Multiple Planar Object Tracking

https://zzcheng.top/MPOT

Zhicheng Zhang Shengzhe Liu Jufeng Yang†

VCIP & TMCC & DISSec, College of Computer Science, Nankai University

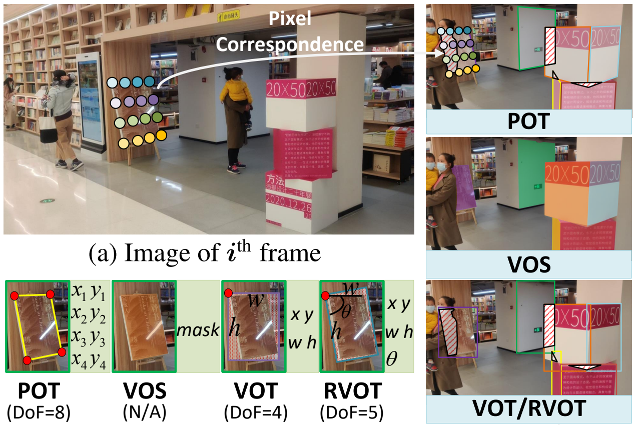
gloryzzc6@sina.com, nkuliusz@163.com, yangjufeng@nankai.edu.cn

Abstract

Tracking both location and pose of multiple planar objects (MPOT) is of great significance to numerous real-world applications. The greater degree-of-freedom of planar objects compared with common objects makes MPOT far more challenging than well-studied object tracking, especially when occlusion occurs. To address this challenging task, we are inspired by amodal perception that humans jointly track visible and invisible parts of the target, and propose a tracking framework that unifies appearance perception and occlusion reasoning. Specifically, we present a dual-branch network to track the visible part of planar objects, including vertexes and mask. Then, we develop an occlusion area localization strategy to infer the invisible part, i.e., the occluded region, followed by a two-stream attention network finally refining the prediction. To alleviate the lack of data in this field, we build the first large-scale benchmark dataset, namely MPOT-3K. It consists of 3,717 planar objects from 356 videos and contains 148,896 frames together with 687,417 annotations. The collected planar objects have 9 motion patterns and the videos are shot in 6 types of indoor and outdoor scenes. Extensive experiments demonstrate the superiority of our proposed method on the newly developed MPOT-3K as well as other two popular single planar object tracking datasets. ***The code and MPOT-3K dataset are released on https://zzcheng.top/MPOT.***

1. Introduction

Tracking multiple planar objects (MPOT) is a fundamental task in computer vision. It aims to explore the motion of planar objects, tracking both the location and pose of multiple targets simultaneously [40, 43, 56]. The planar object is defined as a plane belonging to a specific body [28] (*e.g.*, box, building, or wall) in the form of four ordered vertexes [43, 70]. With the help of MPOT, we can track targets when multiple planar objects exist, *e.g.*, different surfaces of an object [1] or planes from various objects [27]. It has attracted more and more attention attributed to various applications in augmented reality [57, 104], video editing [4, 20, 30], and robot navigation [92].



(a) Image of $\boldsymbol{i}^{\text{th}}$ frame

(b) Different tracking tasks (c) Tracking $\boldsymbol{j}^{\text{th}}$ frame

Figure 1: **Comparison with other vision tasks.** Given an image (a), we present the ground truth for different tasks in (b). The corresponding Degree-of-Freedom (DoF) is reported at the bottom and the details are listed on the right side of each task. In (c), we show the tracking results for box-based tasks (*e.g.*, VOT, RVOT), mask-based tasks (*e.g.*, VOS), and POT, which can find the occluded regions (marked by the red line area) and provide pixel-to-pixel matching correspondence (colored points across frames).

MPOT is closely related to well-known computer vision tasks in RGB tracking [32] (see Tab. 1). Both tasks involve tracking the target across subsequent frames of a video using the ground truth provided in the initial frame. However, it is more challenging in two aspects. 1) Tracking planar objects is of greater Degree-of-Freedom (DoF). As shown in Fig. 1(b), MPOT tracks both the pose and location of the target, which is described by an arbitrary quadrangle (*i.e.*, four independent vertexes $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, whose DoF is 8 [44]). In contrast, it only needs to predict the position and size of an object (x, y, w, h) in video object tracking (VOT), and Rotated VOT (RVOT) additionally requires the rotation angle. Even compared with video object

† Corresponding author.

Table 1: **The comparison of MPOT with other video-related tasks.** We compare MPOT with five RGB tracking tasks and four tasks with auxiliary modality information. The count of DoF comes from [44].

Modality	Task	Object			Scenario		
		Object category	Object selection	Multiple objects	Pixel correspondence	DoF	Occlusion
RGB	MOT [83]	Limited, usually pedestrians or vehicles	Detected	Y	N	4	Y
	MOTS [75]	Limited, usually pedestrians or vehicles	Detected	Y	N	N/A	N
	VOS [66]	Arbitrary	User-specified in the first frame	N	N	N/A	N
	VOT [21]	Arbitrary	User-specified in the first frame	N	N	4	Y
	RVOT [32]	Arbitrary	User-specified in the first frame	N	N	5	Y
	MPOT	Arbitrary	User-specified in the first frame	Y	Y	8	Y
RGB+Depth	RGBDT [91]	Arbitrary	User-specified in the first frame	Y	N	4	Y
	6DOP [50]	Limited by training set	Detected	Y	Y	6	Y
	6DPT [81]	Arbitrary	User-specified in the first frame	N	Y	6	Y
RGB+LIDAR	3dOD [35]	Limited by training set	Detected	Y	N	7	Y
	3dMOT [84]	Limited, usually vehicles	Detected	Y	N	7	Y

segmentation (VOS), an alternative that introduces masks at the pixel level, MPOT is a more challenging task. Because MPOT provides the matched correspondence for each pixel within the object region across frames [60] (*e.g.*, colored points in Fig. 1(a)&(c)), which makes it possible for applications that require positional information like texture mapping [24]. Nevertheless, VOS that tracks the target area instead of per-pixel location can hardly achieve it. 2) Occlusion is another challenge that comes with MPOT. Not only the one in POT that manually occludes the camera, MPOT but introduces the occlusion raised by the layered position of multiple targets relative to the camera [77, 96] (see Fig. 1(c)). Besides, the occlusion is more complex than the ones in multiple object tracking (MOT). As discussed above, when occlusion occurs, MPOT estimates the pixel correspondence controlled by homography matrix $\mathbf{H}$ [3], which tends to be sensitive and have a high condition number that can reach up to 5e^7 [95]. This means that a tiny movement of the invisible part is extremely difficult to track.

To address the aforementioned limitations, our work draws the inspiration that humans consolidate the visible together with invisible parts of the target for tracking [96]. We propose a tracking framework comprised of procedures for appearance **P**erception and occlusion **R**easoning, namely **PRTrack**. To track high DoF planar objects, we reformulate the problem of estimating homography matrix $\mathbf{H}$ as predicting the mask and ordered vertexes for each planar object. With the high-dimensional mask, we can accurately locate the target area at the pixel level. Besides, the ordered vertexes provide per-pixel correspondence across frames for tracking the pose of planar objects. Therefore, in the stage of appearance perception, we propose a dual branch network to predict the ordered vertexes and masks of planar objects based on the historical visible information. For vertexes, we design an encoder with shifted sampling strategy based on the constraint that the vertexes always have a clockwise order. For mask, we aggregate the probability of multiple planar objects with a multi-layered layout, that is, a stack of occluders and occludees. Further, to solve the cases of complex occlusion, we develop an occlusion area localization strategy to indicate the occluded part, by storing the movement of each planar object (*i.e.*, historical $\mathbf{H}$). To be specific, we factorize the sensitive homography matrix, which describes the relative movement of planar objects, into parameters of transition, rotation, and pose. Finally, with the prediction from the mentioned stages, we propose a two-stream self-attention network to jointly refine the predicted planar objects.

Besides, since there is no available dataset in this field, we build a large-scale benchmark dataset, namely MPOT-3K. Specifically, we shoot 356 videos with 3,717 planar objects and 687,417 annotations. The videos are collected under 9 motion patterns, where the relative movement and occlusion are also included to simulate the real-world scene.

The contributions of this paper are two-fold: 1) We collect and annotate the first large-scale benchmark dataset for MPOT, where planar objects are diversely collected expelling bias. Our dataset will be released and boost research in this field. 2) We propose a tracking framework with unified motion and appearance models, which can accurately predict the pose and location of planar objects. Extensive experiments demonstrate the superiority of PRTrack against state-of-the-art approaches.

2. Related Work

2.1. Planar Object Tracking

Benchmark Datasets. Current works focus on single planar object tracking and propose several POT datasets [19, 42, 43, 45, 70, 86]. Metaio [45] is the first dataset collected in the lab and the videos are collected by a monitored camera. Similarly, TMT [70] also shoots videos from the lab, where the annotations are automatically generated by aggregating the results of three planar object trackers. Due to the laboratory setting of the collection, videos in these datasets have similar backgrounds. Thus, POT210 [43] collects video data in the wild under seven motion patterns and encourages the diversity of the recorded videos with complex backgrounds. However, existing benchmarks are insufficient to mimic real-world settings where multiple targets exist. Therefore, our work extends the task into MPOT that

Table 2: **Statistics of tracking datasets.** "I." and "O." indicate indoor and outdoor, respectively. "#" denotes "the number of".

Benchmark	UCSB [19]	TMT [70]	POIC [8]	POT210 [43]	POT280 [42]	MOT16 [58]	MPOT-3K
# Scenes	1	1	20	30	40	14	**42**
# Videos	96	100	20	210	280	14	**356**
# Targets	96	100	20	210	280	1,276	**3,717**
# Annotations	7**K**	-	23**K**	53**K**	70**K**	292**K**	**687K**
Scene Category	I.	I.	I.&O.	I.&O.	I.&O.	O.	**I.&O.**
Multiple Objects	✗	✗	✗	✗	✗	✓	✓

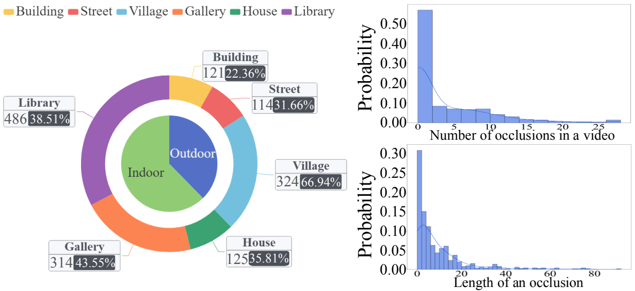
simultaneously tracks multiple planar objects which can be applied in real-world applications like AR, robots, and art.

Methods. POT trackers can be classified into region-based, keypoint-based, and hybrid methods. Region-based approaches [2, 3, 9, 17, 34, 68] focus on the whole planar region and estimate perspective transformation by closing up the template and the warped image. Keypoint-based methods [22, 61, 63, 79, 101] describe a planar object by the detected keypoints [54, 69] and associate objects across frames by computing the similarity between keypoints. As for hybrid methods, previous works [8, 9, 13, 94] exploit the robust feature descriptors for searching the optimal matching. A representative method [8] leverage gradient-based feature as criterion. As progress is made into the deep learning era [37, 82], CLKN [7] integrates deep features into the Lucas-Kanade algorithm for improving matching quality.

2.2. Visual Tracking

Single Object Tracking. It aims to locate and track a single target in a video [39, 48], which can be roughly grouped into box-based tracking [18, 55, 93] and mask-based tracking [41, 47, 60]. SiamFC [5] adopts the correlation layer to search the location which is described by the axis-aligned bounding box. Further, SiamMask [78] introduces rotation to describe the motion and predicts the object mask with the bounding box simultaneously. Meanwhile, mask-based tracking focuses on the accurate location of the targets. STM [62] addresses the problem of appearance change, leading to satisfactory results. Recently, PoST [60] explores the relationship between the mask and contour, building constraints for predicting the polygonal mask.

Multiple Object Tracking. Existing works [14, 58] propose to track multiple objects of human interests simultaneously, with the assumption that all targets belong to a list of predefined classes [10, 11, 26, 36, 51, 90, 100, 102]. A large number of methods [49, 65, 80, 85, 99] are proposed to address the task. With the class assumption, they utilize object detectors to generate candidates in the manner of axis-aligned bounding boxes. Then the candidates are associated into trajectories. While both MPOT and MOT track multiple targets at the same time, the former faces more challenges since it handles arbitrary planar objects with greater DoF, as well as covers unrestricted classes [43].



(a) Occurrence of occlusion in different scenes (b) Statistics about occlusion

Figure 2: **Statistics of occlusion in MPOT-3K.** (a) shows the frequency of occlusion in six types of scenes, in which the number on the left indicates the number of planar objects being occluded and the one in shadow represents the corresponding proportion. (b) illustrates the number of occlusions per video as well as the temporal length per occlusion.

3. The MPOT-3K Dataset

MPOT is being less explored and limited by existing datasets designed to single planar object tracking. In this work, we construct the dataset MPOT-3K for introducing multiple targets. To the best of our knowledge, MPOT-3K is the first large-scale dataset for the challenging task of MPOT. This will help facilitate future research in computer vision (augmented reality, safety surveillance), robot (navigation, manipulation), and art (video editing).

3.1. Data Collecting

Initially, we shoot 840 videos involving the motion of static and moving planar objects. For the static planar objects, we design six motion patterns (*i.e.*, far-near movement, in-plane rotation, out-plane rotation, in-plane movement, motion blur, camera occlusion) by controlling the movements of camera as common practices [43, 86]. For the moving planar objects, we consider the occlusions and relative movements among multiple planar objects and design two motion patterns (*i.e.*, moving objects, moving occlusion). Besides, an unconstrained motion pattern is considered to further combine the above motion patterns. According to [87, 103], we record diverse videos from six types of indoor and outdoor scenes, including library, house, gallery, building, street, and village.

3.2. Data Annotation

MPOT-3K is annotated by seven well-trained annotators. Following [43], we adopt a semi-automatic annotation scheme. First, the annotator labels one frame every five frames. Then, the annotated label is propagated to other unannotated frames by linear interpolation. Finally, the annotator corrects the propagated label. During annotation, we obey the following rules. First, we define what planar object is to be annotated according to [43, 70, 86]. The planar object should be easily presented by four ordered ver-

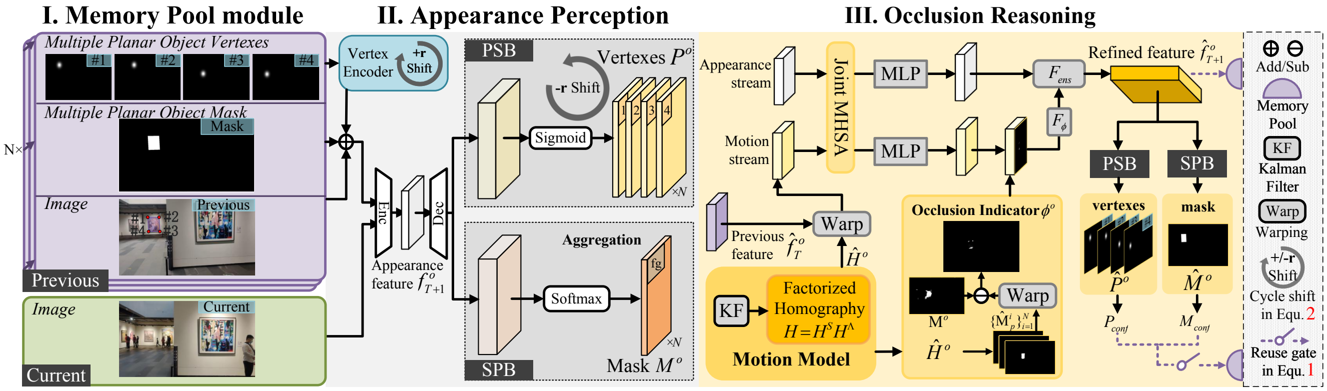


Figure 3: **The pipeline of our method.** Our tracking framework is divided into three stages. First, in appearance perception, we track the visible part of multiple planar objects via a dual-branch network, as coarse output. For the next stage of occlusion reasoning, the occlusion area is localized and then fed into a two-stream self-attention network for refining the predicted target. Finally, the memory pool module restores the high-confident tracked targets.

texes. To ensure that the criteria for selecting plane objects are consistent, we ask all the annotators to vote for filtering out inconsistent objects [52]. Second, we start annotating when the planar object is completely in view and end when the object has been fully occluded or out of view. Third, we annotate the spatial position of the planar object according to two principles: 1) We annotate the planar object which has four apparent lines, since the point is localized by inferring the intersection lines [67]. 2) We maintain the order of vertexes across video frames for pixel correspondence. Combining these principles, we can locate the planar object with four independent vertexes (*i.e.*, a quadrangle).

3.3. Statistics

After collecting and labeling, we yield a large-scale dataset MPOT-3K, where we obtain 356 videos with 3,717 planar objects from 42 scenes. The number of planar objects per video averages 10.44 and can reach 74 at most. MPOT-3K is divided into training, validation, and test sets in a proportion of 80:5:15. The split is performed at the scene level to ensure there's no overlap among splits. Hence the dataset avoids leakage since both the target and scene of the test set are expelled during training.

As shown in Tab. 2, we compare MPOT-3K with six datasets, which come from POT [8, 19, 42, 43, 70] and MOT [58] tasks. MPOT-3K contains over 9.8 times more annotations and 13.2 times more targets in all video frames than the largest POT dataset POT280. The number of targets is almost 3 times of the popular MOT16 dataset. Another strength of MPOT-3K lies in its diversity, which covers 9 motion patterns and 6 types of scenes. Besides, MPOT-3K introduces more complex occlusions. As shown in Fig. 2(a), we observe that occlusion occurs in all the scenes, where 39.9% of planar objects are occluded on average. Fig. 2(b) further illustrates that there are 3.6 occlusions happening in a video on average and each occlusion lasts 9.56 seconds.

4. Methodology

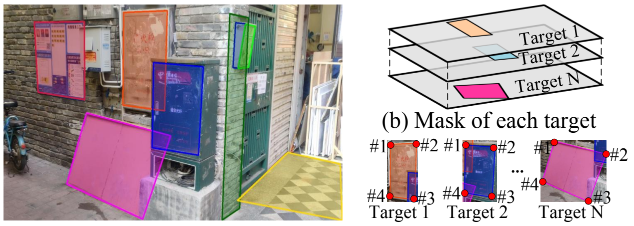
PRTrack consists of three main components (See Fig. 3): memory pool module, appearance perception network, and occlusion reasoning network. The memory pool module (Sec. 4.2) restores the previous predictions and expels the low-confident targets. Guided by previously tracked targets, the appearance perception network (Sec. 4.3) predicts mask together with the vertexes for each planar object. The occlusion reasoning network (Sec. 4.4) further takes all the tracked targets and corresponding occlusion area, which is indicated by the difference between motion-guided results among multiple targets and the predicted one.

4.1. Problem Reformulation

Assume multiple planar objects exists in an RGB video where they are not from the training set, nor have detection results. Given the user-selected ones in the initial frame, the objective is to estimate their position change relative to the beginning, *i.e.*, homography matrix, with which the pose change of targets can be obtained [56]. As shown in Fig. 4, we reformulate the tracking task as predicting the masks of the quadrangle $\widehat{\mathbf{M}} \in \mathbb{R}^{O\times hw}$ and the heatmap of four ordered vertexes $\hat{\mathbf{P}} \in \mathbb{R}^{O\times 4hw}$, where O is the number of planar objects in the current frame. The layered masks represent the location of multiple planar objects [94] and the vertexes are used to track the pose of planar objects [56].

4.2. Memory Pool module

From the viewpoint of human perception, one person tracks the targets by memorizing their historical appearance and trajectory [59]. Inspired by this, we leverage the images and predictions in previous T frames as guidance to track planar objects in the current one. For the previous frames, we record previously predicted heatmaps of ordered vertexes $\mathbf{P_o} \in \mathbb{R}^{T\times 4hw}$, masks $\mathbf{M_o} \in \mathbb{R}^{T\times hw}$, and the images $\mathbf{I_o} \in \mathbb{R}^{T\times 3hw}$ for the o^{th} planar object. The memory pool $\{(\mathbf{P_o}, \mathbf{M_o}, \mathbf{I_o})\}_{o=1}^{N}$ stores the tracked N planar objects.



(a) Image (b) Mask of each target (c) Vertexes of each target

Figure 4: **Illustration of the reformulated form of MPOT.** We predict heatmaps of four ordered vertexes and the mask for each planar object. The stacked masks are built in a layered structure and are responsible for the occluders and occludees.

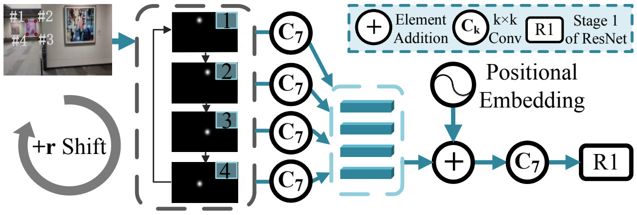


Figure 5: **Illustration of our proposed vertex encoder.** Given the vertex heatmaps, we random sample them by a cyclic shift operation, following a clockwise order and with an offset of r. Note that PSB, as a counterpart, predicts the vertexes via a cyclic shift function with negative r.

However, due to the challenge of accurately tracking multiple planar objects, the error accumulation [46] occurs because of memorizing incorrect predictions. To avoid this issue, we filter out the low-confident predictions. Given the current frame, PRTrack predicts the vertexes $\widehat{P}^o \in \mathbb{R}^{4hw}$ and mask $\widehat{M}^o \in \mathbb{R}^{hw}$ for the oth tracked planar object. We compute the confidence of the mask and vertexes as

$$\begin{aligned} M_{conf} &= min(\{max(\widehat{M}^o)\}_{o=1}^{O}). \\ P_{conf} &= min(\{max(\widehat{P}^o)\}_{o=1}^{O}) \end{aligned} \quad (1)$$

The predictions with the confidences surpassing the threshold are restored in the memory pool module. The reuse gate is only applied during testing.

4.3. Appearance Perception

Our appearance perception network is built on top of the encoder-decoder architecture [62], with a backbone of ResNet50. For the current frame, we extract appearance feature $f_{cur} \in \mathbb{R}^{CHW}$ from an image, where H, W, C denotes the width, height, and the number of channels of the feature. For previous frames, we extract features from vertexes, masks, and images, and sum the three features as $\boldsymbol{f_{pre}} \in \mathbb{R}^{T \times CHW}$. The current image feature and previous target features are fed into the encoder network to yield appearance features $f^o_{T+1} \in \mathbb{R}^{CHW}$ for o^{th} planar object. The plane structure branch and segmentation prior branch finally predict the coarse results of vertexes and masks.

Vertex Encoder with Shifted Sampling Strategy. Considering the vertexes of the planar object provide the pixel correspondence across frames, each vertex is independent against others, thus they are organized in sequential order [47]. Therefore, we design an encoder with shifted sampling strategy to leverage the characteristic of vertexes (*i.e.*, the clockwise order), as the feature extractor. As shown in Fig. 5, a random cyclic shift function $shift(\cdot, +r)$ is first applied to build the connection between the vertex heatmap $P_i \in \mathbb{R}^{HW}$ and its neighbors by shuffling position as

$$shift(P, +r) = [P_{(1+r)\%L}, \ldots, P_{(L+r)\%L}], \quad (2)$$

where L is the number of vertexes in the planar object and $r \in \{0, 1, 2, 3\}$ is a random number. $\%$ is the mod operation. Then, we apply 7 × 7 convolutions [25] to extract the features of four ordered vertexes, separately. For modeling the order between four vertexes, we introduce the cosine positional embedding [74] and add it to the vertex representation. The positional embedding $\boldsymbol{f_{emb}} \in \mathbb{R}^{L \times N}$ contains the N-d embeddings corresponding to the order information for the features of L vertexes. The feature of ordered vertexes is then extracted by a 7×7 convolution to capture long-range spatial interaction between vertexes.

Plane Structure Branch (PSB). Given the shifted feature map of vertexes from the decoder, a 1×1 convolution is attached with the sigmoid function to output the heatmap of the vertexes. The shifted vertex heatmaps are then aligned with the shifted input in Equ. 2 by using the reversed cyclic shift operation $shift(\cdot, -r)$. Finally, we get the heatmaps of o^{th} planar object $P^o \in \mathbb{R}^{HW}$ by upsampling to the same resolution as the original image.

Segmentation Prior Branch (SPB). The mask can accurately locate the planar object at the pixel level and capture the relative movement between objects such as occlusion [55,78]. Therefore, we utilize a 1×1 convolution with the softmax function alongside the foreground/background dimension. Following [62], The probability of background m_{bkg} and each foreground object m_o are obtained by aggregating the output of model $logits \in \mathbb{R}^{O \times HW}$ as

$$m_o = \frac{\exp(logits_o)}{\sum_{i=1}^{O} \exp(logits_i)}, \quad m_{bkg} = \prod_{i=1}^{O} (1 - m_o). \quad (3)$$

Finally, the probability segmentation mask of o^{th} planar object $M^o \in \mathbb{R}^{HW}$ are produced after softmax. The background probability map is attached in the last channel.

4.4. Occlusion Reasoning

With the above coarse output perceiving from the visible part, we move to the next stage of reasoning the occluded part by the motion of planar objects (*i.e.*, homography). In multiple object tracking, reasoning the occlusion by motion has been verified effective [97, 98], where a motion strategy is adopted to predict the future location of the target based on the historical trajectory. Inspired by this, we first develop a homography-guided strategy that locates the occlusion area and correct features from previous frames geometrically. We factorize the homography matrix into motion parameters that describe the common transformations, to estimate the sensitive homography matrix (as discussed in Sec. 1). Then, for tracking both visible and invisible parts of planar objects, we design a two-stream self-attention network that takes both the appearance features of the current frame and the motion-guided features of the previous frame to refine the predicted planar object.

Occlusion Area Localization Strategy. To avoid the sensitivity problem discussed in Sec. 1, we first introduce the geometric definition of homography matrix $\mathbf{H} \in \mathbb{R}^{3\times 3}$ [23], and factorize it into similarity transformation $\mathbf{H^S}$ and residual one $\mathbf{H^\Lambda}$, with eight stable motion parameters of $h_p = (t_x, t_y, \gamma, \theta, k_1, k_2, v_1, v_2)$ as

$$\mathbf{H} = \mathbf{H^S}\mathbf{H^\Lambda} = \begin{bmatrix} \gamma\cos\theta & -\gamma\sin\theta & t_x \\ \gamma\sin\theta & \gamma\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & 0 \\ 0 & \frac{1}{k_1} & 0 \\ v_1 & v_2 & 1 \end{bmatrix}, \tag{4}$$

where t_x, t_y are the transition offset, γ and θ denote the change of scale and angle, and k_1, k_2, v_1, v_2 control the parameters of matrix $\mathbf{H^\Lambda}$. Here, the motion parameters can be computed by solving a transcendental equation with eight variants. The location parameters can be written as $h_l = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$.

Specifically, the location parameters and homography are obtained from the coarse output of appearance perception, where we apply the Argmax function on heatmaps to get the coordinates of vertexes and compute the homography matrix between the coordinates from the current frame and the previous one. Then, the factorized homography h_p and location parameters h_l are fed into the Kalman Filter [29] and yield the corrected ones $\widehat{h}_p, \widehat{h}_l$. Finally, homography matrix $\mathbf{\hat{H}^o}$ for the o^{th} planar object in the current frame can be obtained.

Given the predicted homograhpy, the mask from previous frame $\widehat{M}^o_p$ can be warped [3] by the perspective transformation $warp(\widehat{M}^o_p, \mathbf{\hat{H}^o})$. To locate occlusion area, the intuition is that the heatmap inferred by historical motion is robust to the occlusion on appearance. Meanwhile, the heatmap predicted based on appearance can accurately locate the planar objects in the current frame. As a result, we indicate the area of occlusion as

$$\phi^o = \frac{1}{\sum_i \delta_i} \sum_i \delta_i \cdot \|warp(\widehat{M}^i_p, \mathbf{\hat{H}^i}) - M^o\|, \tag{5}$$

where $\|\cdot\|$ is the distance function computed per pixel and δ_i judges the mask confidence of i^{th} planar object.

Motion Reasoning Network. For reasoning the visible and invisible parts of planar objects, we leverage the occlusion indicator ϕ^o as guidance, where the features f^o_{T+1} from the current frame are refined, together with the warped feature $warp(f^o_T, \mathbf{\hat{H}^o})$ from the previous frame. We denote these two features as f_a and f_m, respectively. The features are first flattened alongside spatial dimension $f_a, f_m \in \mathbb{R}^{c_d \times HW}$. Then, we develop a multi-head self-attention (MHSA) mechanism [31] for extracting occlusion-aware features. For learning joint representation and encouraging the interactions between appearance and motion, we feed both features into MHSA as

$$\begin{cases} X = \text{Concat}(h_1, \dots, h_H) \\ h_i = a_i f \mathbf{W_i^V} \\ a_i = \text{softmax}(\frac{[f\mathbf{W_i^Q}][f\mathbf{W_i^K}]^\top}{\sqrt{d_k}}), \\ f = \text{Concat}(f_m, f_a) \end{cases} \tag{6}$$

where $\mathbf{W_i^Q}, \mathbf{W_i^K}$, and $\mathbf{W_i^V}$ are the matrixes. We then adopt MLP to extract representations for motion feature X_m and appearance feature X_a, respectively. For assembling, we integrate both features guided by indicator ϕ^o

$$F_{occ} = \mathcal{F}_{ens}(\mathcal{F}_\phi(X_a), X_m), \tag{7}$$

where $\mathcal{F}_{ens}$ and $\mathcal{F}_\phi$ are 1×1 convolution layer, $\mathcal{F}_\phi$ concatenates the occlusion indicator with input feature as guidance. Finally, given the mask and vertex heatmaps of the o^{th} occluded planar object, we feed the features into the dual-branch network used in the stage of appearance perception again and obtain the refined vertexes $\widehat{P}^o$ and mask $\widehat{M}^o$.

5. Experiment

5.1. Evaluation Metrics

To evaluate methods on MPOT, following [15], we utilize the CLEAR metric [58] to complement the POT metrics. Note that existing POT metrics estimate the difference between ground truth and prediction in a pairwise manner (*i.e.*, 1-box to 1-box). But it is not suitable for MPOT, since the number of predictions and ground truths are not always equal. Therefore, we match the set of ground truth GT_t and prediction P_t by bipartite matching [33]. Sequentially, we compute the difference of the matched pair by well-known POT distance Alignment Error [43, 45, 70] as

$$E_{AL}(\mathbf{p}, \mathbf{g}) = \sqrt{\frac{\sum_{\mathbf{j=1}}^{\mathbf{4}} (\mathbf{p_j} - \mathbf{g_j})^{\mathbf{2}}}{\mathbf{4}}}, \tag{8}$$

where $\mathbf{p}$ and $\mathbf{g}$ are the quadrangle of the prediction and ground truth. Then, the prediction and ground truth are split into tracked T^t_ϵ, true positive TP^t_ϵ, false negative FN^t_ϵ, and false positive FP^t_ϵ, according to distance ϵ. We set ϵ as 50 in our experiment due to the degree of challenge.

In POT, Precision (Pr_ϵ) is the commonly used metric based on the alignment error [43]. To complement the metrics, we report Recall (Rc_ϵ) to consider the cases of not being tracked. Furthermore, we propose Multiple Planar Object Tracking Distance (D_ϵ) as

$$D_\epsilon = \frac{\sum_{t=1}^T \sum_{tp \in TP^t_\epsilon, q \in T^t_\epsilon} E_{AL}(tp, q)}{\sum_{t=1}^T |TP^t_\epsilon|}. \tag{9}$$

Naturally, we propose the Multiple Planar Object Tracking Accuracy (Acc_ϵ) based on E_{AL} as.

$$Acc_\epsilon = 1 - \frac{\sum_{t=1}^T |FN^t_\epsilon| + |FP^t_\epsilon| + |IDSW^t_\epsilon|}{3 \cdot \sum_{t=1}^T |GT_t|}, \tag{10}$$

where $IDSW^t_\epsilon$ is the set of the identity switch [58] under the threshold ϵ in tth frame, and $|\cdot|$ indicates the mod oper-

Table 3: **Comparison of two groups of methods on MPOT-3K.** The results on five metrics including Success Rate, Multiple Planar Object Tracking Accuracy, Multiple Planar Object Tracking Distance, Precision, Recall, and Speed are reported. ↑ (↓) denotes that the higher (lower) the score, the method performs better. **Bold** and underline indicate the highest and second-highest performance, respectively.

Methods		Traditional						Deep									
Motion Patterns	Metrics	CMT [61]	NCC [71]	CCRE [76]	MI [17]	GOP [8]	Gracker [79]	STM [62]	PoST [60]	LISRD [64]	SMask [78]	SRPN+ [38]	SPoint [16]	SOS [73]	GIFT [53]	HDN [95]	Ours
Overall	$S_{0.8} \uparrow$	09.50	00.19	09.99	26.70	56.03	67.92	09.07	33.52	63.43	64.04	67.23	71.82	72.91	75.86	76.40	**82.51**
	$Acc_\epsilon \uparrow$	64.27	61.95	34.34	37.82	62.28	88.38	52.05	39.55	72.39	73.29	75.37	71.17	70.06	72.99	79.50	**94.59**
	$Pr_\epsilon \uparrow$	37.05	03.64	14.12	23.57	45.25	85.67	25.64	25.02	56.23	58.38	60.70	54.58	53.23	56.41	65.21	**92.85**
	$Rc_\epsilon \uparrow$	10.24	00.55	19.08	38.60	62.72	78.25	23.07	40.74	80.40	69.55	74.45	81.25	85.37	84.02	82.56	**90.78**
	$D_\epsilon \downarrow$	12.30	42.15	17.27	08.68	06.37	05.23	18.79	22.71	06.93	17.06	16.77	07.67	07.76	07.80	07.44	**05.07**
Moving Occlusion	$S_{0.8} \uparrow$	11.34	00.00	10.69	26.47	55.85	63.68	08.78	18.17	73.79	63.29	70.62	70.91	73.42	72.92	74.96	**80.29**
	$Acc_\epsilon \uparrow$	67.25	62.78	56.21	68.10	81.41	92.93	55.49	48.04	91.74	81.46	83.27	89.92	92.28	89.74	88.74	**93.16**
	$Pr_\epsilon \uparrow$	53.48	4.92	33.56	51.82	70.05	76.30	29.63	25.45	86.37	74.57	75.94	84.73	84.35	84.21	82.00	**90.95**
	$Rc_\epsilon \uparrow$	13.47	00.64	32.01	61.57	77.30	81.98	24.37	28.98	80.70	67.62	72.97	85.39	85.03	85.50	84.86	**88.27**
	$D_\epsilon \downarrow$	06.31	42.14	21.29	08.79	04.13	07.92	16.54	27.26	06.41	15.57	14.29	06.06	06.17	06.05	06.24	**05.34**
Moving Objects	$S_{0.8} \uparrow$	09.34	00.69	11.23	24.29	52.21	61.58	08.09	35.52	67.79	62.95	65.49	65.50	70.36	67.02	70.09	**79.12**
	$Acc_\epsilon \uparrow$	69.22	63.86	40.78	47.41	68.96	92.54	53.77	41.96	76.88	80.33	82.72	78.44	69.29	78.59	81.05	**94.29**
	$Pr_\epsilon \uparrow$	60.28	17.10	19.24	32.57	52.50	90.48	28.33	27.13	60.87	68.62	72.09	67.75	52.41	67.97	67.18	**92.27**
	$Rc_\epsilon \uparrow$	22.87	2.19	24.27	53.95	72.51	56.77	25.28	43.96	87.73	75.75	79.22	67.66	86.09	67.79	84.43	**90.44**
	$D_\epsilon \downarrow$	12.71	43.21	17.66	06.85	05.71	**04.11**	21.77	19.93	05.15	14.92	15.48	04.21	04.53	04.16	05.63	05.46
Camera Occlusion	$S_{0.8} \uparrow$	14.61	00.00	13.31	41.79	63.59	66.41	13.80	40.21	85.46	62.46	59.40	88.85	88.28	89.43	84.01	**92.46**
	$Acc_\epsilon \uparrow$	63.54	61.93	34.28	40.28	61.72	87.98	51.72	37.15	74.06	61.66	68.71	72.18	71.90	72.10	84.51	**95.85**
	$Pr_\epsilon \uparrow$	36.89	00.38	18.42	27.84	45.04	88.24	24.59	25.31	56.76	44.83	52.31	54.67	54.42	54.60	73.16	**95.84**
	$Rc_\epsilon \uparrow$	13.19	00.06	28.32	49.74	67.38	73.75	21.70	45.37	84.32	64.86	70.75	87.57	87.15	87.50	84.52	**91.53**
	$D_\epsilon \downarrow$	10.31	29.38	20.10	07.71	06.73	04.63	25.05	20.08	04.82	14.46	14.45	04.81	04.62	04.73	05.18	**04.59**
	$Speed \uparrow$	00.58	01.98	00.31	01.25	01.00	03.14	**35.44**	11.90	08.63	31.52	10.87	09.32	12.51	26.62	16.37	12.64

ation. To fairly compare along with the trajectory level, we adopt the success rate (S_α) with the distance threshold α as

$$S_\alpha = \frac{|\mathbb{C}|}{|\mathbb{I}|}, \mathbb{C} = \{i_n \in \mathbb{I} | \frac{\sum_{t=1}^{T} |T_\epsilon^t|}{\sum_{t=1}^{T} |GT_t|} > \alpha\}, \tag{11}$$

where $\mathbb{I}$ is the set of planar objects in the video.

5.2. Implementation Details

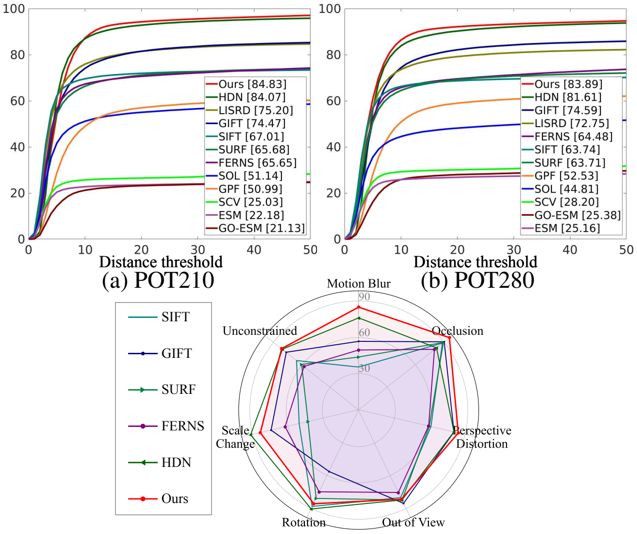
We compare our model with fifteen representative methods. These methods are implemented by official code or open-source libraries [6,72]. We run evaluated single object trackers multiple times for different planar objects. And we equip them with the data association strategy based on [12], for integrating each single object tracker into tracking multiple targets online. Besides, we additionally give the reference frames of ground truth if the single object trackers fail to track. Different from other box-based trackers, mask-based trackers (*e.g.*, STM) directly outputs the mask. Thus, we use a rotated box estimator [88] to transfer the mask. We implement PRTrack using two RTX3090 GPUs. In the training stage, we use data augmentations, including flip, pepper noise, rotation, contrast jittering, and perspective transformation. We adopt a data sampling strategy following [41]. We adopt Smooth L1 loss and Cross-Entropy loss for optimizing the predicted vertex heatmaps and masks. Similar to [62], we pretrain SPB on YouTubeVOS [89] and DAVIS [66] datasets. During inference, we use the ordered vertexes to build the planar objects and estimate the homography matrix. We set the threshold as 0.9 in the reuse gate by grid search and set r in Equ. 2 as 0 during inference.

5.3. Results on MPOT-3K

We compare our PRTrack with the other fifteen trackers on MPOT-3K, which contains six traditional POT trackers, four deep-based generic object trackers, and five deep-based planar object trackers. Tab. 3 reports the overall performance as well as the one in the challenging situations of moving occlusion, moving objects, and camera occlusion.

Deep vs. Traditional Trackers. We can observe that the deep-based trackers have increased by 82.6% on average, on five metrics for overall performance. This is because deep-based trackers avoid taking some assumptions under the laboratory environment, such as the invariance to illumination and appearance change. Note that Gracker performs best in traditional trackers and achieves competitive performance with the deep-based tracker group. However, since MPOT-3K has many small planar objects, it can not extract sufficient information, thus resulting in suboptimal performance (*i.e.*, 10.3% degradation in comparison with ours). In terms of deep-based trackers, STM predicts segmentation masks at the pixel level while ignoring the order of vertexes. Thus, it results in inferior performance, where only 9.0% of targets are correctly tracked.

Planar Object Trackers vs. Generic Object Trackers. Planar object trackers achieve higher performance (58.9% improvement) against generic object trackers, including SRPN+, SMask, STM, and PoST. A predominant reason is that the generic trackers are subject to the assumption of affine transformation, where only the coarse box estimation is required. Therefore, a powerful planar object tracker should be developed in this field.



(a) POT210 (b) POT280

(c) Challenging factors

Figure 6: **Comparison on POT210 and POT280.** The averaged Precision on two datasets with different thresholds are reported in (a) and (b), respectively. (c) further shows the performance under seven challenging factors. The data is extracted from [95].

PRTrack vs. SOTA Trackers. PRTrack outperforms other methods with an average improvement of 6.4%. We also observe a large gain when there exists relative movement among multiple targets. To be specific, when facing challenging situations like moving objects, moving occlusion, and camera occlusion, PRTrack achieves a performance gain of 16.9% against other advanced algorithms.

5.4. Results on POT210 and POT280

To verify the effectiveness of PRTrack on the traditional POT task, we conduct experiments on the POT210 and POT280 datasets, and compare PRTrack with eleven POT methods as shown in Fig. 6. Note that we do not fine-tune PRTrack on these two datasets, again verifying the generalization ability of PRTrack towards unseen scenes. Following [95], the averaged Precision is employed to evaluate all the methods. As the distance threshold E_{AL} increases, the precision tends to be stable when E_{AL}>10. We notice that PRTrack achieves competitive performance in comparison with advanced algorithms. Besides, PRTrack shows its superiority when handling challenging cases such as occlusion and motion blur.

5.5. Ablation Study

We conduct ablation studies on MPOT-3K in Tab. 4 to explore the effectiveness of each component in PRTrack. Our baseline is the appearance perception module, without shifted encoder and auxiliary mask. For simplicity, we use "i&j" to denote the comparison between the i^{th} line and the j^{th} line in the rest of this subsection.

We first perform module-wise ablation studies in the highlighted lines, we observe that PRTrack obtains the largest performance gain of 11.6% (*i.e.*, ③&⑨ and ④&⑩) with the occlusion reasoning module. Then we investigate the variants of each module. For appearance perception, we study how to leverage the order of vertexes. The boosting of 16.3% (*i.e.*, ②&③ and ⑧&⑩) verifies that the proposed encoder avoids learning position bias, where the first vertex commonly comes from the top-left area. For occlusion reasoning, we find the solution to unify the appearance feature and motion feature. We first conduct experiments on alternative solutions of fusion (*i.e.*, ⑤&⑥&⑩), where our strategy achieves the best performance. Because the occlusion indicator provides the change of image against occlusion and joint embedding helps to introduce interaction between features. We also identify the effectiveness of homography factorization via the comparison of ⑦&⑩, with an average boosting of 7.2%. For memory pool, we investigate the accumulated error. With a gain of 4.4% (*i.e.*, ③&④ and ⑨&⑩), we believe the proposed reuse gate can avoid the error derived from restoring the predictions.

Table 4: **Ablation study of components and their variants of PRTrack on the validation split of MPOT-3K.** The lines with background indicate the module-level ablation study. Here AP, OR, and RT are short of appearance perception, occlusion reasoning, and memory pool. SE and Fac denote the shifted encoder and factorized homography matrix. Msk represents the plane structure branch with mask information. Here, '$*$', '$\bullet$', and '$\diamondsuit$' are the fusion strategy (Fus), representing concatenation, attention-based, and indicator-based. Gat indicates the reuse gate.

Id	AP		OR		RT	Metrics				
	SE	Msk	Fac	Fus	Gat	$S_{0.8}\uparrow$	$Acc_\epsilon\uparrow$	$Pr\uparrow$	$Rc\uparrow$	$D_\epsilon\downarrow$
①					✓	45.29	68.01	46.75	46.71	19.56
②		✓			✓	63.76	86.82	82.36	76.96	15.40
③	✓	✓				75.82	91.08	86.22	82.96	12.57
④	✓	✓			✓	78.20	91.43	87.19	86.42	09.95
⑤	✓	✓	✓	✓*	✓	77.53	91.46	**90.41**	86.25	10.43
⑥	✓	✓	✓	✓$^\bullet$	✓	78.99	91.41	88.07	85.87	09.12
⑦	✓	✓		✓$^\diamondsuit$	✓	79.24	91.50	88.10	86.13	08.95
⑧		✓	✓	✓$^\diamondsuit$	✓	76.38	90.56	88.58	82.34	10.59
⑨	✓	✓	✓	✓$^\diamondsuit$		80.44	92.48	89.67	87.54	07.38
⑩	✓	✓	✓	✓$^\diamondsuit$	✓	**80.58**	**92.75**	90.00	**88.03**	**06.39**

6. Conclusion

In this paper, we propose a novel method for tracking multiple planar objects simultaneously. We also build the first large-scale benchmark dataset MPOT-3K which contains sufficient data, is labeled by experienced annotators, and covers various scenes. Extensive experiments are conducted on both MPOT3K and the other two single planar object tracking datasets. The results demonstrate that PRTrack achieves SOTA performance on both tasks.

7. Acknowledgments

This work was supported by the National Key Research and Development Program of China Grant (NO. 2018AAA0100400), Natural Science Foundation of Tianjin, China (NO.20JCJQJC00020), Fundamental Research Funds for the Central Universities, and Supercomputing Center of Nankai University (NKSC).

References

[1] Mark Ashdown, Matthew Flagg, Rahul Sukthankar, and James M Rehg. A flexible projector-camera system for multi-planar displays. In *CVPR*, 2004.

[2] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[3] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IROS*, 2004.

[4] Eric P. Bennett and Leonard McMillan. Proscenium: A framework for spatio-temporal video editing. In *ACM MM*, 2003.

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.

[6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *CVPR*, 2017.

[8] Lin Chen, Haibin Ling, Yu Shen, Fan Zhou, Ping Wang, Xiang Tian, and Yaowu Chen. Robust visual tracking for planar objects using gradient orientation pyramid. *Journal of Electronic Imaging*, 28(1):013007, 2019.

[9] Lin Chen, Fan Zhou, Yu Shen, Xiang Tian, Haibin Ling, and Yaowu Chen. Illumination insensitive efficient second-order minimization for planar object tracking. In *ICRA*, 2017.

[10] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. *arXiv*, 2022.

[11] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, 2023.

[12] Qi Chu, Wanli Ouyang, Bin Liu, Feng Zhu, and Nenghai Yu. Dasot: A unified framework integrating data association and single object tracking for online multi-object tracking. In *AAAI*, 2020.

[13] Alberto Crivellaro and Vincent Lepetit. Robust 3d tracking with descriptor fields. In *CVPR*, 2014.

[14] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *IJCV*, 129(4):845–881, 2021.

[15] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv*, 2020.

[16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[17] Nicholas Dowson and Richard Bowden. Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. *TPAMI*, 30(1):180–185, 2007.

[18] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.

[19] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *IJCV*, 94(3):335, 2011.

[20] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *ACM UIST*, 2008.

[21] SJ Hadfield, R Bowden, and K Lebeda. The visual object tracking vot2016 challenge results. *Lecture Notes in Computer Science*, 9914:777–823, 2016.

[22] Sam Hare, Amir Saffari, and Philip HS Torr. Efficient online structured output learning for keypoint-based object tracking. In *CVPR*, 2012.

[23] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[24] Paul S Heckbert. Survey of texture mapping. *IEEE computer graphics and applications*, 6(11):56–67, 1986.

[25] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017.

[26] S. Hu, X. Zhao, L. Huang, and K. Huang. Global instance tracking: Locating target more like humans. *TPAMI*, 1(1):1–1, 2022.

[27] Linyi Jin, Shengyi Qian, Andrew Owens, and David F. Fouhey. Planar surface reconstruction from sparse views. In *ICCV*, 2021.

[28] Olaf Kähler and Joachim Denzler. Rigid motion constraints for tracking planar objects. In *Joint Pattern Recognition Symposium*, 2007.

[29] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[30] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics*, 40(6):1–12, 2021.

[31] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[32] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ECCVW*, 2018.

[33] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[34] Junghyun Kwon, Hee Seok Lee, Frank C Park, and Kyoung Mu Lee. A geometric particle filter for template-based visual tracking. *TPAMI*, 36(4):625–643, 2013.

[35] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, 2023.

[36] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[38] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[39] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *PR*, 76:323–338, 2018.

[40] Xi Li, Liming Zhao, Wei Ji, Yiming Wu, Fei Wu, Ming-Hsuan Yang, Dacheng Tao, and Ian Reid. Multi-task structure-aware context modeling for robust keypoint-based object tracking. *TPAMI*, 41(4):915–927, 2019.

[41] Yuxi Li, Ning Xu, Jinlong Peng, John See, and Weiyao Lin. Delving into the cyclic mechanism in semi-supervised video object segmentation. In *NeurIPS*, 2020.

[42] Pengpeng Liang, Haoxuanye Ji, Yifan Wu, Yumei Chai, Liming Wang, Chunyuan Liao, and Haibin Ling. Planar object tracking benchmark in the wild. *Neurocomputing*, 454:254–267, 2021.

[43] P. Liang, Y. Wu, H. Lu, L. Wang, C. Liao, and H. Ling. Planar object tracking in the wild: A benchmark. In *ICRA*, 2018.

[44] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *TIP*, 27(8):3676–3690, 2018.

[45] Sebastian Lieberknecht, Selim Benhimane, Peter Meier, and Nassir Navab. A dataset and evaluation methodology for template-based tracking algorithms. In *ISMAR*, 2009.

[46] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia. Video instance segmentation with a propose-reduce paradigm. In *ICCV*, 2021.

[47] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019.

[48] Chang Liu, Xiao-Fan Chen, Chun-Juan Bo, and Dong Wang. Long-term visual tracking: review and experimental comparison. *Machine Intelligence Research*, 19(6):512–530, 2022.

[49] Jiahui Liu, Chirui Chang, Jianhui Liu, Xiaoyang Wu, Lan Ma, and Xiaojuan Qi. Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In *CVPR*, 2023.

[50] Jianhui Liu, Yukang Chen, Xiaoqing Ye, and Xiaojuan Qi. Prior-free category-level pose estimation with implicit space transformation. *arXiv*, 2023.

[51] Jianhui Liu, Yukang Chen, Xiaoqing Ye, Zhuotao Tian, Xiao Tan, and Xiaojuan Qi. Spatial pruned sparse convolution for efficient 3d object detection. *NeurIPS*, 2022.

[52] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *TPAMI*, 33(2):353–367, 2010.

[53] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *NeurIPS*, 2019.

[54] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[55] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s - a discriminative single shot segmentation tracker. In *CVPR*, 2020.

[56] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *TVCG*, 22(12):2633–2651, 2015.

[57] Dmitrii Matveichev and Daw-Tung Lin. Mobile augmented reality: Fast, precise, and smooth planar object tracking. In *ICPR*, 2021.

[58] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv*, 2016.

[59] Roy Allen;Peter Mcgeorge;David G. Pearson;Alan Milne;. Multiple-target tracking: A role for working memory? *Quarterly Journal of Experimental Psychology*, 59(6):1101–1116, 2006.

[60] Gunhee Nam, Miran Heo, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Polygonal point set tracking. In *CVPR*, 2021.

[61] Georg Nebehay and Roman Pflugfelder. Clustering of Static-Adaptive correspondences for deformable object tracking. In *CVPR*, 2015.

[62] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[63] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *TPAMI*, 32(3):448–461, 2009.

[64] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *ECCV*, 2020.

[65] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, 2020.

[66] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv*, 2017.

[67] Yiming Qian and Yasutaka Furukawa. Learning pairwise inter-plane relations for piecewise planar reconstruction. In *ECCV*, 2020.

[68] Rogério Richa, Raphael Sznitman, Russell Taylor, and Gregory Hager. Visual tracking using the sum of conditional variance. In *IROS*, 2011.

[69] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *TPAMI*, 32(1):105–119, 2008.

[70] A. Roy, Xi Zhang, N. Wolleb, C. P. Quintero, and M. Jägersand. Tracking benchmark and evaluation for manipulation tasks. In *ICRA*, 2015.

[71] Glauco Garcia Scandaroli, Maxime Meilland, and Rogério Richa. Improving ncc-based direct visual tracking. In *ECCV*, 2012.

[72] Abhineet Singh and Martin Jagersand. Modular tracking framework: A fast library for high precision tracking. In *IROS*, 2017.

[73] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[75] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[76] Fei Wang and Baba C Vemuri. Non-rigid multi-modal image registration using cross-cumulative residual entropy. *IJCV*, 74(2):201–215, 2007.

[77] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *TIP*, 3(5):625–638, 1994.

[78] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019.

[79] Tao Wang and Haibin Ling. Gracker: A graph-based planar object tracker. *TPAMI*, 40(6):1494–1501, 2017.

[80] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020.

[81] Bowen Wen and Kostas Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *IROS*, 2021.

[82] Changsong Wen, Guoli Jia, and Jufeng Yang. Dip: Dual incongruity perceiving network for sarcasm detection. In *CVPR*, 2023.

[83] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *CVIU*, 193:102907, 2020.

[84] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics. *ECCVW*, 2020.

[85] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021.

[86] Z. Wu, J. Guo, S. Zhang, C. Zhao, and X. Ma. An ar benchmark system for indoor planar object tracking. In *ICME*, 2019.

[87] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[88] Bao Xin Chen and John Tsotsos. Fast visual object tracking using ellipse fitting for rotated bounding boxes. In *ICCVW*, 2019.

[89] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[90] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022.

[91] Jinyu Yang, Zhe Li, Song Yan, Feng Zheng, Aleš Leonardis, Joni-Kristian Kämäräinen, and Ling Shao. Rgbd object tracking: An in-depth review. *arXiv*, 2022.

[92] Shichao Yang and Sebastian Scherer. Monocular object and plane slam in structured environments. *IEEE Robotics and Automation Letters*, 4(4):3145–3152, 2019.

[93] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *CVPR*, 2020.

[94] Zhongguan Zhai, Shang Sun, and Junjie Liu. Tracking planar objects by segment pixels. In *IAECST*, 2021.

[95] Xinrui Zhan, Yueran Liu, Jianke Zhu, and Yang Li. Homography decomposition networks for planar object tracking. In *AAAI*, 2022.

[96] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *CVPR*, 2020.

[97] Yang Zhang, Hao Sheng, Yubin Wu, Shuai Wang, Weifeng Lyu, Wei Ke, and Zhang Xiong. Long-term tracking with deep tracklet association. *TIP*, 29:6694–6706, 2020.

[98] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022.

[99] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 2021.

[100] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Robust multi-object tracking by marginal inference. In *ECCV*, 2022.

[101] Liming Zhao, Xi Li, Jun Xiao, Fei Wu, and Yueting Zhuang. Metric learning driven multi-task structured output optimization for robust keypoint tracking. In *AAAI*, 2015.

[102] Zelin Zhao, Ze Wu, Yueqing Zhuang, Boxun Li, and Jiaya Jia. Tracking objects as pixel-wise distributions. In *ECCV*, 2022.

[103] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2018.

[104] Stefanie Zollmann, Denis Kalkofen, Erick Mendez, and Gerhard Reitmayr. Image-based ghostings for single layer occlusions in augmented reality. In *ISMAR*, 2010.