Neural hand reconstruction using an RGB image

Mengcheng LI¹, Liang AN¹, Tao YU¹, Yangang WANG², Feng CHEN¹, Yebin LIU∗

1. Department of Automation, Tsinghua University, Beijing 100086, China
2. Southeast University, Nanjing 210096, China

*Corresponding author, liuyebin@mail.tsinghua.edu.cn
Received: 30 April 2020 Accepted: 27 May 2020

Citation: Mengcheng LI, Liang AN, Tao YU, Yangang WANG, Feng CHEN, Yebin LIU. Neural hand reconstruction using an RGB image. Virtual Reality & Intelligent Hardware, 2020, 2(3): 276—289
DOI: 10.1016/j.vrih.2020.05.001

Abstract Background This study presents a neural hand reconstruction method for monocular 3D hand pose and shape estimation. Methods Alternate to directly representing hand with 3D data, a novel UV position map is used to represent a hand pose and shape with 2D data that maps 3D hand surface points to 2D image space. Furthermore, an encoder-decoder neural network is proposed to infer such UV position map from a single image. To train this network with inadequate ground truth training pairs, we propose a novel MANOReg module that employs MANO model as a prior shape to constrain high-dimensional space of the UV position map. Results The quantitative and qualitative experiments demonstrate the effectiveness of our UV position map representation and MANOReg module.

Keywords Hand reconstruction; Convolution neural network; Single image; Motion capture

1 Introduction

Accurate markerless 3D hand reconstruction is a crucial task in research and development (R&D) communities such as human computer interaction (HCI), robotics, virtual reality (VR), and augmented reality (AR). Traditional methods that require attached sensors¹² or depth information³⁵ are inconvenient in daily applications. The goal of 3D hand reconstruction is recovering dense hand mesh from camera observations, alternate to hand joint positions. For decades, we have recorded significant successes in extracting 3D hand key-points from depth sensor data⁶⁷. Although depth data can provide indirect information on the shape of visible parts of the hands, the limited sensing range and low resolution of depth data hinder its application in large scale VR scenario or full body motion capture setups. Subsequently, following the discovery of powerful deep learning techniques⁸ and linear statistical hand mesh model⁹ (known as MANO), researchers can infer hand shape and pose from a single RGB input with high-fidelity, and demonstrate promising in-the-wild applications¹⁰–¹³.

Particularly, MANO is a parametric hand model trained from approximately 1000 registered raw hand scans, with the low-dimensional principle components analysis (PCA) parameter space approximating the high-dimensional hand surface space. Although PCA parameter space encodes pose and shape information,

2096-5796/©Copyright 2020 Beijing Zhongke Journal Publishing Co. Ltd., Publishing services by Elsevier B.V. on behalf of KeAi Communication Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by/4.0/).
it is limited by raw scans used for training. Consequently, it is a challenging task to exceed the limited representation space of the MANO model. To address this problem, Ge L et al.\cite{12} proposed generating a 3D hand mesh vertices data from images using a Graph CNN-based approach. However, a synthetic dataset or depth data is required to train a satisfactory model that is more difficult to obtain than key-points annotations or direct image visual cues (e.g., masks). How to exceed MANO with only key-points annotations remains a critical task.

Therefore, we propose a novel method to perform hand reconstruction from an image based on a UV position map with CNN called \textit{neural hand reconstruction}. As demonstrated in face alignment task\cite{14}, a UV position map provides correspondences between image coordinates and UV space positions, and can generate denser 3D mesh from compact UV image representations. It is nontrivial to use a UV position map for hand reconstruction, because the highly non-rigid articulated structure of the hand causes two main challenges. First, there is no existing dataset containing the image and UV position map annotation. Second, high-dimension nature of the UV position map representation introduces a significant challenge for CNN to generate appropriate hand pose and shape without large scale ground truth annotations.

To address these challenges, we propose a fully weakly supervised training framework, that utilizes 2D cues (2D key-points and hand masks) and sparse 3D cues (3D key-points) for supervision. Note that, owing to limited training samples and ultra-high dimension of UV parameter space, model trained without paired image and UV position map will yield unrealistic hand shape. Therefore, we also propose a MANOReg module that utilizes MANO parameter space to explicitly regularize UV parameter space, enabling our network to encode plausible shape and pose together with real visual cues. Some hand mesh reconstruction results from our method are shown in Figure 1.

Finally, we examine the effectiveness and efficiency of UV representation by remeshing sampled mesh.

In summary, our main contributions are as follows:

\textbf{(1) A UV representation of hand mesh.} For the first time, we use a UV position map to represent hand mesh model, to overcome limited representation capacity of linear parametric model (MANO). The representation yields significant hand mesh alignment with visual cues.

\textbf{(2) A novel network structure for semi-supervised training.} We propose a novel neural network and a
weakly supervised strategy to infer a UV position map from an RGB image, together with a MANOReg module to constrain the UV parameter space.

(3) We show promising real-time results. Our method runs at 60fps with GPU acceleration, enabling real-time industrial application.

2 Related works

2.1 Hand model representation

Hand representation is key in hand reconstruction, and has been extensively studied recently. Oikonomidis et al. exploits geometric primitives to simulate hand mesh and scaling weights to adjust bone lengths. To fit to depth image, Wan et al. uses 41 sphere surfaces to approximate hand surfaces. With depth images, Khamis et al. learns surface deformation of human hands using linear blend skinning. Li et al. uses predefined hand gestures to represent hand poses. Recently, Romero et al. described a novel hand statistical model, MANO, that trains mesh deformation space from approximately 1000 registered scans. These methods use limited parameter spaces to represent ultra-high dimensional hand surface data. In contrast, we propose the use of UV position maps that map dense 3D hand surfaces to dense UV image pixels, preserving the high-dimensional nature of the hand surface spaces while reducing the difficulty of inference.

2.2 3D Hand reconstruction

A 3D hand reconstruction task aims to recover hand surface model from depth inputs or RGB inputs. In contrast to hand pose estimation, hand reconstruction seeks to obtain possibly dense surface representation. Using depth images, a common method to reconstruct a human hand is driving pre-scanned hand mesh by estimated hand pose. Furthermore, Wan et al. fits fix-sized spheres to depth images. However, in real-world applications, we cannot assume pre-scanned meshes or pre-defined hand shapes. Using RGB inputs, a typical method involves predicting shape and pose parameters of MANO model with end-to-end neural networks. These methods are summarized in Section 2.3. By the MANO model, we can reconstruct a hand mesh without user-specific scan. However, MANO model parameters restrict the expression ability for hand pose and shape unseen in training raw scans.

2.3 Hand pose and shape from single image

Hand pose and shape estimation from an image is an ill-posed problem, wherein the main challenges are depth ambiguity and occlusion (such as self-occlusion and object occlusion). Currently, single view hand dataset has no dense 3D annotations. Notwithstanding these difficulties, deep-learning-based techniques yield significant results. Boukhayma et al. uses ResNet encoder to directly predict shape and pose parameters of a MANO model. Kulon et al. uses graph convolution to hierarchically recover hand surface points of a fixed geometric topology. In our study, we use a UV position map to express hand mesh and a MANO model to weakly supervise the UV parameter space. By the weakly supervised strategy, it is convenient to train dense hand reconstruction network with only sparse 2D or 3D joint annotations.

3 Proposed method

From Figure 2, our pipeline accepts a single hand RGB image as input. It is thereafter coded to a deep convolutional neural encoder that yields a latent feature. Afterwards, the latent feature is decoded into two
Mengcheng LI et al: Neural hand reconstruction using an RGB image

different decoders. The main decoder generates a UV position map that is further transferred to a final hand reconstructed model. The remaining decoder called MANOReg module, is a regularizer that generates MANO hand model to constrain the UV position map space. These two decoders establish a new semi-supervision pipeline to ensure a high-fidelity mesh output.

Now we present details of the method. Initially, we describe our hand representation in Section 3.1. Furthermore, the weakly supervised learning network used for hand mesh inference is described in Section 3.3 and Section 3.4. Finally, we explain the implementation in Section 3.5.

3.1 Hand representation

We utilize a UV position map \( P \in \mathbb{R}^{H \times H \times 3} \) to represent point clouds of a hand surface, where \( H \) is the resolution (width and height) of our three channels UV position map. Unlike common 2D texture map, the three channels of our UV position map store the 3D coordinates of surface points alternate to surface color. The relationship between the UV position map and hand point clouds is shown in Figure 3. The UV position map is created by manually mapping a 3D deformable human hand model into 2D texture. Particularly, we use open sourced 3D animation software Blender to unwarp the UV positions of the MANO hand model. After the unwarping, the UV position map can generate denser surface point cloud

\[ (R, G, B) = (X, Y, Z) \]

![Figure 3: UV position map. Three channels (R, G, B) of UV position map represent 3D coordinates (X, Y, Z).](image)
corresponding to dense UV position map pixels. This dense nature of the UV position map representation explains the sparse joint locations from training samples with larger parameter space. Using only sparse joints alternate to MANO parameters for supervision, we can exceed the expression space of the MANO PCA representation to significantly explain the hand pose distribution of large scale training dataset.

In mapping a 3D hand mesh to a 2D UV position map, we can obtain the linear transformation between 3D mesh and 2D UV position map:

\[ P = W_iV, \]

where \( V \in \mathbb{R}^{N \times 3} \) is 3D coordinates of the MANO hand surface vertices and \( P \in \mathbb{R}^{H \times W \times 3} \) is the UV position map. By the transformation matrix \( W_i \in \mathbb{R}^{(H \times W) \times N} \), we can map a 3D hand mesh into 2D UV position map. In our implementation, the resolution of UV position map was \( H = 256 \) and the MANO had \( N = 778 \) vertices.

Owing to the lack of a large dataset with UV position map annotations, we use MANO hand mesh as semi-supervised by MANOReg module. MANO is a differentiable hand model driven by pose parameters \( \theta \in \mathbb{R}^{10} \) and shape parameters \( \beta \in \mathbb{R}^{10} \). By adjusting the shape parameters \( \beta \), the mesh surface can vary to fit a different person's hand. Noted that the pose parameter \( \theta \) is a principal component from PCA alternate to rotation angle axis. By properly constraining these PCA components, the network can encode the prior shape of the human hand in UV parameter space. Notably, the MANO model has 45 pose PCA, and we select the first 30 principal components to constrain the hand pose.

Given the pose \( \theta \) and shape \( \beta \) of a hand, we can obtain a 3D hand mesh \( M(\theta, \beta) \) with \( N = 778 \) vertices and \( F = 1538 \) triangular faces, together with a 3D skeleton \( J(\theta, \beta) \) with 21 joints. The original MANO model has 16 joints, and we manually attach 5 joints at the finger tips. The functions \( M(\theta, \beta) \) and \( J(\theta, \beta) \) are differentiable.

3.2 Camera model

To unify hand datasets that may not have camera intrinsic annotations, we use an orthogonal projection camera model. Given a rotation matrix \( R \in SO(3) \), a scale parameter \( S \), and a translation vector \( T \in \mathbb{R}^{3} \), a 2D projection of a 3D mesh \( M(\theta, \beta) \) and skeleton \( J(\theta, \beta) \) can be written as

\[ M^{2D} = S\sigma\left( R\sigma(\theta) - \beta \right) + T, \]
\[ J^{2D} = S\sigma\left( R\sigma(\theta) - \beta \right) + T, \]

where \( \sigma \) is the mapping of orthographic projection.

For hand datasets that provide camera annotations, such as STB, and RHD, we adjust the 3D annotations to solve the inconsistency between the orthographic projection and perspective projection. We apply an adjusting rotation \( R_s \in SO(3) \) to the hand 3D annotation. The new annotation is

\[ A' = R_s(A - t_c) + t_c, \]

where \( A \in \mathbb{R}^{3} \) is an original 3D annotation (such as root joint 3D coordinate under perspective projection), and \( t_c \in \mathbb{R}^{3} \) is the mass center of 3D annotations.

\( R_s \) is determined by the angle between camera direction and \( t_c \) direction. \( R_s \) is the correction of the systematic error caused by different projection.

3.3 Network

The input of our network is a human hand RGB image and the output is a 2D UV position map. To address 280
the difficulty of hand pose estimation, we propose a novel network that includes an encoder and two different decoders. The structure of our network is shown in Figure 2.

Given an input image, our network initially encodes it into a latent feature. Thereafter, the latent feature is decoded by two different decoders to obtain two different hand pose representations: the MANO model and the UV position map. The MANO model encodes hand prior information and can make the output pose more sensible, whereas the UV position map is suitable for the structure 2D convolutional neural network. The constraints of the two representations can enhance the network performance.

3.4 Loss function

There are two decoders in our network with two hand representations. The MANO decoder, MANOReg module, output camera parameters and MANO parameters. The camera parameters include a rotation vector \( r \in \mathbb{R}^3 \), a scale parameter \( S \), and a translation vector \( T \in \mathbb{R}^3 \). The MANO parameters include a pose vector \( \theta \in \mathbb{R}^{10} \) and a shape vector \( \beta \in \mathbb{R}^{10} \). The output of the UV map decoder is a three-channel UV position image. The hand 3D joints \( J^{3d} \) and mesh \( M^{3d} \) can be recovered from these representations. We set the subscript MANO and UV to distinguish two outputs.

We use multiple loss to train our network that can be classified into three categories: pose & shape loss \( L_{\text{MANO}} \) and \( L_{\text{UV}} \), cross loss \( L_{\text{cross}} \), and regularization loss \( L_r \):

\[
L = L_{\text{MANO}} + L_{\text{UV}} + L_{\text{cross}} + L_r, \tag{4}
\]

3.4.1 Pose and shape loss

There are three loss terms to constrain the hand joint and mesh output: a 2D joints loss term \( L^{2d} \), a 3D joints loss term \( L^{3d} \), and a silhouette loss term \( L^s \):

\[
L^i = L^{2d} + \alpha^{3d} L^{3d} + \alpha^s L^s, \quad i \in \{ \text{MANO, UV} \}. \tag{5}
\]

The 2D joints loss term ensures the projection of hand joints to match with the 2D annotations:

\[
L^{2d}_i = \sum \| J^{2d}_i - A^{2d}_i \|, \quad i \in \{ \text{MANO, UV} \}, \tag{6}
\]

where \( J^{2d} \) is the 2D projection of the estimated joints \( J^{3d} \), and \( A^{2d} \) is the ground-truth 2D hand joints annotations.

If the 3D annotations are available (e.g., STB and RHD dataset), we use the 3D joints loss term to penalize the distance error between the ground-truth 3D hand joints position and the output 3D joints:

\[
L^{3d}_i = \sum \| J^{3d}_i - A^{3d}_i \|, \quad i \in \{ \text{MANO, UV} \}, \tag{7}
\]

where \( J^{3d} \) is the output joints coordinates, and \( A^{3d} \) is the ground-truth 3D hand joint annotations. The \( A^{3d} \) is the projection adjusted annotations.

Although given hand joints 3D position, the hand skeleton pose is defined, the surface shape information remains unknown. To address this problem, we introduce a 2D projection silhouette loss term to constrain hand shape. The loss term minimizes the difference between hand mesh projection silhouette and the ground truth hand mask. Most hand datasets do not provide 2D mask annotations and the 2D joints annotations can be obtained. We use Grabcut algorithm\(^{[20]} \) to obtain approximate hand masks by setting 2D joints skeletons as initial seed. The silhouette loss function is

\[
L^s_i = \sum \| \sigma(M^{3d}_i, F) - M \|, \quad i \in \{ \text{MANO, UV} \}, \tag{8}
\]

where \( \sigma \) is the projection operator, \( F \) are faces of hand mesh, and \( M \) is 2D hand silhouette. We use the neural mesh render\(^{[21]} \) to render hand silhouette from 3D mesh, thus ensuring that the operation differentiable.
3.4.2 Cross loss

Our network can generate hand meshes from the MANOReg module and UV decoder. These meshes should be consistent with each other:

\[ L_{\text{cross}} = \|RM(\theta, \beta) - M_{uv}\|_2^2 \]  

where \( M(\theta, \beta) \) is the MANO model mesh recovering from the parameters \( \theta \) and \( \beta \), \( R \) is the global rotation matrix, and \( M_{uv} \) is the hand mesh generated from the UV position map. By minimizing this cross loss term, the UV decoder can learn prior hand shape from the MANO model. Our experiments in Section 4.4 show that the cross loss term can form mesh from the UV position map.

3.4.3 Regularization loss

This loss term ensures that the outputs of the two decoders are acceptable. The first regularization loss term constrains the pose and shape parameters of MANO models. The pose and shape parameters arise from the PCA; therefore, appropriate constraint can reduce the mesh distortions. The second regularization loss term is a smooth loss term for the UV decoder's output.

\[ L_r = \|\theta\|_2^2 + \alpha\|\beta\|_2^2 + \alpha^w \sum_{ij} (|P_{ij} - P_{i} + 1| + |P_{ij} - P_{i+1,j}|), \]  

where \( \theta \) is pose parameters, \( \beta \) is shape parameters, and \( P \) is the UV output image.

3.5 Implementation details

Our network is trained with 4 datasets, namely, STB\(^{[22]}\), RHD\(^{[23]}\), PANOPTIC\(^{[24]}\), and FreiHand\(^{[25]}\). We use an adapted ResNet\(^{50}\)\(^{[8]}\) as the image encoder and a symmetrical network as a UV decoder. The MANOReg module is a two-layer full connection network.

We initially pre-train encoder and MANOReg module. Thereafter, we pre-train the UV decoder while maintaining the weights of the encoder and MANOReg module. After pre-training, we train the network from end-to-end.

We test our network using an NVIDIA 1080Ti GPU. Averagely, our model runs at 61.7 FPS, which means that our network is a real-time model. We build a real-time hand pose capture system using our network. We use a Microsoft Kinect V2 camera to capture hand motion. Although it is an RGB-D camera, we only use RGB image as input. Figure 4 shows some real-time running results.

4 Evaluation

4.1 Evaluation dataset

The STB\(^{[22]}\) dataset contains 12 stereo hand motion sequences, and each sequence has 1500 RGBD images with 3D pose annotations. Adapting to Boukhayma et al.'s work\(^{[10]}\), we use the first 10 sequences as training set and the remaining as the evaluation set. The STB dataset does not have 2D hand mask annotations; hence, we use depth image and 2D pose annotations to obtain an approximate hand mask.

The RHD\(^{[23]}\) is a synthetic dataset with human hand and random background images. It has left and right hands, 3D annotations, and 2D mask annotations. We clip the hand area and laterally invert the image to convert the left hand images to right hands. Furthermore, we clean the dataset and discard the images that have more than half of the key-points outside of images or blocked. After data cleaning, we obtained 54326 training images and 3252 evaluation images.
The PANOPTIC\textsuperscript{[24]} is a multiple view hand dataset captured by the Panoptic studio. It contains 3rd person view hand images with several poses together with 2D pose annotations. We adapt to Boukhayma et al.'s work\textsuperscript{[10]} to obtain approximate 2D hand masks. With no 3D annotations, we apply only the 2D loss to this dataset. The size of the PANOPTIC dataset is 14817 and we classify it into 12000 training images and 2817 evaluation images.

FreiHand\textsuperscript{[25]} is a multi-view hand dataset. It contains images with hand controlling different objects. It provides ground truth MANO parameters that can be used to reconstruct ground truth hand meshes. By the UV position mapping defined in Section 3.1, we can transform ground truth hand meshes to UV position map. The annotations of testing set are not provided; hence, we only use 32560 training images.

4.2 Comparison of key-points

To quantitatively evaluate key-points prediction, we examined the percentage of correct points in 3D space (3D-PCK) and the average 3D distance error of the output skeleton. We evaluated the STB and RHD datasets. The definition of hand joints on the STB dataset is different from that in the MANO model. The hand root joint is defined as the wrist under the MANO model, whereas STB selects the middle of the hand palm as the root joint. To be consistent with these definitions, we adjust the location of the root joint in the STB dataset. We extend the line from the middle finger's root node and the center of the palm; thereafter, we choose twice the length point as hand wrist, which is the root point under MANO. Owing to our correction of the orthographic projection, we apply a rigid alignment before calculating the error. This process includes scale, translation, and rotation alignment.

The average 3D distance error and standard deviation is illustrated in Table 1. Figure 5a shows the evaluation results from the STB dataset. We compare our method with some deep-learning-based methods\textsuperscript{[10,11,13,26,27]} and other methods\textsuperscript{[13,28]}. Our method achieves start-of-the-art results. Figure 5b shows the 3D pose estimation results from RHD. This is a synthetic benchmark wherein we outperform existing methods.

Table 1  Average 3D distance error

<table>
<thead>
<tr>
<th></th>
<th>STB</th>
<th>RHD</th>
<th>FreiHand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (mm)</td>
<td>8.978</td>
<td>12.083</td>
<td>10.440</td>
</tr>
<tr>
<td>Std (mm)</td>
<td>5.215</td>
<td>8.560</td>
<td>8.069</td>
</tr>
</tbody>
</table>
4.3 Mesh comparison

Key-points error represents hand skeleton pose error while surface meshes error represents hand shape error. We compare our model with some other state-of-the-art methods. Noted that the mesh topologies are usually different in different methods, we only qualitatively compare the output meshes in this section.

First, we compare our method with Kulon et al. that uses a Graph-CNN network to estimate human hand mesh\cite{19}. Some results are shown in Figure 6. Our meshes are smooth and complete, whereas meshes from Kulon et al. collapse if hands are self-occluded\cite{19}. Besides, our method only requires a single hand image as input, whereas Kulon et al. requires other additional information, such as camera intrinsic and the 3D

![Figure 6](image_url)

Figure 6  Compare mesh quality. The 1st column and 3rd column are our results, the 2nd and 4th column are results from Kulon et al.\cite{19}
location of root joint, that may not be available in real-world application\cite{19}.

We also show some subjective comparison with Boukhayma et al. in Figure 7\cite{10}. Boukhayma et al. takes a single image input to directly estimate MANO parameters and then reconstructs hand surface meshes. Clearly, our results are more consistent with the input image\cite{10}.

![Figure 7](image)

**Figure 7** Subjective comparison. The 1st row is input image, the 2nd row is our result, and the 3rd row is result from Boukhayma et al.\cite{10}

### 4.4 Ablation study

#### 4.4.1 Cross loss

Most hand dataset do not provide dense UV position map annotations; hence, we apply a cross loss term between the output from the MANOReg module and the UV decoder as weak supervision. We evaluate the difference between using and not using this cross loss term. Qualitatively, for without cross loss, the reconstructed hand mesh from UV output collapses if the UV position map ground truth is not provided. From Figure 8, the STB and RHD datasets do not provide UV position map ground truth, whereas the FreiHand dataset has MANO mesh annotations that can be used to generate UV position map. If there is no cross loss term, the output meshes of STB and RHD datasets collapse. Because the constraint is significantly sparse and cannot supervise dense hand surface. The average 3D distance error is illustrated in Table 2. For the STB and RHD datasets, cross loss improves the network’s performance. For the FreiHand dataset, the results are acceptable if cross loss is not used, because this dataset has UV position map ground truth.

![Figure 8](image)

**Figure 8** Ablation Study. 3D PCK curve on STB dataset with different loss term.

| Table 2 Ablation Study. The average 3D distance error |
|----------------|-------------|-------------|-------------|
| Mean (mm)       | STB         | RHD         | FreiHand    |
| w/o cross loss  | 10.106      | 15.020      | 10.227      |
| w/o 3D loss     | 14.093      | 17.251      | 15.721      |
| ours            | 8.978       | 12.083      | 10.440      |
4.4.2 3D losses

We evaluate the difference between using 2D annotations only (2D joints location and 2D silhouette), and using the 2D and 3D annotations. We compare the STB, RHD, and FreiHand datasets. Table 2 shows the average 3D distance error between the estimated 3D joints and the ground truth. The 3D PCK curve on STB dataset is illustrated in Figure 9. The 3D loss can improve network performance.

![Figure 9 Mesh Comparison](image)

Figure 9 Mesh Comparison. The 1st row contains input images, the 2nd row contains our results, and the 3rd row contains results from networks that do not use cross loss. The 1st column is from the STB dataset, the 2nd column is from the RHD dataset, and the last two columns are from the FreiHand dataset.

4.5 Qualitative results

Figure 10 shows some results of our 3D hand reconstructions. From Figure 10, although the input image has motion blur or self-occlusion, our method is effective and efficient. However, we obtain some failed cases mostly caused by incorrect MANO parameters estimation.

5 Discussion

We present a neural hand reconstruction method for monocular 3D hand pose and shape estimation. We use a common image encoder and a symmetrical decoder to generate a UV position map and a MANO mesh to weakly supervise. Our network can be trained end-to-end with 2D or 3D annotated images notwithstanding if the ground truth UV position map is not available. By weakly supervising the MANO mesh, our UV position map representation can learn the prior shape of a human hand. We obtain state-of-art results on some 3D pose benchmarks and generate significant hand mesh on wild images. A possible extension to this work is appending some offset on the UV position image to improve the mesh details.

Acknowledgements: The authors will like to thank Tsinghua University and Southeast University for supporting the research.
Figure 10  Qualitative Results. (a) input image (b) output skeleton (c) mesh recovery from UV position map (d) output UV position map.

References


3 Li J, Wang J X, Ju Z J. A novel hand gesture recognition based on high-level features. International Journal of
DOI:10.1142/s0219843617500220

DOI:10.1109/tvcg.2020.2973057

DOI:10.3233/jifs-179541

DOI:10.5224/c.25.101

DOI:10.1109/iccv.2013.305

DOI:10.1109/cvpr.2016.90

DOI:10.1145/3130800.3130883

DOI:10.1109/cvpr.2019.01110

DOI:10.1007/978-3-030-01231-1_41

DOI:10.1109/cvpr.2019.01109

DOI:10.1109/wacv.2018.00054

DOI:10.1007/978-3-030-01264-9_33

DOI:10.1109/cvpr.2019.01111

DOI:10.1109/cvpr.2015.7298869

DOI:10.1109/cvpr.2014.88
DOI:10.1109/cvpr.2018.00013


DOI:10.1145/1015706.1015720

DOI:10.1109/cvpr.2018.00411

DOI:10.1109/iccv.2017.525


DOI:10.1007/978-3-030-01252-6_8

DOI:10.1109/cvpr.2018.00017