UV-Based 3D Hand-Object Reconstruction with Grasp Optimization

Ziwei Yu[1]
yuziwei@u.nus.edu
Linlin Yang[1, 2]
yangll@comp.nus.edu.sg
You Xie[3]
you.xie@tum.de
Ping Chen[4]
redcping@gmail.com
Angela Yao[1]
ayao@comp.nus.edu.sg

[1] National University of Singapore, Singapore
[2] University of Bonn, Germany
[3] Technical University of Munich, Germany
[4] Shanghai Subsidiary of Great Wall Motor Co., Ltd., China

Abstract

We propose a novel framework for 3D hand shape reconstruction and hand-object grasp optimization from a single RGB image. The representation of hand-object contact regions is critical for accurate reconstructions. Instead of approximating the contact regions with sparse points, as in previous works, we propose a dense representation in the form of a UV coordinate map. Furthermore, we introduce inference-time optimization to fine-tune the grasp and improve interactions between the hand and the object. Our pipeline increases hand shape reconstruction accuracy and produces a vibrant hand texture. Experiments on datasets such as Ho3D, FreiHAND, and DexYCB reveal that our proposed method outperforms the state-of-the-art.

1 Introduction

3D reconstruction of hand-object interactions is critical for object grasping in augmented and virtual reality (AR/VR) applications [22, 41, 42] We consider the problem of estimating a 3D hand shape when the hand interacts with a known object with a given 6D pose. This set-up lends itself well to AR/VR settings where the hand interacts with a predefined object, perhaps with markers to facilitate the object pose estimation. Such a setting is common, although the majority of previous works [23, 25, 26] consider 3D point clouds as input, while we handle the more difficult case of monocular RGB inputs. Additionally, the previous works [9, 10, 23, 25, 26, 39] are singularly focused on reconstructing feasible hand-object interactions. They aim to produce hand meshes with minimal penetration to the 3D object without regard for the accuracy of the 3D hand pose. We take on the additional challenge of balancing realistic hand-object interactions with accurate 3D hand poses.

Representation-wise, previous hand-object 3D reconstruction works [2, 27, 40, 46] predominantly with the MANO model [37]. MANO is convenient to use, but its accuracy is

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

limited because it cannot represent direct correspondences between the RGB input and the hand surface. This work considers dense representations and, in particular, focuses on UV coordinate maps. UV maps are ideal as they establish dense correspondences between 3D surfaces and 2D images and work well in representing the 3D human body and face [14, 24].

Working with a UV coordinate map has a natural advantage in that we can adopt standard image-based CNN architectures. The CNN captures the pixels' spatial relationships to aid the 3D modelling while remaining fully convolutional. UV coordinate maps can be easily augmented with additional corresponding information, such as surface texture and regions of object contact. This creates a seamless connection between the 3D hand shape, its appearance, and its interactions with objects in a single representation space (see Fig. 1a). To that end, we propose an RGB2UV network that simultaneously estimates the hand UV coordinates, hand texture map, and hand contact mask.

The hand contact mask is a key novelty of our work. We propose a binary UV contact mask (see Fig. 1b-f) that marks contact regions between the hand surface and the interacting object surfaces. To our knowledge, the contact mask is the first dense representation of hand-object contact. Previous methods [19, 43] work sparsely on a per-vertex or per-point basis and have an obvious efficiency-accuracy trade-off. The more vertices or points considered, the more accurate the contact modelling and the higher the computational price. Working in the UV space allows dense contact modelling and improves reconstruction accuracy while remaining efficient.

Normally, accurate hand surface reconstructions in isolation cannot guarantee realistic hand-object interactions; ensuring realistic interactions in vice-versa may result in inaccurate hand reconstructions [19, 20, 31]. The main reason is that the hand and object models (MANO [37], YCB [4]) are all rigid. The rigid assumption is too strong at the contact points, and leads to one mesh penetrating the other [23, 43], even for ground truth poses. To mitigate these errors, we propose an additional optimization-based refinement procedure to improve the overall grasp. The optimization is performed only during inference and reduces the distances from a hand surface that either penetrates or does not make contact with the object surface. This grasp optimization step significantly improves the feasibility of the hand-object interaction while ensuring accurate hand poses.

Our contributions are summarized as follows: (1) We propose a UV-map-based pipeline for 3D hand reconstruction that simultaneously estimates UV coordinates, hand texture, and contact maps in UV space. (2) Our model is the first to explore dense representations to capture contact regions for hand-object interaction. (3) Our grasp optimization refinement procedure yields more realistic and accurate 3D reconstructions of hand-object interactions. (4) Our model achieves state-of-the-art performance on FreiHAND, Ho3D and DexYCB hand-object reconstruction benchmarks.

2 Related Works

Face, Body and Hand Surface Reconstruction. A popular way to represent human 3D surfaces is via a 3D mesh. Previous works estimate mesh vertices either directly [8, 15, 29, 33, 35, 41], or indirectly [2, 28, 46] through a parametric model like SMPL [32] for the body, 3DMM [1] for the face, and MANO [37] for the hand. Parametric models are convenient but cannot provide direct correspondences between the input images and the 3D surface. As such, we learn a dense surface as a UV coordinate map. We are inspired by the success of recent UV works for the human body [16, 45], face [13, 24, 30], and hand [6, 41]. To the

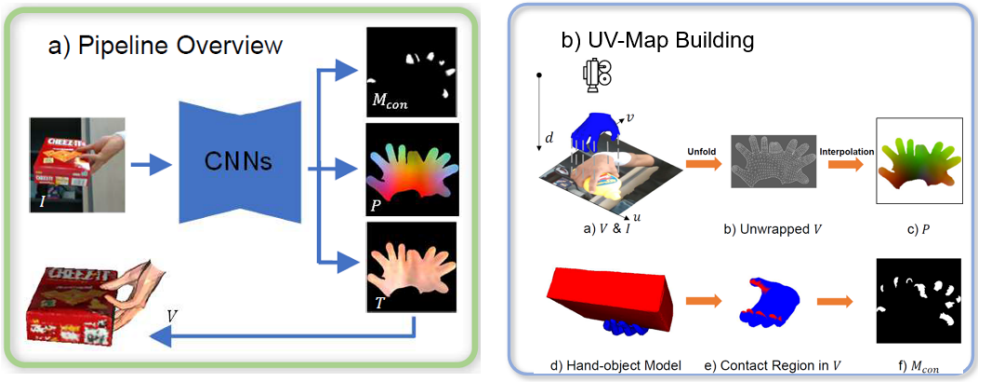

Figure 1: a) We use UV coordinate map P as a dense representation of hand mesh V, and train networks to directly generate P, contact mask M_{con} and texture map T from RGB image I; we then achieve a 3D reconstruction of the hand-object interaction or just the hand. Our method. b) Generation of the ground truth UV maps (a-c) and contact mask (d-f).

best of our knowledge, our work is the first to explore the use of UV coordinate maps in modelling hand-object interactions and capturing the contact regions.

Texture Learning on Hand Surfaces. Shading and texture cues are often leveraged to solve the self-occlusion problem for hand tracking [11, 12]. Recently, [7] verified that texture modelling helps with the self-supervised learning of shape models. Instead of relying on a parametric texture model [34], we apply a UV texture map. UV texture maps are commonly used to model the surface texture of the face [13, 24, 30] and, more recently, the human body [45]. We naturally extend UV texture maps for hand-object interactions.

Hand-Object Interactions. Various contact losses have been proposed [19, 21, 43] to ensure feasible hand-object interactions. Examples such as repulsion loss and attraction loss ensure that the hand makes contact with the object but avoids actual surface penetration. Contact regions, however, are modeled sparsely. Contact points are sparse and selected either statistically [20], or based on hand vertices closest to the object [31]. These methods have an accuracy-efficiency trade-off hinging on the number of considered vertices. In contrast, our UV contact mask establishes a dense correspondence between the contact regions and the local image features. It is both accurate and efficient. Our work is the first to explore a joint position, texture and contact-region UV representation for hand-object interactions.

3 Method

Our method has two components: the RGB2UV network and grasp optimization (see Fig. 2). The RGB2UV network (Sec. 3.1 and 3.2) takes as input an RGB image of a hand-object interaction and outputs a hand UV mask and contact mask. The grasp optimization (Sec. 3.3) adjusts the mesh vertices to refine the hand grasp and reduces penetrations.

3.1 Ground Truth UV, Contact Mask & Texture

UV Coordinate Map. Suppose we are given a 3D hand surface in the form of a 3D mesh V and an accompanying RGB image I (see Fig. 1 a). The hand mesh $V(\mathbf{x}_V) \in \mathbb{R}^{778\times 3}$ stores

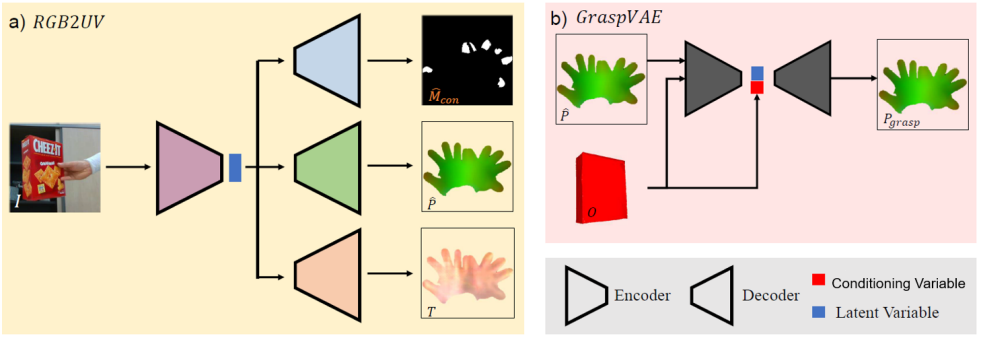

Figure 2: Overview of a) RGB2UV and b) GraspVAE. RGB2UV takes as input an image and outputs the hand contact mask, UV coordinate map, and texture map. GraspVAE further refines the predicted UV coordinate map by conditioning on the interacting object.

the 3D position $\mathbf{x}_V = (x,y,z) \in \mathbb{R}^3$ of the 778 vertices in the camera coordinate space. The image $I(\mathbf{x}_I) \in \mathbb{R}^{H_I \times W_I \times 3}$ records the RGB values of each pixel $\mathbf{x}_I = (u,v) \in \mathbb{R}^{H_I \times W_I}$ in the image plane. I can be considered a projection of V into 2D space, with the mapping

$$(u,v,d) = \mathcal{T}((x,y,z),c), \tag{1}$$

where $\mathcal{T}(\mathbf{x}_V,c)$ is a transformation that can project a vertex at (x,y,z) into (u,v,d) in the image plane with camera parameters c. Here, d denotes the distance from the camera to the image plane. Because I is on the image plane, it only reveals the vertices closest to the camera; we thus consider the vertical distance d from the camera to the vertex (see Fig. 1 b-a), and represent each vertex uniquely with (u,v,d).

To represent all the hand vertices, even those not seen in I, in one 2D plane, we unwrap the MANO mesh with MANO's provided UV unfolding template (see Fig. 1 b-b) [37]) to arrive at a UV coordinate map P with UV coordinates $\mathbf{x}_P \in \mathbb{R}^{H_P \times W_P \times 3}$, where

$$P(\mathbf{x}_P) = (u,v,d). \tag{2}$$

Mapping only the vertices V onto the UV map results in a sparse coordinate map. To generate a dense UV representation P (Fig. 1 b-c), we interpolate values on each triangular mesh face from neighbouring vertices. Given (u,v,d), the corresponding 3D coordinate is given by the $\mathcal{T}^{-1}$ is the inverse transformation of $\mathcal{T}$ from Eq. 2:

$$(x,y,z) = \mathcal{T}^{-1}((u,v,d),c). \tag{3}$$

Contact Mask. Based on P, we propose a binary contact mask $M_{\text{con}} \in \mathbb{R}^{H_P \times W_P}$ to indicate the contact region in hand-object interactions. Like [17], contact vertices are defined as the vertices within 4 mm to the grasped object. We first locate contact vertices $\mathbf{x}_V$ (Fig. 1 b-e) and their corresponding points $\mathbf{x}_P$ in P with Eq. 1 and Eq. 2. The contact mask is defined as:

$$M_{\text{con}}(\mathbf{x}_P) = \begin{cases} 0, & \text{if } dist(\mathbf{v},V_O) > 4\ mm; \\ 1, & \text{if } dist(\mathbf{v},V_O) <= 4\ mm. \end{cases} \tag{4}$$

Similar to P, direct mapping from V to M_{con} generates a sparse representation. For every triangular plane in V, we interpolate all points inside the plane as a contact region if the three vertices are contact vertices to achieve a dense representation of M_{con} (see Fig. 1 b-f).

Texture. The UV coordinates P also conveniently establish a correspondence between the mesh vertices and texture information. The texture map $T \in \mathbb{R}^{H_P \times W_P \times 3}$ gives information on hand appearance. For every point $\mathbf{x}_P$ in T, similar to Eq. 2, we obtain the texture map: $T(\mathbf{x}_P) = I(u,v)$. With Eq. 3, the texture of the vertices in the hand mesh can be tracked in T.

3.2 UV Estimation from an RGB Image

The RGB2UV network, $(\hat{P}, \hat{M}_{\text{con}}, T) = RGB2UV(I)$, estimates the UV coordinate map $\hat{P}$, contact mask $\hat{M}_{\text{con}}$, and texture map T from an input image I. With prepared data tuples (I, P, M_{con}), as described in Sec. 3.1, the network can be trained in a supervised way. Similar to previous works [6, 44, 45], we use a U-Net style encoder-decoder with a symmetric ResNet50 and skip connections between the corresponding encoder and decoder layers.

Our network features three decoders: a hand UV coordinate map decoder, a contact mask decoder, and a UV texture decoder (see Fig 2 a). The encoder and three decoders are trained jointly with five losses:

$$L_{\text{RGB2UV}} = L_P + L_{\text{grad}} + \lambda_1 L_{M_{\text{con}}} + L_V + \lambda_2 L_{\text{texture}} \tag{5}$$

UV Coordinate Map Decoder. This decoder is learned with the hand UV coordinate map loss L_P, hand UV gradient loss L_{grad}. These losses are L1 terms between the ground truth and the predicted UV coordinate maps P and $\hat{P}$, and their respective gradients ∇P and $\nabla \hat{P}$:

$$L_{\text{P}} = |P - \hat{P}| \cdot M \qquad \text{and} \qquad L_{\text{grad}} = |\nabla P - \nabla \hat{P}| \cdot M, \tag{6}$$

where M is a mask in the UV space indicating the entire hand region. M is applied to constrain the consideration of the UV coordinate map and their gradients to only valid regions on the hand's surface in $\hat{P}$ and $\nabla \hat{P}$.

Contact Mask Decoder. This decoder is learned with the L1 contact mask loss $L_{M_{\text{con}}}$ between the ground truth and predicted contact mask M_{con} and $\hat{M}_{\text{con}}$ respectively: $L_{M_{\text{con}}} = |M_{\text{con}} - \hat{M}_{\text{con}}|$. To ensure correctness in the hand pose represented by the coordinate map and contact mask decoders, we add an additional mesh vertex loss L_V. The hand mesh V implicitly represents the pose of the hand, so a loss between ground truth V and estimated $\hat{V}$ improves the accuracy of the estimated pose. The pose loss L_V is defined as: $L_V = |V - \hat{V}|$, where V represents ground truth hand mesh vertices and $\hat{V}$ is sampled from $\hat{P}$.

Texture Decoder. As there is no ground truth for UV texture, we apply self-supervised training to estimate the UV texture map. With given camera parameters c, the ground truth mesh vertices V can be projected into the uv plane to obtain a view-specific hand silhouette S_{gt} via a differentiable renderer[1]. The hand region in RGB image I can be isolated with $I \cdot S_{\text{gt}}$ by applying S_{gt} as a mask. On the other hand, we also render the hand image I_{re} with the estimated UV coordinate and texture map, *i.e.* $\hat{P}$ and T. The texture decoder can then be trained by minimizing the distance between I_{re} and $I \cdot S_{\text{gt}}$. We apply a photometric consistency loss L_{texture} [7] for training, which consists of an RGB pixel loss term L_{pixel} and a structure similarity loss L_{SSIM} based on the structural similarity index [3] SSIM:

$$L_{\text{texture}} = L_{\text{pixel}} + \lambda_3 L_{\text{SSIM}}, \quad L_{\text{pixel}} = |I \cdot S_{\text{gt}} - I_{\text{re}}|, \text{and} \quad L_{\text{SSIM}} = 1 - \text{SSIM}(I \cdot S_{\text{gt}}, I_{\text{re}}). \tag{7}$$

[1] We use the built-in renderer from the `pytorch3D` library [36].

3.3 Grasp Optimization

The RGB2UV network is standalone and does not take any object information into consideration. Even though the ground truth 6D object pose is provided in our setup, this does not ensure feasible hand grasps or naturalistic hand-object interactions. Therefore, we introduce a grasp optimization step for further refinement. Instead of refining the 3D hand mesh directly, which is very high-dimensional with 778 vertices, we work in a latent variable space to reduce the dimensionality of the optimization. Specifically, we learn a conditional VAE with an encoder $En(\cdot)$ and a decoder $De(\cdot)$, which we name GraspVAE.

During inference, the estimated UV coordinate map $\hat{P}$ is encoded into a latent variable $\mathbf{z}$ with GraspVAE's $En(\cdot)$. The latent $\mathbf{z}$ is refined to $\mathbf{z}^*$ through optimization to minimize the penetrations between the object mesh and the estimated hand mesh. Finally, $\mathbf{z}^*$ is decoded with $De(\cdot)$ into a UV coordinate map and converted into a hand mesh. Note the optimization happens during inference only, whereas $En(\cdot)$ and $De(\cdot)$ are learned during training.

GraspVAE is a conditional VAE with a ResNet18 encoder and three transposed convolution layers as the decoder. It is conditioned on the object vertices O (see Fig. 2 b) to estimate a refined UV coordinate map P_{grasp} from the estimated $\hat{P}$ from the RGB2UV network, *i.e.* $P_{\text{grasp}} = \text{GraspVAE}(\hat{P}|O)$. GraspVAE is trained with the combined loss

$$L_{\text{Grasp}} = L_P + L_{\text{grad}} + L_V + \lambda_4 L_{\text{KL}} + L_{\text{pene}}, \quad \text{where} \quad L_{\text{pene}} = \frac{1}{|V_{in}^o|} \sum_{\mathbf{x}_V \in V_{in}^o} \text{dist}(\mathbf{v}, V_O). \tag{8}$$

Here, $L_{\text{KL}} = KL(q(\mathbf{z}|(\hat{P},O))||p)$ is the standard Kullback-Leibler divergence loss used in VAE models, where $\mathbf{z}$ represents the latent variables encoded from input $(\hat{P},O)$. The term $p = \mathcal{N}(0,E)$ denotes a Gaussian prior where E is an identity matrix. The term $q(\mathbf{z}|(\hat{P},O))$ is the distribution of $\mathbf{z}$, while the $\text{dist}(\cdot,\cdot)$ function estimates the closest distance between a hand vertex $\mathbf{x}_V$ and object vertices O. The penetration loss L_{pene} is applied to penalize hand vertices $\mathbf{x}_V \in V_{in}^O$, where V_{in}^O represents the set of hand vertices inside the object. Existing works [19] identify V_{in}^O from the entire hand mesh, which is computationally inefficient. Our contact mask restricts the search area and decreases the processing time. For more details on training a conditional VAE, we refer the reader to [38].

Latent-Space Optimization. During inference, the learned GraspVAE model is used to estimate a refined UV map. Specifically, we encode the given object vertices O and estimated $\hat{P}$, *i.e.* $\mathbf{z}_o = \text{En}(\hat{P},O)$ and solve for an optimized $\mathbf{z}^*$, with $\mathbf{z}_o$ as initialization:

$$\mathbf{z}^* = arg\min_{\mathbf{z}} L_{\text{KL}} + L_{\text{pene}}. \tag{9}$$

The hand UV map is generated with $P^* = De(\mathbf{z}^* \bigoplus O)$, where $\bigoplus$ denotes a concatenation operation. The refined hand mesh V^* can then be generated from P^* with Eqs. 2 and 3. More details are given in the Supplementary.

Hand Only VAE. To further highlight the strengths of our grasp optimization, we introduce a variant of GraspVAE named Hand Only VAE to refine the hand shape accuracy with. Similar to GraspVAE, the encoder of Hand Only VAE is a ResNet18 network and the decoder contains three transposed convolution layers. Hand Only VAE is trained with L_{Grasp} in Eq. 8, but without L_{pene}. Thus, the role of L_{pene} in grasp optimization can be visualized from differences between results of Hand Only VAE and GraspVAE.

Dataset	FreiHand		Ho3D V2					Ho3D V3					DexYCB				
Method	MPJPE	MPVPE	MPJPE	MPVPE	PD	SIV	SD	MPJPE	MPVPE	PD	SIV	SD	MPJPE	MPVPE	PD	SIV	SD
Hampali[10]	-	-	1.07	1.06	12.34	17.28	4.02	-	-	-	-	-	-	-	-	-	-
Hasson[19]	1.33	1.33	1.10	1.12	-	-	-	-	-	-	-	-	-	-	-	-	-
Hasson[20]	1.33	1.33	1.14	1.09	18.44	14.35	4.10	-	-	-	-	-	-	-	-	-	-
GraspField [45]	-	-	1.38	1.36	14.61	14.92	**3.29**	-	-	-	-	-	-	-	-	-	-
Li [31]	-	-	1.13	1.10	11.32	14.67	3.94	1.08	1.09	16.78	10.87	3.90	1.28	1.33	9.47	11.02	3.64
Chen [6]	0.72	0.74	**0.99**	**1.01**	10.25	15.38	4.05	1.25	1.24	17.87	12.56	4.02	1.23	1.13	8.64	12.38	3.36
Dataset GT	-	-	-	-	-	-	-	-	-	-	-	-	0	0	4.58	7.16	1.48
MANO CNN	1.10	1.09	1.30	1.30	17.42	14.2	4.33	1.38	1.37	18.65	14.78	4.36	1.39	1.40	16.24	15.38	4.08
MANO Fit	1.37	1.37	1.58	1.61	21.37	18.0	4.82	1.66	1.65	22.04	15.60	4.65	1.49	1.43	18.56	17.60	4.86
RGB2UV	0.75	0.78	1.08	1.07	11.33	17.24	4.24	1.22	1.23	16.72	13.50	4.27	1.20	1.19	7.03	11.02	3.28
Hand Only	**0.71**	**0.73**	1.04	1.04	9.68	14.1	3.92	**1.08**	**1.04**	13.07	11.77	3.88	**1.09**	**1.02**	6.44	9.32	2.98
Hand+Object	-	-	1.25	1.33	**7.66**	**10.4**	**3.22**	1.28	1.26	**9.67**	**8.66**	**3.01**	1.29	1.27	**5.05**	**7.11**	**2.64**

Table 1: Comparison with state-of-the-art methods. **Best** and Second-best Scores. Our Hand Only achieves the best holistic performance across all comparisons. Note that the PD and SIV of DexYCB ground truth data are non-zero due to the rigid modeling of both the hand and the object. Combining the two inevitably results in some penetration, and this serves as a lower bound in achievable results.

4 Experiments

Implementation Details. The Adam optimizer is applied to train all networks over 80 epochs with a batch size of 64. We start with an initial learning rate of 10^{-4} for all training and lower it by a factor of 10 at the 20th, 40th and 60th epochs. After GraspVAE is trained, we use the learning rate of 10^{-6} to update the latent variables until the loss difference is lower than 10^{-6} for grasp optimization. The parameters are set empirically all to 1, except for $\lambda_{1,2,3} = 10$, $\lambda_4 = 0.001$. The VAE latent space is 128 dimensions.

Datasets. We evaluate on RGB-based hand-object benchmarks HO3D [17, 18], DexYCB [6], and FreiHAND [46]. We compare HO3D with state-of-the-art for both v2 and v3 and report our results through their leaderboard (v2[2], v3[3]). For DexYCB dataset, we use the official "S0" split. Since FreiHAND does not provide object models and object annotations, we only evaluate the hand shape reconstruction on our RGB2UV network and our Hand Only pipeline. Results are reported through their leaderboard[4].

Metrics. For evaluating the hand pose and shape accuracy, we use the mean-per-joint-position-error (MPJPE) (cm) for 3D joints and mean-per-vertex-position-error (MPVPE) (cm) for mesh vertices. For evaluating the hand-object interaction, we measure the penetration depth (PD) (mm), solid intersection volume (SIV) (cm^3) [43] and Simulation Displacement (SD) (cm) [19]. PD is based on the maximum distance of all hand vertices inside the object with respect to their closet object vertices. SIV is defined as the total voxel volume of hand vertices inside the object after converting the object model into 80^3 voxels. SD measures grasp stability in a simulation space that the hand is fixed and the grasped object is subjected to gravity.

4.1 Comparison with the State-of-the-Art

Quantitative Results. Comparison with state-of-the-art results of [6, 17, 19, 20, 31] in Table 1 are based on their released source code and default parameters. Considering only the hand pose and shape accuracy, our Hand Only pipeline obtains the lowest MPJPE and

[2]https://competitions.codalab.org/competitions/22485#results
[3]https://competitions.codalab.org/competitions/33267#results
[4]https://competitions.codalab.org/competitions/21238#results

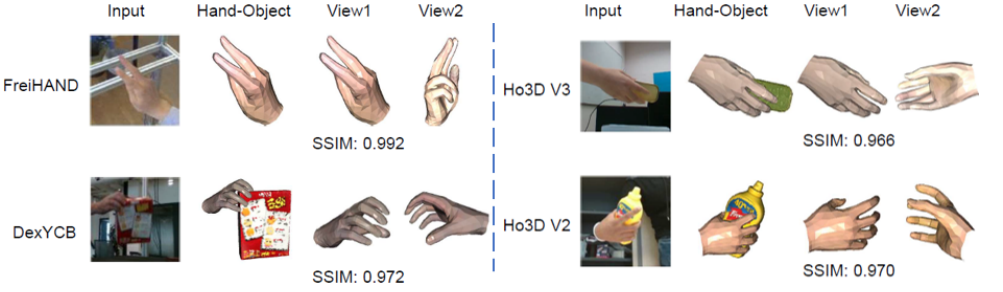

Figure 3: Surface Results: predicted 3D hand surfaces with hand shape. For each quartet, from left to right the columns correspond to RGB input, hand surface with ground truth object, hand surface in camera view, and hand surface in a different view. Besides, similarities between projected hand I_{re} and its ground truth are measured with the SSIM matrix. High SSIM values indicate that our texture decoder can generate reliable textures.

MPVPE on Ho3D V3 and DexYCB. Especially on the DexYCB dataset, compared with the latest works, our method reduces the pose error MPJPE by 10%. For hand-object interaction, our Hand-Object pipeline also achieves the lowest PD, SIV and SD for Ho3D V2 and DexYCB. Comparing the results for the DexYCB dataset, our grasp results verify the effectiveness of our pipeline.

Qualitative Results. Visualizations of our hand surfaces in Fig. 3 verify our texture learning and grasp optimization effectiveness.Additionally, Fig. 4 compares our method to state-of-the-art, demonstrating our improved grasp feasibility. More qualitative results can be found in the supplementary material.

Texture Map Results. Visualizations of our results are all rendered with textures from our texture decoder. Fig. 3 also reports appearance similarity between the projected hand I_{re} and its ground truth via SSIM (higher is better). Our model can even estimate the texture for unobserved vertices, *i.e.* the results of view 2, highlighting our model's ability to generalize appearance. We further compare our hand texture reconstruction results with [7] in the supplementary material and show that our reconstruction results are better.

Contact Mask Results. Fig. 6 shows predicted contact masks for DexYCB. Estimated contact regions are similar to the ground truth. Our generated contact masks $\hat{M}_{\text{con}}$ has an average Intersection over Union (IoU) of 72.09%. Efficiency-wise, our contact mask is 5 times faster than point-based methods [19, 43]. Detailed experimental results are given in the supplementary material.

4.2 Ablation Study

MANO Baselines. We also compare with two simple MANO baselines in Table 1. MANO CNN is a pipeline that regresses the MANO parameters through the differentiable MANO layer [39, 46]. MANO Fit uses inverse-kinematics method to fit MANO parameters from our estimated hand vertices. We outperform these baselines; especially on FreiHAND, our method achieves significant improvements (40% over MANO CNN and MANO Fit).

GraspVAE. The impact of GraspVAE is shown in Table 1 (Hand-Only vs Hand-Object). Grasp optimization worsens MPJPE and MPVPE but improves PD, SIV and SD because the optimization limits penetration between the hand and object at the expense of less accurate

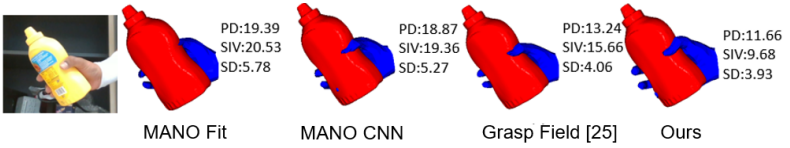

Figure 4: Grasp Results: Comparison of hand-object reconstructions with state-of-the-art. Our method yields a more feasible hand with the lowest PD, SIV and SD.

Figure 5: Grasp Optimization: Predicted hand surfaces with ground truth objects before and after grasp optimization. Differences are highlighted in the boxes. Our grasp optimization significantly reduces PD, SIV and SD, and yields more feasible hand-object interactions.

hand pose and shape. Fig. 5 shows examples before and after optimization. Optimization reduces the penetration and this is verified with lower PD, SIV and SD. Furthermore, compared with RGB2UV pipeline, using the Hand Only VAE refinement reduces pose errors by nearly 10%. This reveals the efficiency of our proposed Hand Only VAE and emphasizes the importance of refinement for predicting UV coordinate maps.

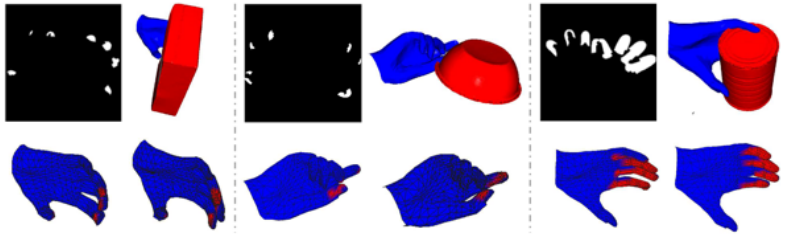

Figure 6: Contact mask results. For each sample, top-left: our predicted contact mask; top-right: our predicted hand surface with ground truth objects; bottom-left: our contact regions from the predicted contact mask; bottom-right: ground truth contact regions.

4.3 Limitations

In our RGB2UV pipeline, we used a silhouette of the hand projection S_{gt} to isolate the hand in the RGB image I. If directly use provided object information, the object can be removed by using an object silhouette but it would break the spatial correlation of hand-object interaction. Therefore, without removing hand object may result in spilling over into the hand texture results; this is especially the case when the area of the hand in the image is very small, or there is significant occlusion from the object (see Fig. 7). These limitations can be improved by considering multi-view information, which would exclude the extreme viewpoints and enable a feasible hand reconstruction to be obtained.

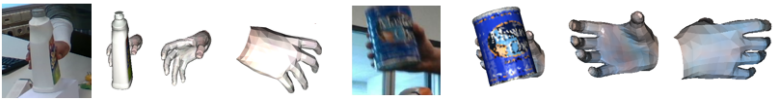

Figure 7: Failure cases. We are limited by only using a single view for texture rendering, and object textures may spill over onto the hand, *i.e*. white and blue hand surfaces from the object.

5 Conclusion

This work proposes a hand surface framework for estimating hand-object interaction from RGB images. We explored UV coordinate maps for hand-object surface modelling and designed the first dense representation to model contact regions. Additionally, we introduce grasp optimization to improve the feasibility of the hand UV coordinate map. Experimental results show that our proposed method outperforms existing 3D hand-object interaction methods.

6 Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (STEM RIE2025 MOE-T2EP20220-0015).

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

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

[3] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *TIP*, 21(4):1488–1499, 2011.

[4] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *ICAR*, 2015.

[5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021.

[6] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *ICCV*, 2021.

[7] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *CVPR*, 2021.

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020.

[9] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. *arXiv preprint arXiv:2112.03028*, 2021.

[10] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *CVPR*, 2022.

[11] Martin de La Gorce, Nikos Paragios, and David J Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, 2008.

[12] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *TPAMI*, 33(9):1793–1805, 2011.

[13] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, 2018.

[14] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

[15] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

[16] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018.

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

[18] Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. Ho-3d_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *arXiv preprint arXiv:2107.00887*, 2021.

[19] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[20] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020.

[21] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from rgb videos. In *3DV*, 2021.

[22] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: non-autoregressive structured modeling for 3d hand pose estimation. In *ECCV*, 2020.

[23] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. *arXiv preprint arXiv:2104.03304*, 2021.

[24] Jiwoo Kang, Seongmin Lee, and Sanghoon Lee. Competitive learning of facial fitting and synthesis using uv energy. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[25] 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.

[26] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *3DV*, 2021.

[27] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *CVPR*, 2015.

[28] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[29] Gun-Hee Lee and Seong-Whan Lee. Uncertainty-aware mesh decoder for high fidelity 3d face reconstruction. In *CVPR*, 2020.

[30] Myunggi Lee, Wonwoong Cho, Moonheum Kim, David Inouye, and Nojun Kwak. Styleuv: Diverse and high-fidelity uv map generative model. *arXiv preprint arXiv:2011.12893*, 2020.

[31] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. *arXiv preprint arXiv:2109.05488*, 2021.

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[33] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020.

[34] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. Parametric hand texture model for 3d hand reconstruction and personalization. In *ECCV*, 2020.

[35] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *ECCV*, 2018.

[36] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[37] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 2017.

[38] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *NeurIPS*, 2015.

[39] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *ECCV*, pages 581–600. Springer, 2020.

[40] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 118(2):172–193, 2016.

[41] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: Hand mesh vertex regression from single depth maps. In *ECCV*, 2020.

[42] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *CVPR*, 2019.

[43] 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 *ICCV*, 2021.

[44] Pengfei Yao, Zheng Fang, Fan Wu, Yao Feng, and Jiwei Li. Densebody: Directly regressing dense 3d human pose and shape from a single color image. *arXiv preprint arXiv:1903.10153*, 2019.

[45] Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *CVPR*, 2020.

[46] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019.