

gSDF: Geometry-Driven Signed Distance Functions for 3D Hand-Object Reconstruction Supplemental Material

Zerui Chen Shizhe Chen Cordelia Schmid Ivan Laptev
Inria, École normale supérieure, CNRS, PSL Research Univ., 75005 Paris, France
firstname.lastname@inria.fr

In the supplementary material, we provide more details of our method and additional results. We first present details of our model architecture in Section A. Then in Section B, we provide more details about solving hand poses from predicted 3D joints using inverse kinematics. Finally, we discuss additional experimental results in Section C.

A. Network Architecture

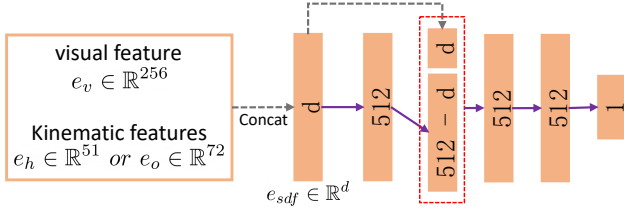


Figure 1. Network architecture used for our hand and object SDF decoders. Following [1, 3], we use five fully-connected layers (marked in purple) for the SDF decoder. The number in the box denotes the dimension of features.

For our SDF decoders (see Figure 2 in the original paper) we adopt the model architecture used in [1, 3] which employ five fully-connected layers as the decoder as illustrated in Figure 1. Given visual feature $e_v \in \mathbb{R}^{256}$ from the input image (Section 3.2) and kinematic features $e_h \in \mathbb{R}^{51}$ or $e_o \in \mathbb{R}^{72}$ from the query point (Section 3.3), we concatenate them together to build a d -dimensional vector e_{sdf} and feed it into the SDF decoder.

B. Hand Kinematics

In this section, we first introduce the forward kinematics and inverse kinematics for the hand as shown in Figure 2(a). Then we present how to use inverse kinematics to calculate hand poses from predicted 3D joints in our method.

Forward Kinematics. Forward kinematics is usually defined as the process to compute posed hand joints $\psi_p \in \mathbb{R}^{21 \times 3}$ from given hand poses (*i.e.*, relative rotations $\theta \in \mathbb{R}^{16 \times 3}$ and

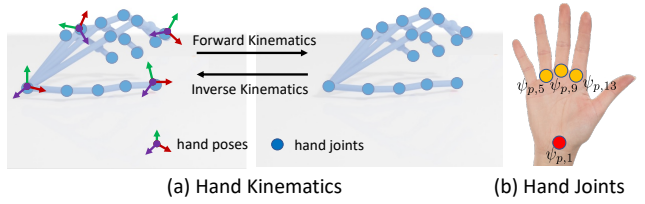


Figure 2. Illustration of hand kinematics. In Figure (a), we show functions of forward kinematics and inverse kinematics. In Figure (b), we show relevant joints (marked in yellow) that are involved in the computation of the hand wrist rotation.

relative translations $\phi \in \mathbb{R}^{16 \times 3}$) and template hand joints $\psi_t \in \mathbb{R}^{21 \times 3}$. The k_{th} joint in ψ_p can be computed as:

$$\begin{aligned} \psi_{p,k} &= R_k \cdot \phi_k + \psi_{p,pa(k)}, \\ R_k &= R_{pa(k)} \cdot \exp(\theta_k), \end{aligned} \quad (1)$$

where R_k denotes the global rotation matrix for the k_{th} joint and $pa(\cdot)$ returns the parent index of the k_{th} joint. $\exp(\cdot)$ denotes *Rodrigues formula* to convert θ_k into the form of the rotation matrix. We follow the inverse order of the kinematic chain to derive the global rotation for the k_{th} joint. For simplicity, we assume that all hands share the same template and set the relative translation as $\phi_k = \psi_{t,k} - \psi_{t,pa(k)}$, which simplifies the computation of Equation 1 to:

$$\begin{aligned} \psi_{p,k} &= R_k \cdot (\psi_{t,k} - \psi_{t,pa(k)}) + \psi_{p,pa(k)}, \\ R_k &= R_{pa(k)} \cdot \exp(\theta_k). \end{aligned} \quad (2)$$

Inverse Kinematics. Given posed hand joints ψ_p and template hand joints ψ_t , inverse kinematics solves relative hand poses (θ, ϕ) that defines the transformations from ψ_t to ψ_p . As we do in forward kinematics, we also omit ϕ in the computation of inverse kinematics and only solves relative hand rotations θ . We first derive the hand wrist rotation matrix $R_1 \in \mathbb{R}^{3 \times 3}$ from the orientation of three connected joints as shown in Figure 2(b) and formulate it as an optimization

Table 1. Comparison with state-of-the-art methods on ObMan.

Method	C _r	P _d	I _v
Hassoret al. [2]	94.8%	1.20	6.25
Karunratanakul et al. [3]	69.6%	0.23	0.20
Chenet al. [1]	95.5%	0.66	2.81
gSDF (Ours)	89.8%	0.42	1.17

Table 2. Comparison with state-of-the-art methods on DexYCB.

Method	C _r	P _d	I _v
Hassoret al. [2]	95.7%	1.15	9.64
Karunratanakul et al. [3]	96.0%	0.92	6.62
Chenet al. [1]	96.6%	1.08	8.40
gSDF (Ours)	95.4%	0.94	6.55

Table 3. Object reconstruction performance with different object kinematic features on DexYCB dataset denotes our re-implementation of the method proposed in Ye et al. [5].

Model	Obj. Pose	CD _o #	FS _o @5"	FS _o @10"
R1 Yeet al. [5]	-	-	0.420	0.630
R2 Yeet al.	-	2.09	0.404	0.663
R3 gSDF	-	1.78	0.411	0.676
R4 (Ours)	X	1.71	0.418	0.689

problem:

$$R_1 = \arg \min_{R \in SO^3} \sum_{i \in \{2f, 5, 9, 13\}} \| \mathbf{p}_{i,j} - \mathbf{R} \mathbf{t}_{i,j} \|^2 \quad (3)$$

where we can apply Singular Value Decomposition (SVD) as in [4] to solve this problem. Then, we follow the hand kinematic chain and solve the 3D rotation recursively for each joint. To this end, we rewrite Equation 2 defined in forward kinematics:

$$\mathbf{R}_{pa(k)}^{-1}(\mathbf{p}_{i,k} - \mathbf{p}_{pa(k)}) = \exp(\mathbf{k}) (\mathbf{t}_{i,k} - \mathbf{t}_{pa(k)}): \quad (4)$$

Then, we could derive the norm and orientation ϕ by computing the dot product and cross product between the vector $\mathbf{R}_{pa(k)}^{-1}(\mathbf{p}_{i,k} - \mathbf{p}_{pa(k)})$ and the vector $\mathbf{t}_{i,k} - \mathbf{t}_{pa(k)}$, respectively.

C. Experimental Results

C.1. Evaluations using additional metrics

To provide a more comprehensive view about our 3D hand poses and object poses to produce object kinematic reconstruction performance, we also report Contact Ratio (C_r), Penetration Depth (P_d) (cm) and Intersection Volume

Table 4. Comparing computational requirements of different models when reconstructing hand and object meshes of resolution 128 × 128 from an image on an NVIDIA 1080Ti GPU.

Method	Input	GPU Memory	Latency
[3]	Image	2357Mb	2.87s
[1]	Image	2847Mb	3.17s
Ours	Image	3425Mb	3.23s
Ours	Video	3764Mb	4.14s

Table 5. Comparison of our method with AlignSDF [1] on DexYCB while using different numbers of backbones (BB).

Model	CD _h #	FS _h @1"	FS _h @5"	CD _o #	FS _o @5"	FS _o @10"
[1]-1BB	0.358	0.162	0.767	1.83	0.410	0.679
Ours-1BB	0.329	0.166	0.787	1.88	0.420	0.689
[1]-2BB	0.344	0.167	0.776	1.81	0.413	0.687
Ours-2BB	0.310	0.172	0.795	1.71	0.426	0.694
Ours-3BB	0.326	0.168	0.784	1.82	0.414	0.679

(I_v) (cm³) for our models. We follow the same process as previous works [1, 3] to compute these metrics. As shown in Table 1 and Table 2, we can observe that our approach can generate results with relatively low Penetration Depth (P_d) and Intersection Volume (I_v) on both the ObMan and DexYCB benchmarks, which suggests that our model can produce physically plausible 3D reconstruction of hand and object meshes. Table 4 compares the speed and memory of different models. Our image model only slightly increases compute compared to [1, 3].

C.2. Comparison with Yeet al. [5]

As Ye et al. [5] is a close work related to ours, we provide more ablation results for comparison with Ye et al. [5] in Table 3. The main differences between Ye et al. [5] and our work are three-fold. Firstly, they focus on 3D hand-held object reconstruction instead of joint hand-object reconstruction. Secondly, they only consider the hand poses for object reconstruction without object poses, and the hand poses are predicted from an off-the-shelf model. Finally, a larger SDF decoder is used in their work while we follow [1, 3] and use a smaller decoder architecture. Therefore, in Table 3, we only compare the object reconstruction performance. We also re-implement Ye et al. [5] (R2 in Table 3) using the same SDF decoder and the same predicted hand poses as ours for a fair comparison. The model in R3 indicates that the joint optimization of hand-object reconstruction is beneficial compared to the model in R2. Our model in R4 uses both hand poses and object poses to produce object kinematic features and achieves the best performance on all the metrics for 3D object reconstruction.

!"#\$%&

/01\$2'()

\$() *+,-&.

Figure 3. Qualitative comparison between AlignSDF [1] and our gSDF. Our approach can produce more realistic hand and object reconstruction results.

C.3. Ablations on the number of backbones

Table 5 reports additional results showing improvements of our method over [1] while using the same number of backbones. We note that all models in this table are trained with the local visual features V_2 defined in Table 3. We observe that gSDF can still outperform AlignSDF [1] under a single backbone setting. For a better comparison, we also

extend AlignSDF to two backbones and train it with the two-stage strategy. 2BB results in Table 5 show that our method outperforms [1] even when both methods use two backbones. We further conduct an experiment with three backbones, where we use three separate backbones for hand and object pose estimation and SDF learning. We observe that 3BB consumes more resources without improving performance.

Figure 5. Failure cases analysis of our method on the DexYCB benchmark.

can still faithfully recover their 3D surfaces.

C.5. Failure cases analysis

In this section, we analyze some typical patterns for our method on the DexYCB benchmark. As shown in Figure 5, our method sometimes makes unreliable predictions in cluttered scenes. Our method uses a hand-relative coordinate system. Hence, the reconstruction of both hands and objects may fail for scenes with heavily occluded hands. Since our method takes monocular RGB frames as the input, reconstructed objects, especially for big objects, might have incorrect scales. For some objects with complex geometric topology, it is still difficult to produce accurate 3D reconstructions under strong motion blur.

References

- [1] Zerui Chen, Yana Hasson, Cordelia Schmid, and Ivan Laptev. AlignSDF: Pose-Aligned signed distance fields for hand-object reconstruction. In *ECCV*, 2022. 1, 2, 3, 4
- [2] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevtykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 2
- [3] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping Field: Learning implicit representations for human grasps. In *3DV*, 2020. 1, 2
- [4] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation. In *CVPR*, 2021. 2
- [5] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What’s in your hands? 3D reconstruction of generic objects in hands. In *CVPR*, 2022. 2

Figure 4. Qualitative results of our model on test images from the ObMan and DexYCB benchmarks. Our approach can produce convincing 3D reconstruction results for different hand grasping poses and challenging objects.

This shows that object pose estimation and SDF learning benefit from a shared backbone in our 2BB asymmetric architecture.

C.4. Qualitative results

In this section, we include more qualitative examples in Figure 4 to show that our approach can reconstruct high-quality hand meshes and object meshes for some challenging cases. We also qualitatively compare our method with a most recent work AlignSDF [1] on both the ObMan and DexYCB benchmarks. As shown in Figure 3, we can observe that our method produces more realistic reconstruction results. Even for some objects with thin structures (e.g., bowl), our method