Occlusion-aware Region-based 3D Pose Tracking of Objects with Temporally Consistent Polar-based Local Partitioning

Leisheng Zhong, Xiaolin Zhao, Yu Zhang, Shunli Zhang, and Li Zhang

***Abstract*—Region-based methods have become the state-of-art solution for monocular 6-DOF object pose tracking in recent years. However, two main challenges still remain: the robustness to heterogeneous configurations (both foreground and background), and the robustness to partial occlusions. In this paper, we propose a novel region-based monocular 3D object pose tracking method to tackle these problems. Firstly, we design a new strategy to define local regions, which is simple yet efficient in constructing discriminative local color histograms. Contrary to previous methods which define multiple circular regions around the object contour, we propose to define multiple overlapped, fan-shaped regions according to polar coordinates. This local region partitioning strategy produces much less number of local regions that need to be maintained and updated, while still being temporally consistent. Secondly, we propose to detect occluded pixels using edge distance and color cues. The proposed occlusion detection strategy is seamlessly integrated into the region-based pose optimization pipeline via a pixel-wise weight function, which significantly alleviates the interferences caused by partial occlusions. We demonstrate the effectiveness of the proposed two new strategies with a careful ablation study. Furthermore, we compare the performance of our method with the most recent state-of-art region-based methods in a recently released large dataset, in which the proposed method achieves competitive results with a higher average tracking success rate. Evaluations on two real-world datasets also show that our method is capable of handling realistic tracking scenarios.**

***Index Terms*—pose estimation, 3D object pose tracking, region-based method, occlusion detection.**

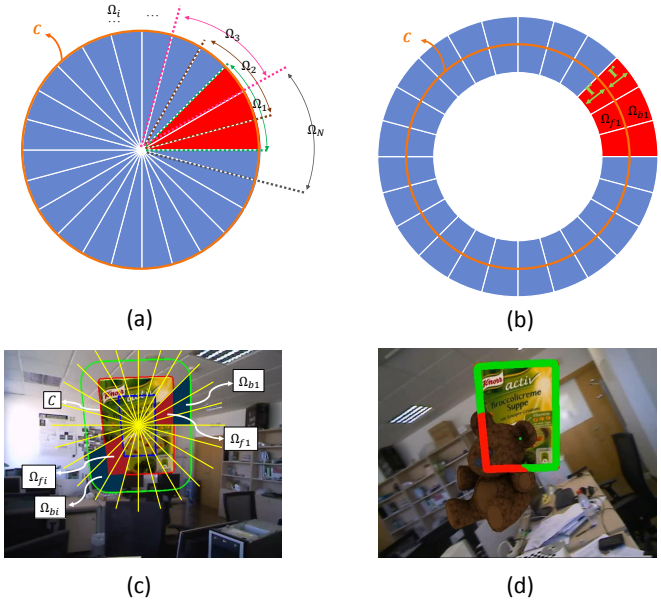

Fig. 1. (a, b) Illustration of the proposed temporally consistent local partitioning strategy with overlapped, fan-shaped local regions. The object is illustrated as a circle for exemplary demonstration. (c) A real example of the proposed local region partitioning strategy. (d) Partial occlusions could be accurately detected in our method.

I. Introduction

TRACKING the 6-DOF pose of a rigid object in monocular videos is an essential problem in computer vision [1]. It is the basic technology in various applications such augmented reality (AR), robotic perception and human-computer interaction [2]–[4]. Recent researches have demonstrated the advantages of region-based methods in real-time 3D object pose tracking among other traditional approaches [5]–[7], especially in some difficult situations. The underlying statistical formulation of region-based method makes it robust to certain kinds of pixel-level outliers, such as a moderate degree of lighting variation, background cluttering, and minor occlusions. However, two main challenges still remain for region-based methods and have limited the application in more complex configurations: (1) Dealing with heterogeneous object and background; (2) Dealing with partial occlusions. We discuss the two challenges in detail as follows.

A. Dealing with Heterogeneous Object and Background

The core of region-based 3D object pose tracking is a probabilistic formulation aiming to maximize the discrimination of statistical foreground and background appearance models [5]. Originally, the statistical information of the image is described by a single global foreground model and a single global background model [5], [8]. In that case, good segmentation results (i.e., posterior probability maps) are only guaranteed in the situation of tracking a homogeneous object in a simple background. The global model is prone to fail for

This work is supported by the National Natural Science Foundation of China under Grant No. 61871248, No. 61503405, No. 61601021, and the Natural Science Foundation of Beijing under Grant No. L172022. *(Corresponding author: Li Zhang.)*

Leisheng Zhong, Yu Zhang, and Li Zhang are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. Leisheng Zhong is also with the Naval Research Academy, Beijing 100073, China. (E-mails: {zls13, zhang-yu15}@mails.tsinghua.edu.cn, chinazhangli@mail.tsinghua.edu.cn)

Xiaolin Zhao is with the School of Aeronautics and Astronautics Engineering, Air Force Engineering University, Xi'an 710038, China. (E-mail: zhaoxiaolin00@mails.tsinghua.edu.cn)

Shunli Zhang is with the School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China. (E-mail: slzhang@bjtu.edu.cn)

heterogeneous objects or in cluttered background, in which case a single statistical model is not sufficient to describe the complex scene, leading to statistical confusion.

The key in dealing with heterogeneous object and cluttered background is to replace the global statistical models with more discriminative local statistical models. In this way, better segmentation results could be achieved even in heterogeneous environments, because locally the color distribution is usually simpler and more distinctive than globally. However, some other problems emerge when localizing the foreground and background regions. Firstly, with the increasing number of partitioned local regions, the number of pixels belonging to each local region decreases accordingly. The local color histograms calculated from these very small number of pixels tend to be unstable and fragile. Secondly, special care is needed to make sure the local regions are temporally consistent across the video frames. In other words, the partitioned local regions of each frame should be reasonably related through time, so that they could be properly identified and updated in each frame. To tackle these issues, the recent state-of-art methods [6], [7], [9] use a large number of overlapped circular image regions around the object contour as the local statistical regions. The temporal consistency is guaranteed by assigning each local region to a 3D model vertex, so that they could be memorized, identified, and correctly updated in multiple frames through time. Since each model vertex is associated with a local color histogram, a very large number of local color histograms are maintained during the whole tracking process (although a second, down-sampled 3D model is utilized in [7], the number of local histograms could still reach 5000 in maximum). This causes runtime issues since too many local histograms need to be updated in each frame. The authors have to use a Monte Carlo approach and randomly update only 100 local histograms per frame.

While the strategy described in [6], [7], [9] is proved to be very efficient, we argue that it is possible to design an even simpler and more concise local region partitioning strategy. As shown in Fig. 1, instead of assigning each local region to a model vertex, we choose to partition the global region into overlapped fan-shaped local regions. Here, each region is identified by the polar coordinates of the pixels relative to the object center. So here the temporal consistency is inherently guaranteed along with the stable spatial relationship between the regions. There is no need to connect the local regions to some other sources, such as 3D model vertices, in order to identify them in the following frames. In other words, the local regions are identified and maintained in the 2D image domain, instead of in the 3D object model domain. Moreover, we could now use much less number of local regions (e.g. 24) and obtain similar or even better segmentation results.

B. Dealing with Partial Occlusions

The second challenge for region-based 3D object pose tracking methods is to handle partial occlusions. Previously, most works choose to handle occlusions in an implicit manner. Region-based methods utilize statistical information of the image, which makes it inherently resistant to certain degrees of occlusion. When a small part of the object is occluded, the other non-occluded parts still provide enough correct information to prevent the statistical models from corruption. But when a large part of the object is occluded, the statistical models would easily get corrupted thus leading to tracking failure.

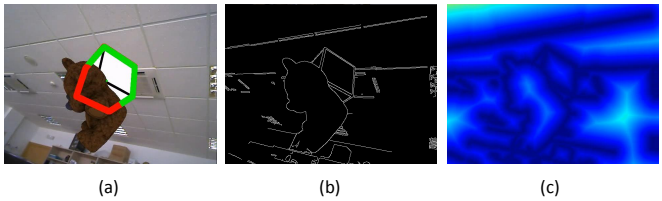

Fig. 2. Handling partial occlusions using edge distance cues. (a) A cube is occluded by a brown bear. The detected occluded pixels are illustrated in red, and the non-occluded pixels are illustrated in green. (b) The edge map. (c) The edge distance map, which is the Distance Transform [10] ($\Phi(\mathbf{x})$) of the edge map. This strategy works well for simple texture-less occluders, and we have further improved it by using edge distance and edge color cues together.

Therefore, we believe it would be a better choice to handle partial occlusions explicitly. In this paper, we exploit two kinds of edge-based information to detect the occluded pixels: the edge distance cues and the edge color cues. The idea of using edge distance cues to handle occlusion has been explored in a previous edge-based tracking method [11], which assumes the occluded object contour points tend to have a large distance to the nearest image edge (as shown in Fig. 2). However, we find that using only the edge distance cues is not always reliable since the above assumption is only correct for simple texture-less occluders, but is often violated when the occluders are well-textured and have a lot of inner edges. So we propose to further improve this strategy by using not only edge distance cues, but also edge color cues. We compare the color of the contour point in the rendered image to the color of its nearest edge point in the video frame. For a non-occluded point, the color difference of these two points would be small since they both belong to the nearby points on the target object. But for an occluded point, the color difference would be large, because the nearest edge point is very likely to locate on the occluder (which would have different color with the target object in most cases). By considering the edge color cues, we could further reduce the influence of the occluded pixels when using edge distance cues alone is not fully reliable. Experiments demonstrate that this occlusion detection strategy works very well in our region-based tracking pipeline.

C. Contribution

This paper aims to tackle the above two main challenges for region-based 3D object pose tracking. To this end, two novel strategies are proposed for dealing with heterogeneous situation and partial occlusions respectively. The main contributions of this paper are:

1. We propose a simple yet efficient local region partitioning strategy based on polar coordinates. The local regions are designed to be overlapped fan-shaped regions, which makes the new strategy both concise and flexible. Our new strategy

could ensure temporal consistency with very little efforts, and could achieve competitive or even superior results with much less number of partitioned local regions.

2. We propose an occlusion handling strategy that explicitly detects the occluded pixels using edge distance and color cues. This strategy is utilized seamlessly in the region-based pose optimization pipeline via an occlusion-aware weight function, which helps to improve the tracking performance when encountering partial occlusions.

The rest of the article is structured as follows: Section II provides a detailed review of the related works, especially focusing on the recent region-based methods and the occlusion handling strategies. We present our method in Section III, including the full derivation of the region-based cost function, together with the proposed novel local region partitioning strategy and the occlusion handling strategy. An extensive experimental evaluation is performed in Section IV. The article concludes in Section V.

II. Related work

There is a large amount of literature in 6-DOF object pose tracking researches [1]. In early years, feature-based [2], [3], [12] and edge-based [13]–[15] methods have been popular. However, feature-based methods require that the object is well-textured so that enough number of local keypoints could be extracted on the object surface. Edge-based methods, on the other hand, are prone to fail in cluttered background which would lead the tracker to be stuck in local minima. In recent years, the so called region-based methods [5], [8], [16] which rely on statistical foreground/background segmentation have proved to achieve state-of-art performance in this task [6], [7], especially in complex and difficult situations. Therefore, we mainly focus on region-based methods, which are most closely related to this work. We also introduce the occlusion handling strategies proposed in previous 3D object pose tracking methods and their shortcomings for comparison with the one we present in this work.

A. Region-based 3D Object Pose Tracking Methods

The most famous work in the family of region-based methods is PWP3D [5], which is the foundation of almost all of the recent state-of-art region-based approaches. Although it utilizes some basic concepts from earlier works [17], [18], PWP3D creatively defines an energy function based on pixel-wise posterior probabilities instead of pixel-wise likelihoods, which contributes to a much better tracking performance. Relying on GPU acceleration, this work also becomes the first region-based method that achieves real-time performance. Several successive works following the spirit of PWP3D have been proposed recently [6]–[9], [19], and each of them tries to address some of the drawbacks of the original algorithm.

The original implementation of PWP3D uses first-order gradient descent for pose optimization, which is very inefficient because the optimal step sizes (for rotation and translation parameters respectively) have to be adjusted experimentally for each model thus are very difficult to decide. It also needs a lot of iterations to ensure convergence. This issue is addressed in [8] by replacing the first-order gradient descent with a second order so-called Gauss-Newton-like optimization strategy. The authors propose to approximate the Hessian matrix using first-order derivatives, so that the optimal step size could be determined automatically. They also optimize their implementation to make it real-time capable using only CPU parallelization and OpenGL rendering. However, the Gauss-Newton-like optimization strategy in [8] is developed from empirical studies and has not been proved mathematically.

Another drawback of PWP3D is that it only uses simple global foreground and background statistical models, which leads to tracking failure in cluttered scenes because the global color models could not sufficiently capture the spatial variation in those cases. Therefore, based on the idea in [20], a local region-based method is proposed in [9], in which they define multiple local regions centered around the object contour points. For each local region, a unique local foreground model and local background model are maintained, which contributes to better segmentation results than the global region model. With the new localized model, this method shows wider basin of attraction and higher convergence scores when evaluating on a 3D object detection dataset (by trying to recover from perturbed poses for each individual image). However the performance of [9] for 3D object pose tracking in continuous video sequences is not clear. Also, no strategy is provided to ensure temporal consistency of the local statistical models through the video.

A more recent work [6] combines the idea from [8] and [9], and further extends the segmentation model by introducing the so-called temporally consistent local color histograms (tclc-histograms). They utilize the second-order Gauss-Newton-like optimization of [8] and the local statistical model of [9]. Furthermore, they propose to ensure the temporal consistency of the local statistical models by assigning each local region to a 3D model vertex, so that they could be memorized, identified, and correctly updated across multiple frames. This strategy proves to be very efficient, and the resulting algorithm currently achieves state-of-art performance. However in [6], a very large number of local color histograms have to be maintained during the whole tracking process, which needs special care and potentially increases the complexity of the algorithm. In this paper, we also propose a temporally consistent local model. But instead of assigning each local region to a model vertex, we choose to partition the global region into overlapped fan-shaped local regions, and identify each region via the polar coordinates of the pixels. Therefore, the temporal consistency is inherently guaranteed in a much simpler manner, and we only need to maintain a much smaller number of local regions.

The same author summarizes their previous works [6], [8] in [7], and expands them by providing a systematic derivation of the Gauss-Newton optimization by means of reformulation the problem as a iteratively reweighted non-linear least-squares problem. They also release a large semi-synthetic 6-DOF object pose tracking dataset. We evaluate our method in this new dataset and compare our results with the most recent state-of-arts.

The authors in [19] propose a hybrid tracker by combining the statistical constraints and the photometric constraints. They

also propose to partition the global region into fan-shaped local regions, but the local regions are not overlapped. So the number of local regions has to be very small, which limits the performance of the tracker in more complex configurations. The design of fan-shaped regions has also been used in other topics in image processing research. In [21], the authors present a novel local binary descriptor named Ring-based Multi-Grouped Descriptor (RMGD), in which a ring-region sampling scheme is introduced to generate pooling region candidates with multiple scales and shapes. In their method, an image patch is first divided into a number of ring regions centered at central of the patch; then each generated ring region is further divided into a number of fan-shaped sub divisions. Only 4, 8 and 16-divisions are used in [21]. Bergamasco *et al.* [22] introduce a fiducial marker characterized by a circular arrangement of dots at fixed angular position in one or more concentric rings. The fiducial marker is built by partitioning a disc in several evenly distributed sectors (which are fan-shaped regions), and then the sectors are further divided into a number of concentric rings.

Some other works also make some improvements in different aspects. In [23], the authors add a boundary term in the original PWP3D energy function. Extensions on mobile phones and RGB-D sensors are first introduced in [24] and [25]. A fast RGB-D algorithm combining the statistical term and the ICP term is proposed in [16].

B. Occlusion Handling in 3D Object Pose Tracking

For comparison with the occlusion handling strategy in this work, we give a brief introduction to the occlusion handling strategies proposed in previous 3D object pose tracking methods.

An occlusion-aware online update rule is proposed in [23]. They suggest to update the appearance model if and only if most of the pixels inside the contour pertain to the foreground (i.e., not occluded). However, they simply rely on the color posterior maps to decide whether a pixel is occluded, which is easily influenced by the occluder itself. Also, they only focus on the update of the color models, but not the more important optimization step.

In [26], the occluded area is explicitly detected by comparison of the current frame and the rendered (non-occluded) template image. Although this occlusion detection strategy proves to be effective, it requires a realistic textured 3D model for rendering, which is often not available.

The authors in [8] propose to handle the mutual occlusion of two known objects by tracking both of them, assuming the occluder is also a known object (whose 3D model is available). But in real occasions, the tracked object could be occluded by various unknown occluders, such as human hands or other deformable objects.

A new occlusion handling strategy is introduced in [11] in the context of edge-based 3D object pose tracking. Based on the assumption that the occluded 3D object contour points tend to be far away from image edges, they assign a weight to each contour point according to the edge distance map. A successive work [27] proposes to perform direction-based pose validation by comparing the edge direction of the projected contour point to its nearest image edge point. The pose validation scheme is further incorporated for non-local searching and failure recovery. In this paper, we borrow this idea and extend it to region-based 3D object pose tracking by considering not only the edge distance cues, but also the edge color cues. We combine the region-based statistical formulation with the edge-based distance and color cues for occlusion handling, which produces a robust occlusion-aware 3D object pose tracker.

Some other 3D object pose tracking methods also try to combine edge-based and region-based methods together. Panin *et al.* [28] propose to integrate color and edge likelihoods for efficient data fusion in a 3D object pose tracking pipeline. Petit *et al.* [29] develop an edge-based 3D pose tracker combining geometrical and color edge information. Seo *et al.* [30] propose a 3D object pose tracking method based on optimal local searching assisted by color statistics. In this paper, we also combine region-based methods with edge cues, but our motivation is to deal with partial occlusions, which is different from those methods.

C. Other Related Methods and Our Assumptions

1) Learning-based 3D Object Pose Tracking: Some previous works have explored the possibility of using learning-based methods for 3D object pose tracking. Krull *et al.* [31] propose to train a random forest that regresses the 3D object coordinates from the RGB-D image. Tan *et al.* [32] also utilize random forest in a RGB-D object tracking framework. More recently, deep learning-based methods have also been proposed for 3D object pose tracking. Garon *et al.* [33] present the first deep learning-based 6-DOF temporal object tracker with RGB-D input. A successive work [34] improves the network architecture in [33] and obtains better tracking performance. Manhardt *et al.* [35] present a novel 6-DOF pose refinement framework using CNN. A new visual loss is designed to drive the pose update by aligning object contours. Li *et al.* [36] proposes a deep neural network which is able to iteratively refine the pose by matching the rendered image against the observed image. Recent advances of learning-based methods have shown great potential in more robust and accurate 3D pose tracking. However, these methods generally need a time-consuming training stage with the help of powerful GPUs. Moreover, the capability of the CNN-based methods for tracking unseen objects is also limited.

2) Monocular 3D Object Detection: Another closely related topic is 3D object detection, in which the 6-DOF pose is determined from a single image, instead of a continuous image sequence. In real tracking applications, a 3D object detection method is required for initialization and re-initialization from tracking loss. Previously, monocular 3D object detection is typically achieved using 2D template matching [37]–[40]. In these methods, both the input image and the templates are transformed into the so-called gradient response maps for fast comparison of the dominant gradient orientations. The recent state-of-art region-based 3D object pose tracking methods [6], [7] also borrow this idea, and propose to match the so-called tclc-histograms for (re-)initialization. The tclc-histograms need

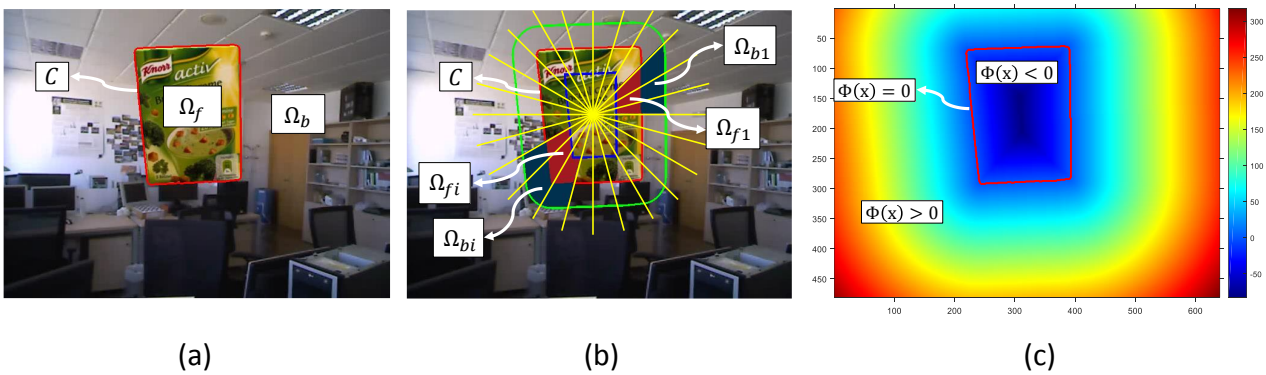

Fig. 3. Overview of the classical model and the proposed model. (a) The classical global region model. (b) The proposed local region model based on temporally consistent polar-based local partitioning. (c) The corresponding signed distance transform $\Phi(\mathbf{x})$.

to be trained in the beginning of a new tracking task by showing the object to the camera from different perspectives. However, as also pointed out by the authors [7], the tclc-histogram-based descriptors will not work in previously unseen environments. Apart from template matching, a lot of deep learning-based 3D object detection methods have been proposed recently [41]–[46]. These methods train deep neural networks either to directly predict the pose parameters [41]–[43], or to first predict the keypoint locations and then estimate the 6-DOF pose via the PnP algorithm [44]–[46]. The deep learning-based methods are currently achieving state-of-art results, with the only drawback that they require powerful GPUs thus are not light-weight enough for some specific applications.

3) Our Assumptions: In this paper, we mainly focus on the frame-to-frame 3D pose tracking problem. We assume an initial object pose (the pose of the first frame) is given, and try to estimate the 6-DOF pose of the object in the subsequent video frames. For evaluation on datasets with ground-truth poses, the initial pose is given by the ground-truth pose of the first frame, as with in most previous works [6], [7], [19], [23], [26], [47], [48]. In real applications, the initial pose could be manually assigned by roughly align the object to a rendered object mask (which corresponds to a known pose). Also, the 3D object detection methods described above can be combined with our method for automatic initialization and reset. Besides, we mainly consider the case of single object pose tracking in this paper, but it is straight forward to extend the method to multiple object pose tracking as described in [5]. For multiple object tracking, we need to initialize multiple trackers for all the objects with their initial poses, and then track each individual object separately. We also assume that the camera is pre-calibrated, so that the intrinsic matrix K is known. For evaluation on datasets, the intrinsic matrix K is usually provided in the dataset. In real applications, we need to first calibrate the camera using some calibration tools, such as the ones provided in Matlab [49] or OpenCV [50], which are typically based on Zhang's calibration method [51].

III. Proposed Method

In this section, we first briefly introduce the mathematical notations and the classical region-based statistical formulation. After that, we will present the proposed temporally consistent polar-based local region partitioning strategy, and the proposed edge-based occlusion handling strategy in detail. In the end, the details of the non-linear pose optimization step are presented based on our new energy function.

A. Region-based 3D Object Pose Tracking

We begin by introducing the classical global region-based model. The global region-based model is depicted in Fig. 3(a). The RGB image is denoted by I. The image region Ω is partitioned into a foreground region Ω_f and a background region $\Omega_b = \Omega \backslash \Omega_f$. Every pixel $\mathbf{x} = (x, y)^T \in \Omega$ has a corresponding color vector $\mathbf{y} = I(\mathbf{x})$. Every foreground pixel $\mathbf{x} \in \Omega_f$ has a corresponding 3D model point $\mathbf{X} = (X, Y, Z)^T$ in the camera coordinate frame and $\mathbf{X}_0 = (X_0, Y_0, Z_0)^T$ in the object coordinate frame. The 3D rigid transformation between these two coordinate frames is defined by a rotation matrix $R \in \mathbb{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, which can be encoded in a 6-DOF pose vector $\mathbf{p} = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)^T \in \mathbb{R}^6$ using Lie algebra representation. Here $\mathbb{SO}(3)$ represents the *special orthogonal group*, which is the collection of 3D rotation matrices [52]:

$$\mathbb{SO}(3) = \left\{ R \in \mathbb{R}^{3\times 3} \,\middle|\, RR^T = I, \det(R) = 1 \right\} \tag{1}$$

By combining rotation and translation, we have:

$$\tilde{\mathbf{X}} = T\tilde{\mathbf{X}}_0 = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{\mathbf{X}}_0 \tag{2}$$

where $T \in \mathbb{SE}(3)$ represents the rigid body transform, and the *tilde-notation* indicates the homogeneous representation $\tilde{\mathbf{X}} = (X, Y, Z, 1)^T$. Here $\mathbb{SE}(3)$ represents the *special Euclidean group*, which is the collection of 3D rigid transformation matrices [52]:

$$\mathbb{SE}(3) = \left\{ T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \,|\, R \in \mathbb{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\} \tag{3}$$

The projection of 3D points into the 2D image plane is described by:

$$\mathbf{x} = \pi\left(K\mathbf{X}\right) \tag{4}$$

where $K \in \mathbb{R}^{3\times3}$ is the intrinsic matrix of the camera, and $\pi\left(\mathbf{X}\right) = \left(X/Z, Y/Z\right)^T$. The goal of 6-DOF pose tracking is to determine the pose parameters $\mathbf{p}$ of the object in each video frame.

In the context of region-based methods, the object in the image is represented by a level-set embedding function (or the signed distance function) $\Phi\left(\mathbf{x}\right)$ [5], as shown in Fig. 3(c). The signed distance function $\Phi\left(\mathbf{x}\right)$ calculates the (signed) Euclidean distance between a pixel $\mathbf{x}$ to its closest contour point. The projected object contour $\mathbf{C}$ is then defined as the zero level-set $\mathbf{C} = \{\mathbf{x}|\Phi\left(\mathbf{x}\right) = 0\}$. We have $\Phi\left(\mathbf{x}\right) < 0$ for the foreground region, and $\Phi\left(\mathbf{x}\right) > 0$ for the background region. The foreground and background statistics are usually represented by a global foreground appearance model $P\left(\mathbf{y}|M_f\right)$ and a global background appearance model $P\left(\mathbf{y}|M_b\right)$, where $\mathbf{y} = I\left(\mathbf{x}\right)$ is the RGB value of a certain foreground or background pixel. The conditional distributions $P\left(\mathbf{y}|M_f\right)$ and $P\left(\mathbf{y}|M_b\right)$ are the likelihoods of a pixel with color $\mathbf{y}$ belonging to foreground or background regions, and are commonly represented with RGB color histograms. Based on this, the posterior distributions of foreground and background pixels could be calculated as [5]:

$$P_f\left(\mathbf{x}\right) = P\left(M_f|\mathbf{y}\right) = \frac{P\left(\mathbf{y}|M_f\right)}{\eta_f P\left(\mathbf{y}|M_f\right) + \eta_b P\left(\mathbf{y}|M_b\right)} \tag{5}$$

$$P_b\left(\mathbf{x}\right) = P\left(M_b|\mathbf{y}\right) = \frac{P\left(\mathbf{y}|M_b\right)}{\eta_f P\left(\mathbf{y}|M_f\right) + \eta_b P\left(\mathbf{y}|M_b\right)} \tag{6}$$

where

$$\eta_f = \sum_{\mathbf{x}\in\Omega} H_e\left(\Phi\left(\mathbf{x}\right)\right), \quad \eta_b = \sum_{\mathbf{x}\in\Omega}\left(1 - H_e\left(\Phi\left(\mathbf{x}\right)\right)\right) \tag{7}$$

and H_e is the smoothed Heaviside step function.

Finally the energy function is formulated as [5]:

$$\begin{aligned} E(\mathbf{p}) = -\sum_{\mathbf{x}\in\Omega} \log[&H_e(\Phi(\mathbf{x}(\mathbf{p})))P_f(\mathbf{x}) \\ &+ (1 - H_e(\Phi(\mathbf{x}(\mathbf{p}))))P_b(\mathbf{x})] \end{aligned} \tag{8}$$

B. Temporally Consistent Polar-based Local Partitioning

As discussed in Section I, the global model is prone to fail for heterogeneous objects and cluttered background because a single global model is no longer descriptive enough in these cases. Hence, we propose a novel local region partitioning strategy based on polar coordinates. As shown in Fig. 1 and Fig. 3(b), the global region is partitioned into overlapped fan-shaped local regions according to the polar coordinates of each pixel relative to the object center. More specifically, the global region is first divided into N_p local *parts*, and the *i*-th part is defined as:

$$\Psi_i = \left\{\mathbf{x}\,\middle|\,(i-1)\times\frac{2\pi}{N_p} \le \theta\left(\mathbf{x},\mathbf{x}_c\right) < i\times\frac{2\pi}{N_p}\right\} \tag{9}$$

where $\mathbf{x}_c = (x_c, y_c)$ is the object center, and $i = 1, 2, ..., N_p$. $\theta\left(\mathbf{x},\mathbf{x}_c\right)$ is the angular element of the polar coordinate of $\mathbf{x}$ relative to $\mathbf{x}_c$:

$$\theta\left(\mathbf{x},\mathbf{x}_c\right) = \mathrm{atan2}\left(x - x_c, y - y_c\right) \tag{10}$$

Here the object center (in the image) $\mathbf{x_c}$ is calculated as the projection of the 3D object center $\mathbf{X_c}$ onto the image plane:

$$\mathbf{x}_c = \pi\left(K\left(R\mathbf{X}_c + \mathbf{t}\right)\right) \tag{11}$$

where R and $\mathbf{t}$ are the pose of the current frame. The object center $\mathbf{x_c}$ is re-calculated for each new frame as the origin of the polar coordinate system.

Then, each local region Ω_i is defined as the combination of N_s continuous parts:

$$\Omega_i = \bigcup_{j=0:N_s-1} \Psi_{\ \mathrm{mod}\ (i+j-1,N_p)+1} \tag{12}$$

where mod() is the modulo operation, $i = 1, 2, ..., N_p$.

As a result, the global region is partitioned into N_p overlapped local regions, with each region containing N_s *subparts* ($N_s = 1$ means no overlapping). Two different examples of our polar-based partitioning results are demonstrated in Fig. 4 with $N_p = 8, N_s = 2$ and $N_p = 24, N_s = 3$.

After the local partitioning, an individual color model could be calculated in each local region, and the local posteriors are now computed as:

$$P_{f_i}\left(\mathbf{x}\right) = P\left(M_{f_i}|\mathbf{y}\right) = \frac{P\left(\mathbf{y}|M_{f_i}\right)}{\eta_{f_i} P\left(\mathbf{y}|M_{f_i}\right) + \eta_{b_i} P\left(\mathbf{y}|M_{b_i}\right)} \tag{13}$$

$$P_{b_i}\left(\mathbf{x}\right) = P\left(M_{b_i}|\mathbf{y}\right) = \frac{P\left(\mathbf{y}|M_{b_i}\right)}{\eta_{f_i} P\left(\mathbf{y}|M_{f_i}\right) + \eta_{b_i} P\left(\mathbf{y}|M_{b_i}\right)} \tag{14}$$

where $P\left(\mathbf{y}|M_{f_i}\right)$ and $P\left(\mathbf{y}|M_{b_i}\right)$ are the local color histograms of the *i*-th region, and

$$\eta_{f_i} = \sum_{\mathbf{x}\in\Omega_i} H_e\left(\Phi\left(\mathbf{x}\right)\right), \quad \eta_{b_i} = \sum_{\mathbf{x}\in\Omega_i}\left(1 - H_e\left(\Phi\left(\mathbf{x}\right)\right)\right) \tag{15}$$

Note that as shown in Fig.1 and Fig. 3(b), the local color histograms are calculated in the limited band around the object contour with the bandwidth r for better distinctiveness, as with the other local region-based methods [6], [9].

The next step is to fuse all the local statistical models and formulate the overall energy function. Similar to [6], [7], we choose to compute the average posteriors from all local histograms that the pixels belong to, instead of computing the average energy over all local regions. The average posterior maps are calculated as:

$$\bar{P}_f\left(\mathbf{x}\right) = \frac{1}{\sum_{i=1}^{N_p} \mathrm{B}_i\left(\mathbf{x}\right)} \sum_{i=1}^{N_p} P_{f_i}\left(\mathbf{x}\right) \mathrm{B}_i\left(\mathbf{x}\right) \tag{16}$$

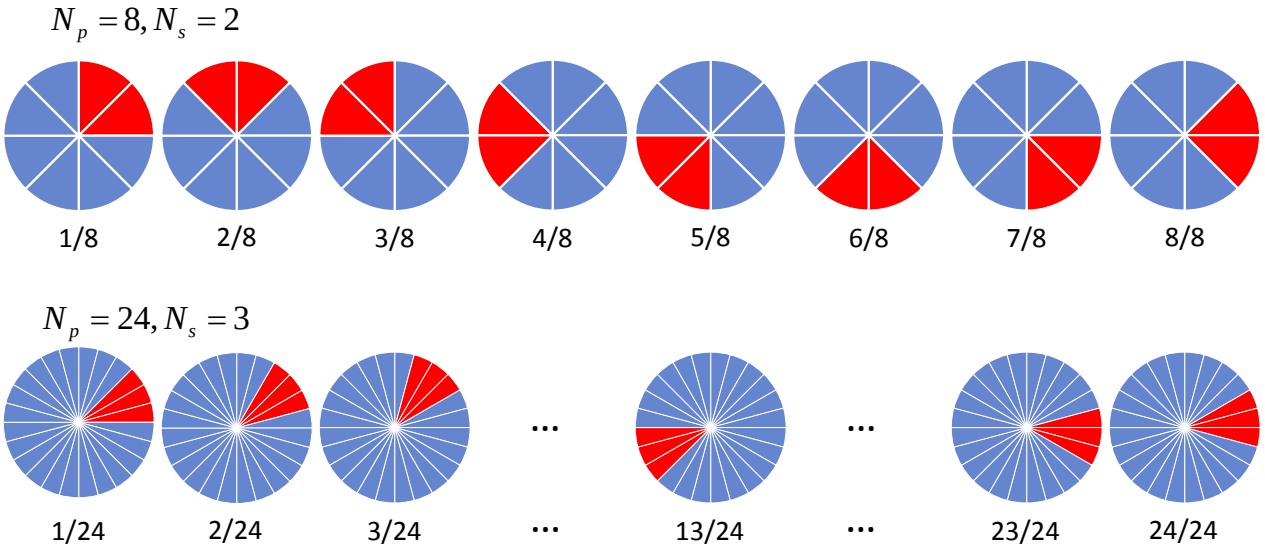

Fig. 4. Two examples of the proposed overlapped fan-shaped local region partitioning.

$$\bar{P}_b(\mathbf{x}) = \frac{1}{\sum_{i=1}^{N_p} \mathrm{B}_i(\mathbf{x})} \sum_{i=1}^{N_p} P_{b_i}(\mathbf{x})\, \mathrm{B}_i(\mathbf{x}) \tag{17}$$

where $\mathrm{B}_i(\mathbf{x})$ is the masking function indicating whether a pixel $\mathbf{x}$ belongs to the *i*-th local region:

$$\mathrm{B}_i(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_i \\ 0 & \forall \mathbf{x} \notin \Omega_i \end{cases} \tag{18}$$

In our case, each pixel belongs to N_s different regions (i.e., $\sum_{i=1}^{N_p} \mathrm{B}_i(\mathbf{x}) = N_s$), and the specific corresponding local regions can be easily determined according to Eq. (12).

Taking the average posteriors into Eq. (8), we obtain the energy function for our localized model:

$$\begin{aligned} E(\mathbf{p}) = -\sum_{\mathbf{x}\in\Omega} \log[H_e(\Phi(\mathbf{x}(\mathbf{p})))\bar{P}_f(\mathbf{x}) \\ + (1 - H_e(\Phi(\mathbf{x}(\mathbf{p}))))\bar{P}_b(\mathbf{x})] \end{aligned} \tag{19}$$

As with in [6], [7], we call the local histograms *temporally consistent* because each local histogram (and the corresponding local image region) is able to be identified and correctly updated across multiple frames. In [6], [7], the local histograms could be identified because each of them is assigned to a 3D model vertex, and the vertex is projected onto the image in each frame as the center of that histogram region. In our case, the local histogram regions are identified simply by the polar coordinates using Eq. (9). Here the only issue is to build the polar coordinate system in each frame by projecting the object model center onto the image plane by Eq. (11). The projected center is then used as the origin of the polar coordinate system, thus each local region is determined on the image according to the polar coordinates of the pixels, ensuring temporal consistency across multiple frames.

The proposed polar-based local region partitioning strategy has several advantages. Firstly, it is very simple and flexible. Stable local parts are partitioned using simple polar coordinates in 2D image domain without extra efforts. Moreover, by changing N_p and N_s, we could easily explore the best settings for the current task according to the scene complexity. Secondly, it ensures temporal consistency in a much simpler manner. Instead of assigning each local region to a model vertex in 3D, we choose to identify the local regions by the polar coordinates of the pixels in 2D. Here the temporal consistency is inherently guaranteed along with the stable spatial relationship between the regions across multiple frames. Thirdly, the overlapped design makes it possible to use more partitions without the risk of unstable color models caused by the deficiency of pixel numbers. The overlapped design also allows to calculate the average posterior membership probability of each pixel, which could contribute to more reliable segmentation results [6], [7]. Finally, a much smaller number of local regions are used in our method while similar or even better tracking performance is achieved, as will be demonstrated in the experiment part.

C. Handling Occlusions Using Edge Distance and Color Cues

Occlusion handling is one of the most difficult problems in 3D object pose tracking. Although some strategies have been proposed in previous works, they are likely to fail in the case of heavy occlusions. Here we present an efficient new occlusion handling strategy based on edge distance and color cues. We add it seamlessly into the region-based tracking pipeline via a simple weight function.

The idea of using edge distance cues to handle occlusion is originally introduced in edge-based tracking [11], in which they assume the occluded object contour points tend to have a large distance to the nearest image edge. However, this assumption is only valid when the occluder is poor-textured and has few inner edges, so that the edge distance cues will not be affected. For more complex and well-textured

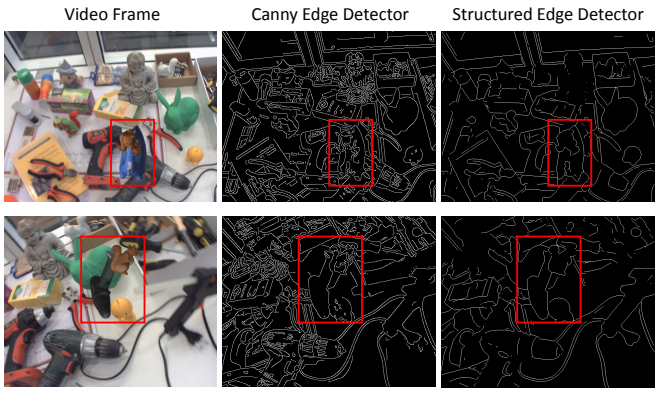

Fig. 5. Comparison between the edge detection results of the Canny Edge Detector [53] and the Structured Edge Detector [54]. The Structured Edge Detector produces better edge maps with less noise, thus is preferred when the occluder has rich texture.

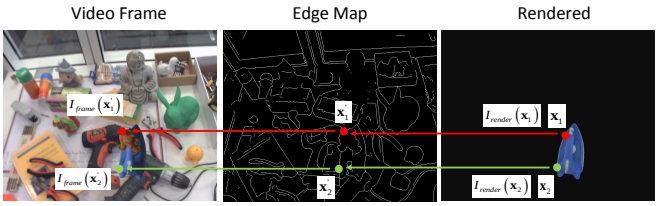

Fig. 6. By comparing the color information between the contour point in the rendered image and its nearest edge point in the video frame, we can further distinguish occluded point (red) from non-occluded point (green).

occluders, only relying on edge distance cues could sometimes be misleading. Therefore, we propose to detect the occluded contour points by using edge distance cues and edge color cues together. When the edge distance is not fully reliable, the edge color could be utilized to further distinguish occluded points from non-occluded points. In most cases, the occluder has different color with the target object, and we could detect the occluded point by comparing the color between the contour point in the rendered image and its nearest edge point in the video frame. Combining the edge distance cues and the edge color cues could contribute to a more robust occlusion handling strategy.

Formally, for each pixel in the evaluation band, we assign an occlusion-aware weight to it:

$$w\left(\mathbf{x}\right) = w_d\left(\mathbf{x}\right) w_c\left(\mathbf{x}\right) \tag{20}$$

where $w_d\left(\mathbf{x}\right)$ is the weight calculated from edge distance cues, and $w_c\left(\mathbf{x}\right)$ is the weight calculated from edge color cues.

Firstly, for the edge distance cues, the weight $w_d\left(\mathbf{x}\right)$ is calculated as:

$$w_d\left(\mathbf{x}\right) = \begin{cases} 1 - \left(\frac{D_e(\mathbf{x})}{c_1}\right)^2 & if\ D_e\left(\mathbf{x}\right) \leq c_1 \\ 0 & otherwise \end{cases} \tag{21}$$

where $D_e\left(\mathbf{x}\right)$ is the distance of $\mathbf{x}$ to its nearest edge point $\mathbf{x}^{'}$ in the video frame, which could be extracted from the distance transform (Fig. 2(c)) of the edge map (Fig. 2(b)). c_1 is the maximum valid distance for a pixel to the nearest image edge.

In order to obtain better edge maps with less noise, we choose to use the random forest-based Structured Edge Detector [54] instead of simple gradient-based edge detectors (such as the Canny Edge Detector [53]). Some examples of the edge maps extracted by the Canny Edge Detector and the Structured Edge Detector are compared in Fig. 5.

Secondly, the weight for the edge color cues is calculated as:

$$w_c\left(\mathbf{x}\right) = \begin{cases} 1 - \left(\frac{D_c(\mathbf{x})}{c_2}\right)^2 & if\ D_c\left(\mathbf{x}\right) \leq c_2 \\ 0 & otherwise \end{cases} \tag{22}$$

where $D_c\left(\mathbf{x}\right)$ is the difference between the color of $\mathbf{x}$ in the rendered image and the color of its nearest edge point $\mathbf{x}^{'}$ in the video frame: $D_c\left(\mathbf{x}\right) = \left\| I_{render}\left(\mathbf{x}\right) - I_{frame}\left(\mathbf{x}^{'}\right) \right\|$. c_2 is the maximum valid color difference.

An example for calculating $w_c\left(\mathbf{x}\right)$ from the edge color cues is demonstrated in Fig. 6. For an occluded point $\mathbf{x}_1$, its nearest edge point $\mathbf{x}_1^{'}$ is likely to locate on the occluder, so the color difference $D_c\left(\mathbf{x}_1\right)$ is large and the weight $w_c\left(\mathbf{x}_1\right)$ is small. On the contrary, for a non-occluded point $\mathbf{x}_2$, its nearest edge point $\mathbf{x}_2^{'}$ is likely to locate near the same object point corresponding to $\mathbf{x}_2$, so the color difference $D_c\left(\mathbf{x}_2\right)$ is small and the weight $w_c\left(\mathbf{x}_2\right)$ is large. By incorporating the edge color cues, the influence of the occluded points could be further reduced even when the edge distance cues are not fully reliable.

After calculating the weights for the object contour points, we spread the weights through the normal direction of the object contour, so that every pixel in the evaluation band (typically ± 8 pixels around the object contour as in previous works) gets its occlusion-aware weight according to its nearest object contour point. Combining the proposed occlusion-aware weight function with the region-based method leads to a slightly different energy function:

$$\begin{aligned} E(\mathbf{p}) = -\sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log[H_e(\Phi(\mathbf{x}(\mathbf{p})))\bar{P}_f(\mathbf{x}) \\ + (1 - H_e(\Phi(\mathbf{x}(\mathbf{p}))))\bar{P}_b(\mathbf{x})] \end{aligned} \tag{23}$$

This is the final energy function we use for 6-DOF pose optimization. With this occlusion-aware weight function, a non-occluded pixel in the evaluation band would be assigned with a large weight, and an occluded pixel in the evaluation band would be assigned with a small weight. Therefore, the influence of the occluded pixels could be minimized, which improves the robustness of our method to partial occlusions.

D. Pose Optimization

To solve the complex non-linear optimization problem, we use a similar Gauss-Newton-based pose optimization strategy as in [7] by rewriting the energy function Eq. (23) as a non-linear iteratively re-weighted least squares problem:

$$E\left(\mathbf{p}\right) = \frac{1}{2}\sum_{\mathbf{x}\in\Omega} \psi\left(\mathbf{x}\right) F^2\left(\mathbf{x}, \mathbf{p}\right) \tag{24}$$

where

$$F(\mathbf{x},\mathbf{p}) = -w(\mathbf{x})\log[H_e(\Phi(\mathbf{x}(\mathbf{p})))\bar{P}_f(\mathbf{x}) + (1-H_e(\Phi(\mathbf{x}(\mathbf{p}))))\bar{P}_b(\mathbf{x})] \quad (25)$$

and $\psi(\mathbf{x}) = \frac{1}{F(\mathbf{x},\mathbf{p})}$.

Then the non-linear optimization problem could be iteratively solved by fixing and alternatingly updating the weights $\psi(\mathbf{x})$. The Jacobian is calculated as:

$$\begin{aligned}\mathbf{J}(\mathbf{x}) &= \frac{\partial F(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} \\ &= -w(\mathbf{x})\frac{\bar{P}_f - \bar{P}_b}{H_e(\Phi(\mathbf{x}))\bar{P}_f + (1-H_e(\Phi(\mathbf{x})))\bar{P}_b} \\ &\quad \times \frac{\partial H_e(\Phi(\mathbf{x}))}{\partial \mathbf{p}}\end{aligned} \quad (26)$$

and

$$\frac{\partial H_e(\Phi(\mathbf{x}))}{\partial \mathbf{p}} = \frac{\partial H_e}{\partial \Phi}\frac{\partial \Phi}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} \quad (27)$$

where $\frac{\partial H_e}{\partial \Phi} = \delta_e(\Phi)$ is the smoothed Dirac delta function, $\frac{\partial \Phi}{\partial \mathbf{x}} = \left[\frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y}\right]$ is calculated using centered finite differences. $\frac{\partial \mathbf{x}}{\partial \mathbf{p}}$ can be derived from eqs. (2, 4), and the details can be found in [7], [9].

The Hessian is then approximated using first-order derivatives [7]:

$$\mathbf{H}(\mathbf{x}) = \psi(\mathbf{x})\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) \quad (28)$$

which leads to the optimal Gauss-Newton update step:

$$\mathbf{\Delta p} = -\left(\sum_{\mathbf{x}\in\Omega}\mathbf{H}(\mathbf{x})\right)^{-1}\sum_{\mathbf{x}\in\Omega}\mathbf{J}(\mathbf{x})^T \quad (29)$$

IV. Experiments

In the following we provide an extensive evaluation of the proposed method. We first clarify some details in our implementation. Then we do a careful ablation study on Rigid Pose Dataset [47] to validate the effectiveness of the proposed new strategies. This is followed by a comprehensive evaluation on the very recent RBOT Dataset [7], in which we compare our results with the most recent state-of-arts. Finally, we test our method on two real-world datasets, the Dense Tracking Dataset [55] and the OPT Dataset [48], to show the performance of our method on real data. For all of the experiments, a commodity desktop computer with Intel i7 quad core CPU @4.0GHz and NVIDIA GeForce GTX970 GPU is used. Our method runs at 20-25Hz using CPU (with GPU used only for rendering).

A. Implementation Details

We generally use similar parameter settings as the recent region-based methods [6], [7], [19]. After successfully tracking the *k*-th frame, the local color histograms are updated as:

$$P(\mathbf{y}|M_{f_i}) = (1-\alpha_f)P^{k-1}(\mathbf{y}|M_{f_i}) + \alpha_f P^k(\mathbf{y}|M_{f_i}) \quad (30)$$

$$P(\mathbf{y}|M_{b_i}) = (1-\alpha_b)P^{k-1}(\mathbf{y}|M_{b_i}) + \alpha_b P^k(\mathbf{y}|M_{b_i}) \quad (31)$$

and the learning rates are set to $\alpha_f = 0.1$ and $\alpha_b = 0.2$ as in [7].

The region radius r in Fig. 1(b) is set to $r = 40$ based on the results of [9] (we observe similar performance using $r = 32 \sim 64$).

The smoothed Heaviside function in Eq. (23) is defined as:

$$H_e(\Phi(\mathbf{x})) = \frac{1}{\pi}\left(-\text{atan}(s\Phi(\mathbf{x})) + \frac{\pi}{2}\right) \quad (32)$$

and the smoothed Dirac delta function $\delta_e(\Phi(\mathbf{x}))$ is derived from $H_e(\Phi(\mathbf{x}))$:

$$\delta_e(\Phi(\mathbf{x})) = \frac{\partial H_e(\Phi(\mathbf{x}))}{\partial \Phi} = -\frac{s}{\pi(1+s^2\Phi^2(\mathbf{x}))} \quad (33)$$

where we use $s = 1.2$ as in [7].

The threshold c_1 in Eq. (21) is set to 20 as in [11], and c_2 in Eq. (22) is set to 80.

We have also tried different settings of the local region partitioning parameter N_p and N_s, and found that $N_p = 24, N_s = 3$ produces the best results in the evaluated datasets. More details are presented in Section IV-B.

B. Ablation Study on Rigid Pose Dataset

We choose to perform the ablation study on Rigid Pose Dataset [47]. This dataset provides semi-synthetic video sequences of 6 different objects under a variety of realistic conditions. The dataset composes of original sequences, noisy sequences and occluded sequences (18 video sequences in total). These sequences are featured with wide-range rotation and translation, object variability and background cluttering. Specifically, for the noisy sequences, random Gaussian noise ($\sigma = 0.1$) is added separately to each color channel in order to test the robustness of the tracker to image noise. For the occluded sequences, another randomly moving object is added to the video frames, which creates realistic occlusion to the first object.

We use the same evaluation metric as in [47]. The error is measured by the largest distance between corresponding vertices transformed by the estimated and the ground-truth poses [47]: $e(\mathbf{p}) = \max_j \|(R\mathbf{v}_j + \mathbf{t}) - (R_{gt}\mathbf{v}_j + \mathbf{t}_{gt})\|$, where $\mathbf{v}_j$ is a vertex of the 3D object model. A frame is successfully tracked if $e(\mathbf{p})$ is smaller than a threshold (we use 10mm as in [47]). Otherwise, we consider the tracker to be lost and reset it with the ground-truth pose. We measure the tracking success rate (SR) throughout the entire sequence, which is defined as the proportion of frames that are successfully tracked (in %): $SR = N_{success}/N_{frames} \times 100\%$, where $N_{success}$ is the number of frames that are successfully tracked and N_{frames} is the total number of frames in the sequence.

Firstly, we demonstrate the influence of the region partitioning parameters (N_p, N_s) by evaluating on the original sequences of Rigid Pose Dataset. We test 10 different settings of (N_p, N_s), and the results are shown in Table I. We list the

TABLE I
EVALUATION RESULTS USING DIFFERENT N_p AND N_s ON THE ORIGINAL SEQUENCES OF RIGID POSE DATASET. (TRACKING SUCCESS RATE IN %, BEST SCORES ARE IN BOLD.)

N_p	1	4	8	16	24	32	48	48	64	128
N_s	1	1	2	2	3	3	1	6	6	16
soda	58.5	59.7	58.7	60.1	**61.9**	60.4	54.6	61.8	60.8	61.6
soup	88.9	88.6	89.1	88.6	**89.8**	87.9	83.6	89.1	88.1	89.6
clown	94.5	95.6	95.1	95.7	**96.6**	95.9	87.4	96.4	95.9	**96.6**
candy	71.3	73.0	72.7	72.7	**74.1**	73.0	68.1	73.7	73.2	73.9
cube	87.4	89.1	89.1	89.4	**90.6**	89.2	81.4	90.4	89.4	90.4
edge	92.2	91.5	91.0	92.0	**93.2**	92.0	87.7	92.8	92.3	93.0
average	82.1	82.9	82.6	83.1	**84.4**	83.1	77.1	84.0	83.3	84.2

TABLE II
ABLATION STUDY ON RIGID POSE DATASET. LOC: METHODS WITH THE PROPOSED LOCAL REGION PARTITIONING STRATEGY. OCC: METHODS WITH THE PROPOSED OCCLUSION DETECTION STRATEGY. (TRACKING SUCCESS RATE IN %, BEST SCORES ARE IN BOLD.)

Method		soda			soup			clown			candy			cube			edge			average
Loc	Occ	ori	noi	occ	ori	noi	occ	ori	noi	occ	ori	noi	occ	ori	noi	occ	ori	noi	occ	
		57.3	53.8	41.0	88.6	85.3	65.5	95.2	92.7	72.0	71.7	61.8	53.9	87.5	83.8	61.6	92.2	89.6	67.6	73.4
√		**61.9**	**57.2**	41.1	**89.8**	87.7	67.2	96.2	**94.0**	71.0	73.2	**64.0**	56.0	**90.6**	**84.0**	65.2	**93.2**	90.4	66.6	75.0
	√	58.5	54.8	**48.5**	88.9	85.2	**75.9**	94.5	91.8	**80.4**	71.3	62.3	64.9	87.4	82.9	**72.9**	92.2	89.9	**78.7**	76.7
√	√	**61.9**	56.5	**48.5**	**89.8**	**90.1**	75.1	**96.6**	93.2	79.0	**74.1**	63.1	**65.0**	**90.6**	**84.0**	**72.9**	**93.2**	**90.8**	77.3	**77.9**

SR scores of each sequence for different settings. The tracking performance generally improves with local partitioning, and the setting of $(N_p = 24, N_s = 3)$ produces the best tracking performance consistently in all the 6 sequences. Similar SR scores are achieved with larger number of partitions (such as $(N_p = 128, N_s = 16)$) at the expense of more computation, thus is not preferred. Moreover, the results indicate that the overlapped design is crucial in the proposed local region partitioning strategy. The tracking performance greatly deteriorates when the global region is divided into a large number of partitions but the local regions are not overlapped (such as $(N_p = 48, N_s = 1)$). The reason is that, in this case, each local region does not contain enough pixels for maintaining a stable statistical model, as discussed in previous sections. As a result, we choose to use $(N_p = 24, N_s = 3)$ in all the following experiments.

Secondly, in order to validate the effectiveness of the two new strategies proposed in this paper, we have tested 4 different versions of the method on the whole Rigid Pose Dataset. The tested methods are 4 variants of our method w/ or w/o the local partitioning and occlusion handling strategies respectively. The tracking success rates are presented in Table II. The results show that the proposed local region partitioning and occlusion handling strategies are effective, and generally improve the robustness of our tracker. Specifically speaking, the local region partitioning strategy contributes to higher tracking success rates especially for the original and noisy sequences, and the occlusion handling strategy helps to improve the tracking performance on occluded sequences by a large margin. Finally, the method with both the proposed local region partitioning and occlusion detection strategies obtains the best average SR score, confirming the effectiveness of the two new strategies.

C. Evaluation on RBOT Dataset

Next, we present a comprehensive evaluation on the very recent RBOT Dataset [7] and compare our results with the recent state-of-art approaches. The RBOT (Region-based Object Tracking) Dataset is a large semi-synthetic dataset containing 72 video sequences with a total number of 72000 video frames. The dataset composes of 4 different variants of videos: regular, dynamic light, noisy, and occlusion. The first variant (regular) is rendered using a static point light source. The second variant (dynamic light) is rendered with a dynamic light source in order to evaluate the robustness of the tracker to lighting variations. The third variant (noisy) further adds artificial Gaussian noise to the images. And the fourth variant (occlusion) contains an additional second object which frequently occludes the first object. A total of 18 objects are used in RBOT Dataset, including both heterogeneous and texture-less objects, as shown in Fig. 7. We use the same evaluation metric as in [7]. The rotation and translation errors are calculated separately as:

Fig. 7. The 18 objects models used in RBOT Dataset. Figures are adapted from [7].

$$e(R) = \cos^{-1}\left(\frac{\text{trace}\left(R^T R_{gt}\right) - 1}{2}\right) \tag{34}$$

TABLE III
EVALUATION RESULTS ON RBOT DATASET. REG: REGULAR; DYN: DYNAMIC LIGHT; NOI: NOISY; OCC: OCCLUSION. (TRACKING SUCCESS RATE IN %, BEST SCORES ARE IN BOLD.)

Variant	Method	Ape	Baking Soda	Bench Vise	Broccoli Soup	Camera	Can	Cat	Clown	Cube	Driller	Duck	Egg Box	Glue	Iron	Koala Candy	Lamp	Phone	Squirrel	Average
REG	ICCV17 [6]	62.1	30.5	95.8	66.2	61.6	81.7	96.7	89.1	44.1	87.7	74.9	50.9	20.2	68.4	20.0	92.3	64.9	98.5	67.0
	TPAMI19 [7]	85.0	39.0	**98.9**	82.4	79.7	87.6	95.9	93.3	78.1	**93.0**	**86.8**	74.6	38.9	81.0	46.8	**97.5**	80.7	**99.4**	79.9
	IJCV19 [19]	82.6	40.1	92.6	85.0	82.8	87.2	98.0	92.9	81.3	84.5	83.3	76.2	56.1	84.6	57.6	90.5	82.6	95.6	80.8
	Ours	**88.8**	**41.3**	94.0	**85.9**	**86.9**	**89.0**	**98.5**	**93.7**	**83.1**	87.3	86.2	**78.5**	**58.6**	**86.3**	**57.9**	91.7	**85.0**	96.2	**82.7**
DYN	ICCV17 [6]	61.7	32.0	94.2	66.3	68.0	84.1	96.6	85.8	45.7	88.7	74.1	56.9	29.9	49.1	20.7	91.5	63.0	98.5	67.0
	TPAMI19 [7]	84.9	**42.0**	**99.0**	81.3	84.3	88.9	95.6	92.5	77.5	**94.6**	**86.4**	**77.3**	52.9	77.9	47.9	**96.9**	**81.7**	**99.3**	81.2
	IJCV19 [19]	81.8	39.7	91.5	85.1	82.6	87.1	98.1	90.7	79.7	87.4	81.6	73.1	51.7	75.9	53.4	88.8	78.6	95.6	79.0
	Ours	**89.7**	40.2	92.7	**86.5**	**86.6**	**89.2**	**98.3**	**93.9**	**81.8**	88.4	83.9	76.8	**55.3**	**79.3**	**54.7**	88.7	81.0	95.8	**81.3**
NOI	ICCV17 [6]	55.9	35.3	75.4	67.4	27.8	10.2	94.3	33.4	8.6	50.9	76.3	2.3	2.2	18.2	11.4	36.6	31.3	93.5	40.6
	TPAMI19 [7]	77.5	**44.5**	**91.5**	82.9	51.7	38.4	95.1	**69.2**	24.4	64.3	**88.5**	11.2	2.9	46.7	32.7	57.3	44.1	**96.6**	56.6
	IJCV19 [19]	**80.5**	35.0	80.9	85.5	58.4	53.5	96.7	65.9	**38.2**	71.8	85.8	29.7	17.0	59.3	34.8	61.1	60.8	93.6	61.6
	Ours	79.3	35.2	82.6	**86.2**	**65.1**	**56.9**	**96.9**	67.0	37.5	**75.2**	85.4	**35.2**	**18.9**	**63.7**	**35.4**	**64.6**	**66.3**	93.2	**63.6**
OCC	ICCV17 [6]	55.2	29.9	82.4	56.9	55.7	72.2	87.9	75.7	39.6	78.7	68.1	47.1	26.2	35.6	16.6	78.6	50.3	77.6	57.5
	TPAMI19 [7]	80.0	**42.7**	91.8	73.5	76.1	81.7	89.8	82.6	68.7	86.7	80.5	67.0	46.6	64.0	43.6	**88.8**	68.6	86.2	73.3
	IJCV19 [19]	77.7	37.3	87.1	78.7	74.6	81.0	93.8	84.3	73.2	83.7	77.0	66.4	48.6	70.8	49.6	85.0	73.8	90.6	74.1
	Ours	**83.9**	38.1	**92.4**	**81.5**	**81.3**	**85.5**	**97.5**	**88.9**	**76.1**	**87.5**	**81.7**	**72.7**	**52.2**	**77.2**	**53.9**	88.5	**79.3**	**92.5**	**78.4**

$$e(\mathbf{t}) = \|\mathbf{t} - \mathbf{t}_{gt}\|_2 \tag{35}$$

The object is considered to be successfully tracked if $e(\mathbf{t})$ is below 5cm and $e(R)$ below 5°. Otherwise the tracker is considered to be lost and is automatically reset to the ground truth. Then we also measure the tracking success rate (SR) of each sequence as the proportion of frames that are successfully tracked (in %).

We compare the results of our method with the three most recent state-of-art region-based methods: ICCV17 [6], TPAMI19 [7] and IJCV19 [19]. Among them, ICCV17 [6] and TPAMI19 [7] are the only two methods having been evaluated on this dataset before, and we test IJCV19 [19] on this dataset using the code provided by the author.

The evaluation results are summarized in Table III. Our method obtains the best SR scores on 50 out of the 72 videos. In particular, for the occluded sequences, our method obtains the best SR scores on 16 out of the 18 sequences, which proves the superiority of the proposed edge-based occlusion handling strategy. Moreover, we also achieve the best average SR scores on all of the 4 video variants. The experimental results have further demonstrated the robustness of our method against the state-of-art region-based methods, thanks to the proposed novel local region partitioning strategy and occlusion handling strategy. More specifically, our method obtains similar or even better tracking performance using much less number of local regions, and the occlusion handling strategy helps to significantly improve the robustness in partial occlusion scenarios. Some sample frames from the 4 video variants of RBOT Dataset and the corresponding tracking results of our method are demonstrated in Fig. 8.

D. Evaluation on Real-world Datasets

We have demonstrated the performance of our method on semi-synthetic datasets in Section IV-B and Section IV-C. In order to show the capability of our method for real data, we further test our method on two real-word datasets: the Dense Tracking Dataset [55] and the OPT Dataset [48].

TABLE IV
EVALUATION RESULTS ON DENSE TRACKING DATASET. (TRACKING SUCCESS RATE (IN %), BEST SCORES ARE IN BOLD.)

Method	Exp#1	Exp#2	ATLAS#1	ATLAS#2
CVPR14 [55]	98.4	97.5	**100**	39.4
TCSVT18 [26]	**100**	**99.7**	**100**	85.1
IJCV19 [19]	99.7	98.4	90.1	78.8
Ours	**100**	98.6	**100**	**87.3**

1) Dense Tracking Dataset: The Dense Tracking Dataset [55] provides real-world video sequences in two challenging environments, including a complex industrial environment, as shown in Fig. 9. The videos contain strong moving light sources, bright specularities (both foreground and background), and motion blurs.

We use the same evaluation metric as in [55]. A rotation error and a translation error are also computed separately the same as Eqs. (34), (35). But here the thresholds are set to 4° (0.07 *radians*) and 5cm as in [55]. The tracking success rate (SR) is measured as described in Section IV-B and Section IV-C.

The evaluation results are shown in Table IV. We compare the performance of our method with the other three state-of-art methods (CVPR14 [55], TCSVT18 [26] and IJCV19 [19]) since they have reported their results on this dataset. The results show that our method achieves competitive tracking performance with these methods on this real-world dataset. More specifically, the proposed method consistently outperforms CVPR14 [55] and IJCV19 [19], especially for the most difficult sequence (ATLAS #2). TCSVT18 [26] also performs very well on this dataset, and obtains similar results with us. The reason is that they incorporate a specifically designed illumination estimation module in their method, which is very beneficial for this dataset since fast illumination change is one of the main challenges.

2) OPT Dataset: In the end, we evaluate our method on another real-world dataset, OPT Dataset [48]. The OPT Dataset is a large 6-DOF object pose tracking dataset, which

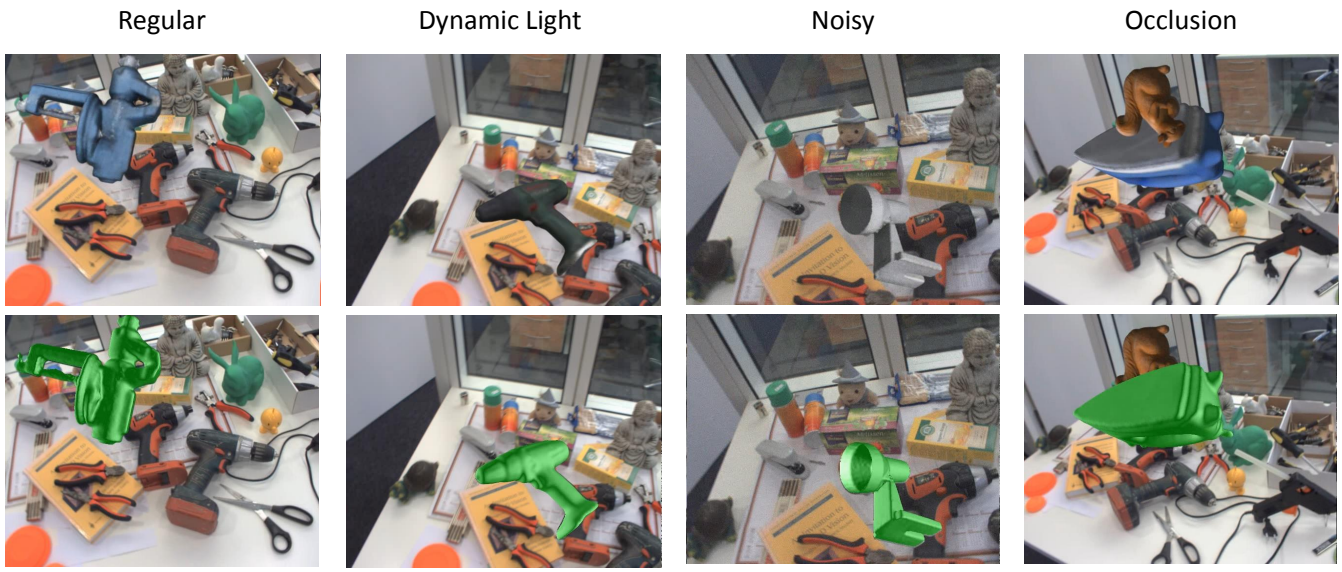

Fig. 8. Some sample frames from the 4 video variants of RBOT Dataset and the corresponding tracking results. For each video variant, the first row shows the input images, and the second row shows the tracking results of our method. The tracking results are illustrated by rendering the 3D object models with the tracked 6-DOF pose in green. (Zoom in for better view.)

TABLE V
EVALUATION RESULTS ON OPT DATASET (AUC SCORES, HIGHER IS BETTER, BEST SCORES ARE IN BOLD, SECOND-BEST SCORES ARE UNDERLINED.)

Method	bike	chest	house	ironman	jet	soda	average
PWP3D [5]	5.358	5.551	3.575	3.915	5.813	5.870	5.014
UDP [56]	6.097	6.791	5.974	5.250	2.342	8.494	5.825
ElasticFusion [57]	1.567	1.534	2.695	1.692	1.858	1.895	1.874
ORB-SLAM2 [58]	10.410	**15.531**	**17.283**	11.198	9.931	**13.444**	**12.966**
TPAMI19 [7]	11.903	11.764	10.150	**11.986**	13.217	8.861	11.314
Ours	**12.831**	12.240	13.613	11.214	**15.441**	9.012	12.392

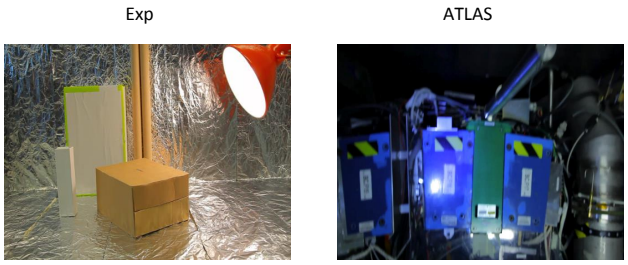

Fig. 9. Two challenging environments in Dense Tracking Dataset [55].

contains 552 real-world sequences of six 3D objects (bike, chest, house, ironman, jet and soda). These sequences are classified into seven different types (translation, zoom in and zoom out, in-plane rotation, out-of-plane rotation, flashing light, moving light and free motion) with different moving speed.

To compare the results of our method to [7] and those provided in [48], we use the same evaluation metric as in [7], [48]. The tracking error is measured by the average distance between corresponding vertices transformed by the estimated and the ground-truth poses: $e(\mathbf{p}) = \underset{j}{\text{avg}} \|(R\mathbf{v}_j + \mathbf{t}) - (R_{gt}\mathbf{v}_j + \mathbf{t}_{gt})\|$, where $\mathbf{v}_j$ is a vertex of the 3D object model. A frame is successfully tracked if $e(\mathbf{p})$ is smaller than $k_e d$ where d is the diameter of the object, and k_e is a tunable coefficient. Then the tracking success rates (SR) are measured by varying $k_e \in [0, 0.2]$. Finally, the overall tracking performance is computed in form of an AUC (area under curve) score, where the SR scores (0∼100) are integrated for all $k_e \in [0, 0.2]$ (so the AUC score is between 0∼20, higher is better). More details of the evaluation protocol on OPT Dataset could be found in [48].

In Table V, we compare our results with 5 other methods available for this dataset: PWP3D [5], the classical region-based 3D object pose tracking method; UDP [56], a monocular 3D object detection method; ElasticFusion [57], a visual SLAM approach which relies on RGB-D data; ORB-SLAM2 [58], another visual SLAM approach which uses gradient-based features; TPAMI19 [7], one of the most recent state-of-art region-based 3D object pose tracking methods. The results show that the proposed method performs significantly better than PWP3D, UDP and ElasticFusion, and obtains comparable results with ORB-SLAM2 and TPAMI19. Among the evaluated 6 objects, our method obtains 2 best AUC scores and 4 second-best AUC scores. The average score of our method ranks second, and is very close to that of ORB-SLAM2. Here ORB-SLAM2 performs especially well for chest, house and soda because these objects are relatively well-textured which is beneficial for feature-based methods. Besides, the symmetrical structure of soda also makes it difficult for region-

based methods such as TPAMI19 and our method.

E. Discussions

1) Limitations of Our Method: Although we have demonstrated the capability of our method on both semi-synthetic and real-world datasets, it still has some limitations.

Firstly, the proposed occlusion detection strategy is able to handle certain degrees of partial occlusion, but would still fail for heavy occlusions. When major parts of the object are occluded, it is very difficult to precisely estimate the 6-DOF pose with only monocular input. Nevertheless, we plan to study this problem by utilizing deep neural networks to interpret the information behind the occluder in our future works.

Secondly, although we have incorporated both the edge distance cues and the edge color cues to further improve the robustness of the proposed occlusion handling strategy, it still has some limitations. For example, when the occluder has a lot of inner edges, and also has similar color with the target object, the proposed strategy would fail to detect the occluded points. However, this case does not frequently happen, and the proposed strategy works well in most common cases. Experiments have also shown that our method performs significantly better than the state-of-arts on occluded sequences (as discussed in Section IV-C), which proves the effectiveness of the proposed occlusion detection strategy.

Thirdly, an efficient 3D object detection module is yet to be developed for pose initialization and reset. In this paper, we mainly focus on robust frame-to-frame pose tracking. The 3D object detection methods (such as the ones discussed in Section II-C-2) could be combined with our method to build a more complete solution. Specifically, for light-weight applications, template-based 3D detection methods (such as LINE-2D [38] and tclc-histograms [6]) could be utilized. When GPU is available, deep learning-based methods (such as [44], [46]) could be incorporated for better performance.

2) Runtime Performance: We compare the runtime of our method with some of the other state-of-art methods evaluated in our experiments in Table VI. The numbers are taken from the corresponding papers or from our evaluation. The average runtime of our method is 41.2 ms per frame, which is a little higher than [6] and [7]. The main reason is the inclusion of the occlusion detection step in our method, which takes about 15 ms per frame. We believe this part could be further optimized in our future work. Moreover, the proposed method is faster than ORB-SLAM2 [58] although ORB-SLAM2 performs relatively better on OPT Dataset.

TABLE VI
RUNTIME PERFORMANCE. (PER FRAME IN MS)

Method	Runtime
CVPR14 [47]	91.8
ICCV17 [6]	12.5∼33.3
TCSVT18 [26]	129.6
TPAMI19 [7]	15.5∼21.8
IJCV19 [19]	47.0
ORB-SLAM2 [58]	67.0
Ours	41.2

V. CONCLUSION

In this paper, we have proposed two novel strategies trying to tackle the two main challenges of region-based 6-DOF object tracking: (1) A novel temporally consistent polar-based local region partitioning strategy for more robust tracking in heterogeneous situations; (2) An efficient edge-based occlusion detection strategy for handling partial occlusions. By incorporating these two novel strategies into the state-of-art region-based scheme, we have presented a robust occlusion-aware local region-based 6-DOF object tracker. A comprehensive evaluation shows that our method achieves competitive or better tracking performance compared to the recent state-of-arts, especially in dealing with partial occlusions.

REFERENCES

[1] V. Lepetit and P. Fua, *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005.
[2] J. P. Lima, F. Simões, L. Figueiredo, and J. Kelner, "Model based markerless 3d tracking applied to augmented reality," *Journal on 3D Interactive Systems*, vol. 1, 2010.
[3] Y. Park, V. Lepetit, and W. Woo, "Multiple 3d object tracking for augmented reality," in *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2008, pp. 117–120.
[4] C. Choi and H. I. Christensen, "Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4048–4055.
[5] V. A. Prisacariu and I. D. Reid, "Pwp3d: Real-time segmentation and tracking of 3d objects," *International Journal of Computer Vision*, vol. 98, no. 3, pp. 335–354, 2012.
[6] H. Tjaden, U. Schwanecke, and E. Schömer, "Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 124–132.
[7] H. Tjaden, U. Schwanecke, E. Schömer, and D. Cremers, "A region-based gauss-newton approach to real-time monocular multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1797–1812, 2019.
[8] H. Tjaden, U. Schwanecke, and E. Schömer, "Real-time monocular segmentation and pose tracking of multiple objects," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 423–438.
[9] J. Hexner and R. R. Hagege, "2d-3d pose estimation of heterogeneous objects using a region based approach," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 95–112, 2016.
[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, vol. 8, no. 1, pp. 415–428, 2012.
[11] B. Wang, F. Zhong, and X. Qin, "Pose optimization in edge distance field for textureless 3d object tracking," in *Computer Graphics International Conference*. ACM, 2017, p. 32.
[12] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.
[13] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.
[14] C. Harris and C. Stennett, "Rapid-a video rate object tracker." in *British Machine Vision Conference (BMVC)*, 1990, pp. 1–6.
[15] B.-K. Seo, J. Park, H. Park, and J.-I. Park, "Real-time visual tracking of less textured three-dimensional objects on mobile platforms," *Optical Engineering*, vol. 51, no. 12, pp. 127 202.1—-127 202.9, 2013.
[16] W. Kehl, F. Tombari, S. Ilic, and N. Navab, "Real-time 3d model tracking in color and depth on a single cpu core," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 745–753.
[17] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 831–844.
[18] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, "Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior," in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 169–182.

[19] L. Zhong and L. Zhang, "A robust monocular 3d object tracking method combining statistical and photometric constraints," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 973–992, 2019.

[20] S. Lankton and A. Tannenbaum, "Localizing region-based active contours," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2029–2039, 2008.

[21] Y. Gao, W. Huang, and Y. Qiao, "Local multi-grouped binary descriptor with ring-based pooling configuration and optimization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4820–4833, 2015.

[22] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 113–120.

[23] S. Zhao, L. Wang, W. Sui, H.-y. Wu, and C. Pan, "3d object tracking via boundary constrained region-based model," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 486–490.

[24] V. A. Prisacariu, O. Kahler, D. W. Murray, and I. D. Reid, "Simultaneous 3d tracking and reconstruction on a mobile phone," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 89–98.

[25] C. Y. Ren, V. Prisacariu, O. Kaehler, I. Reid, and D. Murray, "3d tracking of multiple objects with identical appearance using rgb-d input," in *International Conference on 3D Vision (3DV)*, vol. 1. IEEE, 2014, pp. 47–54.

[26] 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 Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2302–2315, 2018.

[27] B. Wang, F. Zhong, and X. Qin, "Robust edge-based 3d object tracking with direction-based pose validation," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 12 307–12 331, 2019.

[28] G. Panin, E. Roth, and A. Knoll, "Robust contour-based object tracking integrating color and edge likelihoods," in *VMV*, 2008, pp. 227–234.

[29] A. Petit, E. Marchand, and K. Kanani, "A robust model-based tracker combining geometrical and color edge information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 3719–3724.

[30] B.-K. Seo, H. Park, J.-I. Park, S. Hinterstoisser, and S. Ilic, "Optimal local searching for fast and robust textureless 3d object tracking in highly cluttered backgrounds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 99–110, 2014.

[31] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother, "6-dof model based tracking via object coordinate regression," in *Asian Conference on Computer Vision (ACCV)*. Springer, 2014, pp. 384–399.

[32] D. J. Tan, N. Navab, and F. Tombari, "Looking beyond the simple scenarios: Combining learners and optimizers in 3d temporal tracking," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2399–2409, 2017.

[33] M. Garon and J.-F. Lalonde, "Deep 6-dof tracking," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2410–2418, 2017.

[34] M. Garon, D. Laurendeau, and J.-F. Lalonde, "A framework for evaluating 6-dof object trackers," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 582–597.

[35] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6d pose refinement in rgb," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 800–815.

[36] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[37] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 858–865.

[38] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2011.

[39] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision (ACCV)*. Springer, 2012, pp. 548–562.

[40] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3d object detection: A real time scalable approach," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2048–2055.

[41] 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 (RSS)*, 2018.

[42] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 1521–1529.

[43] T.-T. Do, M. Cai, T. Pham, and I. Reid, "Deep-6dpose: Recovering 6d object pose from a single rgb image," *arXiv preprint arXiv:1802.10367*, 2018.

[44] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 292–301.

[45] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[46] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4561–4570.

[47] K. Pauwels, L. Rubio, J. Diaz, and E. Ros, "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2347–2354.

[48] P.-C. Wu, Y.-Y. Lee, H.-Y. Tseng, H.-I. Ho, M.-H. Yang, and S.-Y. Chien, "A benchmark dataset for 6dof object pose tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2017, pp. 186–191.

[49] Matlab, *Matlab Computer Vision System Toolbox*. The MathWorks Inc., Natick, Massachusetts, 2016.

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

[51] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[52] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[53] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. Elsevier, 1987, pp. 184–203.

[54] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1841–1848.

[55] A. Crivellaro and V. Lepetit, "Robust 3d tracking with descriptor fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3414–3421.

[56] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold *et al.*, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3364–3372.

[57] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[58] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.