DOI: 10.1111/cgf.14154
Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L. Zhang
(Guest Editors)

Volume 39 (2020), Number 7

An Occlusion-aware Edge-Based Method for Monocular 3D Object Tracking using Edge Confidence

Hong Huang[1] Fan Zhong[2] Yuqing Sun[1] and Xueying Qin[1]

[1]School of Software, Shandong University, China
huanghone@sina.com, {sun_yuqing,qxy}@sdu.edu.cn
[2]School of Computer Science and Technology, Shandong University, China
zhongfan@sdu.edu.cn

Abstract

We propose an edge-based method for 6DOF pose tracking of rigid objects using a monocular RGB camera. One of the critical problem for edge-based methods is to search the object contour points in the image corresponding to the known 3D model points. However, previous methods often produce false object contour points in case of cluttered backgrounds and partial occlusions. In this paper, we propose a novel edge-based 3D objects tracking method to tackle this problem. To search the object contour points, foreground and background clutter points are first filtered out using edge color cue, then object contour points are searched by maximizing their edge confidence which combines edge color and distance cues. Furthermore, the edge confidence is integrated into the edge-based energy function to reduce the influence of false contour points caused by cluttered backgrounds and partial occlusions. We also extend our method to multi-object tracking which can handle mutual occlusions. We compare our method with the recent state-of-art methods on challenging public datasets. Experiments demonstrate that our method improves robustness and accuracy against cluttered backgrounds and partial occlusions.

CCS Concepts

• ***Computing methodologies*** → ***Mixed / augmented reality; Tracking;***

1. Introduction

Tracking the pose of a rigid object in complex scene is a challenging task which has been widely applied in vision and graphics applications, such as augmented reality, visual servoing, etc. Given an RGB image sequence, the aim of monocular 3D object tracking is to robustly and accurately estimate the six degree-of-freedom (6DOF) pose of a known rigid object with respect to the camera.

In last decades, many different 3D object tracking methods have been proposed and used in practical commercial and industrial application. For well-textured objects, feature-based methods [SL04, VLF04b, HBN07, PLW08] have achieved robust tracking performance because the 3D-2D correspondences can be established by feature points. However, they are not suited to track weakly-textured objects that lack distinctive image feature. For weakly-textured objects, contour edges are the vital visual cues that can be used in most situations. Therefore, edge-based methods have been shown to be suitable alternative. Given the 3D model of the target object, edge-based methods have to establish the 3D-2D correspondences between the 3D model points and object contour points. These correspondences are used as constraints to solve the pose parameters by minimizing the residual distance between projected model contours and image contours. Since the

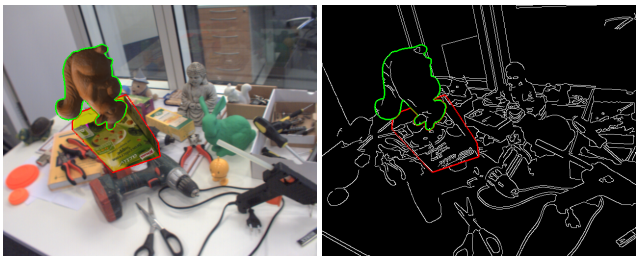

Figure 1: *Critical problem of edge-based method when tracking heterogeneous objects in the presence of partial occlusions and cluttered backgrounds. Left: The projected contour (marked red and green) obtained by projecting the 3D object model into the image plane. Right: Edge map of the scene with cluttered backgrounds and partial occlusions that may cause false object contour edges.*

Fan Zhong and Xueying Qin are corresponding authors.

first edge-based method RAPID [HS90] was proposed, several improved methods have been proposed for better robustness and accuracy [DC02, WVS05, SPP*13, WZQ19]. Though edge-based methods are fast and plausible, clutter edges (both in foreground and background) and partial occlusions often cause incorrect object contour points are searched to establish false 3D-2D correspondences, as shown in Fig. 1, which often lead to tracking drift and failure. In practice, fusing multiple visual cues or using additional sensors could improve robustness, but all the necessary information is not always available in many cases and the computation cost is too high to achieve real-time performance, especially for low-power devices (e.g. smart phone and AR glasses). Therefore, given a monocular RGB camera view, fast and robust 3D object tracking in the presence of clutter edges and partial occlusions is of great importance.

1.1. Motivation

Although current state-of-art monocular 3D object tracking methods are region-based [HSS17, HSSC19], we argue that the edge-based method still can be effective and efficient provided with correct contour correspondences. However, in complex environments the edge-based methods are often ruined due to incorrect correspondences caused by cluttered backgrounds and partial occlusions, etc. To address this problem, we propose a novel edge-based method with improved contour correspondences and pose optimization. We will show that with these improvements, the edge-based method also can achieve state-of-the-art accuracy, and can perform even better than current region-based methods.

To search the correct object contours, region color information is used in [SPP*13, WWZ*15], and leads to more robust tracking performance than previous edge-based methods. However, the region color information used in these works is modelled by a single global foreground and background color histogram, which is insufficient to describe the complex scene, and the searching strategy and pose optimization scheme often fail when the object is partially occluded. Inspired by recent region-based methods [HSS17, HSSC19], we use local color histograms to model the region color information, which can be used to suppress the cluttered points and search the object contour points.

To deal with unavoidable false contour edges caused by cluttered scenes and partial occlusions in pose optimization, robust energy function is proposed in [WZQ17, WZQ19], in which edge distance information is used to penalize the potential false object contour points. However, this strategy is only correct for simple weakly-textured occluding object, and is prone to fail for well-textured occluding object with a lot of inner edges. Instead of considering only edge distance error, our method uses the confidence of the contour points that combines region color cue and edge distance cue to discount the effect of false object contour points, and achieve bettser robustness against partial occlusions and cluttered backgrounds.

1.2. Contributions

As mentioned above, in this paper we propose an improved edge-based 3D tracking method in order to suppress the influence of cluttered backgrounds and partial occlusions. In summary, the contributions of our work are as follows:

- We propose a method to search the contour points with high confidence to be correct correspondences. False contour points are first filtered out based on local color distributions, then each remaining point is assigned a confidence value fusing edge color and distance cues, which would suppress the effect of cluttered and occluded points. The final contour points are selected by maximizing the confidence value.
- For robust pose optimization, we propose to incorporate the confidence values as an adaptive weighting function, and solve the pose parameters in IRLS iterations, which significantly alleviates the influences of unavoidable false object contour points caused by cluttered scenes and partial occlusions.
- Our work shows that the edge-based method also can perform as robust as the region-based method. The proposed method greatly outperforms previous edge-based methods on the challenging RBOT dataset, and also has an average accuracy that is 6-10% higher than the latest region-based method, which is a significant improvements considering its simplicity and efficiency.

2. Related Work

In last decades, lots of different methods have been proposed for monocular 3D object tracking [VF05, MUS15]. In the following, we focus the discussion on monocular 3D object tracking methods, which are closely related to our work. We also introduce some learning-based 3D object methods and 3D object detection methods, and clarify the difference between them. Monocular 3D object tracking aims to estimate the 6DOF object pose using consecutive RGB video frames with the initial pose of the first frame is known. Early keypoint-based methods [SL04, VLF04b, HBN07, PLW08] tackle this problem by matching local keypoint on the object surface, so these methods require the object is well-textured and cannot handle weakly-textured object very well.

Edge-based methods use only object contour points to estimate the object pose, thus can effectively track weakly-textured object. The primary interest of edge-based tracking is to search the object contour points to establish 3D-2D correspondences. Following the RAPID tracker [HS90], edge-based methods usually search for strong gradient responses on 1D search lines perpendicular to the projected contour to find object contours points. [DC02] proposes a very simple strategy that searches the nearest edge points by the intensity discontinuity above a certain threshold. In [MBC01] the authors adopt a pre-computed convolution kernel to extract the object contour points with similar orientation to the projected contour. To avoid trapping in local minima, [VLF04a, WVS05] propose to find multiple-edge hypotheses rather than a single edge hypothesis along the search line, then multiple edge points are used for one projected contour point in pose optimization, so as to improve the robustness in cluttered scene. As high-dimensional statistics, multiple-pose hypotheses methods [KM06, BC11] based on particle filtering are effective for avoiding undesirable error caused by false contour edges. Since the poses are predicted by probabilistic distributions without pose optimization using 3D-2D correspondences, the overall tracking performance is less sensitive to false contour edges. However, the computation cost is usually too

high for real-time tracking, because these methods need to evaluate large state spaces. [SPP*13] exploits region color information of foreground and background to evaluate image edge points, and searches for object contour points in only confident searching directions. Since this searching scheme ignores the affinity of adjacent object contour points, it often fails in highly cluttered backgrounds. So [WWZ*15] proposes to establish the relationship between adjacent object contour points using a graph model, and searches the optimal object contour points with dynamic programming. Since these two works use a global HSV color histogram to model the region color appearance, it is difficult to handle heterogeneous object. Apart from using search line, another way to perform edge-based tracking is to optimize pose with edge distance map, which is distance transform of edges. D^2CO [IP15] proposes to minimize the distance between the projected contour and the query image edges with edge distance map, and considers edge direction as cost measure. This requires to compute a 3D edge distance map by discretizing edge direction into 60 layers, and thus increases computation cost. For dealing with cluttered backgrounds and partial occlusions, [WZQ17, WZQ19] use particle filtering to predict a good initial pose in edge distance map, and assign small weights to occluded or false object contour points in pose optimization. Though these methods do not explicitly search contour edges, they actually search the nearest edge points from the projected contour as the object contour points, because distance transform compute the nearest distance to the edges. This simple searching scheme is unreliable when object and background have a lot of clutter edges. In this paper, we also adjust the weights of object contour points, but instead of considering only edge distance error, we combine edge color and distance cues for better robustness against partial occlusions and cluttered backgrounds.

Recently, region-based methods relying on statistical level-set segmentation are achieving state-of-art performance in complex and difficult conditions. PWP3D [PR12] is the foundation of the recent state-of-art region-based methods. This work uses a single global color histogram to compute the pixel-wise posterior probability, and formulates a unique energy function based on pixel-wise posterior probability for simultaneous 2D segmentation and 3D pose tracking. Relying on GPGPU computing techniques, this work becomes the first region-based method that achieves real-time performance. [HSS16] replaces the first-order gradient descent algorithm used in PWP3D with a second-order Gauss-Newton algorithm, which contributes to better accuracy in pose optimization. Since the global color histogram in PWP3D is not sufficient to describe complex scene, [JH16] proposes to build multiple local color histograms from local circle regions along the projected contour, which can well capture the spatial variation of cluttered background and object appearance. Recent work [HSS17] further extends the local color histograms by introducing temporally consistent local color histograms, in which the posterior probabilities of overlapped local regions are averaged. This contributes to much better segmentation result and robust tracking performance. In this paper, we also use the temporally consistent local color histograms, but they are used for searching object contour points, not for region segmentation. Successive work [HSSC19] summarizes the advances of [HSS16] and [HSS17], and reformulates the Gaussian-Newton optimization problem as an iteratively reweighted non-linear least-squares problem, which leads to drastically faster convergence and highly accurate tracking performance. [ZZ19] proposes a hybrid tracker that combines the statistical constrains from region-based methods and the photometric constrains based on raw pixel information, and shows to be more stable under silhouette pose ambiguities.

Recently, machine learning and deep learning techniques are applied for 6D pose tracking. Previous works [KMB*14, TNT17] train a random forest on RGB-D images and directly regress the object pose. [GL17] presents the first deep learning-based 3D object tracking method using RGB-D images, and the later work [GLL18] improves the network architecture and achieves better tracking performance. [LWJ*18] proposes a deep neural network for 6D pose matching, which is able to iteratively refine the pose by matching the rendered image against the observed image. Although some deep learning-based 3D pose refinement method can be adapted for monocular 3D object tracking, but the training process is very time-consuming and limits its capability for tracking unseen object, so far it is still not adopted in the literature of monocular 3D object tracking.

Compared to 3D object tracking that uses consecutive video frames, 3D object detection uses only a single image to estimate the object pose, so it requires large computation and is less accurate. Early template-based 3D object detection methods [HCI*11, HLI*12, RCT13] first generate a large number of 2D templates that represent the object appearance at different poses, then the object pose is determined by comparing the input image and the templates. Recent deep learning-based 3D object detection methods [KMT*17, TSF18, PLH*19, FRM*20] train the deep networks to directly regress the object pose, or to matching the keypoints and then estimate the pose via PnP algorithm. Due to the large computation burden, 3D object detection is not suitable for accurate pose estimation on low-power devices, but it can be used to estimate the initial pose of tracking and relocate the pose when tracking is lost.

3. Problem Statement

In this work, the pose which transforms 3D model point from the object coordinate to the camera coordinate is represented by homogeneous matrix

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{SE}(3). \tag{1}$$

We assume that the pose $\boldsymbol{T}_{t-1}$ in previous frame is known, By computing the inter-frame motion $\Delta\boldsymbol{T}$, the object pose $\boldsymbol{T}_t$ in current frame can be computed as $\boldsymbol{T}_t = \Delta\boldsymbol{T}\boldsymbol{T}_{t-1}$.

For pose optimization, the inter-frame motion $\Delta\boldsymbol{T}$ is parameterized by twist vector $\Delta\boldsymbol{p} = [w_1, w_2, w_3, t_1, t_2, t_3]^\top \in \mathbb{R}^6$ using Lie algebra representation, and the matrix exponential $\Delta\boldsymbol{T} = \exp(\hat{\boldsymbol{p}}) \in \mathbb{SE}(3)$ maps a twist to its matrix representation. With the object pose $\boldsymbol{T}$, the 3D point $\boldsymbol{X}$ on object surface is projected into the image plane with project function

$$\boldsymbol{x} = \pi(\boldsymbol{K}(\boldsymbol{T}\tilde{\boldsymbol{X}})_{3\times 1}), \tag{2}$$

with $\pi(\boldsymbol{X}) = [X/Z, Y/Z]^\top$. The tilde-notation marks the homogeneous representation $\tilde{\boldsymbol{X}} = [X, Y, Z, 1]^\top$ of $\boldsymbol{X} = [X, Y, Z]^\top = (\tilde{\boldsymbol{X}})_{3\times 1}$. The camera is calibrated, and the intrinsic matrix $\boldsymbol{K}$ is known.

In order to compute the inter-frame motion $\Delta\boldsymbol{T}$, edge-based methods must establish the 3D-2D correspondences between the 3D model points and object contour points. These correspondences are used to set up a non-linear least squares problem that minimizes the distance between the object contour point $\boldsymbol{s}_i$ and the projected contour point $\boldsymbol{m}_i$ corresponding to the 3D model point $\boldsymbol{X}_i$:

$$\begin{aligned} E(\Delta\boldsymbol{T}) &= \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{s}_i - \pi(\boldsymbol{K}(\Delta\boldsymbol{T}\boldsymbol{T}_{t-1}\tilde{\boldsymbol{X}}_i)_{3\times 1})\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{s}_i - \boldsymbol{m}_i\|^2, \end{aligned} \tag{3}$$

where N is the number of $\boldsymbol{s}_i$. By solving this non-linear least squares problem, we obtain the accurate pose of the object. The critical problem is that the object contour point $\boldsymbol{s}_i$ corresponding to the projected contour point $\boldsymbol{m}_i$ must be accurately determined in the image. However, partial occlusions and cluttered backgrounds often cause false object contour points, which easily lead to tracking drift and tracking lost. In this work we aim to accurately search the object contour points and properly handle the unavoidable false object contour points caused by partial occlusions and cluttered backgrounds.

4. Proposed Method

In this section, we first describe our method to search the object contour points, then we present the confidence-based pose optimization method for handling false object contour points. Finally, we extend our method to multi-object tracking.

4.1. Candidate contour points

We use search lines to efficiently establish the correspondences between the projected contour points $\boldsymbol{m}_i$ and the object contour points $\boldsymbol{s}_i$. As shown in Fig. 2, by projecting the 3D model into the image plane we obtain a silhouette mask I_s. The projected contour segments the silhouette mask into a foreground region Ω_f and a background region Ω_b. At each projected contour point $\boldsymbol{m}_i$, a 1D search line l_i is sampled along the unit vector $\boldsymbol{n}_i$ perpendicular to the projected contour. Each search line l_i has N_r sampling points $\boldsymbol{x}_{ij}$ in the foreground region and background region. In order to efficiently access the information of the search lines, similar to [WWZ*15], we build a bundle image B by stacking all search lines. As shown in Fig. 2, for a bundle image B the left part corresponds to the background region Ω_b, the middle column is the projected contour points $\boldsymbol{m}_i$, the right part corresponds to the foreground region Ω_f and the i-th row corresponds to search line l_i. Since the object contour points often exist in the intensity change, we use Canny Edge Detector to extract edges in the video frame and obtain the edge map I_e. For each sampling point $\boldsymbol{x}_{ij}$ we check whether $I_e(\boldsymbol{x}_{ij})$ is an edge point, if so, $\boldsymbol{x}_{ij}$ is labeled as a candidate point $\boldsymbol{h}_{ij}$.

4.2. Clutter points suppression

As shown in Fig. 3, there are many clutter points among the candidate points due to cluttered backgrounds, complex texture of the target object and the occluding object. We exploit the edge color cue which is modelled by the appearance of the foreground region

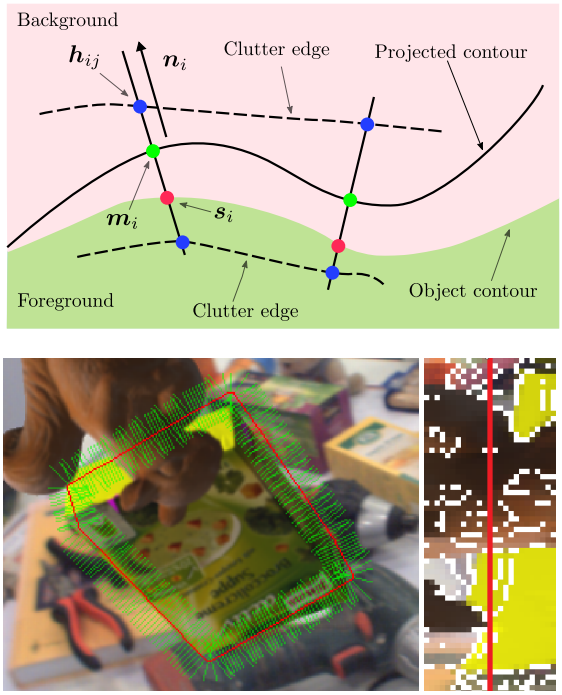

Figure 2: *Searching the candidate contour points. Top: The search lines are sampled along the projected contour. Bottom left: The search lines (green lines) and the projected contour (red dots) of the 3D model in the previous object pose T_{t-1}. Bottom right: Part of the bundle image. The white dots denote the candidate contour points $\boldsymbol{h}_{ij}$.*

and the background region to filter out these clutter points and keep only the candidate points with higher confidence. For a real object contour point in the bundle image B (see Fig. 2), its left area should belong to the background, and its right area should belong to the object surface (foreground). Assuming pixel-wise independence, we define two relative probabilities that the left area $\Phi^-_{\boldsymbol{h}_{ij}}$ of candidate point $\boldsymbol{h}_{ij}$ belongs to the foreground and background as

$$P(\Phi^-_{\boldsymbol{h}_{ij}}|F) = \prod_{\boldsymbol{x}\in\Phi^-_{\boldsymbol{h}_{ij}}} P_f(\boldsymbol{x}) \text{ and } P(\Phi^-_{\boldsymbol{h}_{ij}}|B) = \prod_{\boldsymbol{x}\in\Phi^-_{\boldsymbol{h}_{ij}}} P_b(\boldsymbol{x}), \tag{4}$$

where $P_f(\boldsymbol{x})$ and $P_b(\boldsymbol{x})$ are the foreground and background posterior probabilities of pixel $\boldsymbol{x}$, which are calculated by temporally consistent local color histograms [HSSC19]. The left area $\Phi^-_{\boldsymbol{h}_{ij}}$ of candidate point $\boldsymbol{h}_{ij}$ is defined as 3 sampling pixels from $\boldsymbol{x}_{i,j-1}$ to $\boldsymbol{x}_{i,j-3}$. The two probabilities that the right area $\Phi^+_{\boldsymbol{h}_{ij}}$ of candidate point $\boldsymbol{h}_{ij}$ belongs to the foreground and background are defined as

$$P(\Phi^+_{\boldsymbol{h}_{ij}}|F) = \prod_{\boldsymbol{x}\in\Phi^+_{\boldsymbol{h}_{ij}}} P_f(\boldsymbol{x}) \text{ and } P(\Phi^+_{\boldsymbol{h}_{ij}}|B) = \prod_{\boldsymbol{x}\in\Phi^+_{\boldsymbol{h}_{ij}}} P_b(\boldsymbol{x}), \tag{5}$$

where the right area $\Phi^+_{\boldsymbol{h}_{ij}}$ of candidate point $\boldsymbol{h}_{ij}$ is defined as 3 sampling pixels from $\boldsymbol{x}_{i,j+1}$ to $\boldsymbol{x}_{i,j+3}$. Then the probabilities that the candidate point belongs to the object contour points, the foreground

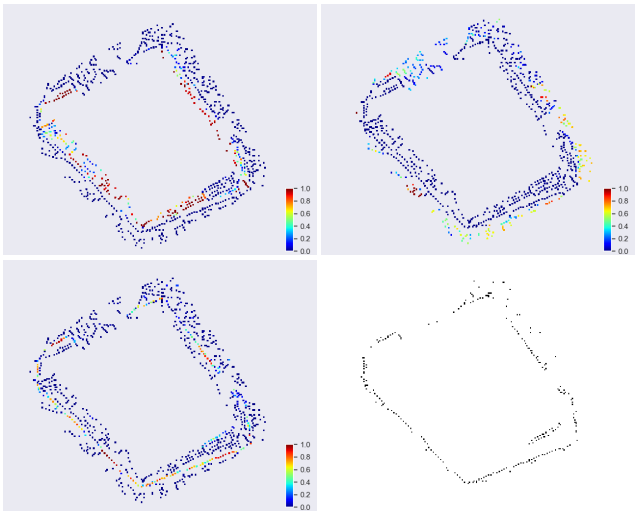

Figure 3: *Clutter points suppression. Top left:The probability map of $P(\boldsymbol{h}_{ij}|F)$. Top right: The probability map of $P(\boldsymbol{h}_{ij}|B)$. Bottom left: The probability map of $P(\boldsymbol{h}_{ij}|C)$. Bottom right: The remaining candidate points after clutter points suppression.*

clutter points and the background clutter points can be defined as

$$\begin{aligned} P(\boldsymbol{h}_{ij}|C) &= P(\Phi^{-}_{\boldsymbol{h}_{ij}}|B)P(\Phi^{+}_{\boldsymbol{h}_{ij}}|F), \\ P(\boldsymbol{h}_{ij}|F) &= P(\Phi^{-}_{\boldsymbol{h}_{ij}}|F)P(\Phi^{+}_{\boldsymbol{h}_{ij}}|F), \\ P(\boldsymbol{h}_{ij}|B) &= P(\Phi^{-}_{\boldsymbol{h}_{ij}}|B)P(\Phi^{+}_{\boldsymbol{h}_{ij}}|B). \end{aligned} \tag{6}$$

Candidate point $\boldsymbol{h}_{ij}$ is considered to be object contour point if $P(\boldsymbol{h}_{ij}|C)$ is larger than both $P(\boldsymbol{h}_{ij}|F)$ and $P(\boldsymbol{h}_{ij}|B)$. We keep the candidate points satisfy this condition, then the clutter points caused by cluttered backgrounds and partial occlusions are filtered out (see Fig. 3, bottom right).

4.3. The confidence of contour points

For the remaining candidate points, we can select the candidate point with maximum $P(\boldsymbol{h}_{ij}|C)$ as the matched object contour point of each search line l_i. However, this naive approach may result in many false object contour points, especially in the presence of partial occlusion and cluttered background. Therefore, we propose to regularize the selection of image contour points with the spatial distance to the projected model contour. When the color probability is not fully reliable, the spatial distance could be used to remove candidate points that may introduce large error.

To indicate the quality of each remaining candidate point, we assign each point $\boldsymbol{h}_{ij}$ a confidence value that combines the confidences of spatial distance and color probability

$$w(\boldsymbol{h}_{ij}) = w_d(\boldsymbol{h}_{ij})w_c(\boldsymbol{h}_{ij}), \tag{7}$$

where $w_d(\boldsymbol{h}_{ij})$ is the confidence calculated with spatial distance, and $w_c(\boldsymbol{h}_{ij})$ is the confidence calculated with color probability. Since the object motion between consecutive frames is not too fast, we assume the correct object contour points tend to have a small

Figure 4: *Left: Searched object contour points $\boldsymbol{s}_i$ (green dots) in the video frame. Right: The confidence $w(\boldsymbol{s}_i)$ of object contour point $\boldsymbol{s}_i$.*

distance to the projected contour. and the occluded object contour points or background clutter points tend to be far away from the projected contour. So we choose to use Tukey weight function to calculate the confidence of candidate point $\boldsymbol{h}_{ij}$ measured by spatial distance

$$w_d(\boldsymbol{h}_{ij}) = \begin{cases} [1-(D(\boldsymbol{h}_{ij})/\lambda_d)^2]^2 & \text{if } D(\boldsymbol{h}_{ij}) \le \lambda_d, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $D(\boldsymbol{h}_{ij})$ is the distance from the candidate point $\boldsymbol{h}_{ij}$ to the projected contour point $\boldsymbol{m}_i$, λ_d is a threshold for the maximum valid distance. This function implies that candidate point $\boldsymbol{h}_{ij}$ which is too far from the projected contour is likely to be outlier point caused by cluttered background or partial occlusions, so it should have low confidence. For better balancing the effect of edge distance and color cues, confidence of candidate point $\boldsymbol{h}_{ij}$ measured by color probability is defined as

$$w_c(\boldsymbol{h}_{ij}) = [1-(1-P(\boldsymbol{h}_{ij}|C))^2]^2. \tag{9}$$

With $w_c(\boldsymbol{h}_{ij})$ increasing with the probability $P(\boldsymbol{h}_{ij}|C)$, thus giving a higher confidence to the candidate $\boldsymbol{h}_{ij}$ on correct object contour. Then we calculate the joint confidence $w(\boldsymbol{h}_{ij})$ at each candidate point on the search line l_i, and select the candidate point with the maximum joint confidence as the object contour point $\boldsymbol{s}_i$ corresponding to the projected contour point $\boldsymbol{m}_i$.

4.4. Pose optimization with edge confidence

With the above approach, only reliable contour points will be selected for pose optimization. However, incorrect contour points are still unavoidable in more complicated cases of cluttered backgrounds and partial occlusions. The effect of these incorrect contour correspondences must be suppressed during pose optimization. For this purpose, we propose to incorporate the confidence of contour points for adaptive weighting the energy terms. Since the confidence function is in the range of $[0,1]$, we directly incorporate it to Eq. (3), and rewrite the energy function as a non-linear iteratively reweighted least squares problem

$$\begin{aligned} E(\boldsymbol{p}) &= \frac{1}{2}\sum_{i=1}^{N} w(\boldsymbol{s}_i)F^2(\boldsymbol{s}_i), \quad \text{where} \\ F(\boldsymbol{s}_i) &= \boldsymbol{n}_i^{\top}(\boldsymbol{s}_i - \boldsymbol{m}_i). \end{aligned} \tag{10}$$

As shown in Fig. 4, a non-occluded contour point is assigned with a large weight, and an occluded contour point is assigned with a small

weight. Therefore, the influence of the occluded contour points is strongly suppressed, which improves robustness to partial occlusions. Using the last estimated pose as an initial value, this problem can be solved by applying the Gauss-Newton optimization with fixed weights and alternatingly using the refined pose for updating the weights $w(\boldsymbol{s}_i)$. We denote the Jacobian of the residual by

$$J(\boldsymbol{s}_i) = \frac{\partial F(\boldsymbol{s}_i)}{\partial \boldsymbol{p}} = \boldsymbol{n}_i^\top \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}}, \tag{11}$$

where $\boldsymbol{n}_i$ is the normal vector of the i-th search line. $\frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}}$ can be derived from Eq. (2), the detail can be found in [HSSC19]. Then the optimal Gauss-Newton update step is computed as

$$\Delta \boldsymbol{p} = -\left(\sum_{i=1}^{N} w(\boldsymbol{s}_i) \boldsymbol{J}(\boldsymbol{s}_i)^\top \boldsymbol{J}(\boldsymbol{s}_i) \right)^{-1} \sum_{i=1}^{N} w(\boldsymbol{s}_i) \boldsymbol{J}(\boldsymbol{s}_i)^\top F(\boldsymbol{s}_i). \tag{12}$$

Finally, the pose is updated by applying this step as composition of the exponential matrix of the twist $\Delta \boldsymbol{p}$ with the previous pose as

$$\boldsymbol{T} \leftarrow \exp(\Delta \hat{\boldsymbol{p}}) \boldsymbol{T}. \tag{13}$$

In order to improve robustness to large inter-frame motion, the pose is iteratively optimized by a hierarchical coarse-to-fine approach as in [HSSC19]. For a video frame, a three level image pyramid is generated with a down-scale factor of 2, then four iterations are performed at the top level, two iterations at the lower level and one iteration at the lowest level.

In pose optimization process, it can easily happen that only a few object contour points are searched due to heavy occlusions and fast illumination changes. The 3D-2D correspondences computed from such few object contour points could typically be less trusted. To handle this case we compute the matching rate which is the proportion of searched object contour points in all sampled projected contour points. If it is less than 10%, we do not perform pose optimization and keep the pose parameters unchanged. When the object is far from the camera or partially moves out of the frame, the object region only contains a small amount of pixels. This could also result in error-prone correspondences that make the optimization converge to a local minimum. We compute the 2D bounding rectangle of the projected 3D bounding box of the model. If its area is less than 3600 pixels, the current pose keeps unchanged and move to the lower pyramid level image, or the estimated pose is accepted and used to update the current pose. This pose updating strategy is important to enforce temporal consistency in pose optimization which improves the robustness of pose tracking, because the error could accumulate in several iterations and degrade the overall optimization quality.

4.5. Extension to multi-object tracking

Similar to [HSSC19], our method can be extended to track multiple objects simultaneously. For this each object is represented by its own 3D model, by rendering all models with a unique color that corresponds to its object index k, we obtain a common silhouette mask image I_s. Then we can extract respective projected contours that segment the silhouette mask image into corresponding foreground regions Ω_f^k and background regions Ω_b^k (see Fig. 5).

When each object is not occluded by other objects, the pose of each object is optimized regardless of the other objects. However, in case of mutual occlusions, the overlapped foreground regions will cause occluded projected contour points, which do not belong to the real contour of the objects. These occluded projected contour points (marked red in Fig. 5) may result in false 3D-2D correspondences, so they must be correctly detected and handled for pose optimization. We use the silhouette mask image I_s and depth image I_d to detect the occluded contour points. For a projected contour point $\boldsymbol{m}_i = \boldsymbol{x}_{ij}$, we check the sampling point $\boldsymbol{x}_{i,j-1} \in \Omega_b^k$ which is in the background region and adjacent to $\boldsymbol{x}_{ij}$, if the object index $I_s(\boldsymbol{x}_{i,j-1})$ is identified as other object and the depth $I_d(\boldsymbol{x}_{i,j-1})$ is less than the depth of the projected contour point $I_d(\boldsymbol{m}_i)$, meaning that other object is in front of the current object, then $\boldsymbol{m}_i$ is considered to be occluded contour point. Since these occluded contour points will move the pose optimization to a wrong direction, they are discarded in pose estimation.

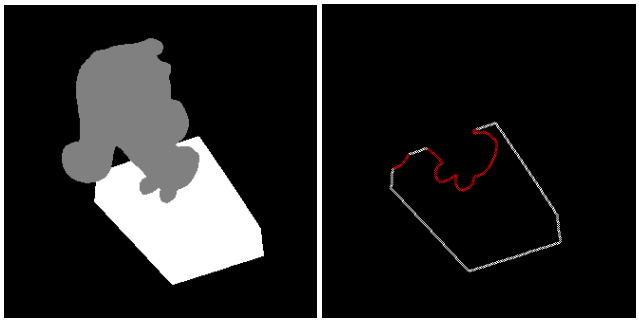

Figure 5: *Occlusion handling when tracking multiple objects. Left: Common silhouette mask I_s, which is generated by rendering each object model with a unique color equals to its object index. A box is occluded by a squirrel in the scene. Right: The projected contour points of the box object. The occluded contour points (marked as red) are discarded in pose optimization.*

5. Experiment

In this section we first provide detail description and parameters settings of the implementation. Then we provide extensive evaluation in public OPT dataset [WLT*17] and RBOT dataset [HSSC19]. We compare the proposed method with two edge-based methods [WWZ*15, WZQ19] and two state-of-art region-based methods [HSSC19, ZZ19]. Finally, we demonstrate the application to augmented reality. For all of these experiments we evaluate our implementation on a laptop with Intel i5 8300H CPU, Intel UHD630 GPU and 16GB RAM.

5.1. Implementation details

We render the 3D models with OpenGL, and obtain the silhouette mask image I_s and depth image I_d. In order to extract projected contour points $\boldsymbol{m}_i$ in silhouette mask image I_s, we first use OpenCV function `findContours` to extract all contour points. Then we discard the contour points on the image border, since they are not the real contour points. Given the depth image I_d, we use OpenGL

function `gluUnProject` to compute the 3D model points $\boldsymbol{M}_i$ corresponding to projected contour points $\boldsymbol{m}_i$, which is required for computing the derivatives of energy function. In order to speed up tracking, all image processing procedures (e.g. canny edge detection and contour points extraction) are restricted in a 2D ROI (region of interest) which is given by expanding the bounding box of the object silhouette by 16 pixels in all directions. The length of the search lines N_r is set to 25 points (12 foreground points, 1 contour point, and 12 background points). The parameter λ_d in Eq. (8) is set to 10.

5.2. Evaluation on OPT dataset

The OPT dataset [WLT*17] contains 6 objects and 552 real-world image sequences with a total number of 79,968 frames at 1920×1080 px resolution. The dataset covers different camera motion and lighting condition. However, it does not contain cluttered backgrounds and partial occlusions. The objects are placed in entirely white background (see Fig. 6). We use the same evaluation metric as that in [WLT*17]. The pose error is computed as

$$e(\boldsymbol{T}) = \frac{1}{n}\sum_{i=1}^{n} \|(\boldsymbol{T}\boldsymbol{X}_i - \boldsymbol{T}_{gt}\boldsymbol{X}_i)\|, \tag{14}$$

where $\boldsymbol{T}_{gt}$ is the ground truth pose, $\boldsymbol{X}_i$ is a vertex of the 3D object model. The tracking is considered to be successful if $e(\boldsymbol{T}) < k_e d_m$, where d_m is the diameter of the 3D model (the largest distance between vertices) and k_e is a tunable threshold. Then the tracking success rates (the percentage of successfully tracked frames) are measured by varying $k_e \in [0, 0.2]$ in a precision plot. The tracking performance is evaluated in form of AUC (area under curve) score, where the tracking success rates (0∼100) are integrated for all $k_e \in [0, 0.2]$ (so the range of AUC score is $[0, 20]$, higher is better). For each image sequence, the ground truth pose is only used to initialize tracking at the first frame, and the pose is never reset to ground truth when tracking is lost.

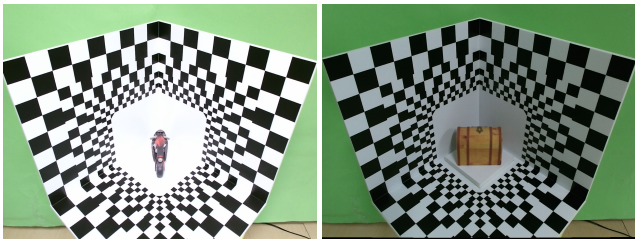

Figure 6: *Sample images of the OPT dataset.*

Table 1: *Evaluation results in the OPT dataset (AUC scores). TR: translation, ZO: zoom, IR: in-plane rotation, OR: out-of-plane rotation, FL: flashing light, ML: moving light, FM: free motion.*

Method	TR	ZO	IR	OR	FL	ML	FM
[WWZ*15]	6.89	**7.18**	**9.89**	7.96	**10.67**	10.30	1.22
[WZQ19]	5.27	5.07	5.92	5.41	0.53	9.06	1.29
[HSSC19]	9.11	6.48	7.48	8.36	7.26	8.17	1.87
Ours	**9.13**	6.53	9.61	**9.99**	9.85	**10.69**	**4.71**

Tab. 1 shows the results for each method with respect to different motion patterns and lighting conditions. Authors of [ZZ19] do not provide results in the OPT dataset, so it is not listed here. Our method performs comparable with the edge-based method [WWZ*15], and better than the RBOT method [HSSC19] and [WZQ19]. Compared to [WWZ*15], our method shows obvious advantage in case of translation (TR) and free motion (FM) conditions, because [WWZ*15] cannot find blurred edges in frames containing motion blur, and it cannot properly handle large inter-frame motion without hierarchical coarse-to-fine pose optimization strategy. Note that as mentioned above, the OPT dataset contains only white background, in which case the advantages of our method and [HSSC19] cannot be shown. [WWZ*15] use dynamic programming to optimize the object contour, so in this case it can be more stable. However, in real environments with cluttered backgrounds and occlusions, the performance of [WWZ*15] will be reduced dramatically, as will be shown below. The results also show the limitation of [WZQ19]. Since distance transform always searches edge points nearest to the projected contour point, this simple searching scheme will cause many false object contour points when the object or background contains clutter edges. Note that the pose is never reset to ground truth when tracking is lost, so the evaluation results in OPT dataset may be misleading. For example, if tracking is lost in the early frames, the score will be very low, such as the scores in free motion (FM) condition, but this does not mean the tracker cannot track the object in the remainder frames, if the tracker is reset with ground truth pose.

5.3. Evaluation on RBOT dataset

The RBOT dataset [HSSC19] contains 18 objects and 72 image sequences. The objects vary in shape and texture, including well-textured and weakly-textured objects. Each image sequence contains 1000 frames of 640 × 512 px resolution. This dataset is semi-synthetic, in which the virtual objects are rendered with OpenGL and composed with a real background video captured with a hand-held camera. In order to increase realism, the objects are rendered with anti-aliasing and blurred using a 3 × 3 Gaussian kernel. In order to increase complexity, for each object the dataset composes 4 different variants of image sequences: regular, dynamic light, noisy, and occlusion. We handle the occlusions in two different ways as in [HSSC19]. For the case of *modelled* occlusion, both the varying object and the occluding object are tracked simultaneously and the occlusions are handled using method described in Sec. 4.5. For the case of *unmodelled* occlusion, only the varying object is tracked, and the pose of the occluding object is unknown, so it is more difficult to be handled. With the additional occlusion variant, we actually need to process 5 variants of 90 image sequences (total number of 90,000 frames). We use the same evaluation metric as that in [HSSC19]. For each frame we compute the translation error $e(\boldsymbol{t}) = \|\boldsymbol{t} - \boldsymbol{t}_{gt}\|$ and the the rotation error

$$e(\boldsymbol{R}) = \cos^{-1}\left(\frac{trace(\boldsymbol{R}^\top \boldsymbol{R}_{gt}) - 1}{2}\right). \tag{15}$$

If $e(\boldsymbol{t})$ is below 5 cm and $e(\boldsymbol{R})$ below 5°, the tracking is considered to be successful. Otherwise, the tracking is considered to be lost, and the pose is reset to the ground truth.

Table 2: *Evaluation results in the RBOT dataset (Tracking success rates in %). REG: Regular, DYN: Dynamic light, NOI: Noisy, UOC: Unmodelled occlusion, MOC: Modelled occlusion. The 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	[WWZ*15]	13.3	13.3	16.1	16.0	15.1	14.6	19.3	14.0	13.9	16.6	16.5	11.0	13.4	15.6	8.7	15.9	13.5	18.2	14.7
	[WZQ19]	22.4	19.4	22.9	21.3	23.3	19.8	20.7	23.1	22.7	22.4	20.5	21.2	19.7	22.2	21.0	23.0	24.9	22.6	21.8
	[HSSC19]	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
	[ZZ19]	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	**91.9**	**44.8**	**99.7**	**89.1**	**89.3**	**90.6**	97.4	**95.9**	**83.9**	**97.6**	**91.8**	**84.4**	**59.0**	**92.5**	**74.3**	97.4	**86.4**	**99.7**	**86.9**
DYN	[WWZ*15]	13.1	12.2	15.3	15.9	14.4	14.2	18.8	12.3	14.0	15.8	16.2	12.7	13.0	14.9	8.8	14.3	12.4	18.1	14.2
	[WZQ19]	20.7	19.9	23.7	21.5	23.5	22.4	20.1	21.9	22.7	23.8	19.9	22.4	18.2	24.4	20.8	22.4	24.4	23.6	22.0
	[HSSC19]	84.9	42.0	**99.0**	81.3	84.3	**88.9**	95.6	92.5	77.5	94.6	86.4	77.3	52.8	77.9	47.9	96.9	81.7	99.3	81.2
	[ZZ19]	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	**91.8**	**42.3**	98.9	**89.9**	**91.3**	87.8	97.6	**94.5**	**84.5**	**98.1**	**91.9**	**86.7**	**66.2**	**90.9**	**73.2**	**97.1**	**89.2**	**99.6**	**87.3**
NOI	[WWZ*15]	10.4	10.3	12.3	14.5	6.5	8.4	16.6	11.4	6.9	9.3	17.5	3.7	5.6	11.4	5.8	9.6	7.3	12.9	10.0
	[WZQ19]	19.8	21.4	21.2	21.0	22.0	20.3	19.8	21.2	22.4	20.4	20.4	20.7	**19.1**	20.4	20.9	20.6	21.0	20.8	20.7
	[HSSC19]	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
	[ZZ19]	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	**89.0**	**45.0**	89.5	**90.2**	**68.9**	38.3	95.9	**72.8**	20.1	**85.5**	**92.2**	26.8	15.8	**66.2**	**52.2**	58.3	**65.1**	**98.4**	**65.0**
UOC	[WWZ*15]	12.1	12.0	14.6	13.9	13.5	14.7	18.4	13.1	13.2	16.8	17.0	11.6	11.6	15.1	8.0	14.3	14.3	16.7	13.9
	[WZQ19]	21.8	19.2	23.4	21.2	22.1	20.2	21.6	21.1	22.3	22.5	19.5	20.5	18.6	23.5	20.7	22.4	24.7	23.0	21.5
	[HSSC19]	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
	[ZZ19]	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	**86.2**	**46.3**	**97.8**	**87.5**	**86.5**	**86.3**	**95.7**	**90.7**	**78.8**	**96.5**	**86.0**	**80.6**	**59.9**	**86.8**	**69.6**	**93.3**	**81.8**	**95.8**	**83.6**
MOC	[HSSC19]	82.0	42.0	95.7	81.1	78.7	83.4	92.8	87.9	74.3	91.7	84.8	71.0	49.1	73.0	46.3	90.9	76.2	96.9	77.6
	Ours	**87.8**	**45.5**	**98.1**	**87.2**	**89.0**	**89.8**	**95.1**	**91.4**	**77.4**	**97.1**	**87.7**	**83.0**	**62.5**	**88.6**	**69.7**	**94.1**	**86.0**	**98.9**	**84.9**
Runtime	[WWZ*15]	42.6	40.7	40.8	39.1	46.8	44.5	40.1	42.5	44.1	43.2	38.2	47.4	41.6	40.8	51.4	43.0	48.0	39.6	43.1
	[HSSC19]	33.2	36.1	37.7	**32.4**	33.6	**34.7**	34.9	**26.9**	**31.9**	36.1	**31.5**	32.3	**33.6**	34.5	**28.8**	37.8	33.8	35.5	33.6
	Ours	**32.1**	36.1	**33.6**	32.8	**32.6**	34.8	**33.0**	30.2	32.6	**34.0**	31.8	**32.2**	33.9	**33.9**	31.4	**35.6**	**33.1**	**33.6**	**33.1**

Tab. 2 summarizes the tracking success rates (SR) for all sequences. The results show that our method achieves the best SR scores on 78 out of the 90 sequences, and obtains the highest average SR score on all of the 5 variants. The edge-based methods [WWZ*15, WZQ19] achieve poor accuracy, because their contour searching schemes are sensitive to cluttered backgrounds. Our method improves the average SR score by about 6% compared to the recent state-of-art region-based methods [HSSC19, ZZ19]. This can be explained by the proposed contour searching strategy, since the edge confidence $w(\boldsymbol{s}_i)$ that combines edge color and distance cues can suppress the clutter edges in cluttered backgrounds. Specifically, for the sequences of unmodelled occlusion variant our method achieves the best SR scores on all 18 sequences, and the average SR score is improved by 10% compared with the state-of-art. This proves the advantage of the proposed occlusion handling strategy, since edge confidence $w(\boldsymbol{s}_i)$ generally reduces the influence of the occluded edges in pose optimization. However, our method performs less accurate for objects with a less distinct color (e.g. Can, Cube and Glue), owing to the similarity in the appearance of these objects to the background, and the tracking performance is worse for objects with symmetrical structure (e.g. Baking Soda, Glue and Koala Candy), due to the silhouette ambiguity in some pose.

In order to verify the effectiveness of using edge color and distance cues for improving the tracking accuracy, we setup up different combinations of edge color and distance cues in object contour points searching and pose optimization, and summarize the average tracking SR scores in Tab. 3. When only using the proposed edge color cue in object contour points searching (CS column), the candidate points $\boldsymbol{h}_{ij}$ on the search line l_i with maximum $P(\boldsymbol{h}_{ij}|C)$ is selected as the object contour point $\boldsymbol{s}_i$. The tracking performance outperforms the previous edge-based methods [WWZ*15, WZQ19] by a large margin, which proves that the edge color probability

Table 3: *The effect of different combinations of edge color and spatial distance for contour searching and pose optimization. The shown accuracy is the same as that in Tab. 2. CS=color based (contour) searching, CDS=color+distance based searching, DW=distance based weighting, CDW=color+distance based weighting. Please see the text for more explanations.*

Variant	CS	CDS	CDS+ DW	CDS+ CDW
REG	73.4	76.8	82.1	**86.9**
DYN	72.6	76.6	82.4	**87.3**
NOI	44.9	49.6	59.2	**65.0**
UOC	62.3	66.2	75.1	**83.6**
MOC	71.6	74.6	78.7	**84.9**
Average	64.9	68.7	75.5	**81.5**

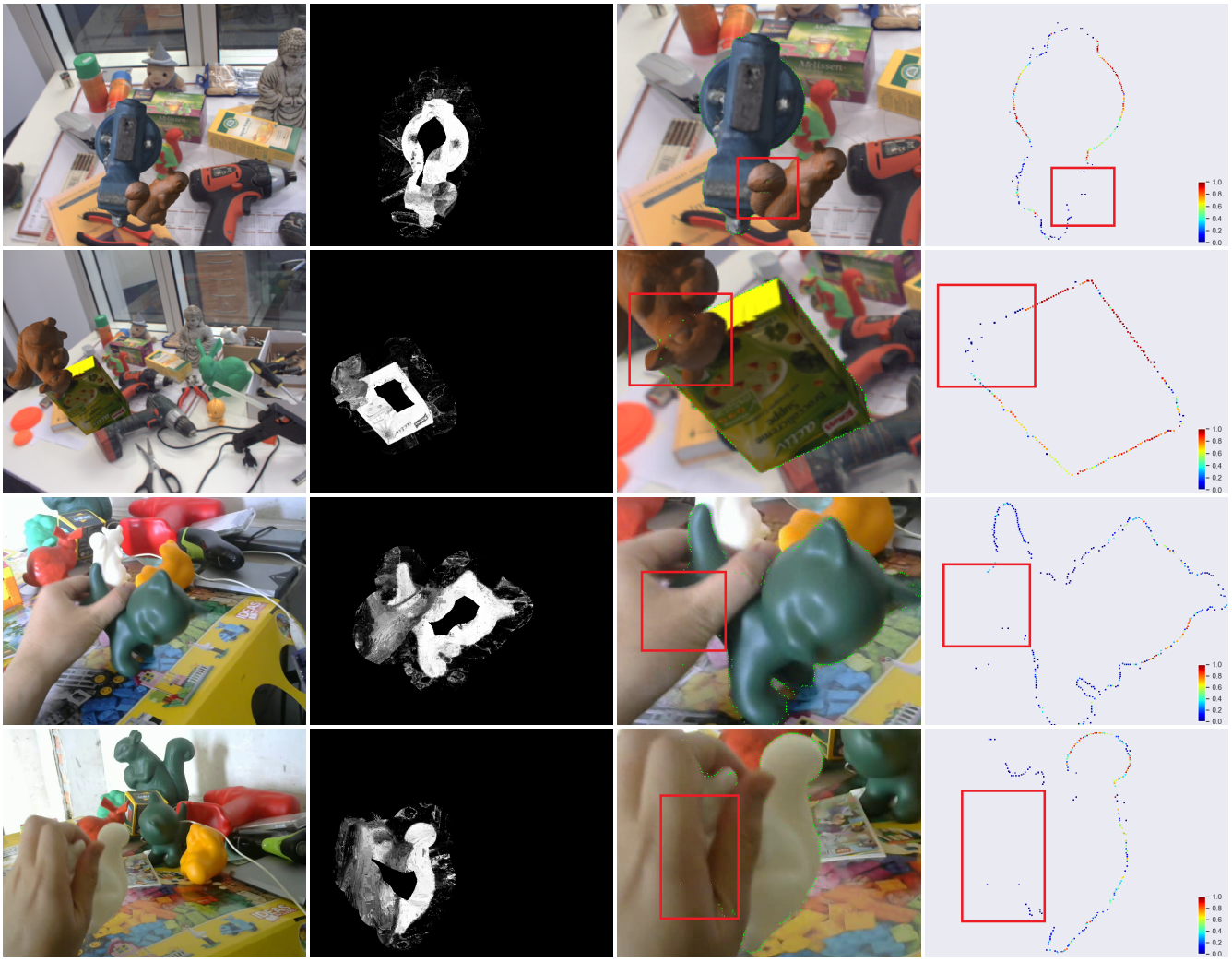

Figure 7: *Analysis of edge confidence for dealing with partial occlusions in cluttered backgrounds. The first to the fourth column: input image, foreground probability map, searched object contour points (green dots) and confidence of each object contour point.*

$P(\boldsymbol{h}_{ij}|C)$ can suppress clutter points as described in Sec. 4.2. When using the proposed edge confidence which combines edge color and edge distance in object contour points searching (CDS column), the average SR scores is improved by 4%, which proves the effectiveness of the confidence-based searching strategy proposed in Sec. 4.3. Based on this, when only using traditional edge distance to weight the energy function (CDS+DW column), the tracking performance already outperforms the current state-of-the-arts. When using the the proposed edge confidence which combines both edge color and edge distance to weight the energy function (CDS+CDW column), the average SR score of our method outperforms [HSSC19, ZZ19] by ∼7%, which proves the effectiveness of the confidence-based pose optimization method proposed in Sec. 4.4.

5.4. Analysis of edge confidence

In order to demonstrate the effectiveness of the edge confidence in object contour searching and pose optimization, as shown in Fig. 7, we select some frames that contain partial occlusions and cluttered backgrounds, and visualize the confidence of each object contour point in the confidence map. The frames in the first and second rows are selected from RBOT dataset, in which the target objects are occluded by a squirrel object. The frames in the third and fourth rows are selected from real world videos, in which the target objects are occluded by user's hand. Partial occlusion results in many misclassified pixels (outliers) in the corresponding probability map which will lead to many false object contour points in the occluded region, but the clutter points caused by occlusions are almost filtered out by the proposed clutter points suppressing strategy, and the remaining clutter points are assigned with small confidences (see the points in red rectangle), so we can reduce the negative impact of these clutter

Figure 8: *Application to augmented reality. Top: RGB video frames. Bottom: Augmented frames. Based on the estimated pose, the augmentations remain precise in the scenes containing cluttered backgrounds and partial occlusions.*

points in pose optimization and improve robustness against partial occlusions.

5.5. The time cost

Tab. 2 also shows the average runtime (in ms) measured with the first three image sequences of each object. Note that the runtime is related with several factors, including the mesh size of the 3D model, the distance of the object from the camera (affecting the size of the object projection). Our method achieves real-time performance (30 - 40 Hz) for image frames with a resolution of 640×512, and the average runtime is 33.1 ms per frame (rendering: 56.2%, object contour points searching: 19.6%, pose optimization: 5.7%, and the update of color histogram: 18.5%), which is comparable with [HSSC19] and less then [WWZ*15]. The runtime of [WZQ19] is too high ($\sim$1000 ms) due to the time-consuming particle filtering process, so it is not listed in Tab. 2. Currently, we do not utilize parallel computing for acceleration, the runtime of our method can be further reduced by exploiting multi-thread or GPGPU computing techniques.

5.6. Application to augmented reality

Augmented reality (AR) needs to seamlessly insert virtual objects in an real image sequence. In order to accomplish this task, it requires the virtual objects to be aligned with the real objects in an accurate and visually acceptable way. Our method can solve this problem due to its high accuracy and real-time performance. As shown in Fig. 8, the real objects can be realistically occluded by the virtual augmentations since their poses are accurately estimated. What makes it more reliable for AR systems, is that our method can handle fair amounts of cluttered backgrounds and unmodelled occlusions (e.g. by user's hand) when the user manipulate the objects in complex scenario. Therefore, this further allows AR systems to use arbitrary real object (with known 3D model) as input device for motion tracking.

6. Conclusion and Limitation

In this paper, we proposed an edge-based method to improve the robustness of monocular 3D object tracking against cluttered backgrounds and partial occlusions. In searching of the optimal object contour points, candidate points are first evaluated with edge color to filter out numerous false edge points, and the optimal object contour points are searched by comparing their edge confidence which is measured by region color and spatial distance. Then in pose optimization, the edge confidence is also used to weight the energy terms at each object contour points in order to reduce the influence of false object contour points. As the newly defined edge confidence captures the color and distance properties of the object contour points, the object contour point searching and pose optimization are more robust to cluttered backgrounds and partial occlusions.

However, our method still has the following limitations considering more general applications. Firstly, our method uses only the object contour points for pose estimation, it is therefore prone to fail for the objects with symmetrical structure (e.g. Baking Soda, Glue and Koala Candy from the RBOT dataset), since the projected contour are ambiguous in some pose. In future work, we consider combining the inner structural information of the object to further improve the tracking performance. Secondly, our method is still prone to fail when the object is heavily occluded or moves out of the camera's view, since we only focus on pose estimation using consecutive frames, and do not involve automatic pose recovery when tracking is lost. In future work, we consider utilizing 3D object detection for the relocalization of our 3D object tracking system.

7. Acknowledgements

The authors gratefully acknowledge the anonymous reviewers for their comments to help us to improve our paper, and also thank for their enormous help in revising this paper. This work is partially supported by NSF of China (Nos.61672326, 61572290), the Major Project of NSF Shandong

Province (No.ZR2018ZB0420), Industrial Internet Innovation and Development Project in 2019 of China, and Zhijiang Lab (No. 2020NB0AB02).

References

[BC11] BROWN J. A., CAPSON D. W.: A framework for 3d model-based visual tracking using a gpu-accelerated particle filter. *IEEE Transactions on Visualization and Computer Graphics 18*, 1 (2011), 68–80. 2

[DC02] DRUMMOND T., CIPOLLA R.: Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (2002), 932–946. 2

[FRM*20] FELIX H., RODRIGUES W. M., MACÊDO D., SIMÕES F., OLIVEIRA A. L., TEICHRIEB V., ZANCHETTIN C.: Squeezed deep 6dof object detection using knowledge distillation. In *arXiv preprint arXiv:2003.13586* (2020). 3

[GL17] GARON M., LALONDE J.-F.: Deep 6-dof tracking. *IEEE Transactions on Visualization and Computer Graphics 23*, 11 (2017), 2410–2418. 3

[GLL18] GARON M., LAURENDEAU D., LALONDE J.-F.: A framework for evaluating 6-dof object trackers. In *ECCV* (2018), pp. 582–597. 3

[HBN07] HINTERSTOISSER S., BENHIMANE S., NAVAB N.: N3m: Natural 3d markers for real-time object detection and pose estimation. In *ICCV* (2007), pp. 1–7. 1, 2

[HCI*11] HINTERSTOISSER S., CAGNIART C., ILIC S., STURM P., NAVAB N., FUA P., LEPETIT V.: Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 5 (2011), 876–888. 3

[HLI*12] HINTERSTOISSER S., LEPETIT V., ILIC S., STEFAN HOLZER GARY BRADSKI K. K., NAVAB N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV* (2012), pp. 548–562. 3

[HS90] HARRIS C., STENNETT C.: Rapid-a video-rate object tracker. In *BMVC* (1990), pp. 1–6. 2

[HSS16] HENNING T., SCHWANECKE U., SCHOMER E.: Real-time monocular segmentation and pose tracking of multiple objects. In *ECCV* (2016), pp. 423–438. 3

[HSS17] HENNING T., SCHWANECKE U., SCHOMER E.: Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms. In *ICCV* (2017), pp. 124–132. 2, 3

[HSSC19] HENNING T., SCHWANECKE U., SCHÖMER E., CREMERS D.: A region-based gauss-newton approach to real-time monocular multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*, 8 (2019), 1797–1812. 2, 3, 4, 6, 7, 8, 9, 10

[IP15] IMPEROLI M., PRETTO A.: D^2co: Fast and robust registration of 3d textureless objects using the directional chamfer distance. In *Proceedings of International Conference on Computer Vision Systems* (2015), pp. 316–328. 3

[JH16] JONATHAN H., HAGEGE R. R.: 2d-3d pose estimation of heterogeneous objects using a region based approach. *International Journal of Computer Vision 118*, 1 (2016), 95–112. 3

[KM06] KLEIN G., MURRAY D. W.: Full-3d edge tracking with a particle filter. In *BMVC* (2006), pp. 1119–1128. 2

[KMB*14] KRULL A., MICHEL F., BRACHMANN E., GUMHOLD S., IHRKE S., ROTHER C.: 6-dof model based tracking via object coordinate regression. In *ACCV* (2014), pp. 384–399. 3

[KMT*17] KEHL W., MANHARDT F., TOMBARI F., ILIC S., NAVAB N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV* (2017), pp. 1521–1529. 3

[LWJ*18] LI Y., WANG G., JI X., XIANG Y., FOX D.: Deepim: Deep iterative matching for 6d pose estimation. In *ECCV* (2018), pp. 683–698. 3

[MBC01] MARCHAND E., BOUTHEMY P., CHAUMETTE F.: A 2d–3d model-based approach to real-time visual tracking. *Image and Vision Computing 19*, 13 (2001), 941–955. 2

[MUS15] MARCHAND E., UCHIYAMA H., SPINDLER F.: Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics 32*, 12 (2015), 2633–2651. 2

[PLH*19] PENG S., LIU Y., HUANG Q., ZHOU X., BAO H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR* (2019), pp. 4561–4570. 3

[PLW08] PARK Y., LEPETIT V., WOO W.: Multiple 3d object tracking for augmented reality. In *ISMAR* (2008), pp. 117–120. 1, 2

[PR12] PRISACARIU V. A., REID I. D.: Pwp3d: Real-time segmentation and tracking of 3d objects. *International Journal of Computer Vision 98*, 3 (2012), 335–354. 3

[RCT13] RIOS-CABRERA R., TUYTELAARS T.: Discriminatively trained templates for 3d object detection: A real time scalable approach. In *ICCV* (2013), pp. 2048–2055. 3

[SL04] SKRYPNYK I., LOWE D. G.: Scene modelling, recognition and tracking with invariant image features. In *ISMAR* (2004), pp. 110–119. 1, 2

[SPP*13] SEO B.-K., PARK H., PARK J.-I., HINTERSTOISSER S., ILIC S.: Optimal local searching for fast and robust textureless 3d object tracking in highly cluttered backgrounds. *IEEE Transactions on Visualization and Computer Graphics 20*, 1 (2013), 99–110. 2, 3

[TNT17] TAN D. J., NAVAB N., TOMBARI F.: Looking beyond the simple scenarios: Combining learners and optimizers in 3d temporal tracking. *IEEE Transactions on Visualization and Computer Graphics 23*, 11 (2017), 2399–2409. 3

[TSF18] TEKIN B., SINHA S. N., FUA P.: Real-time seamless single shot 6d object pose prediction. In *CVPR* (2018), pp. 292–301. 3

[VF05] VINCENT L., FUA P.: Monocular model-based 3d tracking of rigid objects: a survey. *Foundations and Trends in Computer Graphics and Vision 1*, 1 (2005), 1–89. 2

[VLF04a] VACCHETTI L., LEPETIT V., FUA P.: Combining edge and texture information for real-time accurate 3d camera tracking. In *ISMAR* (2004), pp. 48–56. 2

[VLF04b] VACCHETTI L., LEPETIT V., FUA P.: Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 10 (2004), 1385–1391. 1, 2

[WLT*17] WU P.-C., LEE Y.-Y., TSENG H.-Y., HO H.-I., YANG M.-H., CHIEN S.-Y.: A benchmark dataset for 6dof object pose tracking. In *ISMAR* (2017), pp. 186–191. 6, 7

[WVS05] WUEST H., VIAL F., STRIEKER D.: Adaptive line tracking with multiple hypotheses for augmented reality. In *ISMAR* (2005), pp. 62–69. 2

[WWZ*15] WANG G., WANG B., ZHONG F., QIN X., CHEN B.: Global optimal searching for textureless 3d object tracking. *The Visual Computer 31*, 6-8 (2015), 979–988. 2, 3, 4, 6, 7, 8, 10

[WZQ17] WANG B., ZHONG F., QIN X.-Y.: Pose optimization in edge distance field for textureless 3d object tracking. In *Proceedings of the Computer Graphics International Conference* (2017), pp. 1–6. 2, 3

[WZQ19] WANG B., ZHONG F., QIN X.-Y.: Robust edge-based 3d object tracking with direction-based pose validation. *Multimedia Tools and Applications 78*, 6 (2019), 12307–12331. 2, 3, 6, 7, 8, 10

[ZZ19] ZHONG L., ZHANG L.: A robust monocular 3d object tracking method combining statistical and photometric constraints. *International Journal of Computer Vision 127*, 8 (2019), 973–992. 3, 6, 7, 8, 9