IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 15, 2021, accepted January 9, 2022, date of publication January 18, 2022, date of current version January 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3144153

DET: Depth-Enhanced Tracker to Mitigate Severe Occlusion and Homogeneous Appearance Problems for Indoor Multiple-Object Tracking

CHENG-JEN LIU[1] AND TSUNG-NAN LIN[1,2], (Senior Member, IEEE)
[1]Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan

Corresponding author: Tsung-Nan Lin (tsungnan@ntu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2634-F-002-037 and Grant MOST 110-2218-E-002-018-MBK.

ABSTRACT Multiple-object tracking has long been a topic of interest since it plays an important role in many computer vision applications. Existing works are mostly designed for outdoor tracking, such as video surveillance and autonomous driving. However, the behaviors of objects in outdoor tracking scenarios do not fully reflect the tracking challenges in indoor tracking environments. In outdoor tracking scenarios, pedestrians and vehicles usually move uniformly from place to place on a simple straight path, and target appearances are usually different. In contrast, in indoor scenarios, such as choreographed performances, the dynamic behaviors of dancers lead to severe occlusions, and similar costumes present a homogeneous appearance problem. These severe occlusion and homogeneous appearance problems in indoor tracking lead to noticeable degradation in the performance of existing works. In this paper, we propose a depth-enhanced tracking-by-detection framework and a semantic matching strategy combined with a scene-aware affinity measurement method to mitigate occlusion and homogeneous appearance problems significantly. In addition, we introduce an indoor tracking dataset and increase the diversity of existing benchmark datasets for indoor tracking evaluation. We conduct experiments on both the proposed indoor tracking dataset and the latest MOT benchmarks, MOT17 and MOT20. The experimental results show that our method consistently outperforms other works on the convincing HOTA metric across the benchmarks and greatly reduces the number of identity switches by 20% compared to that of the second-best tracker, DeepSORT, in our proposed indoor MOT benchmark dataset.

INDEX TERMS Affinity measurement, computer vision, data association, depth estimation, multiple-object tracking, indoor tracking dataset.

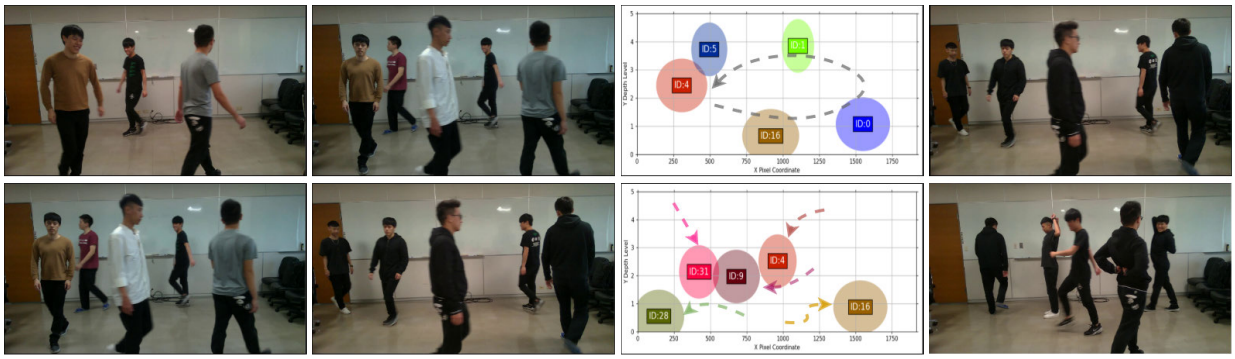
I. INTRODUCTION

The goal of multiple-object tracking (MOT) is to consistently localize and identify several objects in a video sequence. It plays an important role in many video analysis applications, such as video surveillance [1], autonomous driving [2], and sports analysis [3].

Existing works mostly follow the tracking-by-detection framework due to its simplicity and effectiveness and perform MOT tracking in a two-stage manner. In the first stage, an object detector is used to detect objects of interest in the current video frame. In the second stage, the detected objects are associated with tracks in the previous frame to form trajectories.

With the success of convolutional neural networks (CNNs) [4] in different computer vision tasks, many works [5]–[7] have replaced critical components, such as detection modules and feature extraction modules, in the tracking-by-detection framework with CNN networks and have focused on effectively learning the tracking task in an end-to-end manner in 2D space. Although these works greatly improved the tracking performance on public benchmark datasets, they cannot perform tracking well in indoor tracking scenarios, such as choreographed performances and stage performances. In these scenarios, people might dress similarly, and there are many occlusions between objects due

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

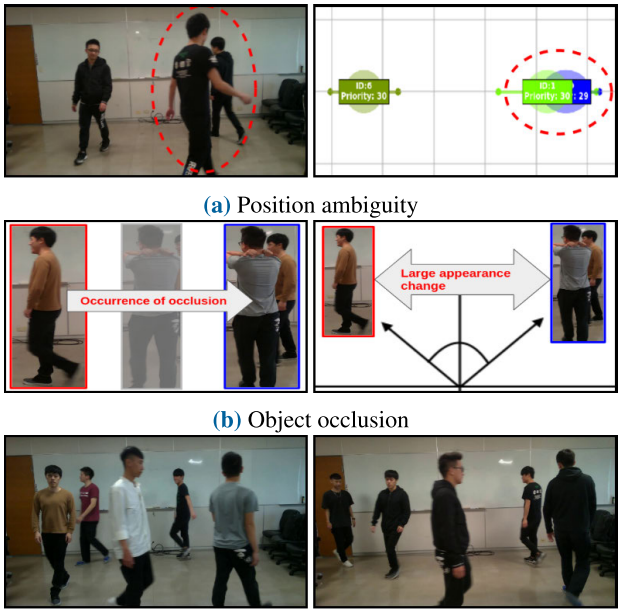


(a) Number of people (b) Appearance similarity (c) Movement pattern (d) Diversity of poses

FIGURE 1. All of the tracking scenarios in the proposed NTU-MOTD tracking dataset can be categorized in terms of (a) the number of people, (b) appearance similarity, (c) movement pattern, and (d) diversity of poses.

to frequent physical interactions and complex movements; it is difficult for existing works relying solely on 2D spatial information and appearance information to deal with these conditions.

Existing MOT benchmark datasets cover a variety of categories. The MOT16 [8] tracking dataset contains pedestrian videos, the KITTI [9] autonomous driving dataset contains car view videos, and UA-DETRAC [10] contains traffic videos. The video sequences in these datasets are mostly captured in outdoor scenes, and the objects of interest are usually pedestrians who are dressed differently and move in a single direction. These datasets do not genuinely reveal the tracking challenges depicted in Figure 2 in indoor tracking scenarios. In indoor tracking scenarios, such as choreographed performances and stage performances, people move around unpredictably and have frequent physical interactions with others, which usually leads to position ambiguity and severe occlusion in 2D space. The position ambiguity problem, as shown in Figure 2(a), occurs in 2D space when two or more objects are visually crowded together, which means that trackers cannot distinguish the objects well through spatial information. Occlusion between objects, as shown in Figure 2(b), leads to corrupted feature representation of the occluded object, which means that trackers tend to consider the same object as two different entities before and after the occlusion. In addition, to make a choreographed scene harmonious, people usually wear similar costumes and have a similar appearance, as shown in Figure 2(c). This makes it difficult for tracking algorithms to differentiate people only by appearance information.



(a) Position ambiguity

(b) Object occlusion

(c) Similar appearance

FIGURE 2. Indoor MOT tracking challenges. (a) shows that trackers cannot differentiate objects because of position ambiguity in 2D space when the objects are crowded together. The right diagram is a bird's-eye view. (b) shows that the target object, with a brown shirt, is occluded by an object with a gray shirt, and the occlusion causes a significant change in the feature space to the occluded object in the right diagram, which makes trackers tend to differentiate images of the same entity as two different people. (c) shows that it is difficult for trackers and even humans to differentiate objects with similar appearances.

In this work, we present an indoor tracking dataset called NTU-MOTD that increases the diversity of existing MOT benchmark datasets and better reveals the tracking challenges in indoor tracking environments. The proposed dataset covers a variety of tracking scenarios to evaluate the robustness of the trackers used in indoor tracking scenarios. Specifically, the tracking scenarios are categorized by four factors, as shown in Figure 1: 1) the number of people, 2) the appearance similarity between objects, 3) the movement patterns of objects, and 4) the diversity of object poses. Along with the raw videos, we also provide high-quality depth maps collected from an Intel-RealSense L515 LiDAR camera [11] to allow researchers to utilize them in future works.

To address the challenges shown in Figure 2, we propose a depth-enhanced tracker (DET) that enhances the tracking-by-detection framework by incorporating a depth estimation module and propose a semantic matching strategy combined with the scene-aware affinity measurement method

to dynamically combine different types of features given different types of scenes and the states of tracks to obtain better associations. Specifically, we extend the tracking state space from 2D to 3D with the depth estimation module. Under the 3D state space, the occlusion problem and positional ambiguity issue in 2D space can be solved effectively, as there cannot be multiple objects occupying the same 3D position, and we can determine the occlusion status between objects based on their relative positions on the z-axis. Before measuring the affinity values between objects, the type of scene is first determined, and the tracker selects different discriminative spatial features given the type of scene to perform affinity measurement. Generally, the proposed tracker fuses the spatial feature with the appearance feature so that it can track objects well even when they have similar appearances. To reduce the number of incorrect pairing results in the data association, we propose a semantic matching strategy that exploits the properties of tracks under different states and adopts suitable affinity functions to perform association in a more fine-grained manner.

The key contributions of our work are fourfold:

- A depth-enhanced tracking-by-detection framework combined with a scene detection module is proposed to address the challenging occlusion problem and positional ambiguity issue in indoor tracking scenarios. Furthermore, the scene detection module makes our tracker robust to both indoor tracking and outdoor tracking.
- A semantic matching strategy combined with scene-aware affinity measurement is proposed to dynamically combine different discriminative features to perform fine-grained data association and solve the homogeneous appearance problem effectively.
- An indoor tracking dataset called NTU-MOTD[1] is presented that increases the diversity of existing MOT benchmark datasets and can better reflect the robustness and performance of trackers in indoor environments. In addition to the raw videos, high-quality depth maps are provided.
- Comprehensive experiments are conducted on both the proposed indoor dataset and the public outdoor datasets MOT17 and MOT20. Our tracker beats other trackers in various tracking scenarios on the convincing HOTA metric and reduces the number of identity switches by almost 20% compared to the widely adopted DeepSORT on the indoor dataset.

II. RELATED WORKS

A. MOT BENCHMARK DATASET

Existing MOT tracking datasets can be roughly divided into two categories based on their purpose: video surveillance and autonomous driving.

Regarding video surveillance, a great number of different tracking datasets [8], [12]–[15] focus on tracking pedestrians and vehicles in different tracking scenarios. [12] provides video sequences of the same street view under multiple camera viewpoints. [8] provides video sequences with different crowd densities recorded by either movable or fixed-point cameras. [13] provides video sequences taken by drones from a bird's-eye view. [14] provides video sequences in urban scenarios created by exploiting the highly photorealistic video game GTA-V.

Regarding autonomous driving, the number of tracking datasets is relatively small compared with that of video surveillance datasets. Fortunately, with the surge in demand for autonomous driving technology, an increasing number of high-quality autonomous driving datasets have been released. [9], [16], [17] all provide information from multiple sensors, such as LiDAR point clouds, RGB image frames, GPS locations, and IMU measurements, for perception purposes. The video sequences are primarily traffic flows mixed with vehicles, cyclists, and pedestrians.

In contrast to these datasets focused on pedestrian and vehicle tracking in outdoor environments, the proposed dataset is acquired in indoor environments and simulates choreographed and stage performance scenes, presenting frequent occlusions and objects with similar appearances. These situations tend to cause performance degradation when performing tracking in indoor tracking environments in practice.

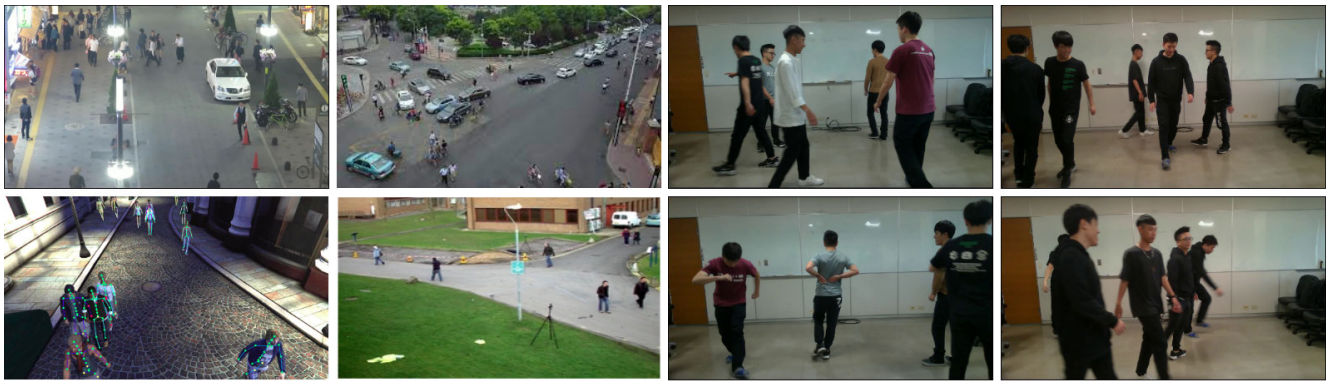
B. MOT TRACKERS

According to the information they use, trackers can be divided into two categories: offline trackers and online trackers. Offline trackers can utilize future information, such as objects in future frames, to perform tracking in the current frame. Online trackers can only use past information up to the current frame to perform tracking.

Typically, offline trackers formulate the tracking problem as an optimization problem. Several objects in a batch of frames are used to form a network graph. Offline trackers then apply optimization techniques to find the optimal solution to link nodes in the graph together to form trajectories. [18] considered the optimization problem as an energy minimization problem and adopted integer programming [19] to solve it. [20], [21] formed the graph as a network flow graph and applied either the min-cost network flow [22] algorithm to link nodes together or the message passing technique used in a graph convolutional neural network (GCN) [23] to aggregate node features and perform link prediction between them.

Online trackers usually follow the tracking-by-detection framework. Objects are first detected in the image, and then a greedy assignment method, such as Hungarian assignment [24], is conducted to link the detected objects with the tracks in the previous frame with minimum cost to form trajectories. Recent works have focused on designing end-to-end learning frameworks to solve tracking problems or on improving various submodules in tracking-by-detection frameworks with corresponding deep learning models. [6], [25], [26] proposed end-to-end feature extraction and feature matching for data association using either the GCN approach or CNN approach. Various object detectors [27]–[30] have

[1] https://pc217.ee.ntu.edu.tw



(a) Snapshots of existing datasets (b) Snapshots of the NTU-MOTD dataset

FIGURE 3. An overview of existing tracking datasets and the proposed tracking dataset. (a) shows video snapshots from the MOT16, VisDrone, JTA, and PETS2009 tracking datasets. (b) shows video snapshots of the proposed dataset.

been proposed and incorporated into tracking pipelines to improve the tracking performance. [31] further leveraged the box regression head in Faster-RCNN [30] to act as the motion estimator instead of a linear Kalman filter [32] and to greatly reduce the false negative and false-positive detections that largely affect the tracking performance. To improve the data association process, [33]–[35] enhanced the feature extraction module to extract discriminative and stable object feature representations. In detail, [33] trained a CNN model with a modified cosine metric loss to obtain discriminative feature representations. [34] extracted long-term and short-term features and combined them to obtain robust feature representations. [35] estimated the occlusion levels of objects using learned visibility maps to control the mechanism for updating feature representations.

Compared to previous works limiting the tracking space in 2D space, we extend the tracking space from 2D to 3D with the aid of a depth estimation module and solve the critical occlusion problem effectively with 3D spatial information rather than by generating visibility maps or response maps from 2D images as in previous works. With discriminative 3D spatial information, the ambiguity caused by similar appearances can also be resolved by combining spatial information and appearance information evenly to leverage the discriminativeness of 3D spatial features.

III. PROPOSED DATASET

The proposed NTU-MOTD tracking dataset aims at presenting a wide variety of indoor tracking scenarios and highlighting the challenges exhibited in indoor MOT tracking. In Figure 3 and Table 1, we show the difference between existing datasets and our dataset visually as well as an overview of the sequences included in the dataset. Compared to existing datasets, mostly presenting outdoor video sequences and focusing on tracking pedestrians, the proposed dataset provides indoor video sequences and aims at tracking dynamic objects. In the next sections, we introduce the dataset in detail.

A. DATASET COLLECTION ENVIRONMENT

The video sequences in the dataset were all recorded in an empty indoor environment. We used an Intel-RealSense LiDAR camera L515 to record RGB images and real depth map information. When recording the videos, we set the camera at a distance of approximately 5 meters from the depth of field and kept the camera at a medium viewpoint for shooting. Under these settings, the camera can accommodate roughly 5 to 7 people without overcrowding in the scene.

B. DATASET STATISTICS

To cover a variety of indoor tracking scenarios, we define four different recording conditions to realize in the dataset. These recording conditions are the number of people in the video, the similarity of the appearances of objects, the movement patterns of objects in the space, and the diversity of poses. Under each recording condition, we further specify two different scenes. Combining all the situations, there are 16 videos representing each tracking scenario and 4 longer versions of videos fusing different tracking scenarios together. The statistics of the dataset are shown in Table 2.

We provide the detections predicted by Mask-RCNN along with the video sequences in the same format as those in the MOT16 dataset. The bounding boxes detected in each frame have already undergone nonmaximum suppression (NMS) to filter out redundant bounding boxes. Specifically, the intersection-over-union (IoU) threshold of NMS for the region proposal network (RPN) head in the Mask-RCNN is 0.7, and the IoU threshold of NMS for the region of interest (ROI) head in the Mask-RCNN is 0.5. A breakdown of the detections on individual sequence is provided in Table 3.

We encourage future works to use the detections provided in the dataset, as the performance of trackers is highly dependent on the underlying detector. In this way, when evaluating tracker performance, we can make a fair comparison and confidently attribute the improvement to components other than the detector in the tracking-by-detection framework.

TABLE 1. Statistics of the NTU-MOTD dataset.

Name	FPS	Resolution	Frames	Tracks	Boxes	Camera	Viewpoint	Scene
3p_da_pm_pp	30	1920x1080	655	3	1,759	static	medium	indoor
3p_da_pm_up	30	1920x1080	675	3	1,780	static	medium	indoor
3p_da_um_pp	30	1920x1080	715	3	1,853	static	medium	indoor
3p_da_um_up	30	1920x1080	643	3	1,572	static	medium	indoor
3p_sa_pm_pp	30	1920x1080	674	3	1,738	static	medium	indoor
3p_sa_pm_up	30	1920x1080	664	3	1,757	static	medium	indoor
3p_sa_um_pp	30	1920x1080	667	3	1,728	static	medium	indoor
3p_sa_um_up	30	1920x1080	634	3	1,664	static	medium	indoor
5p_da_pm_pp	30	1920x1080	631	5	2,422	static	medium	indoor
5p_da_pm_up	30	1920x1080	636	5	2,444	static	medium	indoor
5p_da_um_pp	30	1920x1080	641	5	2,472	static	medium	indoor
5p_da_um_up	30	1920x1080	648	5	2,516	static	medium	indoor
5p_sa_pm_pp	30	1920x1080	710	5	2,552	static	medium	indoor
5p_sa_pm_up	30	1920x1080	650	5	2,521	static	medium	indoor
5p_sa_um_pp	30	1920x1080	673	5	2,526	static	medium	indoor
5p_sa_um_up	30	1920x1080	645	5	2,472	static	medium	indoor
3p_da_40sec	30	1920x1080	1200	3	3,723	static	medium	indoor
3p_sa_40sec	30	1920x1080	1200	3	3,728	static	medium	indoor
5p_da_60sec	30	1920x1080	1800	5	9,483	static	medium	indoor
5p_sa_60sec	30	1920x1080	1800	5	9,470	static	medium	indoor

TABLE 2. Description of 20 video sequences in the NTU-MOTD dataset.

Name	Description
3p_da_pm_pp	3 people with diverse appearances move around in a predicted motion pattern with little pose variation.
3p_da_pm_up	3 people with diverse appearances move around in a predicted motion pattern with large pose variation.
3p_da_um_pp	3 people with diverse appearances move around in an unpredicted motion pattern with little pose variation.
3p_da_um_up	3 people with diverse appearances move around in an unpredicted motion pattern with large pose variation.
3p_sa_pm_pp	3 people with similar appearances move around in a predicted motion pattern with little pose variation.
3p_sa_pm_up	3 people with similar appearances move around in a predicted motion pattern with large pose variation.
3p_sa_um_up	3 people with similar appearances move around in an unpredicted motion pattern with little pose variation.
3p_sa_um_up	3 people with similar appearances move around in an unpredicted motion pattern with large pose variation.
5p_da_pm_pp	5 people with diverse appearances move around in a predicted motion pattern with little pose variation.
5p_da_pm_up	5 people with diverse appearances move around in a predicted motion pattern with large pose variation.
5p_da_um_pp	5 people with diverse appearances move around in an unpredicted motion pattern with little pose variation.
5p_da_um_up	5 people with diverse appearances move around in an unpredicted motion pattern with large pose variation.
5p_sa_pm_pp	5 people with similar appearances move around in a predicted motion pattern with little pose variation.
5p_sa_pm_up	5 people with similar appearances move around in a predicted motion pattern with large pose variation.
5p_sa_um_pp	5 people with similar appearances move around in an unpredicted motion pattern with little pose variation.
5p_sa_um_up	5 people with similar appearances move around in an unpredicted motion pattern with large pose variation.
3p_da_40sec	A fusion version of indoor tracking scenarios. 3 people dressed differently are shown in the video.
3p_sa_40sec	A fusion version of indoor tracking scenarios. 3 people dressed similarly are shown in the video.
5p_da_60sec	A fusion version of indoor tracking scenarios. 5 people dressed differently are shown in the video.
5p_sa_60sec	A fusion version of indoor tracking scenarios. 5 people dressed similarly are shown in the video.

TABLE 3. Detection bounding box statistics.

Name	nDet.	nDet./fr.	min height	max height
3p_da_pm_pp	1,759	2.7	37.18	630.61
3p_da_pm_up	1,780	2.6	42.63	594.65
3p_da_um_pp	1,853	2.6	40.15	560.70
3p_da_um_up	1,572	2.4	38.92	664.01
3p_sa_pm_pp	1,738	2.6	28.85	755.58
3p_sa_pm_up	1,757	2.6	43.35	713.93
3p_sa_um_pp	1,728	2.6	41.13	522.45
3p_sa_um_up	1,664	2.6	43.18	538.91
5p_da_pm_pp	2,422	3.8	28.79	721.99
5p_da_pm_up	2,444	3.8	41.30	705.30
5p_da_um_pp	2,472	3.9	33.72	595.82
5p_da_um_up	2,516	3.9	31.18	663.43
5p_sa_pm_pp	2,552	3.6	35.51	584.48
5p_sa_pm_up	2,521	3.9	40.79	770.33
5p_sa_um_pp	2,526	3.8	34.46	583.68
5p_sa_um_up	2,472	3.8	39.77	635.29
3p_da_40sec	3,723	3.1	52.79	845.41
3p_sa_40sec	3,728	3.1	44.84	837.74
5p_da_60sec	9,483	5.3	23.51	847.73
5p_sa_60sec	9,470	5.3	47.67	855.23

C. GROUND-TRUTH ANNOTATION

The ground-truth trajectories in the video sequences are annotated manually with the aid of the SORT [36] tracker to provide the baseline annotations. The annotation process is the same as that in other popular annotation tools, such as CVAT [37]. An automatic labeling tracker is first used to obtain the baseline annotations. Given the initial annotations, incorrect labels of trajectories can be corrected or broken trajectories of the same identity can be linked to obtain better annotation results.

An important labeling principle we follow is that an object, no matter how long it is occluded, has the same label as before it became occluded when it appears on the screen again. This principle is necessary for evaluating the ability of trackers to preserve the identities of objects in a long-term tracking process. However, this does not hold in the popular MOT16 [8] benchmark dataset. As stated in its original paper,

- *If a person leaves the field of view and appears at a later point, they are assigned a new ID.*

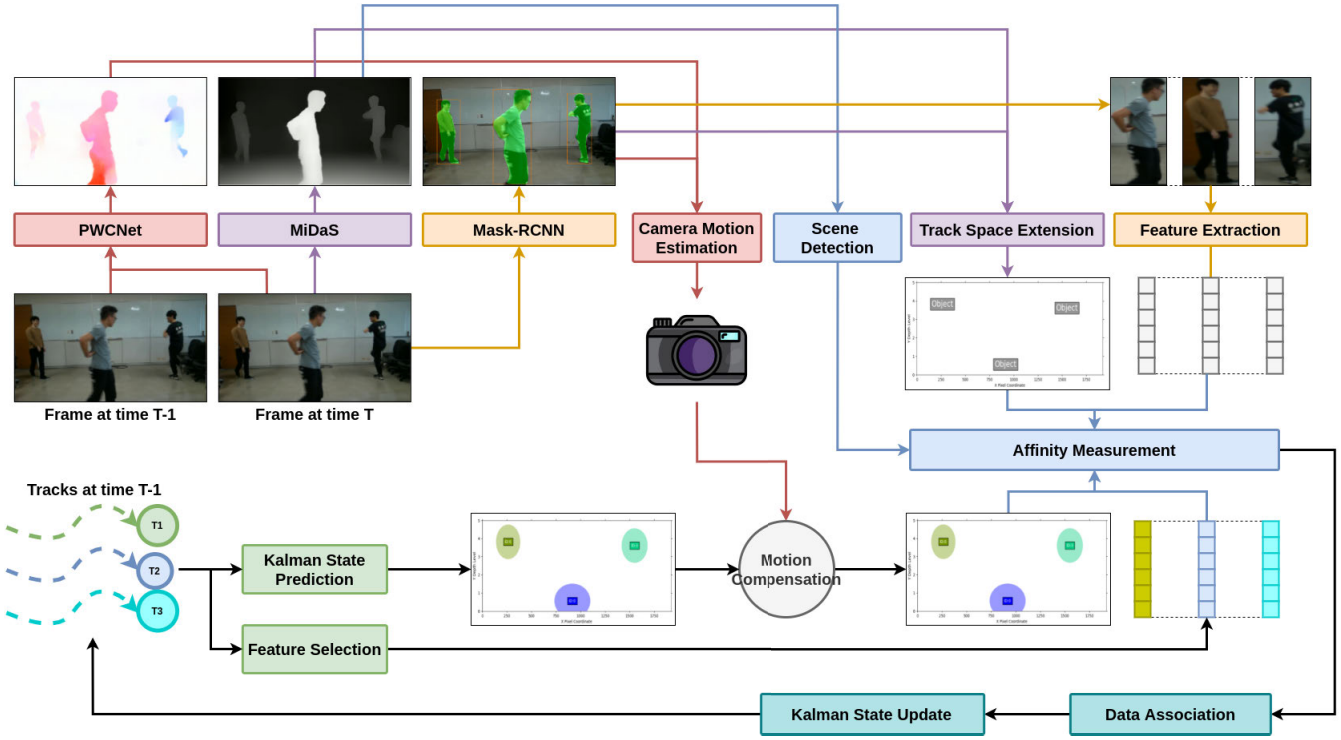


FIGURE 4. Depth-enhanced tracking-by-detection framework. At time t, flow estimation, depth estimation, and instance segmentation are performed on the current frame. Then, 2D detected objects are combined with the depth map to extend the state space from 2D to 3D, and their feature representations are computed with the feature extraction module. Before linking the objects with the tracks, the candidate feature vector of each track is selected, and each track at time $t-1$ is carried to the current frame using a Kalman filter with a calibrated motion vector. After that, affinity values between objects and tracks are computed with different policies given the states of the tracks and the type of the current scene. Finally, the Kalman states of the tracks are updated with the associated objects.

- *If a target reappears after a prolonged period such that its location is ambiguous during occlusion, it will reappear with a new ID.*

Therefore, an object is assigned a new ID whenever it leaves the scene and enters the scene again or reappears after a long period of occlusion. We believe that MOT16 does not consistently label tracking objects. This makes the performance evaluation slightly unfaithful. The dataset we propose consistently retains the identity information of the object, which can faithfully reflect trackers' abilities to preserve object identities.

IV. PROPOSED METHOD

Severe occlusion and homogeneous appearance problems are significant and critical for MOT tracking, especially in indoor tracking scenarios. To address these problems, we propose a depth-enhanced tracker (DET) incorporating a depth estimation module into the typical tracking-by-detection framework to solve the severe occlusion problem effectively, and we introduce the semantic matching strategy combined with the scene-aware affinity measurement method to make the critical association process robust to different tracking scenarios, such as outdoor scenes and indoor scenes in which objects have similar appearances. The experimental results show that our tracker is more robust to different tracking environments and can outperform others by a large margin, specifically in indoor tracking scenarios.

The proposed tracking pipeline is illustrated in Figure 4. Flow maps, depth maps, and objects are first extracted from every incoming frame. Then, the tracking state space of objects is extended from 2D to 3D with the depth map, and the feature representation of each object is computed with the feature extractor. Before measuring the affinities between objects and tracks, the spatial feature of each track is predicted from the previous frame to the current frame using a Kalman filter, and a dedicated appearance feature vector is selected for each track. After that, affinity values between objects and tracks are computed with different policies given the state of the tracks and the type of the current scene. Finally, the Hungarian assignment is conducted to link each object to the proper track, and the spatial feature of each track is updated with the associated object. In the next sections, we introduce the method in detail.

A. EXTENDING THE TRACKING SPACE TO SOLVE THE SEVERE OCCLUSION PROBLEM IN INDOOR TRACKING

Tracking in a 2D space will inevitably encounter position ambiguity and occlusion problems. When two or more objects are visually close together, they occupy almost the same position (x, y), which results in position ambiguity,

and trackers cannot properly differentiate them. Additionally, physical interactions between objects commonly lead to occlusion, which results in corrupted appearance features. The situation becomes more serious when performing indoor tracking, as objects have more interactions with others and move around more dynamically than pedestrians in outdoor environments. To solve these problems, we extend the tracking state space from 2D to 3D. In a 3D space, each object has a unique position (x, y, z), and the occlusion status of an object can be effectively determined based on its position relative to others.

Given an image frame, the associated depth map is inferred by MiDaS [38], a robust depth estimation model pretrained on several depth datasets across a variety of environments. The value range of the depth map is normalized between 0 and 1. In addition, the 2D bounding boxes and segmentation masks of objects are extracted with the Mask-RCNN [39]. Then, to extend the tracking state space from 2D to 3D, a target segmentation mask is used to filter out target depth pixels with the cropped depth map, and the mean value of these filtered depth pixels is multiplied with a scale factor to represent the virtual depth level z. As shown in Figure 5, since the cropped depth map may contain depth pixels from the background, target, and occluders, there are several peaks in the depth distribution. Therefore, to determine the unique depth peak representing the depth of the target, the occlusion-aware segmentation mask of the target is used to filter out only the depth pixels belonging to itself.

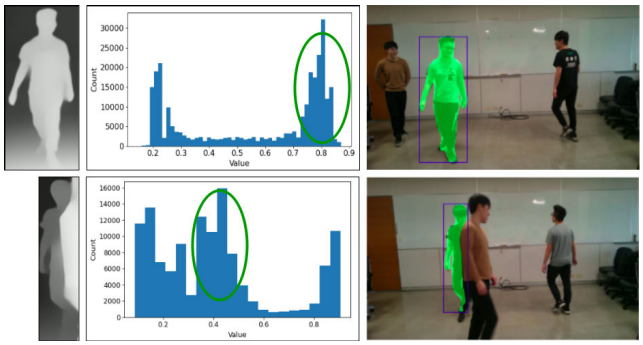


FIGURE 5. Distributions of depth pixel values. The leftmost column shows the cropped depth map of the object. The middle column shows the distribution of the depth pixel values with several peaks in the cropped depth map. The green circle indicates the true peak for the target, and the target depth pixels are filtered out with the target segmentation mask. The rightmost column shows the selected target with a segmentation mask in the raw image frame.

To solve the occlusion problem effectively, the estimated virtual depth level (z) and the center position (x, y) of each object are combined to extend the state space from 2D to 3D. The combined spatial feature (x, y, z) is maintained with the proposed 3D Kalman filter to handle noisy observations, such as false-negative and false-positive detections and inaccurate depth maps. As shown in Figure 6, in 3D space, each track has its own unique position (x, y, z), and the tracker can leverage the uniqueness of the 3D position to differentiate

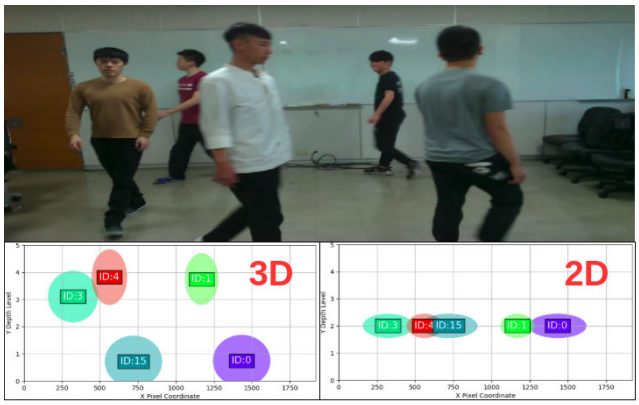


FIGURE 6. Bird's-eye view in 2D and 3D tracking state spaces. Each contour in the diagram represents a tracked object. In 3D space, each track can be effectively differentiated given their positions. However, in 2D space, objects are squeezed together, and it is difficult to differentiate them when they have ambiguous positions.

objects. Furthermore, the occlusion status of each object can be determined by its overlap ratio with others along the x-axis and its relative position to others along the z-axis. For precise appearance feature management, the tracker can refuse to add a corrupted appearance feature to the feature set of the target object if the object is occluded by others.

B. SCENE-AWARE SPATIAL FEATURE SELECTION AND APPEARANCE FEATURE EXTRACTION

The process of measuring affinity values between tracks and objects usually considers multiple pieces of information. This information should be discriminative and unique so that trackers can rely on it to effectively identify different objects and associate objects that are the same. To obtain a good affinity matrix between tracks and objects, we mainly utilize two kinds of discriminative information: spatial features and appearance features. In addition, to make our tracker robust to different tracking environments, we leverage the proposed scene detection method to dynamically select robust spatial features given the type of the recognized scene. In the description below, we first describe how the type of the current scene is determined. Then, we describe how to extract proper spatial features and appearance features before estimating the affinity matrix between tracks and objects.

Since the prediction of the depth estimation module is not very accurate in an outdoor environment, as shown in Figure 7, the proposed scene detector is used to recognize the type of the current scene and provide a signal to the tracker so that it can dynamically select different robust spatial features given the type of scene. Specifically, we refer to 2D spatial features in outdoor tracking environments and 3D spatial features in indoor tracking environments.

To determine whether the current frame is an outdoor scene or an indoor scene, we train a shallow CNN model to perform binary classification on the depth map. We sample 3,000 outdoor depth maps generated by MiDaS from the MOT16 training sequences and 3,000 indoor depth maps generated

FIGURE 7. Inaccurate depth estimation in outdoor environments.

by MiDaS from the proposed dataset. The model is trained with a binary cross-entropy loss and Adam optimizer with 50 epochs. Given a resized depth map of resolution 256×256, the model outputs a 2-dimensional vector representing the probability that the depth map belongs to an outdoor scene or an indoor scene.

Following the procedures described in Sec. IV-A, we can extract the extended 3D spatial features of objects from the current frame and obtain the 3D spatial features of the tracks maintained with the 3D Kalman filter in the previous frame. Before estimating the affinity values based on the spatial features between tracks and objects, the spatial features of the existing tracks in the previous frame are predicted for the current frame. Then, if the predicted type of the current scene is an outdoor scene, we project the 3D spatial features into 2D space to deal with the inaccurately estimated depth maps in outdoor environments. In addition, to address possible track drifting caused by a moving camera, we perform dense optical flow estimation with the PWCNet [40] and consider the mean xy offset of the background pixels as the global camera motion to calibrate the predicted tracks in the (x, y) dimension.

Regarding the appearance features, we use a ResNet50 [41] CNN model trained on the training sequences of the MOT16 dataset with online triplet loss [42] to extract the normalized 128-dimensional feature vector for each object, and the previously matched feature vectors are stored in the fixed-size feature pool of the target tracks. The size of the feature pool of each track is determined by the frame rate of the video. Typically, we store the appearance information for each second. Therefore, if the frame rate of the video is 30 FPS, then the size of the feature pools is 30. When performing affinity measurement based on the appearance features, the mean feature vector of the feature pool of each track is considered in computing the affinity values with the observed appearance features of objects.

C. SEMANTIC MATCHING STRATEGY FOR SOLVING THE HOMOGENEOUS APPEARANCE PROBLEM IN INDOOR TRACKING

During the tracking process, the state of each track continuously changes. Under different states, the tracks exhibit different properties. To take full advantage of the properties of tracks under different states, we design a four-state finite state machine representing the life cycle of each track to capture the semantic context in different states; the states are *tentative*, *tracked*, *lost*, and *dead*. Under each state, an appropriate affinity function utilizing different combinations of heterogeneous information, such as spatial features, appearance features, and IoUs, is adopted to perform semantic data association. This dynamic process makes our tracker more robust to different tracking contexts. One example is combining the discriminative 3D spatial features with the appearance features to effectively solve the homogeneous appearance problem in indoor tracking where targets are dressed similarly. The experiments consistently show the improvement in tracking with the proposed semantic matching strategy. In the description below, we first describe how we compute the affinity values based on different types of information and then demonstrate how we leverage them to perform semantic data association.

For the spatial affinity, given the calibrated spatial features of existing tracks $U = \{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n\}$ and the newly detected observations $X = \{\vec{x}_1, \vec{x}_2, \ldots \vec{x}_m\}$, we compute the affinity value regarding spatial features with the Mahalanobis distance between track i and object j with respect to either the (x, y, z) or (x, y) dimension given the type of the current scene.

$$d_{i,j} = (\vec{x}_j - \vec{u}_i)^T \Sigma_i^{-1} (\vec{x}_j - \vec{u}_i) \tag{1}$$

where Σ_i is the associated feature covariance matrix maintained by the Kalman filter with respect to track i. Then, to allow the spatial affinity to be integrated with the later appearance affinity, we normalize the distance values with a softmax operation so that the orders of magnitude of the spatial affinity and appearance affinity are in the same range [0, 1].

$$p_{i,j} = \frac{e^{-d_{i,j}}}{\sum_{k=1}^{m} e^{-d_{i,k}}} \tag{2}$$

Accordingly, a position-based affinity matrix $A^{(1)}$ can be constructed, where $A^{(1)}_{i,j} = p_{i,j}$.

For the appearance affinity, we refer to the commonly used cosine similarity to measure the affinity values between tracks and objects. In detail, for each candidate pair consisting of a track i and observation j, we use the mean appearance feature vector $\vec{x}_i$ of the feature pool of the track and the extracted appearance feature vector $\vec{y}_j$ of the observation in the current frame to compute the cosine similarity.

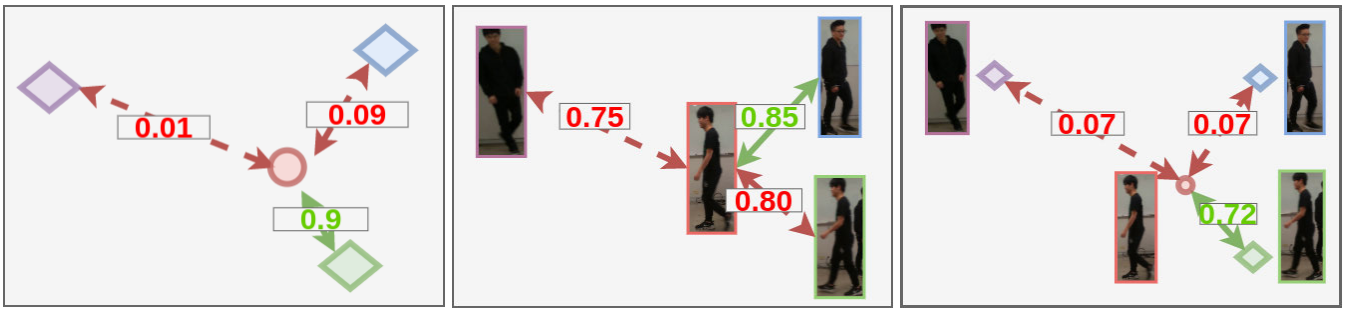
$$s_{i,j} = max(\frac{\vec{x}_i \cdot \vec{y}_j}{\|\vec{x}_i\| \times \|\vec{y}_j\|}, 0) \tag{3}$$

Accordingly, an appearance-based affinity matrix $A^{(2)}$ can be constructed, where $A^{(2)}_{i,j} = s_{i,j}$.

By combining these two affinity matrices, $A^{(1)}$ and $A^{(2)}$, with the Hadamard product operation

$$A = A^{(1)} \odot A^{(2)}, \tag{4}$$

we can obtain a scene-aware affinity matrix A with equal importance of both kinds of information. As shown in Figure 8, with this heterogeneous affinity matrix with equal



(a) Mahalanobis softmax distance (b) Cosine similarity (c) Combined affinity function

FIGURE 8. Affinity measurement with equal importance of spatial features and appearance features. (a): the Mahalanobis softmax distance measures affinity values based on spatial features. (b): the cosine similarity measures affinity values based on appearance features. (c): combined affinity function. The incorrect associations in (b) can be corrected based on the discriminative spatial features in (a), and vice versa.

importance of different types of information, we can correct the association result with the discriminative spatial feature when objects have similar appearances. This allows our tracker to effectively solve the homogeneous appearance problem for indoor tracking.

To capture the current semantic context of the target tracks, we define a four-state finite state machine for each track, as shown in Figure 9. When a track is first created, it enters the *tentative* state. When 3 continuous observations are matched, it enters the *tracked* state; otherwise, it enters the *dead* state. Once in the *tracked* state, if a track is mismatched for 3 consecutive frames, it enters the *lost* state. When an observation is matched to a track in the *lost* state, that track immediately returns to the *tracked* state. If a track is mismatched for 30 consecutive frames, it enters the *dead* state and is discarded by the tracker. In what follows, we describe how we select the proper affinity function to perform data association with the tracks in each state.

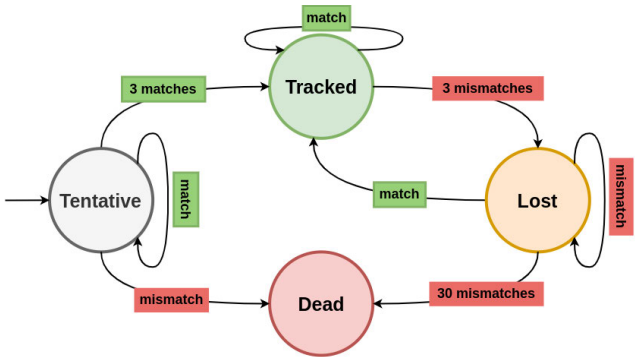


FIGURE 9. Finite state machine describing the life cycle of each track.

Because tracks in the *tentative* state might be false-positive tracks, a strict affinity function combining the Mahalanobis softmax distance and the IoU is used to reduce the number of false-positive tracks. Because the IoU is sensitive to pose changes and noise in detection, we consider it a strict metric. The information used here includes spatial features and object

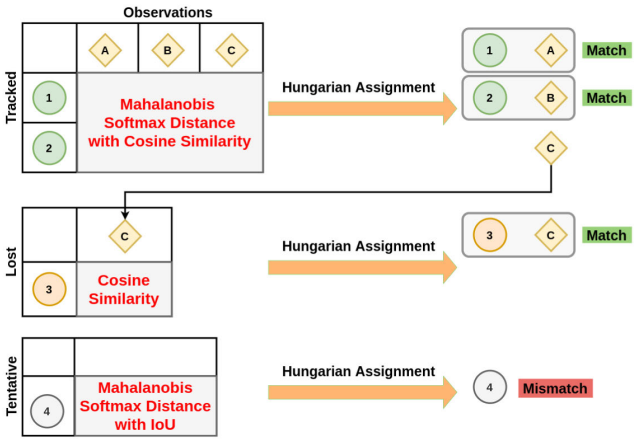


FIGURE 10. Order of execution of the matching process.

shapes, and the affinity matrix is constructed with Equation 1, Equation 2, and the IoU.

Because tracks in the *tracked* state have been located and identified by the tracker at a certain time, they should have position continuity and appearance consistency. Therefore, an affinity function combining the Mahalanobis softmax distance regarding either 2D or 3D spatial features and the cosine similarity is used to perform data association. Consequently, even if the appearance information becomes inaccurate under occlusions, the tracker can rely on discriminative spatial features to obtain good matching results. The information used here includes spatial features and appearance features, and the affinity matrix is constructed with Equation 4.

Because tracks in the *lost* state have been lost by the tracker for several frames, they are likely to exhibit only appearance consistency. Therefore, an affinity function using cosine similarity is used to perform data association. The information used here includes only appearance features, and the affinity matrix is constructed with Equation 3.

Finally, to improve the tracking consistency and reduce the number of false-positive tracks, we execute the matching algorithm in the following order: we first match the *tracked*

(a) Snapshots of the MOT17 (2 diagrams on the left) and MOT20 (2 diagrams on the right) datasets

(b) Snapshots of the NTU-MOTD dataset

FIGURE 11. A visual comparison between MOT17, MOT20 and the proposed NTU-MOTD dataset. (a) shows video snapshots from the MOT17 and MOT20 datasets. (b) shows video snapshots from the proposed dataset.

set, then the *lost* set, and finally, the *tentative* set. The earlier that data association is performed for a set of tracks, the higher the chance that they will be matched to observations. The overall process is illustrated in Figure 10.

V. EXPERIMENTS

A. EVALUATION DATASET FOR INDOOR TRACKING

For indoor tracking evaluation, we conduct experiments on the proposed NTU-MOTD indoor tracking dataset, which covers a variety of tracking scenarios in indoor environments. There are 20 video sequences representing different tracking scenarios under different conditions, such as different numbers of people, appearance similarity between objects, movement patterns of objects, and diversity of object poses. The provided depth maps collected from the LiDAR camera are not used in the benchmark evaluation, but we use them in the ablation study to demonstrate the improvement of tracking with more accurate depth estimation. The statistics of the dataset can be found in Table 1. A visual comparison with the outdoor evaluation dataset is shown in Figure 11.

B. EVALUATION DATASET FOR OUTDOOR TRACKING

For outdoor tracking evaluation, we conduct experiments on the public MOT benchmark datasets MOT17 and MOT20. The video sequences in MOT17 are mostly recorded in outdoor environments with different crowd densities and camera viewpoints, and those in MOT20 are recorded in outdoor environments with very crowded scenarios and high camera viewpoints. Specifically, we perform evaluation only on the training sequences and not the test sequences, as ground-truth annotations are not provided in the test dataset, and they can only be evaluated on the benchmark website. Additionally, the trackers evaluated on the test sequences on the benchmark website use their own generated detections, which is not a fair comparison, as the quality of detections greatly affects the tracking performance. A visual comparison with the indoor evaluation dataset is shown in Figure 11.

C. EVALUATION METRICS

We use the same evaluation metrics as in the latest MOTChallenge benchmark. Among the metrics, we mainly focus on HOTA [43] because it is a more balanced and convincing metric than the previous ones, such as MOTA [44] and IDF1 [45]. Here is the formula of HOTA,

$$\begin{aligned} HOTA_\alpha &= \sqrt{DetA_\alpha \cdot AssA_\alpha} \\ &= \sqrt{\frac{\sum_{c\in TP_\alpha} AssIoU_\alpha(c)}{|TP_\alpha| + |FN_\alpha| + |FP_\alpha|}} \\ HOTA &= \int_{0<\alpha\leq 1} HOTA_\alpha \\ &\approx \frac{1}{19} \sum_{\substack{\alpha=0.05 \\ \alpha+=0.05}}^{0.95} HOTA_\alpha \end{aligned}$$

$DetA_\alpha$ measures the percentage of aligning detections, and $AssA_\alpha$ measures the percentage of aligning trajectories, averaged over all detections. The α value represents the localization threshold used to determine whether a pair of detections are aligned together or not. TP_α, FP_α, and FN_α represents the number of true-positive, false-positive, and false-negative tracks under localization threshold α.

From the formula of HOTA above, the tracking performance related to the tracking coverage ($DetA_\alpha$) and tracking association ($AssA_\alpha$) are considered simultaneously and computed with variants of the Jaccard index. This makes HOTA a balanced metric rather than one biased toward tracking coverage like MOTA or tracking association like IDF1.

In addition to HOTA, we include MOTA, IDF1, FP, FN, and IDs in the experimental tables. The formula of MOTA is as follows,

$$MOTA = 1 - \frac{FN + FP + IDs}{\sum_i GTi}$$

where FP and FN measure the number of false-positive tracks and false-negative tracks, IDs measures the number of

identity switches of all tracks during the tracking process, and GT_i means the number of ground truth track in frame i.

The formula of IDF1 is as follows,

$$IDF1 = \frac{2 * IDTP}{2 * IDTP + IDFP + IDFN}$$

where $IDTP$, $IDFP$, and $IDFN$ represents the number of true-positive trajectories, false-positive trajectories, and false-negative trajectories.

IDF1 computes the value with detections across frames, which form the trajectory, while MOTA computes the value with individual detection in each frame. Both MOTA and IDF1 give normalized values between 0 and 1 that measure the performance of tracking coverage and tracking association, respectively.

D. ABLATION STUDY

In this section, we conduct comprehensive experiments related to different algorithmic components in the proposed tracker to demonstrate the effectiveness of each component and find proper hyperparameter settings for each component.

1) TRACKING WITH DIFFERENT OBJECT DETECTORS

To show the influence of the object detection model on the tracking performance, we run the tracker with different object detectors, including DPM [46], SDP [27], POI [47], Faster-RCNN [30], and Mask-RCNN [39]. DPM, SDP, and Faster-RCNN are provided with the MOT17 dataset. POI is pretrained on an additional pedestrian dataset and surveillance dataset. Mask-RCNN is pretrained on the COCO dataset [48] and included in the PyTorch library. We filter out the detection results with confidence values below 0.8 and evaluate our tracker with different detection inputs on the single ground-truth source from MOT16, as shown in Table 4. The experimental results show that different object detectors will affect the overall performance of the tracker in almost every respect. Therefore, it is important to select a proper object detector to track objects well.

2) TRACKING WITH DIFFERENT DEPTH ESTIMATION MODELS

To show the influence of the depth estimation model on the tracking performance, we run the tracker with different depth estimation models, including LeRes [49], MiDaS [38], and a physical LiDAR sensor, on the proposed NTU-MOTD dataset. LeRes and MiDaS are both monocular depth estimation models based on deep learning. Compared with MiDaS, LeRes can generate more detailed high-resolution depth maps. The physical LiDAR sensor, an Intel-RealSense L515 camera, can generate sparse depth maps precisely and accurately with limited LiDAR points. We compare the performances of trackers on the NTU-MOTD dataset with the different depth estimation models in Table 5 and show their runtimes in Figure 12. In addition to our tracker, we include two 2D trackers, SORT [36] and DeepSORT [33], in the table to show the improvement obtained by using depth information. The experimental results show that depth information can help the tracker improve tracking association and mitigate the position ambiguity problem in 2D space if the underlying depth estimation model can separate objects in the z dimension precisely and accurately. Since LeRes has the worst accuracy when predicting the depth maps, it has the least improvement, as shown in Table 5.

TABLE 4. The proposed tracker DET run with different detectors on the MOT16 training dataset.

MOT16 Training Dataset						
Detector	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DPM [46]	425	83905	135	32.48%	23.50%	26.96%
SDP [27]	1676	43430	379	63.20%	58.80%	51.92%
POI [47]	3483	44640	309	62.60%	56.13%	50.92%
Faster-RCNN [30]	1435	54239	286	57.02%	49.32%	48.64%
Mask-RCNN [39]	9784	53861	355	53.28%	42.03%	42.97%

TABLE 5. The proposed tracker DET runs with different depth estimation models on the NTU-MOTD dataset. Two additional 2D trackers, SORT and DeepSORT, are included to compare with our tracker when employing depth information.

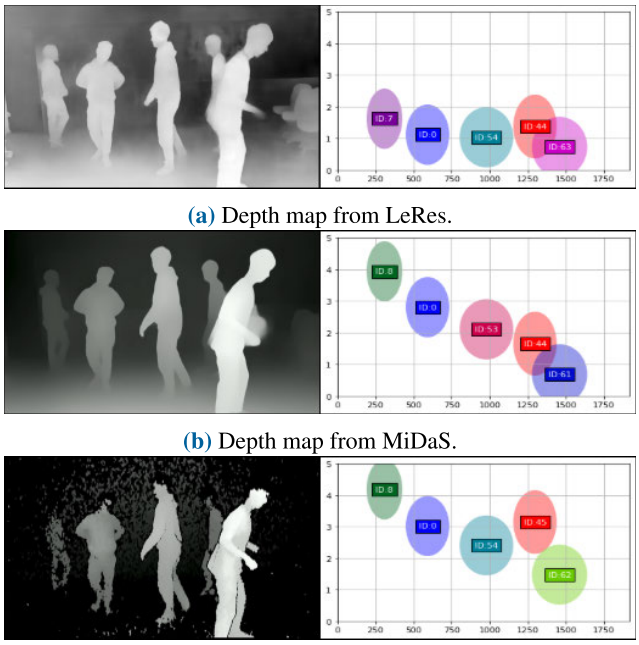
NTU-MOTD Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
SORT [36]	**2238**	1843	2124	20.67%	88.04%	29.75%
DeepSORT [33]	2321	2011	213	66.49%	91.32%	66.49%
DET (LeRes)	3423	543	182	73.15%	92.05%	69.34%
DET (MiDaS)	3354	563	170	79.67%	92.12%	73.99%
DET (Sensor)	3391	**539**	**160**	**79.83%**	**92.12%**	**74.48%**

3) DEPTH EXTRACTION WITH OR WITHOUT SEGMENTATION MASKS

In addition to the depth map information, the method of extracting the depth value of an object from the depth map will affect the performance of the tracker. We use two different methods to retrieve the depth value of an object. One filters out the depth pixels of the target object through its segmentation mask and takes the average pixel value as the depth value; the other randomly samples the depth pixels within the region of the bounding box of the target object and takes the average pixel value as the depth value. An illustration of these two methods is shown in Figure 13. For the experiment, we run our tracker with two different depth extraction methods under two different segmentation models, YOLACT [50] and Mask-RCNN [39], on the NTU-MOTD dataset. As shown in Table 6, regardless of which object segmentation model is used, using object masks can more accurately extract depth information and improve the performance of the tracker.

4) TRACKING WITH DIFFERENT MATCHING STRATEGIES

To verify whether the proposed semantic matching strategy shown in Figure 10 is the best among other matching strategies, we run our tracker on the NTU-MOTD dataset under eight different combinations of matching functions. These matching functions include mahalanobis softmax distance in

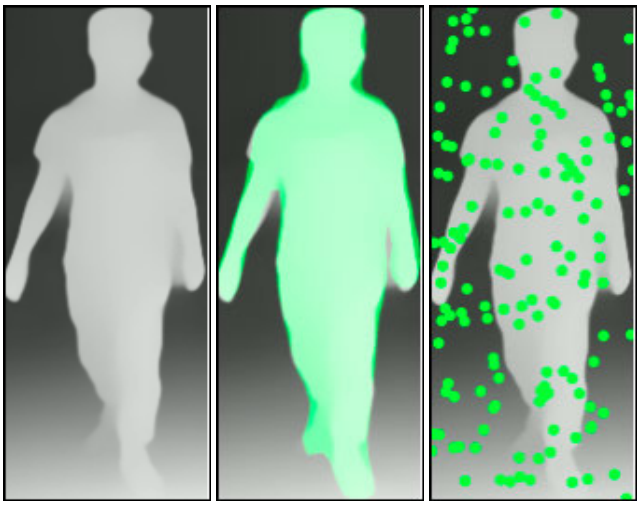


(a) Depth map from LeRes.

(b) Depth map from MiDaS.

(c) Depth map from the LiDAR camera.

FIGURE 12. Runtimes of the proposed tracker DET with different depth estimation models. The diagrams on the left show the depth maps generated from different models. The diagrams on the right show the bird's-eye views of different runtimes.



(a) Depth map (b) Masked depth (c) Sampled depth

FIGURE 13. Two different methods of extracting the depth value from the (a) depth map. The green pixels in the diagrams represent the depth pixels used to extract the depth value of an object. (b) uses a segmentation mask to filter out the target depth pixels. (c) samples the target depth pixels randomly within the bounding box of the object.

Equation 2 with cosine similarity in Equation 3 (mahacos), cosine similarity (cos), and mahalanobis softmax distance with IoU (mahaiou). The matching thresholds for different matching functions are all 0.3 by default. As shown in Table 7, our tracker performs best with the highest HOTA value 73.99% when using the proposed semantic matching strategy to perform the matching process.

TABLE 6. The proposed tracker DET uses different methods to extract the depth values of objects. Those with the "Box" suffix use random sampling inside the bounding box to extract the depth value. Those with the "Box+Mask" suffix use an object mask to filter out the target depth pixels to obtain the depth value.

NTU-MOTD Dataset with Mask-RCNN						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DET (Box)	3395	**559**	189	75.99%	92.07%	71.07%
DET (Box+Mask)	**3354**	563	**170**	**79.67%**	**92.12%**	**73.99%**
NTU-MOTD Dataset with YOLACT						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DET (Box)	4401	2755	477	41.43%	85.29%	43.10%
DET (Box+Mask)	**4221**	**2696**	**431**	**43.87%**	**85.84%**	**45.79%**

TABLE 7. The proposed tracker DET uses different combinations of matching functions to perform the matching process. "maha" stands for the mahalanobis softmax distance function, "cos" stands for the cosine similarity function, and "iou" stands for the IoU function. For "mahacos", it means we combine "maha" and "cos" together to compute the final affinity matrix. The proposed semantic matching strategy is the one using the (mahacos, cos, mahaiou) configuration.

NTU-MOTD Dataset								
Affinity Function			FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
Tracked State	Lost State	Tentative State						
mahacos	mahacos	cos	3603	549	199	75.67%	91.62%	71.44%
cos	cos	mahaiou	3411	568	173	76.11%	92.00%	71.96%
cos	mahacos	cos	3690	543	185	76.64%	91.49%	72.25%
cos	cos	cos	3762	**536**	175	76.91%	91.38%	72.37%
mahacos	mahacos	mahaiou	**3206**	586	174	76.94%	**92.36%**	72.38%
cos	mahacos	mahaiou	3333	577	168	76.84%	92.14%	72.40%
mahacos	cos	cos	3697	530	180	79.62%	91.51%	73.77%
mahacos	**cos**	**mahaiou**	3354	563	**170**	**79.67%**	92.12%	**73.99%**

5) TRACKING WITH DIFFERENT MATCHING THRESHOLDS

To find a suitable configuration of matching thresholds for different matching functions in the proposed semantic matching strategy as shown in Figure 10, we run our tracker on the NTU-MOTD dataset under eight different combinations of matching thresholds and choose the best one among them for subsequent experiments. The matching threshold determines whether a matching track-object pair is valid. If the matching cost is lower than the matching threshold, then it is considered a valid match. As shown in Table 8, the choice of matching thresholds for the matching functions in the semantic matching algorithm will affect the performance of the whole tracker. Therefore, we choose the best threshold with the highest

TABLE 8. The proposed tracker DET uses different combinations of matching thresholds for the matching functions in the proposed semantic matching strategy.

NTU-MOTD Dataset								
Threshold Value			FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
mahacos	cos	mahaiou						
0.5	0.5	0.3	3699	518	**148**	69.32%	91.59%	67.54%
0.5	0.5	0.5	4023	**474**	153	69.95%	91.04%	67.70%
0.3	0.5	0.5	4212	519	216	71.01%	90.47%	68.58%
0.3	0.5	0.3	3852	566	195	71.07%	91.11%	68.86%
0.5	0.3	0.5	3702	478	158	71.13%	91.64%	68.96%
0.5	0.3	0.3	3412	524	152	71.59%	92.12%	69.37%
0.3	0.3	0.5	3663	520	179	79.64%	91.59%	73.82%
0.3	**0.3**	**0.3**	**3354**	563	170	**79.67%**	**92.12%**	**73.99%**

HOTA value 73.99%, and we use a threshold value of 0.3 for all matching functions in the semantic matching strategy for later experiments.

6) TRACKING WITH OR WITHOUT SEMANTIC MATCHING STRATEGY

To verify the effectiveness and robustness of the proposed semantic matching strategy, we run our tracker on the MOT17, MOT20, and NTU-MOTD datasets with the semantic matching strategy and two homogeneous matching strategies, uniformly using either IoU or cosine similarity as the affinity function for tracks in different states to perform the matching process. As shown in Table 9, our tracker consistently improves the performance in HOTA across the datasets with the semantic matching strategy. This demonstrates the effectiveness and robustness of the semantic matching strategy.

TABLE 9. The proposed tracker DET run with the semantic matching strategy and two other homogeneous matching strategies, using either IoU or cosine similarity as the affinity function to perform the matching process.

MOT17 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DET (IoU)	5370	195699	3623	36.69%	39.24%	34.81%
DET (cos)	3968	**182788**	863	50.08%	**44.31%**	42.73%
DET (semantic)	**3401**	187129	**799**	**51.91%**	43.21%	**43.85%**
MOT20 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DET (IoU)	13529	472335	12648	41.83%	56.06%	38.63%
DET (cos)	15120	460561	7343	48.89%	57.44%	41.80%
DET (semantic)	**13257**	**459433**	**7303**	**48.90%**	**57.70%**	**41.81%**
NTU-MOTD Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
DET (IoU)	3617	1288	1859	21.79%	86.97%	30.88%
DET (cos)	3762	**536**	175	76.91%	91.38%	72.37%
DET (semantic)	**3354**	563	**170**	**79.67%**	**92.12%**	**73.99%**

7) TRACKING WITH DIFFERENT TRANSITION POLICIES OF FINITE-STATE MACHINE

To find a suitable transition policy for the finite-state machine used in our tracker as shown in Figure 9, we run our tracker under eight different transition policies on the NTU-MOTD dataset and choose the best policy as our final transition policy for the finite-state machine. Specifically, we change the counter value for each state transition action. For example, to change the state of an object from the *tentative* state to the *tracked* state, there need to be 3 continuous detection matches. As shown in Table 10, different policies will affect the performance related to tracking coverage, such as FP and FN, and the best policy among them is (3, 3, 1). This means that each track needs 3 continuous matches with detections to change from the *tentative* state to the *tracked* state, 3 continuous mismatches to change from the *tracked* state to the *lost* state, and only 1 match to change from the *lost* state to the *tracked* state. We fix the counter setting (3, 3, 1) for the finite-state machine in our tracker to conduct other experiments.

TABLE 10. The proposed tracker DET run with different transition policies for the underlying finite-state machine. S_1 stands for tentative state, S_2 stands for tracked state, and S_3 stands for lost state. The values for $S_1 \rightarrow S_2$ and $S_3 \rightarrow S_2$ are the numbers of continuous matches with detections. The value for $S_2 \rightarrow S_3$ is the number of continuous mismatches.

NTU-MOTD Dataset							
Transition Policy			FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
$S_1 \rightarrow S_2$	$S_2 \rightarrow S_3$	$S_3 \rightarrow S_2$						
1	3	1	4728	**444**	322	70.37%	89.41%	67.95%
1	1	1	3164	1394	365	70.74%	90.51%	67.97%
1	3	3	3203	765	308	70.27%	91.76%	68.25%
1	1	3	1758	1907	320	73.63%	92.32%	70.33%
3	3	3	1561	1454	175	73.80%	**93.85%**	70.76%
3	1	3	**992**	3098	**156**	75.58%	91.82%	70.93%
3	1	1	2101	2331	203	78.22%	91.07%	72.04%
3	**3**	**1**	3354	563	170	**79.67%**	92.12%	**73.99%**

8) TRAINING SCENE DETECTOR WITH DIFFERENT TYPES OF INPUT IMAGES

For the scene detector used in the scene-aware affinity measurement, we design the model architecture as shown in Table 11. The model extracts the feature representation from the input image with a feature backbone consisting of convolutional layers and outputs a 2-dimensional probability vector to indicate whether the input image is an outdoor scene or an indoor scene. We train the model with different types of input images. One is RGB images, and the other is grayscale depth maps. The numbers of training samples for indoor scenes and outdoor scenes are both 3000, and we evaluate the accuracy of the model on 1000 images, with 500 indoor images and 500 outdoor images. The images are all randomly sampled from the MOT16 training sequences and the NTU-MOTD dataset. We label the images from the NTU-MOTD dataset as indoor samples and the ones from MOT16 as outdoor samples, with the exception that some of the video sequences are in indoor environments. Finally, we train the model with 50 epochs, and the batch size is 64. The model is optimized with the Adam optimizer with a 0.001 learning rate. As shown in Table 12, the model can accurately predict the type of

TABLE 11. Overview of the scene detector model architecture. The last layer performs a further softmax operation to convert the final vector to a probability vector.

Layer Name	Batch Size/Stride	Output Size
Conv1	3 x 3 / 1	16 x 256 x 256
MaxPool1	2 x 2 / 2	16 x 128 x 128
Conv2	3 x 3 / 1	32 x 128 x 128
MaxPool2	2 x 2 / 2	32 x 64 x 64
Conv3	3 x 3 / 1	64 x 64 x 64
MaxPool3	2 x 2 / 2	64 x 32 x 32
Conv4	3 x 3 / 1	128 x 32 x 32
MaxPool4	2 x 2 / 2	128 x 16 x 16
Conv5	3 x 3 / 1	256 x 16 x 16
MaxPool5	2 x 2 / 2	256 x 8 x 8
AvgPool	8 x 8 / 1	256
Dense1		64
Dense2		2

TABLE 12. Classification accuracy of the scene detector trained with different input data types.

Scene Detector	
Input Data Type	Classification Accuracy
RGB Images	0.99
Grayscale Depth Maps	0.99

image with either RGB images or depth maps. Therefore, we choose depth maps as the input data to the scene detector to best reduce the background noise information in the image during inference.

9) TRACKING WITH OR WITHOUT SCENE-AWARE AFFINITY MEASUREMENT

To verify the effectiveness and robustness of the proposed scene-aware affinity measurement choosing 2D or 3D spatial information dynamically to compute affinity values between objects according to the type of scene, we run the tracker with three different settings. One restricts the tracker to using only 2D spatial information (x, y) to compute affinity values, and the other restricts the tracker to using only 3D spatial information (x, y, z) to compute affinity values. The last setting lets the tracker dynamically select 2D or 3D information to calculate the affinity value according to the type of processing frame. As shown in Table 13, we conduct experiments on the MOT17, MOT20, and NTU-MOTD datasets and include two 2D trackers in the table to show the contribution of 3D depth information as well. Our tracker can consistently perform best with the scene-aware affinity measurement in various scenarios, which demonstrates the robustness and effectiveness of the scene-aware affinity measurement. The reason the scene-aware affinity measurement is important to the tracker is that it can instruct the tracker to avoid using unreliable 3D depth information in outdoor scenes, as shown in Figure 14, and can switch to reliable 3D depth information in indoor scenes to mitigate position ambiguity and occlusion problems in 2D space.

TABLE 13. The proposed tracker DET uses different types of spatial information to compute affinity values between objects. 2D stands for (x, y) information, and 3D stands for (x, y, z) information. The dynamic method leverages the scene-aware affinity measurement to dynamically choose 2D or 3D information to perform the affinity measurement.

MOT17 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
SORT [36]	5370	190797	4266	37.79%	40.51%	35.58%
DeepSORT [33]	5486	**178291**	1103	49.36%	**45.12%**	42.58%
DET (2D)	3483	187237	827	51.15%	43.14%	43.48%
DET (3D)	3473	186879	842	51.90%	43.25%	43.78%
DET (Dynamic)	**3401**	187129	**799**	**51.91%**	43.21%	**43.85%**
MOT20 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
SORT [36]	**5297**	506584	8423	44.71%	54.14%	39.86%
DeepSORT [33]	8291	487153	8331	43.68%	55.60%	37.67%
DET (2D)	13257	459433	7303	48.90%	57.70%	41.81%
DET (3D)	13872	462786	8129	47.25%	57.27%	40.58%
DET (Dynamic)	13257	**459433**	**7303**	**48.90%**	**57.70%**	**41.81%**
NTU-MOTD Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
SORT [36]	**2238**	1843	2124	20.67%	88.04%	29.75%
DeepSORT [33]	2321	2011	213	66.49%	91.32%	66.49%
DET (2D)	3372	**535**	213	79.19%	92.06%	73.70%
DET (3D)	3354	563	170	79.67%	92.12%	73.99%
DET (Dynamic)	3354	563	**170**	**79.67%**	**92.12%**	**73.99%**

(a) Failure case 1. **(b)** Failure case 2.

FIGURE 14. Inaccurate depth maps in outdoor scenes.

10) COMPUTATIONAL COMPLEXITY OF THE PROPOSED TRACKER

To measure the computational efficiency and complexity of the tracker, we use the public ptflops [51] library to get the theoretical number of floating-point operations and the memory consumption of the deep learning models included in our tracker. As shown in Table 14, the total memory consumption of all models is less than 1 GB. The models can be loaded onto modern GPUs, such as RTX3090, at the same time to perform MOT tracking.

TABLE 14. Computational complexity and memory consumption of the deep learning models included in the proposed DET tracker.

Model Name	Input Size	Params (M)	FLOPs (G)	Complexity
MiDaS [38]	3x1920x1080	344.06	884.30	O(1)
PWCNet [40]	3x1920x1080	9.37	726.72	O(1)
Mask-RCNN [39]	3x1920x1080	44.18	390.86	O(n)
Person-reID [41]	3x128x256	23.77	8.16	O(n)
Scene Detector	3x256x256	0.41	0.68	O(1)
Total		421.79	2010.72	O(n)

E. MOT BENCHMARK EVALUATION

We summarize the final settings of our tracker in the following description. Our tracker uses MiDaS to perform depth estimation to obtain the depth map from the image frame. If the detection annotation provides object segmentation information, our tracker will use the segmentation masks of the objects to compute their depth values; otherwise, the tracker will adopt a random sampling method to determine the objects' depth values. The scene detector will classify the current processing frame as an indoor scene when the probability value is larger than 0.5. The matching thresholds for the matching functions in the semantic matching strategy are all 0.3. Any matching pair with a matching cost less than

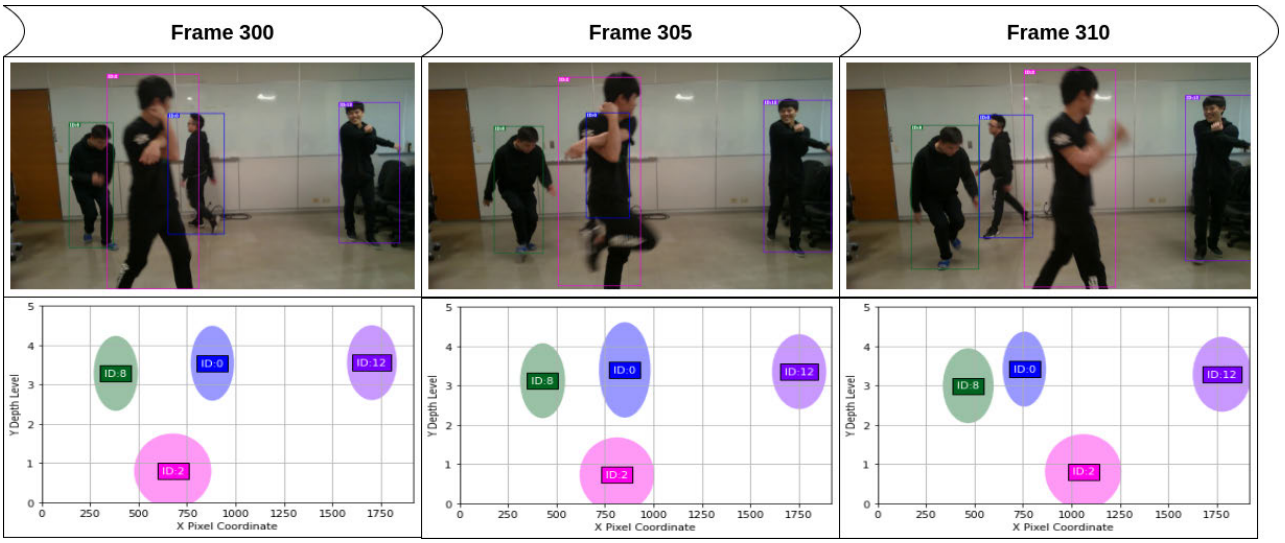


FIGURE 15. Runtime of DET performing tracking on NTU-MOTD dataset. Due to the discriminative 3D spatial information, DET can accurately track objects with similar appearances even when occlusion occurs between objects in frame 305.

TABLE 15. Benchmark evaluations on the MOT17, MOT20, and NTU-MOTD datasets. The first-place and second-place values are represented in red and blue, respectively.

MOT17 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
UMA [52]	12779	**157597**	**1005**	**62.51%**	**49.13%**	**49.36%**
JDE [53]	**7218**	185220	1418	**53.42%**	42.46%	43.56%
MOTDT [54]	3953	179460	1168	51.71%	**45.21%**	42.89%
SORT [36]	5370	190797	4266	37.79%	40.51%	35.58%
DeepSORT [33]	5486	**178291**	1103	49.36%	45.12%	42.58%
DET (ours)	**3401**	187129	**799**	51.91%	43.21%	**43.85%**
MOT20 Training Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
UMA [52]	17607	**431639**	9085	36.73%	**59.61%**	32.63%
JDE [53]	**5848**	543836	**6918**	**48.20%**	50.94%	**40.82%**
MOTDT [54]	12288	491970	7950	44.91%	54.86%	39.00%
SORT [36]	**5297**	506584	8423	44.71%	54.14%	39.86%
DeepSORT [33]	8291	487153	8331	43.68%	55.60%	37.67%
DET (ours)	13257	**459433**	**7303**	**48.90%**	**57.70%**	**41.81%**
NTU-MOTD Dataset						
Tracker	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
UMA [52]	6879	**1174**	1341	30.95%	81.90%	34.22%
JDE [53]	**2058**	1470	473	**76.48%**	**92.29%**	**69.34%**
MOTDT [54]	**2180**	3657	458	55.76%	87.87%	55.60%
SORT [36]	2238	1843	2124	20.67%	88.04%	29.75%
DeepSORT [33]	2321	2011	**213**	66.49%	91.32%	66.49%
DET (ours)	3354	**563**	**170**	**79.67%**	**92.12%**	**73.99%**

0.3 will be considered a valid matching candidate. The feature extraction module is ResNet50 [41] pretrained on the MOT16 training dataset with metric learning to extract the feature representations of objects. The state transition policy of the finite state machine for all tracks is that 3 continuous matches will change the state of a track from *tentative* to *tracked*, 3 continuous mismatches will change the state of a track from *tracked* to *lost*, and 1 match will immediately change the state of a track from *lost* to *tracked*.

We compare our tracker with several state-of-the-art online trackers on the latest MOT benchmarks, MOT17 and MOT20, and the proposed NTU-MOTD dataset. For the MOT17 and MOT20 benchmark evaluations, we use the public detection annotations and uniformly filter out detections with confidence scores less than 0.8 in MO17 and 0.0 in MOT20 for all trackers. For the proposed NTU-MOTD dataset, we use the detection results along with the segmentation masks generated by Mask-RCNN and uniformly filter out detections with confidence scores less than 0.8. The experimental results are organized in Table 15, and the qualitative results of our tracker are shown in Figure 16. Except for the MOT17 benchmark, on which the UMA tracker overfits, our tracker consistently beats the other trackers on a convincing metric, HOTA, across the datasets. Since the depth estimation model is more precise and accurate in indoor environments, the performance improvement of our tracker is more significant on the NTU-MOTD dataset than on MOT17 and MOT20, which are mostly outdoor video sequences.

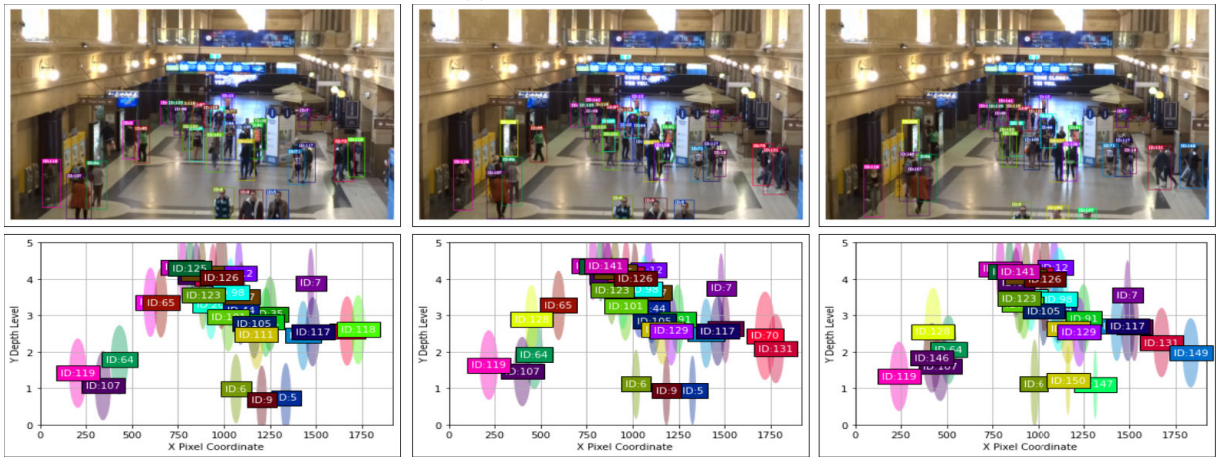
To make the performance comparison of the trackers in Table 15 more intuitive, we rank all trackers based on their HOTA values and sum the ranks of each tracker across the datasets to represent their final performance. As shown in Table 16, our tracker has the best total rank among all the trackers. This demonstrates the robustness and competitiveness of our tracker in various tracking scenarios.

To identify the difficult cases for our tracker in indoor tracking environments, we inspect the tracking result of each video sequence in Table 17. We consider the cases with HOTA values less than 80% as difficult cases, and mark their

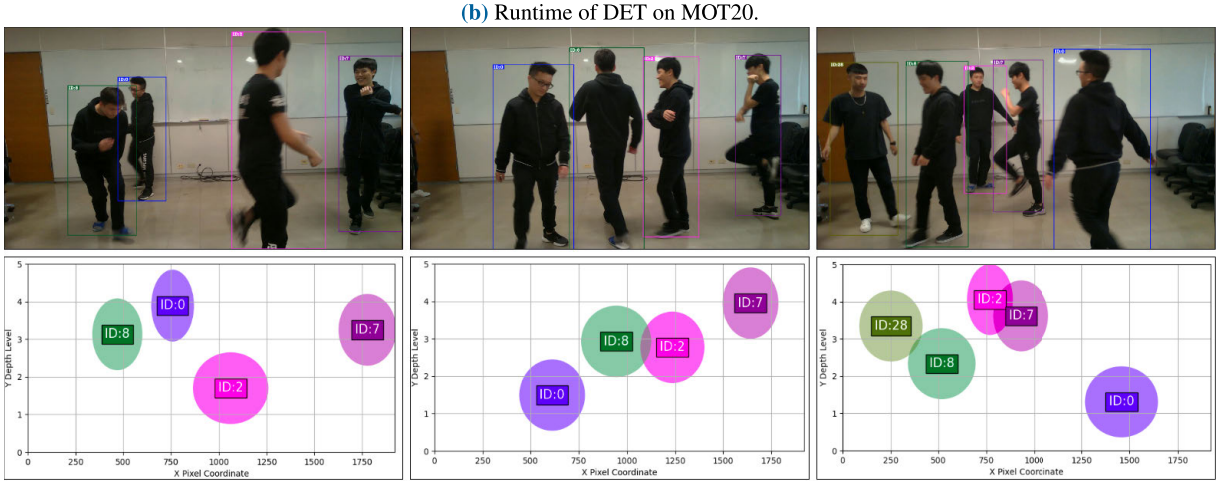
TABLE 16. MOT benchmark evaluations on the MOT17, MOT20, and NTU-MOTD datasets. The best value is presented in red.

Rank Table for MOT Benchmarks				
Tracker	MOT17 ↓	MOT20 ↓	NTU-MOTD ↓	Total Rank ↓
UMA [52]	**1**	6	5	12
JDE [53]	3	2	2	7
MOTDT [54]	4	4	4	12
SORT [36]	6	3	6	15
DeepSORT [33]	5	5	3	13
DET (ours)	2	**1**	**1**	**4**

(a) Runtime of DET on MOT17.



(b) Runtime of DET on MOT20.



(c) Runtime of DET on NTU-MOTD.

FIGURE 16. Qualitative results of DET on MOT17, MOT20, and NTU-MOTD. The top row shows the runtime result on MOT17, the middle row shows the result on MOT20, and the bottom row shows the result on NTU-MOTD.

names in red. In general, our tracker can perform tracking well, with very few identity switches in most cases. As shown in Figure 15, even when occlusion occurs between objects with similar appearances, the proposed tracker can still track objects accurately most of the time. However, under some difficult tracking scenarios in which crowded people with similar appearances move around dynamically, our tracker still has room for improvement, as shown by the lines with sequence names in red in Table 17.

We believe that a nonlinear motion model can be adopted to better deal with the dynamics of objects. Instead of the Kalman filter, the interacting multiple model (IMM)

TABLE 17. Detailed evaluation results of DET on the NTU-MOTD dataset. The sequence names in red represent difficult cases for DET. The tracking scenario of a video sequence is indicated by its name. For example, 5p_sa_um_up denotes 5 people with similar appearances moving around unpredictably with unpredictable pose changes.

NTU-MOTD Dataset						
Sequence	FP ↓	FN ↓	IDs ↓	IDF1 ↑	MOTA ↑	HOTA ↑
3p_da_pm_pp	62	16	0	97.59%	95.11%	84.04%
3p_da_pm_up	43	27	0	97.86%	95.70%	83.74%
3p_da_pm_up	60	19	8	95.83%	94.86%	82.01%
3p_da_um_up	60	9	2	97.27%	94.86%	82.73%
3p_sa_pm_pp	41	18	0	98.17%	96.31%	84.95%
3p_sa_pm_up	56	21	0	97.57%	95.08%	82.28%
3p_sa_um_pp	51	16	6	87.05%	95.35%	78.24%
3p_sa_um_up	59	13	0	97.64%	95.21%	83.36%
5p_da_pm_pp	100	58	2	96.21%	92.54%	80.27%
5p_da_pm_up	100	42	0	96.63%	93.17%	79.92%
5p_da_um_pp	96	35	6	87.47%	93.56%	76.37%
5p_da_um_up	170	46	10	93.87%	89.11%	77.85%
5p_sa_pm_pp	192	48	0	94.65%	88.94%	77.46%
5p_sa_pm_up	196	73	10	73.26%	86.73%	65.74%
5p_sa_um_pp	160	40	8	74.58%	90.49%	65.63%
5p_sa_um_up	150	39	12	65.29%	90.19%	59.81%
3p_da_40sec	190	7	8	96.49%	93.87%	90.44%
3p_sa_40sec	139	7	13	80.50%	95.28%	80.79%
5p_da_60sec	735	15	34	56.89%	90.56%	60.27%
5p_sa_60sec	694	14	51	57.89%	89.81%	59.28%

filter [55] or a deep learning approach might serve as a better motion model for tracking dynamic objects. To better address pose changes, training a pose-invariant ReID model with additional body keypoints acting as supervision signals should reduce the impact of the intraclass variance associated with the same person in different poses.

VI. LIMITATIONS AND FUTURE WORK

The proposed NTU-MOTD currently only provides video sequences with 3 to 5 people. Crowded scenes are not included in the dataset. To accommodate more people when recording the video, the camera needs to be placed at a higher viewpoint. In future work, we will design more complex tracking scenarios and collect video sequences in crowded scenes.

Under crowded scenes, we believe trackers follow with the tracking-by-detection framework will have a significant performance drop, and the processing speed will become very slow due to a tremendous amount of objects in the crowded scene. To speed up the processing speed of our tracker in crowded scenes, we will design a model with multitask learning to jointly extract essential information, such as detections, ReID features, and depth map, in a single model forward pass, and train the model on crowded tracking datasets.

VII. CONCLUSION

We introduce an indoor tracking dataset showing a variety of indoor tracking scenarios and increasing the diversity of the existing MOT benchmark dataset. Severe occlusion and homogeneous appearance problems are fully reflected in the proposed dataset. We believe the proposed dataset will enable future research to solve indoor MOT tracking problems.

We propose a depth-enhanced tracking-by-detection framework and a semantic matching strategy combined with scene-aware affinity measurement to effectively mitigate the severe occlusion and homogeneous appearance problems in indoor tracking. We demonstrate the effectiveness and generality of the proposed method on both indoor and outdoor benchmark datasets. The experimental results highlight the importance of considering depth information in tracking tasks. We hope that our work will open the door for future works to incorporate depth information in performing tracking and to design all-in-one training frameworks to learn tracking problems with multiple supervision signals, such as depth maps, bounding boxes, and flow maps, rather than using separate dedicated models to perform inference, which adds additional computational complexity.

REFERENCES

[1] R. T. Collins, A. J. Lipton, T. Kanade, and H. Fujiyoshi, "A system for video surveillance and monitoring," *VSAM Final Rep.*, vol. 2000, nos. 1–68, p. 1, 2000.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[3] T. Zhang, B. Ghanem, and N. Ahuja, "Robust multi-object tracking via cross-domain contextual information for sports video analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 985–988.

[4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[5] C. Ma, Y. Li, F. Yang, Z. Zhang, Y. Zhuang, H. Jia, and X. Xie, "Deep association: End-to-end graph-based learning for multiple object tracking with conv-graph neural network," in *Proc. Int. Conf. Multimedia Retr.*, 2019, pp. 253–261.

[6] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, Jan. 2021.

[7] P. Chu and H. Ling, "FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6172–6181.

[8] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[10] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Understand.*, vol. 193, Apr. 2020, Art. no. 102907.

[11] F. Lourenço and H. Araujo, "Intel realsense SR305, D415 and l515: Experimental evaluation and comparison of depth estimation," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 362–369.

[12] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proc. 12th IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, Dec. 2009, pp. 1–6.

[13] P. Zhu *et al.*, "VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, pp. 496–518.

[14] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 430–446.

[15] 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," 2020, *arXiv:2003.09003*.

[16] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, and B. Caine, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2446–2454.

[17] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.
[18] W. Brendel, M. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *Proc. CVPR*, Jun. 2011, pp. 1273–1280.
[19] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*, vol. 271. Springer, 2014, pp. 67–70.
[20] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[21] G. Braso and L. Leal-Taixe, "Learning a neural solver for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6247–6257.
[22] L. R. Ford, "Network flow theory," RAND Corp., Santa Monica, CA, USA, Tech. Rep., 1956. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Network+flow+theory&btnG=
[23] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.
[24] R. Jonker and T. Volgenant, "Improving the Hungarian assignment algorithm," *Oper. Res. Lett.*, vol. 5, no. 4, pp. 171–175, Oct. 1986.
[25] I. Papakis, A. Sarkar, and A. Karpatne, "GCNNMatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization," 2020, *arXiv:2010.00067*.
[26] H. Lee, I. Kim, and D. Kim, "VAN: Versatile affinity network for end-to-end online multi-object tracking," in *Proc. Asian Conf. Comput. Vis.*, 2020. [Online]. Available: https://openaccess.thecvf.com/content/ACCV2020/papers/Lee_VAN_Versatile_Affinity_Network_for_End-to-end_Online_Multi-Object_Tracking_ACCV_2020_paper.pdf
[27] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.
[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 21–37.
[29] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.
[31] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.
[32] R. J. Meinhold and N. D. Singpurwalla, "Understanding the Kalman filter," *Amer. Statistician*, vol. 37, no. 2, pp. 123–127, Oct. 1981.
[33] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
[34] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang, "Multi-object tracking with multiple cues and switcher-aware classification," 2019, *arXiv:1901.06129*.
[35] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4836–4845.
[36] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
[37] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin, B. Hoff, D. Kruchinin, A. Zankevich, DmitriySidnev, M. Markelov, M. Chenuet, A. Melnikov, J. Kim, L. Ilouz, N. Glazov, R. Tehrani, S. Jeong, V. Skubriev, S. Yonekura, and T. Truong. (Aug. 2020). *OpenCV/CVat: V1.1.0*. [Online]. Available: https://doi.org/10.5281/zenodo.4009388
[38] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2019, *arXiv:1907.01341*.
[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "MasK R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.
[40] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[42] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[43] J. Luiten, A. Ošep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, Feb. 2021.
[44] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Proc. 6th IEEE Int. Workshop Vis. Surveill., Conjunct. (ECCV)*, May 2006, vol. 90, no. 91, pp. 1–8.
[45] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 17–35.
[46] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 437–446.
[47] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 36–42.
[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 740–755.
[49] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9685–9694.
[50] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166.
[51] V. Sovrasov. (2019). *Flops Counter for Convolutional Networks in Pytorch Framework*. [Online]. Available: https://github.com/sovrasov/flops-counter.pytorch/
[52] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6768–6777.
[53] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K.: Springer, Aug. 2020, pp. 107–122.
[54] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[55] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.

CHENG-JEN LIU received the B.S. degree in computer science from the National Taiwan Cheng Kung University, Tainan, Taiwan, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering with the National Taiwan University, Taipei, Taiwan. He is also an Embedded Software Intern responsible for power management at Nvidia Corporation. His research interests include computer vision, multiple-object tracking, and embedded systems.

TSUNG-NAN LIN (Senior Member, IEEE) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1989, and the M.S. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1993 and 1996, respectively. He joined EPSON Research and Development, Inc., San Jose, CA, USA, and EMC Corporation, Hopkinton, MA, USA. Since February 2002, he has been with the Department of Electrical Engineering, Graduate Institute of Communication Engineering, National Taiwan University. He was the Director of the Division of Network Management, Computer and Information Networking Center, National Taiwan University, and the Vice President and the Director General of the Cybersecurity Technology Institute, Institute for Information Industry. He is a member of the Phi Tau Phi Scholastic Honor Society.