HOLD: Category-agnostic 3D Reconstruction of Interacting Hands and Objects from Video

Zicong Fan[1,2] Maria Parelli[1] Maria Eleni Kadoglou[1]
Xu Chen[1,2,†] Muhammed Kocabas[1,2] Michael J. Black[2] Otmar Hilliges[1]
[1]ETH Zürich, Switzerland [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany

The SupMat, comprising this document and a video, along with our code and pre-trained models, is available here. We will post updates and project news on this link.

A. Pose initialization

Hand pose estimation: To initialize the hand pose, we first crop around the hand with the hand detector from 100DoH [20], and use an off-the-shelf hand regressor, METRO [15], to infer the MANO hand mesh. We empirically found that METRO, a non-parametric method, is more stable in predicting hand global rotation during object occlusion than parametric regressors. To obtain MANO parameters for each estimated hand mesh with MANO topology, we register the MANO model to the predicted meshes. For each frame in a sequence, we compute the volume of the watertight MANO mesh. We then fit a 1D Gaussian to the volumes for each sequence. We remove frames with volumes that are smaller than the mean of the Gaussian by 2.0 standard deviations or more. Although it is a simple heuristic, we found this method to be effective in removing degenerate hand mesh predictions because non-parametric hand regressors tend to produce meshes with small volumes in degenerate cases. We use spherical interpolation (SLERP) to infill MANO poses for missing frames. We experiment with 2D keypoint detectors such as OpenPose [21] and MediaPipe [25], but they are often very noisy during object occlusion. Instead, we project the initial hand mesh from METRO to image space and use this as the 2D keypoints. We found that the 2D keypoints from METRO are more stable than those from existing 2D keypoint detectors.

Object pose estimation: Since we focus on a category-agnostic setting, most existing object pose estimators [2] are unsuitable for our setting because they are designed for specific categories. To obtain initial object pose estimates of a novel object without category-level supervision, we perform structure-from-motion. We first obtain object masks from an off-the-shelf segmentation network [6]. The masks are used to create images with object-only pixels for structure-from-motion (SfM). We use HLoc [18] for SfM with SuperGlue [19] and SuperPoints [7] and 40 keypoints for multi-view matching. Since not all frames provide object poses, we use SLERP to infill missing frames. SfM often provides noisy point clouds. Therefore, we clean each point cloud automatically by first computing its median center, and the distance of the 20-percentile point to the center. Points that are outside of $1.5\times$ of this distance are removed. We then subtract the point cloud with its median to center the point cloud around zero. After centering the point cloud, we normalize it such that the point cloud radius is 1.0. Similar to hand 2D keypoints, we project 3D points from this point cloud to obtain 2D object keypoints for the alignment.

Hand-object alignment: Since SfM only reconstructs the point clouds up to a scale, we need to figure out the scale (size factor) s of an object template. This is important because the object's translation, denoted as $\mathbf{t}_o$, is calculated in the camera's coordinate system assuming the scale is 1. Further, the hand's translation, represented by $\mathbf{t}_h$, is in the coordinate system of the cropped image around the hand. Both these poses need to be aligned to the same space and share the same camera intrinsic matrix $\mathbf{K}$.

To align the hand and the object in a shared camera coordinate space, we optimize an energy function,

$$\mathcal{E} = \sum_t \mathcal{E}_h + \mathcal{E}_o + \omega_{\text{smooth}} \cdot \mathcal{E}_{\text{smooth}} \tag{1}$$

where, for each frame, $\mathcal{E}_h$ is for the hand pose, $\mathcal{E}_o$ is for the object pose, and $\mathcal{E}_{\text{smooth}}$ is for smoothness, weighted by ω_{smooth}. In particular, a hand energy term is defined as

$$\mathcal{E}_h(\mathbf{R}_h, \mathbf{t}_h) = \omega_{2D} \cdot \rho(\mathbf{J}^t - \hat{\mathbf{J}}^t) \tag{2}$$

where $\mathbf{J}^t$ and $\hat{\mathbf{J}}^t$ are the fitted and target 2D keypoints of the hand joints at frame t respectively. The object energy term is defined as $\mathcal{E}_o(s, \mathbf{R}_o, \mathbf{t}_o) =$

$$\omega_{2D} \cdot \rho(\mathbf{X}^t_o - \hat{\mathbf{X}}^t_o) + \omega_z \cdot \left\|-\mathbf{c}^t_{o,z}\right\|^+_1 + \omega_{\text{contact}} \left\|\mathbf{c}^t_h - \mathbf{c}^t_o\right\|_1 \tag{3}$$

where $\mathbf{X}^t_o$ and $\hat{\mathbf{X}}^t_o$ are the fitted and the target 2D projection of the object point cloud at frame t; $\mathbf{c}^t_{o,z}$ is the z-component

† Work done prior joining Google

of the object mean vertex; $\|\cdot\|_1^+$ is a clamped L1 loss that set negative values to 0.0; $\mathbf{c}_h$ and $\mathbf{c}_o$ are the mean of the hand and object meshes. Finally, we have the smoothness term

$$\mathcal{E}_{\text{smooth}}(s, \mathbf{t}_o, \mathbf{t}_h) = \left\|\mathbf{c}_h^t - \mathbf{c}_h^{t+1}\right\|_2^2 + \left\|\mathbf{c}_o^t - \mathbf{c}_o^{t+1}\right\|_2^2. \tag{4}$$

For the 2D projection terms, following [1], we use the Geman-McClure [8] loss $\rho(\cdot)$ with a sigma of 25.0. We empirically choose ω_{smooth} and ω_{contact} to be 100.0, ω_{2D} to be 1.0 and ω_z to be 1000.0 for all sequences.

Intuitively, the hand energy term enforces the fitted hand parameters to have the same 2D projection as the initial pose estimate; the object energy term enforces the same 2D projection as the initial pose, avoids the object center (its z-component) to be behind the camera, and encourages hand-object proximity; the smoothness term maintains the smoothness of the overall sequence.

We use a procedural fitting approach in which we first fit the hand with the hand energy term for 4000 iterations with a learning rate of 0.01, then we freeze the hand parameters and fit the object energy term for 15000 iterations with a learning rate of 0.01. Finally, we allow both hand and object parameters to be optimized and jointly fit for all terms for 3000 iterations with a learning rate of 0.001. The fitting often takes less than 5 minutes on a GeForce RTX 2080 Ti.

B. Pose refinement

As mentioned in the main paper, after training HOLD-Net, we have more accurate hand and object shapes. The goal of this pose refinement is to encourage better hand-object contact and better pixel alignment of the hand and the object by optimizing $\{\mathbf{t}_h, \mathbf{R}_o, \mathbf{t}_o, \beta, s\}$.

Occlusion-aware loss: To enhance the pixel-alignment of the hand and object meshes, we use Soft Rasterizer [16] to render hand and object amodal [14] masks $\mathcal{M}_h$ and $\mathcal{M}_o$. Suppose the hand and object segmentation masks from off-the-shelf prediction are $\hat{S}_h$ and $\hat{S}_o$. Following [26], we use an occlusion-aware mask loss

$$\mathcal{L}_{mask} = \left\|\mathcal{M}_h - \hat{S}_h\right\| \cdot (1 - \hat{S}_o) + \left\|\mathcal{M}_o - \hat{S}_o\right\| \cdot (1 - \hat{S}_h) \tag{5}$$

where all masks are binary masks (1 indicates the corresponding class of interest). Intuitively, when fitting the hand masks, with the term $(1 - \hat{S}_o)$, the loss will not penalize cases where the hand falls on the object part of the image, and vice versa when fitting the object masks.

Refinement details: The total fitting loss $\mathcal{L}_{\text{fitting}}$ is defined as $\mathcal{L}_{\text{fitting}} = \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}$ where we choose λ_{contact} to be 100.0 and λ_{mask} to be 1000.0. Due to limited GPU memory, for a given sequence, we uniformly sample 51 frames, and fit frame-independent parameters (β, s) while allowing other parameters to change. Once (β, s) are initialized, we freeze them and optimize $\{\mathbf{t}_h, \mathbf{R}_o, \mathbf{t}_o\}$ with their initial values on a batch of 51 consecutive frames at a time to fit the entire sequence. We fit 300 iterations for each batch, which takes around 30 minutes to fit a sequence around 250 frames on an A100 GPU.

Object	Sequence name
bleach	ABF12
bleach	ABF14
potted meat	GPMF12
potted meat	GPMF14
cracker box	MC1
cracker box	MC4
power drill	MDF12
power drill	MDF14
sugar box	ShSu10
sugar box	ShSu12
mustard	SM2
mustard	SM4
mug	SMu1
mug	SMu40
banana	BB12
banana	BB13
scissors	GSF12
scissors	GSF13

Table A. HO3D sequences for hand-object reconstruction

C. Experiment details

Inverted sphere parametrization: Our compositional implicit model consists of a hand model, an object model and a dynamic background. Following [9, 27], to query a point (x, y, z) with the background model, we first convert the point to a quadruple

$$\rho(x', y', z', 1/r) \tag{6}$$

where $x'^2 + y'^2 + z'^2 = 1$ and $r = \sqrt{x^2 + y^2 + z^2} > 1$. This quadruple point format is used as the query point for the background model.

Sampling on a ray: We use error-bounded sampling from VolSDF by iteratively upsampling z-values within a specified error bound to obtain 3D points. Sampling stops when this error is below a threshold. The opacity values of these points are used to derive the final samples via inverse transform sampling. See VolSDF [24] for details.

HO3D sequences: We use HO3D-v3 [10] to evaluate our method. The sequences can be found in Table A. Note that the 3D annotations are not used to train our method; we only use the RGB raw sequences for training.

To compare with Hampali *et al.* [11] for in-hand object scanning, since there is no code released for this method, we

Object	Sequence name
35: power drill	MDF14
10: potted meat	GPMF12
3: cracker box	MC1
6: mustard	SM2
4: sugar box	ShSu12
21: bleach	ABF14
25: mug	SMu1
19: pitcher base	AP13

Table B. HO3D sequences for in-hand object scanning

train HOLD on the same sequences (see Table B). For qualitative results, Figure A shows a side-by-side comparison with Hampali *et al.* [11] and the ground-truth meshes. Since Hampali *et al.* [11] only release the reconstructed objects in point cloud format, the figure shows a dense point cloud comparison. Our method captures noticeably more fine-grained details and fewer artifacts compared to Hampali *et al.* [11], and the results are more similar to the ground-truth. For example, the bottle in the middle has hand-like artifacts in the Hampali *et al.* [11] result, while ours does not. This is due to our compositional implicit model that explicitly disentangles hands and objects in object reconstruction.

Intrinsics: For evaluation purposes, we use intrinsics from HO3D; for in-the-wild experiments, we use intrinsics from SfM. Both intrinsics give reasonable results.

Training details: We train each sequence with Adam [13] and randomly optimize 10 images from the sequence at each iteration with the loss:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{segm}}\mathcal{L}_{\text{segm}} + \lambda_{\text{sdf}}\mathcal{L}_{\text{sdf}} \\ + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}} + \lambda_{\text{eikonal}}\mathcal{L}_{\text{eikonal}}. \quad (7)$$

During the first 30k iterations, λ_{segm} linearly decays from 1.1 to 0.1; λ_{sdf} linearly increases from 0.0 to 1.0. The eikonal loss weight λ_{eikonal} is always 0.00001 and the MANO prior loss λ_{sdf} is always 5.0. To avoid over-regularization, the eikonal loss is only applied when it is greater than 8e-4. We use an L1 loss for $\mathcal{L}_{\text{rgb}}, \mathcal{L}_{\text{sdf}}, \mathcal{L}_{\text{sparse}}$, and L2 for $\mathcal{L}_{\text{segm}}, \mathcal{L}_{\text{eikonal}}$. We apply the sparsity loss $\mathcal{L}_{\text{sparse}}$ for the hand if a given ray's closest distance to the hand mesh is beyond 1cm, and for the object if it is beyond 5cm. At each iteration, we sample 256 points around the hand surfaces with Gaussian noise using a standard deviation of 0.008, as well as uniformly sample 51 points within a tight bounding box for hand or object; for the object, we use a standard deviation of 0.03. We choose these hyperparameters based on qualitative observations for all experiments. Since hyperparameter tuning is costly, we chose the them empirically and did not perform extensive tuning.

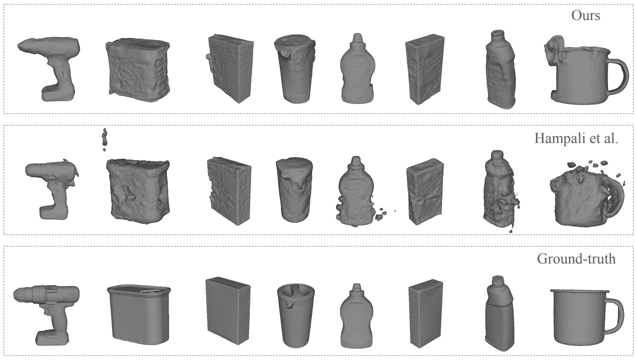

Figure A. **Comparison with Hampali *et al.* [11] for in-hand object scanning**. Our method (top) has significantly more details and fewer artifacts compared to Hampali *et al.* [11] (middle) and our results more closely resemble the groundtruth (bottom).

Loss terms	MPJPE $[mm]\downarrow$	CD$_h$ $[cm^2]\downarrow$	mIoU [%] ↑
BL	**22.1**	41.9	83.9
BL + $\mathcal{L}_{\text{mask}}$	**22.1**	249.0	83.2
BL + $\mathcal{L}_{\text{contact}}$	**22.1**	46.8	49.9
BL + $\mathcal{L}_{\text{mask}}$ + $\mathcal{L}_{\text{contact}}$	**22.1**	**9.1**	**86.3**
BL* + $\mathcal{L}_{\text{mask}}$ + $\mathcal{L}_{\text{contact}}$	**24.9**	14.4	87.7
BL* + $\mathcal{L}_{\text{mask}}$ + $\mathcal{L}_{\text{contact}}$ + $\mathcal{L}_{\text{bio}}$	25.1	**13.7**	**87.1**

Table C. **Ablation on pose refinement losses**. We ablate the effect of different refinement losses on a baseline (BL) before the refinement. * denotes allowing hand joint rotations to be optimizable.

D. Additional experiments

Ablation on fitting losses: We have experimented with different setups for optimizing hand and object poses. However, we found the current loss formulation is the most consistent in providing improvements. Table C shows the evaluation metrics after fitting a baseline model with different loss combinations. To measure pixel-alignment for hands and objects, we use mean interaction-over-union (mIoU) between the ground-truth segmentation masks and the one rendered from the optimized hand and object models. For computational efficiency, here we randomly choose one sequence for each object in HO3D to ablate this experiment. We see that, compared to mask-only, or contact-only fitting, fitting with both the mask loss $\mathcal{L}_{\text{mask}}$ and the contact loss $\mathcal{L}_{\text{mask}}$ is crucial for hand-relative object poses (see CD$_h$) while maintaining the pixel alignment of hand and object meshes (see mIoU). In our experiments, we have explored the possibility of optimizing hand joints during the fitting procedure, as indicated in experiments marked with an asterisk ($*$). However, this approach has not yielded consistently improved results in terms of MPJPE. Additionally, we investigated the integration of a biomechanical loss, denoted as $\mathcal{L}_{\text{bio}}$ from [23]. Contrary to expectations, this integration resulted in a deterioration of MPJPE performance. Consequently, to maintain simplicity and efficacy, we have

Noise Level	MPJPE [mm] ↓	CD [cm^2] ↓	F10 [%] ↑	CD$_h$ [cm^2] ↓
GT	**2.4**/20.5	0.5/**0.2**	97.2/**98.5**	**1.8**/16.1
GT + 1× noise	**18.4**/25.0	0.5/**0.4**	94.5/**96.4**	**6.0**/17.1
GT + 2× noise	38.9/**36.6**	18.7/**6.0**	36.1/**72.2**	62.4/**42.4**

Table D. **Analyze the effect of noise in poses**. Given different noise levels, we compare HOLD before or after pose refinement (left/right in the table). For all scenarios, object canonical geometries improve after pose refinement (see CD and F10).

decided to adhere to our current loss formulation.

Effect of noise in poses: Table D shows HOLD's performance on HO3D sequences (1 sequence per object) with various noise levels. The ground-truth (GT) poses (including 6D object poses, hand joint angles, and 6D hand rotation and translation) were altered with rotational (ϵ_{rot}) and translational (ϵ_{transl}) noise from Gaussian distributions $\mathcal{N}(0, 8°)$ and $\mathcal{N}(0, 16mm)$, termed "1× noise." Post-noise, we refined the poses (see Sec. 3.3 main paper) and trained HOLD-Net with the new poses and assessed HOLD-Net's effectiveness pre- and post-refinement (left/right in the table). Noise levels were also increased by 2×. Although in the main manuscript, we do not optimize hand poses (only hand translation), here we allow hand poses to be optimized for completeness. We see that when the noise in the poses is relatively low (see GT and "1× noise"), pose refinement introduces error in hand poses (MPJPE). This makes sense as our pose refinement only fit hands into silhouettes, which is insufficient for improving reasonably accurate poses. However, when the hand poses are extremely noisy ("2× noise"), the refinement improves the hand poses in MPJPE. This is also the case for the object's reconstruction in the hand coordinate (see CD$_h$) as this metric measures the spatial alignment between the hand and the object. Interestingly, across all noise levels, the canonical object geometries always improve after pose refinement (see CD and F10) even when ground-truth is used.

Results with random poses: For completeness and to facilitate future comparison, Table E shows the results of banana and scissors sequences from Table A using random object poses. For the hand poses, we use the same hand pose estimator in the main manuscript. For the object poses, we perturb ground-truth object rotations with Gaussian noise $\mathcal{N}(0, 32°)$. Intuitively, with a 95.5% chance, rotation noises fall within the confidence interval $[-64°, 64°]$. To be consistent with HOLD in the main manuscript, we freeze hand poses during energy-based fitting before the final training.

Segmentation masks: For each sequence, we use SAM-Track [6] with point-prompting to obtain hand and object segmentation masks. Obtaining such masks takes around 30 seconds for each sequence (requiring three to five clicks at the first frame). Table F shows that HOLD performs better with SAM-Track masks compared to STCN [5].

Sequence	MPJPE [mm] ↓	CD [cm^2] ↓	F10 [%] ↑	CD$_h$ [cm^2] ↓
BB12	18.9	2.1	82.3	39.8
BB13	19.4	2.1	82.4	77.5
GSF12	21.1	6.7	67.8	46.2
GSF13	19.1	6.9	65.4	63.2
Average†	19.6	4.4	74.5	56.6
Average‡	23.2	1.3	91.6	21.4

Table E. **Random pose performance for banana and scissors**. Results of HOLD using random poses: †Average across the banana and scissors sequences; ‡Average across sequences here and in the main manuscript.

	MPJPE [mm] ↓	CD [cm^2] ↓	F10 [%] ↑	CD$_h$ [cm^2] ↓
STCN	25.0	0.5	95.6	19.0
SAM-Track	**24.8**	**0.3**	**97.5**	**14.1**

Table F. **HOLD performance with different segm. masks**.

E. Discussion

In this paper, we present the first method that reconstructs category-agnostic articulated hands and objects from monocular videos. Being the first step, ours is not perfect.

First of all, during hand-object interaction, our hands could heavily occlude the object throughout the entire video sequence. Also, the object can have self-occlusion. Since our method relies on RGB supervision from video images, under-observed regions on the object could potential lead to artifacts due to the lack of proper regularization. As an example, Figure B (b) shows that artifact appears when the right side of the object handle is less observed in a video. However, there are huge advancement in text-to-3D diffusion priors [17] and they can be used to hallucinate under-observed object regions. Another aspect is to consider long-term interaction settings of the same object, which should provide a full observation of the object in interaction.

Our model relies on detector-based SfM to provide initial object pose estimates. This is challenging for objects with uniform or poor texture and for objects with thin structures. As an example, Figure B (c) shows that SfM struggles with the two sides of the handle for a kettle sequence due to the poor texture and symmetric shape of this object, resulting in an handle-like artifact. Orthogonal to our work, recently there has been a surge of detector-free SfM models [12, 22] that can potentially provide object poses in such cases. Further, object poses can be noisy due to motion blur and hand-object occlusion, which could also lead to artifacts. One can explore correcting the object poses by enforcing a consistency loss with 2D keypoints between consecutive frames.

Our method relies on the MANO skinning weights for deformation and uses K-nearest neighbours to find correspondences between the observation space and the canonical space. To further increase the realism of the learned hand model, one can try to learn the skinning weights from videos and leverage root-finding to find better correspon-

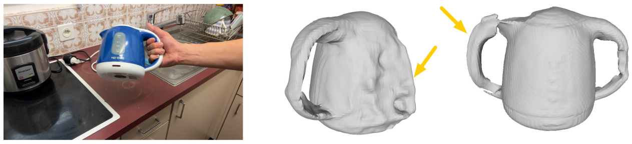

Figure B. **Failure cases**. In this sequence, the right side of the handle of the object has significantly fewer images for HOLD to learn its shape. The reconstruction has artifact in the under-observed side (see b). Also, SfM gets confused by the two sides of the kettle due to its simple texture and creates an artifacts on the object model (see c).

dences between the observation space and the canonical space [3, 4]. Moreover, when some hand parts are under-observed, the hand model tends to have MANO-like shape for those regions. A direction would be to learn a prior on realistic hand texture and shape from in-the-lab hand scans.

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European conference on computer vision*. Springer, 2016. 2

[2] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 1

[3] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *International Conference on Computer Vision (ICCV)*, 2021. 5

[4] Xu Chen, Tianjian Jiang, Jie Song, Max Rietmann, Andreas Geiger, Michael J. Black, and Otmar Hilliges. Fast-SNARF: A fast deformer for articulated neural fields. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. 5

[5] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 4

[6] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 1, 4

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Computer Vision and Pattern Recognition Workshops (CVPRw)*, 2018. 1

[8] Stuart Geman and Donald E. McClure. Bayesian image analysis: An application to single photon emission tomography. *Proceedings of the American Statistical Association*, 1985. 2

[9] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2Avatar: 3D avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[10] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[11] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. In-hand 3D object scanning from an RGB sequence. *Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[12] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *arxiv*, 2023. 4

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 3

[14] Ke Li and Jitendra Malik. Amodal instance segmentation. In *European Conference on Computer Vision (ECCV)*. Springer, 2016. 2

[15] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[16] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[17] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *International Conference on Learning Representations (ICLR)*, 2022. 4

[18] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[19] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[20] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[21] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[22] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[23] 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 *International Conference on Computer Vision (ICCV)*, 2021. 3

[24] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 2

[25] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. MediaPipe hands: On-device real-time hand tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRw)*, 2020. 1

[26] 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)*. Springer, 2020. 2

[27] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2