GRASPTrack: Geometry-Reasoned Association via Segmentation and Projection for Multi-Object Tracking

Xudong Han*, Pengcheng Fang*, Yueying Tian, Jianhui Yu, Xiaohao Cai, Daniel Roggen, Philip Birch †,

***Abstract*—Multi-object tracking (MOT) in monocular videos is fundamentally challenged by occlusions and depth ambiguity, issues that conventional tracking-by-detection (TBD) methods struggle to resolve owing to a lack of geometric awareness. To address these limitations, we introduce GRASPTrack, a novel depth-aware MOT framework that integrates monocular depth estimation and instance segmentation into a standard TBD pipeline to generate high-fidelity 3D point clouds from 2D detections, thereby enabling explicit 3D geometric reasoning. These 3D point clouds are then voxelized to enable a precise and robust Voxel-Based 3D Intersection-over-Union (IoU) for spatial association. To further enhance tracking robustness, our approach incorporates Depth-aware Adaptive Noise Compensation, which dynamically adjusts the Kalman filter process noise based on occlusion severity for more reliable state estimation. Additionally, we propose a Depth-enhanced Observation-Centric Momentum, which extends the motion direction consistency from the image plane into 3D space to improve motion-based association cues, particularly for objects with complex trajectories. Extensive experiments on the MOT17, MOT20, and DanceTrack benchmarks demonstrate that our method achieves competitive performance, significantly improving tracking robustness in complex scenes with frequent occlusions and intricate motion patterns.**

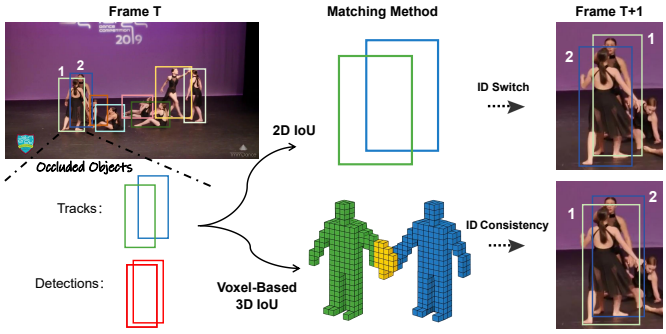

Fig. 1: An illustration of associating occluded detections in crowded scenes. In the presence of heavy occlusion, conventional 2D IoU-based matching can lead to ID switches owing to spatial ambiguity between overlapping objects. To address this, we propose a Voxel-Based IoU metric that operates in 3D space, enabling more accurate association by capturing fine-grained volumetric overlap and handling partial occlusions with improved spatial reasoning.

I. Introduction

Multi-object tracking (MOT) is a critical task in computer vision with many applications, such as autonomous driving [1], robotic navigation [2], and sports analytics [3]. Most MOT methods typically follow the tracking-by-detection (TBD) paradigm, where objects are detected independently in each frame and associated across frames based on motion and appearance cues. These MOT methods typically rely on 2D bounding box detection and frame-wise association through metrics such as the Intersection-over-Union (IoU). Despite their efficiency, these approaches inherently lack geometric awareness, making them vulnerable to object interactions, depth ambiguity, and occlusions.

Current MOT methods face several challenges in real-world scenarios. One critical problem is the occlusion. When multiple objects at different depths overlap in the 2D image plane, even short-term partial occlusions can result in heavy overlap, leading to identity switches that IoU-based matching struggles to resolve. Another significant challenge is accurately modelling motion. For instance, objects moving along the optical axis of the camera may undergo substantial 3D motion with minimal 2D positional changes, leading to erroneous velocity estimates and association failures. To mitigate such issues, several existing works [4], [5] attempt to infer pseudo-depth from 2D cues. However, these methods rely on strong scene assumptions and typically produce imprecise depth estimations. In addition, other methods [6], [7] utilize a monocular depth estimation model to obtain depth maps, but typically extract 3D features from the entire 2D bounding box. This process introduces significant noise from the background and even occluding objects, degrading the quality of the object's 3D representation.

To address these limitations, this study proposes a depth-aware MOT framework that explicitly incorporates geometric reasoning into the tracking pipeline, called GRASPTrack. Our approach leverages advanced models in monocular depth estimation and segmentation to enrich scene understanding from a single image. Specifically, we use a segmentation model to generate a precise instance mask for each object. This mask guides the creation of a clean, high-fidelity 3D point cloud from the dense depth map produced by a monocular depth estimation model. To enhance spatial matching, these point clouds are transformed into voxel representations, enabling a Voxel-Based 3D IoU for robust association and better reflecting their true spatial extent.

Additionally, we enhance motion modeling in the presence of occlusions. Traditional Kalman Filters [8] rely on fixed

* Equal Contribution.
† Corresponding author.

process noise assumptions, which fail to adapt to the increased uncertainty introduced by occlusions. We propose a Depth-aware Adaptive Noise Compensation (DANC) method that dynamically adjusts the process noise covariance in the Kalman filter based on the severity of occlusion, ensuring more conservative and reliable state updates under uncertainty. Furthermore, the Observation-Centric Momentum (OCM) introduced in OC-SORT [9] leverages motion direction consistency to improve association robustness. We introduce a Depth-enhanced Observation-Centric Momentum (DOCM) to extend motion direction consistency modeling from 2D to 3D space. By calculating the motion direction consistency using full 3D state vectors, our method provides a more robust motion cue, leading to more reliable data association. We evaluate our method on several challenging datasets, such as MOT17 [10], MOT20 [11] and DanceTrack [12]. Experimental results show that our method achieves highly competitive performance among tracking-by-detection methods.

The main contributions of this study are as follows:

- We propose GRASPTrack, a novel depth-aware MOT framework that integrates geometric reasoning into the tracking pipeline, significantly enhancing robustness under occlusion. We leverage monocular depth estimation and segmentation masks to reconstruct high-fidelity 3D point clouds from 2D detection. These are voxelized to enable Voxel-Based 3D IoU for object association, while mask-guided refinement effectively suppresses background and occluder noise.
- We introduce DANC, a dynamic Kalman filter process noise adjustment mechanism that accounts for occlusion severity. In addition, we extend the Kalman filter state vector using depth information to enable accurate spatial state estimation in 3D space.
- We propose DOCM to extend the motion direction consistency in 3D space, improving motion-based association under complex scenarios.
- Extensive experimental results and comparison are conducted on challenging benchmark datasets.

II. Background and Related Work

A. Tracking by Detection

Many current multi-object tracking methods follow the TBD paradigm [9], [13]–[15]. These methods use a detector to detect objects in each frame and associate them across various frames. Early TBD methods, such as SORT [13], relied on the Kalman Filter for motion prediction and the IoU between predicted and detected bounding boxes for association. DeepSORT [16] introduced a ReID-based appearance similarity in the cost matrix to enhance robustness and handle longer-term occlusions where the IoU would fail. ByteTrack [14] introduced a simple and effective heuristic for associating low-confidence detections separately to recover objects during occlusion. OCSORT [9] enhanced the robustness of handling occlusions by improving the linear motion assumption in the Kalman filter. Deep OC-SORT [17] integrated appearance features and camera motion compensation. UCMCTrack [18] proposed a method that handles camera motion in object tracking by replacing the standard IoU metric with a Mapped Mahalanobis Distance on the ground plane. TBD methods have shown that the combination of a strong detector with a simple association strategy can yield competitive tracking performance. Therefore, we chose to follow the TBD paradigm in this study.

B. Depth Information in MOT

Adding depth information as a form of spatial context is a key strategy for making multi-object tracking more robust, particularly in crowded scenes. In the domain of 3D MOT, trackers such as AB3DMOT [19] and CenterPoint [20] leverage explicit 3D sensors, such as LiDAR, to track objects in true 3D space. However, these approaches depend on specialized and costly hardware, which restricts their widespread application. This has motivated the development of methods that can infer 3D information from a more accessible single 2D image, which implicitly contains depth cues through perspective projection. Approaches using a single camera have largely followed two directions. The first uses pseudo-depth heuristics to infer a relative depth order from an object's position in the 2D frame. SparseTrack [4] leveraged pseudo-depth to separate objects along the depth axis and divided the detected objects into multiple sparse subsets at different depths. CAMOT [5] incorporated a pseudo-depth state directly into its Kalman filter. The second direction involves the use of a monocular depth estimation model to generate a depth map. QuoVadis [6] used these maps to create a bird's-eye view (BEV) representation for forecasting. However, these prior studies are limited because they either relied on coarse geometric heuristics or used depth information merely as an auxiliary cue to improve tracking performance. In this study, we propose a more robust and holistic integration of 3D geometric reasoning by integrating more precise depth information to enhance the robustness of the tracker in complex and occluded scenes.

III. Method

GRASPTrack enhances the TBD paradigm with a depth-aware framework composed of three main components. We first introduce a Depth-Aware Voxelization and 3D IoU Computation module, which converts segmented depth maps into voxel grids for geometric matching. This is followed by a DANC module that incorporates depth cues into state prediction. Finally, a DOCM module models motion consistency in 3D space. All components are coherently designed around depth, forming a fully integrated framework for depth-aware multi-object tracking.

A. Depth-Aware Voxelization and 3D IoU

As illustrated in Figure 2(c), GRASPTrack recovers accurate 3D spatial representations of objects from monocular RGB images. The monocular image is fed into two foundational models in parallel: Depth Anything v2 [21], which performs high-quality depth estimation with enhanced cross-scene generalization and improved reconstruction of fine-grained depth details, and EfficientTAM [22], which generates segmentation

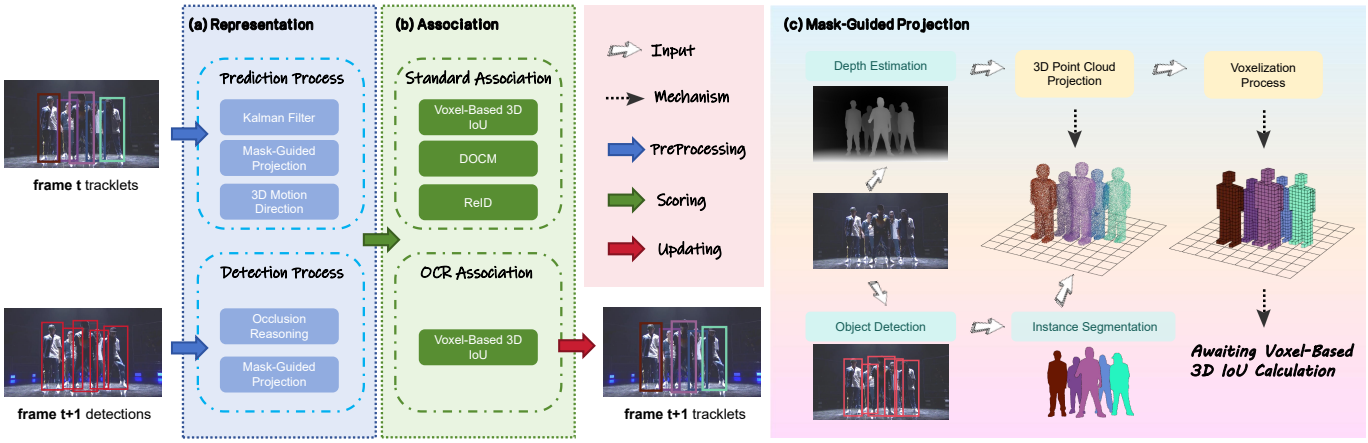

Fig. 2: The pipeline of the proposed GRASPTrack. **(a)** Representation: We estimate the probable location of each tracked object in the current frame by leveraging a Kalman filter. To reconstruct the 3D geometry of an object, we apply a mask-guided projection that combines depth estimation and instance segmentation cues. To facilitate more accurate and efficient association, the resulting point cloud is voxelized to enable voxel-based 3D reasoning. For each detection, we first assess its occlusion state and accordingly adapt the Kalman filter's noise covariance in subsequent frames; **(b)** Association: We compute the Voxel-Based 3D IoU and DOCM (Depth-enhanced Observation-Centric Momentum) to capture geometric similarity, while appearance similarity is measured using a ReID model. In the OCR association stage, only the Voxel-Based 3D IoU is employed; and **(c)** Process of our Mask-Guided Projection.

masks for the objects using box prompts. For each input frame $I_t \in \mathbb{R}^{H\times W\times 3}$, we estimate a dense depth map $D_t \in \mathbb{R}^{H\times W}$ using a monocular depth estimation network as follows:

$$D_t = f_{\text{depth}}(I_t), \tag{1}$$

where $f_{\text{depth}}(\cdot)$ denotes the depth estimation model.

Given a set of bounding boxes $\mathcal{B}_t = \{b_i^t\}_{i=1}^{N_t}$ from both the detector and tracker, with each $b_i^t = [x_1^i, y_1^i, x_2^i, y_2^i, s_i^t]$, we obtain the corresponding binary segmentation masks using EfficientTAM, i.e.,

$$M_i^t = f_{\text{seg}}(I_t, b_i^t), \quad M_i^t \in \{0,1\}^{H\times W}, \tag{2}$$

where $M_i^t(u,v) = 1$ indicates that pixel (u,v) belongs to object i at time t.

1) Mask-Guided Projection: Using the estimated depth map D_t and mask M_i^t, we reconstruct a per-object 3D point cloud by projecting pixels within the mask region into camera coordinates using the standard camera model. For each pixel within the segmentation mask, the 3D coordinates are computed using the standard pinhole camera projection equations:

$$Z = D_t(u,v),\ X = \frac{(u-c_x)Z}{f_x},\ Y = \frac{(v-c_y)Z}{f_y}, \tag{3}$$

where (u,v) are pixel coordinates of the projection plane, Z is the depth value, (c_x, c_y) is the center point of the box corresponding to the object, and (f_x, f_y) are the focal lengths in the x and y directions, respectively. For each object i at time t, we construct a 3D point cloud by collecting all valid projected points within its segmentation mask, i.e.,

$$\mathcal{P}_i^t = \{\mathbf{p} = [X,Y,Z]^\top \mid M_i^t(u,v) = 1, D_t(u,v) > 0\}, \tag{4}$$

where $M_i^t(u,v) = 1$ indicates pixels within the object's segmentation mask, and $D_t(u,v) > 0$ ensures valid depth values. This formulation ensures that only valid depth values within the precise segmentation boundary of the object are considered, providing a more accurate 3D representation than using entire bounding boxes. This mask-guided projection eliminates the background and occluder pixels, ensuring that only valid object regions contribute to the 3D geometry.

2) Voxelization Process: While 3D point clouds $\mathcal{P}_i^t$ offer fine-grained geometric details, traditional 3D IoU computations typically rely on fitting coarse 3D bounding boxes, which fail to capture the true object shape [23]. To better preserve geometric fidelity while enabling efficient pairwise comparison, we adopt a voxel-based representation that discretizes each $\mathcal{P}_i^t$ into a binary occupancy grid. This allows us to compute the 3D Intersection-over-Union (IoU) directly on the volumetric shape, yielding a more accurate and robust similarity metric. Unlike the voxelization adopted in detection frameworks [24], which is used solely for feature extraction before regressing a bounding box, our voxel grid is used exclusively during evaluation. Each voxel stores a single binary occupancy value and does not participate in network training or inference.

To ensure consistent voxelization across different frames and object pairs, we establish a unified 3D coordinate system. Given two sets of 3D point clouds $\mathcal{P}_i^{\text{det}}$ and $\mathcal{P}_j^{\text{trk}}$ representing detections and tracks respectively, we compute the overall spatial bounds:

$$\mathbf{p}_{\min} = \min\Big(\min_{\mathbf{p}\in\mathcal{P}_i^{\text{det}}} \mathbf{p}, \min_{\mathbf{p}\in\mathcal{P}_j^{\text{trk}}} \mathbf{p}\Big), \tag{5}$$

$$\mathbf{p}_{\max} = \max\Big(\max_{\mathbf{p}\in\mathcal{P}_i^{\text{det}}} \mathbf{p}, \max_{\mathbf{p}\in\mathcal{P}_j^{\text{trk}}} \mathbf{p}\Big), \tag{6}$$

where $\mathbf{p}_{\min}, \mathbf{p}_{\max} \in \mathbb{R}^3$ define the global 3D bounding volume that encompasses both point clouds. This way ensures that all the point clouds share the same voxel coordinate.

We discretize the continuous 3D space into a regular voxel grid using a voxel size parameter δ_v, which determines the spatial resolution of the discretization. The voxel size δ_v controls the fundamental trade-off between computational efficiency and spatial precision, smaller values provide finer granularity but increase memory usage and computation time. In our implementation, we set $\delta_v = 0.4$ to balance the accuracy and efficiency for typical object scales in multi-object tracking scenarios.

For each point cloud $\mathcal{P}$, we transform the 3D coordinates into discrete voxel indices as follows:

$$\mathbf{v}(\mathbf{p}) = \left\lfloor \frac{\mathbf{p} - \mathbf{p}_{\min}}{\delta_v} \right\rfloor, \tag{7}$$

where $\mathbf{p} = [x, y, z]^\top$ is a 3D point, and $\mathbf{v}(\mathbf{p})$ represents the corresponding voxel index. To ensure valid indices, we apply boundary constraints to keep all indices within the computed grid dimensions. Since multiple points may map to the same voxel, we perform de-duplication by retaining only unique voxel indices.

We create a sparse binary occupancy grid $\mathcal{V} \in \{0,1\}^{N_x \times N_y \times N_z}$ where each voxel is marked as occupied if it contains at least one point from the object. This sparse representation is crucial for computational efficiency because typical object point clouds occupy only a small fraction of the total voxel space. The resulting occupancy grid provides a discretized volumetric representation that captures the essential 3D structure of each object while enabling efficient intersection and union operations for the IoU computation. Building upon this sparse volumetric encoding, we next describe how 3D IoU is efficiently computed between voxelized objects.

3) Voxel-Based 3D IoU Computation: Given two voxelized occupancy grids $\mathcal{V}_i$ and $\mathcal{V}_j$ representing objects i and j respectively, we compute the 3D IoU between objects, as illustrated in Figure 1, following the standard intersection-over-union formulation adapted to voxelized volumes, i.e.,

$$\text{IoU}_{3D}(\mathcal{V}_i, \mathcal{V}_j) = \frac{|\mathcal{V}_i \cap \mathcal{V}_j|}{|\mathcal{V}_i \cup \mathcal{V}_j|}. \tag{8}$$

The intersection $|\mathcal{V}_i \cap \mathcal{V}_j|$ counts the number of voxels occupied in both grids, which is computed through element-wise logical `AND` operations across all voxel positions. Similarly, the union $|\mathcal{V}_i \cup \mathcal{V}_j|$ counts the voxels occupied in either grid, obtained through element-wise logical `OR` operations. This voxel-based IoU computation provides several advantages over the traditional 2D IoU. First, it captures precise volumetric overlap rather than only projected area overlap, making it robust to changes in the viewpoint and camera motion. Second, it naturally handles complex object shapes and partial occlusions by considering the 3D spatial occupancy.

B. Depth-Aware Adaptive Noise Compensation

The traditional KF in current MOT methods [9], [13]–[15] use a fixed process noise parameter, which limits the robustness of the tracking algorithm under occlusion and geometric ambiguity. Occluded objects may exhibit unpredictable motion patterns that are not captured by simple constant velocity models. To enhance tracking performance in such challenging conditions, we propose the DANC, which dynamically adjusts process noise parameters.

1) Extended State Representation.: We extend the Kalman filter state vector to incorporate the object depth and its velocity, enabling 3D-aware motion modeling:

$$\mathbf{x}_t = [x, y, s, r, d, \dot{x}, \dot{y}, \dot{s}, \dot{d}]^\top, \tag{9}$$

where (x, y) denotes the object center in image coordinates, s is the object area, r is the aspect ratio, and d is the estimated object depth. The terms $(\dot{x}, \dot{y}, \dot{s}, \dot{d})$ represent the respective velocities. The depth value d is obtained by first employing EfficientTAM to generate precise object segmentation masks within detection bounding boxes, and then computing the average depth from the corresponding segmented regions in the depth map provided by Depth Anything v2, ensuring accurate depth representation that focuses solely on the object's actual geometry rather than background interference. The extended state representation enables a depth-aware adjustment of the Kalman noise to maintain stable predictions when the targets approach or recede rapidly.

2) Occlusion Status Determination: We dynamically adjust process noise covariance based on the occlusion level of the detected object. When an object is occluded, the reliability of both its motion model and measurements decreases, increasing the uncertainty in the Kalman filter. Let $\mathcal{D} = \{1, 2, ..., N\}$ represent all detections in the current frame, where N is the total number of detections. To determine whether an object $i \in \mathcal{D}$ is occluded, the IoU is calculated between i and all other objects $j \in \mathcal{D} \setminus \{i\}$. The occlusion status is determined using a depth-based criterion:

$$\text{occ}(i) = \begin{cases} \text{True}, & \text{if } \exists j \in \mathcal{D} \setminus \{i\} : \text{IoU}(b_i, b_j) \\ & \quad > \tau_{\text{IoU}} \text{ and } d_i > d_j, \\ \text{False}, & \text{otherwise}, \end{cases} \tag{10}$$

where b_i and b_j are the bounding boxes of objects i and j respectively, d_i and d_j are their corresponding depth values, and τ_{IoU} is the spatial overlap threshold. This process ensures that object i is evaluated against all other objects in the frame to comprehensively detect the occlusion scenarios.

3) Adaptive Noise Scaling: For occluded objects, we adaptively scale the process noise to account for the increased uncertainty. We compute the occlusion score $\mathcal{O}_i$ as the maximum IoU overlap with all occluding objects:

$$\mathcal{O}_i = \max_{j \in \mathcal{D} \setminus \{i\}} \{\text{IoU}(b_i, b_j) \mid \text{occ}(i) = \text{True}\}. \tag{11}$$

The adaptive noise scaling λ_i is then determined based on the occlusion strength:

$$\lambda_i = \begin{cases} 1 + \alpha \times \mathcal{O}_i, & \text{if occ}(i) = \text{True}, \\ 1, & \text{otherwise}, \end{cases} \tag{12}$$

where α is the occlusion sensitivity factor that controls the amplification intensity of noise scaling in response to occlusion severity. Therefore, the process noise covariance is adjusted accordingly as follows:

$$\mathbf{Q}_t^{\text{adaptive}} = \lambda_i \cdot \mathbf{Q}_{\text{base}}, \tag{13}$$

Tracker	MOT17				MOT20			
	HOTA↑	IDF1↑	MOTA↑	AssA↑	HOTA↑	IDF1↑	MOTA↑	AssA↑
Motion:								
ByteTrack [14]	63.1	77.3	80.3	62.0	61.3	75.2	77.8	59.6
C-BIoU [25]	64.1	79.7	81.1	63.7	-	-	-	-
MotionTrack [26]	65.1	80.1	**81.1**	65.1	62.8	76.5	78.0	61.8
OC-SORT [9]	63.2	77.5	78.0	63.4	62.4	76.3	75.7	62.5
SparseTrack [4]	65.1	80.1	81.0	65.1	63.4	77.3	**78.2**	62.8
UCMCTrack [18]	65.8	81.1	80.5	66.6	62.8	77.4	75.7	63.4
Motion & Appearance:								
Quo Vadis [6]	63.1	77.7	80.3	62.1	61.5	75.7	77.8	59.9
Bot-SORT [27]	65.0	80.2	80.5	65.5	63.3	77.5	77.8	62.9
GHOST [28]	62.8	77.1	78.7	-	61.2	75.2	73.7	-
StrongSORT [29]	64.4	79.5	79.6	64.4	62.6	77.0	73.8	64.0
Deep OCSORT [17]	64.9	80.6	79.4	65.9	63.9	79.2	75.6	65.7
DiffMOT [30]	64.5	79.3	79.8	64.6	61.7	74.9	76.7	60.5
OFTrack [31]	64.1	78.8	80.1	63.3	63.4	76.9	75.6	62.7
GRASPTrack	**66.1**	**81.7**	80.4	**66.9**	**64.5**	**80.1**	77.5	**66.1**

TABLE I: Performance comparison on the MOT17 & MOT20 test set. The best results are shown in bold.

where $\mathbf{Q}_{\text{base}}$ denotes the default noise. This mechanism ensures more conservative updates in the presence of occlusions. Multiplying the process covariance by the scale factor λ_i deliberately widens the predicted uncertainty, boosting the Kalman gain so that fresh measurements dominate whenever an object is occluded. Because λ_i grows linearly with the occlusion score, the filter shifts smoothly from normal confidence to a more cautious mode under heavy occlusion, all without retuning the base noise matrix.

C. Depth-Enhanced Observation-Centric Momentum.

The OCM introduced in OC-SORT considers the motion direction consistency modeling of an object in the association. The original OCM calculates motion direction angles using 2D center coordinates, where the angle θ is computed as $\theta = \arctan(\frac{v_j - v_i}{u_j - u_i})$ for two points (u_i, v_i) and (u_j, v_j) representing object center coordinates at different time steps. Although effective in 2D scenarios, this approach cannot adequately model motion consistency when depth variations are significant. However, the OCM only relies on the velocity direction of an object in the 2D image plane and fails to capture depth-related motion consistency, particularly when objects exhibit significant displacements along the depth axis.

To address this, we propose DOCM, which operates in 3D space. Instead of computing the motion direction solely from 2D center displacements, we extend the representation to incorporate depth-aware trajectories. Let (u_i, v_i, d_i) and (u_j, v_j, d_j) denote the 2D center coordinates and depth values of objects at two different time steps. The corresponding 3D displacement vector $\mathbf{v}_{3D}$ is defined as:

$$\mathbf{v}_{3D} = [u_j - u_i,\, v_j - v_i,\, d_j - d_i]^\top. \tag{14}$$

We evaluate the motion consistency by measuring the cosine similarity between the historical and current 3D motion vectors:

$$\mathcal{C}_{\text{VDC}} = \frac{\mathbf{v}_{\text{hist}} \cdot \mathbf{v}_{\text{curr}}}{\|\mathbf{v}_{\text{hist}}\| \cdot \|\mathbf{v}_{\text{curr}}\|}, \tag{15}$$

where $\mathbf{v}_{\text{hist}}$ connects two previous observations on the same trajectory and $\mathbf{v}_{\text{curr}}$ links the last track position with the current detection.

Tracker	HOTA↑	IDF1↑	MOTA↑	AssA↑
Motion:				
ByteTrack	47.3	52.5	89.5	31.4
C-BIoU	60.6	61.6	91.6	45.4
MotionTrack	58.2	58.6	91.3	41.7
OC-SORT	55.1	54.9	92.2	40.4
SparseTrack	55.5	58.3	91.3	39.1
UCMCTrack	63.6	65.0	88.9	51.3
Motion & Appearance:				
GHOST	56.7	57.7	91.3	39.8
StrongSORT	55.6	55.2	91.1	38.6
Deep OCSORT	61.3	61.5	92.3	45.8
DiffMOT	62.3	63.0	**92.8**	47.2
OFTrack	63.4	65.6	91.2	48.7
GRASPTrack	**65.3**	**66.2**	92.4	**52.1**

TABLE II: Performance comparison on the DanceTrack test set. The best results are shown in bold.

IV. Experiments

A. Datasets and Evaluation Metrics

1) Datasets.: We evaluate our proposed framework on three MOT benchmarks: MOT17 [10], MOT20 [11] and DanceTrack [12]. MOT17 and MOT20 datasets are standard benchmarks commonly employed in the MOT community, featuring various challenging real-world scenarios including dense crowds, frequent occlusions, and diverse camera angles. MOT17 provides annotated pedestrian tracking data with sequences captured from different perspectives, while MOT20 presents denser scenes to evaluate tracking methods under extreme occlusion and crowd conditions. In contrast, DanceTrack specifically targets challenging tracking scenarios characterized by uniform appearances and complex, diverse motions in dance performance scenes. Utilizing these diverse benchmarks allows for a comprehensive evaluation of our framework across various and realistic tracking challenges.

2) Evaluation.: We adopt standard evaluation metrics commonly used in MOT, including MOTA [32], IDF1 [33], HOTA [34], and AssA [34]. MOTA evaluates overall tracking accuracy, combining detection accuracy with identity consistency, while IDF1 specifically measures the accuracy of maintaining object identities throughout tracking. AssA is used to evalu-

ate the association performance. HOTA provides a balanced evaluation, capturing both association accuracy and detection performance.

3) Implementation Details.: Our proposed framework builds upon the OC-SORT baseline, integrating additional modules for depth estimation and segmentation. Specifically, we utilize the pretrained ViT-B Depth Anything v2 model [21] for zero-shot monocular depth estimation and ViT-S EfficientTAM [22] for precise instance segmentation. Depth maps are predicted with Depth Anything v2 and linearly scaled to the interval [0, 255]. For a fair comparison, we use the publicly available YOLOX [35] detector weights developed by ByteTrack [14]. Because the evaluated video sequences lack camera intrinsics (f_x, f_y), we first estimate them by interactively aligning a projected ground-plane grid with each image, following the method introduced in UCMCTrack [18]. The voxel size parameter δ_v for Voxel-Based 3D IoU computation is set to 0.4, balancing computational efficiency and accuracy. For our Depth-Aware Adaptive Noise Compensation (DANC), the occlusion sensitivity factor α that controls the amplification intensity of noise scaling is set to 3. The spatial overlap threshold τ_{IoU}, used to determine pairwise occlusion based on 3D IoU, is set to 0.6. During the association phase, we performed separate matching processes for high- and low-score detections, following the ByteTrack, with thresholds set to 0.6 and 0.1, respectively. We also employ the ReID model following the same settings as in DiffMOT [30]. All the experiments were conducted using a GeForce NVIDIA A100 GPU.

B. Comparison with State-of-the-art Methods

1) MOT Challenge.: In Table I, we compare the performance of GRASPTrack with the state-of-the-art TBD methods on the MOT17 and MOT20 datasets. To ensure fairness, all methods are evaluated using the same detection results and standardized evaluation protocols. From the comparison, our method demonstrates superior performance on both MOT17 and MOT20, achieving HOTA scores of 66.1 and 64.5, respectively. The results demonstrate the good efficiency and robustness of our method against complex scenes with occlusions.

2) DanceTrack.: To demonstrate the performance of our method in complex and occluded scenarios, we test our model on the DanceTrack dataset, as shown in Table II. Our results demonstrate superior performance compared to other methods and obtain a 65.3 HOTA score. The results indicate that our method can effectively handle challenging scenes with diverse motions and occlusions.

C. Ablation Study

To validate the effectiveness of our proposed depth-aware multi-object tracking framework, we conduct comprehensive ablation studies on the validation set of DanceTrack. The ablation experiments are designed to analyze four key aspects: (1) the contribution of each proposed component, (2) the impact of the Voxel Grid Size parameter, (3) the influence of the Occlusion Sensitivity Factor, and (4) the impact of the 3D Point Cloud Generation Strategy.

Appearance	3D IoU	DANC	DOCM	HOTA ↑	AssA ↑	IDF1 ↑
				52.1	35.3	51.6
✓				58.0	42.3	57.7
✓	✓			61.5	47.5	61.6
✓		✓		58.9	42.6	58.6
✓	✓	✓		62.3	48.1	63.6
✓	✓	✓	✓	**62.8**	**49.2**	**64.2**

TABLE III: Ablation study of the GRASPTrack components. 3D IoU is Voxel-Based 3D IoU.

VGS (δ_v)	HOTA ↑	AssA ↑	IDF1 ↑	FPS
0.2	62.3	48.9	63.8	9.3
0.4	**62.8**	**49.2**	**64.2**	13.1
0.6	62.0	48.8	63.4	14.0
0.8	61.8	48.6	63.1	14.8
1.0	61.6	48.5	62.9	15.1

TABLE IV: Impact of Voxel Grid Size (VGS) δ_v on the validation set of DanceTrack.

1) Component Ablation: In Table III, we systematically evaluated the contribution of each proposed component of GRASPTrack by progressively incorporating them into the OC-SORT baseline. The three key innovations are Voxel-Based 3D IoU, DANC and DOCM. Our experiments demonstrate that each component provides substantial improvements to baseline performance. The Voxel-Based 3D IoU computation enhances object association by replacing the traditional 2D IoU with volumetric similarity measures, enabling robust tracking in complex scenes with occlusions. The DANC improves tracking robustness by dynamically adjusting the process noise parameters based on detected occlusion events, which is particularly beneficial in occluded scenarios. The integration of DOCM provides the most substantial performance gain by extending motion consistency modeling from 2D to 3D space, effectively capturing complex motion patterns. The cumulative effect of all three components results in a comprehensive depth-aware MOT framework that significantly outperforms the baseline OC-SORT method on the DanceTrack dataset.

2) Voxel Grid Size: In Table IV, we conducted extensive experiments to determine the optimal voxel grid size parameter δ_v for our Voxel-Based 3D IoU, systematically varying its value from 0.2 to 1.0 in increments of 0.2. The experimental results demonstrate that $\delta_v = 0.4$ achieves the highest tracking performance on the DanceTrack dataset, yielding the best balance among HOTA (62.8), AssA (49.2), and IDF1 (64.2) metrics. When δ_v is too small (0.2), the voxel grid becomes excessively fine-grained, leading to sparse occupancy patterns that are sensitive to depth estimation noise and resulting in increased computational overhead, as indicated by the lowest FPS (9.3). Conversely, when δ_v is too large (0.8-1.0), the voxel grid becomes overly coarse, losing critical spatial details required for accurate object discrimination, though FPS performance improves (14.8 to 15.1 FPS). The optimal value of 0.4 not only provides sufficient spatial resolution to capture meaningful volumetric overlaps and maintains robustness against depth estimation uncertainties but also achieves a reasonable computational efficiency (13.1 FPS).

OSF (α)	HOTA ↑	AssA ↑	IDF1 ↑
1	61.9	48.6	63.7
2	62.3	49.1	63.8
3	**62.8**	**49.2**	**64.2**
4	62.2	49.0	64.0
5	61.8	48.7	63.5

TABLE V: Impact of OSF (Occlusion Sensitivity Factor) α on the validation set of DanceTrack.

Method	HOTA ↑	AssA ↑	IDF1 ↑
Mask-Guided Proj.	**62.8**	**49.2**	**64.2**
BoundingBox Proj.	61.7	48.4	63.1

TABLE VI: Performance impact of the 3D point cloud generation strategy on the validation set of DanceTrack.

3) Occlusion Sensitivity Factor: We investigated the impact of the occlusion sensitivity factor α in our depth-aware Kalman filtering mechanism by systematically varying its value from 1 to 5. As shown in Table V, our results reveal that $\alpha = 3$ provides the optimal balance for robust tracking performance on the DanceTrack dataset. This parameter controls the intensity of the process noise amplification during the occlusion events. When α is too small (1–2), the noise compensation mechanism becomes insufficient to account for the increased uncertainty during occlusion events, resulting in overconfident motion predictions that fail to adapt to unpredictable motion patterns. Conversely, when α is too large (4–5), the noise compensation becomes excessive, causing the Kalman filter to become overly permissive and potentially associate incorrect detections with existing tracks, leading to identity switches. The optimal value of 3 effectively addresses the motion uncertainty introduced by occlusion while maintaining sufficient discriminative power for accurate data association, and is particularly well suited for the dynamic and interactive motion patterns characteristic of group dancing scenarios.

4) 3D Point Cloud Generation Strategy: In Table VI, we conducted experiments to validate the effectiveness of our mask-guided 3D point cloud generation strategy by comparing it with alternative approaches. We compare two different strategies: (1) Mask-guided projection using EfficientTAM to obtain segmentation masks of objects (our method) and (2) Full bounding box projection using all pixels within the detection boxes. Our experimental results on the DanceTrack dataset demonstrate that the mask-guided approach achieves the best performance with a HOTA score improvement of 1.1% over the full bounding box method. The mask-guided strategy effectively eliminates background noise and occluder interference, leading to cleaner 3D point clouds and more accurate voxel-based 3D IoU calculations. In contrast, the full bounding box approach suffers from background contamination, particularly in crowded scenes where objects frequently overlap. Furthermore, we observe that stronger base detectors significantly enhance the effectiveness of our method. Detailed experimental results and ablation studies are provided in the Appendix.

V. Conclusion

This paper presents GRASPTrack, a depth-aware multi-object tracking framework that combines monocular depth estimation and instance segmentation to reconstruct high-fidelity 3D point clouds for individual objects, enabling explicit 3D geometric reasoning beyond the 2D plane. By voxelizing these mask-guided point clouds, we compute Voxel-Based 3D IoU for robust object association under heavy occlusion. We further introduce DANC, which adaptively scales Kalman filter process noise based on occlusion severity, and DOCM, which incorporates depth into motion modeling to enhance trajectory continuity. Extensive experiments demonstrate the effectiveness and robustness of our approach in comparison to the state-of-the-art methods.

References

[1] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454. I

[2] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitani, and J. Kautz, "Glamr: Global occlusion-aware human mesh recovery with dynamic cameras," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 038–11 049. I

[3] L. Torres-Ronda, E. Beanland, S. Whitehead, A. Sweeting, and J. Clubb, "Tracking systems in team sports: a narrative review of applications of the data and sport specific analysis," *Sports Medicine-Open*, vol. 8, no. 1, p. 15, 2022. I

[4] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. I, II-B, **??**

[5] F. Limanta, K. Uto, and K. Shinoda, "Camot: Camera angle-aware multi-object tracking," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6479–6488. I, II-B

[6] P. Dendorfer, V. Yugay, A. Osep, and L. Leal-Taixé, "Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 657–15 671, 2022. I, II-B, **??**

[7] T. Khurana, A. Dave, and D. Ramanan, "Detecting invisible people," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3174–3184. I

[8] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. soc. mat. mexicana*, vol. 5, no. 2, pp. 102–119, 1960. I

[9] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9686–9696. I, II-A, II-A, III-B, **??**

[10] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016. I, IV-A1

[11] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv preprint arXiv:2003.09003*, 2020. I, IV-A1

[12] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 20 993–21 002. I, IV-A1

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

[14] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European conference on computer vision*. Springer, 2022, pp. 1–21. II-A, II-A, III-B, **??**, IV-A3

[15] X. Han, N. Oishi, Y. Tian, E. Ucurum, R. Young, C. Chatwin, and P. Birch, "Ettrack: enhanced temporal motion predictor for multi-object tracking," *Applied Intelligence*, vol. 55, no. 1, p. 33, 2025. II-A, III-B

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

[17] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," in *2023 IEEE International conference on image processing (ICIP)*. IEEE, 2023, pp. 3025–3029. II-A, **??**

[18] K. Yi, K. Luo, X. Luo, J. Huang, H. Wu, R. Hu, and W. Hao, "Ucmc-track: Multi-object tracking with uniform camera motion compensation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 7, 2024, pp. 6702–6710. II-A, **??**, IV-A3

[19] X. Weng, J. Wang, D. Held, and K. Kitani, "Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics," *arXiv preprint arXiv:2008.08063*, 2020. II-B

[20] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793. II-B

[21] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2024. III-A, IV-A3

[22] Y. Xiong, C. Zhou, X. Xiang, L. Wu, C. Zhu, Z. Liu, S. Suri, B. Varadarajan, R. Akula, F. Iandola *et al.*, "Efficient track anything," *arXiv preprint arXiv:2411.18933*, 2024. III-A, IV-A3

[23] S. Shin, K. Zhou, M. Vankadari, A. Markham, and N. Trigoni, "Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4060–4069. III-A2

[24] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499. III-A2

[25] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 4799–4808. **??**

[26] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 17 939–17 948. **??**

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

[28] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 13 813–13 823. **??**

[29] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023. **??**

[30] W. Lv, Y. Huang, N. Zhang, R.-S. Lin, M. Han, and D. Zeng, "Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19 321–19 330. **??**, IV-A3

[31] Z. Song, R. Luo, L. Ma, Y. Tang, Y.-P. P. Chen, J. Yu, and W. Yang, "Temporal coherent object flow for multi-object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 7, 2025, pp. 6978–6986. **??**

[32] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, p. 246309, 2008. IV-A2

[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. IV-A2

[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, no. 2, pp. 548–578, 2021. IV-A2, IV-A2

[35] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021. IV-A3