Reconstructing Hand-Held Objects from Monocular Video

Di Huang
dihuanginfo@gmail.com
The University of Sydney
Australia

Xiaopeng Ji
xp.ji@cad.zju.edu.cn
Zhejiang University
China

Xingyi He
hexingyi8@gmail.com
Zhejiang University
China

Jiaming Sun
suenjiaming@gmail.com
Image Derivative Inc.
China

Tong He
tonghe90@gmail.com
Shanghai AI Laboratory
China

Qing Shuai
s_q@zju.edu.cn
Zhejiang University
China

Wanli Ouyang
wanli.ouyang@sydney.edu.au
Shanghai AI Laboratory
The University of Sydney
Australia

Xiaowei Zhou*
xwzhou@zju.edu.cn
State Kay Lab of CAD&CG
Zhejiang University
China

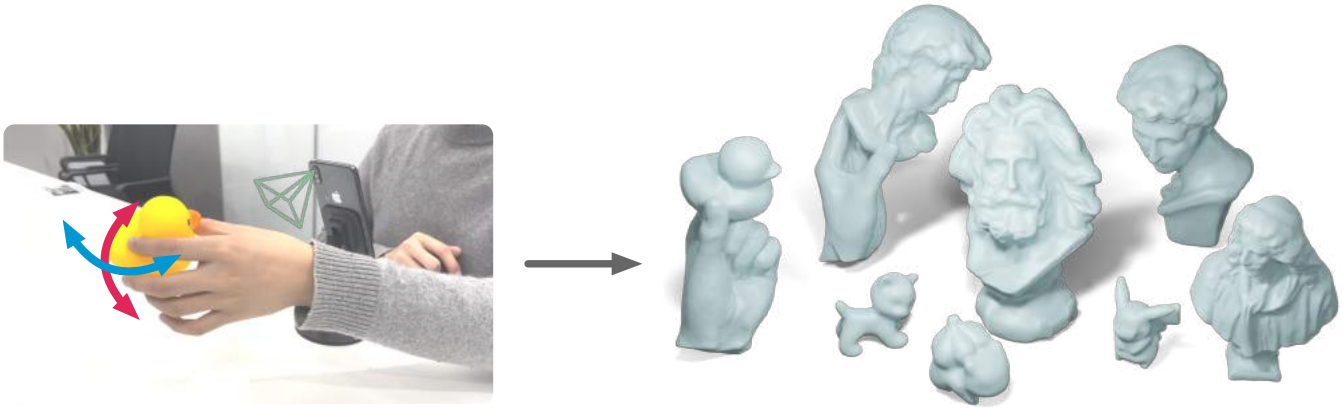

Figure 1: Hand-held object reconstruction. We propose a novel approach that reconstructs a moving hand-held object from a video captured by a static RGB camera, which is a common real-life scenario. The proposed approach does not require any prior knowledge about the object and thus can be widely applicable. The data capture process (left) and reconstructed 3D meshes (right) are visualized.

ABSTRACT

This paper presents an approach that reconstructs a hand-held object from a monocular video. In contrast to many recent methods that directly predict object geometry by a trained network, the proposed approach does not require any learned prior about the object and is able to recover more accurate and detailed object geometry. The key idea is that the hand motion naturally provides multiple views of the object and the motion can be reliably estimated by a hand pose tracker. Then, the object geometry can be recovered by solving a multi-view reconstruction problem. We devise an implicit neural representation-based method to solve the reconstruction problem and address the issues of imprecise hand pose estimation, relative hand-object motion, and insufficient geometry optimization for small objects. We also provide a newly collected dataset with 3D ground truth to validate the proposed approach. The dataset and code will be released at https://dihuangdh.github.io/hhor.

*Corresponding author.

SA '22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea

ACM ISBN 978-1-4503-9470-3/22/12…$15.00
https://doi.org/10.1145/3550469.3555401

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**.

KEYWORDS

Object reconstruction, joint hand-object reconstruction

ACM Reference Format:
Di Huang, Xiaopeng Ji, Xingyi He, Jiaming Sun, Tong He, Qing Shuai, Wanli Ouyang, and Xiaowei Zhou. 2022. Reconstructing Hand-Held Objects from Monocular Video. In *SIGGRAPH Asia 2022 Conference Papers (SA '22*

Conference Papers), December 6–9, 2022, Daegu, Republic of Korea. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3550469.3555401

1 INTRODUCTION

Reconstructing a 3D object from 2D images is vital for many applications such as augmented reality, 3D printing, and robotic manipulation. Existing approaches [Izadi et al. 2011; Schönberger et al. 2016] mainly focus on the setting of capturing a static object with an RGB camera or depth sensor rotating around the object. This work investigates a different setting where the object is held by a moving hand with a fixed grasping gesture in front of a static RGB camera, as shown in Figure 1. This setting is not only common in daily scenarios, e.g., video conferencing, but also potentially provides a more user-friendly object capture procedure without the need to walk around the object.

The hand-held object capture leads to the following challenges that make existing reconstruction methods inapplicable. First, relative motion between the object and the camera needs to be solved for video-based 3D reconstruction. Existing systems generally rely on structure from motion (SfM) algorithms [Schönberger and Frahm 2016]. However, SfM assumes rigid scenes while in our setting, the object is moving independently of the background. Another alternative is to track the 6DoF pose of the object. But traditional object pose estimation methods either need object CAD models [Tjaden et al. 2016, 2017] or rely on rich textures for reliable feature tracking [Lowe 2004], both of which cannot be satisfied in our setting. Recent works [Chen et al. 2020; Wang et al. 2019] adopt learning-based methods for object pose estimation but need training on the same object category. Second, even if the object motion is given, dense reconstruction of a textureless object from RGB images is still difficult, particularly when the object mask is not provided. Moreover, the object is partially occluded by the hand, which makes object pose tracking and reconstruction even harder. To solve these challenges, some recent works resort to a learning-based approach that learns object shape prior to making it possible to estimate object shape from a single view with neural networks [Choy et al. 2016; Fan et al. 2017; Hasson et al. 2019; Karunratanakul et al. 2020]. While these works show promising results, the learned single-view reconstruction networks can hardly generalize to new object categories.

In this paper, we propose a novel framework to reconstruct a moving hand-held object from a monocular video without the need to know any prior of the object. The key insight is that the physical constraint between hand and object provides important motion cues for reconstructing the object: while the 3D motion of the object is hard to track, we are able to infer object motion by tracking hand motion based on a learning-based hand pose estimator. For dense geometry reconstruction, instead of using traditional multi-view stereo algorithms that cannot handle textureless objects, we propose to leverage the recent advances in neural representation-based methods [Mildenhall et al. 2020; Wang et al. 2021a], which directly optimize 3D scene geometry and appearance represented by implicit functions with differentiable rendering. However, directly using such a representation leads to poor reconstruction quality. We find three main issues that degrade the reconstruction quality: 1) imprecise hand pose, 2) relative motion between the hand and object, and 3) inefficient optimization for the object geometry due to its small size compared to the hand. We propose three solutions to address these challenges. First, we simultaneously optimize the neural scene representation and the relative pose between the hand and camera. Second, we equip the neural scene representation with a learnable deformation field for modeling the relative motion between the hand and object. Third, we leverage 2D object masks to adaptively sample more rays on the object during training to make the network more focused on the object instead of the hand. Note that by taking an additional background image, the object mask can be obtained using background subtraction and hand segmentation, without the need to know the object category.

To validate the proposed approach, we collect a new hand-held object dataset that consists of 4K videos of 35 objects, 14 of which are paired with ground-truth meshes obtained by a commercial 3D scanner. This dataset covers various types of daily objects. The experiments show that our approach is able to accurately reconstruct objects and hands and outperforms several baseline methods.

2 RELATED WORK

Multi-view 3D Reconstruction. Conventional multi-view 3D reconstruction methods [Furukawa and Ponce 2007; Kutulakos and Seitz 2000; Yao et al. 2018] follow a two-stage pipeline: They first estimate camera parameters by SfM and then use multi-view stereo (MVS) to reconstruct the scene from calibrated images. A representative system in this category is COLMAP [Schönberger and Frahm 2016; Schönberger et al. 2016], which first estimates multi-view depth maps and then fuses depth data into 3D models. There is a recent trend to solve the multi-view 3D reconstruction by recovering the implicit scene representation using volume rendering. NeRF [Mildenhall et al. 2020] proposes to represent the scene as a radiance field of density and color. NeuS [Wang et al. 2021a] improves the NeRF representation by replacing the density radiance field with the Signed Distance Field (SDF), leading to much better reconstruction quality. These works still require camera poses from SfM, while some more recent works [Lin et al. 2021a; Wang et al. 2021b] aim to get rid of the SfM phase by simultaneously optimizing the radiance field and camera poses. However, they are prone to local optimum without proper camera pose initialization [Wang et al. 2021b]. Existing multi-view 3D reconstruction methods mentioned above generally rely on SfM to estimate camera parameters for each frame. Unfortunately, SfM cannot work in our setting which aims to reconstruct moving hand-held objects from a monocular RGB video. First, standard SfM assumes rigid scenes, while in our setting, the object is moving independently of the background. Second, the hand and object are often low-textured, causing standard SfM to fail.

Single-view 3D Reconstruction. Another line of works aims at reconstructing 3D objects from single-view input by learning object shape prior. According to the different types of representations, single-view 3D reconstruction methods can be categorized as voxel-based [Choy et al. 2016; Dai et al. 2017; Riegler et al. 2017; Tatarchenko et al. 2017; Wang et al. 2017], point-based [Fan et al. 2017; Insafutdinov and Dosovitskiy 2018; Jiang et al. 2018; Mandikal et al. 2018], and mesh-based [Groueix et al. 2018; Wang et al. 2018]. Voxels are suitable for CNNs to process but suffer from limited

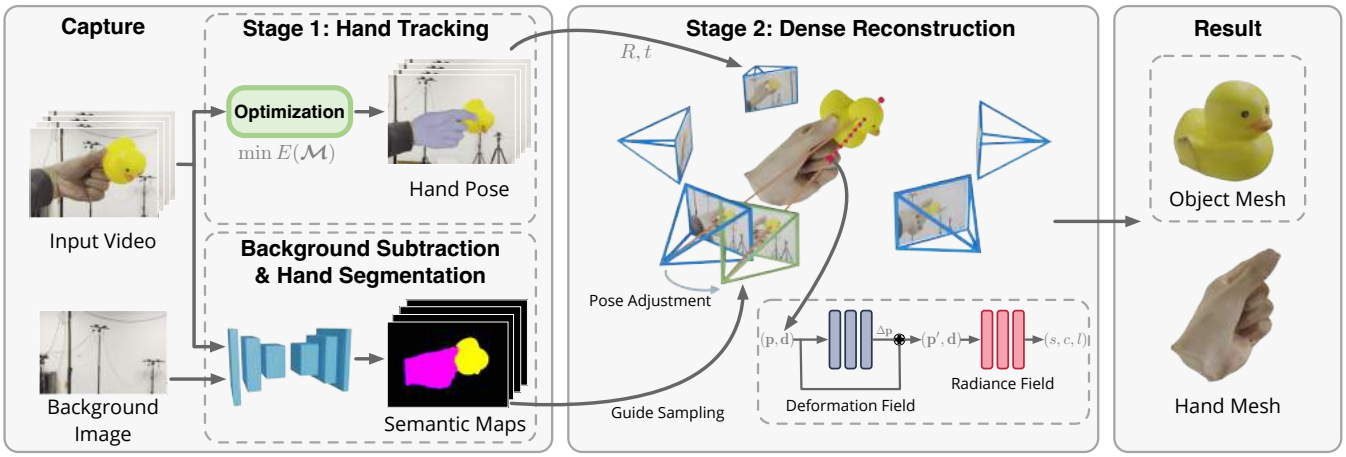

Figure 2: The pipeline of our approach, which consists of two stages. **Hand tracking:** by minimizing the reprojection error of hand keypoints detected by a learned detector, the 3D hand pose and the camera motion relative to it are recovered. **Dense reconstruction**: an implicit neural representation-based method is employed to reconstruct the SDF and color fields of the hand and object. Three additional modules are proposed: the pose adjustment to compensate for imprecise hand pose tracking, the deformation field to model the relative motion between the hand and object, and the semantics-guided sampling to improve object reconstruction quality.

resolution. Point clouds and meshes are compact representations, but difficult to be generated by neural networks. To overcome the limitations of above 3D representations, recent works propose to leverage implicit neural representations [Chen and Zhang 2019; Chibane et al. 2020; Mescheder et al. 2019; Park et al. 2019; Xu et al. 2019]. In these methods, the geometry and appearance of a 3D scene are represented as continuous functions of spatial locations, which are approximated by learned neural networks. All the above single-view 3D reconstruction methods need to learn strong object shape prior from massive training data. Given the fact that there is a lack of large-scale 3D object datasets with real images and corresponding 3D models, previous methods are often trained on synthetic datasets like ShapeNet [Chang et al. 2015] and suffer from limited generalization capability, particularly when the object category of the test image is unseen before. Moreover, the quality of single-view reconstruction is inherently inferior to multi-view methods.

Hand Tracking. Hand tracking is an active research topic in computer vision. Early works focus on tracking hand motion with a depth sensor. [Tompson et al. 2014] uses a deep convolutional network to extract features and then apply inverse kinematics for accurate hand pose. [Taylor et al. 2016] presents a hand tracking system for real-time AR applications. The system uses a smooth hand model and non-linear optimization to achieve impressive hand tracking performance. [Mueller et al. 2019] proposes a novel model which is capable of tracking two interacting hands in real-time. [Hampali et al. 2020] jointly estimates the hand pose and 6D object pose using one or several RGB-D cameras and differentiable rendering. Recently, hand tracking from RGB input has been paid more attention. [Han et al. 2020; Mueller et al. 2018] split the hand tracking problem into two stages. They estimate the 3D keypoints locations first and then apply a skeleton fitting process to get the final tracking results. Other works[Boukhayma et al. 2019; Ge et al. 2019; Kulon et al. 2020] track the hand by direct estimating pose and shape parameters of a parametric hand model, e.g., the MANO model [Romero et al. 2017].

In-hand Capture. Since hands frequently interact with objects, reconstructing objects in hands is an essential problem for in-hand AR/VR applications. Early in-hand scanning uses depth images as input. [Rusinkiewicz et al. 2002] proposes an in-hand 3D model acquisition system, which allows the user to manipulate the object and see the scanning mesh in real time. The system captures local surface patches, then aligns and integrates them into a complete 3D mesh. Later work [Weise et al. 2008] improves the system with efficient registration failure detection and three better registration methods. Since the system heavily relies on object textures and often fails for textureless objects, [Tzionas and Gall 2015] proposes to use the hand contact points as the additional registration energy. [Zhang et al. 2019] proposes a novel pipeline to simultaneously reconstruct hand pose and object shape using two RGBD cameras, which is later simplified to one RGBD camera by [Zhang et al. 2021]. Recent works focus on the simultaneous reconstruction of hand and object from a single RGB image. [Hasson et al. 2019] jointly reconstructs the hand and objects by using two separate branches: one branch estimates MANO parameters, and the other reconstructs the object shape in the normalized coordinate space. [Karunratanakul et al. 2020; Ye et al. 2022] use implicit representations to recover the object shape. However, all learning-based methods are limited to a few known object categories included in the training set. In contrast, our method works on RGB input and does not rely on any learned object prior to recover high-fidelity object meshes.

3 DATA CAPTURE AND PRE-PROCESSING

3.1 Data Capture

Our system only uses a still monocular RGB camera as the capture device (e.g., iPhone or iPad). We first use the camera to take a snapshot of the background scene, which is used to generate the foreground mask later. Then, we capture a short grasping video

for hand-object reconstruction. During the data capture, users only need to grasp and rotate the object in front of the camera. The camera is fixed during the capture and the object is firmly grasped by the hand, which means that the relative motion between the hand and the object is small. Each video contains around 1800 frames with a 30 fps frame rate and 4K (2160×3840) resolution.

3.2 Hand-object Segmentation

Our method requires hand-object segmentation maps to help separate the hand and object. We design a two-step framework to generate 2D semantic maps. First, we leverage background matting [Lin et al. 2020] to extract the foreground masks, which contain both the hand and object. Second, we segment the hand from the foreground masks with a hand segmentation network. Notably, we collect a new egocentric-view hand segmentation dataset to train the network. The whole segmentation process does not rely on any object prior and can be widely generalized to various types of objects.

4 METHODS

This section describes the proposed approach for reconstructing a hand-held object from a monocular RGB video without any prior of the object. The proposed approach consists of two main stages: hand tracking (Section 4.1) and dense reconstruction (Section 4.2). An overview of our approach is presented in Figure 2.

4.1 Hand Tracking

We build our hand tracking method on the widely used parametric hand model MANO [Romero et al. 2017]. The hand mesh is defined by two sets of parameters: pose ($\boldsymbol{\theta}$) and shape ($\boldsymbol{\beta}$). The pose parameters affect the joint angles except for the wrist joint, and the shape parameters control the person-specific shape deformations. $\boldsymbol{R} \in \mathrm{SO}(3)$ and $\boldsymbol{T} \in \mathbb{R}^3$ are the relative rotation and translation between the hand and canonical MANO space, which also indicate the relative motion between the hand and camera. $\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{\theta}$ and $\boldsymbol{\beta}$ share the same parametrization with MANO. We denote the optimized hand model in this article as $\boldsymbol{M} = \{\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{R}, \boldsymbol{T}\}$. The paired joint locations of the MANO model can be denoted as $J(\boldsymbol{M})$.

Given a set of RGB images $\mathcal{I}$ captured by the monocular camera, hand tracking aims to fit the hand model parameters $\boldsymbol{\mathcal{M}} = \{\boldsymbol{M}_t\}_{t=1}^{N_T}$ to image observations, where N_T is the number of frames. Each $\boldsymbol{M}_t$ consists of $\boldsymbol{\theta}_t$, $\boldsymbol{\beta}_t$, $\boldsymbol{R}_t$ and $\boldsymbol{T}_t$. Some previous works [Han et al. 2020; Mueller et al. 2018, 2019; Taylor et al. 2016] follow a similar formulation but usually optimize hand parameters $\boldsymbol{M}_t$ per frame, considering only the short-term temporal constraints. Instead, since there are no significant hand pose changes in our setting, we propose to optimize the full-sequence hand parameters $\boldsymbol{\mathcal{M}}$ at once and share the hand parameters for the whole video: $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \cdots = \boldsymbol{\theta}_t$, $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \cdots = \boldsymbol{\beta}_t$. Sharing hand parameters reduces the number of optimization parameters and enforces pose consistency across the whole video, which makes the hand model fitting much faster and more robust to occlusions (See Appendix C). Specifically, we minimize the following energy function to optimize the hand parameters:

$$\boldsymbol{\mathcal{M}} = \min_{\boldsymbol{\mathcal{M}}} (E_{2D} + \omega_1 E_t + \omega_2 E_{reg}). \tag{1}$$

The error term E_{2D} evaluates the consistency between the recovered hand model and the input video:

$$E_{2D} = \sum_t ||\pi(J(\boldsymbol{M}_t)) - J_t||_2^2, \tag{2}$$

where $\mathcal{J} = \{J_t\}_{t=1}^{N_T}$ are detected 2D joints in all frames, π is the projection operator that projects 3D hand keypoints to the image plane using the camera intrinsic matrix $\boldsymbol{K}$ initialized by image width and height. $\boldsymbol{M}_t$ is the t-th hand model parameters. $J(\boldsymbol{M}_t)$ is the 3D keypoints of the t-th hand model.

E_t is a term to force temporal smoothness and E_{reg} is a regularization term to avoid abnormal hands:

$$E_t = \sum_t ||\boldsymbol{R}_t - \boldsymbol{R}_{t-1}|| + ||\boldsymbol{T}_t - \boldsymbol{T}_{t-1}||, \tag{3}$$

$$E_{reg} = \sum_t ||\boldsymbol{\theta}_t||_2^2 + ||\boldsymbol{\beta}_t||_2^2. \tag{4}$$

We set ω_1 and ω_2 in Equation (1) as 1e-4 and 5e-4 respectively in all experiments.

Directly fitting a hand model to 2D keypoint observations is highly non-linear and sensitive to initialization. Similar to SPIN [Kolotouros et al. 2019], we improve the fitting convergence by initializing hand model parameters for each frame with a neural network. Specifically, we train a MANO estimation network to provide the initialization. Then, we optimize the hand parameters in a multi-stage manner. First, we freeze the $\boldsymbol{\theta}_t$ and $\boldsymbol{\beta}_t$ and only optimize the $\boldsymbol{R}_t$ and $\boldsymbol{T}_t$ with E_{2D}. Then, we optimize $\boldsymbol{\mathcal{M}}$ by all the energy function in Equation (1). During the optimization process, we use the LBFGS as the optimizer. For more accurate 2D keypoint detection under egocentric-view and hand-object interactions, we train a 2D keypoint detection network on our mixed dataset (See Appendix B.1).

After hand tracking, we are able to convert the video frames to calibrated multi-view images in the hand/object-centric coordinates by the estimated $\{\boldsymbol{R}_t, \boldsymbol{T}_t\}$, and $\boldsymbol{K}$.

4.2 Dense Reconstruction

For dense reconstruction, we choose the Signed Distance Function (SDF) as the implicit surface representation. The SDF representation is capable of representing the high-quality object surface, from which meshes can be easily extracted by marching cubes [Lorensen and Cline 1987]. To recover the SDF, we leverage the recent differentiable SDF rendering technique [Wang et al. 2021a], which converts the SDF to a radiance field, renders images with volume rendering, and compares rendered images with input images to optimize the SDF.

4.2.1 Surface representation. We follow [Wang et al. 2021a] and represent the 3D geometry and appearance to be reconstructed as:

$$[s(\mathbf{p}), c(\mathbf{p}, \mathbf{d})] = F(\mathbf{p}, \mathbf{d}), \tag{5}$$

where F is an MLP network. F takes 3D location $\mathbf{p}$ and viewing direction $\mathbf{d}$ as input, and predicts RGB color $c(\mathbf{p}, \mathbf{d})$ and SDF value $s(\mathbf{p})$. Positional encoding functions are applied to $\mathbf{p}$ and $\mathbf{d}$ for capturing high-frequency signals.

As for rendering, for each pixel we sample a set of points along the camera ray passing through this pixel, which are denoted by $\{\mathbf{p}(z) | \mathbf{p}(z) = \mathbf{o} + z\mathbf{d}, z \in [z_n, z_f]\}$ where $\mathbf{o}$ is the origin of the

camera, $\mathbf{d}$ is the viewing direction of each image pixel, z_n and z_f denote the near and far bounds of the ray. Note that $\mathbf{o}$ and per-pixel $\mathbf{d}$ can be computed from hand parameters $\{\boldsymbol{R_t}, \boldsymbol{T_t}\}$ and $\boldsymbol{K}$. Then, the color of the pixel can be composed as

$$\hat{C} = \int_{z_n}^{z_f} \omega(z) c(\mathbf{p}(z), \mathbf{d}) dz, \tag{6}$$

where $\omega(z)$ is the weight function for accumulating colors. $\omega(z)$ is the product of $T(z)$ and $\rho(z)$. $T(z)$ measures the total transmittance accumulated from z_n to z. $\rho(z)$ is the density function that indicates the probability of occupancy. We leverage the 3D hand keypoints $J(\boldsymbol{M})$ to automatically determine the z_n and z_f, which are required to tediously manually set and tune in the previous method. Following NeuS [Wang et al. 2021a], an unbiased and occlusion-awareness function $\omega(z)$ is used to convert SDF values $s(\mathbf{p}(z))$ to $T(z)$ and $\rho(z)$:

$$T(z) = \exp(-\int_{z_n}^{z_f} \rho(z) dz), \tag{7}$$

$$\rho(z) = \max\left(\frac{-\frac{d\Phi_h}{dz}(s(\mathbf{p}(z)))}{\Phi_h(s(\mathbf{p}(z)))}, 0\right), \tag{8}$$

where $\Phi_h(x) = (1 + e^{-hx})^{-1}$ is the Sigmoid function, h^{-1} is a trainable parameter during optimization. In practice, the above formulations are numerically approximated using quadrature.

4.2.2 Camera Pose Refinement. Due to the heavy occlusion between the hand and object, the camera poses (relative to the object) generated by hand tracking tend to be imprecise and noisy, which directly degrades the reconstruction quality. To alleviate this problem, we propose simultaneously optimizing the SDF and camera poses. By regarding camera poses as a set of optimizable parameters, they are gradually refined for lower rendering loss during the network training process.

During the simultaneous optimization, we use the coarse-to-fine strategy for more accurate camera pose refinement, following [Lin et al. 2021a; Park et al. 2021]. Specifically, we weight the positional encoding for $\mathbf{p}$ as $\lambda(\mathbf{p}) = (\mathbf{p} \ldots, w_k(n_s) \cdot sin(2^k \pi \mathbf{p}), w_k(n_s) \cdot cos(2^k \pi \mathbf{p}) \ldots)$ where $k \in [0, L-1]$. In our paper, we set L as 6. $n_s \in [0, N_s]$ denotes the current training step and N_s is the total training steps. w_k is a weight function defined as:

$$w_k(n_s) = \frac{1}{2}[1 - \cos(\mathrm{clamp}(\frac{2\mathrm{n_s L}}{\mathrm{N_s}} - \mathrm{k}, 0, 1) \cdot \pi)]. \tag{9}$$

Starting from $n_s = 0$, the positional encodings are gradually activated. When $n_s = \frac{N_s}{2}$, all λ are enabled. A similar coarse-to-fine strategy is also applied to $\mathbf{d}$. Our results and ablation demonstrate that the simultaneous optimization of camera poses and radiance field significantly improves the reconstruction quality.

4.2.3 Deformation Field. Even though we assume the object is grasped firmly, the relative motion between the hand and object is inevitable. The minor relative motion breaks the rigid body assumption of our multi-view reconstruction and results in artifacts in the reconstructed 3D model. Inspired by [Park et al. 2021], we equip the deformation field network W to our model to compensate for the minor relative motion. The network W maps a 3D location $\mathbf{p}$ from each frame to a point $\mathbf{p'}$ in a canonical space: $W : (\mathbf{p}, \mathbf{e}) \rightarrow \mathbf{p'}$, where $\mathbf{e}$ is the per-frame temporal embedding. With the help of the deformation field, the relative motion between the hand and object can be modeled and optimized along with the reconstruction process, which allows us to accurately recover the object geometry in the canonical space.

Specifically, the motion of each sampled point is represented by a rigid transformation in SE(3). A simple linear network is used to represent W, which takes $\mathbf{p}$ and $\mathbf{e}$ as input and outputs the 6D rotation and translation of $\mathbf{p}$.

4.2.4 Leveraging Semantics. So far, we can reconstruct the SDF of the object and the hand. Two problem remains. First, because the hand often occupies a large image region, uniformly sampling tends to optimize a high-quality hand to minimize the render loss and pay less attention to the object, which leads to low object reconstruction quality, particularly for small objects. Second, the recovered SDF contains both the hand and the object, which requires post-processing to separate the object from the hand. We leverage semantic information to solve the above problems.

Semantics-guided sampling. Since the uniform ray sampling strategy will make the network optimize less for the object part of the SDF, we use the semantic maps to guide ray sampling to focus more on the object than the hand. Specifically, during the training process, we gradually decrease the number of sampled rays in the hand areas and increase the number in object areas. We sample 80% of rays from the object region and 20% of the hand and background region. Such a training strategy can significantly improve the sampling efficiency and reconstruction quality of the object, while maintaining the high quality of hand mesh and avoiding irregular SDF values for regions that lack supervision.

Semantic head. Inspired by semantic-nerf [Zhi et al. 2021], we add an additional semantic head to the network F, which enables F to predict 3D semantic labels: $[s(\mathbf{p}), l(\mathbf{p}), c(\mathbf{p}, \mathbf{d})] = F(\mathbf{p}, \mathbf{d})$, where l is the semantic logits. Per-ray semantic logits $\hat{L}$ can be rendered similar to color rendering Equation (6):

$$\hat{L} = \int_{z_n}^{z_f} \omega(z) l(\mathbf{p}(z), \mathbf{d}) dz. \tag{10}$$

The 3D semantic prediction is rendered and supervised by the 2D semantic maps.

4.2.5 Training Loss. We utilize multiple loss terms to optimize our hand-object SDF. To compute the loss, we sample N_r rays on each image and N_p points along each ray. The total loss function is defined as:

$$L = L_c + \lambda_m L_m + \lambda_e L_e + \lambda_l L_l + \lambda_d L_d. \tag{11}$$

L_c is a color rendering loss between rendered pixel colors $\hat{C}$ and true values C of sampled rays in training images:

$$L_c = \frac{1}{N_r} \sum_i ||C_i - \hat{C}_i||_2^2. \tag{12}$$

The Eikonal term [Gropp et al. 2020] L_e and mask term L_m are used to regularize the SDF:

$$L_e = \frac{1}{N_r N_p} \sum_{i,j} (|\nabla s(\mathbf{p}_{i,j})| - 1)^2, \tag{13}$$

$$L_m = \mathrm{CE}(\mathrm{M_i}, \hat{\mathrm{O}}_\mathrm{i}), \tag{14}$$

where the CE denotes the cross-entropy loss. M_i is the mask value and $\hat{O}_i$ is the sum of weight $\omega(t)$ along the ray. We set λ_e as 0.1, same with NeuS, but use a larger $\lambda_m = 5.0$ to remove the influence of inconsistent motion between the foreground and background. L_l is the semantic logits rendering loss, which uses a multi-class cross-entropy loss to optimize the semantic branch:

$$L_l = \mathrm{CE}(\mathrm{L_i}, \hat{\mathrm{L}}_\mathrm{i}), \tag{15}$$

where L_i denotes the ground-truth semantic labels. λ_l is set as 0.1 in experiments. L_d is a deformation regularization term proposed in [Tretschk et al. 2021]:

$$L_d = \frac{1}{N_r N_p} \sum_{i,j} \rho(\mathbf{p}_{i,j}) \cdot |\mathrm{div}(\mathbf{p}'_{\mathrm{i,j}} - \mathbf{p}_{\mathrm{i,j}})|^2, \tag{16}$$

where λ_d is set as 10.0 empirically.

4.2.6 Post-processing. Given the learned neural implicit field F, the 3D geometry of hand and object can be reconstructed from the explicit SDF volume by querying F. Then, the predicted 3D semantic labels of each vertex are used to separate the hand and the object. Finally, Poisson Reconstruction [Kazhdan et al. 2006] is used to fill the holes on the reconstructed object mesh caused by occlusion from the hand.

5 EXPERIMENTS

5.1 Implementation

We use a single NVIDIA TITAN RTX GPU to run the whole reconstruction pipeline. For a one-minute 4K video, the hand motion capture step takes 15 minutes, including 8 minutes for hand 2D keypoint estimation and 7 minutes for MANO fitting. Then, the video is downsampled from 30fps to 6fps to perform the segmentation and reconstruction. The segmentation networks take 6 minutes. The neural SDF reconstruction requires 30 hours to converge. Our implementation is based on the PyTorch [Paszke et al. 2019] library.

5.2 Dataset

Since no existing dataset can satisfy our setting, we collect a new 3D object reconstruction dataset, called Hand-held Object Dataset (HOD). The HOD dataset contains 35 objects, which is divided into two subsets named *Sculptures* and *Daily Objects*. The *Sculptures* has five human sculptures with complex geometries and pure white textures. The *Daily Objects* consists of 30 daily objects with various shapes and appearances. All of the *Sculptures* and nine of the *Daily Objects* are paired with high-fidelity scanned meshes as ground truth geometries for evaluation.

5.3 Evaluations

5.3.1 Metrics and Baseline. We use Chamfer Distance (CD) as the metric for quantitative evaluation. Since the reconstructed and the ground-truth mesh lay in different coordinate systems, we normalize each mesh to unit size and register the reconstructed mesh to the ground-truth mesh using the point-to-point iterative closest point (ICP) algorithm.

We compare our method against the state-of-the-art multi-view reconstruction methods and learning-based hand-object reconstruction methods. We use NeuS [Wang et al. 2021a] as the multi-view

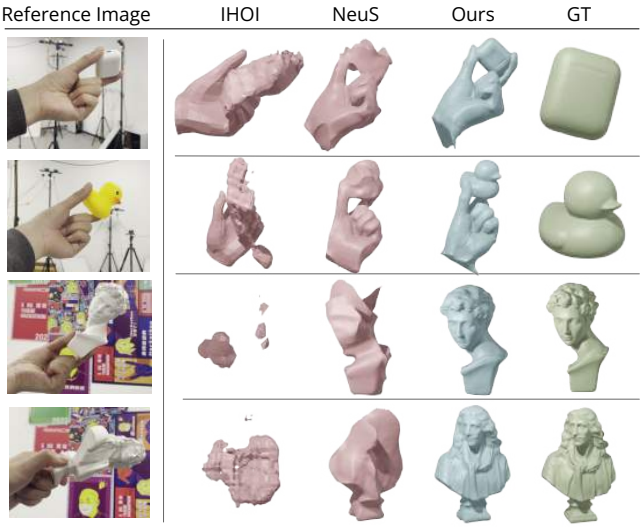

Figure 3: Qualitative results of hand-held object reconstruction. We show the joint reconstruction of the hand and the object for the top two examples, and the segmented object meshes for the bottom two examples. IHOI [Ye et al. 2022] is a learning-based single-view reconstruction method with the reference frame as input. NeuS [Wang et al. 2021a] uses the same video frames and camera poses as ours as input. *The full figure with more examples and compared methods is in Appendix D.8.*

baseline method which our method is built upon. The original NeuS uses COLMAP to obtain camera poses which does not work in our setting, so we feed the camera poses estimated from our hand tracking stage to NeuS. To the best of our knowledge, there is no previous learning-based method that reconstructs hand-object interactions from a monocular video. We then compare our method against ObMan [Hasson et al. 2019], GF [Karunratanakul et al. 2020] and IHOI [Ye et al. 2022], all of which reconstruct hand-object interactions from a single image using deep neural networks. Since learning-based methods take a single image as input, we evaluate them on each frame of the video and report the average accuracy.

5.3.2 Quantitative evaluation. We provide quantitative results for both *Sculptures* and *Daily Objects* subsets in Table 1. The results show that our method outperforms ObMan, GF, and IHOI with much lower CD using the same hand-held object capture video as input. As learning-based methods heavily rely on the learned object shape prior to predicting the 3D object geometry, they don't work well for objects beyond the training dataset. NeuS obtains more reasonable results than the single-view prediction methods but still fails to reconstruct accurate shapes as the original NeuS cannot handle the imprecise pose estimation and relative hand-object motion in our problem. Our approach outperforms the baseline method by a large margin, which validates the effectiveness of the three proposed components.

5.3.3 Qualitative evaluation. We show the qualitative results of hand-held object reconstruction in Figure 3 (full figure in Appendix D.8). NeuS can generate the approximate shape of the target object

Table 1: Quantitative results of object reconstruction. The metric is the Chamfer Distance.

ID	ObMan	GF	IHOI	NeuS	Ours
Orange	3.426	158.735	12.055	3.963	**0.304**
Plastic Box	1.969	24.645	3.866	1.515	**0.433**
Rubber Duck	3.984	59.713	7.676	2.035	**0.521**
Robot	2.250	27.589	4.703	1.354	**0.207**
Cat	8.808	56.905	15.334	11.192	**0.225**
AirPods	0.721	24.561	6.978	1.081	**0.083**
Bottle	0.711	271.218	3.995	0.875	**0.293**
Case	2.438	28.630	7.065	0.929	**0.242**
Pingpong	8.588	112.620	12.759	4.381	**0.408**
Apollo	3.387	45.322	4.255	8.724	**0.164**
David	1.806	40.947	3.652	3.348	**0.191**
Giuliano	1.448	20.794	4.139	0.873	**0.094**
Marseille	3.012	60.904	4.698	1.918	**0.181**
Moliere	1.737	28.963	4.054	5.290	**0.145**
mean	3.163	68.682	6.802	3.391	**0.249**

but with many sharp artifacts. This is mainly due to the imprecise camera poses and not considering the small relative motion between the hand and object. The learning-based method IHOI cannot reconstruct unseen objects and generate invalid mesh for the given image. In contrast, our reconstructed meshes are more fidelity than IHOI and NeuS.

5.3.4 Comparison with static object capture. To demonstrate the reconstruction ability of our proposed hand-held object capture pipeline, we conduct additional comparative experiments with COLMAP [Schönberger et al. 2016] and NeuS [Wang et al. 2021a] in the *static object capture* setting. Specifically, we capture an extra one-minute 4K video for each object by placing the object on a flat table and slowly moving the camera surrounding the object to capture the video. Textured papers are put beneath the object to provide extra textures for more accurate camera pose estimation of SfM.

The qualitative and quantitative results are shown in Figure 4 and Appendix Table 3. Even with hand-held capturing, our method obtains higher object reconstruction quality than COLMAP with a static capture setting. However, our method cannot recover as many fine details as NeuS. The reason is that the input of NeuS has accurate camera poses and little occlusion. In contrast, our input videos contain heavy hand-object occlusions, and it is hard to recover accurate camera poses. Nevertheless, this comparison shows that our pipeline is able to achieve competitive reconstruction quality that is close to static object capture, while the data capture process is more user-friendly.

5.3.5 Robustness. We demonstrate the robustness of our method with different grasping gestures and different motion speeds in Appendix D.1 and D.2.

5.4 Ablation Study

In this paper, we propose several modifications to NeuS to improve the reconstruction quality. We demonstrate their effectiveness by sequentially adding them to the vanilla NeuS model. As shown

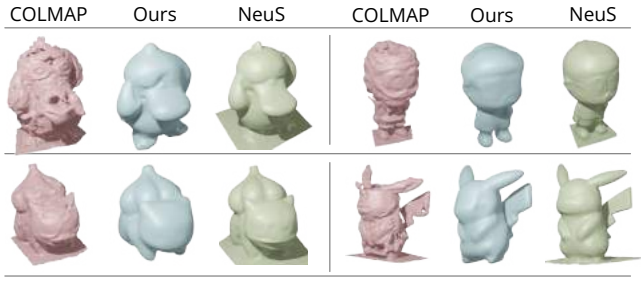

Figure 4: Comparison with static object capture. Note that the inputs to COLMAP and NeuS are videos of static object capture by putting the object on a table, while our inputs are videos of hand-held object capture by moving the object in front of the camera.

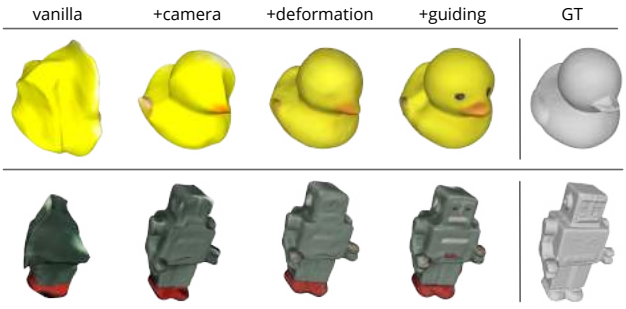

Figure 5: Qualitative results of ablation study. *vanilla* indicates the mesh reconstructed with original NeuS, *+camera* indicates the camera refinement introduced in Section 4.2.2, *+deformation* indicates the deformation field introduced in Section 4.2.3, and *+guiding* indicates the semantics-guided sampling introduced in Section 4.2.4.

in Figure 5, the vanilla NeuS suffers from incorrect camera poses due to imprecise hand pose tracking. By simultaneously optimizing the camera poses, the optimized SDF improves but still contains sharp artifacts. By adding the deformation field, the network F can capture relative motion between the hand and object, which removes the irregular sharp edges. Finally, the semantics-guided sampling makes the network more focused on the object instead of the hand, enabling more object details to be reconstructed. Please see Appendix D.6 for quantitative results.

6 LIMITATIONS AND FUTURE WORK

In this paper, we showed that it was possible to reconstruct a moving hand-held object which may be textureless from a monocular video, without using any shape prior and training data of the object. There are some limitations of this work. First, Poisson Reconstruction is used as post-processing for hole filling, which may not be suitable for large holes or thin structures. More sophisticated learning-based shape completion methods could be used. Second, depth-based in-hand capture allows the user to see the reconstruction in real-time, while our method takes hours to reconstruct the object. Leveraging recent techniques [Chen et al. 2022; Müller et al. 2022] to accelerate our method is left as future work. Finally, our method assumes a fixed camera, a single grasping gesture, and small relative motion between hands and objects. Extending our method to relax these constraints would be an interesting direction to explore.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from ZJU-SenseTime 3D Vision Lab and the Open Project Program of the State Key Lab of CAD&CG (No.A2213), Zhejiang University.

REFERENCES

Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 2019. 3d hand shape and pose from images in the wild. In *CVPR*.

Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *The European Conference on Computer Vision (ECCV)*.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Anpei Chen, Zexiang Xu, Andreas Geiger, , Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. arXiv:2203.09517 [cs.CV]

Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. 2020. Learning canonical shape space for category-level 6d object pose and size estimation. In *CVPR*.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*.

Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *CVPR*.

Julian Chibane, Gerard Pons-Moll, et al. 2020. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems* 33 (2020).

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*.

Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *CVPR*.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *CVPR*.

Yasutaka Furukawa and Jean Ponce. 2007. Accurate, dense, and robust multiview stereopsis. *CVPR* (2007).

Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D Hand Shape and Pose Estimation from a Single RGB Image. In *CVPR*.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *ICML*.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*.

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. 2020. HOnnotate: A Method for 3D Annotation of Hand and Object Poses. In *CVPR*.

Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. 2020. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 87–1.

Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

Eldar Insafutdinov and Alexey Dosovitskiy. 2018. Unsupervised Learning of Shape and Pose with Differentiable Point Clouds. In *NeurIPS*.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 559–568.

Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. 2018. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In *ECCV*.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Recognition (CVPR)*.

Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael Black, Krikamol Muandet, and Siyu Tang. 2020. Grasping Field: Learning Implicit Representations for Human Grasps. (2020).

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Vol. 7.

Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. 2019. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *ICCV*.

Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. 2020. Weakly-Supervised Mesh-Convolutional Hand Reconstruction in the Wild. In *CVPR*.

Kiriakos N Kutulakos and Steven M Seitz. 2000. A theory of shape by space carving. *International journal of computer vision* 38, 3 (2000), 199–218.

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021a. BARF: Bundle-Adjusting Neural Radiance Fields. In *IEEE International Conference on Computer Vision (ICCV)*.

Fanqing Lin, Connor Wilhelm, and Tony Martinez. 2021c. Two-Hand Global 3D Pose Estimation Using Monocular RGB. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2373–2381.

Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021b. End-to-End Human Pose and Mesh Reconstruction with Transformers. In *CVPR*.

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. 2020. Real-Time High-Resolution Background Matting. *arXiv* (2020), arXiv–2012.

William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH* (1987).

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* (2004).

Priyanka Mandikal, KL Navaneet, Mayank Agarwal, and R Venkatesh Babu. 2018. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. *arXiv preprint arXiv:1807.07796* (2018).

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *European Conference on Computer Vision (ECCV)*.

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. 2019. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. *ICCV* (2021).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *SIGGRAPH Asia* 36, 6 (Nov. 2017).

Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. 2002. Real-time 3D model acquisition. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 438–446.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *CVPR*.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*.

Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *CVPR*.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*.

Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer. 2016. Real-time monocular segmentation and pose tracking of multiple objects. In *ECCV*.

Henning Tjaden, Ulrich Schwanecke, and Elmar Schomer. 2017. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In *ICCV*.

Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)* 33, 5 (2014), 1–10.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Dimitrios Tzionas and Juergen Gall. 2015. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*. 729–737.

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. 2019. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021a. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *arXiv preprint arXiv:2106.10689* (2021).

Weiyue Wang, Qiangui Huang, Suya You, Chao Yang, and Ulrich Neumann. 2017. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. In *ICCV*.

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. 2021b. NeRF−−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064* (2021).

Thibaut Weise, Bastian Leibe, and Luc Van Gool. 2008. Accurate and robust registration for in-hand modeling. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. 2019. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 2019. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In *NeurIPS*.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *European Conference on Computer Vision (ECCV)* (2018).

Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. 2022. What's in your hands? 3D Reconstruction of Generic Objects in Hands. (2022).

Hao Zhang, Zi-Hao Bo, Jun-Hai Yong, and Feng Xu. 2019. InteractionFusion: real-time reconstruction of hand poses and deformable objects in hand-object interactions. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–11.

Hao Zhang, Yuxiao Zhou, Yifei Tian, Jun-Hai Yong, and Feng Xu. 2021. Single Depth View Based Real-Time Reconstruction of Hand-Object Interactions. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–12.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. 2021. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Christian Zimmermann and Thomas Brox. 2017. *Learning to Estimate 3D Hand Pose from Single RGB Images*. Technical Report. arXiv:1705.01389. https://lmb.informatik.uni-freiburg.de/projects/hand3d/ https://arxiv.org/abs/1705.01389.

Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russel, Max Argus, and Thomas Brox. 2019. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. In *ICCV*. "https://lmb.informatik.uni-freiburg.de/projects/freihand/"

A DATASET DETAILS

Figure 6: HOD dataset. The left figure shows five textureless *Sculptures* with complex geometry details. The right figure shows some examples of the *Daily Objects* in the HOD dataset, which contains various shapes and textures.

An overview of the dataset objects is given in Figure 6. To inspire future hand-held object reconstruction works, each object is captured with extra videos for two additional settings: 1) **Hand-held Object Reconstruction with Different Gestures**. The classic static object scanning recovers the bottom surface of the object by scanning the object a second time with a different object placement. Likewise, for hand-held object reconstruction, it should be possible to reconstruct the occluded region by changing the position of contact points. We then capture an additional video for each object with a different grasping gesture. The main challenge of solving this setting is building the accurate correspondence between two partial object meshes and fusing two slightly different meshes. 2) **Hand-held Object Reconstruction with Large Hand-object Relative Motion**. Unlike our work, a more user-friendly setting should allow significant relative motion between the hand and object. Thus, we capture a video with the large hand-object relative motion for each object. To solve this setting, future work should be able to infer the object pose by human manipulation and fuse the temporal observations to a complete object mesh.

There are several datasets of hand-object interactions. HO3D [Hampali et al. 2020] and ContactPose [Brahmbhatt et al. 2020] are the representatives. Compared to them, our dataset is more suitable for the task of hand-held object reconstruction for several reasons: First, both ContactPose and HO3D use multiple cameras while each camera only sees a part of the object, which makes HO3D and ContactPose not suitable for the monocular reconstruction setting. Second, compared with HO3D, the objects in our dataset are firmly grasped by users, while HO3D allows significant relative motion between the hand and object. Third, compared with ContactPose, our dataset has more realistic objects. ContactPose uses 3D-printed replica objects with no texture. Our dataset contains objects from daily-life scenarios with various shapes and appearances.

B DETAILS OF AUXILIARY NETWORK TRAINING

B.1 Hand 2D/MANO network training

The input video of our method is captured in the egocentric view and contains strong occlusions due to the hand-object interactions. However, off-the-shelf 2D hand keypoints networks [Simon et al. 2017] and MANO estimation networks [Kulon et al. 2020; Lin et al. 2021b] are mostly trained for third-person perspective and no hand-object interactions. To solve this problem and improve the hand tracking performance, we propose to train new hand networks with a mixture of egocentric-view and hand-object interaction datasets. Specifically, we use FreiHand [Zimmermann et al. 2019], InterHand [Moon et al. 2020], MTC [Xiang et al. 2019], RHD [Zimmermann and Brox 2017], Ego3D [Lin et al. 2021c] and HO3D [Hampali et al. 2020] to improve the hand tracking performance. For the network architecture, we use ResNet50 as the 2D hand keypoints estimation backbone network, and HMR [Kanazawa et al. 2018] as the MANO estimation backbone network.

B.2 Hand segmentation network training

Since none of the current hand segmentation networks performs well on egocentric-view and hand-object interaction scenarios, we collect and annotate a new hand segmentation dataset. The dataset contains 3000 egocentric hand-object interaction images with manually labeled hand segmentation labels. For the network architecture, we choose the DeepLabv3+ [Chen et al. 2018] as our backbone network.

C HAND TRACKING WITH PARAMETERS SHARING

Figure 7: Effectiveness of parameters sharing during hand tracking. We compare our hand tracking system with and without sharing hand pose ($\boldsymbol{\theta}$) and shape ($\boldsymbol{\beta}$) parameters. Without sharing hand parameters, the hand tracking module easily tracks wrong hand poses due to heavy hand-object occlusions. In contrast, by sharing hand parameters across the entire video, the tracking module is able to get the correct hand poses even in extreme cases.

Here we compare our hand tracking method with and without parameters sharing. As mentioned in the Section 4.1, sharing parameters uses the same pose ($\boldsymbol{\theta}$) and shape ($\boldsymbol{\beta}$) parameters for each frame. Such parameters sharing manner makes the hand tracking process much faster due to fewer optimization parameters and easier convergence. According to our experiments on 1800 frames of video, the fitting process takes about 4600 seconds without sharing the hand parameters. However, the hand fitting process only takes about 420 seconds when the hand parameters are shared.

Another benefit of sharing hand parameters is the tracking robustness under heavy hand-object occlusions. As shown in Figure

7, without sharing hand parameters, the hand tracking method is sensitive to keypoints outliers and often fails to produce correct hand poses due to the heavy occlusion. However, by sharing hand parameters across the whole sequence, the tracking process can use other frames to correct wrong estimations, resulting in better tracking results.

D EXPERIMENTS FOR DENSE RECONSTRUCTION

D.1 Results for different motion speeds

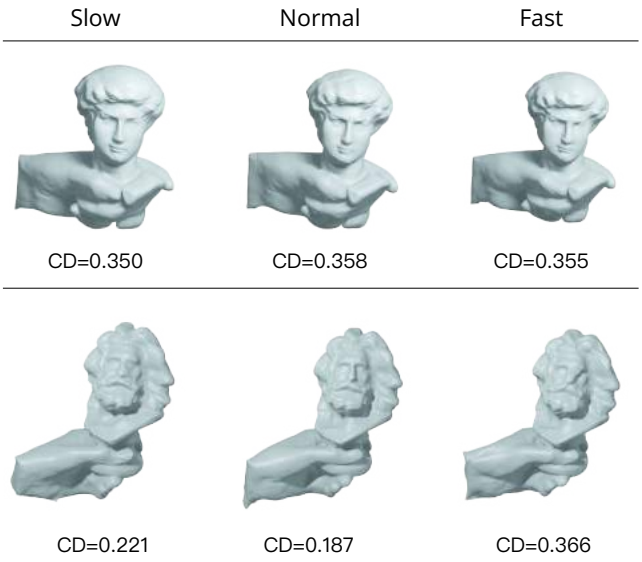

Figure 8: Experiments of different moving speeds. We do experiments to validate the robustness of our method at different grasping motion speeds. Our method is able to generate high-quality 3D geometry for the *Slow* and *Normal*, and only cause small artifacts for the *Fast*. The 'CD' refers to the Chamfer Distance of object reconstruction.

We validate our reconstruction method at different motion speeds. We collect videos at three different speeds with the same grasping gesture. *Normal* is the normal speed at which humans daily interact with objects. *Slow* and *Fast* are half the movement speed and twice the motion speed of *Normal*. The results of the joint hand-object reconstruction are shown in the figure 8. It turns out that our algorithm works well for both the *Slow* and *Normal* speeds. For faster motion, our reconstruction results may cause some artifacts, but are still correct for most regions.

D.2 Results for different grasping gestures

We also validate our reconstruction algorithm with different grasping gestures. As shown in Figure 9, different grasping gestures lead to varying levels of occlusions. No matter how the hand holds the object, our method is capable of generating high-fidelity joint hand-object reconstruction meshes since we do not use any gesture-related prior or assume specific grasping types. However, the object-only reconstruction contains obvious artifacts when heavy occlusions happen. The reason is that we use Poisson Reconstruction as post-processing for hole filling. When a large part of the object is occluded or covered by the hand, the Poisson Reconstruction cannot fill the correct surface for the missing part, resulting in artifacts, see *Strong Occlusion* row in Figure 9. In conclusion, gestures are irrelevant for joint hand-object reconstruction. However, for object-only reconstruction, weak occlusion gestures make it easier for Poisson Reconstruction to fill in the holes.

Table 2: Quantitative comparison with GT-selected meshes. The metric is the Chamfer Distance between the reconstructed and the ground-truth mesh.

ID	ObMan	GF	IHOI	Ours
Orange	0.814	1.080	1.936	**0.304**
Plastic Box	**0.282**	0.605	0.575	0.433
Rubber Duck	1.926	2.504	1.794	**0.521**
Robot	1.028	1.055	1.010	**0.207**
Cat	4.891	6.128	3.667	**0.225**
AirPods	0.314	0.416	0.377	**0.083**
Bottle	**0.245**	0.408	0.303	0.293
Case	0.641	0.957	0.757	**0.242**
Pingpong	1.869	5.286	2.567	**0.408**
Apollo	0.784	1.371	0.788	**0.164**
David	0.849	1.262	0.650	**0.191**
Giuliano	0.481	0.589	0.525	**0.094**
Marseille	1.217	1.175	1.118	**0.181**
Moliere	0.786	0.800	0.597	**0.145**
mean	1.152	1.688	1.190	**0.249**

D.3 Comparison with GT-selected meshes

To demonstrate the quality of our reconstructed meshes, we also compare them with the meshes of learning-based methods selected by the ground-truth. Specifically, we run the learning-based hand-object reconstruction method per frame, align them with the

Table 3: Quantitative comparison with temporal smoothed meshes. The metric is the Chamfer Distance between the reconstructed and the ground-truth mesh.

ID	ObMan	GF	IHOI	Ours
Orange	0.552	1.454	2.314	**0.304**
Plastic	7.325	0.761	1.208	**0.433**
Rubber Duck	3.330	2.662	2.390	**0.521**
Robot	1.862	1.007	2.236	**0.207**
Cat	10.605	5.900	6.226	**0.225**
AirPods	0.210	0.338	0.432	**0.083**
Bottle	0.263	1.258	**0.281**	0.293
Case	1.586	1.972	1.967	**0.242**
Pingpong	4.262	4.471	1.756	**0.408**
mean	3.332	2.203	2.090	**0.302**

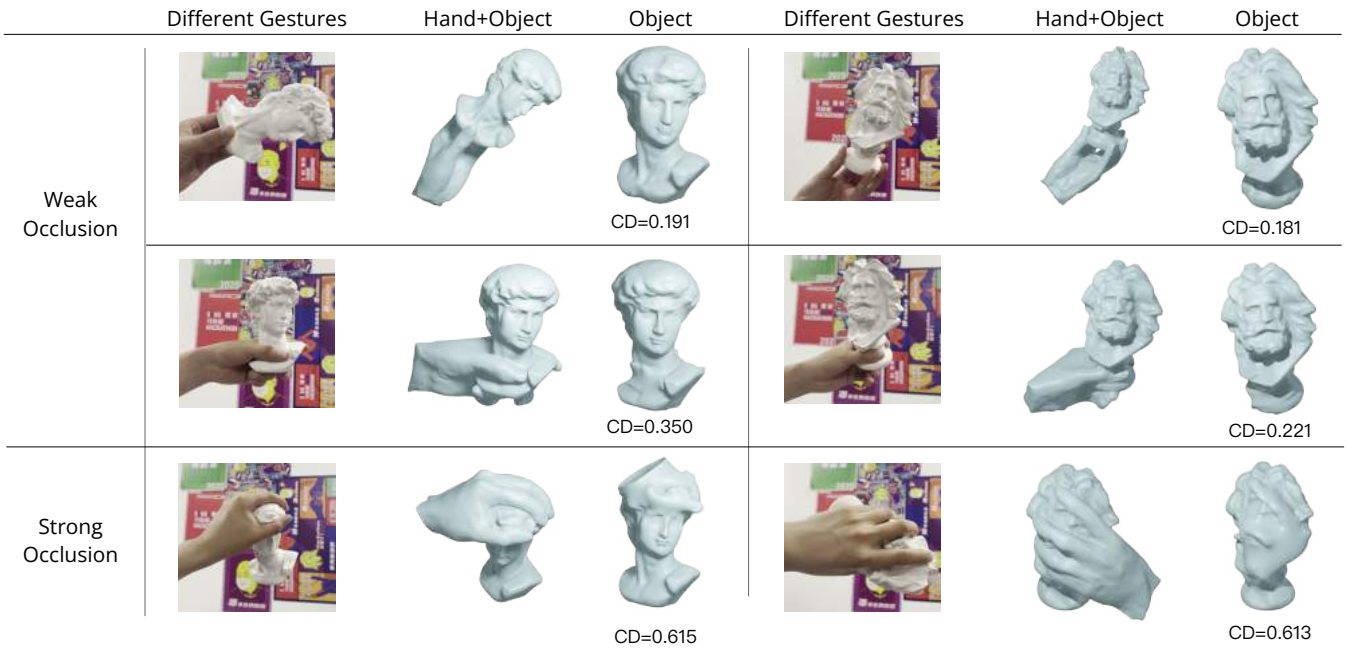

Figure 9: Experiments of different grasping gestures. We conduct experiments to verify whether grasping gestures affect the final reconstruction results. When the grasping gesture covers only a small part of the object, both joint hand-object reconstruction and final object-only results are high-fidelity. Our method can still generate high-quality joint hand-object reconstruction results when grasping gestures cover a large portion of the object. However, the limitations of our post-processing make it challenging to fill in holes under heavy occlusions. The 'CD' refers to the Chamfer Distance of object reconstruction.

ground-truth by ICP, calculate the Chamfer Distance, and select the most similar mesh with the ground-truth. The table 2 shows that even compared with the GT-selected meshes of learning-based approaches, our method still gets a lower Chamfer Distance by a large margin. The only exception is the *Plastic Box* and *Bottle* compared with ObMan. We argue that the reason is those two objects are cylinder-like. According to our experiments, ObMan is prone to generate cylinder-like geometries. For this reason, ObMan gains better Chamfer Distance results for the two instances. However, ObMan heavily relies on learned object prior and often generates cylinder-like shapes for every unseen object, thus cannot generate correct geometry for other instances. In contrast, our method can generalize to various types of objects and gain much better performance, especially for those with complex geometries or low texture information.

D.4 Comparison with temporal smoothed meshes

A simple strategy to boost the performance of learning-based methods is leveraging temporal coherence. We apply temporal smoothing to learning-based methods and compare them with the proposed pipeline. Specifically, we run learning-based methods to reconstruct the object mesh for each frame. The reconstructed object meshes are then aligned with the ground-truth mesh by ICP and voxelized to 3D occupancy volumes. We average the 3D occupancy volume across the whole temporal sequence, then set a threshold and use marching cubes to extract the final mesh. The quantitative results are shown in Table 3. By leveraging temporal coherence, the learning-based methods are able to fuse multi-frame observations and get better performance. Compared with learning-based methods with temporal coherence, our method still gets a better reconstruction quality and lower Chamfer Distance by a large margin. The only exception is the *Bottle*, which has a cylinder-like shape whose shape prior is easier to learn by learning-based methods (Discussed in Section D.3).

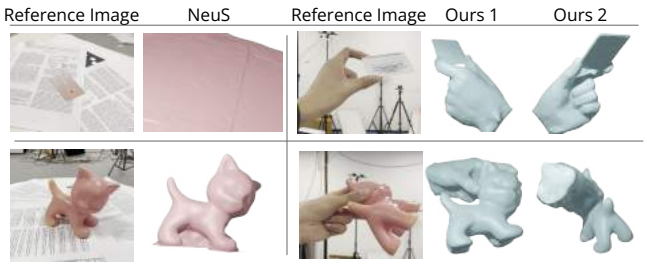

Figure 10: Comparison with static object capture. The proposed setting is able to reconstruct objects that cannot stand vertically, or the bottom surfaces are heavily occluded.

Table 4: Quantitative comparison with static object capture. The metric is the Chamfer Distance between the reconstructed and the ground-truth mesh.

ID	COLMAP	NeuS	Ours
Orange	0.260	**0.232**	0.304
Plastic	0.350	**0.062**	0.433
Rubber Duck	1.917	**0.326**	0.521
Robot	**0.118**	0.123	0.207
Cat	1.049	**0.173**	0.225
AirPods	0.895	0.228	**0.083**
Bottle	0.043	**0.032**	0.293
Case	0.483	**0.172**	0.242
Pingpong	1.081	**0.037**	0.408
mean	0.688	**0.154**	0.302

Table 5: Quantitative results of ablation study. The metric is the PSNR of the target object region.

ID	vanilla	+camera	+deformation	+guiding
Orange	20.082	21.897	22.576	**28.756**
Plastic	17.156	18.260	18.989	**25.215**
Rubber Duck	18.453	19.474	21.307	**28.903**
Robot	17.083	18.522	19.506	**24.857**
Cat	17.797	20.456	20.490	**26.456**
AirPods	17.945	20.300	21.819	**28.427**
Bottle	17.804	18.655	19.123	**25.804**
Case	19.323	20.649	21.096	**28.468**
Pingpong	17.277	18.956	19.627	**28.508**
mean	18.102	19.685	20.504	**27.266**

D.5 Comparison with static object capture

In this paper, we propose a novel object reconstruction manner by grasping and moving the object in front of the camera. Compared with classic static object capture, the proposed method has several potential benefits: First, our method is able to reconstruct objects that cannot be easily placed on the floor or table. For example, the *card* from our HOD dataset is impossible to stand vertically on the floor. As shown in the first row of Figure 10, NeuS with static object capture cannot reconstruct the card and outputs a plane for the whole table. However, our method is able to reconstruct the *card* by grasping it in hand. Second, static object capture requires placing the object on a plane, which makes it hard to reconstruct the bottom surface. As shown in the second row of Figure 10, static object capture cannot reconstruct the bottom part of the *cat* due to occlusion, while our setting allows users to scan the object more freely (as long as the object can be grasped).

D.6 Effectiveness of the three proposed modules

We evaluate the PSNR metric in Table 5 as a supplement of Section 5.4. The higher PSNR value indicates a better novel view quality and mesh details. According to the table, the three solutions show their ability to solve the three main issues that degrade the reconstruction quality.

D.7 Effectiveness of camera pose initialization

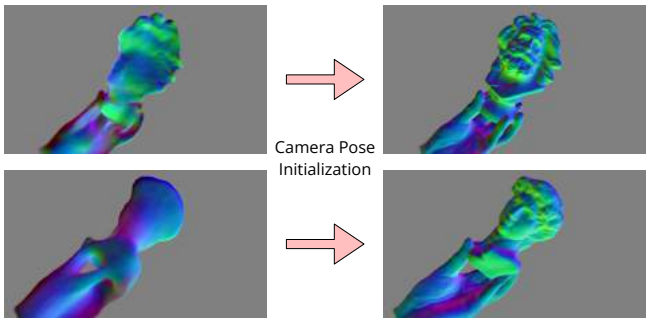

Figure 11: Effectiveness of camera pose initialization. We use hand tracking to initialize the camera poses of dense reconstruction. Here we compare the reconstructed normal map with and without camera pose initialization.

We validate our dense reconstruction performance with and without using hand tracking as the camera pose initialization. The reconstructed normal map is shown in Figure 11. Even if our dense reconstruction stage can optimize the camera poses during the optimization phase, it leads to coarse and blurred meshes without camera pose initialization. This is because the camera refinement requires a good initialization to achieve correct camera pose refinement. With hand tracking, our method is able to generate more detailed object geometry.

D.8 More results

We give a full figure of method comparison in Figure 12, and more reconstructed results in Figure 13.

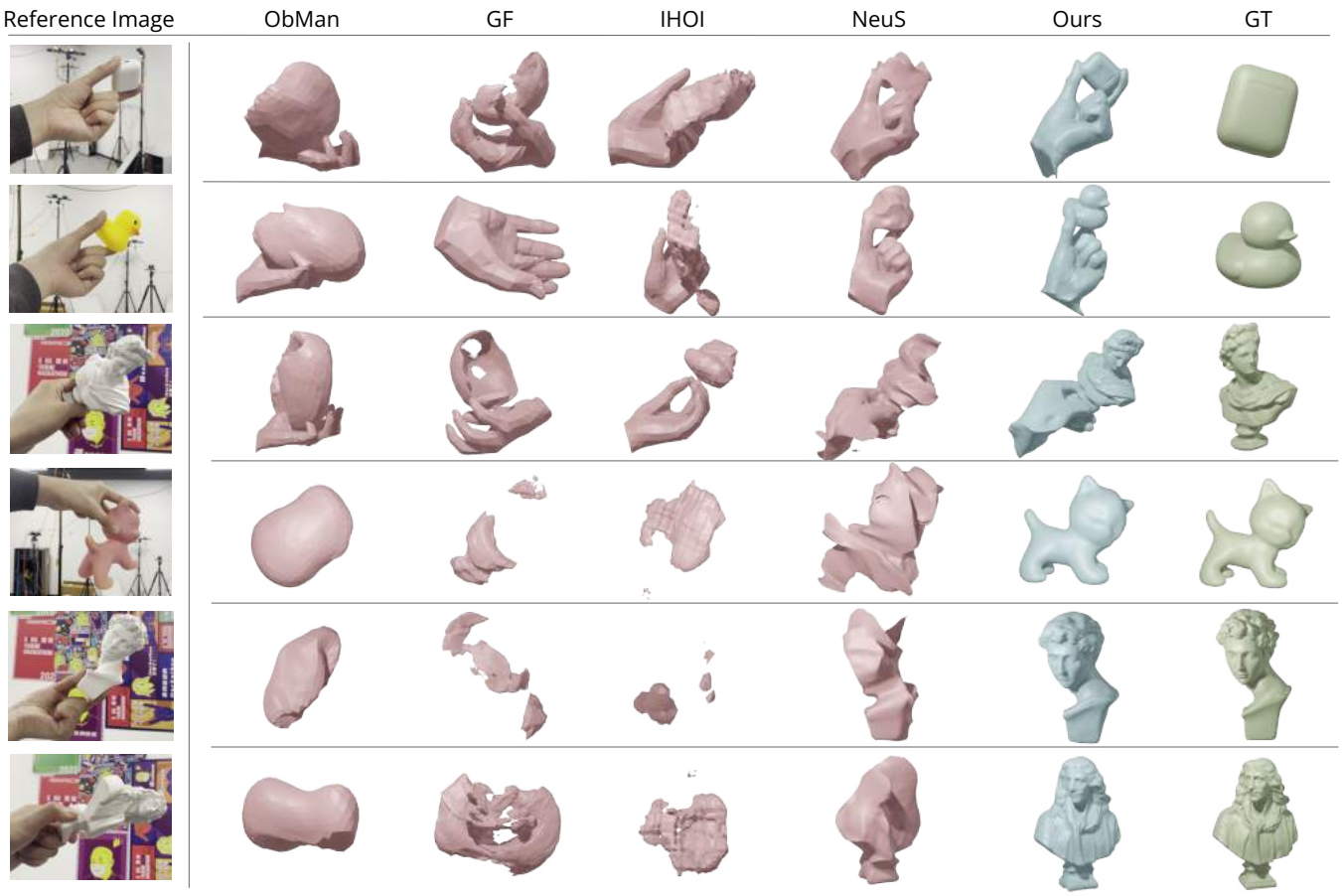

Figure 12: A full figure of method comparison. The top three rows are the joint reconstruction of the hand and object, and the bottom three rows are the separated objects. Learning-based methods ObMan [Hasson et al. 2019], GF [Karunratanakul et al. 2020] and IHOI [Ye et al. 2022] cannot generalize to unseen objects; NeuS generates wrong shapes with sharp artifacts. Our method can recover high-quality meshes for both joint hand-object reconstruction and separated object reconstruction.

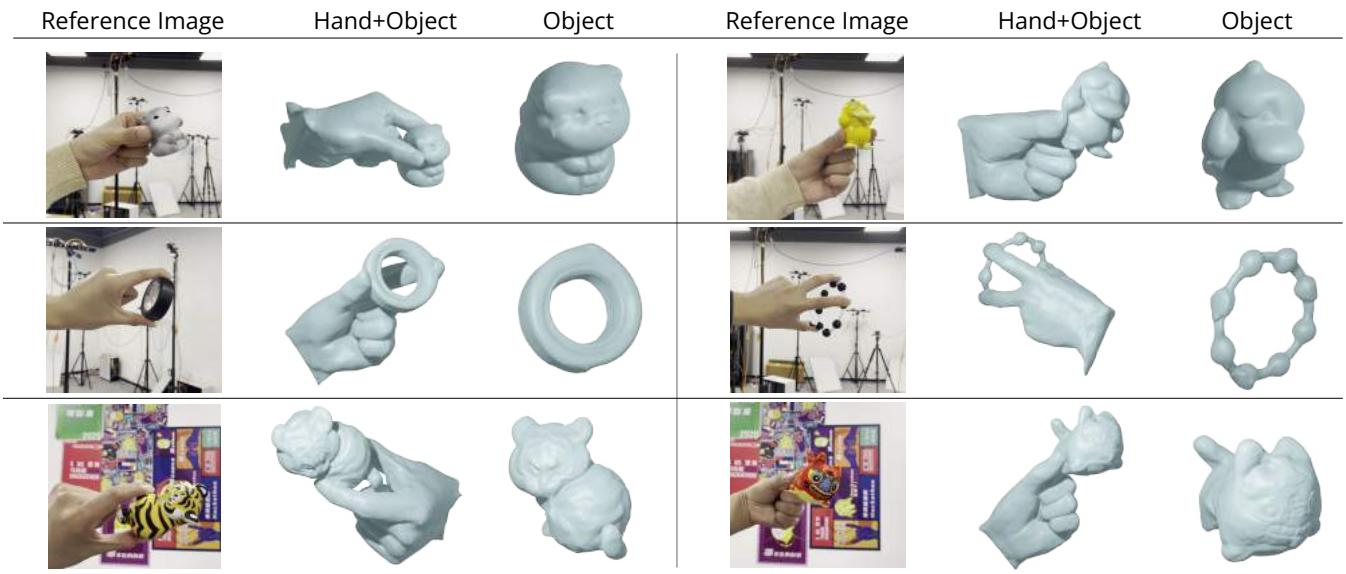

Figure 13: More results of our hand-held object reconstruction method.