Supplementary Material of
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

Appendix

The appendix is organized as follows. Sec. A gives a detailed derivation of the optimization of the object-centered model and the joint framework. Sec. B presents additional test results in our synthetic multi-view dataset and shows more intermediate results. Sec. C shows more details and examples of the proposed BCOT benchmark and evaluates state-of-the-art monocular 3D tracking methods comprehensively. Finally, Sec. D compares BCOT with other 3D object tracking datasets in detail.

A. Optimization

In this section, we first derive the Jacobian matrix in detail based on the object-centered model, based on which we solve the pose for the proposed joint optimization framework.

A.1. Optimization of the Object-centered Model

As shown in the manuscript, we translate the camera-centered model to the object-centered model and then expand it, i.e.:

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

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

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

$$= \pi\left(\boldsymbol{K}\left(\begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix}\right)_{3\times1}\right) \tag{4}$$

$$= \pi\left(\boldsymbol{K}\left(\begin{bmatrix} t_{11}X_o+t_{12}Y_o+t_{13}Z_o+t_{14} \\ t_{21}X_o+t_{22}Y_o+t_{23}Z_o+t_{24} \\ t_{31}X_o+t_{32}Y_o+t_{33}Z_o+t_{34} \\ 1 \end{bmatrix}\right)_{3\times1}\right) \tag{5}$$

$$= \pi\left(\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} A \\ B \\ C \end{bmatrix}\right) \tag{6}$$

$$= \begin{bmatrix} \frac{f_xA+c_xC}{C} \\ \frac{f_yB+c_yC}{C} \end{bmatrix}, \tag{7}$$

where $A = t_{11}X_o + t_{12}Y_o + t_{13}Z_o + t_{14}$, $B = t_{21}X_o + t_{22}Y_o + t_{23}Z_o + t_{24}$, and $C = t_{31}X_o + t_{32}Y_o + t_{33}Z_o + t_{34}$.

The object-centered model and the camera-centered model are obtained from the camera projection model, which is irrelevant to the feature extraction of the tracking method. Therefore, mapping the camera-centered model to the object-centered model is universal and can be replaced in all 3D tracking methods.

The Jacobian matrix under $\boldsymbol{O}_o$ can be formulated as:

$$\boldsymbol{J}_o(\boldsymbol{x}) = \frac{\partial F}{\partial \Delta\boldsymbol{\xi}_o} = \frac{\partial F}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_o}\frac{\partial \boldsymbol{X}_o}{\partial \Delta\boldsymbol{\xi}_o}. \tag{8}$$

$\frac{\partial F}{\partial \boldsymbol{x}}$ is related to the energy function, which is unique to different methods and irrelevant to the coordinate frame. $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_o}\frac{\partial \boldsymbol{X}_o}{\partial \Delta\boldsymbol{\xi}_o}$ is derived based on the object-centered model, which is common to each method, replacing $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_c}\frac{\partial \boldsymbol{X}_c}{\partial \Delta\boldsymbol{\xi}_c}$ based on the camera-centered model.

Then we derivative $\boldsymbol{x}$ as:

$$\frac{\partial \boldsymbol{x}}{\partial \Delta\boldsymbol{\xi}_o} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_o}\frac{\partial \boldsymbol{X}_o}{\partial \Delta\boldsymbol{\xi}_o}, \tag{9}$$

where

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_o} = \begin{bmatrix} \frac{\partial x}{\partial X_o} & \frac{\partial x}{\partial Y_o} & \frac{\partial x}{\partial Z_o} \\ \frac{\partial y}{\partial X_o} & \frac{\partial y}{\partial Y_o} & \frac{\partial y}{\partial Z_o} \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} \frac{(f_x\frac{\partial A}{\partial X_o}+c_x\frac{\partial C}{\partial X_o})C-(f_xA+c_xC)\frac{\partial C}{\partial X_o}}{C^2} & \frac{\partial x}{\partial Y_o} & \frac{\partial x}{\partial Z_o} \\ \frac{(f_y\frac{\partial B}{\partial X_o}+c_y\frac{\partial C}{\partial X_o})C-(f_yB+c_yC)\frac{\partial C}{\partial X_o}}{C^2} & \frac{\partial y}{\partial Y_o} & \frac{\partial y}{\partial Z_o} \end{bmatrix} \tag{11}$$

and

$$\frac{\partial \boldsymbol{X}_o}{\partial \Delta\boldsymbol{\xi}_o} = \frac{\partial exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X}}{\partial \Delta\boldsymbol{\xi}_o} \tag{12}$$

$$= \lim_{\Delta\boldsymbol{\xi}_o\to\boldsymbol{0}} \frac{exp(\Delta\hat{\boldsymbol{\xi}}_o)exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X} - \exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X}}{\Delta\boldsymbol{\xi}_o} \tag{13}$$

$$= \lim_{\Delta\boldsymbol{\xi}_o\to\boldsymbol{0}} \frac{(\boldsymbol{I} + \Delta\hat{\boldsymbol{\xi}}_o)exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X} - \exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X}}{\Delta\boldsymbol{\xi}_o} \tag{14}$$

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

$$= \lim_{\Delta\boldsymbol{\xi}_o\to\mathbf{0}} \frac{\Delta\hat{\boldsymbol{\xi}}_o exp(\hat{\boldsymbol{\xi}}_o)\boldsymbol{X}}{\Delta\boldsymbol{\xi}_o} \tag{15}$$

$$= \lim_{\Delta\boldsymbol{\xi}_o\to\mathbf{0}} \frac{\begin{bmatrix} \Delta\hat{\boldsymbol{\phi}}_o & \Delta\boldsymbol{\rho}_o \\ \mathbf{0}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{RX}+\boldsymbol{t} \\ 1 \end{bmatrix}}{\Delta\boldsymbol{\xi}_o} \tag{16}$$

$$= \lim_{\Delta\boldsymbol{\xi}_o\to\mathbf{0}} \frac{\begin{bmatrix} \Delta\hat{\boldsymbol{\phi}}_o(\boldsymbol{RX}+\boldsymbol{t})+\Delta\boldsymbol{\rho}_o \\ 0 \end{bmatrix}}{\Delta\boldsymbol{\xi}_o} \tag{17}$$

$$= \begin{bmatrix} \boldsymbol{I} & -(\boldsymbol{RX}+\boldsymbol{t})^\wedge \\ \mathbf{0}^\top & \mathbf{0}^\top \end{bmatrix} \tag{18}$$

$$= [\boldsymbol{I} \quad -\hat{\boldsymbol{X}}_o]. \tag{19}$$

$\boldsymbol{\phi}_o$ and $\boldsymbol{\rho}_o$ are the rotation and translation components of $\boldsymbol{\xi}_o$, and $\boldsymbol{R}$ and $\boldsymbol{t}$ are the corresponding rotation matrix and translation vector of $\boldsymbol{T}$.

A.2. Optimization of the Joint Framework

For the i-th camera, we can calculate the Jacobian matrix of the object-centered model as described above, i.e.:

$$\boldsymbol{J}_o^i(\boldsymbol{x}) = \frac{\partial F^i}{\partial \Delta\boldsymbol{\xi}_o} = \frac{\partial F^i}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}_o} \frac{\partial \boldsymbol{X}_o}{\partial \Delta\boldsymbol{\xi}_o}. \tag{20}$$

We use the Gauss-Newton method for optimization, where the second-order Taylor approximation of the energy function E is formulated as:

$$\begin{aligned} E(\boldsymbol{\xi}_o{+}\Delta\boldsymbol{\xi}_o) \approx E(\boldsymbol{\xi}_o) + \sum\nolimits_{i=1}^{N} \sum\nolimits_{\boldsymbol{x}\in\Omega^i} \boldsymbol{J}_o^i(\boldsymbol{x})\Delta\boldsymbol{\xi}_o \\ + \frac{1}{2} \sum\nolimits_{i=1}^{N} \sum\nolimits_{\boldsymbol{x}\in\Omega^i} \Delta\boldsymbol{\xi}_o^\top \boldsymbol{J}_o^{i\top}(\boldsymbol{x}) \boldsymbol{J}_o^i(\boldsymbol{x})\Delta\boldsymbol{\xi}_o. \end{aligned} \tag{21}$$

In Eq. 21, the second-order derivative is dropped when calculating the Hessian matrix. Then the update step in the object-centered model can be formulated as:

$$\begin{aligned} \Delta\boldsymbol{\xi}_o = -\Big(\sum\nolimits_{i=1}^{N} \sum\nolimits_{\boldsymbol{x}\in\Omega^i} \boldsymbol{J}_o^{i\top}(\boldsymbol{x}) \boldsymbol{J}_o^i(\boldsymbol{x})\Big)^{-1} \\ \cdot \sum\nolimits_{i=1}^{N} \sum\nolimits_{\boldsymbol{x}\in\Omega^i} \boldsymbol{J}_o^{i\top}(\boldsymbol{x}). \end{aligned} \tag{22}$$

Finally, we map $\Delta\boldsymbol{\xi}_o$ to each camera coordinate frame, i.e.:

$$\Delta\boldsymbol{T}^i = {}^{c_i}\boldsymbol{T}_o exp(\Delta\hat{\boldsymbol{\xi}}_o)({}^{c_i}\boldsymbol{T}_o)^{-1}. \tag{23}$$

B. Additional Multi-view Tracking Results on the Synthetic Data

The synthetic dataset contains three modes of multi-view data, including: *1) Object moves freely with fixed cameras*, *2) Object rotates only with fixed cameras* and *3) Cameras move freely*.

In the manuscript, we give the results of binocular tracking and multi-view tracking results in mode 1. In this section, we first provide the rest results in mode1 and mode 2 and then give some intermediate results to show the effectiveness of our method. Fig. 1 shows the spatial distribution of cameras. The red cameras constitute the $plane$-type, and the green cameras with the other two cameras on the plane constitute the $cone$-type.

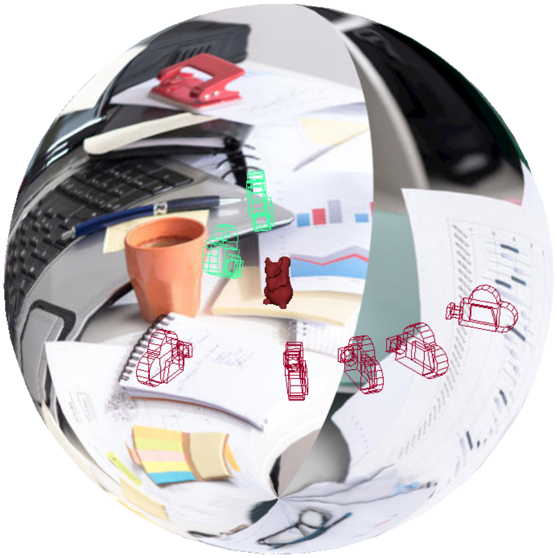


Figure 1. Spatial distribution of cameras in first two modes. The red cameras and the object constitute a $plane$, while the green cameras are outside the plane, constituting the $cone$ with the other two cameras on the plane.

B.1. Trinocular Tracking Result in Mode 1

We select several groups of cameras for the trinocular tracking evaluation in mode 1. The selection principle is that the included angle between the first camera and the second camera is equal to the included angle between the second camera and the third camera. Based on this, we select 8 sets of data where the first four are the object and the cameras constitute a $plane$, and the last four are the object and the cameras constitute a $cone$. The camera angles are 5°, 10°, 30°, and 45°, respectively. Table 1 shows the evaluation results.

Overall, the rotation and translation errors decrease with the camera angle increase in both $plane$ and $cone$ cameras. We can find more interesting consequences by combining Table 3 in the manuscript and Table 1, that is, 1) when the object and the cameras on one $plane$, the precision depends on the two cameras with the largest included angle, and adding cameras between them will reduce the overall precision. 2) The precision of the $cone$ cameras is worse than that of the $plane$ cameras if the included angle is the same.

The reason for this phenomenon is that the cameras with a large included angle can eliminate uncertainty, but adding a camera between them actually introduces the new uncer-

Camera Angle	Mono.	5° *plane*	10° *plane*	30° *plane*	45° *plane*	5° *cone*	10° *cone*	30° *cone*	45° *cone*
Camera Index	C-0	C-0/C-1/C2	C-0/C-2/C3	C-0/C-4/C6	C-0/C-5/C-7	C-0/C-1/C9	C-0/C-2/C10	C-0/C-4/C11	C-0/C-5/C-12
r(°)	1.62	1.25	1.21	0.93	**0.68**	1.28	1.24	0.96	0.77
tx(mm)	4.36	2.51	1.37	0.48	**0.36**	2.97	1.77	0.52	0.40
ty(mm)	2.39	1.37	0.71	**0.27**	**0.27**	1.68	1.02	0.45	0.38
tz(mm)	22.09	12.70	6.58	1.06	**0.45**	15.26	9.04	1.74	0.85
Lost Number	21	10	5	0	0	14	9	0	0

Table 1. Trinocular tracking evaluation on *Object moves freely with fixed cameras* mode.

Camera Angle	Mono.	30°	45°	60°	90°	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
r(°)	1.41	1.15	1.05	1.06	**0.79**	0.93	0.87
tx(mm)	0.39	0.64	0.55	0.38	0.36	**0.22**	0.31
ty(mm)	0.37	0.13	0.13	**0.12**	0.13	0.45	0.42
tz(mm)	15.07	2.48	1.37	0.61	**0.36**	1.92	1.39
Lost Number	9	0	0	0	0	0	0

Table 2. Binocular tracking evaluation on *Object rotates only with fixed cameras* mode.

Camera Angle	Mono.	30° *plane*	45° *plane*	30° *cone*	45° *cone*
Camera Index	C-0	C-0/C-1/C-3	C-0/C-2/C-4	C-0/C-1/C-5	C-0/C-2/C-6
r(°)	1.41	1.12	0.86	0.93	**0.85**
tx(mm)	0.39	0.39	0.39	0.34	**0.31**
ty(mm)	0.37	**0.12**	**0.12**	0.31	0.28
tz(mm)	15.07	0.90	**0.60**	1.55	0.99
Lost Number	9	0	0	0	0

Table 3. Trinocular tracking evaluation on *Object rotates only with fixed cameras* mode.

tainty in the view direction, resulting in increased error. Therefore, in practical applications, our primary purpose is to eliminate the uncertainty of the object pose through different views, meaning that increasing the camera angle is more preferred than increasing the camera number.

If we use multiple cameras, we should also increase the included angle between them as much as possible, uniformly distributed in space.

B.2. Binocular Tracking Results in Mode 2

Table 2 shows the binocular tracking evaluation results in mode 2, i.e., *Object rotates only with fixed cameras*. In this mode, C-0 to C-4 and the object constitute a $plane$, and C-5 and C-6 are outside the plane. Since the object only rotates, the translation errors in the X-axis and Y-axis directions during tracking are tiny. The X-axis translation error of some binocular tracking is larger than that of monocular tracking, which is caused by the geometric shape of the object.

Generally, when the camera angle is within 90°, as the camera angle increases, the rotation and translation errors gradually decrease, especially the translation in the Z-axis direction, which is consistent with the conclusions in the manuscript.

Since C-5 and C-6 are looking at the object from a higher position, the translation components of the X-axis and the Y-axis are a little different from C-1 to C-4, where the X-axis component is better but Y-axis component is worse.

B.3. Trinocular Tracking Results in Mode 2

Table 3 gives the trinocular evaluation results under mode 2. For the translation component, the $plane$ pattern is better than the $cone$ pattern, i.e., the precision depends on the two cameras with the largest angle. For the rotation component, the precision of the two patterns is very close, which is caused by the object only rotating.

Combining Table 2 and Table 3, we find that adding cameras between the two cameras will reduce the tracking accuracy, which is consistent with the conclusions above. For example, the Z-axis translation precision of C-0/C-4 is 0.36mm, while the corresponding precision in C-0/C-2/C-4 is only 0.60mm. At the same time, the rotation precision is also reduced.

B.4. Intermediate Results

This section analyzes the detailed effect of our multi-view method through the intermediate results. Fig. 2 shows the binocular tracking results results on real data. The first row is the input images of the two cameras with 90° included angle, the second row is the tracking result of TPAMI19, which performs the monocular tracking, and the third row is our multi-view tracking result. The C-1 image is the result of rendering with $\boldsymbol{T}_1$, and it can be seen that the reprojection region of TPAMI19 can precisely match the input image visually. The C-2 image is the rendering result of $\boldsymbol{T}'_2$, that is, transforming $\boldsymbol{O}_{c_1}$ to $\boldsymbol{O}_{c_2}$ by $\boldsymbol{T}'_2 =^2\boldsymbol{T}_1\boldsymbol{T}_1$ for rendering. We can see that TPAMI19 has an obvious translation error in the camera view direction. Our joint optimization can get the precise pose, resulting in the reprojection region on each camera image is precise. Fig. 3 shows the intermediate results on the synthetic data (*Cameras move freely* mode), which is consist with Fig. 2.

Fig. 4 is another set of results on the synthetic data. We use two cameras with 90° included angle to estimate the object pose and then observe the object with three other views, i.e., 30°, 45°, and 60°. We enlarge the image for better observation, where the purple contour in the figure is the rendering result, and we can see that they can be visually aligned with the object contour precisely in all views.

Fig. 5 is the trinocular tracking result on real data, which is also enlarged for better observation. We can see that

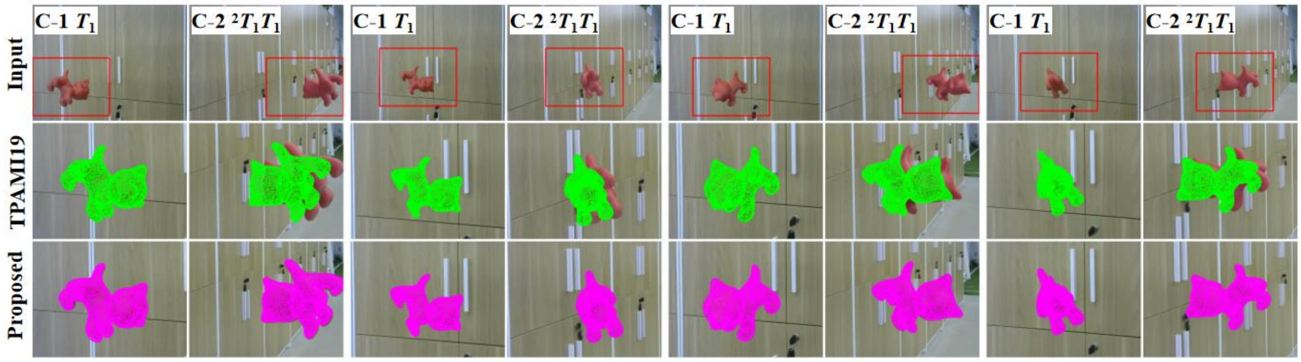


Figure 2. Binocular tracking results results on the real data. There are 4 sets of images captured from C-1 and C-2. TPAMI19 perform the monocular tracking on C-1 and map it to C-2 for display through ${}^2\boldsymbol{T}_1\boldsymbol{T}_1$, where we can see a large translation error in the camera view direction. Our method can joint the information of two cameras to get precise visual results under each camera.

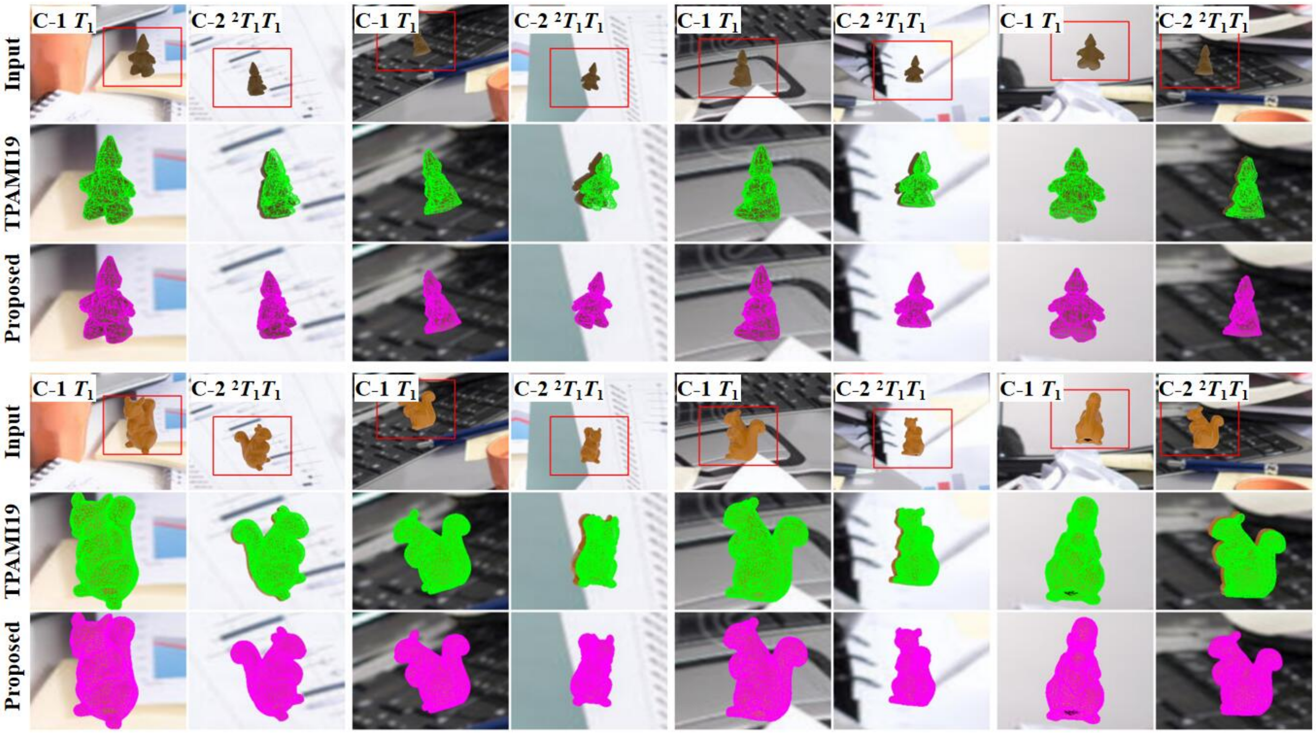


Figure 3. Binocular tracking results on the synthetic data (*cameras freely move*).

our method can get excellent visual performances under all views.

C. Monocular Tracking Evaluation on the BCOT Benchmark

In this section, we will give more details of the BCOT benchmark. Then we show more examples and evaluate state-of-the-art monocular tracking algorithms comprehensively.

C.1. More Details of the BCOT Benchmark

Time cost and iterations. The tracking time of the proposed multi-view tracking method depends on the selected basic monocular tracking method and the number of cameras used. There is no extra time needed to convert the basic coordinate frame to the object-centered coordinate frame.

For pose annotation when constructing BCOT Benchmark, the optimization executes 5 rounds (7 iterations per round), while for normal cases, only 1 round is required.

Sequences discarded. As stated in the manuscript, we

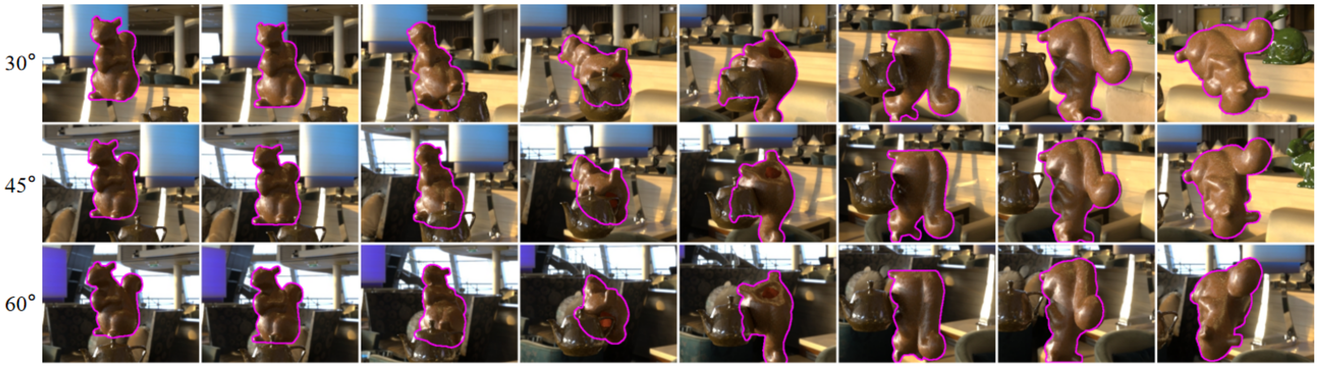


Figure 4. We use the basic camera (0° camera) and 90° camera to estimate the object pose and then observe the tracking result with three other perspectives, i.e., 30°, 45°, and 60° cameras. The object is precisely aligned with the image in all views.

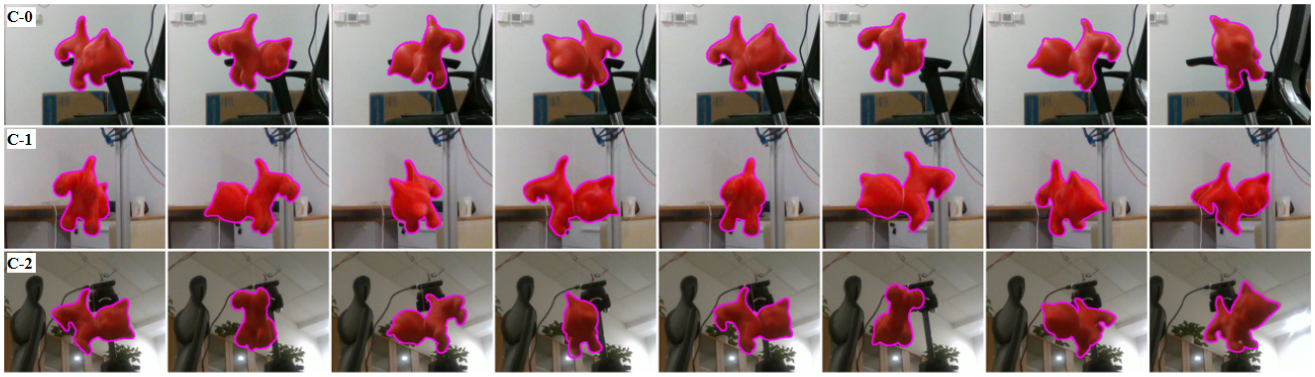


Figure 5. Trinocular tracking results on the real data. C-0 to C-2 represent the images captured by 3 cameras, respectively, and our method can get precise results in each view.

will discard sequences with large errors. Specifically, we discarded 36 sequences, i.e., 20×22-404=36.

C.2. More Examples of the BCOT Benchmark

Fig. 6–8 shows more BCOT benchmark examples. The blue contour is the result rendered according to the annotation pose. Our benchmark is markerless and can annotate high-precision poses.

C.3. Monocular Tracking Evaluation

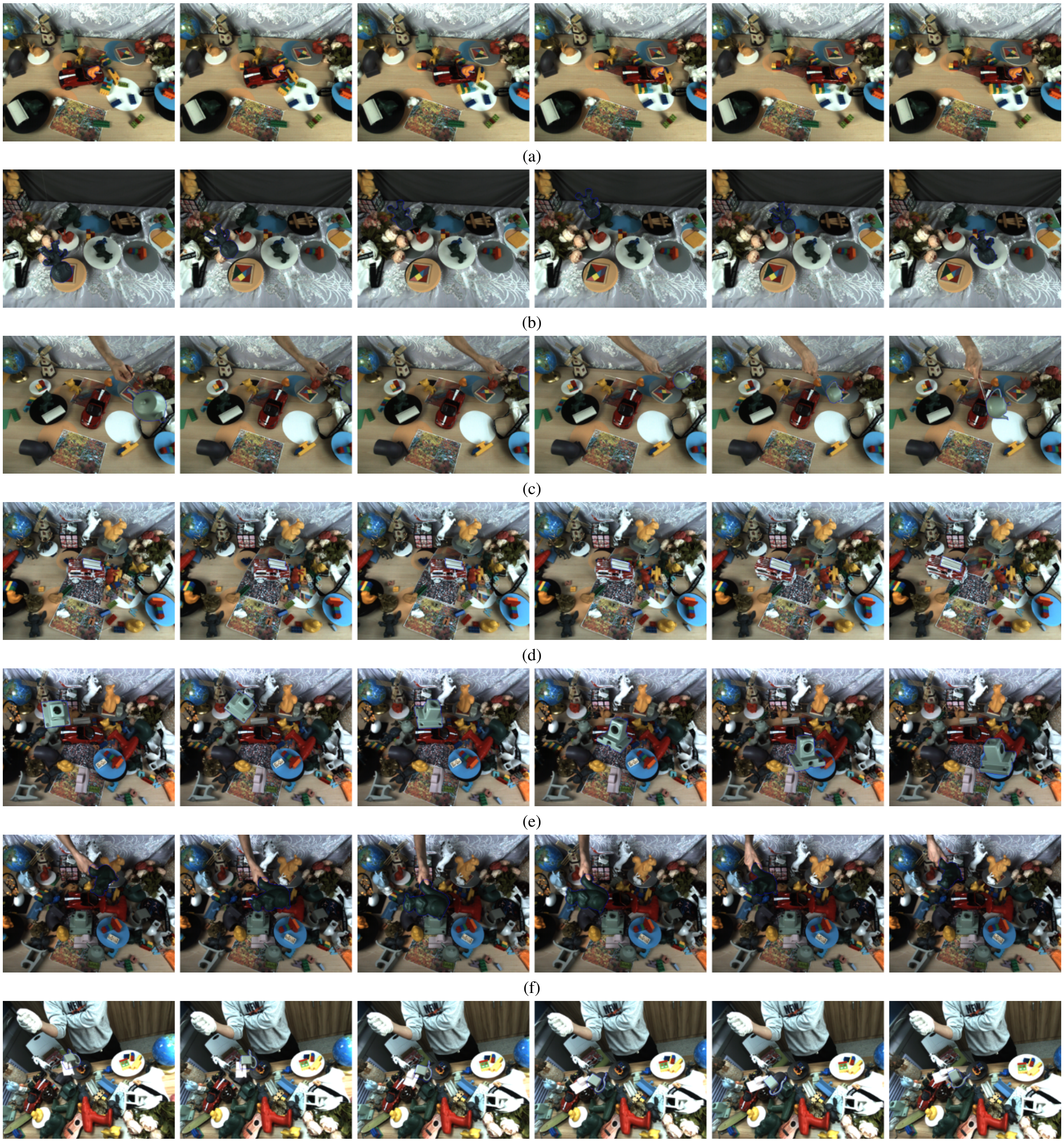
We further analyze the performance of the methods in indoor and outdoor scenes. Table 4 and Table 5 respectively show the accuracy of different methods in indoor and outdoor scenes. Fig. 9 and Fig. 10 respectively show the AUC scores of the ADD metric in the indoor scene and the outdoor scene.

Analysis. Through comparative analysis, the ACCV2020 gets the best performance overall. But as stated in the manuscript, it has some limitations when directly used. Besides, the origin of the object model coordinate frame needs to be set at the center of the model, which may limit the application scenario.

With further analysis, it can be seen from Table 4 that in indoor scenes, TVCG2021 achieves the best performance in ADD metric and $2°$,$2cm$ metric. This indicates that TVCG2021 has higher tracking precision in complex scenes, where the complexity of indoor scenes in the BCOT dataset is higher than that of outdoor scenes. In outdoor scenes, the background is relatively simple, ACCV2020 shows the best tracking performance, but TVCG2021 still has the highest rotation precision.

The reason is ACCV2020 uses prerendered templates in fixed discrete view angles (in order for acceleration), which introduces angular errors in templates and reduces its precision in rotation estimation. On the contrary, TVCG2021 render templates online, and there is no error in templates. The translation is insensitive to the small angular error of templates and thus is less affected.

In addition, it is found from Fig. 10 that MTAP2019 shows a high AUC score under the ADD metric. This is because MTAP2019 can obtain high translation precision in a simple background (also shown in the $2cm$ metric in Table 5), and the ADD error depends more on the translation error. However, except for ACCV2020 and MTAP2019, the



(a)

(b)

(c)

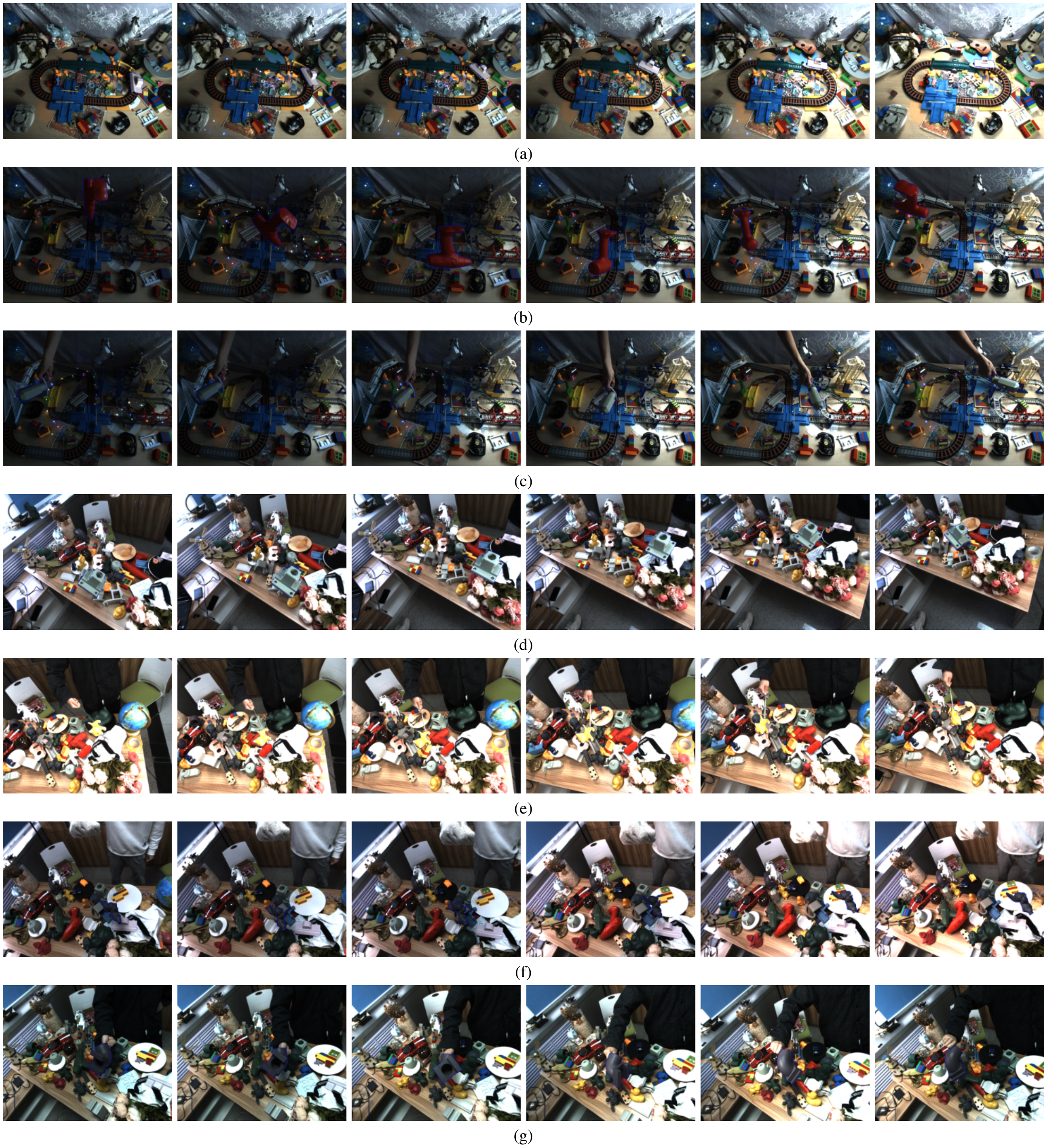
(d)

(e)

(f)

(g)

Figure 6. More examples of the BCOT benchmark, where the blue contour is rendered according to the annotation pose. (a) Ape model, static camera set, easy scene, translation movement. (b) Deadpool model, static camera set, easy scene, suspension movement. (c) Teapot model, static camera set, easy scene, handheld movement. (d) RTI Arm model, static camera set, complex scene, translation movement. (e) Lamp Clamp model, static camera set, complex scene, suspension movement. (f) Squirrel model, static camera set, complex scene, handheld movement. (g) RJ45 Clip model, movable camera set, complex scene, suspension movement, occlusion.



(a)

(b)

(c)

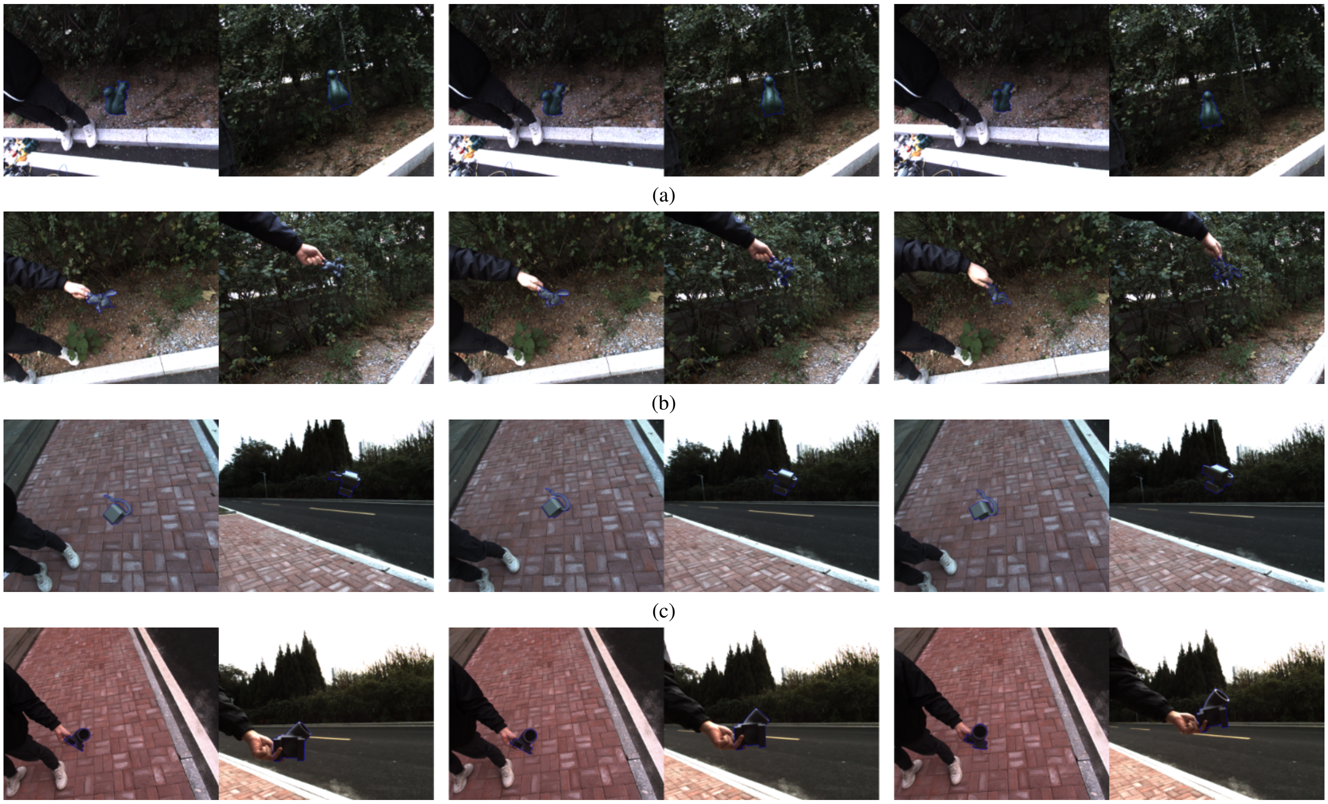
(d)

(e)

(f)

(g)

Figure 7. More examples of the BCOT benchmark, where the blue contour is rendered according to the annotation pose. (a) Wall Shelf model, static camera set, dynamic light, translation movement. (b) Driller model, static camera set, dynamic light, suspension movement. (c) 3D Touch model, static camera set, dynamic light, handheld movement. (d) Lamp Clamp model, movable camera set, complex scene, suspension movement. (e) Cat model, movable camera set, complex scene, handheld movement. (f) Stitch model, movable camera set, complex scene, dynamic light, suspension movement. (g) Tube model, movable camera set, complex scene, dynamic light, handheld movement.



(a)

(b)

(c)

(d)

Figure 8. More examples of the BCOT benchmark, where the blue contour is rendered according to the annotation pose. The outdoor scenes provide both two views. (a) Squirrel model, outdoor scene 1, movable camera set, suspension movement. (b) Stitch model, outdoor scene 1, movable camera set, handheld movement. (c) RJ45 Clip model, outdoor scene 2, movable camera set, suspension movement. (d) Tube model, outdoor scene 2, movable camera set, handheld movement.

Method	ADD$-0.02d$	ADD$-0.05d$	ADD$-0.1d$	$5°, 5cm$	$5°$	$5cm$	$2°, 2cm$	$2°$	$2cm$	Time(ms)
MTAP2019	7.0	32.5	61.6	57.9	58.3	97.8	15.4	16.9	75.1	8.6
TPAMI2019	17.7	43.5	66.5	75.0	75.8	92.0	44.8	47.5	76.0	33.3
CGF2020	18.7	45.5	70.6	83.2	83.5	96.0	52.9	56.8	81.2	32.2
ACCV2020	9.8	43.1	76.4	**88.2**	**88.4**	**99.6**	45.6	49.2	**87.8**	**3.5**
C&G2021	13.5	42.6	69.2	83.1	84.0	96.6	43.8	47.6	79.9	19.2
JCST2021	21.7	52.0	76.9	87.0	87.4	97.8	55.8	58.5	86.0	38.8
TVCG2021	**23.6**	**55.5**	**78.5**	87.1	87.4	97.3	**58.2**	**60.6**	87.1	32.8

Table 4. Comparison of monocular 3D tracking methods of indoor scenes.

Method	ADD$-0.02d$	ADD$-0.05d$	ADD$-0.1d$	$5°, 5cm$	$5°$	$5cm$	$2°, 2cm$	$2°$	$2cm$	Time(ms)
MTAP2019	3.2	32.9	69.0	49.3	49.9	97.9	8.0	9.0	82.0	9.1
TPAMI2019	2.9	14.0	43.3	80.3	84.3	91.4	34.8	49.5	55.7	36.6
CGF2020	2.1	10.3	38.1	85.4	87.5	95.4	33.6	52.6	53.9	34.3
ACCV2020	**12.5**	**49.0**	**77.6**	**90.3**	**90.7**	**99.4**	**46.5**	50.0	**87.8**	**3.3**
C&G2021	2.7	15.2	41.8	81.7	85.8	92.7	30.8	46.1	55.1	18.4
JCST2021	3.8	17.6	49.2	87.1	89.1	96.3	41.9	55.7	64.2	37.9
TVCG2021	3.9	16.8	47.9	87.2	90.3	94.7	41.3	**56.7**	60.7	37.8

Table 5. Comparison of monocular 3D tracking methods of outdoor scenes.

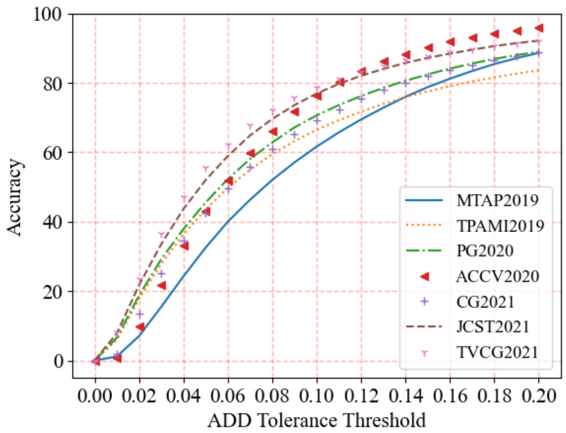


Figure 9. Indoor scene tracking accuracy under various ADD error tolerance thresholds.

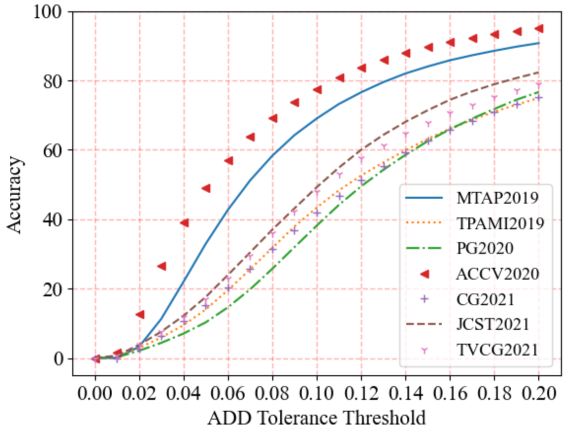


Figure 10. Outdoor scene tracking accuracy under various ADD error tolerance thresholds.

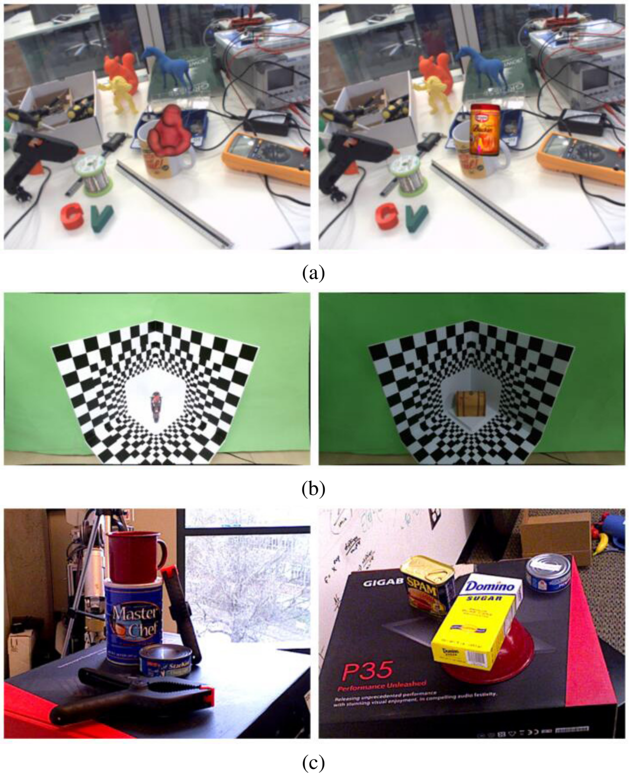
AUC scores of ADD in other methods have decreased in outdoor scenes, indicating that their translation errors have increased. This shows domain differences between outdoor and indoor scenes, which may affect the tracking method.

D. Comparison with Other Tracking Datasets

Other datasets used for 3D object tracking include RBOT, OPT, and YCB-Video. Fig. 11 shows some examples of other datasets.

Datas. Fig. 11(a) is an example of the RBOT dataset. It is semi-synthetic with rendered foreground objects. The synthesized dynamic light, noise, occlusion, and object motion are very different from real scenes, which also prevents them from being used by the learning-based methods.

Fig. 11(b) is an example of the OPT dataset. It is a real scene dataset, and the GT pose is calculated by artificial markers. Objects in the dataset are always stationary, surrounded by white areas and a large number of artificial markers.


(a)

(b)

(c)

Figure 11. Examples of other datasets. (a) RBOT dataset. (b) OPT dataset. (c) YCB-Video dataset.

Fig. 11(c) shows an example of the YCB-Video dataset. It is also a real scene dataset, which calculates the GT pose of objects through depth data. As stated in the manuscript, this annotation method will suffer from large errors. In addition, objects in YCB-Video also remain stationary.

The main feature of BCOT is real-scene and high-precision. It is markerless, and the camera and object are both dynamic. Besides, its labeling error also achieves the highest precision. Currently, the learning-based and the optimization-based methods are studied separately on different benchmarks (RBOT v.s. YCB-Video). We believe that one important reason is the lacking of high-precision real-scene datasets, which now can be addressed with BCOT.