Rethinking the Implicit Segmentation of 3D Object Tracking under Vision Foundation Model

Jiachen Li[1,2], Rui Xing[3], Chunjing Wang[1,2,*], Mingle Zhou[1,2], Min Li[1,2], Gang Li[1,2], and Xueying Qin[4]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

[2]Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan, China

[3]Interdisciplinary Center, Shandong University, Jinan, China

[4]School of Software, Shandong University, Jinan, China

jcli@qlu.edu.cn wangchj@sdas.org

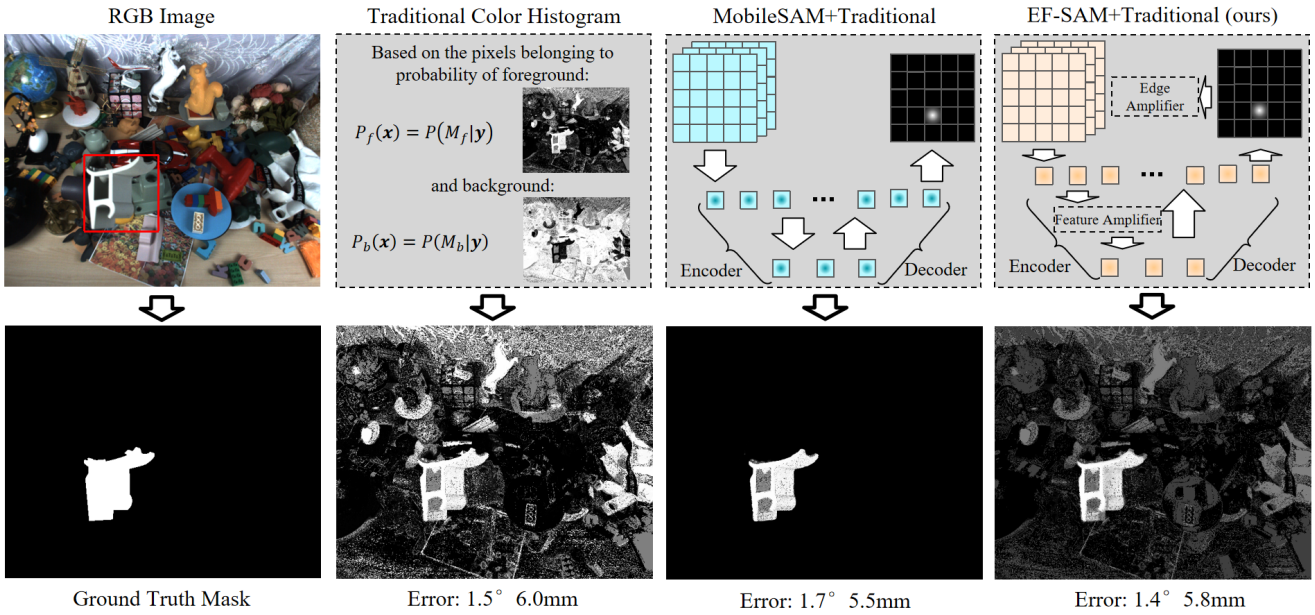

Figure 1: The relationship between the quality of the probability map obtained by implicit segmentation and the final tracking accuracy is still unclear in complex scenes. Sometimes a seemingly poor probability map may have higher precision than one that appears better. We propose Fuse-Tracker with the EF-SAM model to enhance 3D object tracking accuracy by improving the quality of the probability map.

Abstract

In 3D object tracking, the quality of the probability map obtained from implicit segmentation is the most critical factor affecting tracking accuracy. In this work, we found that in the task of 3D object tracking, vision foundation models can assist traditional color histograms in improving the prediction quality of probability maps. However, traditional color histograms cannot be replaced entirely, even though their visual results are unsatisfactory. In response to the characteristics of the 3D object tracking, we modify the SAM model to improve the precision of the predicted probability map, named Edge Filter SAM (EF-SAM). EF-SAM can accurately calculate the probability values near the object contours while improving the computational speed through the edge enhancement module and knowledge distillation. Subsequently, we use a concise fusion framework to integrate the probability maps predicted by EF-SAM and traditional color histograms. Based on the fused probability map, we estimate the ob-

*Corresponding Author

ject pose using traditional analytical optimization methods. In this way, we combine the advantages of feature extraction from vision foundation models and the precision advantages of traditional optimization methods. In the experiment, we compare our method with other 3D object tracking methods on the BCOT and RBOT datasets and test the performance of 4 variants of SAM in the 3D object tracking task. The results show that the proposed method achieves SOTA results (71.6% vs 70.4% on BCOT dataset).

Keywords: 3D Object Tracking, Implicit Segmentation, Pose Estimation

1. Introduction

Template-based monocular RGB 3D object tracking aims to estimate the 6DoF (Degree of Freedom) pose of an object relative to the camera in consecutive frames, given the CAD model. The high precision and speed characteristics of 3D object tracking make it a key technology in practical applications [54], widely used in video assembly, human-computer interaction, and robotics.

The core step of template-based texture-less 3D object tracking is the implicit segmentation of the object foreground. It does not directly segment the foreground object but calculates the probability of image pixel points belonging to the foreground and background. Then, the object pose is estimated by maximizing the probabilities of the foreground and background regions. However, the intrinsic connection between the segmentation quality and final tracking accuracy remains unclear. As shown in Fig. 1, the implicit segmentation result with poor visual quality (obtained by calculating the color histogram [30]) can achieve good accuracy. On the other hand, the segmentation result with good visual quality (predicted by MobileSAM [48]) is not optimal. The implicit segmentation results in the figure are displayed using the probability map of the foreground.

Recently, the proposal of the vision foundation model of segmentation [17] has shown the potential to replace traditional probability maps. However, vision foundation models usually focus on the extraction of general visual features and semantic features. For 3D object tracking, the features around the object contour are the key to tracking accuracy. Existing vision foundation models of segmentation do not pay special attention to this point. On the other hand, after we obtain good features using vision foundation models, appropriate solutions are required to combine the extracted feature with the optimization methods in tracking.

Based on the abovementioned problems, we propose the Edge Filter SAM (EF-SAM) model with an edge enhancement module and knowledge distillation to calculate the accurate probability map around the object contour. Then, we propose a concise fusion strategy (Fuse-Tracker) and rethink how implicit segmentation impacts 3D object tracking and improves accuracy. Our contributions are summarized as follows:

- We propose a high-efficiency and high-precision Edge Filter SAM (EF-SAM) model, which uses an edge enhancement module to improve the accuracy of probability value in the regions near the object contours. Meanwhile, we employ knowledge distillation to enhance the model's computational speed while maintaining performance.
- We use a concise fusion method (called Fuse-Tracker) of the probability maps for traditional color histograms and vision foundation models of segmentation, and combine it with the traditional optimization method, greatly enhancing the tracking accuracy of 3D objects.
- We conduct a detailed analysis of the impact of probability maps obtained from 4 SAM variants on tracking accuracy. Our method achieves SOTA results on both the BCOT and RBOT datasets.

2. Related Work

Our focus is on texture-less 3D object tracking based on monocular RGB data. Around the research topic, we will mainly introduce 3D object tracking, vision foundation models of segmentation and the application of 3D object tracking.

2.1. 3D Object Tracking

For 3D object tracking of textureless objects, keypoint extraction is ineffective, allowing us to rely only on edge or color features. Essentially, both methods align the object's projected contour with the image's contour to optimize the pose. The difference is that edge-based methods [15, 7, 34, 36] explicitly find correspondences between edge points, while color-based methods (also named region-based methods) [25, 37, 31, 14, 32] implicitly search for object contours by maximizing foreground-background color differences. Additionally, some methods fuse multiple features [52, 19, 30, 51] to solve for object poses.

Another branch is the learning-based methods [22, 21, 6, 16, 47], which use deep neural networks to extract features and optimize poses. This kind of method requires a large amount of training data. Even though rendering can generate a mass of synthetic data, the domain gap between synthetic and real data still exists, leading to overfitting and weak generalization performance of the network. Furthermore, the network's deep features and structure still need more scientific explanation. Thus, despite the advantages of learning-based methods in feature extraction, their accuracy, speed, and generalization performance still need to

catch up to optimization methods based on traditional features due to the lack of a reasonable way to utilize features. However, deep networks have been widely used for pose estimation based on the single frame [24, 10], as they do not require very high precision.

There are also methods based on the depth camera [35, 33, 44, 23] and multi-view cameras [18], which exhibit higher tracking accuracy but have limited applicability.

2.2. Vision Foundation Model of Segmentation

Unlike traditional neural network-based segmentation methods, the Segment Anything Model (SAM) [17] is a prompt-based vision foundation model of segmentation based on the Vision Transformer (ViT) [8]. SAM can segment objects prompted by points or bounding boxes, or automatically identify all objects in an image and generate masks. The zero-shot generalization ability of SAM provides a new approach for calculating probability maps in implicit segmentation for 3D object tracking tasks. However, the large number of parameters in SAM makes it unsuitable for tasks like 3D tracking that require real-time performance.

Lightweighting SAM is a crucial research direction in the vision foundation model of segmentation. Among them, FastSAM [50] uses the instance segmentation branch of YOLACT [1] to replace SAM's image encoder branch, reducing its size and achieving a higher speed increase while retaining accuracy. MobileSAM [48] uses TinyViT [42] as the image encoder branch to reduce the number of parameters and uses knowledge distillation [12] on the encoder to decouple it from the decoder to improve accuracy while significantly increasing the computational speed. EdgeSAM [53] replaces the image encoder branch with a CNN branch and optimizes the promotion strategy. EfficientSAM [45] adopts a self-supervised MAE method [11] based on MobileSAM. In addition, some of the latest models [26, 49, 40, 46] have shown excellent results. Additionally, methods [4, 3, 5] present work on SAM distillation and its extended applications.

The 3D object tracking relies on the quality of implicit segmentation around the object's foreground. Although SAM series models have strong segmentation capabilities, they are still prone to incorrect segmentation results around the object's contour, significantly affecting tracking accuracy. Therefore, continued improvement is still needed for scenarios of 3D object tracking, especially the segmentation quality near the contour.

2.3. Applications of 3D Object Tracking

3D object tracking is widely used in robotics, augmented reality, and human-computer interaction. Method [39] uses 3D object tracking for video assembly guidance, interacting with users and give feedback. GBOT [20] further considers hand occlusion during object tracking and interaction. ASDF [27] associates pose with status, improving perception accuracy and interactive experience. On the other hand, SLAM [2] technology is also widely used in interaction, helping users perceive the scene and interact through real-time map construction and camera pose estimation. Regarding perceptual terminals, the Minilag Filter [28] uses a lightweight back update and compensation strategy to improve the visual performance of tracking.

For applications, pose estimation precision is the key to user experience. Although template-based 3D object tracking has the advantages of high speed and high precision, its tracking accuracy still needs to improve under complex conditions such as similar foreground and background colors, fast motion, and occlusion. We need to improve the accuracy further to enhance the user's experience in the human-computer interaction environment.

3. Method

This section will introduce the EF-SAM model and the Fuse-Tracker, both of which aim to improve the tracking accuracy of 3D objects.

3.1. Edge Filter SAM Model

The region around the object's contour has the most significant impact on the tracking accuracy of 3D objects [29], as region-based methods require the reprojected contour to be aligned as closely as possible with the implicit segmentation contour. However, general vision foundation models of segmentation do not usually pay special attention to the contours of objects. When the background is complex, directly using a vision foundation model to segment the foreground of the object often does not yield precise results in the contour regions.

We propose the EF-SAM model, further improving the SAM model's inference speed while ensuring the segmentation precision around the object contour. By implicitly segmenting the image and integrating the Fuse-Tracker in Sec. 3.2, EF-SAM model achieves a balance of speed and accuracy of the vision foundation model.

The framework is shown in Fig. 2. An effective and lightweight visual backbone is an important component in the segmentation task. We leverage the feature extractor of lightweight MaxViT-tiny [38] and integrate it into our EF-SAM model as encoder. During image processing, EF-SAM focuses on the quality of features near the object contour. We propose the edge enhancing module including feature amplifier and edge amplifier after the encoder to process features, enhancing the module's segmentation capabilities around the object contour regions and highlighting the edges of objects. In addition, EF-SAM retains the prompt-guided mask decoder from classical SAM [17] for object segmentation.

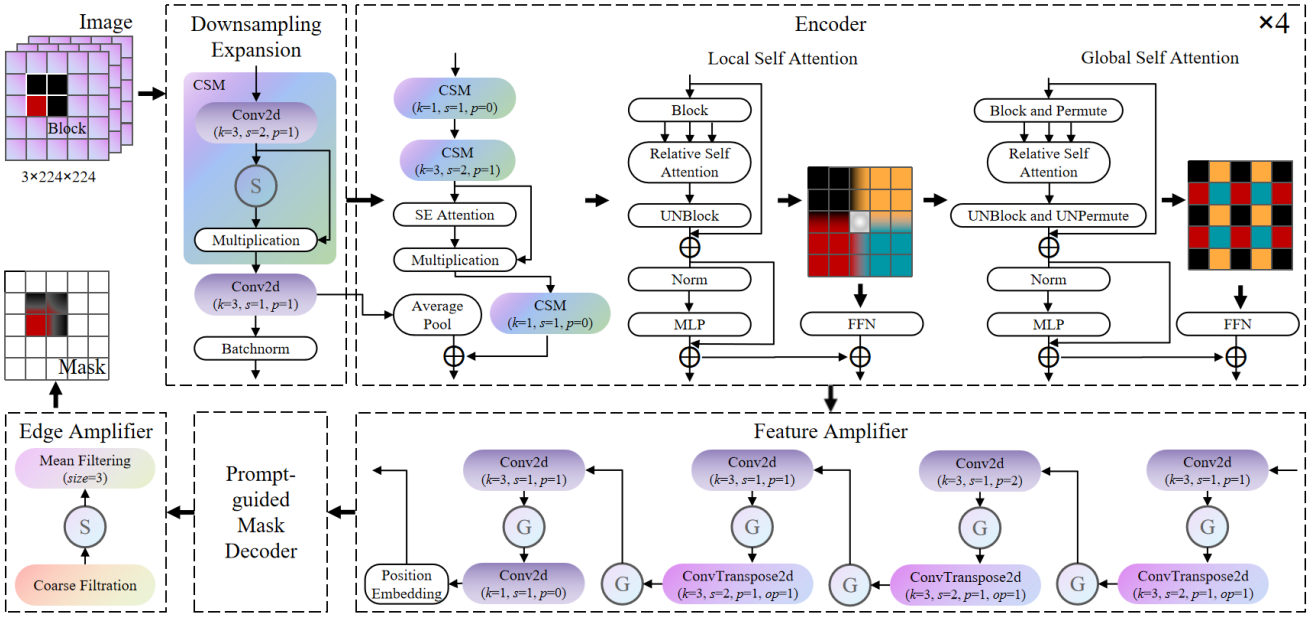

Figure 2: The EF-SAM model. The image sequentially passes through the Downsampling Expansion module and the MaxViT-tiny to obtain encoded features. Then, the Feature Amplifier and Edge Amplifier enhance the edge features near the object contours. Specifically, Block represents operations includes reshape, permute, and contiguous, UNBlock represents the reverse operations of Block, Relative Self-Attention represents the multi-head attention mechanism, FFN stands for the Feed Forward layer in the transformer, S represents the Sigmoid function, and G denotes the GELU activation function.

3.1.1 Encoder

Specifically, the encoder of the EF-SAM model first resizes the image to 3×224×224 through interpolate operation. Compared to the classic SAM, which adjusts to 3×1024×1024, our smaller-scale image resizing can greatly enhance segmentation speed. Correspondingly, this may also damage image information, thereby reducing segmentation quality. Therefore, given the premise of low-resolution input, enhancing the segmentation capability around the object contour regions and strengthening edge features are the main problems that EF-SAM will address.

First, the image is downsampled by the downsampling expansion module to a shape of C×112×112. Then, it enters the encoder of the network, which consists of 4 stages with identical structures. The resolution of each stage is half that of the previous stage, and the number of channels (hidden dimensions) is doubled compared to the previous stage. Each sub-stage undergoes downsampling through 2 CSM (Convolution + Sigmoid + Multiplication) modules. Following that, the SE (Squeeze-and-Excitation) Attention module [13] is applied to enhance the network's generalization and trainability. One CSM module with a kernel size of 1 is then used for relative positional encoding, replacing the explicit positional encoding layer.

Afterward, the feature passes through the local and global self-attention modules. The local self-attention module uses the Block operation for windowing and performs local self-attention computation. After restoration using the UNBlock operation, the FFN (Feed Forward) layer independently transforms the features at each position, enhancing the model's expressive power and allowing the network to learn more complex feature representations. The global self-attention module builds upon the local self-attention module by using the permute operation to enable global interaction among the windowed features, where other modules are the same.

After passing through the local and global self-attention modules, the output shape is 2C×56×56. Similarly, the outputs of the remaining 3 stages are 4C×28×28, 8C×14×14, and 16C×7×7, respectively. By computing locally and globally on different spatial scales, the computational complexity is reduced while ensuring its effectiveness.

3.1.2 Edge Enhancement

To address the issue of inaccurate segmentation in regions around object contours, we propose the feature amplifier and edge amplifier module, which enhances the probability maps around object contours based on filters.

Feature Amplifier. Initially, the feature amplifier re-

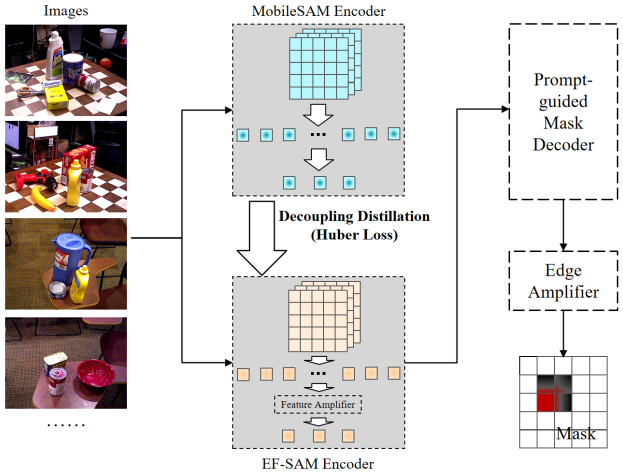

Figure 3: Decoupled knowledge distillation process. We guide MaxViT-tiny with the encoder of MobileSAM to obtain a lightweight student network.

ceives the output from the encoder and undergoes feature amplification through 3 Conv2d, GELU, and ConvTranspose2d operations to obtain a shape of 16C×64×64. By mapping operations, high-dimensional feature information is obtained, ensuring that edge information is not lost. Subsequently, Conv2d, GELU, and Conv2d operations are applied to reduce the number of channels, resulting in a shape of 256×64×64. Channel reduction removes redundant information, ensuring the lightweight nature of the network.

It is worth noting that, considering the need for pretraining, relying solely on 4 CSM modules with a kernel size of 1 to replace the position encoding layer is insufficient. We perform a random position embedding operation, which serves as a trainable parameter that participates in gradient optimization, so the word vector will also be generated when the model training is completed. Finally, the encoded features enter the traditional SAM's prompt-guided mask decoder for decoding, resulting in an output shaped as the original size.

Edge Amplifier. Next, the edge amplifier performs filtering on the probability maps. We first use a simple step function for coarse filtering of the probability maps, followed by a sigmoid activation to normalize their probability distribution. After normalization, the probability maps undergo mean filtering, improving the filtering effect and reducing computational complexity.

Furthermore, we perform an additional threshold operation on the probabilities when normalization, setting all probabilities less than 0.5 to 0.5, resulting in a threshold distribution of probabilities ranging from [0.5-1.0]. This is because during feature extraction the network undergoes several downsampling and upsampling steps, so probabilities near object edges (especially on the edges) tend to converge to 0.5. On the other hand, the color itself on the edges already blends the foreground and background probabilities due to the imaging principle of the camera. That is, edge points should inherently belong to both the foreground and background equally. Therefore, we mimic this process through thresholding. Although this process may affect regions with probabilities less than 0.5, it does not affect the overall probability distribution. Additionally, the Fuse-Tracker in Sec. 3.2 will fuses the probability map predicted by EF-SAM model with the probability map obtained from traditional color histograms, which can also reduce the impact of these regions.

3.1.3 Knowledge Distillation

The SAM [17] mentions that training SAM-ViT-H requires 256 A100 GPUs and takes up to 68 hours. Even with an Encoder of ViT-B, 128 GPUs are needed. Such a high consumption of resources undoubtedly hinders researchers from reproducing or improving the model. Therefore, for the training process of the EF-SAM model, we are inspired by studies such as MobileSAM [48], EdgeSAM [53], and EfficientSAM [45], and adopt a decoupled distillation technique.

Fig. 3 illustrates the training process of the EF-SAM model. First, the teacher model of MobileSAM and the student model of EF-SAM encoder are loaded. The input for MobileSAM is $b\times3\times1024\times1024$, and for EF-SAM model, it is $b\times3\times224\times224$, where b represents the batch size. Then, the dataset is loaded and input simultaneously into MobileSAM and EF-SAM model. MobileSAM obtains image embedding in the same way as the classical segment anything model [17]. EF-SAM model will use the Huber function [9] to calculate the loss and gradient backpropagate with the image embedding. The Huber function is applied to the feature maps, and the distillation method is feature-based distillation.

3.2. Fuse-Tracker

Deep learning, especially the vision foundation model, is well known to excel in feature extraction. However, due to the high degree of freedom and nonlinear nature of the pose, the results estimated by directly regressing are usually not precise enough [33]. Therefore, combining the feature extraction advantages of vision foundation models with the precision advantages of traditional analytical optimization is a direct and effective strategy.

We draw inspiration from region-based optimization methods [25], which transform color features into probability maps. These probability maps describe the probability of image pixels belonging to the foreground or background. Then, the object pose is solved by maximizing the difference between the foreground and background colors. For-

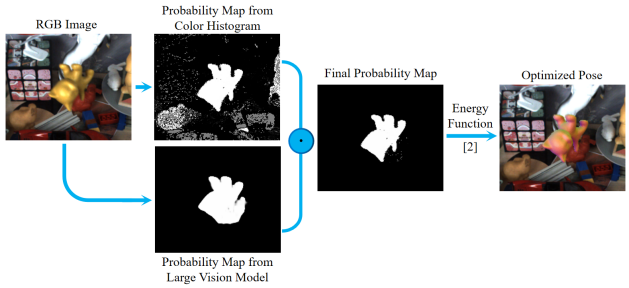

Figure 4: The framework of the proposed Fuse-Tracker. The Fuse-Tracker integrates the advantages by multiplying the two types of probability maps. It then estimates the pose based on the traditional analytical optimization method [30], combining the feature extraction advantages of vision foundation models with the precision advantages of analytical optimization.

tunately, the segmentation problem naturally produces the required probability map, providing an effective way to utilize the features of vision foundation models and serving as the intersection with analytical optimization.

As shown in the Fig. 4, we can obtain the probability map in two ways: the traditional color probability histogram ($\boldsymbol{P}_t$) and the vision foundation model of segmentation ($\boldsymbol{P}_l$). Next, we integrate them by multiplying every corresponding image point:

$$\boldsymbol{P} = \boldsymbol{P}_t \odot \boldsymbol{P}_l \tag{1}$$

Please note that this is a very straightforward method. We choose to integrate the two probability map types because each has unique advantages.

For the probability map obtained from the traditional color histogram [30] we use here, although it is weaker in feature expression, it is computationally fast and can ensure real-time updates for each frame. The update strategy is as follows:

$$p^k(\boldsymbol{y} \mid m) = 0.2p(\boldsymbol{y} \mid m) + 0.8p^{k-1}(\boldsymbol{y} \mid m) \tag{2}$$

k is the frame index, m represents the color model of the foreground or background, and $\boldsymbol{y}$ represents the pixel color. p represents the probability of color $\boldsymbol{y}$ occurring for the given color model m. During each update, 80% of the historical information is retained. By updating the histogram in real-time, it is possible to ensure the capture of the real-time characteristics of the image sequence in an unknown scene.

The advantage of the probability map from the vision foundation model lies in its combination of rich prior knowledge and zero-shot generalization capability. However, semantic segmentation errors are prone to appear in some complex scenes, especially around object contours, as shown in the Fig. 4.

$\boldsymbol{P}_t$ and $\boldsymbol{P}_l$ exhibit complementary characteristics in feature representation. $\boldsymbol{P}_l$ encodes the prior knowledge embedded in the pre-trained model, whereas $\boldsymbol{P}_t$ updates the conventional color-histogram features on-the-fly using the current frame. Consequently, the two types of features are statistically complementary. Integrating the two types of probability maps can combine the advantages of the real-time update of the color histogram probability map and the rich prior knowledge of the vision foundation model. Eq. 1 multiplies the two probability maps, which means only when the probabilities of both types of probability maps are high enough that the point can truly be determined as the foreground. Therefore, it can avoid one type of feature failing in complex scenes.

In addition, we can also directly use the probability map from the vision foundation model to replace the one from the traditional color histogram or calculate the average of the two, as shown as:

$$\boldsymbol{P} = \boldsymbol{P}_l \tag{3}$$

$$\boldsymbol{P} = 0.5 \times (\boldsymbol{P}_t + \boldsymbol{P}_l) \tag{4}$$

In the experimental section, we will present the results of each fusion method and demonstrate many interesting conclusions.

Next, we solve the pose using the energy function of method [30], which includes the region term, feature point term, and depth term. The depth energy term is not considered since we only use RGB input images here.

In this way, we integrate the probability map of the traditional color histogram and the vision foundation model, and the analytical optimization method, fully leveraging their characteristics to enhance tracking accuracy.

4. Experiment

In the experiments, we use a desktop computer with an Intel(R) Core(TM) i9-13900 @3.0GHz CPU, NVIDIA GeForce RTX 4090 GPU, and 64GB RAM.

4.1. Datasets and Metrics

BCOT Dataset. The BCOT [18] dataset is a real-world dataset where the ground-truth poses are annotated using a binocular tracking method. It contains 20 scenes, 22 objects, 404 sequences, and 126K images. The evaluation metrics include the n° $n$cm metric (n = 2, 5) and the ADD metric (with thresholds of 0.02d, 0.05d, and 0.1d, where d represents the longest side of the object's bounding box). The ground-truth pose is reset when the tracking error exceeds 5°, or the translation error exceeds 5cm. Additionally, the BCOT dataset provides binocular view data, allowing for observing results from different viewpoints.

RBOT Dataset. The RBOT dataset [37] is a synthetic dataset that provides accurate ground-truth poses. It contains 18 objects, 4 variants, 72 sequences and 72K images.

Table 1: Comparison of the BCOT dataset [18]. For the compared methods, * represents the results by running the source code provided by the authors, while the rest are the results in the paper [18].

Method	ADD-0.02d ↑	ADD-0.05d ↑	ADD-0.1d ↑	5°, 5cm ↑	5° ↑	5cm ↑	2°, 2cm ↑	2° ↑	2cm ↑	Avg. ↑
RBOT [37]	11.7	31.6	57.1	77.1	79.2	91.7	40.8	48.3	67.8	56.1
RBGT [31]	10.9	45.5	76.9	89.0	89.3	99.5	46.0	49.5	87.8	66.0
SLOT [14]	**15.6**	39.8	66.1	87.1	88.5	96.3	51.4	**59.0**	76.4	64.5
SRT3D [32]*	12.5	49.4	82.1	93.1	93.2	**99.9**	53.6	56.6	91.9	70.3
ICG+ [30]*	12.5	48.4	81.5	**93.8**	**93.9**	**99.9**	**54.5**	57.5	91.8	70.4
Ours	14.8	**54.7**	**85.1**	92.9	93.0	99.8	54.2	56.0	**93.6**	**71.6**

The evaluation metric is based on the 5° 5cm, where tracking is considered correct if the rotation error is less than 5° and the translation error is less than 5cm in the current frame; otherwise, it is deemed incorrect, and the ground-truth pose is reset. The percentage of correctly tracked frames calculates the final accuracy.

4.2. Implementation Details

The use of SAM models. We test many SAM models in the experiments, including our proposed EF-SAM model. Unless otherwise specified, we use the bounding box as the prompt when using these SAM models.

Obtaining the bounding box. Given the known initial pose of the 3D object tracking, we can render the object and acquire a foreground mask. Subsequently, we calculate a tightly bounding box based on this mask. However, it is important to note that directly utilizing the initial pose for rendering may introduce deviations in the resulting bounding box. Therefore, we first use the pre-search strategy of method [29] to correct the pose along the X and Y axes, obtaining a more accurate bounding box.

Training Data of Knowledge Distillation. During the training of the EF-SAM model, we select the test set of the classic 3D object tracking dataset YCB-Video [43] as the training data, totaling approximately 20K images. The reason for choosing the YCB-Video is that it contains both simple and complex objects and uses different object combinations with scenes, representing the typical data characteristics of 3D object tracking.

Additionally, the final testing of our method is conducted on the BCOT and RBOT datasets, using only the YCB-Video dataset for training, demonstrating the generalization ability of the proposed method.

Probability map normalization. We represent the probability of a point belonging to the foreground and background with the foreground probability map and background probability map, respectively. In the experimental section, unless specifically stated otherwise, EF-SAM normalizes the probability map to [0.5-1.0] as mentioned in Sec. 3.1.2, while traditional color histograms normalizes the probability map to [0-1.0].

Table 2: Comparison of the RBOT dataset [37]. For the compared methods, * represents the results by running the source code provided by the authors, while the rest are the results in their original paper.

Method	Reg. ↑	Dyn. ↑	Noi. ↑	Occ. ↑	Avg. ↑
RBOT [37]	79.9	81.2	56.6	73.3	72.8
RBGT [31]	90.0	90.6	71.5	85.6	84.4
SLOT [14]	89.9	90.7	69.6	88.9	84.8
SRT3D [32]	94.2	94.6	81.7	93.2	90.9
NLT [36]	95.2	95.4	83.2	**94.9**	92.2
TRM [29]	95.4	94.9	86.2	93.2	92.4
ICG+ [30]*	96.3	96.1	86.5	93.3	93.0
DeepAC [41]	95.6	95.6	88.0	94.0	93.3
Ours	**96.6**	**96.0**	**89.3**	93.1	**93.8**

4.3. Quantitative Results

4.3.1 BCOT dataset

Table 1 presents the tracking results on the BCOT dataset [18]. We first compare traditional methods, including RBOT [37], RBGT [31], SLOT [14], SRT3D [32], and ICG+ [30], where ICG+ [30] only uses RGB images as input. All these methods use traditional color histograms for implicit segmentation of objects, calculating the probability of each pixel in the image belonging to the foreground and background, and then optimizing the pose using traditional methods. Among these methods, RBOT [37] and SLOT [14] use local color histograms, while RBGT [31], SRT3D [32], and ICG+ [30] use global histograms. Among these methods, ICG+ [30] used local and global uncertainty during the optimization process, showing the best results.

Compared to traditional methods, our method has achieved optimal results, with an average score of 71.6%. The pose optimization part of our method is consistent with ICG+ [30], which demonstrates the effectiveness of the probabilistic map computation in our method. In addition, we find that our method performs better in the translation component, with more significant improvements in metrics of ADD-n and 2cm, indicating higher optimization precision. This is attributed to its handling of edge features around the object contours, confirming that object contours

Table 3: Comparison of accuracy on the BCOT dataset [18] by integrating different SAM models. The time refers to the total time of the algorithm, including the time for probability map prediction and pose optimization.

Method	ADD-0.02d ↑	ADD-0.05d ↑	ADD-0.1d ↑	5°, 5cm ↑	5° ↑	5cm ↑	2°, 2cm ↑	2° ↑	2cm ↑	Avg. ↑	Time (ms) ↓
FastSAM [50]	14.7	52.7	81.3	87.0	87.1	99.5	42.7	44.2	91.2	66.7	30
EdgeSAM [53]	10.1	41.8	76.6	91.2	91.4	99.5	49.1	52.5	89.0	66.8	20
EdgeSAM (3x) [53]	9.4	38.9	74.6	91.3	91.5	99.5	48.6	52.4	88.0	66.0	20
EfficientSAM (t) [45]	10.9	44.5	80.6	92.9	92.9	**99.8**	53.6	56.3	91.8	69.3	30
EfficientSAM (s) [45]	12.6	50.1	85.0	**93.2**	**93.3**	**99.8**	**55.6**	**57.3**	**94.2**	71.2	55
MobileSAM [48]	12.7	50.2	84.3	92.6	92.6	**99.8**	54.1	55.9	93.6	70.6	20
EF-SAM (ours)	**14.8**	**54.7**	**85.1**	92.9	93.0	**99.8**	54.2	56.0	93.6	**71.6**	**13**

Table 4: Ablation study of the normalization way of the BCOT dataset [18].

Method	Scope	ADD-0.02d ↑	ADD-0.05d ↑	ADD-0.1d ↑	5°, 5cm ↑	5° ↑	5cm ↑	2°, 2cm ↑	2° ↑	2cm ↑	Avg. ↑
EF-SAM (ours)	[0.5, 1.0]	**14.8**	**54.7**	**85.1**	**92.9**	**93.0**	**99.8**	**54.2**	**56.0**	**93.6**	**71.6**
EF-SAM (ours)	[0, 1.0]	10.3	42.6	74.8	87.8	88.1	99.1	44.6	47.9	86.2	64.6

Table 5: Ablation study of the fusion way of the BCOT dataset [18].

Method	Fusion way	ADD-0.02d ↑	ADD-0.05d ↑	ADD-0.1d ↑	5°, 5cm ↑	5° ↑	5cm ↑	2°, 2cm ↑	2° ↑	2cm ↑	Avg. ↑
EF-SAM (ours)	Eq. 1	**14.8**	**54.7**	**85.1**	**92.9**	**93.0**	**99.8**	**54.2**	**56.0**	**93.6**	**71.6**
EF-SAM (ours)	Eq. 3	4.3	20.7	45.7	61.3	63.1	96.1	13.9	19.0	59.1	42.6
EF-SAM (ours)	Eq. 4	4.4	21.2	45.9	63.1	64.8	96.1	14.6	20.1	59.1	43.3

are a critical factor in influencing accuracy, as mentioned in the paper [29]. On the other hand, our method is not as good as traditional methods in rotation optimization. This is because traditional methods propose refined processing strategies for occlusion and spatial scale, which our method has not yet carefully considered. These will be the directions for future improvement.

4.3.2 RBOT Dataset

Table 2 shows the comparison results on the RBOT dataset [37]. We further compare our method with two other traditional methods, namely NLT [36] and TRM [29]. NLT [36] is an edge-based method that employs an out-of-plane search strategy to enhance tracking accuracy under large translation conditions. TRM [29] is a feature fusion-based method that integrates region features and point features. The average accuracy ranking on the RBOT dataset [37] is basically consistent with the BCOT dataset [18], and our method still achieves the optimal results.

In the occlusion variant, our method does not have a significant advantage, as we have not adopted any strategies related to occlusion. In the other three variants, our method achieves optimal results, indicating that the probability map computed by our method can adapt to regular, dynamic light, and noisy scenarios and exhibits good generalization performance.

DeepAC [41] is a learning-based method. It is similar to our proposed method in that it uses a deep neural network to compute features and then employs traditional optimization methods to solve for object pose. However, DeepAC differs in that it utilizes the network to predict a boundary map and then optimizes the pose based on this edge feature. Experimental results show that DeepAC performs slightly worse than our proposed method. More importantly, its extensibility is weaker. Specifically, our method utilizes the probability map as the integration point with traditional pose optimization, allowing it to be combined with any segmentation model, thereby offering stronger extensibility. In the future, it can be combined with more powerful segmentation models without requiring modifications to the tracking base.

4.4. Ablation Study

4.4.1 Combined with Different SAM Model

The Fuse-Tracker we proposed can integrate the probability map of traditional color histograms with the probability map of EF-SAM. Due to its concise fusion approach, it can also be integrated with other SAM models. In this section, we integrate the traditional probability map with the probability map predicted by 4 other SAM models and test the tracking results, including FastSAM [50], EdgeSAM [53], EfficientSAM (t) [45], and MobileSAM [48]. The experimental results are shown in Table 3, where the Method column indicates the different SAM models used for fusion.

Our method achieves the highest average accuracy, demonstrating the effectiveness of EF-SAM's probability map prediction. In comparison, the accuracy of EfficientSAM (s) [45] is only 0.4% lower. However, it employs a more complex backbone network, with a total inference

time of 55ms per frame, which is significantly slower than EF-SAM. Its base version, EfficientSAM (t) [45], has an inference time of 30ms per frame, but its accuracy is only 69.3%. The performance of the other SAM models is inferior to EF-SAM, as shown in the table.

Runtime. The runtime of our method mainly consists of two parts, and the time shown in Table 3 represents the total time. The first part is the prediction time of the probability map for EF-SAM, which is approximately 8ms. The second part is the pose optimization time, which is approximately 5ms. EF-SAM not only achieves the highest accuracy but also the fastest speed, ensuring its real-time capability.

4.4.2 Normalization Ways

For the task of 3D object tracking, the accuracy of the probability map in the regions near the object's contour has a greater impact on tracking precision. As described in Sec. 3.1.2, we limit the probability range to [0.5-1.0] to suppress the influence of inaccurate probability values around the contour.

Table 4 shows the quantitative effect of this module. Limiting the probabilistic graph to [0.5, 1.0] can stabilize the overall probability distribution. During the probability normalization process, this operation ensures that the lower limit of the probability distribution will not be out of range. Although this operation may affect some regions with lower probability values, the impact of these regions on the overall probability distribution is relatively small. This is because the main part of the foreground probability distribution usually concentrates on the higher probability value regions. In comparison, the lower probability value regions often only account for a smaller proportion. Therefore, by limiting the range of the probability map, it can effectively avoid the instability of the overall probability distribution caused by a few low-probability regions, thereby enhancing the stability and reliability of the tracking.

Moreover, the fusion mechanism of Fuse-Tracker (Eq. 1) can further reduce the impact of these regions on overall tracking performance. This fusion mechanism can not only compensate for the shortcomings of a single probability map but also further optimize the tracking results through weight adjustment. Therefore, even when some low-probability regions are limited, the fusion mechanism can still ensure the improvement of overall tracking performance, allowing the tracking system to maintain high accuracy and stability in complex scenarios.

4.4.3 Fusion Ways

As described in the method section, we use Eq. 1 to fuse the probability maps of the traditional color histogram and the vision foundation model. Here, we will test other fusion ways, including using only the probability map obtained from the vision foundation model (Eq. 3) and the average of the two probability maps (Eq. 4).

Table 5 shows the test results of different fusion ways on the BCOT dataset. We found that the EF-SAM model experiences a significant performance drop when using other fusion methods. This is because the normalization in the EF-SAM model scales the probability maps to the range of [0.5, 1.0]. As a result, the probability distribution after fusion no longer falls within the [0, 1.0], leading to degraded results. The results further illustrate the advantage of the pointwise multiplication fusion way (Eq. 1), as it can always maintain a reasonable probability distribution.

4.5. Qualitative Results

Fig. 5 shows the predicted masks and final tracking results during the tracking process. The second column of Fig. 5 displays the masks predicted by ICG+ [30] and integrated with each SAM model, while the third column shows the magnified results. Visually, there are only minor differences in the masks after fusion for each method. However, we can still find that the results of EF-SAM near the object contours are somewhat blurred, which is the result of our targeted processing and has achieved higher accuracy in quantitative results. The last column shows the tracking results.

5. Limitations

We propose the EF-SAM model and Fuse-tracker, a concise yet effective fusion framework that combines the feature extraction advantages of vision foundation models with the precision advantages of traditional analytical optimization. However, this paper still has some limitations.

The analysis of the probability map quality is mainly qualitative. In the problem of 3D object tracking, how to quantify the quality of the probability map is an issue that needs to be further addressed. On the other hand, regarding the 3D object tracking problem, when an object is occluded, the distribution of the probability map will inevitably be affected. The proposed method does not analyze or address occlusion, which is an important direction for our future work.

6. Conclusion

The proposed EF-SAM model can improve the precision of probability predictions while ensuring interface speed, and the Fuse-Tracker can greatly enhance the tracking accuracy of 3D objects. Through in-depth analysis and contemplation of this paper, we can draw the following main conclusions:

1) Although vision foundation models show outstanding advantages in image feature extraction, they still cannot completely replace traditional color histograms in the

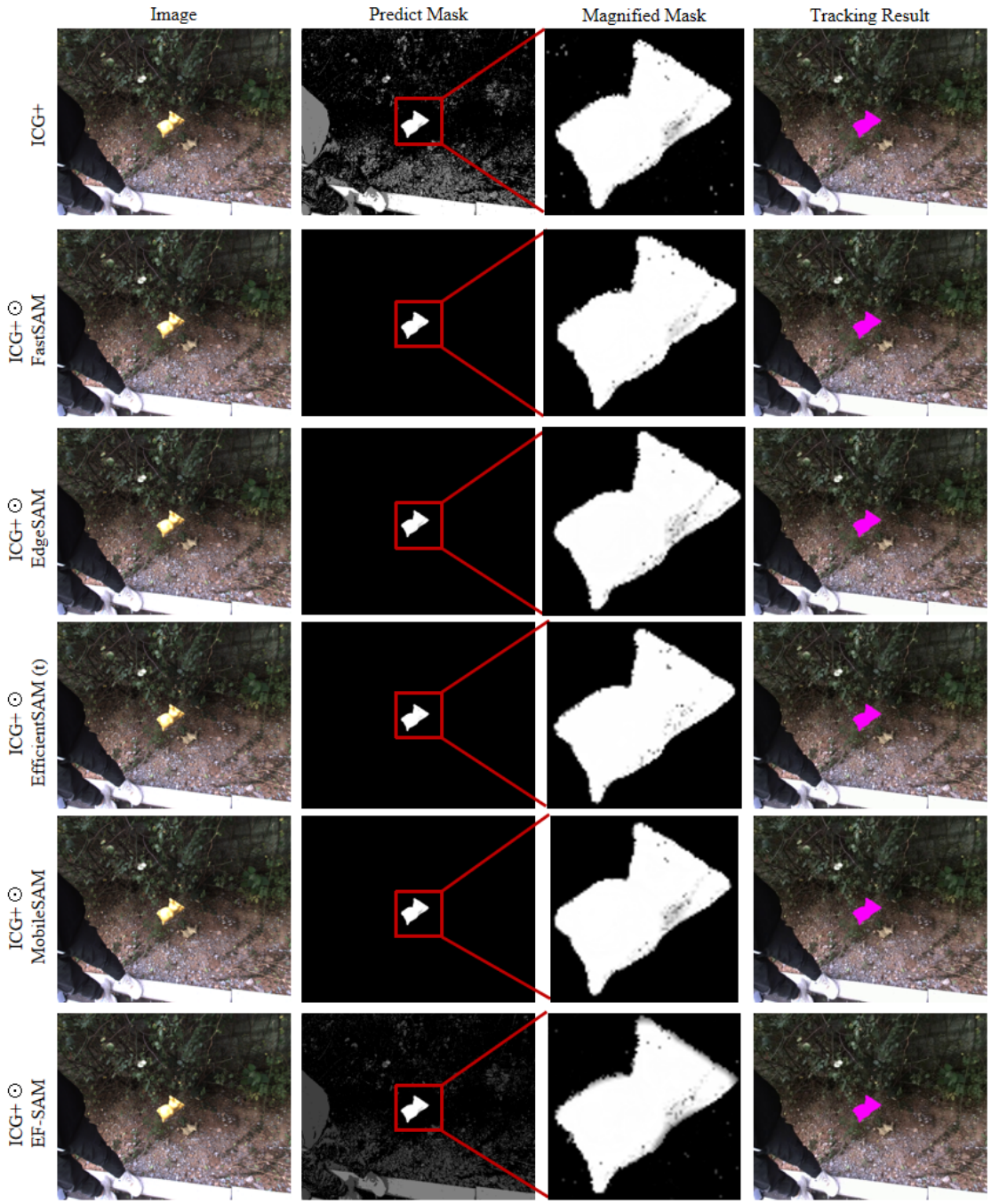

Figure 5: Mask prediction and tracking results of the BCOT dataset [18].

problem of 3D object tracking. Color histograms can still complement the vision foundation models.

2) From a visual effect perspective, good probability map segmentation results do not have an obvious correlation with the final tracking quality.

3) The fusion way of point-wise multiplication can fully exert the advantages of traditional color histograms and vision foundation models. In contrast, using the average fusion way or a single probability map will result in worse performance.

In future work, we will improve upon the limitations of this paper to further enhance the accuracy of 3D object tracking. Additionally, we will investigate end-to-end pose estimation methods and explore the potential of vision foun-

dation models in pose regression.

Acknowledgement

This work was supported in part by the Key R&D Program of Shandong Province, China (No. 2025CXGC010111), the Qilu University of Technology (Shandong Academy of Sciences) School of Computer Science and Technology Pairing Program (No. 2024JDJH04), the Taishan Scholars Program (No. tsqnz20240834), and the project ZR2024QF286 supported by Shandong Provincial Natural Science Foundation.

References

[1] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. YOLACT: real-time instance segmentation. In *ICCV*, pages 9156–9165. IEEE, 2019. 3

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Robotics*, 37(6):1874–1890, 2021. 3

[3] T. Chen, L. Zhu, C. Ding, R. Cao, Y. Wang, Z. Li, L. Sun, P. Mao, and Y. Zang. SAM fails to segment anything? - sam-adapter: Adapting SAM in underperformed scenes: Camouflage, shadow, and more. *CoRR*, abs/2304.09148, 2023. 3

[4] T. Chen, L. Zhu, C. Ding, R. Cao, Y. Wang, S. Zhang, Z. Li, L. Sun, Y. Zang, and P. Mao. SAM-adapter: Adapting segment anything in underperformed scenes. In *ICCV (Workshops)*, pages 3359–3367. IEEE, 2023. 3

[5] Z. Chen, Z. Zhu, Y. Zhang, J. Hou, G. Shi, and J. Wu. Segment any events via weighted adaptation of pivotal tokens. *arXiv preprint arXiv:2312.16222*, 2023. 3

[6] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. PoseRBPF: A rao-blackwellized particle filter for 6D object pose tracking. *IEEE Trans. Robotics*, 37(5):1328–1342, 2021. 2

[7] Y. Dong, L. Ji, S. Wang, P. Gong, J. Yue, R. Shen, C. Chen, and Y. Zhang. Accurate 6DoF pose tracking for texture-less objects. *IEEE Trans. Circuits Syst. Video Technol.*, 31(5):1834–1848, 2021. 2

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 3

[9] K. Gökcesu and H. Gökcesu. Generalized huber loss for robust learning and its efficient minimization for a robust statistics. *CoRR*, abs/2108.12627, 2021. 5

[10] Y. Hai, R. Song, J. Li, and Y. Hu. Shape-constraint recurrent flow for 6D object pose estimation. In *CVPR*, pages 4831–4840. IEEE, 2023. 3

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988. IEEE, 2022. 3

[12] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 3

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017. 4

[14] H. Huang, F. Zhong, and X. Qin. Pixel-wise weighted region-based 3D object tracking using contour constraints. *IEEE Trans. Vis. Comput. Graph.*, 28(12):4319–4331, 2022. 2, 7

[15] H. Huang, F. Zhong, Y. Sun, and X. Qin. An occlusion-aware edge-based method for monocular 3D object tracking using edge confidence. *Comput. Graph. Forum*, 39(7):399–409, 2020. 2

[16] S. Iwase, X. Liu, R. Khirodkar, R. Yokota, and K. M. Kitani. RePOSE: Fast 6D object pose refinement via deep texture rendering. In *ICCV*, pages 3283–3292. IEEE, 2021. 2

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. B. Girshick. Segment anything. In *ICCV*, pages 3992–4003. IEEE, 2023. 2, 3, 5

[18] J. Li, B. Wang, S. Zhu, X. Cao, F. Zhong, W. Chen, T. Li, J. Gu, and X. Qin. BCOT: A markerless high-precision 3D object tracking benchmark. In *CVPR*, pages 6687–6696. IEEE, 2022. 3, 6, 7, 8, 10

[19] J. Li, F. Zhong, S. Xu, and X. Qin. 3D object tracking with adaptively weighted local bundles. *J. Comput. Sci. Technol.*, 36(3):555–571, 2021. 2

[20] S. Li, H. Schieber, N. Corell, B. Egger, J. Kreimeier, and D. Roth. GBOT: graph-based 3D object tracking for augmented reality-assisted assembly guidance. In *VR*, pages 513–523. IEEE, 2024. 3

[21] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6D pose estimation. *Int. J. Comput. Vis.*, 128(3):657–678, 2020. 2

[22] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6D pose refinement in RGB. In *ECCV (14)*, volume 11218 of *Lecture Notes in Computer Science*, pages 833–849. Springer, 2018. 2

[23] J. Nie, A. Xu, Z. Bao, Z. He, X. Lv, and M. Gao. Context matching-guided motion modeling for 3D point cloud object tracking. *IEEE Trans. Circuits Syst. Video Technol.*, 35(3):2289–2300, 2025. 3

[24] S. Peng, X. Zhou, Y. Liu, H. Lin, Q. Huang, and H. Bao. PVNet: Pixel-wise voting network for 6DoF object pose estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(6):3212–3223, 2022. 3

[25] V. A. Prisacariu and I. D. Reid. PWP3D: real-time segmentation and tracking of 3D objects. *Int. J. Comput. Vis.*, 98(3):335–354, 2012. 2, 5

[26] N. Ravi, V. Gabeur, Y. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C. Wu, R. B. Girshick, P. Dollár, and C. Feichtenhofer. SAM 2: Segment anything in images and videos. *CoRR*, abs/2408.00714, 2024. 3

[27] H. Schieber, S. Li, N. Corell, P. Beckerle, J. Kreimeier, and D. Roth. ASDF: assembly state detection utilizing late fusion by integrating 6D pose estimation. *CoRR*, abs/2403.16400, 2024. 3

[28] X. Song, W. Xie, J. Li, N. Wang, F. Zhong, G. Zhang, and X. Qin. Minilag filter for jitter elimination of pose trajectory

in AR environment. In *ISMAR*, pages 950–959. IEEE, 2023. 3

[29] X. Song, W. Xie, J. Li, N. Wang, F. Zhong, G. Zhang, and X. Qin. 3D object tracking for rough models. *Comput. Graph. Forum*, 42(7), 2023. 3, 7, 8

[30] M. Stoiber, M. Elsayed, A. E. Reichert, F. Steidle, D. Lee, and R. Triebel. Fusing visual appearance and geometry for multi-modality 6DoF object tracking. In *IROS*, pages 1170–1177, 2023. 2, 6, 7, 9

[31] M. Stoiber, M. Pfanne, K. H. Strobl, R. Triebel, and A. Albu-Schäffer. A sparse gaussian approach to region-based 6DoF object tracking. In *ACCV (2)*, volume 12623 of *Lecture Notes in Computer Science*, pages 666–682. Springer, 2020. 2, 7

[32] M. Stoiber, M. Pfanne, K. H. Strobl, R. Triebel, and A. Albu-Schäffer. SRT3D: A sparse region-based 3D object tracking approach for the real world. *Int. J. Comput. Vis.*, 130(4):1008–1030, 2022. 2, 7

[33] M. Stoiber, M. Sundermeyer, and R. Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3D tracking of textureless objects. In *CVPR*, pages 6845–6855. IEEE, 2022. 3, 5

[34] X. Sun, J. Zhou, W. Zhang, Z. Wang, and Q. Yu. Robust monocular pose tracking of less-distinct objects based on contour-part model. *IEEE Trans. Circuits Syst. Video Technol.*, 31(11):4409–4421, 2021. 2

[35] D. J. Tan, N. Navab, and F. Tombari. Looking beyond the simple scenarios: Combining learners and optimizers in 3D temporal tracking. *IEEE Trans. Vis. Comput. Graph.*, 23(11):2399–2409, 2017. 3

[36] X. Tian, X. Lin, F. Zhong, and X. Qin. Large-displacement 3D object tracking with hybrid non-local optimization. In *ECCV (22)*, volume 13682 of *Lecture Notes in Computer Science*, pages 627–643. Springer, 2022. 2, 7, 8

[37] H. Tjaden, U. Schwanecke, E. Schömer, and D. Cremers. A region-based gauss-newton approach to real-time monocular multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1797–1812, 2019. 2, 6, 7, 8

[38] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. C. Bovik, and Y. Li. Maxvit: Multi-axis vision transformer. In *ECCV (24)*, volume 13684 of *Lecture Notes in Computer Science*, pages 459–479. Springer, 2022. 3

[39] B. Wang, G. Wang, A. Sharf, Y. Li, F. Zhong, X. Qin, D. Cohen-Or, and B. Chen. Active assembly guidance with online video parsing. In *VR*, pages 459–466. IEEE Computer Society, 2018. 3

[40] K. Wang, K. Chen, C. Li, Z. Tu, and B. Luo. Adapting segment anything model to multi-modal salient object detection with semantic feature fusion guidance. *arXiv preprint arXiv:2408.15063*, 2024. 3

[41] L. Wang, S. Yan, J. Zhen, Y. Liu, M. Zhang, G. Zhang, and X. Zhou. Deep active contours for real-time 6-DoF object tracking. In *ICCV*, pages 13988–13998. IEEE, 2023. 7, 8

[42] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *ECCV (21)*, volume 13681 of *Lecture Notes in Computer Science*, pages 68–85. Springer, 2022. 3

[43] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018. 7

[44] J. Xiao, Y. Ma, W. Yang, and T. Zhang. Learning adaptive conceptual prototypes for 3D single object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 3

[45] Y. Xiong, B. Varadarajan, L. Wu, X. Xiang, F. Xiao, C. Zhu, X. Dai, D. Wang, F. Sun, F. N. Iandola, R. Krishnamoorthi, and V. Chandra. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. *CoRR*, abs/2312.00863, 2023. 3, 5, 8, 9

[46] X. Xu, H. Chen, L. Zhao, Z. Wang, J. Zhou, and J. Lu. EmbodiedSAM: Online segment any 3D thing in real time. In *ICLR*. OpenReview.net, 2025. 3

[47] Y. Xu, K. Lin, G. Zhang, X. Wang, and H. Li. RNNPose: Recurrent 6-DoF object pose refinement with robust correspondence field estimation and pose optimization. In *CVPR*, pages 14860–14870. IEEE, 2022. 2

[48] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S. Bae, S. Lee, and C. S. Hong. Faster segment anything: Towards lightweight SAM for mobile applications. *CoRR*, abs/2306.14289, 2023. 2, 3, 5, 8

[49] Y. Zhang, T. Cheng, R. Hu, L. Liu, H. Liu, L. Ran, X. Chen, W. Liu, and X. Wang. EVF-SAM: early vision-language fusion for text-prompted segment anything model. *CoRR*, abs/2406.20076, 2024. 3

[50] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang. Fast segment anything. *CoRR*, abs/2306.12156, 2023. 3, 8

[51] L. Zhong, M. Lu, and L. Zhang. A direct 3D object tracking method based on dynamic textured model rendering and extended dense feature fields. *IEEE Trans. Circuits Syst. Video Technol.*, 28(9):2302–2315, 2018. 2

[52] L. Zhong and L. Zhang. A robust monocular 3D object tracking method combining statistical and photometric constraints. *Int. J. Comput. Vis.*, 127(8):973–992, 2019. 2

[53] C. Zhou, X. Li, C. C. Loy, and B. Dai. EdgeSAM: Prompt-in-the-loop distillation for on-device deployment of SAM. *CoRR*, abs/2312.06660, 2023. 3, 5, 8

[54] F. Zhou, H. B. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *ISMAR*, pages 193–202. IEEE Computer Society, 2008. 2