

Point2Pose: Occlusion-Recovering 6D Pose Tracking and 3D Reconstruction for Multiple Unknown Objects via 2D Point Trackers

Tzu-Yuan Lin¹, Ho Jae Lee¹, Kevin Doherty^{2§}, Yonghyeon Lee¹, and Sangbae Kim¹

¹ Massachusetts Institute of Technology, USA
{tzuyuan, hjlee201, yhl, sangbae}@mit.edu

² Boston Dynamics, USA
kdoherty@mit.edu

Abstract. We present *Point2Pose*, a model-free method for causal 6D pose tracking of multiple rigid objects from monocular RGB-D video. Initialized only from sparse image points on the objects, our approach tracks multiple unseen objects without requiring object CAD models or category priors. Point2Pose leverages a 2D point tracker to obtain long-range correspondences, enabling instant recovery after complete occlusion. Simultaneously, the system incrementally reconstructs an on-line Truncated Signed Distance Function (TSDF) representation of the tracked targets. Alongside the method, we introduce a new multi-object tracking dataset comprising both simulation and real-world sequences, with motion-capture ground truth for evaluation. Experiments show that Point2Pose trades some single-object pose accuracy for broader model-free tracking capabilities, including multi-object tracking and recovery from complete occlusion. Project page: <https://point2pose.github.io/>.

Keywords: Multi-object 6D pose tracking · Shape reconstruction

1 Introduction

Estimating the 6D pose of rigid objects from visual observations is a core problem in robotics and augmented reality. Reliable pose tracking enables agents to plan, interact with, and manipulate objects. While recent progress in object pose estimation and tracking has been substantial, many high-performing systems assume access to object CAD models [12, 32], category-level priors [26], or require multi-view reconstruction before tracking [32]. These assumptions limit deployment in open-world settings, where robots must interact with previously unseen objects without pre-built geometric models.

² § Work was conducted in personal time and independently of the author’s affiliated organization.

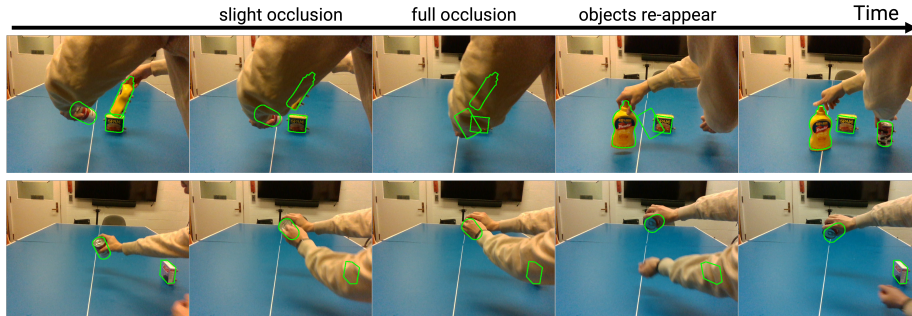


Fig. 1: Point2Pose tracks multiple unknown objects through severe hand-object occlusion and recovers their poses when they reappear, without requiring object CAD models.

Many existing model-free trackers [29, 31] are primarily designed for single-object tracking and often struggle to recover from complete occlusions. This limitation becomes particularly pronounced in egocentric settings involving multi-object manipulation, where objects may undergo full occlusion due to robot-arm motion, inter-object crossovers, or temporary camera motion away from the workspace. In such cases, tracking can fail entirely, as these methods typically rely on inter-frame feature matching and require brittle relocalization mechanisms.

To address these challenges, we present Point2Pose, a model-free method for causal 6D pose tracking of multiple unknown rigid objects from a single RGB-D video stream. Point2Pose jointly tracks object poses online while incrementally reconstructing object-centric 3D models for each tracked object. Our key idea is to use a long-range 2D point tracker as a persistent data-association module, which eliminates the need for expensive and unreliable long-horizon feature matching. By lifting these tracked 2D points using depth data, we construct object-centric 3D keypoint maps and recover poses through map-based registration. In addition, this design naturally supports simultaneous multi-object tracking through aggregated object-specific query sets and enables immediate recovery after complete occlusion when objects reappear, without requiring a separate heavyweight and potentially brittle relocalization stage.

Furthermore, we introduce YCBMultiTrack, a new dataset for dynamic multi-object RGB-D pose tracking with both synthetic and real-world sequences. Existing benchmarks such as HO3D [11] and YCB-InEOAT [30] focus on single-object manipulation, while YCB-Video [34] mainly contains static multi-object scenes and egocentric datasets such as HOT3D [1] rely on multi-view RGB streams without dense depth. In contrast, YCBMultiTrack captures dynamic multi-object motion with inter-object occlusions and provides ground-truth object poses from simulation or motion capture. In summary, our main contributions include:

- We propose **Point2Pose**, a model-free method for 6D pose tracking of multiple unknown rigid objects from RGB-D video, using long-range 2D point tracks for persistent data association and recovery after complete occlusion.

- We introduce **YCBMultiTrack**, a new dataset for dynamic multi-object RGB-D pose tracking with synthetic and real-world sequences.
- Experiments on public benchmarks and **YCBMultiTrack** show that Point2Pose extends model-free 6D tracking to challenging multi-object scenarios with complete occlusion and object re-entry.
- Code and dataset will be released on the project page.

2 Related Work

6D object pose estimation and tracking. 6-DoF object pose estimation and tracking infer the 3D position and orientation of a target object, expressed relative to a chosen reference frame (e.g., the CAD canonical frame, the object pose in the first frame, or the camera frame). Pose estimation compares the current observation to a reference model and estimates the object’s pose in the current frame. It typically relies on a reference model, either an existing CAD model, pre-trained on selected objects [12], or one reconstructed from multi-view images [32]. When performed in real time, pose estimation operates independently at each frame, avoiding drift accumulation and enabling natural recovery from abrupt motion or tracking failure. However, it does not enforce temporal consistency and may result in frame-to-frame jitter. In contrast, pose tracking leverages temporal coherence to propagate and refine pose estimates over time, maintaining a dynamically consistent estimate of the object’s motion [5]. It can be formulated as either model-based or model-free approaches [29]. By exploiting motion continuity, pose tracking achieves smooth and computationally efficient updates, but at the risk of drift accumulation and reduced robustness under prolonged occlusions. Several methods, e.g., MegaPose [20], and FoundationPose [32], can perform both global pose estimation and tracking using separate coarse estimation and local refinement steps. These methods can use the (faster) local refinement step to perform frame-to-frame tracking given an initial estimate. However, these methods struggle to regain tracking after occlusion, and data association is a challenge in the multi-object setting. CosyPose [19] and KMOPS [33] are able to provide pose estimates for multiple static objects in a scene, but do not consider the problem of persistent temporal tracking. In contrast, our method is designed to track multiple objects that are dynamically moving, maintaining precise 6D trajectories even through complex inter-object interactions and temporary disappearances.

Simultaneous tracking and reconstruction. When an object CAD model is not available *a priori*, tracking methods must infer object pose from online observations, often while building or updating an object representation. BundleTrack [29] pioneered this direction by using a graph-based optimization that segments the object and tracks it via temporal feature matching. However, its reliance on short-term frame-to-frame correspondences makes it susceptible to drift and failure during total occlusions. BundleSDF [31] improves upon this by using a neural implicit field to represent the object, enabling joint optimization of pose and shape. However, its neural-field optimization is computationally

expensive, making multi-object tracking difficult, and recovery after prolonged complete occlusion remains challenging. Recent advancements have explored alternative representations, such as 6DOPE-GS [16], which leverages 3D Gaussian Splatting for real-time tracking and reconstruction. While Gaussian Splatting offers impressive rendering speeds and detail, these methods still struggle with re-localization after a target object completely leaves and re-enters the camera view. In this work, we bridge these gaps by utilizing a long-range 2D point tracker as the primary data-association engine. Unlike previous methods that rely on local descriptors or dense volumetric updates, our approach uses persistent point queries to maintain identity across long temporal gaps. This allows our method to reconstruct object-centric models for multiple items simultaneously and achieve immediate re-localization after complete occlusion.

Object tracking datasets. H03D [11] and YCBInEOAT [30] are well-established for evaluating pose tracking methods but are limited to single-object manipulation scenarios. Datasets that incorporate multiple objects, such as YCB-Video [34], predominantly feature static scenes where only the camera is in motion, failing to capture the complexities of independent object trajectories and inter-object occlusions. Recent large-scale egocentric benchmarks like HOT3D [1] capture multi-object interactions but rely entirely on multi-view RGB and monochrome streams, lacking the dense depth information. To address this gap, YCBMultiTrack is designed specifically for multi-object RGB-D pose tracking under dynamic motion and occasional complete occlusions, and includes both synthetic and real sequences with motion-capture object pose annotations.

3 Methodology

Our goal is to track the 6D poses of multiple rigid objects and simultaneously reconstruct their 3D meshes from a stream of monocular RGB-D images. We assume a fixed camera frame C rigidly attached to the world and denote an RGB-D video by $\{\mathcal{F}_t\}_{t=0}^T$, where $\mathcal{F}_t = (I_t, D_t)$ contains an RGB image $I_t \in \mathbb{R}^{3 \times H \times W}$ and a depth image $D_t \in \mathbb{R}^{H \times W}$. The rigid transformation that maps 3D points from frame A to frame B is represented by $T_A^B \in \text{SE}(3)$, which can also be interpreted as the pose of frame A expressed in frame B .

For each object i , we rigidly attach a frame O_i to the object, which we refer to as the *object frame*. We denote the pose of object frame at time t expressed in C as $T_{O_{i,t}}^C$, and we initialize it to coincide with the camera frame C , i.e., $T_{O_{i,0}}^C = I_{4 \times 4}$. Then, multi-object tracking reduces to estimating $T_{O_{i,t}}^C = T_{O_{i,0}}^C T_{O_{i,t}}^{O_{i,0}}$.

An overview of the proposed method is shown in Fig. 2. Given a few user-provided points on the objects, segmentation masks for each object are obtained using Segment Anything 2 [27]. Given these masks, we sample object-specific 2D points and track them over time using a long-range point tracker, such as Track Any Points [7, 8] or CoTracker [17, 18]. The tracked points are lifted to 3D using depth measurements and transformed into the object frame to form an object-centric keypoint map, which we simply refer to as the map. As the object rotates

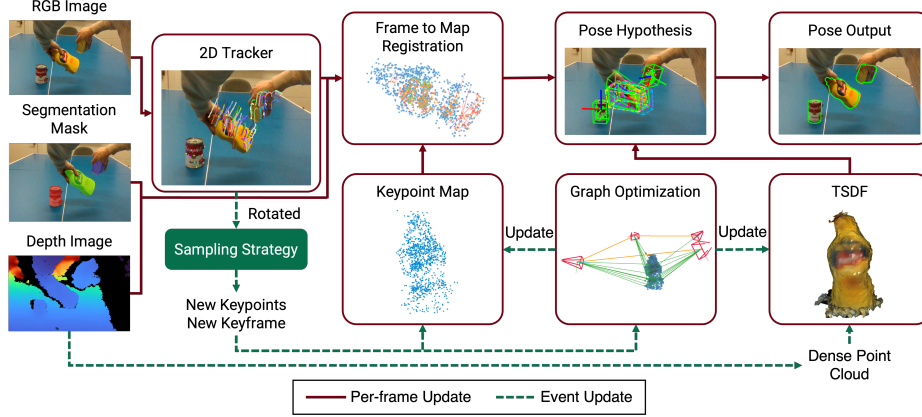


Fig. 2: Overview of the Point2Pose framework. Starting from RGB-D frames and object masks, sparse object keypoints are tracked over time and used to register each frame to an object-centric map. The segmented depth observations are fused into TSDF maps for reconstruction and pose refinement, while pose-graph optimization maintains consistent trajectories across keyframes.

and new surfaces become visible, additional points are sampled and added to expand the map over time. Frames where new points are sampled are referred to as keyframes. Using the natural correspondences from tracking, object poses are estimated by solving a registration problem between the current keypoint observations and the map, which we refer to as *frame-to-map registration*. Meanwhile, we perform online graph optimization to improve temporal stability and update a Truncated Signed Distance Function (TSDF) representation of the mesh model online, which is used to refine pose estimation among the pose hypotheses.

3.1 2D Point Tracker

Given an RGB video $\{I_t\}_{t=0}^T$ and a set of query points, where each query point is defined as $q = (t_q, u_q)$ with t_q the query frame index (i.e., the frame in which the point is queried) and $u_q \in \mathbb{R}^2$ the query pixel location in I_{t_q} , a 2D point tracker estimates the corresponding pixel location in I_t , denoted $u_t \in \mathbb{R}^2$. In addition, the tracker outputs a binary visibility indicator $\delta_t \in \{0, 1\}$ and an uncertainty score $\sigma_t \in [0, 1]$ for each prediction.

In our multi-object setting, we maintain an object-specific query set $\mathcal{Q}_i$ for each object i , while tracking all query points jointly in a single tracker pass. Therefore, the tracker serves as a long-range data-association module, producing temporally consistent correspondences over long time horizons, bypassing the need for explicit feature matching between consecutive frames. This property enables instant recovery when the object reappears after complete occlusion. Since tracking is performed at the point level without object-specific setup, the

same mechanism naturally supports multi-object tracking by operating on the aggregated query set.

3.2 Sampling Strategy and Keypoints Map

When the initially tracked points become occluded or rotate out of view, additional points must be sampled to maintain tracking. These points should be chosen to jointly maximize trackability and spatial diversity to reduce the risk of degenerate pose estimation. To this end, we detect candidate keypoints using SuperPoint [6] within the segmented mask, and use the detector confidence score as a proxy for trackability. Given a set of candidate points $\{(p_k, s_k)\}_{k=1}^{N_k}$, where p_k is the pixel location and $s_k \in [0, 1]$ is the confidence score from SuperPoint, we iteratively select $K \leq N_k$ points using a greedy approach. Let $\mathcal{K}$ denote the set of all keypoints already stored in the map. At each step, we choose the candidate that maximizes the following objective function, which balances trackability and spatial spread:

$$J_k = \lambda s_k + (1 - \lambda) \min\left(\frac{d_k}{r_{ideal}}, 1\right) - \beta \rho(d_k), \quad (1)$$

where d_k is the minimum distance between the candidate point p_k and the union of the previously selected point and $\mathcal{K}$, r_{ideal} is a parameter controlling the ideal spacing radius, and $\lambda \in [0, 1]$ is a weighting parameter. We additionally include a clustering cost $\rho(d_k) = \max(0, \frac{r_{min} - d_k}{r_{min}})$ that penalizes candidates that fall below a minimum acceptable separation distance r_{min} weighted by β . Intuitively, the first term rewards trackability, the second term encourages the selected points to maintain an ideal spatial spread, and the final term strictly penalizes candidates that are too close to existing tracks.

The newly selected K keypoints are then lifted to 3D using the camera matrix and the depth measurements. These points are subsequently transformed into the object frame using the current estimated pose. Rather than immediately integrating these new keypoints into the map, they are initialized in a pending state to mitigate the impact of depth noise and incorrect pose estimate. A pending point is promoted to a keypoint only if it passes a multi-frame verification process. Each pending point receives a score of +1 or +0 at every time step based on the new observation. If the accumulated score exceeds a threshold within a fixed time window, the point is promoted to a keypoint and added to the map. Further details are provided in the Appendix.

3.3 Frame-to-Map Registration for Pose Estimation

The point tracker provides the 2D pixel location of each keypoint in the current camera frame along with an estimate of its visibility. By filtering out non-visible keypoints and back-projecting the remaining points using depth, we obtain 3D keypoint observations in the current camera frame and establish correspondences

with the keypoint map. This allows us to recover the pose of the object by solving a corresponded point cloud registration problem:

$$T_{O_{i,t}}^C = \arg \min_{T \in SE(3)} \sum_{n=1}^N \|\tilde{p}_n - Tp_n\|^2, \quad (2)$$

where $\{(\tilde{p}_n, p_n)\}_{n=1}^N$ denotes the set of correspondences between the 3D keypoint observations $\tilde{p}_n$ in the camera frame and their corresponding points p_n in the map, where N is the number of keypoint observations in the current frame. This least-squares objective can be solved analytically using a Singular Value Decomposition (SVD) approach.

However, because 2D point tracks often contain a high ratio of outliers due to severe occlusions or repetitive local textures, solving (2) directly across all putative matches can yield catastrophic pose errors. To address this, we propose a multi-hypothesis registration strategy designed to explicitly handle geometric degeneracies and perceptual aliasing. For instance, consider a mug rotating around its vertical axis: 2D point trackers may incorrectly predict nearly static trajectories for keypoints on the symmetric cylindrical body, while only a small subset of points on the handle reflects the true rotational motion. In such cases, single-hypothesis outlier-robust methods such as RANSAC can be dominated by the more numerous but incorrect correspondences on the body. By sequentially generating a diverse set of pose hypotheses, our approach preserves the valid transformation among the candidates.

Multi-Hypothesis Pose Generation and Selection. For each object frame O , we generate a set of pose hypotheses $\mathcal{H} = \{(T_O^C)_h\}_{h=1}^H$ using a sequential RANSAC scheme paired with SVD. In each iteration, standard RANSAC identifies the candidate transformation T_O^C that maximizes the number of geometric inliers, after which the corresponding consensus set is removed from the pool of correspondences. We repeat this procedure on the remaining points. This greedy extraction strategy ensures that $\mathcal{H}$ captures not only the dominant motion but also alternative viable poses, preventing the system from prematurely committing to a spatially clustered set of outlier tracks.

We select the best pose hypothesis by evaluating its geometric consistency with the TSDF, whose construction is described later in the section. Although the hypotheses are generated from sparse tracked keypoints, the final selection uses dense 3D observations extracted from the current masked depth image. For each pose hypothesis, the dense observed points are transformed into the object frame, and the absolute TSDF value of the transformed points is computed. The hypothesis with the minimum TSDF score is selected.

SDF Refinement. To further improve the registration accuracy of the selected hypothesis $\bar{T}_O^C$, we perform an iterative pose refinement step that minimizes the dense point-to-implicit-surface distance induced by the TSDF. Given a dense point cloud $\mathcal{P}_{\text{cur}}$ extracted from the current depth image, we formulate a robust nonlinear least-squares problem that drives the transformed observations onto the zero-level isosurface of the TSDF volume $\Phi(\cdot)$.

Let $T \in \text{SE}(3)$ denote the object pose. Starting from the initial estimate $T_0 = \bar{T}_O^C$, we iteratively solve for a pose increment $D \in \text{SE}(3)$ by minimizing

$$D^* = \arg \min_{D \in \text{SE}(3)} \sum_{p \in \mathcal{P}_{\text{cur}}} \rho_H \left(\Phi(T_i^{-1} D^{-1} p)^2 \right), \quad (3)$$

where $\rho_H(\cdot)$ denotes the Huber loss [15]. After each iteration, the pose estimate is updated as $T_{i+1} = D^* T_i$ for $i = 0, \dots, L$. The spatial gradients of the TSDF are computed numerically, and the optimal transformation D^* is obtained using the Levenberg–Marquardt algorithm [21, 23].

3.4 Graph Optimization

To maintain a globally consistent 3D map and correct for accumulated drift over time, for each object, we jointly optimize the keyframe poses and the keypoints using an online factor graph formulation. This optimization is performed whenever a new keyframe is created.

For factor-graph optimization, we adopt the equivalent perspective of a fixed object observed by a moving virtual camera, which yields a standard landmark observation model. As a result, the factor graph is parameterized using inverse pose variables $X_m := T_C^O(m)$. Let $\mathcal{X} = \{X_0, \dots, X_{M_i}\}$ be the set of inverse keyframe poses of a given object, and $\mathcal{P} = \{p_0, \dots, p_{N_i}\}$ denote the 3D coordinates of the keypoints in the object frame. We solve for the optimal configurations for $\mathcal{X}$ and $\mathcal{P}$ by minimizing the following nonlinear least-squares objective function:

$$\mathcal{L}(\mathcal{X}, \mathcal{P}) = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{obs}}. \quad (4)$$

For simplicity, we will drop the object index i .

Prior Loss. To prevent gauge ambiguity, the prior loss anchors the first keyframe pose to the identity:

$$\mathcal{L}_{\text{prior}} = \|\log(X_0)^\vee\|_{\Sigma_{\text{prior}}}^2, \quad (5)$$

where $\log(\cdot)^\vee$ denotes the vectorized Lie algebra representation of the pose in $\mathfrak{se}(3)$ obtained via the logarithm map from $\text{SE}(3)$, and $\|e\|_\Sigma^2 = e^\top \Sigma^{-1} e$ denotes the Mahalanobis norm.

Pose Consistency Loss. The pose consistency loss penalizes deviations of keyframe poses from the frame-to-map registration results. Let $\tilde{X}_m^{m-1}$ denote the estimated relative pose between two consecutive keyframes obtained from the registration. The pose consistency loss is defined using a relative motion constraint rather than an absolute pose estimate and is formulated as

$$\mathcal{L}_{\text{reg}} = \sum_{m=1}^M \rho_H \left(\left\| \log \left((\tilde{X}_m^{m-1})^{-1} X_{m-1}^{-1} X_m \right)^\vee \right\|_{\Sigma_{\text{reg}, m}}^2 \right), \quad (6)$$

where $\Sigma_{\text{reg}, m}$ is the registration covariance matrix, which may vary between keyframes.

Observation Loss. Under the inverse pose parameterization, each keyframe pose X_m acts as a virtual camera observing static keypoints p_n defined in the object frame. Let Ω denote the set of keypoint observations, where $(m, n) \in \Omega$ indicates that keypoint p_n is visible in keyframe m with a valid depth observation. Under this parameterization, $X_m^{-1}p_n$ represents the keypoint expressed in the camera frame of keyframe m . Let $\tilde{z}_{m,n} \in \mathbb{R}^3$ denote the corresponding 3D observation expressed in the camera frame. The observation loss is defined as the discrepancy between the predicted keypoint location $X_m^{-1}p_n$ and the observed 3D point $\tilde{z}_{m,n}$ for $(m, n) \in \Omega$.

To measure this discrepancy, we represent observations using a bearing-range representation (spherical coordinates) and define the loss as the sum of directional and radial errors. Let $\phi: \mathbb{R}^3 \rightarrow \mathbb{R}^4$ be a mapping that converts a 3D point p to its bearing-range representation, $\phi(p) = \left[\frac{p}{\|p\|}, \|p\| \right]$, where $\frac{p}{\|p\|}$ denotes the unit direction (bearing) and $\|p\|$ denotes the radial distance. The observation loss is then defined as

$$\mathcal{L}_{\text{obs}} = \sum_{(m,n) \in \Omega} \rho_H(d^2(\phi(\tilde{z}_{m,n}), \phi(X_m^{-1}p_n))), \quad (7)$$

where $d^2((u_1, r_1), (u_2, r_2))$ denotes the summation of the angle between u_1 and u_2 and $|r_1 - r_2|$. We refer the reader to the Appendix for detailed information.

Solving the Optimization. We implement and solve this nonlinear least-squares problem using the Levenberg-Marquardt optimizer [21, 23] provided by the GTSAM library [4, 9]. At each keyframe insertion, the factor graph is updated and re-optimized to obtain refined inverse keyframe poses $\mathcal{X}^*$ and keypoint coordinates $\mathcal{P}^*$. The optimized graph maintains a globally consistent keyframe-keypoint map, resulting in a more consistent pose tracking result.

3.5 3D Reconstruction

With the globally consistent object trajectory from the factor graph optimization, we aggregate the segmented RGB-D observations into a dense 3D model. We represent the target object using an object-centric volumetric grid defined in the object frame and fuse the depth measurements via a Truncated Signed Distance Function (TSDF) [3], where we follow the approach in [25] for projective TSDF update. The TSDF is updated online whenever a new keyframe is added. For final mesh extraction, we use the marching cubes approach [22] and apply a filter to remove the disconnected component introduced by depth noise. Beyond reconstruction, the fused TSDF also serves as a dense geometric representation for pose-hypothesis selection and refinement.

4 Experiments

We evaluate Point2Pose on two public RGB-D pose-tracking benchmarks and our newly introduced YCBMultiTrack dataset. We compare against BundleSDF [31], which performs tracking and reconstruction without object CAD models, and FoundationPose [32], which is evaluated with provided object CAD models.

4.1 Datasets

Public benchmarks. We evaluate on two standard RGB-D pose-tracking benchmarks: H03D [11] and YCBInEOAT [30]. Following BundleSDF [31], we use the H03D_v3 evaluation split, which contains 13 hand-object interaction sequences with 4 objects. We also evaluate on 9 YCBInEOAT sequences with 5 YCB objects manipulated by a dual-arm robot, where objects are generally smaller in the image, and ground-truth poses are manually annotated.

YCBMultiTrack. YCBMultiTrack is built from a subset of the YCB object [2] and contains both synthetic and real-world RGB-D sequences. The two subsets serve complementary evaluation purposes. The synthetic subset provides controlled multi-object tracking sequences with exact ground-truth poses from simulation, enabling systematic evaluation of multiple simultaneously tracked objects. The real-world subset focuses on more challenging interaction scenarios, including severe occlusion, complete object disappearance, and object re-entry, with ground-truth poses obtained from a motion-capture system. Together, these sequences complement existing object tracking benchmarks by covering both controlled multi-object evaluation and real-world occlusion-heavy tracking. Each sequence includes RGB-D images, SAM2 [27] per-object segmentation masks, object canonical mesh poses in the camera frame $T_{M_i}^C(t)$ for each object i , camera intrinsics, and visibility annotations indicating whether each object is visible or fully occluded.

1. *Synthetic.* We use seven objects from the YCB dataset and generate sequences containing one single-object and three two-object scenarios. Example data are shown in Fig. 3. The objects move along either linear or circular trajectories to create diverse motion patterns and partial occlusion conditions. We simulate the motions and render RGB-D observations in Isaac Lab [24], which provides ground-truth depth and object poses. This allows us to evaluate tracking algorithms in a controlled setting with realistic object geometry and known ground truth.

2. *Real-World.* We use a total of five objects from the YCB dataset and generate sequences containing five single-object, four two-object, and two three-object scenarios. Example sequences are shown in Fig. 1. We use an Intel RealSense D435i camera to collect RGB-D data. To obtain near ground-truth object pose labels, we use an OptiTrack motion capture system. Specifically, we attach four to five markers to each object and record their 6D poses during the sequence. The camera extrinsic matrix is calibrated using an AprilTag [28] with known ground-truth pose. After optimization-based time synchronization, we obtain aligned RGB-D frames and motion capture measurements for accurate pose labeling.

Evaluation metrics. Following prior work [13, 29, 31], we evaluate pose tracking with the area under the curve (AUC) of ADD and ADD-S, computed using the ground-truth mesh. ADD measures exact pose error and ADD-S accounts for object symmetries via closest-point matching. We evaluate reconstruction quality using Chamfer distance between the reconstructed and ground-truth meshes.

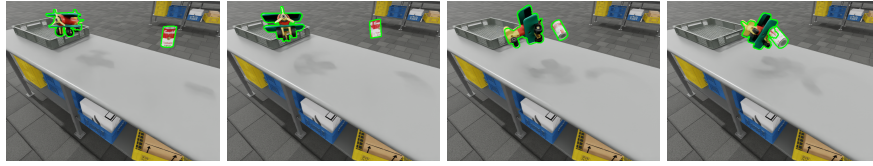


Fig. 3: Example from the YCBMultiTrack-Synthetic dataset: a two-object sequence (toy airplane and tomato soup can) with circular motion and mutual occlusions.

4.2 Results on HO3D and YCBInEOAT

Table 1 reports results on the HO3D benchmark, and Table 2 reports results on the YCBInEOAT dataset. FoundationPose achieves strong pose tracking results, but it uses additional object-level information not assumed by BundleSDF or Point2Pose. Among methods without object CAD models, Point2Pose achieves comparable ADD-S AUC to BundleSDF on HO3D, with a higher mean ADD-S AUC, but lower ADD AUC and Chamfer Distance. On YCBInEOAT, BundleSDF achieves higher mean ADD-S and ADD AUC.

We observe lower ADD AUC on a subset of sequences, such as **AP12** in HO3D and **bleach v1** in YCBInEOAT. These cases contain largely textureless visible object surfaces, making long-range point tracking more difficult. As shown in Fig. 4 (a), **AP12** exposes weaker texture than **AP14**, which contains clearer visual features such as white labels and a cap. This results in less stable 2D point tracks and reduced registration accuracy. A similar failure mode appears in the YCBInEOAT bleach sequences: in **v1**, only a small textured region is visible, which can induce geometric degeneracies in the tracked points. Nevertheless, Point2Pose still achieves reasonable ADD-S AUC in these edge cases, suggesting that the estimated poses often remain coarsely aligned even when orientation-sensitive ADD accuracy degrades.

To better understand the effect of object texture, we further analyze how visible texture influences point tracking and downstream pose tracking. We use Sobel gradient magnitude [10] as a quantitative proxy for local image texture. We evaluate 2D tracker accuracy by projecting object keypoints with ground-truth poses and comparing them with tracker predictions. Figure 4 (b) compares sequences containing the same pitcher object under different viewpoints, which expose different amounts of visible texture. We observe that lower-texture views generally lead to larger 2D point-tracking errors and lower pose-tracking accuracy. This suggests that low-texture views affect Point2Pose primarily by reducing the reliability of long-range 2D correspondences, which then degrades frame-to-map registration. Since our pipeline is modular with respect to the point tracker, future advances in texture-robust point tracking could directly improve this failure mode.

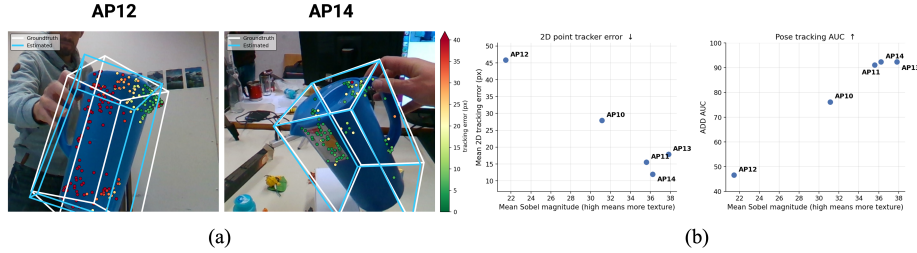


Fig. 4: Texture analysis on H03D pitcher sequences. (a) Comparison of a low-texture view (AP12) and a view with stronger local texture (AP14). White and cyan boxes show ground-truth and estimated poses. In AP12, many high-error tracks occur on largely textureless surfaces. In contrast, the visible labels and cap in AP14 provide stronger local texture, leading to more reliable low-error tracks, resulting in better pose alignment. (b) Mean Sobel gradient magnitude is used as a proxy for visible texture. Sequences with weaker texture generally exhibit larger 2D point-tracking errors and lower ADD AUC, suggesting that reduced correspondence reliability degrades downstream pose tracking.

Table 1: Results on the H03D dataset [11]. ADD-S/ADD AUC are percentages over the 0–0.1 m threshold range. CD denotes Chamfer Distance for reconstruction evaluation. FoundationPose [32] is evaluated with provided CAD models. Bold numbers compare only between CAD-free methods.

Metric	Setting	Method	AP10	AP11	AP12	AP13	AP14	MPM10	MPM11	MPM12	MPM13	MPM14	SB11	SB13	SM1	Mean
ADD-S (%) ↑	CAD	FoundationPose	96.33	95.86	96.29	96.21	96.35	97.80	97.66	98.30	98.21	97.40	97.35	98.12	Fail	97.16
	CAD-free	BundleSDF	95.82	96.07	96.96	96.20	97.17	90.94	96.29	96.16	59.05	97.18	97.06	97.69	96.88	93.34
		Ours	93.71	96.18	84.33	96.27	96.13	95.31	97.08	97.12	96.58	96.42	94.35	97.58	89.15	94.63
ADD (%) ↑	CAD	FoundationPose	91.58	90.50	85.80	89.87	92.83	95.40	95.58	96.75	96.62	94.44	94.92	96.54	Fail	93.40
	CAD-free	BundleSDF	89.69	91.01	94.54	92.80	94.81	78.78	91.69	91.33	33.71	94.16	93.73	95.25	94.23	87.36
		Ours	76.19	91.12	46.64	92.36	92.39	73.99	88.16	93.86	63.19	85.56	85.99	94.90	65.95	80.79
CD (cm) ↓	CAD-free	BundleSDF	0.52	0.58	0.66	0.66	0.98	0.71	0.49	0.48	0.71	0.47	0.45	0.46	0.44	0.58
		Ours	1.20	0.89	2.97	0.96	0.86	0.48	0.28	0.70	0.59	0.74	2.15	0.68	0.80	1.02

4.3 Results on YCBMultiTrack

We present results on the YCBMultiTrack dataset. Since both FoundationPose and BundleSDF support only single-object tracking, we run them sequentially for each object in a multi-object sequence. This setting gives the baselines a slight advantage in computational resources, as it never needs to share memory or compute across objects within a frame.

Table 3a reports performance on the synthetic split, where accurate depth and ground-truth poses are available. Under these controlled conditions, FoundationPose achieves the best performance with the provided object meshes, as objects are largely continuously visible and long-term occlusion recovery is not strongly stressed. Among methods without object CAD models, our method performs consistently better across objects and scenes. Qualitative results of our method for the airplane–tomato sequence are shown in Fig. 3.

Real-world results are shown in Table 3b. In these sequences, objects frequently undergo complete occlusion and may even leave the camera view. Under

Table 2: Results on the YCBInE0AT dataset [30]. ADD-S/ADD AUC are percentages over the 0–0.1 m threshold range. FoundationPose [32] is evaluated with provided CAD models. Bold numbers compare only between CAD-free methods.

Metric	Setting	Method	bleach		cracker		mustard		sugar		tomato	Mean
			v1	v2	v1	v2	v1	v2	v1	v2		
ADD-S (%) ↑	CAD	FoundationPose	96.78	95.59	96.41	93.30	97.70	97.00	97.10	93.79	96.35	96.00
	CAD-free	BundleSDF	94.00	95.47	95.60	91.72	96.32	96.28	96.12	90.56	fail	94.51
		Ours	82.88	93.90	96.37	92.50	95.26	95.65	94.31	87.60	95.52	92.67
ADD (%) ↑	CAD	FoundationPose	92.39	92.48	93.31	88.45	95.41	94.83	94.65	89.24	91.52	92.48
	CAD-free	BundleSDF	87.71	90.99	91.73	84.06	90.44	93.08	90.10	84.41	fail	89.07
		Ours	63.96	87.75	93.56	85.74	89.89	89.90	89.00	78.84	87.35	85.11

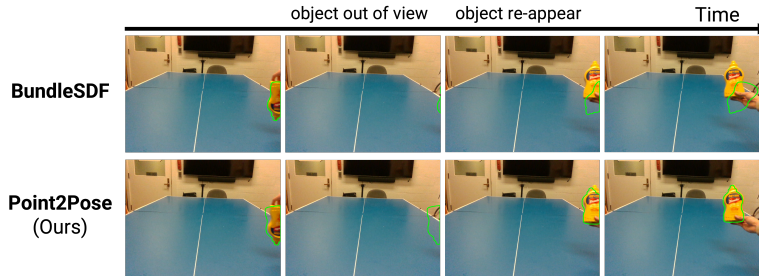


Fig. 5: Results on a fully occluded scene from YCBMultiTrack-Real. After the object becomes completely occluded and later re-enters the scene, our method successfully recovers the object pose, whereas BundleSDF [31] loses track and preserves a stale object orientation.

these conditions, FoundationPose loses track despite using the provided object CAD mesh. BundleSDF exhibits a similar failure mode after complete occlusion. In contrast, our method remains reliable while tracking multiple objects simultaneously. Fig. 1 shows a representative three-object scenario with severe arm-induced occlusions. When the objects reappear, our method quickly recovers their poses. Fig. 5 presents a direct comparison with BundleSDF. While BundleSDF tracks correctly before the object leaves the view, its relocalization after re-entry is biased by the previous estimate used for initialization, resulting in an incorrect pose. These results highlight the key advantage of Point2Pose for simultaneous multi-object tracking and pose recovery after prolonged occlusion.

4.4 Runtime

Point2Pose runs at 2–10 Hz depending on tracker resolution and track count. The 2D point tracker is the main bottleneck, with runtime scaling roughly linearly with the number of tracks. Since the system is modular and our Python implementation is not runtime-optimized, faster point trackers or engineering improvements could further reduce latency.

Table 3: Results on the YCBMultiTrack dataset. ADD-S/ADD AUC are reported as percentages over the 0–0.1 m threshold range. FoundationPose [32], denoted as FP, is evaluated with provided ground-truth object meshes, while BundleSDF [31] and Ours do not use object CAD models. Bold numbers compare only between methods without object CAD models.

(a) YCBMultiTrack-Synthetic										(b) YCBMultiTrack-Real									
Category	Object	ADD-S (%) $\uparrow$			ADD (%) $\uparrow$					Category	Object	ADD-S (%) $\uparrow$			ADD (%) $\uparrow$				
		CAD		CAD-free	CAD		CAD-free					CAD		CAD-free	CAD		CAD-free		
		FP	BundleSDF	Ours	FP	BundleSDF	Ours					FP	BundleSDF	Ours	FP	BundleSDF	Ours		
One object	sugar	98.44	69.80	96.97	97.59	67.18	91.98			One object	tomato	45.10	fail	71.66	32.07	fail	36.60		
	pitcher	98.48	94.53	92.60	98.01	79.15	79.27				mustard	41.42	40.82	90.34	35.55	35.37	78.39		
Two objects	meat	98.57	76.67	97.13	98.03	72.13	93.06			Two objects	pudding	13.86	44.48	90.81	4.21	24.87	73.22		
	sugar	99.07	72.09	96.87	98.84	69.16	92.31				meat	41.23	74.88	88.48	37.84	41.65	76.25		
	scissors	98.95	12.36	41.80	98.41	6.70	39.65				bleach	44.43	61.91	94.80	26.69	43.78	88.80		
	master	97.13	96.24	97.31	94.91	54.20	90.73				tomato	40.00	55.51	95.63	35.75	34.12	89.15		
	airplane	99.01	96.02	88.46	98.60	92.21	55.07				pudding	42.04	fail	95.13	35.96	fail	91.63		
	tomato	97.43	96.17	98.20	96.21	63.66	81.41				tomato	33.71	93.48	87.21	31.75	83.44	44.87		
Mean		98.39	76.74	88.67	97.58	63.05	77.94			Three objects	pudding	22.82	fail	84.99	13.09	fail	70.46		
											mustard	18.22	12.19	95.35	12.11	12.04	91.02		
											meat	97.13	fail	96.49	94.48	fail	92.34		
											mustard	29.01	70.08	91.01	27.08	29.19	81.72		
											meat	47.98	fail	90.02	40.82	fail	68.29		
											tomato	24.46	fail	90.36	19.96	fail	60.96		
										Three objects	bleach	90.92	fail	92.65	83.72	fail	81.89		
											tomato	94.21	94.59	94.05	70.13	79.92	83.67		
											pudding	28.23	fail	72.14	17.59	fail	41.75		
Mean		42.49	62.08	89.43	35.23	42.34	74.17												

Table 4: Component-wise ablation on H03D.

Method	ADD-S (%) $\uparrow$	ADD (%) $\uparrow$
Proposed	95.07	82.76
No multi-hypothesis	89.81	65.52
No SDF refinement	94.40	77.37
No graph optimization	94.56	78.80

4.5 Ablation Studies

Component-Wise Ablation. We evaluate the contribution of the main modules on H03D in Table 4. The no multi-hypothesis setting estimates the pose using a two-step procedure: an initial SVD-based alignment, followed by outlier removal using a fixed threshold and a second SVD using the remaining inliers. In the no SDF refinement setting, the pose from the multi-hypothesis stage is used directly without the SDF refinement step. In the no graph optimization setting, the estimated pose is used without performing graph-based global refinement. Removing the multi-hypothesis module leads to the largest performance degradation, while SDF refinement and graph optimization further improve pose accuracy through local and global refinement. Overall, these results confirm that the three modules are complementary and together enable robust and accurate pose estimation.

Long Range v.s. Short Horizon Tracking. To isolate the role of long-range tracking, we introduce P2P-SH(10), where points are permanently discarded after 10 consecutive invisible frames. All other components are unchanged. Across two occlusion-heavy YCBMultiTrack-Real sequences, Point2Pose recov-

ers 5/5 occlusion events within 30 frames after re-emergence, whereas P2P-SH(10) recovers 0/5 events. On the mustard sequence, Point2Pose achieves 93.79 ADD-S AUC, and 84.36 ADD AUC, while P2P-SH(10) degrades to 38.52, and 31.44, respectively. This ablation directly supports our claim that persistent long-range point identity is essential for recovery from prolonged or complete occlusion.

5 Limitations

The proposed method has several limitations. First, the underlying 2D point tracker relies on sufficiently discriminative image texture to maintain stable correspondences and may become less reliable on textureless or repetitive surfaces. Second, the approach depends on accurate instance segmentation; noisy or incomplete masks may introduce background points that degrade registration. Third, tracking many objects can increase memory usage due to the large number of tracked points, although this can be mitigated through practical engineering choices such as point subsampling or pruning inactive keypoints, as well as improved point-tracking models. Finally, our use of a classical TSDF representation prioritizes simplicity and runtime efficiency, but may yield lower reconstruction fidelity compared to recent learning-based approaches. Exploring modern neural reconstruction methods (e.g., EfficientNeRF [14]) while maintaining online fusion and tight coupling with tracking is an interesting direction for future work.

6 Conclusion

We presented *Point2Pose*, a causal, model-free method for multi-object 6D pose tracking and 3D reconstruction from monocular RGB-D video that can recover from complete occlusion. By leveraging modern 2D point trackers for long-range data association, our method maintains object identity through severe occlusions and enables recovery when objects reappear. We improve robustness under outlier-heavy correspondences with multi-hypothesis frame-to-map registration and TSDF-based hypothesis selection and refinement, while an online factor-graph optimization maintains global consistency and supports object-centric TSDF reconstruction. We also introduced *YCBMultiTrack*, a new synthetic and real-world benchmark for multi-object tracking under severe occlusions. Together, these results show that Point2Pose provides a practical model-free approach to multi-object tracking and reconstruction with recovery from complete occlusion.

Acknowledgements

This work was supported in part by the Technology Innovation Program (RS-2024-00427719) funded by MOTIE, Korea and the KIAT Global Industry Technology Cooperation Center Program (P246800183).

References

1. Banerjee, P., Shkodrani, S., Moulon, P., Hampali, S., Han, S., Zhang, F., Zhang, L., Fountain, J., Miller, E., Basol, S., et al.: Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7061–7071 (2025)
2. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The ycb object and model set: Towards common benchmarks for manipulation research. In: *2015 international conference on advanced robotics (ICAR)*. pp. 510–517. IEEE (2015)
3. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. pp. 303–312 (1996)
4. Dellaert, F., Kaess, M.: *Factor Graphs for Robot Perception*. *Foundations and Trends in Robotics*, Vol. 6 (2017), <http://www.cs.cmu.edu/~kaess/pub/Dellaert17fnt.pdf>
5. Deng, X., Mousavian, A., Xiang, Y., Xia, F., Bretl, T., Fox, D.: Poserbpf: A rao-blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics* **37**(5), 1328–1342 (2021)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 224–236 (2018)
7. Doersch, C., Luc, P., Yang, Y., Gokay, D., Koppula, S., Gupta, A., Heyward, J., Rocco, I., Goroshin, R., Carreira, J., et al.: Bootstap: Bootstrapped training for tracking-any-point. In: *Proceedings of the Asian Conference on Computer Vision*. pp. 3257–3274 (2024)
8. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: TAPIR: Tracking any point with per-frame initialization and temporal refinement. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 10061–10072 (2023)
9. Frank Dellaert and GTSAM Contributors: borglab/gtsam (May 2022). <https://doi.org/10.5281/zenodo.5794541>, <https://github.com/borglab/gtsam>
10. Gonzalez, R.C., Woods, R.E.: *Digital image processing*. Pearson, 4th edn. (2018)
11. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3d annotation of hand and object poses. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 3196–3206 (2020)
12. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11632–11641 (2020)
13. He, Y., Wang, Y., Fan, H., Sun, J., Chen, Q.: Fs6d: Few-shot 6d pose estimation of novel objects. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 6814–6824 (2022)
14. Hu, T., Liu, S., Chen, Y., Shen, T., Jia, J.: Efficientnerf efficient neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 12902–12911 (2022)
15. Huber, P.J.: Robust estimation of a location parameter. In: *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer (1992)
16. Jin, Y., Prasad, V., Jauhri, S., Franzius, M., Chalvatzaki, G.: 6dope-gs: Online 6d object pose estimation using gaussian splatting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 8032–8043 (2025)

17. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Ruppel, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6013–6022 (2025)
18. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Ruppel, C.: Cotracker: It is better to track together. In: European conference on computer vision. pp. 18–35. Springer (2024)
19. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: European conference on computer vision. pp. 574–591. Springer (2020)
20. Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D., Sivic, J.: Megapose: 6d pose estimation of novel objects via render & compare. In: Conference on Robot Learning. pp. 715–725. PMLR (2023)
21. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics* **2**(2), 164–168 (1944)
22. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353. Association for Computing Machinery, New York, NY, United States (1998)
23. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics* **11**(2), 431–441 (1963)
24. Mittal, M., Roth, P., Tigue, J., Richard, A., Zhang, O., Du, P., Serrano-Muñoz, A., Yao, X., Zurbrugg, R., Rudin, N., et al.: Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831* (2025)
25. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011)
26. Ponimatkin, G., Cifka, M., Soucek, T., Fourmy, M., Labbé, Y., Petrik, V., Sivic, J.: 6d object pose tracking in internet videos for robotic manipulation. In: International Conference on Learning Representations. vol. 2025, pp. 32569–32596 (2025)
27. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: International Conference on Learning Representations. vol. 2025, pp. 28085–28128 (2025)
28. Wang, J., Olson, E.: Apriltag 2: Efficient and robust fiducial detection. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4193–4198. IEEE (2016)
29. Wen, B., Bekris, K.: Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8067–8074. IEEE (2021)
30. Wen, B., Mitash, C., Ren, B., Bekris, K.E.: se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Oct 2020). <https://doi.org/10.1109/iros45743.2020.9341314>, <http://dx.doi.org/10.1109/IROS45743.2020.9341314>

31. Wen, B., Tremblay, J., Blukis, V., Tyree, S., Müller, T., Evans, A., Fox, D., Kautz, J., Birchfield, S.: Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 606–617 (2023)
32. Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17868–17879 (2024)
33. Wu, Y.K., Shen, Y., Huang, T., Fang, I., Chen, J.C., et al.: Kmops: Keypoint-driven method for multi-object pose and metric size estimation from stereo images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7730–7739 (2026)
34. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. Robotics: Science and Systems XIV (2018)