a novel mobile robot, it is essential to obtain the three-dimensional coordinates of specified targets based on tracking boxes. Most existing multi-target tracking algorithms neglect the inherent constraints of the novel mobile robot, such as insufficient computational power, dynamically complex working environments, and irregularly occluded targets. To address these limitations, we propose a robust tracking algorithm with occlusion resistance (hereinafter referred to as ROTrack). ROTrack compensates for the predictions of Kalman filter (KF) by incorporating Inertial Measurement Unit (IMU) information, enabling the tracker to achieve more accurate tracking in dynamic environments. Additionally, MobileSAM is employed to handle occlusion issues and obtain the correct three-dimensional coordinates of the targets. At the same time, a depth-triggered segmentation strategy is proposed to reduce computational resource consumption. The effect of ROTrack is demonstrated through alignment between IMU signals and Camera Motion Compensation (CMC) data in BoT-SORT. Real-world tracking tests validate the robustness and real-time capability of ROTrack.**

I. INTRODUCTION

In modern society, novel mobile robots, such as wheel-legged robots, quadruped robots and humanoid robots, play a key role in search and rescue, logistics and transportation. The role of multi-target tracking in these tasks is to identify and track targets in real time, so as to improve the robot's perception, decision making and action capabilities.

Up to now, numerous multi-target tracking algorithms have been developed, making outstanding contributions to addressing challenges such as target appearance variations and target diversity. However, these algorithms are typically designed for relatively static environments. Moreover, most target tracking algorithms are deployed on regular servers with sufficient computational resource. Unfortunately, the environments and challenges that mobile robots encounter are considerably more complex and diverse.

- **Complex dynamic environments.** Previous researches have predominantly focused on static environments, whereas the working environments for mobile robots involves dynamic scenes. The dynamic scenes come from two primary sources: the changes of environment and the camera movements. For example, in search and rescue or other scenarios, it is almost inevitable for the robots to encounter violent motions due to rough terrain conditions, complex tasks and other factors, which further lead to camera movements, including camera shake, translation, tilt and so on. These factors often cause motion blur or changes in size, potentially resulting in the failure of robot tracking.
- **Occlusion issue.** It is essential for mobile robots to obtain the three-dimensional coordinates of the target in real time. Unfortunately, in complex environments, targets may be irregularly occluded by other objects. Conventional multi-target tracking algorithms are devoted to getting accurate tracking boxes even in the presence of occlusion. However, when it comes to situations where the target is occluded, even if the algorithm can obtain accurate tracking boxes, it may get the depth of the obstacle rather than the target. Therefore, the calculated three-dimensional coordinates of the target are also incorrect.
- **Computational resource limitation.** Tracking multiple targets in large-scale environments demands substantial computational and planning capabilities. With the improvement in graphics card performance, the majority of multi-target tracking algorithms do not need to worry about the limitation of computational resource. However, mobile robots are often affected by factors such as power supply, space constraints, and payload limitations, which result in limited computational resource of onboard computers.

Aiming at the limitations of target tracking for mobile robots, we propose a novel multi-target tracker named ROTrack and integrate it into BoT-SORT [1]. ROTrack is designed for deployment on mobile robots. Experiments indicate that ROTrack maintains impressive stability and robustness. Its real-time capability fully meets the planning and control frequency requirements of the robot. The primary contributions of our work can be summarized as follows:

- To address the dynamic complexities of the robot working scenes, the IMU information is used during the tracking process to estimate the motion of mobile robot, and correct the predictions of KF appropriately. We denote this as Robot Motion Compensation (RMC). By improving the prediction results of KF, the tracking performance of the tracker can be enhanced effectively in dynamic environment. Moreover, the IMU information

*This work is supported by the National Natural Science Foundation of China under Grant 62203235, the Key Projects of the Joint Fund of the National Natural Science Foundation of China under Grant U22A2050, and the Joint Fund of Guangdong Basic and Applied Basic Research Fund under Grant 2022A1515110046

The authors are with the Institute of Robotics and Automatic Information System, College of Artificial Intelligence, Nankai University, Tianjin 300353, China, and also with the Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen 518083, China (E-mail of corresponding author* Biao Lu: lubiao@mail.nankai.edu.cn)

can be obtained directly without cumbersome computation, which contributes to the real-time efficiency of ROTrack.

- We propose a tracking solution that integrates MobileSAM [2] to resist occlusion. When the tracking point is occluded, MobileSAM is used to segment the unoccluded area of the target within the tracking box. Then ROTrack obtains the actual three-dimensional coordinates and updates the tracking point. Due to the robust performance of MobileSAM, tracking targets are not limited to humans. Hence, ROTrack can be conveniently extended to other tasks.
- Because of the great resource consumption, it is impractical for mobile robots to segment every frame. Therefore, a depth-triggered segmentation strategy is proposed. This method is based on depth perception and performs segmentation only when the target is occluded. In this way, the resource consumption and the computational burden of the robot are reduced.

II. Related Work

Dynamic target tracking is a focal research point in the field of mobile robots [3]. For instance, Fast-Tracker 2.0 [4] employs OpenPose [5] as a human detection system. Cao et al. [6] presents TCTrack for drone tracking. In addition, there are some works [7]–[10] have alseo studied this field. Multi-target tracking generally enables robots to perform more complex tasks than single-target tracking.

At present, the mainstream of multi-target tracking algorithms is detection-based tracking. CenterNet [11] is widely recognized for its simplicity and efficiency, making it a popular choice in various detection approaches [12], [13]. Similarly, many methods [14], [15] favor the YOLO series [16], [17] detectors for their excellent balance of accuracy and speed.

Most recent tracking-by-detection algorithms rely on motion models. The widely-known Kalman filter is employed to model object motion and predict tracking boxes in new frames. The Intersection over Union (IoU) metric is then used to compute the similarity between the detection boxes and the predicted boxes. While some researches use deep-appearance cues to distinguish and re-identify (ReID) objects, dynamic scenes may contain cases of severe occlusion or motion blur, in which ReID feature might not be reliable [18]. Furthermore, ReID causes significant resource consumption, which is not friendly to novel robots performing various tasks.

In recent years, joint trackers [19], [20] have been applied in various methods due to their low computational cost and satisfactory performance. In addition to tracking targets, mobile robots can also perform other tasks such as detection and obstacle avoidance. Tracking-by-detection trackers often exhibit better adaptability than joint trackers in these complex tasks. Their modular design allows for convenient migration to different tasks. If the detection algorithm performs poorly in specific scenarios, it can be easily replaced with a more suitable detector without redesigning the entire system. Therefore, these methods are more conducive to application on mobile robot platforms.

In complex dynamic environments, the motion of the robot can significantly influence and induce movement of the camera, which may lead to nonlinear motion of the targets in image coordinate and incorrect prediction of KF. Many researchers [21], [22] have employed CMC by aligning frames through image registration using Enhanced Correlation Coefficient (ECC) maximization [23] or matching features such as ORB [24]. After extracting image feature points, the affine matrix is solved by RANSAC [25] to modify KF. However, the affine matrix may fail to acquire if the image lacks enough feature points. Moreover, the image registration also results in additional resource consumption, leading to increased computational demands and potential performance degradation, thereby impacting the overall efficiency of the system. As an alternative, the Kalman filter is modified by the IMU information, which can be easily obtained from the robot to carry out the robot motion compensation.

The Segment Anything Model (SAM) [26] is a large computer vision model designed to address image segmentation tasks, particularly achieving precise instance segmentation in various application fields. SAM is used in this paper to address occlusion issues. Considering the limitation in computational power, we use MobileSAM instead. It is a lightweight version of SAM, being over 60 times smaller, yet performs on par with the original SAM [2]. Due to its superior performance and higher versatility, it is more suitable for mobile robots.

III. Method

In this section, we introduce the three main modifications and enhancements of ROTrack for multi-target tracking-based tracking-by-detection methods. These improvements have been integrated into the well-known BoT-SORT. We drop the ReID module and only use IoU scores as matching criteria. The flow chart of ROTrack is shown in Fig. 1.

A. Robot Motion Compensation

In a dynamic scenario, the position of the bounding box in the image plane may shift dramatically, potentially resulting in increased ID switches or false positives. Based on this consideration, IMU information is used to compensate for robot motion. The RealSense D435i camera along with its integrated IMU is utilized to ensure that the relative position of the IMU and the camera remains constant, even when the robot is in motion. Before RMC, extrinsics are used to unify the coordinate systems of the IMU and camera. After that, the compensation matrix I_{k-1}^{k} is obtained from the IMU signals. The complete expression of I_{k-1}^{k} is presented as follows:

$$I_{k-1}^{k} = [IR_{2x2} \,|\, IT_{2x1}] = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & t_x \\ \sin(\theta_z) & \cos(\theta_z) & t_y \end{bmatrix} \quad (1)$$

where IR_{2x2} is the matrix of the rotation part, IT_{2x1} is the matrix of the translation part. θ_z is the rotation angle about the z axis (perpendicular to the image direction), obtained

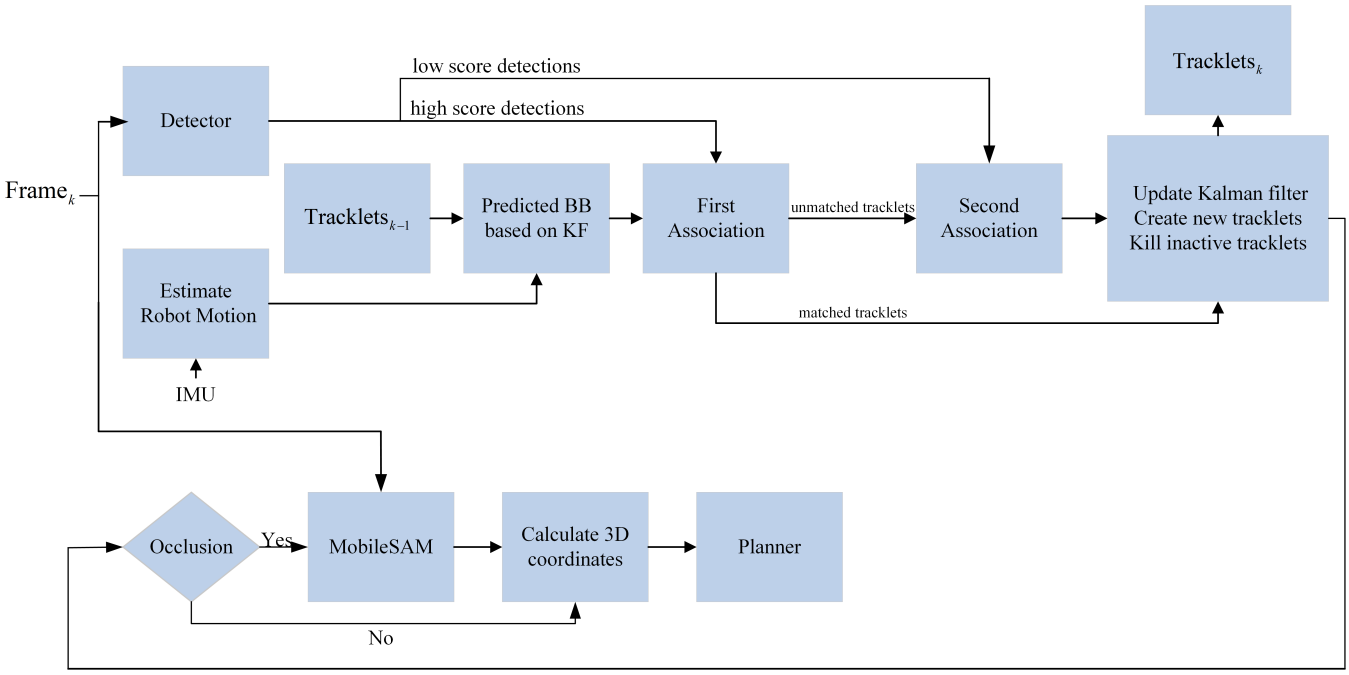


Fig. 1. Overview of ROTrack pipeline.

by integrating the rotation speed measured by the gyroscope. t_x, t_y are the amount of translation of pixels in the frames.

For the translation matrix, it is difficult to accurately measure them directly using the IMU. This is because the accelerometer data must be integrated twice to obtain the translation component, and the precision is insufficient due to the measurement noise. Fortunately, the camera movement caused by the robot comes mostly from rotation. Specifically, the targets are usually far enough away from the lens, and the translations attenuate quickly with increasing depth. As a result, translation of the camera usually causes negligible changes in the image [27]. On the other hand, the rotation motion of the camera leads to a relatively obvious translation of pixels in the two-dimensional image. Since the frame rate is fast and the rotation between consecutive frames is small, the translation in the image can be approximately computed from the rotation as follows:

$$t_x = f_x \tan(\theta_x) \tag{2}$$

$$t_y = f_y \tan(\theta_y) \tag{3}$$

where f_x and f_y are the focal length in the camera's intrinsic parameters. θ_x and θ_y are the rotation angles around the x axis and y axis respectively, which are also obtained by integrating the rotation speed measured by the gyroscope. In the end, the Savitzky-Golay filter is used to refine and smooth the rotation and translation of the output.

The bounding box then needs to be transformed from the coordinate system of frame $k-1$ to the coordinates of the next frame k. Following the work of Nir et al. [1], the robot motion compensation step can be performed by the following equations:

$$\hat{x}'_{k|k-1} = \widetilde{IR}^{k}_{k-1}\hat{x}_{k|k-1} + \widetilde{IT}^{k}_{k-1} \tag{4}$$

$$P'_{k|k-1} = \widetilde{IR}^{k}_{k-1}P_{k|k-1}\widetilde{IR}^{k^\top}_{k-1} \tag{5}$$

where $\hat{x}_{k|k-1}$ and $\hat{x}'_{k|k-1}$ are the predicted state vectors of KF at time k before and after RMC, respectively. $P_{k|k-1}$ and $P'_{k|k-1}$ are the prediction covariance matrix of KF before and after RMC, respectively. Finally, $\hat{x}'_{k|k-1}$ and $P'_{k|k-1}$ are used to update the KF as follows:

$$K_k = P'_{k|k-1}H_k^\top \left(H_k P'_{k|k-1}H_k^\top + R_k\right)^{-1} \tag{6}$$

$$\hat{x}_{k|k} = \hat{x}'_{k|k-1} + K_k\left(z_k - H_k\hat{x}'_{k|k-1}\right) \tag{7}$$

$$P_{k|k} = (I - K_k H_k)\, P'_{k|k-1} \tag{8}$$

where K_k is the Kalman gain, R_k is the measurement noise covariance, H_k is the observation matrix. $\hat{x}_{k|k}$, $P_{k|k}$ are the posterior state estimation and covariance estimation, respectively. At this point, robot motion compensation is completed.

B. Anti-occlusion Method Combined with MobileSAM

1) Get three-dimensional coordinates: After obtaining the target's tracking box through ROTrack, the three-dimensional coordinates of the center point within the tracking box need to be calculated and taken as the tracking point. Firstly, the RealSense D435i depth camera is used to obtain the depth frame aligned with the color frame and its intrinsics. After that, the depth of the tracking point in depth-frame is determined. In the end, the 3D coordinates of the tracking point in the camera coordinate system are computed according to the pixel coordinates, depth and intrinsics of the depth image.

When multiple targets present, ROTrack allows obtaining coordinates for all targets or choosing the one of interest for tracking.

In the practical applications, the tracking point may be occluded, thus the obtained depth may not be the actual depth of the target. Camera coordinates calculated from the wrong depth can also be biased. Due to the complexity of the environment, the occlusion area is random. Therefore, it is necessary to segment the unoccluded part of the target to accurately obtain the depth of tracking point.

2) Use MobileSAM to segment the target: MobileSAM works based on deep learning and convolutional neural networks. The image segmentation task is realized by learning the features and patterns of different objects in the image. MobileSAM mainly includes three components. The image encoder based on Vision Transformer (ViT) [28] is used to generate image embedding [29]. Then there is the prompt encoder, which is used to prompt the area of the model that needs to be segmented. Specifically, the points and boxes are considered as prompts which are represented using positional encodings. Following that, there is the mask decoder, which can effectively map image embeddings, prompt embeddings, and output token to a mask. This section is designed with reference to the Transformer decoder [30] section.

The process of segmenting the unoccluded part of the target is shown in Fig. 2. When the target is occluded, a certain frame to be segmented is input into the image encoder, and the obtained tracking box is input to the prompt encoder as a prompt. After that, the mask is segmented by MobileSAM, namely, the unoccluded part of the target. Finally, the depth of the mask is obtained, and the three-dimensional coordinates of the target are calculated. Note that MobileSAM allows prompt to be culled as a background point. In other words, the part outside the prompt can be split. In the case that the obstacle is similar to the target, obtaining a more accurate mask can be achieved by utilizing the point within the occluded area as the background prompt. Please refer to section IV for specific experimental results.

C. Depth-triggered segmentation method based on depth perception

It is not possible to segment every frame on a mobile robot, because even if MobileSAM is mobile friendly, it still takes up a considerable amount of resources. In the tracking process, the tracking point is not occluded most of the time, hence there is no need to segment each frame. Therefore, we rely on depth information to judge the occlusion state, and segmentation is triggered only when the tracking point is occluded. Since the camera frame rate is considerably fast relative to the moving speed of the target, the change in depth information of the target is minimal from frame $k-1$ to frame k if no occlusion occurs. This makes it possible to determine whether occlusion has occurred based on depth.

The flow chart of the whole triggering process is shown in Fig. 3. At the beginning, ROTrack uses the center point of the tracking box as the tracking point. The depth of the tracking point in the latest frame is recorded at the same time. When

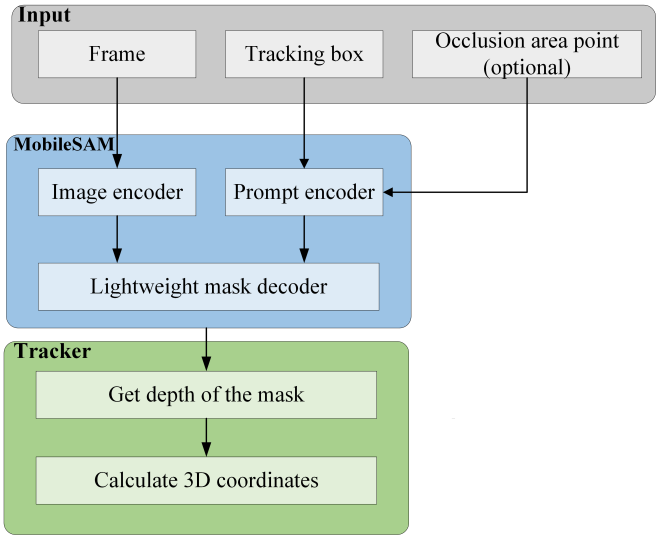


Fig. 2. Segmentation flow chart.

the difference between the current depth and the previous depth exceeds the threshold at a certain time, it indicates that the depth has a sudden change. This can be caused by occlusion or the disappearance of the target. There is also a case that due to the impact of light reflection, insufficient texture, etc., the camera may fail to obtain the depth of the tracking point. Under such circumstances, the obtained depth value is 0. In these cases, MobileSAM is triggered to segment and get the target mask. Then the unoccluded pixel is selected as the new tracking point. At the same time, the position of the point relative to the center point is recorded, so that the tracking point is found directly based on this relative position in the next frame, without the need to segment again. In other words, the image only needs to be segmented once when the depth changes suddenly.

In this process, ROTrack continues to record the depth of the center point of the tracking box. When the depth of the center point is consistent with the tracking point, it indicates that the occlusion disappears. At this time, the center point is reused as the tracking point. This step can better improve the robustness and execution speed of ROTrack.

IV. EXPERIMENTS

All experiments are implemented using PyTorch. Metrics are calculated on a desktop with 12th Gen Intel(R) Core(TM) i7-12700H 2.30GHz and NVIDIA GeForce RTX 3070 GPU. In the real world, our test images are taken on a wheel-legged robot as shown in Fig. 4. It is equipped with a RealSense D435i camera and a Jetson Orin that acts as an image processing unit. We use the publicly available YOLOX [31] detector. The image segmentation uses the model publicly available in [2]. The threshold for triggering segmentation is set to 0.1 m. Other parameters such as the detection score threshold are the same as [1].

Subsequently, we will demonstrate the effect of ROTrack through RMC testing and occlusion resistance testing. Specifically, the RMC testing illustrates the ability to

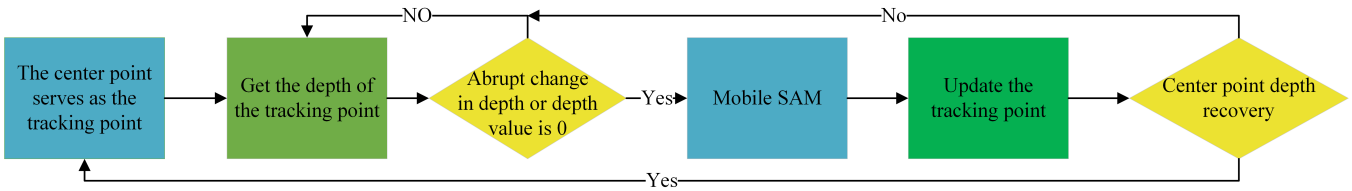


Fig. 3. Depth-triggered segmentation strategy.

enhance robustness of tracking in dynamic environments without additional resources consumption. The occlusion resistance testing demonstrates the capability to accurately obtain the three-dimensional coordinates of occluded targets.

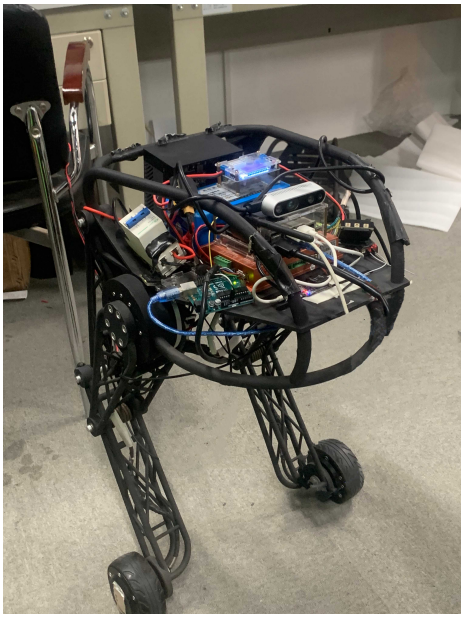


Fig. 4. The standing posture of the wheel-legged robot.

A. *Performance evaluation of robot motion compensation*

It is difficult to test the performance of ROTrack on a fixed, generic target tracking datasets, because the IMU signals need to be obtained in real time. We will illustrate the effect of robot motion compensation in the following ways. Firstly, the compensation matrix is obtained by CMC based on image matching. After that, following the settings in ByteTrack [18], the scores are tested on MOT17 and MOT20 datasets to demonstrate the effect after compensating for the KF. Finally, the rotation and translation signals obtained by IMU are aligned with the CMC data to show that the robot motion compensation through IMU can achieve similar effects to the camera motion compensation.

Since ROTrack needs to be deployed on a wheel-legged robot, the utilized detector is a lightweight YOLOX-Tiny trained by [18] for MOT17, MOT20. All experiments use the same tracking parameters. The widely accepted CLEAR metrics [32], including Multiple-Object Tracking Accuracy (MOTA), False Positive (FP), False Negative (FN), ID Switch (IDSW), etc., IDF1 [33] and Higher-Order Tracking Accuracy (HOTA) [34] are used to evaluate different aspects of the tracking performance. We compare the results on BoT-SORT and ByteTrack, which are ranked first and second in MOTChallenge, respectively. Moreover, we evaluate the speed (FPS) of these trackers in Hz on Jetson Orin. The experimental results are shown in Tables I to III.

TABLE I. Tests on the MOT17 dataset

Tracker	MOTA	IDF1	FP	FN	IDs	HOTA
ByteTrack	77.3	71.5	4912	20440	513	61.8
BoT-SORT(without CMC)	77.4	73.1	5250	19686	732	63.2
BoT-SORT(with CMC)	78.2	75.4	5270	19125	479	64.5

TABLE II. Tests on the MOT20 dataset

Tracker	MOTA	IDF1	FP	FN	IDs	HOTA
ByteTrack	68.5	65.0	138459	295916	2731	50.8
BoT-SORT(without CMC)	68.7	65.1	139055	293297	2968	50.9
BoT-SORT(with CMC)	68.7	65.0	139165	293044	2990	50.8

TABLE III. Frame rate tests on Jetson Orin

Tracker	FPS
ByteTrack	23.7
BoT-SORT(without CMC)	25.2
BoT-SORT(with CMC)	17.6
Ours	23.3

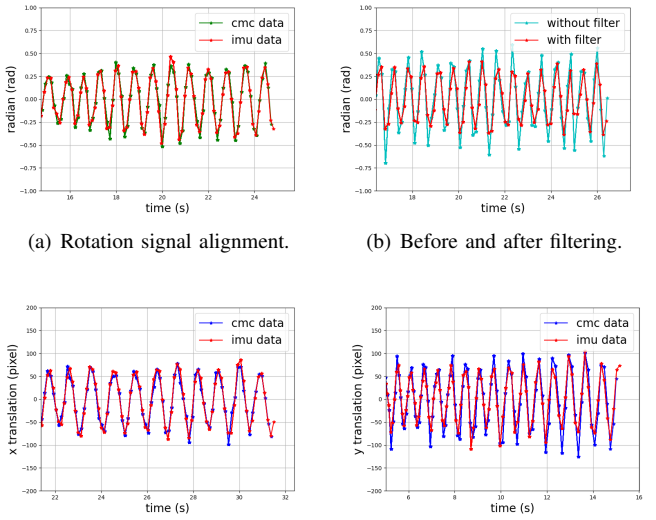
MOT17. The tracker with CMC achieves higher scores in metrics such as MOTA, IDF1, HOTA, etc., compared with the tracker without CMC. In different scenarios, high MOTA and IDF1 indicate that CMC can significantly improve the tracking robustness and accuracy.

MOT20. Compared to MOT17, MOT20 has more congestion and occlusion. The effect is basically the same as the original result after adding CMC. This is due to the limitations of CMC. Specifically, camera motion estimation may fail due to lack of background key points in scenes with high density of dynamic objects. Wrong camera movements can lead to unexpected tracker behavior [1]. However, there is no such limitation for robot motion compensation in ROTrack, because it relies on the movement data returned by the IMU instead of the camera image.

FPS. On computers running on mobile platforms, such as Jetson Orin, the frame rate of a tracker with CMC is significantly reduced. The IMU-based compensation in

ROTrack has basically no influence on the frame rate, which enables ROTrack to match the frame rate of trackers without CMC.

It will be shown that IMU compensation can achieve the same effect as CMC in terms of accuracy and other aspects by aligning IMU signals with CMC data. In this step, the Savitzky-Golay filter is applied to smooth and denoise the output IMU signals. The effect of the filter is drawn using the rotation as an example. The alignment of each signal is shown in Fig. 5.



(a) Rotation signal alignment. (b) Before and after filtering.

(c) X-direction signal alignment. (d) Y-direction signal alignment.

Fig. 5. Signals alignment in all directions.

The signals in all directions can be smoothed well by filtering. The signals of the IMU in terms of rotation and translation can be well matched to the CMC, which indicates that compensating with IMU signals can achieve similar effects to CMC. Since the IMU does not need to consider the limitations mentioned above, ROTrack can be used in any high-density dynamic complex scene without consuming additional computational resource.

Fig. 6 shows a real-world test scenario for ROTrack. The figure is derived from the submitted video, which contains camera movement caused by rotation. The left side is the prediction of the conventional Kalman Filter-based tracking system that do not utilize IMU data for robot motion compensation. The right side is the prediction after robot motion compensation by ROTtrack. It can be seen that the KF prediction after compensation is more accurate and in line with its expected position.

Fig. 7 shows the difference in x direction with or without RMC in the actual test. The mean value of the difference is 2.9 cm, the mean square error is 15.1 cm^2, and the maximum value can reach 10 cm. These are not small errors for robot tracking, especially at frame rate above 20 fps. When conducting robot planning, these errors can result in significant deviations in the calculated velocity. For example, at a frame rate of 20 fps, if the tracking point has a deviation

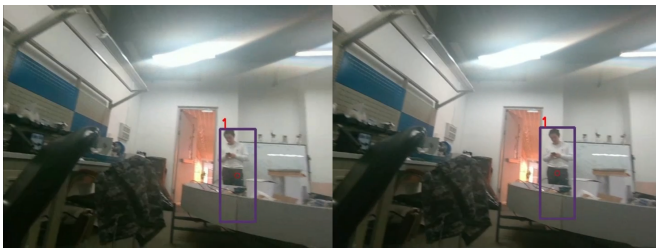


Fig. 6. Comparison before (left) and after (right) robot motion compensation.

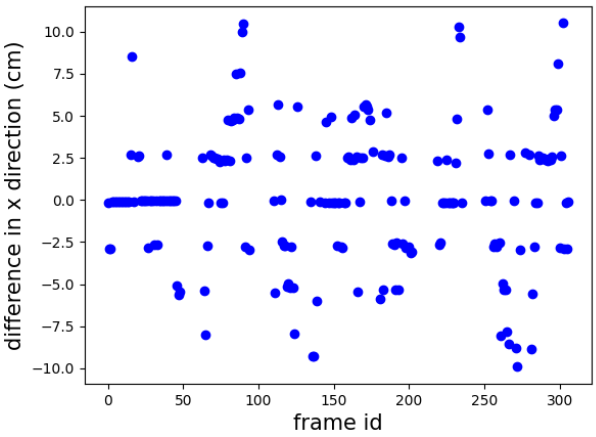
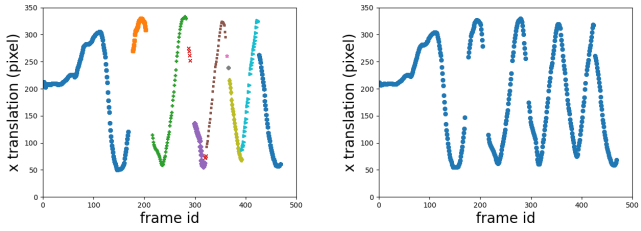


Fig. 7. The difference in the x direction between adding RMC and not adding RMC.



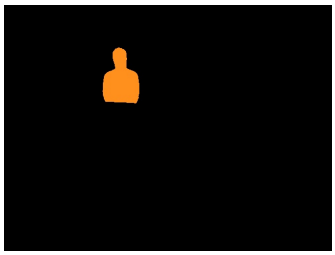
(a) Without RMC. (b) With RMC.

Fig. 8. The tracking results of the submitted video. Different colored dots represent different IDs.

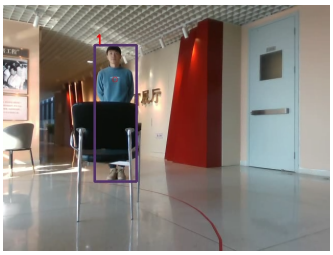
of 10 cm, the calculated velocity will have a deviation of 2 m/s. After adding RMC, the tracking frame can be more stable and the error can be effectively reduced.

Fig. 8 represents the tracking results of the submitted video. The video mainly shows the changes of view in the x direction, while the changes of y and z directions are small. Therefore, the variation of the x direction offset with the frame numbers is plotted. There is only one person in the video, so there is only one ID. In the absence of RMC during motion blur, ID switches are more drastic. The addition of RMC makes the character ID more stable and does not change dramatically with the shaking of the camera.

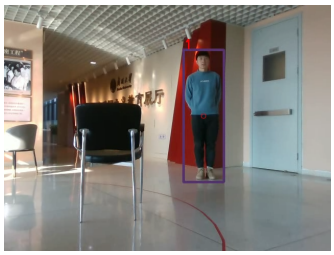
(a) The tracking point is occluded.



(b) Target mask.



(c) Update tracking point.



(d) The center point is recovered as the tracking point.

Fig. 9. Use MobileSAM to segment when occluding. The red dot in the bounding box is the tracking point.

(a) Without background point prompt.

(b) Incorrect segmentation results in incorrect tracking point.

(c) With background point prompt.

(d) Correct segmentation results in correct tracking point.

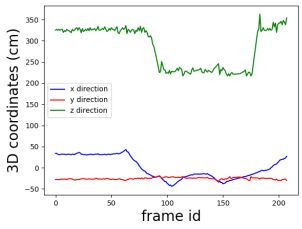
Fig. 10. Comparison with and without background point prompt in the case that the obstacle is large in proportion and similar in color.

B. Performance evaluation of occlusion resistance

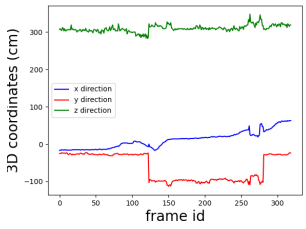
Fig. 9 shows the anti-occlusion process of ROTrack. At first, the center point of the target is taken as the tracking point, and its three-dimensional coordinates are obtained for tracking. As the tracking point is occluded, segmentation is triggered at this time. The tracking box is used as a prompt to get the mask, and the three-dimensional coordinates of the unoccluded part are obtained for tracking. In the end, when the obstacle disappears, the center point of the target is tracked again to avoid additional computational resource consumption.

As shown in Fig. 10, in some special cases, the color of the obstacle may be identical to that of the target, or the obstacle may occupy the majority of the target. This situation may result in obtaining the wrong mask. At this time, the occlusion point can be added as the background prompt to obtain a more accurate mask. The performance of MobileSAM is described in [2], and we have omitted the step of proof in this paper for the sake of simplicity.

Fig. 11(a) shows the variation in the three-dimensional coordinates of the target when occlusion occurs without the proposed method. At this point, there is a sudden change in the x direction and z direction (depth), while the change in y direction is minor. This is because during tracking, the height of the robot remains constant in the experimental field, which indicates that the change in the y direction is less affected by depth. However, the x-direction coordinate is more significantly affected by depth. In Fig. 11(b), the variation in the three-dimensional coordinates of the target with anti-occlusion is depicted. In this case, the x and z directions change smoothly, while the y direction has a sudden change. This is because we have updated the tracking



(a) Not add anti-occlusion.



(b) Add anti-occlusion.

Fig. 11. Comparison of 3D coordinates with and without anti-occlusion.

point in the case where the target is occluded from below (as shown in Fig. 9(c)), and the tracking point is now on the upper body of the person. This change might induce actions like raising the head or straightening the legs in different robots. In real-world environments, the occluded parts of the target are random. ROTrack can ensure that the obtained 3D coordinates represent the actual target rather than those of obstacles.

Because of the trigger mechanism, MobileSAM is not executed all the time. It is executed only when the target is occluded or fails to obtain the target depth. Therefore, the drop in frame rate caused by segmentation can be negligible.

V. CONCLUSION

In order to address the limitations of deploying multi-target tracking algorithms on novel mobile robots, such as the limitation of computational resource and the complexity of working environment, we propose a robust anti-occlusion

multi-target tracker named ROTrack. The robot motion compensation in ROTrack can make the robot work better in complex dynamic environment, and it does not cause extra computational resource consumption. We demonstrate the effect of ROTrack by matching various signals of IMU with CMC data and real-world testing. ROTrack uses MobileSAM to solve the problem of 3D coordinates acquisition failure caused by occlusion during robot tracking. It also further reduces the waste of computational resource through a depth-based triggered segmentation strategy. We hope that this work will be attractive in multi-target tracking applications for novel mobile robots.

REFERENCES

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

[2] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.

[3] M. Zhang, X. Liu, D. Xu, Z. Cao, and J. Yu, "Vision-based target-following guider for mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9360–9371, 2019.

[4] N. Pan, R. Zhang, T. Yang, C. Cui, C. Xu, and F. Gao, "Fast-tracker 2.0: Improving autonomy of aerial tracking with active vision and human location regression," *IET Cyber-Systems and Robotics*, vol. 3, no. 4, pp. 292–301, 2021.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[6] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, and C. Fu, "Tctrack: Temporal contexts for aerial tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 798–14 808.

[7] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, and J. A. Chambers, "Multi-target tracking and occlusion handling with learned variational bayesian clusters and a social force model," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1320–1335, 2015.

[8] A. A. I. Aly, A. Abbasimoshaei, and T. A. Kern, "Developing a vr training environment for fingers rehabilitation," in *13th International Conference on Human Haptic Sensing and Touch Enabled Computer Applications, EuroHaptics 2022*, 2022, pp. 331–333.

[9] V. Bharathi and K. Sakthivel, "Unmanned mobile robot in unknown obstacle environments for multi switching control tracking using adaptive nonlinear sliding mode control method," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 3, pp. 3513–3525, 2022.

[10] A. Abbasimoshaei, T. Stein, T. Rothe, and T. A. Kern, "Design and impedance control of a hydraulic robot for paralyzed people," in *8th RSI International Conference on Robotics and Mechatronics, ICRoM 2020*, 2020.

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6569–6578.

[12] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3876–3886.

[13] 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.

[14] 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.

[15] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Transmot: Spatial-temporal graph transformer for multiple object tracking," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4870–4880.

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[18] 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.

[19] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[20] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8844–8854.

[21] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[22] 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.

[23] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2564–2571.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[27] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," *CSTR*, vol. 1, no. 2, p. 13, 2011.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[29] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 280–296.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

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

[32] 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.

[33] 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.

[34] 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.