BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects

Bowen Wen Jonathan Tremblay Valts Blukis Stephen Tyree Thomas Müller
Alex Evans Dieter Fox Jan Kautz Stan Birchfield
NVIDIA

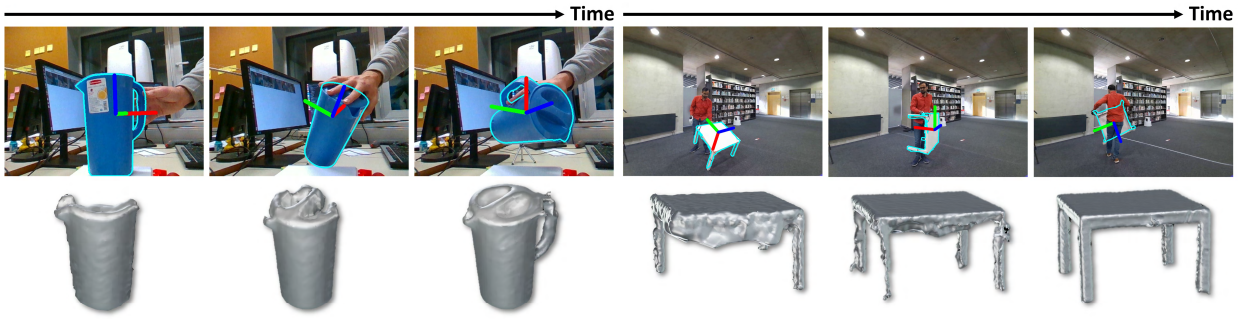

Figure 1. Given a monocular RGBD sequence and 2D object mask (in the first frame only), our method performs causal 6-DoF tracking and 3D reconstruction of an unknown object. Without any prior knowledge of the object or interaction agent, our method generalizes well, handling flat and untextured surfaces, specular highlights, thin structures, severe occlusion, and a variety of interaction agents (human hand / body / robotic arm). The visualized meshes are directly output by the method.

Abstract

We present a near real-time (10Hz) method for 6-DoF tracking of an unknown object from a monocular RGBD video sequence, while simultaneously performing neural 3D reconstruction of the object. Our method works for arbitrary rigid objects, even when visual texture is largely absent. The object is assumed to be segmented in the first frame only. No additional information is required, and no assumption is made about the interaction agent. Key to our method is a Neural Object Field that is learned concurrently with a pose graph optimization process in order to robustly accumulate information into a consistent 3D representation capturing both geometry and appearance. A dynamic pool of posed memory frames is automatically maintained to facilitate communication between these threads. Our approach handles challenging sequences with large pose changes, partial and full occlusion, untextured surfaces, and specular highlights. We show results on HO3D, YCBInEOAT, and BEHAVE datasets, demonstrating that our method significantly outperforms existing approaches. Project page: https://bundlesdf.github.io/

1. Introduction

Two fundamental (and closely related) problems in computer vision are 6-DoF ("degree of freedom") pose tracking and 3D reconstruction of an unknown object from a monocular RGBD video. Solving these problems will unlock a wide range of applications in areas such as augmented reality [34], robotic manipulation [22, 70], learning-from-demonstration [71], and sim-to-real transfer [1, 15].

Prior efforts often consider these two problems separately. For example, neural scene representations have achieved great success in creating high quality 3D object models from real data [3, 40, 44, 59, 68, 81]. These approaches, however, assume known camera poses and/or ground-truth object masks. Furthermore, capturing a static object by a dynamically moving camera prevents full 3D reconstruction (*e.g.*, the bottom of the object is never seen if resting on a table). On the other hand, instance-level 6-DoF object pose estimation and tracking methods often require a textured 3D model of the test object beforehand [24, 28, 66, 72, 73] for pre-training and/or online template matching. While category-level methods enable generalization to new object instances within the same category [7, 27, 62, 67, 74], they struggle with out-of-distribution object instances and unseen object categories.

To overcome these limitations, in this paper we propose to solve these two problems jointly. Our method assumes that the object is rigid, and it requires a 2D object mask in the first frame of the video. Apart from these two requirements, the object can be moved freely throughout the video, even undergoing severe occlusion. Our approach is similar in spirit to prior work in object-level SLAM [35, 36, 50–52, 64, 85], but we relax many common assumptions, allowing us to handle occlusion, specularity, lack of visual texture and geometric cues, and abrupt object motion. Key to our method is an online pose graph optimization process, a concurrent Neural Object Field to reconstruct the 3D shape and appearance, and a memory pool to facilitate communication between the two processes. The

robustness of our method is highlighted in Fig. 1.

Our contributions can be summarized as follows:

- A novel method for causal 6-DoF pose tracking and 3D reconstruction of a novel unknown dynamic object. This method leverages a novel co-design of concurrent tracking and neural reconstruction processes that run online in near real-time while largely reducing tracking drift.
- We introduce a hybrid SDF representation to deal with uncertain free space caused by the unique challenges in a dynamic object-centric setting, such as noisy segmentation and external occlusions from interaction.
- Experiments on three public benchmarks demonstrate state-of-the-art performance against leading methods.

2. Related Work

6-DoF Object Pose Estimation and Tracking. 6-DoF object pose estimation infers the 3D translation and 3D rotation of a target object in the camera's frame. State-of-the-art methods often require instance- or category-level object CAD models for offline training or online template matching [24,25,60,67], which prevents their application to novel unknown objects. Although several recent works [32,45,58] relax the assumption and aim to quickly generalize to novel unseen objects, they still require pre-capturing posed reference views of the test object, which is not assumed in our setting. Aside from single-frame pose estimation, 6-DoF object pose tracking leverages temporal information to estimate per-frame object poses throughout the video. Similar to their single-frame counterparts, these methods make various levels of assumptions, such as training and testing on the same objects [28, 38, 54, 63, 69, 72] or pretraining on the same category of objects [30, 38, 65]. BundleTrack [69] shares the closest setting to ours, generalizing pose tracking instantly to novel unknown objects. Differently, however, our co-design of tracking and reconstruction with a novel neural representation not only results in more robust tracking as validated in experiments (Sec. 4), but also enables an additional shape output, which is not possible with [69].

Simultaneous Localization and Mapping. SLAM solves a similar problem to the one addressed in this work, but focuses on tracking the camera pose w.r.t. a large static environment [41, 56, 61, 85]. Dynamic-SLAM methods usually track dynamic objects by frame-model Iterative Closest Point (ICP) combined with color [33,49,50,77], probabilistic data association [55], or 3D level-set likelihood maximization [48]. Models are simultaneously reconstructed on-the-fly by aggregating the observed RGBD data with the newly tracked pose. In contrast, our method leverages a novel Neural Object Field representation that allows for automatic on-the-fly fusion [10], while dynamically rectifying historically tracked poses to maintain multi-view consistency. We focus on the object-centric setting including dynamic scenarios, in which there is often a lack of texture or geometric cues, and severe occlusions are frequently introduced by the interaction agent—difficulties that rarely happen in traditional SLAM. Compared to static scenes studied in object-level SLAM [35, 36, 51, 52, 64], dynamic interaction also allows observing different faces of the object for more complete 3D reconstruction.

Object Reconstruction. Retrieving a 3D mesh from images has been extensively studied using learning based methods [26, 40, 80]. With recent advances in neural scene representation, high quality 3D models can be reconstructed [3,40,44,59,68,81], though most of these methods assume known camera poses or ground-truth segmentation and often focus on static scenes with rich texture or geometric cues. In particular, [47] presents a semi-automatic method with a similar goal but uses manual object pose annotations to retrieve a textured model of the object. In contrast, our method is fully automatic and operates over the video stream causally. Another line of research leverages human hand or body priors to resolve object scale ambiguity or refine object pose estimations via contact/collision constraints [4,6,16,18,21,23,31,76,79,84]. In contrast, we do not assume specific knowledge of the interaction agent, which allows us to generalize to drastically different forms of interactions and scenarios, ranging from human hand, human body to robot arms, as shown in the experiments. This also eliminates another possible source of error from imperfect human hand/body pose estimation.

3. Approach

An overview of our method is depicted in Fig. 2. Given a monocular RGBD input video, along with a segmentation mask of the object of interest *in the first frame only*, our method tracks the 6-DoF pose of the object through subsequent frames and reconstructs a textured 3D model of the object. All processing is causal (no access to future frames) The object is assumed to be rigid, but no specific amount of texture is required—our method works well with untextured objects. In addition, no instance-level CAD model of the object, nor category-level prior (*e.g.*, training on the same object category beforehand), is needed.

3.1. Coarse Pose Initialization

To provide a good initial guess for the subsequent online pose graph optimization, we compute a coarse object pose estimate $\tilde{\xi}_t \in \mathrm{SE}(3)$ between the current frame $\mathcal{F}_t$ and the previous frame $\mathcal{F}_{t-1}$. First, the object region is segmented in $\mathcal{F}_t$ by leveraging an object-agnostic video segmentation network [8]. This segmentation method was chosen because it does not require any knowledge of the object or the interaction agent (*e.g.*, a human hand), thus allowing our framework to be applied to a wide range of scenarios and objects.

Feature correspondences in RGB between $\mathcal{F}_t$ and $\mathcal{F}_{t-1}$ are established via a transformer-based feature matching network [57], which was pretrained on a large collection

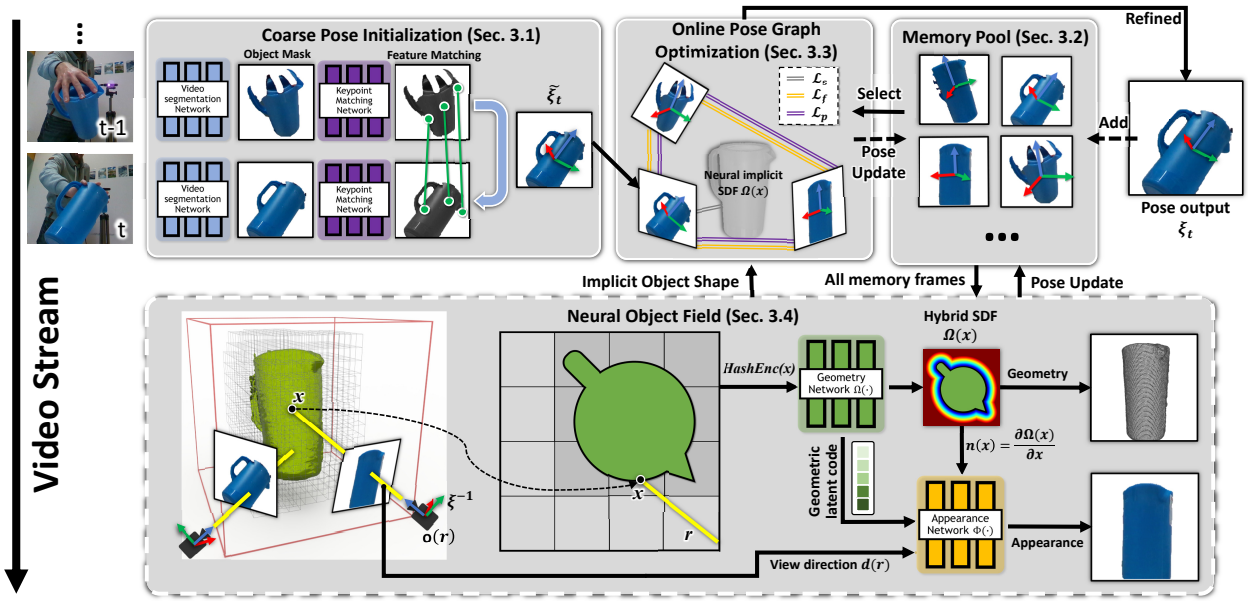

Figure 2. Framework overview. First, features are matched between consecutive segmented images, to obtain a coarse pose estimate (Sec. 3.1). Some of these posed frames are stored in a memory pool, to be used and refined later (Sec. 3.2). A pose graph is dynamically created from a subset of the memory pool (Sec. 3.3); online optimization refines all the poses in the graph jointly with the current pose. These updated poses are then stored back in the memory pool. Finally, all the posed frames in the memory pool are used to learn a Neural Object Field (in a separate thread) that models both geometry and visual texture (Sec. 3.4) of the object, while adjusting their previously estimated poses.

of internet photos [29]. Together with depth, the identified correspondences are filtered by a RANSAC-based pose estimator [11] using least squares [2]. The pose hypothesis that maximizes the number of inliers is then selected as the current frame's coarse pose estimation $\tilde{\xi}_t$.

3.2. Memory Pool

To alleviate catastrophic forgetting, which can cause long-term tracking drift, it is important to retain information about past frames. A common approach exploited by prior work is to fuse each posed observation into an explicit global model [43, 50, 53]. The fused global model is then used to compare against the subsequent new frames for their pose estimation (frame-to-model matching). However, such an approach is too brittle for the challenging scenarios considered in this work, for at least two reasons. First, any imperfections in the pose estimates will be accumulated when fusing into the global model, causing additional errors when estimating the pose of subsequent frames. Such errors frequently occur when there is insufficient texture or geometric cues on the object, or this information is not visible in the frame. Such errors accumulate over time and are irreversible. Second, in the case of long-term complete occlusion, large motion changes make registration between the global model and the reappearing frame observation difficult and suboptimal.

Instead, we introduce a keyframe memory pool $\mathcal{P}$ that stores the most informative historical observations. To build the memory pool, the first frame $\mathcal{F}_0$ is automatically added, thus setting the canonical coordinate system for the novel unknown object. For each new frame, its coarse pose $\tilde{\xi}_t$ is updated by comparing to the existing frames in the memory pool, as described in Sec. 3.3, to yield an updated pose ξ_t. The frame is only added to $\mathcal{P}$ when its viewpoint (described by ξ_t) is deemed to sufficiently enrich the multi-view diversity in the pool while keeping the pool compact.

More specifically, ξ_t is compared with the poses of all existing memory frames in the pool. Since in-plane object rotation does not provide additional information, this comparison takes into account rotational geodesic distance while ignoring rotation around the camera's optical axis. Ignoring this difference allows the system to allocate memory frames more sparsely in the space while maintaining a similar amount of multi-view consistency information. This trick enables jointly optimizing a wider range of poses, compared to previous work (*e.g.*, [69]), when selecting the same number of memory frames to participate in the online pose graph optimization.

3.3. Online Pose Graph Optimization

Given a new frame $\mathcal{F}_t$ with its coarse pose estimation $\tilde{\xi}_t$ (Sec. 3.1), we select a subset of (no more than) K memory frames from the memory pool to participate in online pose graph optimization. The optimized pose corresponding to the new frame becomes the output estimated pose ξ_t. This step is implemented in CUDA for near real-time processing, making it sufficiently fast to be applied to every new frame, thus resulting in more accurate pose estimations as the object is tracked throughout the video.

As described below (Sec. 3.4), the Neural Object Field is also used to assist in this optimization process. Every frame in the memory pool has associated with it a binary flag

$b(\mathcal{F})$ indicating whether the pose of this particular frame has had the benefit of being updated by the Neural Object Field. When a frame is first added to the memory pool, $b(\mathcal{F}) =$ FALSE. This flag remains unchanged through subsequent online updates until the frame's pose has been updated by the Neural Object Field, at which point it is forever set to TRUE.

Concurrent with updating the pose of the new frame $\mathcal{F}_t$, all the poses of the subset of frames selected for the online pose graph optimization are also updated to the memory pool, as long as their flag is set to FALSE. Those frames whose flag is set to TRUE continue to be updated by the more reliable Neural Object Field process, but they cease being modified by the online pose graph optimization.

Selecting Subset of Memory Frames. We constrain the number of memory frames participating in the pose graph optimization to be no more than K for efficiency. Early in the video, when $|\mathcal{P}| \leqslant K$, no selection is needed, and all frames in the memory pool are used. When the size of the memory pool grows to be larger than K, a selection process is applied with the goal of maximizing the multi-view consistency information. Prior efforts select keyframes by exhaustively searching pair-wise feature correspondences and solving a spanning tree [41], which is either too time-consuming for real-time processing, or simply based on a fixed time interval [53], which is less effective in our object-centric setting. Therefore, we propose instead to efficiently select the subset $\mathcal{P}_{pg} \subset \mathcal{P}$ of memory frames by leveraging the current frame's coarse pose estimation $\tilde{\xi}_t$ (obtained in Sec. 3.1). Specifically, for each frame $\mathcal{F}^{(k)}$ in the memory pool, we first compute the point normal map and compute the dot product between these normals and the ray direction in the new frame's camera view to test their visibility. If the point cloud visibility ratio in the new frame $\mathcal{F}_t$ is above a threshold (0.1 for all experiments), we further measure the viewing overlap with $\mathcal{F}_t$ by computing the rotation geodesic distance between $\xi^{(k)}$ and $\tilde{\xi}_t$ while ignoring the in-plane rotation (as described above). Finally we select the K memory frames with the maximum viewing overlap (smallest distance) to participate in the pose graph optimization along with $\mathcal{F}_t$. Therefore, $|\mathcal{P}_{pg}| = K$.

Optimization. In the pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the nodes consist of $\mathcal{F}_t$ and the above selected subset of memory frames: $\mathcal{V} = \mathcal{F}_t \cup \mathcal{P}_{pg}$, so $|\mathcal{V}| = K + 1$. The objective is to find the optimal poses that minimize the total loss of the pose graph:

$$\mathcal{L}_{pg} = w_s \mathcal{L}_s(t) + \sum_{i \in \mathcal{V}, j \in \mathcal{V}, i \neq j} \left[w_f \mathcal{L}_f(i,j) + w_p \mathcal{L}_p(i,j) \right], \quad (1)$$

where $\mathcal{L}_f$ and $\mathcal{L}_p$ are pairwise edge losses [69], and $\mathcal{L}_s$ is an additional unary loss. The scalar factors w_f, w_p, w_s are all set to 1 empirically. The loss

$$\mathcal{L}_f(i,j) = \sum_{(p_m, p_n) \in C_{i,j}} \rho \left(\left\| \xi_i^{-1} p_m - \xi_j^{-1} p_n \right\|_2 \right) \quad (2)$$

measures the Euclidean distance of the RGBD feature correspondences $p_m, p_n \in \mathbb{R}^3$, where ξ_i denotes the object pose in frame $\mathcal{F}^{(i)}$, and ρ is the Huber loss [19] for robustness. The set of correspondences $C_{i,j}$ between frames $\mathcal{F}^{(i)}$ and $\mathcal{F}^{(j)}$ is detected by the same network introduced in Sec. 3.1, where we run batch inference in parallel for efficiency. The loss

$$\mathcal{L}_p(i,j) = \sum_{p \in I_i} \rho \left(\left| n_i(p) \cdot \left(T_{ij}^{-1} \pi_{D_j}^{-1} (\pi_j (T_{ij} p)) - p \right) \right| \right) \quad (3)$$

measures the pixel-wise point-to-plane distance via reprojective association, where $T_{ij} \equiv \xi_j \xi_i^{-1}$ transforms from $\mathcal{F}^{(i)}$ to $\mathcal{F}^{(j)}$, π_j denotes the perspective projection mapping onto image I_j associated with $\mathcal{F}^{(j)}$, $\pi_{D_j}^{-1}$ represents the inverse projection mapping via looking-up the depth image D_j at the pixel location, $n_i(p)$ denotes the normal via looking-up the normal map of $\mathcal{F}^{(i)}$ at pixel location $p \in I_i$ associated. Lastly, the unary loss

$$\mathcal{L}_s(t) = \sum_{p \in I_t} \rho \left(\left| \Omega(\xi_t^{-1} (\pi_D^{-1}(p))) \right| \right) \quad (4)$$

measures the point-wise distance to the neural implicit shape using the current frame, where $\Omega(\cdot)$ denotes the signed distance function from the Neural Object Field as will be discussed in Sec. 3.4. The Neural Object Field weights are frozen in this step. This unary loss is taken into account only after the initial training of the Neural Object Field has converged.

The poses are represented as inversions of camera poses w.r.t. the object, parametrized using Lie Algebra, fixing the coordinate frame of the initial frame as the anchor point. We solve the entire pose graph optimization via the Gauss-Newton algorithm with iterative re-weighting. The optimized pose corresponding to $\mathcal{F}_t$ becomes its updated pose ξ_t. For the rest of the selected memory frames, their optimized poses in the memory pool are also updated to rectify possible errors computed earlier in the video, unless $b(\mathcal{F}) =$ TRUE, as mentioned earlier.

3.4. Neural Object Field

A key to our approach is learning an object-centric neural signed distance field that learns multi-view consistent 3D shape and appearance of the object while adjusting memory frames' poses. It is learned per-video and does not require pre-training in order to generalize to novel unknown objects. This Neural Object Field trains in a separate thread parallel to the online pose tracking. At the start of each training period, the Neural Object Field consumes all the memory frames (along with their poses) from the pool and begins learning. When training converges, the optimized poses are updated to the memory pool to aid subsequent online pose graph optimization, which fetches these up-

dated memory frame poses each time to alleviate tracking drift. The learned SDF is also updated to the subsequent online pose graph to compute the unary loss $\mathcal{L}_s$ described in Sec. 3.3. The Neural Object Field training process is then repeated by grabbing new memory frames from the pool.

Object Field Representation. Inspired by [82], we represent the object by two functions. First, the geometry function $\Omega : x \mapsto s$ takes as input a 3D point $x \in \mathbb{R}^3$ and outputs a signed distance value $s \in \mathbb{R}$. Second, the appearance function $\Phi : (f_{\Omega(x)}, n, d) \mapsto c$ takes the intermediate feature vector $f_{\Omega(x)} \in \mathbb{R}^3$ from the geometry network, a point normal $n \in \mathbb{R}^3$, and a view direction $d \in \mathbb{R}^3$, and outputs the color $c \in \mathbb{R}^3_+$. In practice, we apply multi-resolution hash encoding [39] to x before forwarding to the network. The normal of a point in the object field can be derived by taking the first-order derivative on the signed distance field: $n(x) = \frac{\partial \Omega(x)}{\partial x}$, which we implement by leveraging automatic differentiation in PyTorch [46]. For both directions n and d, we embed them by a fixed set of low-order spherical harmonic coefficients (order 2 in our case) to prevent overfitting that could discourage the object pose update (represented as inversion of camera poses w.r.t. the object, as mentioned above), in particular the rotations.

The implicit object surface is obtained by taking the zero level set of the signed distance field: $S = \{x \in \mathbb{R}^3 \mid \Omega(x) = 0\}$. The SDF object representation Ω has two major benefits compared to [37] in our setting. First, when combined with our efficient ray sampling with depth guided truncation (described below), it enables the training to converge quickly within seconds for online tracking. Second, implicit regularization guided by the normals encourages smooth and accurate surface extraction. This not only provides a satisfactory object shape reconstruction as one of our final goals, but also in return provides more accurate frame-to-model loss $\mathcal{L}_s$ for the online pose graph optimization.

Rendering. Given the object pose ξ of a memory frame, an image is rendered by emitting rays through the pixels. 3D points are sampled at different locations along the ray:

$$x_i(r) = o(r) + t_i d(r), \tag{5}$$

where $o(r)$ and $d(r)$ are the ray origin (camera focal point) and ray direction, respectively, both of which depend on ξ; and $t_i \in \mathbb{R}_+$ governs the position along the ray.

The color c of a ray r is integrated by near-surface regions:

$$c(r) = \int_{z(r)-\lambda}^{z(r)+0.5\lambda} w(x_i)\Phi(f_{\Omega(x_i)}, n(x_i), d(x_i))\, dt, \tag{6}$$

$$w(x_i) = \frac{1}{1+e^{-\alpha\Omega(x_i)}} \frac{1}{1+e^{\alpha\Omega(x_i)}}, \tag{7}$$

where $w(x_i)$ is the bell-shaped probability density function [68] that depends on the distance from the point to the implicit object surface, *i.e.*, the signed distance $\Omega(x_i)$. α (set to a constant) adjusts the softness of the probability density distribution. The probability reaches a local maximum at the surface intersection. $z(r)$ is the depth value of the ray from the depth image. λ is the truncation distance. In Eq. (6), we ignore the contribution from empty space that is more than λ away from the surface to reduce over-fitting from the empty space in the neural field in order to improve pose updates. We then only integrate up to a 0.5λ penetrating distance to model self-occlusion [68]. An alternative to directly using the depth reading $z(r)$ to guide the integration would be to infer the zero-crossing surface from $\Omega(x_i)$. However, we found this requires denser point sampling and slower training convergence compared to using the depth.

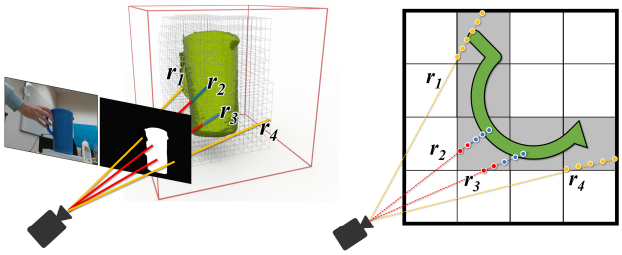

Figure 3. **Left:** Octree-voxel representation for efficient ray tracing, using the predicted binary mask from the video segmentation network (Sec. 3.1), which contains errors. Rays can land inside the mask (shown as red) or outside (yellow). **Right:** 2D top-down illustration of the neural volume and point sampling along the rays with hybrid SDF modeling. Blue samples are near the surface.

Efficient Hierarchical Ray Sampling. For efficient rendering, we construct an Octree representation [12] before training by naively merging the point clouds of the posed memory frames. We then perform hierarchical sampling along the rays. Specifically, we first uniformly sample N points bounded by the occupancy voxels (gray boxes in Fig. 3), terminating at $z(r)+0.5\lambda$. A custom CUDA kernel was implemented to skip the sampling of intermediate unoccupied voxels. Additional samples are allocated around the surface for higher quality reconstruction: Instead of importance sampling based on the SDF predictions, which requires multiple forward passes through the network [37,68], we draw N' point samples from a normal distribution centered around the depth reading $\mathcal{N}(z(r), \lambda^2)$. This results in $N+N'$ total samples, without querying the more expensive multi-resolution hash encoding or the networks.

Hybrid SDF Modeling. Due to the imperfect segmentation and external occlusions, we propose a hybrid signed distance model. Specifically, we divide the space into three regions to learn the SDF (see Fig. 3):

- *Uncertain free space:* These points (yellow in the figure) correspond to the background in the segmentation mask or to pixels with missing depth values, for which the observation is unreliable. For instance, at ray r_1's pixel location in the binary mask, the finger's occlusion results in background prediction, even though it actually corresponds to the pitcher handle. Naively ignoring the back-

ground for emitting the ray would lose the contour information, causing bias. Therefore, instead of fully trusting or ignoring *uncertain free space*, we assign a small positive value ϵ to be potentially external to the object surface so that it can quickly adapt when a more reliable observation is available later:

$$\mathcal{L}_u = \frac{1}{|\mathcal{X}_u|} \sum_{x \in \mathcal{X}_u} (\Omega(x) - \epsilon)^2. \tag{8}$$

- *Empty space:* These points (red in the figure) are in front of the depth reading up to a truncation distance, making them almost certainly external to the object surface. We apply L_1 loss to the truncated signed distance to encourage sparsity:

$$\mathcal{L}_e = \frac{1}{|\mathcal{X}_e|} \sum_{x \in \mathcal{X}_e} |\Omega(x) - \lambda|. \tag{9}$$

- *Near-surface space:* These points (blue in the figure) are near the surface, no more than $z(r) + 0.5\lambda$ distance behind the depth reading to model self-occlusion. This space is critical for learning the sign flipping in SDF and the zero level set. We approximate the near-surface SDF by projective approximation for efficiency:

$$\mathcal{L}_{surf} = \frac{1}{|\mathcal{X}_{surf}|} \sum_{x \in \mathcal{X}_{surf}} \left(\Omega(x) + d_x - d_D\right)^2, \tag{10}$$

where $d_x = \|x - o(r)\|_2$ and $d_D = \left\|\pi^{-1}(z(r))\right\|_2$ are the distance from ray origin to the sample point and the observed depth point, respectively.

Training. The trainable parameters include the multi-resolution hash encoder, Ω, Φ, and the object pose updates in the tangent space parametrized in Lie Algebra $\Delta\bar{\xi} \in \mathbb{R}^{(|\mathcal{P}|-1)\times 6}$, wherein we freeze the first memory frame's pose to be the anchor point. The training loss is:

$$\mathcal{L} = w_u\mathcal{L}_u + w_e\mathcal{L}_e + w_{surf}\mathcal{L}_{surf} + w_c\mathcal{L}_c + w_{eik}\mathcal{L}_{eik}, \tag{11}$$

where $\mathcal{L}_c$ denotes the L_2 loss over the foreground color for appearance network supervision:

$$\mathcal{L}_c = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \left\|\Phi(f_{\Omega(x)}, n(x), d(r)) - \bar{c}(r)\right\|_2, \tag{12}$$

and $\mathcal{L}_{eik}$ is the Eikonal regularization [13] over the SDF in *near-surface space*:

$$\mathcal{L}_{eik} = \frac{1}{|\mathcal{X}_{surface}|} \sum_{x \in \mathcal{X}_{surface}} (\|\nabla\Omega(x)\|_2 - 1)^2. \tag{13}$$

Unlike [68] which requires ground-truth mask as input, we do not perform mask supervision, since the predicted mask is often noisy from the network.

4. Experiments

4.1. Datasets

To evaluate our method, we consider three real-world datasets with drastically different forms of interactions and dynamic scenarios. For results on wild application and static scenes, see project page.

HO3D [14]: This dataset contains the RGBD video of a human hand interacting with YCB objects [5], captured by Intel RealSense camera at close range. Ground truth is automatically generated from multi-view registration. We adopt the most recent version HO-3D_v3 and test on the official evaluation set. This results in 4 different objects, 13 video sequences, and 20428 frames in total.

YCBInEOAT [72]: This dataset contains the ego-centric RGBD videos of a dual-arm robot manipulating the YCB objects [5] captured by Azure Kinect camera at mid range. There are three types of manipulation: (1) single arm pick-and-place, (2) within-hand manipulation, and (3) pick-and-place with handoff between arms. Although this dataset was originally developed to evaluate pose estimation approaches relying on CAD models, we do not provide any object prior knowledge to the evaluated methods. There are 5 different objects, 9 videos, and 7449 frames in total.

BEHAVE [4]: This dataset contains the RGBD video of a human body interacting with the objects, captured at far range by a pre-calibrated multi-view system with Azure Kinect cameras. However, we constrain our evaluation to the single-view setting, where severe occlusions frequently occur. We evaluate on the official test split excluding the deformable objects. This results in 16 different objects, 70 videos/scenes, and 107982 frames in total.

4.2. Metrics

We separately evaluate pose estimation and shape reconstruction. For 6-DoF object pose, we compute the area under the curve (AUC) percentage of *ADD* and *ADD-S* metrics [17, 69, 75] using ground-truth object geometry. For 3D shape reconstruction, we compute the chamfer distance between the final reconstructed mesh and ground-truth mesh in the canonical coordinate frame defined by the first image of each video. More details can be found in the appendix.

4.3. Baselines

We compare against DROID-SLAM (RGBD) [61], NICE-SLAM [85], KinectFusion [43], BundleTrack [69] and SDF-2-SDF [53] using their open-source implementations with the best tuned parameters. We additionally include the baseline results from their leaderboard. Note that methods such as [20, 42] focus on deformable objects and the root 6-DoF tracking and fusion are often based on [43], whereas we focus on rigid objects that are dynamically moving. We thus omit their comparisons. The inputs to each evaluated method are the RGBD video and the first frame's mask indicating the object of interest. We augment the comparison methods with the same video segmentation masks used in our framework for fair comparison, to focus on 6-DoF object pose tracking and 3D reconstruction performance. In the case of tracking failure, no re-initialization is performed to test long-term tracking robustness.

DROID-SLAM [61], NICE-SLAM [85] and KinectFu-

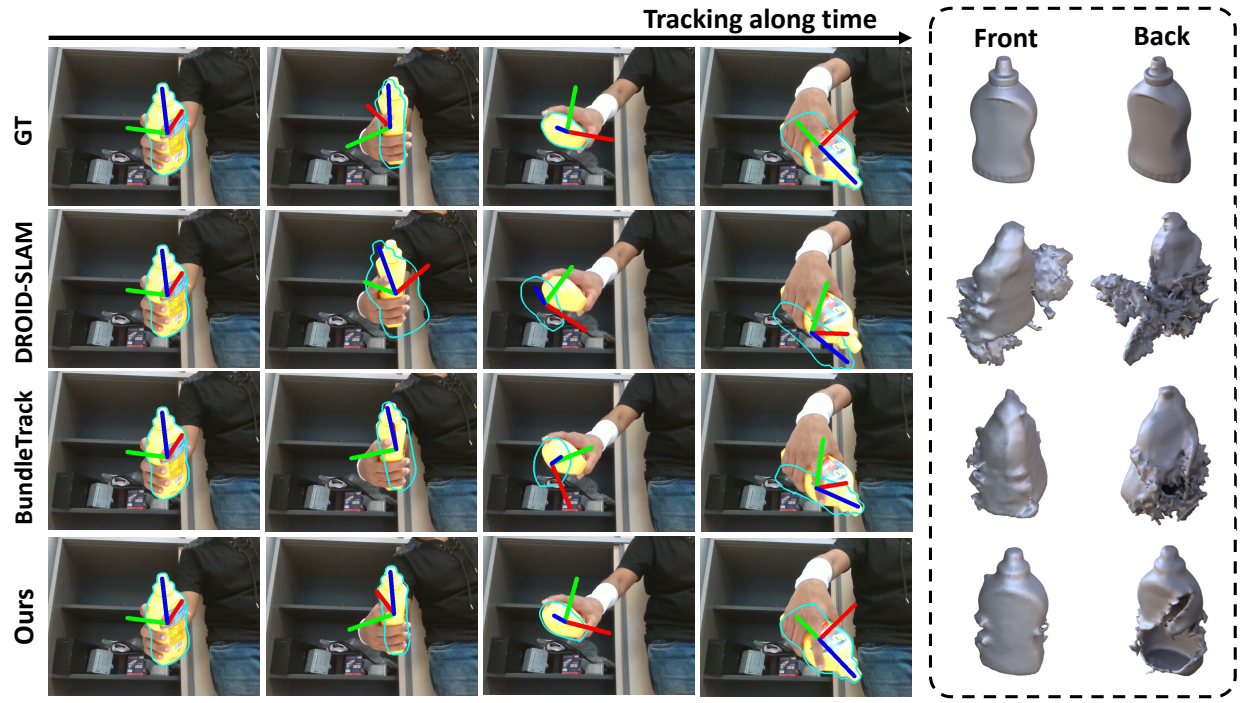

Figure 4. Qualitative comparison of the three most competitive methods on HO3D Dataset. **Left:** 6-DoF pose tracking visualization, where the contour (cyan) is rendered with the estimated pose. Note, as shown in the 2nd column, that our predicted pose sometimes corrects errors in the ground truth. **Right:** Front and back view of the final reconstructed shape output by each method. Due to hand occlusions, some parts of the object are never visible in the video. Meshes are rendered from the same viewpoint, though significant drift of DROID-SLAM and BundleTrack results in erroneously rotated meshes.

sion [43] were originally proposed for camera pose tracking and scene reconstruction. When given the segmented images, they run in an object-centric setting. Since DROID-SLAM [61] and BundleTrack [69] cannot reconstruct an object mesh, we augment these methods with TSDF Fusion [9, 83] for shape reconstruction evaluation. For NICE-SLAM [85] and our method, we initialize the neural volume's bound using only the first frame's point cloud (to preserve causal processing, we cannot access future frames).

4.4. Comparison Results on HO3D

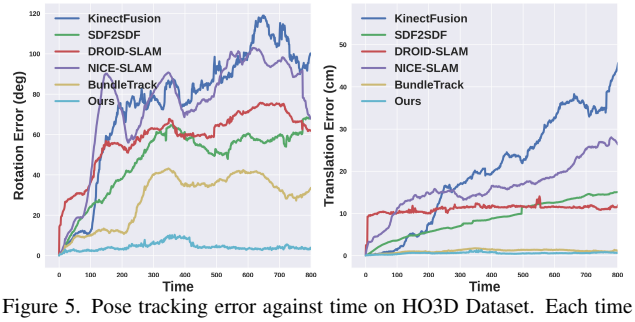

Figure 5. Pose tracking error against time on HO3D Dataset. Each time stamp's result is averaged across all videos. **Left:** Rotation error measured by geodesic distance. **Right:** Translation error.

Quantitative results on HO3D are shown in Tab. 1 and Fig. 5. Our method outperforms the comparison methods by a large margin on both 6-DoF pose tracking and 3D reconstruction. For DROID-SLAM [61], NICE-SLAM [85] and KinectFusion [43], when working in an object-centric setting, significantly less texture or geometric (purely planar or cylindrical object surfaces) cues can be leveraged for tracking, leading to poor performance. Fig. 5 presents the tracking error against time to study the long-term tracking drift. While BundleTrack [69] achieves similarly low translation error as our approach, it struggles on the rotation estimation. In contrast, our method maintains a low tracking error throughout the video. We provide per-video quantitative results in the appendix.

Fig. 4 shows example qualitative results of the three most competitive methods. Despite multiple challenges such as severe hand occlusions, self-occlusions, little texture cues in intermediate observations and strong lighting reflections, our method keeps tracking accurately along the video and obtains dramatically higher quality 3D object reconstruction. Notably, our predicted pose is sometimes more accurate than ground-truth, which was annotated by multi-camera multi-view registration leveraging hand priors.

	Pose		Reconstruction
	ADD-S (%) ↑	ADD (%) ↑	CD (cm) ↓
NICE-SLAM [85]	22.29	8.97	52.57
SDF-2-SDF [53]	35.88	16.08	9.65
KinectFusion [43]	25.81	16.54	15.49
DROID-SLAM [61]	64.64	33.36	30.84
BundleTrack [69]	92.39	66.01	52.05
Ours	**96.52**	**92.62**	**0.57**

Table 1. Comparison on HO3D Dataset. ADD and ADD-S are AUC percentage (0 to 0.1 m). Reconstruction is measured by chamfer distance.

	Pose		Reconstruction
	ADD-S (%) ↑	ADD (%) ↑	CD (cm) ↓
NICE-SLAM [85]	23.41	12.70	6.13
SDF-2-SDF [53]	28.20	14.04	2.61
KinectFusion [43]	46.39	34.68	4.63
DROID-SLAM [61]	32.12	20.39	2.34
BundleTrack [69]	93.01	87.26	2.81
BundleTrack* [69]	92.53	**87.34**	-
MaskFusion* [50]	41.88	35.07	-
TEASER++* [78]	81.17	57.91	-
Ours	**93.77**	86.95	**1.16**

Table 2. Comparison on YCBInEOAT Dataset. ADD and ADD-S are AUC percentage (0 to 0.1 m). Reconstruction is measured by chamfer distance.

4.5. Comparison Results on YCBInEOAT

Quantitative results on YCBInEOAT are shown in Tab. 2. This dataset captures the interaction between the robot arms and the object from an ego-centric view, which leads to challenges due to the constrained camera view and severe occlusions by the robot arms. For completeness, in this table we also include additional baseline methods from [69].[1] The results from these methods, indicated by asterisk (*), are simply copied from [69]. Note that, in the case of (non-asterisk) BundleTrack, we re-run the algorithm with the same segmentation masks as ours for fair comparison, and we augment with TSDF Fusion for reconstruction evaluation (same as Tab. 1). We omit the re-running for MaskFusion* [50] and TEASER++* [78] due to their relatively poorer performance.

Our approach sets a new benchmark record on ADD-S metric and chamfer distance in 3D reconstruction, while obtaining comparable performance with the previous state-art-art method on ADD metric. In particular, while BundleTrack [69] achieves competitive object pose tracking, it does not obtain satisfactory 3D reconstruction results. This demonstrates the benefits of our co-design of tracking and reconstruction.

4.6. Comparison Results on BEHAVE

	Pose		Reconstruction
	ADD-S (%) ↑	ADD (%) ↑	CD (cm) ↓
DROID-SLAM [61]	56.14	32.29	11.24
BundleTrack [69]	59.06	45.03	19.27
KinectFusion [43]	38.37	28.45	9.36
NICE-SLAM [85]	28.80	11.93	36.03
SDF-2-SDF [53]	25.71	10.05	35.99
Ours	83.63	67.52	4.66

Table 3. Comparison on BEHAVE Dataset. ADD and ADD-S are AUC percentage (0 to 0.5 m). Reconstruction is measured by chamfer distance.

Quantitative results on BEHAVE are shown in Tab. 3. We refer to the supplemental material for more detailed results. In our setting of single-view and zero-shot transfer without leveraging human body priors, this dataset exhibits extreme challenges. For instance, (i) there are long-term complete occlusions when the human carries the object and faces away from the camera; (ii) severe motion blur and abrupt displacement frequently occur due to the human freely swinging the object; (iii) the objects are of diverse properties and vary greatly in size; (iv) the video is captured at a distance from the camera, making it difficult for depth sensing. Therefore, evaluation on this benchmark pushes the boundary to a more difficult setting. Despite these challenges, our method is still able to perform long-term robust tracking in most scenarios and performs significantly better than previous methods.

[1]For fair comparison, we only include baselines from [69] that—like our method—do not require instance- or category-level object knowledge.

4.7. Ablation Study

Ablations	Pose		Reconstruction
	ADD-S (%) ↑	ADD (%) ↑	CD (cm) ↓
Ours w/o memory	82.05	56.96	-
Ours w/o NOF	93.09	76.69	-
Ours-GPG	93.82	78.82	-
Ours w/o hybrid SDF	85.31	73.57	2.62
Ours w/o compact mem pool	87.48	59.99	0.90
Ours	**96.52**	**92.62**	**0.61**

Table 4. Ablation study of our design choices. *Ours w/o memory* removes the memory related modules and only performs frame-to-frame coarse pose estimation. *Ours w/o NOF* removes the Neural Object Field module and $\mathcal{L}_s$ in Eq. (1). *Ours-GPG* replaces the Neural Object Field by global pose graph optimization using all memory frames. It runs in a separate thread concurrently same as Neural Object Field. *Ours w/o hybrid SDF* only considers foreground rays in the mask instead of hybrid SDF modeling. *Ours w/o compact mem pool* adopts similar strategy of selecting frames to add into the memory pool as well as selecting the subset memory frames for pose graph optimization as in [69].

We investigate the effectiveness of our design choices on HO3D dataset given its more accurate pose annotations. The results are shown in Tab. 4. *Ours w/o memory* achieves dramatically worse performance as there is no mechanism to alleviate tracking drift. For *Ours-GPG*, even with similar amount of computation, it struggles on objects or observations with little texture or geometric cues due to handcrafted losses. Aside from object pose tracking, *Ours w/o memory*, *Ours w/o NOF* and *Ours-GPG* lack the module for 3D object reconstruction. *Ours w/o hybrid SDF* ignores the contour information and can be biased by false positive segmentation when rectifying the memory frames' pose. These lead to less stable pose tracking and more noisy final 3D reconstruction. *Ours w/o compact mem pool*, when under the same computational budget, leads to insufficient pose coverage during pose graph optimization and Neural Object Field learning, as mentioned in Sec. 3.2.

5. Conclusion

We presented a novel method for 6-DoF object tracking and 3D reconstruction from a monocular RGBD video. Our method only requires segmentation of the object in the initial frame. Leveraging two parallel threads that perform online graph pose optimization and Neural Object Field representation respectively, our method is able to handle challenging scenarios, such as fast motion, partial and compete occlusion, lack of texture, and specular highlights. On several datasets we have demonstrated state-of-the-art results compared with existing methods. Future work will be aimed at leveraging shape priors to reconstruct unseen parts.

References

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 1

[2] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9(5):698–700, 1987. 3

[3] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. 1, 2

[4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. BEHAVE: Dataset and method for tracking human object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2022. 2, 6

[5] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set. *IEEE Robotics and Automation Magazine*, 22(3), Sept. 2015. 6

[6] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12417–12426, 2021. 2

[7] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 11973–11982, 2020. 1

[8] Ho Kei Cheng and Alexander G Schwing. XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model. In *ECCV*, 2022. 2

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996. 7

[10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 2

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[12] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A PyTorch library for accelerating 3D deep learning research. https://github.com/NVIDIAGameWorks/kaolin, 2022. 5

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, pages 3789–3799, 2020. 6

[14] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 6

[15] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. DeXtreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv preprint arXiv:2210.13702*, 2022. 1

[16] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 571–580, 2020. 2

[17] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. FS6D: Few-shot 6D pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6814–6824, 2022. 6

[18] Yinghao Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. InterCap: Joint markerless 3D tracking of humans and objects in interaction. In *German Conference on Pattern Recognition (GCPR)*, volume 13485 of *Lecture Notes in Computer Science*, pages 281–299, 2022. 2

[19] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 4

[20] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 362–379. Springer, 2016. 6

[21] Yuheng Jiang, Suyi Jiang, Guoxing Sun, Zhuo Su, Kaiwen Guo, Minye Wu, Jingyi Yu, and Lan Xu. NeuralHOFusion: Neural volumetric rendering under human-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6155–6165, 2022. 2

[22] Daniel Kappler, Franziska Meier, Jan Issac, Jim Mainprice, Cristina Garcia Cifuentes, Manuel Wüthrich, Vincent Berenz, Stefan Schaal, Nathan Ratliff, and Jeannette Bohg. Real-time perception meets reactive motion generation. *IEEE Robotics and Automation Letters*, 3(3):1864–1871, 2018. 1

[23] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327, 2011. 2

[24] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D

pose estimation. In *European Conference on Computer Vision*, pages 574–591, 2020. 1, 2

[25] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D pose estimation of novel objects via render & compare. In *6th Annual Conference on Robot Learning (CoRL)*, 2022. 2

[26] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy Mitra, and Leonidas J Guibas. Pix2Surf: Learning parametric 3D surface models of objects from images. In *European Conference on Computer Vision (ECCV)*, pages 121–138, 2020. 2

[27] Xiaolong Li, Yijia Weng, Li Yi, Leonidas Guibas, A. Lynn Abbott, Shuran Song, and He Wang. Leveraging SE(3) equivariance for self-supervised category-level object pose estimation from point clouds. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:15370–15381, 2021. 1

[28] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 1, 2

[29] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018. 3

[30] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. Keypoint-based category-level object pose tracking from an RGB sequence with uncertainty estimation. In *International Conference on Robotics and Automation (ICRA)*, 2022. 2

[31] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3D hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14687–14697, 2021. 2

[32] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. *ECCV*, 2022. 2

[33] Lu Ma, Mahsa Ghafarianzadeh, David Coleman, Nikolaus Correll, and Gabe Sibley. Simultaneous localization, mapping, and manipulation for unsupervised object discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 2

[34] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(12):2633–2651, 2015. 1

[35] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. In *International Conference on 3D Vision (3DV)*, pages 32–41, 2018. 1, 2

[36] Nathaniel Merrill, Yuliang Guo, Xingxing Zuo, Xinyu Huang, Stefan Leutenegger, Xi Peng, Liu Ren, and Guoquan Huang. Symmetry and uncertainty-aware object SLAM for 6DoF object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14901–14910, 2022. 1, 2

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 5

[38] Norman Müller, Yu-Shiang Wong, Niloy J Mitra, Angela Dai, and Matthias Nießner. Seeing behind objects for 3D multi-object tracking in RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6071–6080, 2021. 2

[39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 5

[40] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 1, 2

[41] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2, 4

[42] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 6

[43] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 3, 6, 7, 8

[44] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5589–5599, 2021. 1, 2

[45] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10710–10719, 2020. 2

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 5

[47] Timothy Patten, Kiru Park, Markus Leitner, Kevin Wolfram, and Markus Vincze. Object learning for 6D pose estimation and grasping from RGB-D videos of in-hand manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4831–4838, 2021. 2

[48] Carl Yuheng Ren, Victor Prisacariu, David Murray, and Ian Reid. STAR3D: Simultaneous tracking and reconstruction of 3D objects using RGB-D data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1561–1568, 2013. 2

[49] Martin Rünz and Lourdes Agapito. Co-Fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017. 2

[50] Martin Runz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, 2018. 1, 2, 3, 8

[51] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013. 1, 2

[52] Akash Sharma, Wei Dong, and Michael Kaess. Compositional and scalable object SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11626–11632, 2021. 1, 2

[53] Miroslava Slavcheva, Wadim Kehl, Nassir Navab, and Slobodan Ilic. SDF-2-SDF registration for real-time 3D reconstruction from RGB-D data. *International Journal of Computer Vision (IJCV)*, 126(6):615–636, 2018. 3, 4, 6, 7, 8

[54] Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6855–6865, 2022. 2

[55] Michael Strecke and Jörg Stückler. EM-fusion: Dynamic object-level SLAM with probabilistic data association. In *Proceedings IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[56] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6229–6238, 2021. 2

[57] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2

[58] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022. 2

[59] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021. 1, 2

[60] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 2

[61] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual slam for monocular, stereo, and RGB-D cameras. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:16558–16569, 2021. 2, 6, 7, 8

[62] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6D object pose and size estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 530–546, 2020. 1

[63] Henning Tjaden, Ulrich Schwanecke, and Elmar Schomer. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 124–132, 2017. 2

[64] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14540–14549, 2020. 1, 2

[65] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-PACK: Category-level 6D pose tracker with anchor-based keypoints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066, 2020. 2

[66] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16611–16621, 2021. 1

[67] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2642–2651, 2019. 1, 2

[68] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 5, 6

[69] Bowen Wen and Kostas Bekris. BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8067–8074, 2021. 2, 3, 4, 6, 7, 8

[70] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6401–6408. IEEE, 2022. 1

[71] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *RSS*, 2022. 1

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

[73] 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. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6210–6217. IEEE, 2020. 1

[74] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. CAPTRA: Category-level pose tracking for rigid and articulated objects from point clouds. *ICCV*, 2021. 1

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

[76] Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. CHORE: Contact, human and object reconstruction from a single RGB image. In *European Conference on Computer Vision (ECCV)*, October 2022. 2

[77] Binbin Xu and et al. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *ICRA*, 2019. 2

[78] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Trans. Robotics*, 37(2):314–333, Apr. 2021. 8

[79] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11097–11106, 2021. 2

[80] Zhenpei Yang, Zhile Ren, Miguel Angel Bautista, Zaiwei Zhang, Qi Shan, and Qixing Huang. FvOR: Robust joint shape and pose optimization for few-view object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2497–2507, 2022. 2

[81] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[82] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2492–2502, 2020. 5

[83] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *CVPR*, 2017. 7

[84] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[85] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 1, 2, 6, 7, 8