This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

BCOT: A Markerless High-Precision 3D Object Tracking Benchmark

Jiachen Li[1], Bin Wang[1], Shiqiang Zhu[2], Xin Cao[1], Fan Zhong[1], Wenxuan Chen[2],
Te Li[2]*, Jason Gu[3] and Xueying Qin[1]*
[1]Shandong University [2]Zhejiang Lab [3]Dalhousie University

Abstract

Template-based 3D object tracking still lacks a high-precision benchmark of real scenes due to the difficulty of annotating the accurate 3D poses of real moving video objects without using markers. In this paper, we present a multi-view approach to estimate the accurate 3D poses of real moving objects, and then use binocular data to construct a new benchmark for monocular textureless 3D object tracking. The proposed method requires no markers, and the cameras only need to be synchronous, relatively fixed as cross-view and calibrated. Based on our object-centered model, we jointly optimize the object pose by minimizing shape re-projection constraints in all views, which greatly improves the accuracy compared with the single-view approach, and is even more accurate than the depth-based method. Our new benchmark dataset contains 20 textureless objects, 22 scenes, 404 video sequences and 126K images captured in real scenes. The annotation error is guaranteed to be less than 2mm, according to both theoretical analysis and validation experiments. We re-evaluate the state-of-the-art 3D object tracking methods with our dataset, reporting their performance ranking in real scenes. Our BCOT benchmark and code can be found at https://ar3dv.github.io/BCOT-Benchmark/.

1. Introduction

Template-based 3D object tracking aims to estimate the accurate 6DOF pose of moving objects with known 3D models. It is an essential task of computer vision [21], and is widely used in applications that desire high-precision 3D object pose, such as augmented reality, robotic grasping, etc. Despite the rapid development of single-frame 6DOF pose estimation methods [32, 41], for video analysis 3D tracking can be more accurate and more efficient, and thus is indispensable.

Because it is difficult to annotate the accurate 3D pose of a moving object in the real video, it is a great challenge to evaluate 3D tracking methods in real scenes. Previous works use only synthetic datasets or datasets with low precision, the movement of object and camera is also limited. Currently the main adopted datasets include RBOT [38], OPT [45] and YCB-Video [47]. The RBOT dataset is semi-synthetic with rendered moving objects in the real background image sequences, which may be different from real video in camera effects and object movement. The OPT dataset is real captured, but with numerous artificial markers around the object, and the objects are not allowed to move. The YCB-Video is a real RGB-D dataset without markers; however, it contains only static objects, and the pose annotation contains a significant error that prevents it from being used in high-precision scenarios (as required by many AR applications). The above limitations are especially important for the learning-based tracking methods [43, 44], due to the domain difference as well as the bias to the visual cues and object movements in the training dataset.

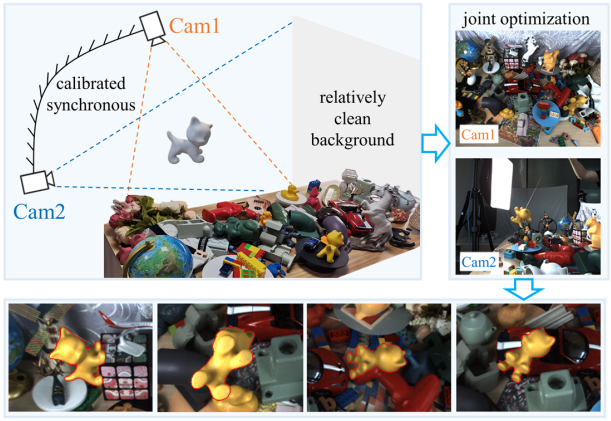


Figure 1. The capturing camera set is composed of two approximately orthogonal cameras. Based on the binocular data and proposed joint optimization framework, we can organize the benchmark with precise annotated pose as rendered in red contour.

Table 1 lists and compares related datasets, including some datasets for single-frame pose estimation [1, 13, 14, 18, 47]. Note that even with the depth camera, it is still difficult to annotate the accurate 3D pose, due to the depth er-

*Corresponding author: Xueying Qin (qxy@sdu.edu.cn) and Te Li (lite@zhejianglab.com)

Dataset	data type	marker-less	depth	outdoor	dynamic object	for tracking	objects	sequences	frames
Linemod [13]	real	×	✓	×	×	×	13	-	15K
Linemod Occlusion [1]	real	×	✓	×	×	×	8	-	1.2K
T-LESS [14]	real	×	✓	×	×	×	30	-	49K
HomebrewedDB [18]	real	×	✓	×	×	×	33	-	17K
TOD [27]	real	×	✓	×	×	×	20	-	64K
StereOBJ-1M [26]	real	×	×	✓	×	×	18	-	397K
YCB-Video [47]	real	✓	✓	×	×	✓	21	-	134K
OPT [45]	real	×	✓	×	×	✓	6	552	101K
RBOT [38]	semi-synthetic	✓	×	×	✓	✓	18	72	72K
BCOT (Ours)	real	✓	×	✓	✓	✓	20	404	126K

Table 1. Dataset Comparisons. Our BCOT benchmark is the only real scene benchmark that provides dynamic objects. The benchmark doesn't have invasive artificial markers and includes indoor and outdoor scenes.

ror and mis-alignment with RGB image, especially around the object boundary [47]. On the other hand, the datasets with artificial markers [1, 13, 14, 18], will not only damage the naturalness of the scene, but also limit the object movement. Therefore, a high-precision markerless pose estimation method is required in order to build a real-life 3D tracking benchmark, which according to our knowledge has not been addressed in previous works.

In this paper we propose a markerless multi-view approach to address the above problem. Our method can estimate the high-precision 3D pose of video objects from two orthogonal views of RGB images captured by the high-resolution, high-speed and synchronous cameras, as illustrated in Fig. 1. To deal with textureless objects, we adopted a shape-based method, which does not rely on point correspondences between views. The object pose is solved by jointly optimizing the shape re-projection error of multiple views, in a novel object-centered pose estimation framework. Based on the proposed approach we contribute a new 3D tracking benchmark, namely **BCOT (BinoCular Object Tracking) benchmark**, which contains accurately annotated real videos, with both camera and objects can move freely. The maximum annotation error of the BCOT benchmark is $2mm$, achieving the best annotation precision at present. The major contributions can be summarized as follows.

- We propose a multi-view approach that can estimate the accurate 3D pose of real video objects. Our method is markerless and suitable for textureless and moving objects, and thus provides a way to annotate real-life tracking videos.
- We build a 3D tracking benchmark of real scenes with high-precision ground truth (GT) poses, with annotation accuracy guaranteed by both theoretical analysis and validation experiments.
- We comprehensively evaluate the existing SOTA 3D object tracking methods on the proposed BCOT benchmark.

2. Related Work

Related datasets. Early 3D tracking algorithms usually use their own collected video sequences as test data [3, 31, 33, 37, 39, 42], which scale is small and difficult to reflect the performance of the algorithm.

In recent years, some large-scale 3D object tracking datasets have been released. The RBOT dataset [38] is a semi-synthetic dataset with the real scene background and the rendered object. It provides absolute GT poses but lacks authenticity. The motion trajectory of the virtual object is also pre-set, and all sequences share the same trajectory, limiting the diversity of object motion. The OPT dataset [45] is a real scene dataset, which uses artificial markers to calculate the GT pose. However, the marker occupies most background areas, making the background invasive.

For pose estimation datasets based on the single frame [1, 6, 8, 13, 14, 18, 26, 27, 47], because they do not annotate the pose on the sequence, or the annotation precision is insufficient, they cannot effectively evaluate the 3D tracking method.

Monocular textureless 3D object tracking. Textureless 3D object tracking can be divided into edge-based methods [2, 5, 9, 29, 34, 39, 40, 42, 46] and region-based methods [11, 33, 35–38, 49] according to the feature used. There are also some methods based on feature fusion, which use multiple features explicitly [23, 24, 48] or implicitly [15, 16, 42] to achieve better results.

In recent years, some methods based on deep learning have been explored [4, 7, 25, 43, 44], but their performance is still not comparable to those based on traditional features.

Multi-view tracking. Multi-view geometry [10] is widely used in computer vision, which estimates the camera and object pose through corresponding feature matching [13, 14, 17, 28]. However, the constraints will fail for textureless objects because the stable point or edge features cannot be extracted and matched when the object lacks the texture.

Li et al. [22] and Labbé et al. [20] propose a two-step

multi-view optimization framework for textureless objects. They first estimate the object pose using the image feature under every single camera and then minimize a reconstruction loss (irrelevant to image feature) in the world coordinate frame, where [20] uses 2D projection points and [22] uses 3D points. This kind of strategy separates the features and coordinate system, limiting the precision. Besides, they are single frame pose estimation methods whose accuracy is lower than the tracking method.

3. Multi-view Pose Estimation

Since the single-view RGB-based method can make use of only 2D reprojection error for 3D pose estimation, it is error-prone in the direction of the camera view (Z-axis), as shown in Fig. 2. In multi-view tracking, each camera has a different view direction, and the uncertainty thus can be greatly reduced.

3.1. The Object-centered Model

We should note that all previous 3D tracking methods are based on the camera coordinate frame to optimize the object pose, which can not directly establish the association between multiple cameras. So we first select a basic coordinate frame. Considering that all cameras are directly associated with objects, we use the object-centered coordinate frame $\boldsymbol{O}_o$ for pose optimization, which takes the center of the object model as the coordinate origin, as shown in Fig. 2. Please note that object template coordinate frame $\boldsymbol{O}_t$ is different from $\boldsymbol{O}_o$. The origin of $\boldsymbol{O}_t$ can be at any position, which is determined by the CAD model. At the same time, we need to know the relative position between the cameras. Here, we call the model based on the camera coordinate frame $\boldsymbol{O}_c$ the camera-centered model, and the model based on the object-centered coordinate frame $\boldsymbol{O}_o$ is called the object-centered model.

When using multi-view information, the coordinate frames between multiple cameras are no longer independent. So the pose cannot be estimated independently in each camera coordinate frame $\boldsymbol{O}_{c_i}$ but should be solved jointly based on the basic coordinate frame $\boldsymbol{O}_o$. The subscript i represents the camera index. We need to re-derive the camera projection model and pose updating process. The projection model in $\boldsymbol{O}_o$ is formulated as:

$$\boldsymbol{x} = \pi(\boldsymbol{K}({}^{c}\boldsymbol{T}_t\,\tilde{\boldsymbol{X}}_t)_{3\times1}) \tag{1}$$

$$= \pi(\boldsymbol{K}({}^{o}\boldsymbol{T}_c^{-1}\,{}^{o}\boldsymbol{T}_c\,{}^{c}\boldsymbol{T}_t\,\tilde{\boldsymbol{X}}_t)_{3\times1}) \tag{2}$$

$$= \pi(\boldsymbol{K}({}^{o}\boldsymbol{T}_c^{-1}\,\tilde{\boldsymbol{X}}_o)_{3\times1}). \tag{3}$$

$\boldsymbol{X}_m$ represents the 3D point of the object, and the subscript m represents the corresponding coordinate frame $\boldsymbol{O}_m$. $\tilde{\boldsymbol{X}} = (X, Y, Z, 1)^\top$ is homogeneous representation of the $\boldsymbol{X} = (X, Y, Z)^\top = (\tilde{\boldsymbol{X}})_{3\times1}$ and $\pi(\boldsymbol{X}) = [X/Z, Y/Z]^\top$.

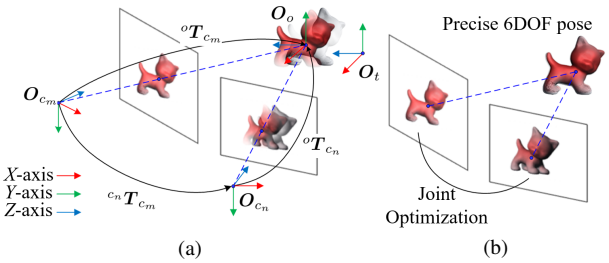


Figure 2. Error analysis and elimination of monocular 3D tracking. (a) When only using the left view for optimization, the predicted 3D position (in gray) is very different from that of the GT position (in red), even if there is a plausible visual result. This means there is a large translation error along the viewing direction, where the re-projection error can be observed in the right view. (b) Through the proposed joint optimization framework, we can obtain the precise 6DOF pose, which can render the exact contour on each image.

$\boldsymbol{x}$ is the 2D point on the image and $\boldsymbol{K}$ is the pre-calibrated camera intrinsic parameter. ${}^{n}\boldsymbol{T}_m$ represents the coordinate transformation from $\boldsymbol{O}_m$ to $\boldsymbol{O}_n$. In particular, ${}^{c}\boldsymbol{T}_t$ also represents the object pose in $\boldsymbol{O}_c$. ${}^{o}\boldsymbol{T}_c$ can be calculated from the $\boldsymbol{O}_t$, $\boldsymbol{O}_o$ and the initial pose (i.e., the pose of the previous frame). In practical applications, it can also select $\boldsymbol{O}_t$, $\boldsymbol{O}_{c_i}$, or other unified coordinate frames.

3.2. Joint Pose Estimation

The monocular energy function in $\boldsymbol{O}_o$ is formulated as:

$$\Delta\boldsymbol{\xi}_o = \arg\min_{\Delta\boldsymbol{\xi}_o} \sum\nolimits_{\boldsymbol{x}\in\Omega} F(\boldsymbol{x}, \boldsymbol{\xi}_c', {}^{o}\boldsymbol{T}_c), \tag{4}$$

and we then map the updated pose increment $\Delta\boldsymbol{\xi}_o$ to $\boldsymbol{O}_c$ by $\Delta\boldsymbol{T}_c = {}^{c}\boldsymbol{T}_o exp(\Delta\hat{\boldsymbol{\xi}}_o){}^{c}\boldsymbol{T}_o^{-1}$. $\boldsymbol{\xi}_c'$ is the initial pose in $\boldsymbol{O}_c$. F represents the arbitrary energy function for 3D tracking, here we adopt the region-based pose estimation method introduced in [38]. F in $\boldsymbol{O}_o$ can be re-formulated as:

$$F(\boldsymbol{x}, \boldsymbol{\xi}_c', {}^{o}\boldsymbol{T}_c) = F(\boldsymbol{x}, \boldsymbol{\xi}_o) \tag{5}$$

$$\begin{aligned} = -\log\big(&H_e(\Phi(\boldsymbol{x}(\boldsymbol{\xi}_o)))P_f(\boldsymbol{x}) \\ &+ (1 - H_e(\Phi(\boldsymbol{x}(\boldsymbol{\xi}_o))))P_b(\boldsymbol{x})\big), \end{aligned} \tag{6}$$

which minimizes the shape reprojection error with respect to a rendered shape template Φ and estimated soft object segmentation P_f, P_b. The original F in method [38] is associated with the pose $\boldsymbol{\xi}_c$ in $\boldsymbol{O}_c$, and we use the object-centered model to associate F with $\boldsymbol{\xi}_o$, and solve for the pose increment in $\boldsymbol{O}_o$. For the case of multiple views, we propose to optimize the following joint energy function:

$$\Delta\boldsymbol{\xi}_o = \arg\min_{\Delta\boldsymbol{\xi}_o} \sum\nolimits_{i=1}^{N} \sum\nolimits_{\boldsymbol{x}\in\Omega^i} F^i(\boldsymbol{x}, \boldsymbol{\xi}_{c_i}', {}^{o}\boldsymbol{T}_{c_i}) \tag{7}$$

$$= \arg\min_{\Delta\boldsymbol{\xi}_o} \sum\nolimits_{\boldsymbol{x}\in\Omega} F(\boldsymbol{x}, \boldsymbol{\xi}_o), \tag{8}$$

in which N represents the number of views. Ω^i represents optimized points set in each view. Through the object-centered model, we combine all the optimized points in each view to form an optimized area Ω, where all sample points are independent. Eq. 7 uses maximum likelihood estimation to maximize the color probability difference between the foreground and background to solve the pose. It becomes a summation by taking the logarithm. Therefore, all sampling points can be jointed, and Eq. 8 can constraint the pose by unifying the image features and coordinate system.

During tracking, the object can freely move. The camera movement needs to be discussed in two cases. The first case is that the relative spatial position of the cameras is fixed, and the transformation between the cameras can be calibrated in advance. The second case is that the cameras move freely. At this time, we need to calibrate the camera in real-time during the tracking. For example, put an artificial marker [19] in the scene or use SLAM [30] or other technologies, but this way will introduce the calibration error. For data collection we prefer the first approach, which allows the cameras to move with fixed relative pose (see section 4.3).

Optimization. We use the Gauss-Newton method to solve the joint energy function. Please refer to the supplementary materials for details.

4. BCOT Benchmark

Based on our high-precision multi-view tracking method, we construct the BCOT benchmark.

4.1. Data Collection

We utilize two high-resolution, high-speed cameras (MER-131-210U3) to capture images synchronously in the data collection stage, using Cam1 and Cam2 to indicate. The camera exposure time is $5ms$ ($200FPS$) so that no motion blur will occur. The image resolution is 1280×1024. When storing data, limited by the transmission bandwidth of the USB3.0 interface, the images of the two cameras can only be stored at a speed of $60FPS$, but the storing speed doesn't influence the exposure time. The included angle between the two cameras is approximately 90°. Their relative position in the scene is fixed, and we pre-calibrate them in advance. During collection, the camera set and the object can move freely in the scene.

We then use the binocular images taken by the Cam1 and Cam2 cameras to annotate the object pose. As shown in Fig. 1, the image taken by Cam1 is the complex scene designed in advance, and the image taken by Cam2 is the relatively clean background to improve the annotation precision. The pose annotation process does not need to consider the calculation efficiency. We increase the number of iterations in the optimization process to ensure convergence. The initial pose of the first frame can be roughly set manually. With the help of the spatial position relationship constraints between the cameras, it can be optimized to a precise value. We annotate the object pose frame by frame under the high frame rate ($60FPS$) image. At this time, the pose increment between frames is smaller, which can obtain higher precision. After annotating the entire sequence, we downsample the frames and resolutions, i.e., $30FPS$, 640×512 resolution. This strategy will annotate a more precise object pose.

Finally, the images with annotated object poses are provided to users. In this way, we can collect images of complex scenes and provide precise poses of objects without any invasion of artificial markers, ensuring the authenticity of the scene and the movement of the object.

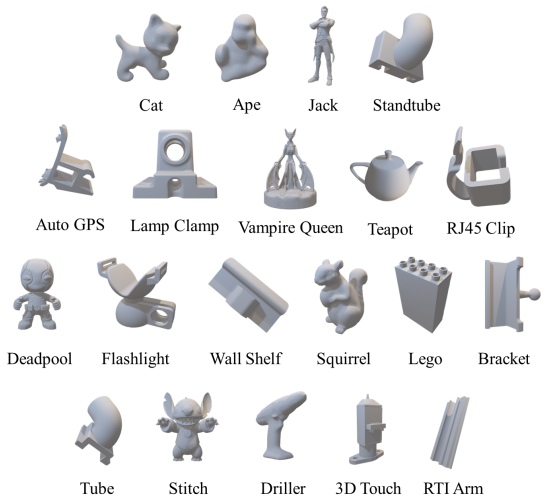
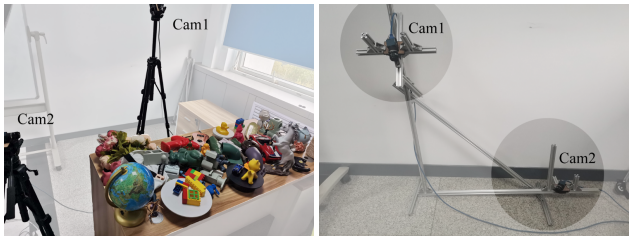


Figure 3. 3D Models in the BCOT Benchmark.



(a) (b)

Figure 4. Camera layout: (a) mounted on tripods; (b) mounted on a movable bracket.

4.2. 3D Models

The BCOT benchmark contains 20 objects, as shown in Fig. 3. The first row is the irregular object, and the second row is the hollow object. The last three rows are symmetrical objects. The real object is 3D printed according to the model and painted with one single color to ensure texture-

less. Besides, all objects have reflective properties. The range of the longest side of the model bounding box is from 91.7mm to 229.5mm.

4.3. Scenes

The BCOT benchmark contains a variety of scene attributes and multiple motion modes, which are partially combined to a total of 22 scenes.

Static camera set. The camera set in the scene are fixed, so its background is basically static, with only a few objects moving slowly on the turntable. Objects move freely in the scene. The camera layout is shown in Fig. 4(a).

Movable camera set. The camera set are fixed on a bracket, as shown in Fig. 4(b). The relative pose between the cameras is fixed, and the camera set can be moved freely. The other configurations are the same as the static camera set. When moving the bracket, the camera on it may shake very slightly. Therefore, we will slowly move the bracket to ensure annotating precision.

Indoor scenes. For the indoor scene, we only provide users with images taken by Cam1, where the background is delicately designed, as shown in Fig. 1. At the same time, we construct simple scenes and complex scenes, respectively.

Outdoor scenes. For the outdoor scene, the images taken by the two cameras have the same priority, so we provide the images taken by two cameras to the user.

Motion modes. The object motion is mainly divided into three modes. 1) Translation movement: tie the object to the toy car while manipulating the car to produce the free translation. 2) Suspension movement: use a transparent fishing line to tie the object to perform suspension movement. At the same time, move the apex to generate random motion and rotation. 3) Handheld movement: people hold the object to move freely.

Dynamic light. In indoor scenes, we added dynamic lighting sources to increase the complexity of the scene. In outdoor scenes, natural light at different times will naturally produce light changes.

Occlusion. In the suspension movement mode, we tie two objects at the same time to create mutual occlusion. Because occlusion will impact the precision of our multi-view tracking method, there are fewer occlusion sequences in the BCOT benchmark.

4.4. Data Post-processing

Some models may face considerable challenges in some specific scenes, e.g., reflective, symmetrical, and fast-rotating scenes, where our joint optimization framework cannot precisely track objects. If the reprojection error on the original resolution image is greater than 2 pixels, we will discard the sequence. In the BCOT benchmark, we provide 404 valid sequences.

Dataset	RGB-D datasets	TOD [27]	StereOBJ-1M [26]	BCOT
Error	$\geq 17mm$	$3.4mm$	$2.3mm$	$<$**2.0***mm*

Table 2. Annotation error in 3D space.

4.5. Why Use Binocular Data?

The critical factor determining our tracking precision is the included angle between the cameras (see section 5.1). The orthogonal angle can already constrain the object in space and eliminate the uncertainty of the pose. New uncertainties may be introduced to affect the precision if other cameras are added in between the orthogonal cameras.

If the cameras are arranged uniformly in the space under various viewing, objects can be constrained to the greatest extent. But it will also bring storage and camera movement restrictions, and camera synchronization will also introduce extra errors. Therefore, to balance precision and operability, we adopted two cameras.

4.6. Error Analysis

The error of pose annotation mainly comes from three aspects: calibration error between cameras, camera synchronization, and estimation error of the proposed multi-view tracking method. The calibration between the cameras is an offline process, so it can be considered precise. We then use the system clock to ensure the synchronization between cameras.

We comprehensively analyze the errors caused by camera synchronization and pose estimation. In the binocular data, we analyze the annotation error by observing the reprojection error of the object contour. Cameras with 90° included angles can constrain objects from 3 directions, one camera constrains the X and Y axes, and the other camera constrains the Z-axis. Ideally, the projection contour under the annotation pose can be precisely aligned with the object contour in each image.

The reprojection pixel error can be observed from the binocular data. We then use the camera projection model in Eq. 1 to convert to spatial position error. In the formula, ${}^{c}\boldsymbol{T}_{t}$ is the annotated object pose, which converts the object template coordinate frame $\boldsymbol{O}_{t}$ to the camera coordinate frame $\boldsymbol{O}_{c}$. ${}^{c}\boldsymbol{T}_{t}$ is the rigid transformation matrix, and $\tilde{\boldsymbol{X}}_{t}$ is the model vertices, so the error of $\tilde{\boldsymbol{X}}_{c}$ in the camera coordinate frame is equivalent to the annotation error ${}^{c}\boldsymbol{T}_{t}$. We then expand the formula to get the mapping relationship:

$$x = \frac{f_x}{Z_c} X_c + c_x, \quad y = \frac{f_y}{Z_c} Y_c + c_y. \tag{9}$$

c_x and c_y are constants, so X_c and Y_c are proportional to x and y, respectively. In the original data of our benchmark, $\frac{f_x}{Z_c}$ and $\frac{f_y}{Z_c}$ are both in $[1, 2]$, which means each pixel error corresponds to a spatial 3D error of $0.5-1mm$. In

Included Angle	Mono.	5°	10°	20°	30°	45°	60°	90°	120°	5°	10°	30°	45°
Camera Index	C-0	C-0/C-1	C-0/C-2	C-0/C-3	C-0/C-4	C-0/C-5	C-0/C-6	C-0/C-7	C-0/C-8	C-0/C-9	C-0/C-10	C-0/C-11	C-0/C-12
r(°)	1.62	1.31	1.22	1.17	1.07	0.94	0.87	0.76	**0.62**	1.33	1.27	1.06	0.82
tx(mm)	4.36	3.27	2.12	1.12	0.80	0.57	0.41	**0.28**	0.31	3.18	2.01	0.53	0.40
ty(mm)	2.39	1.80	1.15	0.55	0.34	0.28	0.26	0.23	**0.22**	1.82	1.19	0.45	0.38
tz(mm)	22.09	16.67	10.64	5.11	2.86	1.35	0.67	**0.28**	0.37	16.30	10.48	1.72	0.93
Lost Number	21	15	10	4	1	0	0	0	0	18	11	0	0

Table 3. Binocular tracking evaluation on *Object moves freely with fixed cameras* mode.

all sequences, we ensure that the pixel errors in two views are both within 2 pixels, so the maximum spatial error of the proposed benchmark is $2mm$, which is the benchmark with the highest annotation precision at present. Table 2 shows the annotation error compared to the other datasets. The RGB-D sensor has a random error standard deviation of $17mm$[1], and the keypoints-based annotation methods [27] and [26] have the RMSE (Root Mean Squared Error) of $3.4mm$ and $2.3mm$.

4.7. Evaluation Metric

We use $n°$, $n\ cm$ and ADD metric to evaluate the monocular tracking method.

n°, **n** cm. The tracking is considered accurate when the rotation error is less than $n°$ and the translation error is less than $n\ cm$ [38]. This value is usually set to 5°, $5cm$. If it is greater than this pre-set value, the initial pose will be reset to the GT pose. For indoor objects, $5cm$ is usually a large threshold, so we will adjust the value of n to re-evaluate the monocular 3D tracking method. Given the GT translation $\boldsymbol{t}$ and rotation $\boldsymbol{R}$, and the predicted translation $\hat{\boldsymbol{t}}$ and rotation $\hat{\boldsymbol{R}}$, the translation error and rotation error are defined as:

$$e(\boldsymbol{t}) = \|\hat{\boldsymbol{t}} - \boldsymbol{t}\|_2, \tag{10}$$

$$e(\boldsymbol{R}) = \cos^{-1}\left(\frac{1}{2}(\text{trace}\left(\hat{\boldsymbol{R}}^{\top}\boldsymbol{R}\right) - 1)\right). \tag{11}$$

ADD metric. ADD metric [12] represents the average distance between the model points in the predicted pose and the GT pose. When the error is less than the preset value, the tracking is considered correct. Although our BCOT benchmark contains many symmetrical objects, we have not used the ADD-S metric for symmetrical objects because we have the pose of the previous frame as the prior information. The ADD metric is formulated as:

$$ADD = \frac{1}{M}\sum_{i=1}^{M}\|(\hat{\boldsymbol{R}}\boldsymbol{X} + \hat{\boldsymbol{t}}) - (\boldsymbol{R}\boldsymbol{X} + \boldsymbol{t})\|. \tag{12}$$

5. Experiments

In this section, we conduct a detailed evaluation of the proposed joint optimization framework and the BCOT benchmark. We also show more results in the supplementary material. Our experimental environment is on a laptop with Intel(R) Core(TM) i7-8565U @1.8GHz CPU, NVIDIA GeForce MX250 GPU, and 8GB RAM.

5.1. Multi-View Tracking Evaluation

The precision of the proposed joint optimization framework is the basis for the construction of the BCOT benchmark. This section uses the synthetic data to prove that the method can obtain sufficient annotation precision. The synthetic data contains 4 objects, namely Cat, $Clown$, $Driller$ and $Squirrel$, and there diameters are $127.6mm$, $142.3mm$, $229.5mm$, and $194.3mm$. Each object includes 3 modes of multi-view data, i.e.: 1) *Object moves freely with fixed cameras*, 2) *Object rotates only with fixed cameras* and 3) *Cameras move freely*. Its resolution is 640×480px.

Binocular tracking result. We first perform the binocular tracking evaluation on the mode 1, as shown in Table 3. The basic camera C-0 and other cameras constitute binocular data. C-0 to C-8 are on an arc, forming a $plane$ with the object. C-9 to C-12 are outside the plane, and C-0/C-1/C-9, C-0/C-2/C-10, C-0/C-3/C-11 and C-0/C-4/C-12 constitute 4 sets of $cone$-type cameras with the object. The data in the table is the average error of the two views, which also averaged all 4 objects. Lost Number represents the number of tracking failure, i.e., the rotation error is larger than 5°, or the translation error is larger than $5cm$, under the C-0 coordinate frame (first camera of the set). When the object is lost, we reset the GT pose.

The second column of Table 3 is the tracking result under C-0 with only monocular data. Overall, the rotation and translation errors of our binocular tracking gradually decrease with the included angle increasing between 5° to 90°, and this tendency is especially obvious in the Z-axis direction. When the included angle is 90°, the Z-axis translation error is $0.28mm$, which is less than 2‰ of the diameter of the model.

The object may be lost in the case that the camera included angle is small. This is because the uncertainty of the pose cannot be eliminated in a small view angle. As the included angle increases, while the object may be lost in one view, the other view can constrain the object so that the joint optimization will pull the object back to the correct pose.

Multi-view tracking result. We further introduce the

[1]Azure Kinect DK hardware specifications: https://docs.microsoft.com/en-us/azure/kinect-dk/hardware-specification

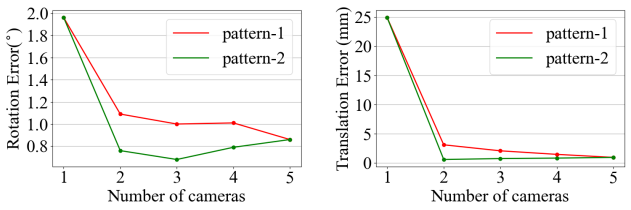


Figure 5. Multi-view tracking results. With the increase in the number of cameras, the error trend of two patterns.

Included Angle	Mono.	Freely	Freely	Freely
Camera Index	C-0	C-0/C-1	C-0/C-2	C-0/C-1/C-2
r(°)	1.47	0.59	0.80	**0.50**
tx(mm)	1.18	0.24	0.26	**0.21**
ty(mm)	1.22	0.26	0.18	**0.16**
tz(mm)	12.96	0.78	0.33	**0.32**
Lost Number	17	0	0	0

Table 4. Multi-view tracking evaluation on *Cameras move freely* mode.

Reso.(width)	320	640	1280	1920	2560	4096
r(°)	1.01	0.68	0.40	0.37	**0.34**	0.36
tx(mm)	0.28	0.21	0.13	0.10	0.07	**0.06**
ty(mm)	0.40	0.21	0.11	0.08	0.05	**0.03**
tz(mm)	0.28	0.20	0.12	0.10	0.08	**0.06**
Lost Number	0	0	0	0	0	0

Table 5. Binocular tracking evaluation on different resolution.

error trend when the camera number is gradually increasing. We take C-0 as the basic camera and append cameras according to the following two patterns, i.e., the camera included angle is appended from small to large (pattern-1, C-0/C-4/C-5/C-6/C-7), and the camera included angle is appended from large to small (pattern-2, C-0/C-7/C-5/C-6/C-4). The error trends of translation and rotation are shown in Fig. 5.

We can see that translation and rotation errors of pattern-1 show a downward trend. But in pattern-2, adding a camera between the existing cameras may increase the error, especially the translation component. Moreover, the multi-view performance in most cases is inferior to the binocular tracking with a large included angle, i.e., the green point of 2 cameras. All of these show that the camera included angle has a decisive effect on the tracking precision. The results of mode 2 are consistent with the results of mode 1. The other results can be found in the supplementary material.

Cameras move freely. We next evaluate the *Cameras move freely* mode. The relative transformation between the cameras is continuously changing, and we use the ground truth poses of each camera to calculate it. The average error is shown in Table 4.

We can see that when the two cameras are moving freely, our method can track precisely. The precision is slightly improved by extending to 3 cameras. This is because some small included angle situations during the tracking may increase the error, and appending cameras with larger included angle can reduce this kind of error. Overall, the free movement of the two cameras can already satisfy multi-view tracking in general scenes.

Evaluation with different resolutions. Table 5 shows the evaluation of the precision with different resolutions under binocular tracking. We use Cat model for the evaluation, whose error is lower than the average error due to its unique structure. The camera included angle here is 90°, and the resolution gradually increases from 320×240px to 4096×3072px. As the resolution increases, the precision of rotation and translation also gradually increases. When the resolution is 2560px, each axis's translation error is less than $0.1mm$, reaching sub-millimeter level.

5.2. Evaluation on BCOT Benchmark

BCOT benchmark examples. Fig. 6 shows examples of the BCOT benchmark. The red contour is rendered according to the annotation pose, and it can accurately align with the object contour on the image.

Monocular 3D tracking method evaluation. Table 6 shows the evaluation results of $n°$, $n\ cm$ and ADD metric. The d in the ADD metric represents the longest side of the bounding box of the object model. For all the metrics in the table, if the rotation error is larger than 5° or the translation error is larger than $5cm$, we reset the GT pose. All methods use the code provided by the author. ACCV2020 [35] achieves the highest accuracy under 5°, $5cm$. However, it needs to pre-render the template when meeting the new object, which will cost several minutes. Considering 5°, $5cm$ is a relatively relaxed metric for tracking, we further test the accuracy under 2°, $2cm$. We find that TVCG2021 [15] has higher rotation accuracy, and ACCV2020 [35] has higher translation accuracy.

Fig. 7 shows the tracking accuracy under different ADD error tolerances. The unit of the horizontal axis is the model side length d. A more detailed comparative comparison can be found in the supplementary material.

5.3. Limitation and Future Work

Although our multi-view joint optimization framework can achieve sufficient precision for benchmark construction, it still has some limitations. We need to keep the relationship between cameras relatively fixed for multi-view tracking, which limits many application scenarios. In future work, we will explore high-precision optimization methods with freely moving cameras. For the BCOT benchmark, to ensure precision, the movement speed of the object is relatively slow. In addition, there is currently no proper method to evaluate the rotation error of the annotation, and we will explore it in future work.

Figure 6. Examples of the BCOT benchmark, where the red contour is rendered according to the annotation pose: (a) Vampire Queen model, static camera set, easy scene, handheld movement; (b) Flashlight model, static camera set, complex scene, dynamic light, translation movement; (c) Bracket model, movable camera set, complex scene, suspension movement; (d) Deadpool model, movable camera set, complex scene, occlusion, suspension movement; (e) Standtube model, movable camera set, outdoor scene, handheld movement, providing both two views.

Method	ADD−0.02d	ADD−0.05d	ADD−0.1d	5°, 5cm	5°	5cm	2°, 2cm	2°	2cm	Time(ms)
MTAP2019 [40]	5.5	32.7	64.6	54.4	54.9	97.8	12.4	13.7	77.9	8.8
TPAMI2019 [38]	11.7	31.6	57.1	77.1	79.2	91.7	40.8	48.3	67.8	34.6
CGF2020 [16]	12.0	31.3	57.5	84.1	85.1	95.7	45.1	55.1	70.1	33.0
ACCV2020 [35]	10.9	**45.5**	**76.9**	**89.0**	**89.3**	**99.5**	46.0	49.5	**87.8**	**3.5**
C&G2021 [23]	9.1	31.5	58.1	82.5	84.7	95.0	38.5	47.0	69.9	18.9
JCST2021 [24]	14.4	38.1	65.7	87.0	88.1	97.2	50.2	57.3	77.2	38.5
TVCG2021 [15]	**15.6**	39.8	66.1	87.1	88.5	96.3	**51.4**	**59.0**	76.4	34.8

Table 6. Comparison of monocular 3D tracking methods.

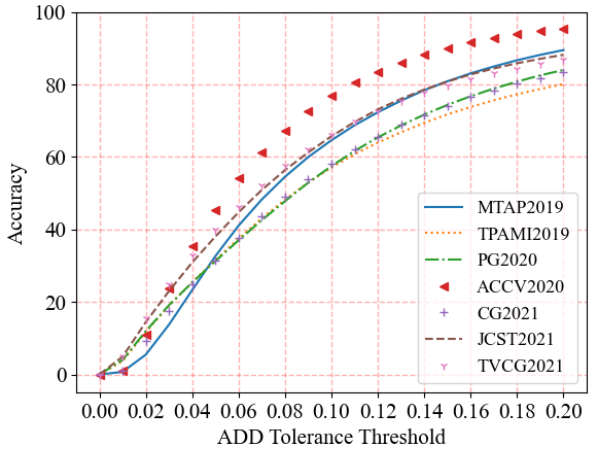


Figure 7. Overall tracking accuracy under various ADD error tolerance thresholds.

6. Conclusion

We have proposed a joint optimization framework for multi-view textureless 3D object tracking, based on which we further construct a real scene benchmark with high-precision annotation poses. Increasing the included angle of cameras to eliminate the pose uncertainty is the key to improving precision. The orthometric two cameras have reached enough precision, and increasing the number of cameras will not significantly improve tracking precision anymore. We comprehensively evaluated SOTA 3D object tracking methods on the proposed BCOT benchmark. At the same time, the benchmark provides real data for deep learning model training, which enables future research on tracking methods based on deep learning.

Acknowledgements: This work is partially supported by the National Key R&D Program of China (No. 2020YFB1708903), Zhejiang Lab (No. 2020NB0AB02), and NSF of China (No. 62172260).

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV (2)*, volume 8690 of *Lecture Notes in Computer Science*, pages 536–551. Springer, 2014. 1, 2

[2] Changhyun Choi and Henrik I. Christensen. Robust 3D visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. *I. J. Robotics Res.*, 31(4):498–519, 2012. 2

[3] Alberto Crivellaro and Vincent Lepetit. Robust 3D tracking with descriptor fields. In *CVPR*, pages 3414–3421. IEEE Computer Society, 2014. 2

[4] Alberto Crivellaro, Mahdi Rad, Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. Robust 3D object tracking from monocular images using stable parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1465–1479, 2018. 2

[5] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):932–946, 2002. 2

[6] Haoshu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *CVPR*, pages 11441–11450. Computer Vision Foundation / IEEE, 2020. 2

[7] Mathieu Garon and Jean-François Lalonde. Deep 6-DOF tracking. *IEEE Trans. Vis. Comput. Graph.*, 23(11):2410–2418, 2017. 2

[8] Till Grenzdörffer, Martin Günther, and Joachim Hertzberg. YCB-M: A multi-camera RGB-D dataset for object recognition and 6DoF pose estimation. In *ICRA*, pages 3650–3656. IEEE, 2020. 2

[9] Chris Harris and Carl Stennett. RAPID - a video rate object tracker. In *BMVC*, pages 1–6. BMVA Press, 1990. 2

[10] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 2

[11] Jonathan Hexner and Rami R. Hagege. 2D-3D pose estimation of heterogeneous objects using a region based approach. *Int. J. Comput. Vis.*, 118(1):95–112, 2016. 2

[12] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, pages 858–865. IEEE Computer Society, 2011. 6

[13] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV (1)*, volume 7724 of *Lecture Notes in Computer Science*, pages 548–562. Springer, 2012. 1, 2

[14] Tomas Hodan, Pavel Haluza, Stepán Obdrzálek, Jiri Matas, Manolis I. A. Lourakis, and Xenophon Zabulis. T-LESS: an RGB-D dataset for 6D pose estimation of texture-less objects. In *WACV*, pages 880–888. IEEE Computer Society, 2017. 1, 2

[15] Hong Huang, Fan Zhong, and Xueying Qin. Pixel-wise weighted region-based 3D object tracking using contour constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 2, 7, 8

[16] Hong Huang, Fan Zhong, Yuqing Sun, and Xueying Qin. An occlusion-aware edge-based method for monocular 3D object tracking using edge confidence. *Comput. Graph. Forum*, 39(7):399–409, 2020. 2, 8

[17] Huan Jin, Gang Qian, and Stjepan Rajko. Real-time multi-view 3D object tracking in cluttered scenes. In *ISVC (2)*, volume 4292 of *Lecture Notes in Computer Science*, pages 647–656. Springer, 2006. 2

[18] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: RGB-D dataset for 6D pose estimation of 3D objects. In *ICCV Workshops*, pages 2767–2776. IEEE, 2019. 1, 2

[19] Hirokazu Kato and Mark Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IWAR*, pages 85–94. IEEE Computer Society, 1999. 4

[20] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. *ECCV*, 2020. 2, 3

[21] Vincent Lepetit and Pascal Fua. Monocular model-based 3D tracking of rigid objects: A survey. *Found. Trends Comput. Graph. Vis.*, 1(1), 2005. 1

[22] Chi Li, Jin Bai, and Gregory D. Hager. A unified framework for multi-view multi-class object pose estimation. In *ECCV (16)*, volume 11220 of *Lecture Notes in Computer Science*, pages 263–281. Springer, 2018. 2, 3

[23] Jiachen Li, Xiuqiang Song, Fan Zhong, and Xueying Qin. Fast 3D texture-less object tracking with geometric contour and local region. *Comput. Graph.*, 97:225–235, 2021. 2, 8

[24] Jia-Chen Li, Fan Zhong, Songhua Xu, and Xueying Qin. 3D object tracking with adaptively weighted local bundles. *J. Comput. Sci. Technol.*, 36(3):555–571, 2021. 2, 8

[25] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. *Int. J. Comput. Vis.*, 128(3):657–678, 2020. 2

[26] Xingyu Liu, Shun Iwase, and Kris M. Kitani. StereOBJ-1M: Large-scale stereo image dataset for 6D object pose estimation. *CoRR*, abs/2109.10115, 2021. 2, 5, 6

[27] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. KeyPose: Multi-view 3D labeling and keypoint estimation for transparent objects. In *CVPR*, pages 11599–11607. Computer Vision Foundation / IEEE, 2020. 2, 5, 6

[28] Manolis I. A. Lourakis and Xenophon Zabulis. Model-based pose estimation for rigid objects. In *ICVS*, volume 7963 of *Lecture Notes in Computer Science*, pages 83–92. Springer, 2013. 2

[29] Éric Marchand, Patrick Bouthemy, and François Chaumette. A 2D-3D model-based approach to real-time visual tracking. *Image Vis. Comput.*, 19(13):941–955, 2001. 2

[30] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. 4

[31] Karl Pauwels, Leonardo Rubio, Javier Díaz, and Eduardo Ros. Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. In *CVPR*, pages 2347–2354. IEEE Computer Society, 2013. 2

[32] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6DoF pose estimation. In *CVPR*, pages 4561–4570. Computer Vision Foundation / IEEE, 2019. 1

[33] Victor Adrian Prisacariu and Ian D. Reid. PWP3D: real-time segmentation and tracking of 3D objects. *Int. J. Comput. Vis.*, 98(3):335–354, 2012. 2

[34] Byung-Kuk Seo, Hanhoon Park, Jong-Il Park, Stefan Hinterstoisser, and Slobodan Ilic. Optimal local searching for fast and robust textureless 3D object tracking in highly cluttered backgrounds. *IEEE Trans. Vis. Comput. Graph.*, 20(1):99–110, 2014. 2

[35] Manuel Stoiber, Martin Pfanne, Klaus H. Strobl, Rudolph Triebel, and Alin Albu-Schäffer. A sparse gaussian approach to region-based 6DoF object tracking. In *ACCV (2)*, volume 12623 of *Lecture Notes in Computer Science*, pages 666–682. Springer, 2020. 2, 7, 8

[36] Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer. Real-time monocular segmentation and pose tracking of multiple objects. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 423–438. Springer, 2016. 2

[37] Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In *ICCV*, pages 124–132. IEEE Computer Society, 2017. 2

[38] Henning Tjaden, Ulrich Schwanecke, Elmar Schömer, and Daniel Cremers. A region-based gauss-newton approach to real-time monocular multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1797–1812, 2019. 1, 2, 3, 6, 8

[39] Bin Wang, Fan Zhong, and Xueying Qin. Pose optimization in edge distance field for textureless 3D object tracking. In *CGI*, pages 32:1–32:6. ACM, 2017. 2

[40] Bin Wang, Fan Zhong, and Xueying Qin. Robust edge-based 3D object tracking with direction-based pose validation. *Multim. Tools Appl.*, 78(9):12307–12331, 2019. 2, 8

[41] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *CVPR*, pages 16611–16621. Computer Vision Foundation / IEEE, 2021. 1

[42] Guofeng Wang, Bin Wang, Fan Zhong, Xueying Qin, and Baoquan Chen. Global optimal searching for textureless 3D object tracking. *Vis. Comput.*, 31(6-8):979–988, 2015. 2

[43] Bowen Wen and Kostas E. Bekris. BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models. *CoRR*, abs/2108.00516, 2021. 1, 2

[44] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E. Bekris. se(3)-TrackNet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains. In *IROS*, pages 10367–10373. IEEE, 2020. 1, 2

[45] Po-Chen Wu, Yueh-Ying Lee, Hung-Yu Tseng, Hsuan-I Ho, Ming-Hsuan Yang, and Shao-Yi Chien. A benchmark dataset for 6DoF object pose tracking. In *ISMAR Adjunct*, pages 186–191. IEEE Computer Society, 2017. 1, 2

[46] Harald Wuest, Florent Vial, and Didier Stricker. Adaptive line tracking with multiple hypotheses for augmented reality. In *Fourth IEEE / ACM International Symposium on Mixed and Augmented Reality (ISMAR 2005), 5-8 October 2005, Vienna, Austria*, pages 62–69. IEEE Computer Society, 2005. 2

[47] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018. 1, 2

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

[49] Leisheng Zhong, Xiaolin Zhao, Yu Zhang, Shunli Zhang, and Li Zhang. Occlusion-aware region-based 3D pose tracking of objects with temporally consistent polar-based local partitioning. *IEEE Trans. Image Process.*, 29:5065–5078, 2020. 2