Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains

Bowen Wen
Rutgers University
bw344@scarletmail.rutgers.edu

Chaitanya Mitash
Amazon
cmitash@amazon.com

Kostas Bekris
Rutgers University
kostas.bekris@cs.rutgers.edu

Abstract

Tracking the 6D pose of objects in video sequences is important for robot manipulation. This work presents se(3)-TrackNet, a data-driven optimization approach for long-term, 6D pose tracking. It aims to identify the optimal relative pose given the current RGB-D observation and a synthetic image conditioned on the previous best estimate and the object's model. The key contribution in this context is a novel neural network architecture, which appropriately disentangles the feature encoding to help reduce domain shift, and an effective 3D orientation representation via Lie Algebra. Consequently, even when the network is trained solely with synthetic data can work effectively over real images. Comprehensive experiments over multiple benchmarks show se(3)-TrackNet achieves consistently robust estimates and outperforms alternatives, even though they have been trained with real images. The approach runs in real time at 90.9Hz. Code, data and supplementary video for this project are available at[1] *https://github.com/wenbowen123/iros20-6d-pose-tracking*

1. Introduction

Robotic tasks, such as object manipulation, often require to track the pose of an object. Pose estimation from a single snapshot can initiate a manipulation pipeline [26, 30]. Purposeful manipulation, however, such as placement [15, 17] frequently involves dynamic object pose change. Some pose estimation approaches are relatively fast and can re-estimate pose from scratch for every frame [22, 25, 26, 28]. This can be redundant, however, and often leads to less coherent estimations over consecutive frames, which negatively impact manipulation.

Temporal tracking of object poses over sequences of images can greatly improve speed while maintain or even improve pose quality [14, 20, 21, 27, 29]. Nevertheless, many traditional methods that depend on hand-crafted likelihood functions and features, require extensive hyper-parameter tuning when adapted to novel object categories or environments. On the other hand, data-driven techniques [7, 12] require real-world training data, which are difficult to acquire and label in the context of 6D poses.

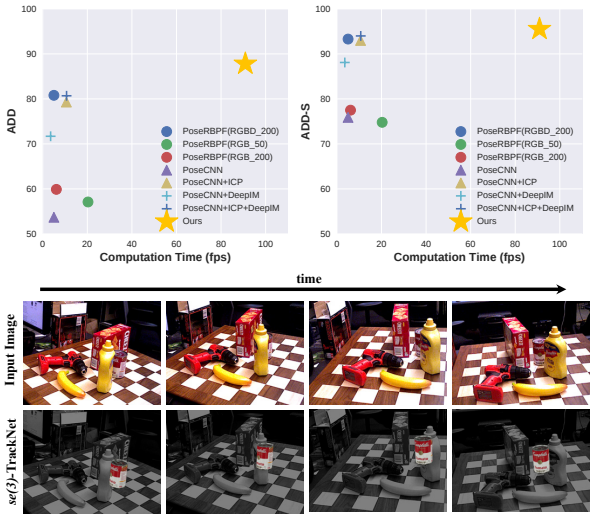

Figure 1: **Top:** Performance w.r.t. computation time evaluated on the YCB-Video dataset according to the area under the curve (AUC) metric for the ADD and ADD-S objectives [30]. The proposed approach achieves more accurate tracking and is significantly faster than alternatives. **Bottom:** Pose predicted by the *se*(3)-TrackNet without any re-initialization, which is able to recover from complete occlusion.

This work proposes a data-driven optimization strategy to keep long-term track of an object's 6D pose robustly. The contributions are the following:

1. A novel deep neural network for 6D object pose tracking, where a smart feature-encoding disentanglement technique enables more efficient sim-to-real transfer.
2. A Lie Algebra representation of 3D orientations, which allows effective learning of the residual pose transforms given a proper loss function.
3. A training pipeline over synthetic data that employs domain randomization [24] in the context of pose tracking.
4. A novel benchmarking dataset for 6D pose tracking in the context of multiple different robotics manipulation tasks, with groundtruth 6D pose annotation.

Experiments indicate that the proposed network achieves state-of-art results on the YCB-Video benchmark without

[1] This work has been previously accepted to appear at IEEE IROS 2020.

re-initialization in contrast to prior work [4, 12]. It is also significantly faster at 90.9Hz, as shown in Fig. 1.

2. Related Work

6D Object Pose Estimation: Learning-based techniques have shown promise in directly regressing the 6D object pose from image data [8, 10, 13, 16, 22]. Nevertheless, given the complexity of the 6D challenge, a large amount of pose annotated training data is required to achieve satisfactory results. This is often more challenging than labeling for object classification or detection. Given that a pose is re-estimated in every frame, estimation techniques often trade-off speed for accuracy [30] while still insufficient for real-time application. This is desirable for manipulation that often involves dynamic scene updates. In contrast, $se(3)$-TrackNet exploits temporal information to achieve higher accuracy and faster response than state-of-art single-image pose estimation methods while using only synthetic data for training.

6D Object Pose Tracking: For setups where CAD object models are available, prior approaches [3, 10, 29] harness GPUs for probabilistic filtering. They often require, however, hand-crafted feature design and carefully tuned hyper-parameters, which do not generalize easily. Recent work [4] proposed a data-driven Rao-Blackwellized particle filter, achieving promising results on the YCB Video benchmark. It suffers, however, from severe occlusions and requires re-initialization from costly pose estimation. Another effort computes relative transformations between consecutive frames by optimizing for the discrepancy between consecutive observations [11, 19, 20, 23, 31]. The most related effort to the current paper leverages the FlowNetSimple network [6] for learning 6D pose tracking. It requires, however, occasional re-initialization and has to be trained at least partially with real data.

3. Approach of $se(3)$-TrackNet

The objective is to compute the 6D pose of an object $T_t \in SE(3)$ at any time $t > 0$, given as input:

- a 3D CAD model of the object,
- its initial pose, $T_0 \in SE(3)$, computed by any single-image based 6D pose estimation technique, and a
- sequence of RGB-D images $O_\tau, \tau \in \{0, 1, ..., t-1\}$ from previous time stamps and the current observation O_t.

This work proposes a data-driven optimization technique to track the 6D object pose. The cost function for the optimization is encoded and learned by a novel neural network architecture, trained with only synthetically generated data. In every time step, it computes a relative pose ΔT_τ from the previous frame as indicated in Fig. 2.

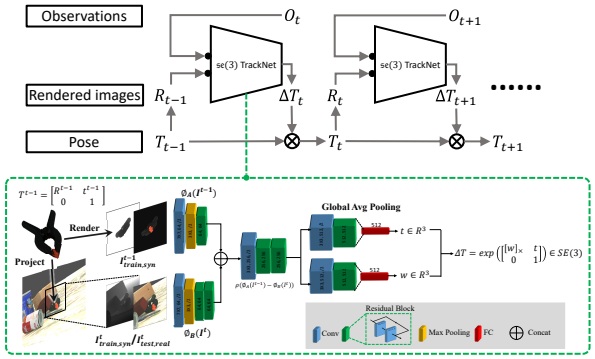

Figure 2: Overview: At any given time t the current observation O_t and the rendering R_{t-1} of the object model based on the previously computed pose $_{t-1}$ are passed to the $se(3)$-TrackNet via two separate feature encoders ϕ_B and ϕ_A respectively. Both inputs are synthetic during training while at test time, the input to ϕ_B is a real observed image. The encoders' outputs are concatenated and used to predict the relative pose ΔT_t between the two images, with decoupled translation and rotational. ΔT_t is then combined with T_{t-1} to compute T_t. The same process repeats along the video.

3.1. Tracking on $SE(3)$ Manifolds with Residuals

Optimization in this domain operates over cost functions defined for the object poses $\bar{\xi}, \xi$ that measure the discrepancy ε between the features extracted from the images:

$$\varepsilon = \rho(\phi_{I_1}(\bar{\xi}) - \phi_{I_2}(\xi)),$$

where ρ is a predefined robust loss function, and $\phi(\cdot)$ can be direct pixel intensity values, such as in [5], point-to-point discrepancy or its variations [1], pre-designed features [9] or the combinations from any of the above [19].

Given the current observation O_t, and the pose computed in the previous timestamp ξ_{t-1}, the goal of this work is to find a relative transformation $\Delta\xi$ that takes the object from ξ_{t-1} to the pose captured by the current observation. This can be formulated as an optimization problem. Let R denote the image corresponding to the object model's rendering at the given pose. Then, the optimal relative transform is:

$$\Delta\xi^* = \underset{\Delta\xi}{\operatorname{argmin}}\{\rho(\phi_{O_t}(\xi_t) - \phi_R(\xi_{t-1} \boxplus \Delta\xi))\}$$

An analytical solution is to perform Gauss-Newton optimization, which often requires appropriate choice of a robust cost function and hand-crafted features. Another problem arises when different modalities are involved and suitable tuning of relative weights struggles to generalize to new scenarios. Instead, this work proposes a novel neural network architecture that implicitly learns to calibrate the residual between the features extracted from the current observation and the rendered image conditioned on previous pose estimate to resolve the relative transform in the tangent space $\Delta\xi \in se(3)$.

3.2. Neural Network Design

The proposed neural network is shown in Fig. 2. The network takes as input a pair of images, I^{t-1}: rendered from the previous pose estimation, and I^t: the current observation. The images are 4-channel RGB-D data. During training, both inputs are synthetically generated images

$\phi(I^{t-1}_{syn,train}; I^{t}_{syn,train})$, while for testing the current timestamp input comes from a real sensor, $\phi(I^{t-1}_{syn,test}; I^{t}_{real,test})$. The $se(3)$-TrackNet uses two separate input branches for I^{t-1} and I^{t}. The weights of the feature encoders are not shared so as to disentangle feature encoding. This is different from related work [12], where the two images are concatenated into a single input. A shared feature extractor worked in the previous work when both real and synthetic data are available during training. The property of the latent space $\phi(I^{t-1}_{syn,train}; I^{t}_{real,train})$ could still be partly preserved when tested on real world test scenarios $\phi(I^{t-1}_{syn,test}; I^{t}_{real,test})$. This representation, however, does not generalize to training exclusively on synthetic data.

In particular, let's denote the latent space features trained on purely synthetic data as $\phi_A(I^{t-1}_{syn,train})$ and $\phi_B(I^{t}_{syn,train})$. When tested on real world data, the latent space features are $\phi_A(I^{t-1}_{syn,test})$ and $\phi_B(I^{t}_{real,test})$. By this feature encoding disentanglement, domain gap reduces to be between $\phi_B(I^{t}_{syn,train})$ and $\phi_B(I^{t}_{real,test})$, while $\phi_A(I^{t-1}_{syn,train})$ and $\phi_A(I^{t-1}_{syn,test})$ can be effortlessly aligned between the training and test phase without the need for tackling the domain gap problem.

A relative transform can be predicted by the network via end-to-end training. The transformation is represented by Lie algebra as $\Delta\xi = (t,w)^T \in se(3)$, where the prediction of w and t are disentangled into separate branches and trained by L_2 loss:

$$L = \lambda_1 ||w - \bar{w}||_2 + \lambda_2 ||t - \bar{t}||_2$$

where λ_1 and λ_2 has been simply set to 1 in experiments. Given $\Delta\xi$, the current pose estimate is computed as $T^t = exp(\Delta\xi) \cdot T^{t-1}$, as described in Sec. 3.1. The details of the network architecture will be described in the released code.

3.3. Synthetic Data Generation via PPDR

Domain randomization provides variability at training time, such that at test time, the model is able to generalize to real-world data [24]. Prior work implements the idea of domain randomization by directly sampling object poses from a predetermined distribution [24, 25, 30], which can lead to unrealistic penetration or pile configurations, adversely introducing undesired bias to depth data distribution during learning. This work therefore leverages the complementary attributes of domain randomization and physically-consistent simulation for the synthetic data generation process, namely PPDR (Physically Plausible Domain Randomization). Specifically, object poses are initialized randomly where collision between objects or distractors could occur, which is then followed by a number of physics simulation steps so that objects are separated or fall onto the table without collision. Meanwhile, other complex or intractable physical properties such as lighting, number of objects, distractor textures are randomized. For each scene, a pair of data $I^{t-1}_{syn,train}$ and $I^{t}_{syn,train}$ are generated. It is later utilized as the input to the network for training. Both images are cropped and resized into a fixed resolution 176×176 before feeding into the network - similar to prior work [12].

The next step is to bridge the domain gap of depth data via *bidirectional alignment*, operated over the synthetic depth data during training time and real depth data during test time. Specifically, during training time, random Gaussian noise and random pixel corruption augmentation steps are applied to the synthetic depth data $D^{t}_{syn,train}$ at branch B to resemble a real noisy depth image. In contrast, during test time, bilateral filtering is carried out on the real depth image so as to smooth sensor noise and fill holes to be aligned with the synthetic domain.

4. Experiments

This section evaluates the proposed approach and compares against state-of-the-art 6D pose tracking methods as well as single-image pose estimation methods on a public benchmark. It also introduces a new benchmark developed as part of this work, which corresponds to robot manipulation scenarios. Experiments are conducted on a standard desktop with Intel Xeon(R) E5-1660 v3@3.00GHz processor with a NVIDIA Tesla K40c GPU. Both quantitative and qualitative results demonstrate $se(3)$-TrackNet achieves the state-of-art results in terms of both accuracy and speed, while using only synthetic training data.

4.1. Datasets

YCB-Video Dataset This dataset [30] captures 92 RGB-D video sequences over 21 YCB Objects [2] arranged on table-tops. Objects' groundtruth 6D poses are annotated in every frame. The evaluation closely follows the protocols adopted in comparison methods [4, 12, 26, 30] and reports the AUC (Area Under Curve) results on the keyframes in 12 video test sequences evaluated by the metrics of $ADD = \frac{1}{m}\sum_{x \in M} ||Rx + T - (\hat{R}x + \hat{T})||$ which performs exact model matching, and $ADD-S = \frac{1}{m}\sum_{x_1 \in M} \min_{x_2 \in M} ||Rx_1 + T - (\hat{R}x_2 + \hat{T})||$ [30].

YCBInEOAT Dataset In order to evaluate the 6D tracking performance in the setup of dynamic moving object, in this work, a novel dataset, referred to as "YCBInEOAT Dataset", is developed in the context of robotic manipulation, where various robot end-effectors are included: a vacuum gripper, a Robotiq 2F-85 gripper, and a Yale T42 Hand [18]. The manipulation sequences consider 5 YCB objects outlined in Table 1. Each video sequence is collected from a real manipulation performed with a dual-arm *Yaskawa Motoman SDA10f*. In general, there are 3 types of manipulation tasks performed: (1) single arm pick-and-place, (2) within-hand manipulation, and (3) pick to hand-off between arms to placement. RGB-D images are captured by an *Azure Kinect* sensor mounted statically on the

	DOPE [25]		DenseFusion [26]		PoseCNN+DeepIM [12]		DeepIM tracking [12]		RGF [10]		Wüthrich's [29]		*se*(3)-TrackNet		PoseRBPF [4]		*se*(3)-TrackNet		*se*(3)-TrackNet	
Modality	RGB		RGBD		RGBD		RGB		Depth		Depth		RGBD		RGBD		RGBD		RGBD	
Type	detection		detection		detection		tracking		tracking		tracking		tracking		tracking		tracking		tracking	
Initial pose from	-		-		-		groundtruth		groundtruth		groundtruth		groundtruth		PoseCNN		PoseCNN		PoseCNN	
Re-initialization (Total)	No		No		No		Yes (290)		No		No		No		Yes (2)		No		Yes (2)	
Train data	Syn		Real+Syn		Real+Syn		Real+Syn		-		-		Syn		Syn		Syn		Syn	
Objects	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
002_master_chef_can	-	-	-	96.40	78.00	96.30	89.00	93.80	46.23	90.17	55.60	90.68	93.86	96.29	90.50	95.10	93.84	95.92	93.84	95.92
003_cracker_box	55.90	69.80	-	95.50	91.40	95.30	88.50	93.00	56.95	72.26	96.38	97.19	96.52	97.20	88.20	93.00	96.42	97.12	96.42	97.12
004_sugar_box	75.70	87.10	-	97.5	97.60	98.20	94.30	96.30	50.38	72.65	97.14	97.94	97.58	98.14	92.90	95.50	97.56	98.13	97.56	98.13
005_tomato_soup_can	76.10	85.10	-	94.60	90.30	94.80	89.10	93.20	72.44	91.60	64.74	89.55	94.96	97.17	90.00	93.80	94.81	97.10	94.81	97.10
006_mustard_bottle	81.90	90.90	-	97.20	97.10	98.00	92.00	95.10	87.71	98.19	97.12	97.95	95.76	97.37	91.90	96.30	95.73	97.36	95.73	97.36
007_tuna_fish_can	-	-	-	96.60	92.20	98.00	92.00	96.40	28.67	52.93	69.14	93.32	86.46	91.09	91.10	95.30	86.46	91.08	86.46	91.08
008_pudding_box	-	-	-	96.50	83.50	90.60	80.10	88.30	12.69	17.98	96.85	97.89	97.93	98.39	85.80	92.00	97.90	98.37	97.90	98.37
009_gelatin_box	-	-	-	98.10	98.00	98.50	92.00	94.40	49.10	70.72	97.46	98.37	97.81	98.42	96.30	97.50	97.74	98.46	97.74	98.46
010_potted_meat_can	39.40	52.40	-	91.30	82.20	90.30	78.00	88.90	44.09	45.57	83.71	86.69	77.81	84.16	68.70	77.90	36.45	60.28	74.51	82.38
011_banana	-	-	-	96.60	94.90	97.60	81.00	90.50	93.33	97.74	86.27	96.07	94.90	97.18	74.20	86.90	40.04	78.81	84.62	95.15
019_pitcher_base	-	-	-	97.10	97.40	97.90	90.40	94.70	97.93	98.18	97.30	97.74	96.75	97.45	86.80	94.20	96.71	97.43	96.71	97.43
021_bleach_cleanser	-	-	-	95.80	91.60	96.90	81.70	90.50	95.87	97.28	95.23	97.16	95.94	97.25	86.00	93.00	95.89	97.23	95.89	97.23
024_bowl	-	-	-	88.20	8.10	87.00	38.80	90.60	24.25	82.40	30.37	97.15	80.91	94.46	25.50	94.20	39.12	95.56	39.12	95.56
025_mug	-	-	-	97.10	94.20	97.60	83.20	92.00	59.99	71.18	83.15	93.35	91.53	96.88	90.90	97.10	91.56	96.88	91.56	96.88
035_power_drill	-	-	-	96.00	97.20	97.90	85.40	92.30	97.94	98.35	97.09	97.82	96.42	97.40	93.90	96.10	96.38	97.38	96.38	97.38
036_wood_block	-	-	-	89.70	81.10	91.50	44.30	75.40	45.68	62.51	95.48	96.87	95.16	96.70	20.10	89.10	33.91	95.92	33.91	95.92
037_scissors	-	-	-	95.20	92.70	96.00	70.30	84.50	20.94	38.60	4.17	16.20	95.68	97.55	76.10	85.60	95.67	97.54	95.67	97.54
040_large_marker	-	-	-	97.50	88.90	98.20	80.40	91.20	12.17	18.90	35.58	53.02	92.15	95.99	92.00	97.10	89.01	94.23	89.01	94.23
051_large_clamp	-	-	-	72.90	54.20	77.90	73.90	84.10	62.84	80.12	61.25	72.35	94.71	96.93	48.50	94.80	71.60	96.88	71.60	96.88
052_extra_large_clamp	-	-	-	69.80	36.50	77.80	49.30	90.30	67.48	69.65	93.73	96.58	91.74	95.76	40.30	90.10	64.58	95.80	64.58	95.80
061_foam_brick	-	-	-	92.50	48.20	97.60	91.60	95.50	69.99	86.55	96.76	98.11	93.65	96.71	81.10	95.70	40.66	94.67	40.66	94.67
ALL	-	-	-	93.10	80.70	94.00	79.30	91.00	59.18	74.29	78.01	90.21	93.05	95.71	80.80	93.30	84.46	93.87	87.81	95.52
Speed (fps)	4.31		16.67		0.09		12.00		11.76		12.93		90.90		5.00		90.90		90.90	

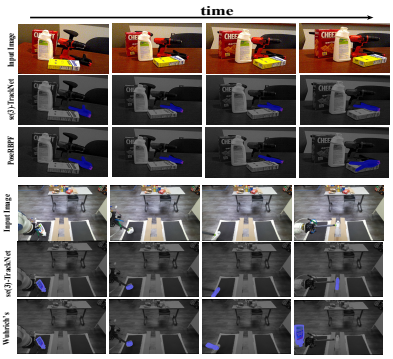

Figure 3: **Left:** Comparing the performance of $se(3)$-TrackNet (Gray) with state-of-the-art techniques on the *YCB Video*. The approach significantly outperforms the competing approaches over the ADD metric, which considers semantic information during pose evaluation. It also achieves the highest success rate over the ADD-S metric both in cases of initialization with the ground-truth pose and when initialized with the output of PoseCNN [30] (rightmost two columns). **Top-right:** Qualitative results for tracking the "large-clamp" object in the YCB-Video dataset. **Bottom-right:** Tracking results for "bleach-cleanser" being manipulated by a vacuum gripper in the *YCBInEOAT* dataset.

robot with a frequency of 20 to 30 Hz. Similar to the YCB-Video, ADD and ADD-S metrics are adopted for evaluation. Ground-truth 6D object poses in camera's frame have been accurately annotated *manually* for each frame of the video.

4.2. Results on YCB-Video

Fig. 3 present the evaluation over the *YCB-Video* dataset. Although this dataset contains real world annotated training data, $se(3)$-TrackNet does not use any of them but is trained solely on synthetic data generated by aforementioned pipeline. It is compared with other state-of-art 6D object pose detection approaches [12, 25, 26, 30] and 6D pose tracking approaches [4, 10, 12, 29], where publicly available source code[2] is used to evaluate [10, 29], while other results are adopted from the respective publications. All the compared tracking methods except PoseRBPF are using ground-truth pose for initialization. PoseRBPF [4] is the only one that is initialized using predicted poses from PoseCNN [30]. For fairness, two additional experiments using the same initial pose as PoseRBPF[3] are performed and presented in the rightmost two columns of Table I, one is without any re-initialization, and the other allows re-initialization by PoseCNN twice (same as in PoseRBPF) after heavy occlusions. The prior work [12] was originally proposed to refine the pose output from any 6D pose estimation detection method, but also extends to RGB-based tracking. It has to be re-initialized by PoseCNN when the last 10 frames have an average rotation greater than 10 degrees or an average translation greater than 1 cm, which happens every 340 frames on average as reported [12].

4.3. Results on YCBInEOAT-Dataset

Table 1 shows the quantitative results evaluated by the area under the curve for ADD and ADD-S on the developed *YCBInEOAT* dataset. On this benchmark, the tracking approaches with publicly available source code could be directly evaluated [10, 29]. Pose is initialized with ground-truth in the first frame and no re-initialization is allowed. Example qualitative results are demonstrated by Fig. 3. Abrupt motions, extreme rotations and slippage within the end-effector expose challenges for 6D object pose tracking. Nevertheless, $se(3)$-TrackNet is sufficiently robust to perform long-term reliable pose tracking.

Objects	RGF [10]		Wüthrich's [29]		*se*(3)-TrackNet	
	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
003_cracker_box	34.78	55.44	79.00	88.13	90.76	94.06
021_bleach_cleanser	29.40	45.03	61.47	68.96	89.58	94.44
004_sugar_box	15.82	16.87	86.78	92.75	92.43	94.80
005_tomato_soup_can	15.13	26.44	63.71	93.17	93.40	96.95
006_mustard_bottle	56.49	60.17	91.31	95.31	97.00	97.92
ALL	29.98	39.90	78.28	89.18	92.66	95.53

Table 1: Results evaluated on YCBInEOAT-dataset by AUC (Area Under Curve) for ADD and ADD-S.

4.4. Ablation Study

An ablation study investigates the importance of different modules of the proposed approach. It is performed for the *large clamp* object from the *YCB-Video dataset* and is presented in the accompanying table. The initial pose is given by ground-truth and no re-initialization is allowed. **No physics** implies that physics simulation is removed during synthetic data generation. For **No depth**, the depth modality is removed in both training and inference stage. **Shared encoder** means I^t and I^{t-1} are passed to the same feature encoder. **Quaternion** replaces the proposed rotation representation by quaternion. **Shape-Match Loss** [30] does not require the specification of symmetries. However, it loses track of the object early in the video.

Criteria	ADD	ADD-S
Proposed	94.71	96.93
No physics	91.88	95.76
No depth	75.65	87.22
Shared encoder	0.28	0.28
Quaternion	93.58	96.39
Shape-Match Loss	1.93	5.48

[2] https://github.com/bayesian-object-tracking/dbot

[3] We thank the authors for providing the initial poses they used in their original paper [4].

References

[1] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*. International Society for Optics and Photonics, 1992.

[2] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *IEEE ICAR 2015*.

[3] Changhyun Choi and Henrik I Christensen. Rgb-d object tracking: A particle filter approach on gpu. In *IROS*, 2013.

[4] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 2021.

[5] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE T-PAMI 2017*.

[6] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *ICCV*, 2015.

[7] Mathieu Garon and Jean-François Lalonde. Deep 6-dof tracking. *IEEE transactions on visualization and computer graphics*, 23(11):2410–2418, 2017.

[8] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *CVPR*, pages 11632–11641, 2020.

[9] Paul Hebert, Nicolas Hudson, Jeremy Ma, Thomas Howard, Thomas Fuchs, Max Bajracharya, and Joel Burdick. Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In *ICRA 2012*.

[10] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. Depth-based object tracking using a robust gaussian filter. In *ICRA*, 2016.

[11] David Joseph Tan, Federico Tombari, Slobodan Ilic, and Nassir Navab. A versatile learning-based 3d temporal tracker: Scalable, robust, online. In *ICCV 2015*.

[12] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *ECCV 2018*.

[13] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *ICCV 2019*.

[14] Leon Magnus, Svenja Menzenbach, Max Siebenborn, Niklas Funk, Boris Belousov, and Georgia Chalvatzaki. Object tracking for robotic assembly dataset creation and tracking evaluation.

[15] Chaitanya Mitash, Rahul Shome, Bowen Wen, Abdeslam Boularias, and Kostas Bekris. Task-driven perception and manipulation for constrained placement of unknown objects. *IEEE Robotics and Automation Letters*, 2020.

[16] Chaitanya Mitash, Bowen Wen, Kostas Bekris, and Abdeslam Boularias. Scene-level pose estimation for multiple instances of densely packed objects. *CoRL*, 2019.

[17] Andrew S. Morgan*, Bowen Wen*, Junchi Liang, Abdeslam Boularias, Aaron M. Dollar, and Kostas Bekris. Vision-driven compliant manipulation for reliable, high-precision assembly tasks. *RSS*, 2021.

[18] Lael Odhner, Raymond R Ma, and Aaron Dollar. Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy. *IEEE Transactions on Automation Science and Engineering 2013*.

[19] Karl Pauwels and Danica Kragic. Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking. In *IROS 2015*.

[20] Tanner Schmidt, Richard A Newcombe, and Dieter Fox. Dart: Dense articulated real-time tracking. In *RSS 2014*.

[21] Manuel Stoiber, Martin Pfanne, Klaus H Strobl, Rudolph Triebel, and Alin Albu-Schaeffer. A sparse gaussian approach to region-based 6dof object tracking. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[22] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV 2018*.

[23] Henning Tjaden, Ulrich Schwanecke, Elmar Schömer, and Daniel Cremers. A region-based gauss-newton approach to real-time monocular multiple object tracking. *TPAMI 2018*.

[24] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS 2017*.

[25] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *CoRL*, 2018.

[26] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. *CVPR*, 2019.

[27] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, 10/2020 2020.

[28] Bowen Wen, Chaitanya Mitash, Sruthi Soorian, Andrew Kimmel, Avishai Sintov, and Kostas E Bekris. Robust, occlusion-aware pose estimation for objects grasped by adaptive hands. *ICRA*, 2020.

[29] Manuel Wüthrich, Peter Pastor, Mrinal Kalakrishnan, Jeannette Bohg, and Stefan Schaal. Probabilistic object tracking using a range camera. *IROS 2013*.

[30] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *RSS*, 2018.

[31] Leisheng Zhong and Li Zhang. A robust monocular 3d object tracking method combining statistical and photometric constraints. *IJCV 2019*.