End-to-end Hand Mesh Recovery from a Monocular RGB Image

Xiong Zhang[1*], Qiang Li[1*], Hong Mo[2], Wenbo Zhang[1†], Wen Zheng[1]
[1]Y-tech, Kwai, [2]State Key Laboratory of VR, Beihang University
[1]{zhangxiong,liqiang03,zhangwenbo,zhengwen}@kuaishou.com, [2]mandymo@buaa.edu.cn

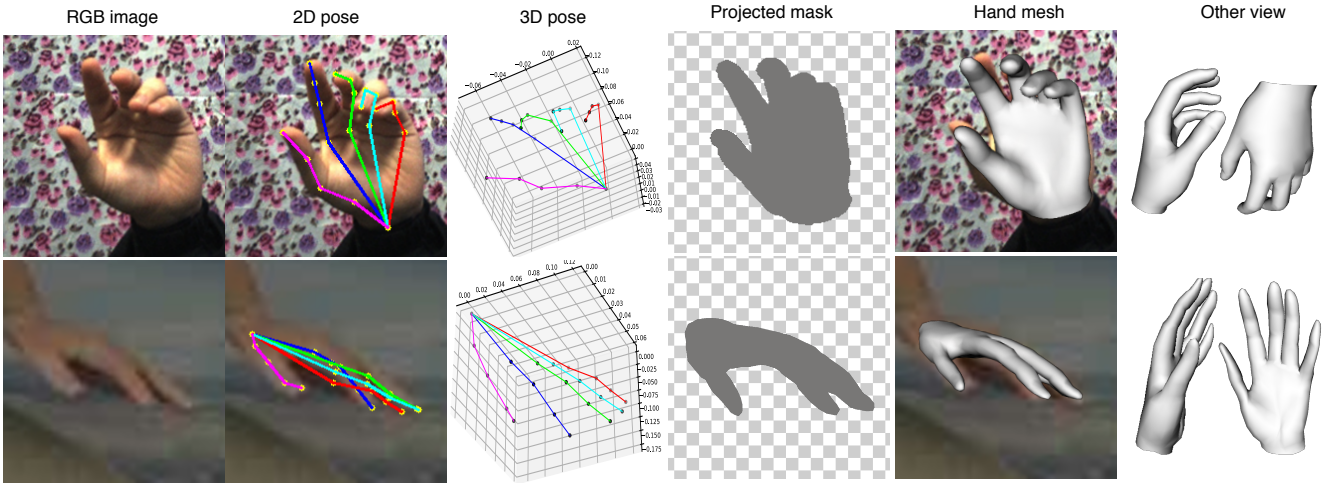

Figure 1: **Hand Mesh Recovery.** Our framework provides a unified scheme for monocular hand image understanding. The first column shows two examples from the STB [55] and RHD [59] dataset respectively. The second and third columns present the estimated 2D joints and 3D joints locations. The fourth column demonstrate the projection mask of the reconstructed hand mesh, while the last two columns illustrate the recovered mesh from several viewpoints. Note that our framework can recover the mesh in a reasonable way even in the presence of severe occlusions.

Abstract

In this paper, we present a HAnd Mesh Recovery (HAMR) framework to tackle the problem of reconstructing the full 3D mesh of a human hand from a single RGB image.[1] *In contrast to existing research on 2D or 3D hand pose estimation from RGB or/and depth image data, HAMR can provide a more expressive and useful mesh representation for monocular hand image understanding. In particular, the mesh representation is achieved by parameterizing a generic 3D hand model with shape and relative 3D joint angles. By utilizing this mesh representation, we can easily compute the 3D joint locations via linear interpolations between the vertexes of the mesh, while obtain the 2D joint locations with a projection of the 3D joints. To this end, a differentiable re-projection loss can be defined in terms of the derived representations and the ground-truth labels, thus making our framework end-to-end trainable. Qualitative experiments show that our framework is capable of recovering appealing 3D hand mesh even in the presence of severe occlusions. Quantitatively, our approach also outperforms the state-of-the-art methods for both 2D and 3D hand pose estimation from a monocular RGB image on several benchmark datasets.*

1. Introduction

Hand image understanding targets the problem of recovering the spatial configuration of hands from natural RGB or/and depth images, which has many applications such as human-machine interaction [20, 40], and virtual/augmented reality [18, 33]. Estimating the spatial configuration is very challenging due to the diverse appearance variation, self occlusions and complex articulations. While many

[1]The code is available at https://github.com/MandyMo/HAMR.
*Indicates equal contribution.
†Corresponding author.

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.00244

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

existing works considered markerless image-based hand image understanding, most of them require depth cameras [1, 11, 23, 34, 41, 44, 45, 53] or multi-view images [2, 14, 38, 43, 51] to handle the above difficulties. As a result, most of these methods are impractical for real-world scenarios where only monocular RGB images are available.

For the monocular RGB setting, many recent works have investigated the specific problems of 2D and 3D hand pose estimation [7, 15, 24, 38, 39, 59]. Considering 2D hand pose, Simon *et al.* [38] utilized bootstrapping to estimate 2D joint locations from single-view RGB images, while the approach has to leverage multi-view images for 3D cases. Gomez-Donoso *et al.* [14] introduced a simple yet effective deep learning architecture for 2D hand pose estimation and also built a multi-view 3D hand pose dataset. Note that single RGB image based 2D hand pose estimation is prone to unreasonable predictions under self occlusions and complex configurations.

Regarding 3D hand pose estimation from monocular RGB, an obvious difficulty resides in the perspective ambiguity caused by missing depth information. Zimmermann and Brox [59] presented the first learning based formulation which augments deep networks to learn a network-implicit 3D articulation prior for resolving perspective ambiguities. Panteleris *et al.* [29] formulated the 3D pose estimation from 2D pose as an inverse kinematics (IK) problem which constrains the solution space to only plausible hand articulations. Mueller *et al.* [24] also introduced several geometry-consistent rules to enforce human hand constraints to relief the hand parsing ambiguity.

Besides enforcing geometric constraints, several recent works have investigated intricate learning strategies. Spurr *et al.* [39] designed a generative hand model represented by a coherent latent space across RGB, 2D and 3D pose modalities, which enables semi-supervised learning. Iqbal *et al.* [15] introduced an intermediate 2.5D pose representation which can be easily estimated from RGB images, and then provided a solution to reconstruct the 3D pose from 2.5D. Cai *et al.* [7] proposed a weakly-supervised method for 3D pose regression by leveraging depth images during training. Though effective at large, the two types of strategies are insufficient to fully capture the inherent properties and diversity of hand articulations.

Aside from the perspective ambiguity, another challenge is the lack of high-quality annotated 3D hand pose data. To address this issue, Zimmermann *et al.* [59] turned to render synthetic data with 3D models, from which the ground truth annotations of 3D joints can be easily obtained. Mueller *et al.* [24] employed CycleGAN [57] to synthesize training data which can enhance the performance of 3D hand pose estimation to some extend. However, networks trained on synthetic data do not generalize well to real-world images due to the domain shift.

In this paper, we propose a hand mesh recovery (HAMR) framework which is also able to produce 2D and 3D pose estimation as side products, as shown in Fig. 1. In contrast to current research on 2D or 3D hand pose estimation, HAMR provides a more comprehensive and flexible mesh representation for hand RGB image understanding. Inside its core, HAMR parameterizes a generic 3D hand model to define the mesh representation in terms of shape and relative 3D joint angles. With this mesh representation, the 3D joint locations can be computed by linearly interpolating between mesh vertexes. In addition, one can obtain the 2D joint locations with a simple projection of the 3D joints.

As a result, HAMR provides a graceful approach to handle perspective ambiguities and data insufficiency. Firstly, the introduced parametric model in HAMR has implicitly characterized the inherent properties of hand comprehensively. This is also validated by our ablation study that adding several specific geometric constraints over fingers can yield very slight improvements. Secondly, since HAMR is end-to-end trainable thanks to the differentiable re-projection loss, our method can exploit enormous incompletely annotated 2D and 3D training samples. The effectiveness of HAMR is evaluated via extensive qualitative and quantitative experiments on benchmark datasets.

2. Related Work

In literature, 3D reconstruction is a long standing problem that has received considerable attention in computer vision and computer graphics. Here, we present a brief overview of two particular topics on 3D face and human body reconstruction from a single RGB image.

Monocular Face Reconstruction. Thanks to the 3D morphable model (3DMM) released by [4, 8, 32], the 3D face reconstruction from single RGB has achieved great success in recent yeas. For example, MoFA [47] provided an end-to-end trainable framework by leveraging 3DMM to define a differential parametric decoder for face pose, shape, expression, skin reflectance and scene illumination. As a step forward, Tewari et al. [46] further considered a regressor for the face properties in addition to the differential parametric model. Genova *et al.* [13] proposed a method for training a regression network for fitting parameters of 3DMM using only unlabeled photographs. Sengupta *et al.* [37] implemented an end-to-end learning framework for producing an decomposition of an unconstrained human face image into shape, reflectance and illuminance.

Monocular Human Body Recovery. Similar progress have been achieved on human body mesh recovery from single RGB. For instance, Bogo *et al.* [5] presented the first method to automatically estimate the 3D pose of the human body as well as its 3D shape from a single unconstrained image. Kanazawa *et al.* [16] described an end-to-end framework for recovering a full 3D mesh of a human body from

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

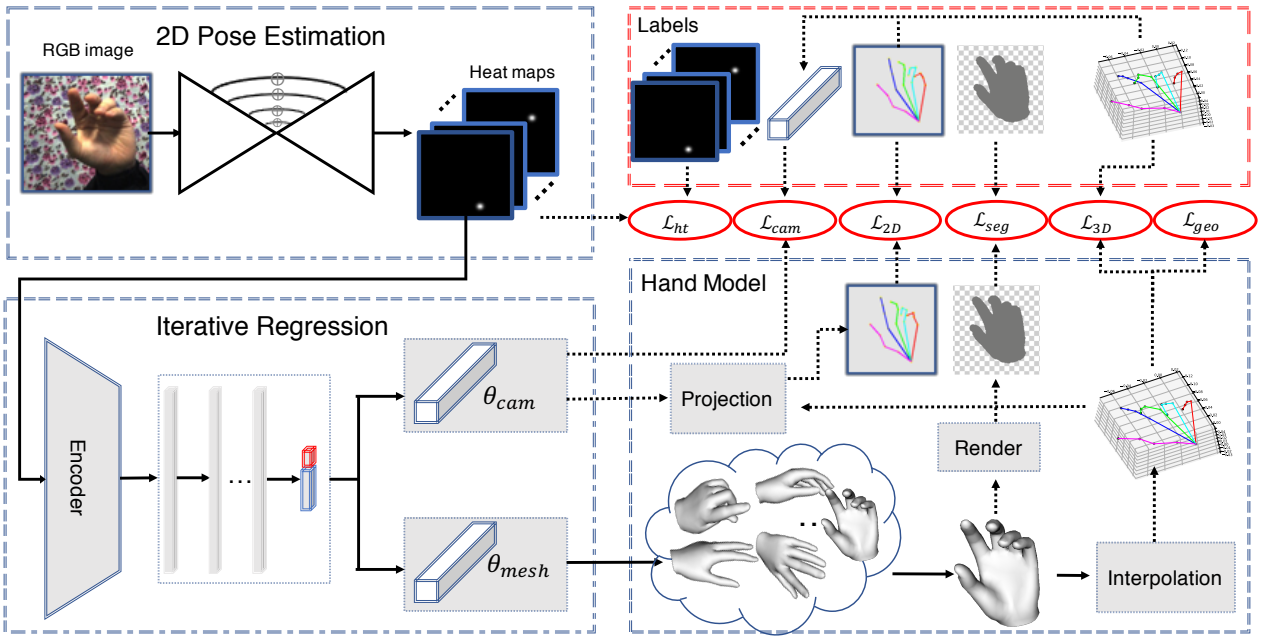

Figure 2: **Framework Architecture**. Our framework consist of three parts: (1) a 2D pose estimation block aims at estimating 2D heat-maps from a single RGB image, (2) a regression module that regresses for 3D parameters $\Theta = \{\theta_{mesh}, \theta_{cam}\}$ by a iterative way, (3) a mesh recovery part for generating the mesh representation, the 2D and 3D joints locations from Θ.

a single RGB image by employ a CNN to regression of the parameters of a parametric model SMPL [19]. Similar to [16], Omran *et al.* [28] performed a novel network structure for human mesh recovery and achieve state-of-the-art result on task of 3D human pose estimation. Recently, BodyNet [50] provided an end-to-end trainable network that considered a volumetric 3D loss, a multi-view re-projection loss, and several kinds of intermediate supervision.

Aside from the above researche works, hand reconstruction from monocular RGB has not drawn sufficient attention from the community. To the best of our knowledge, HAMR provides one of the first approaches to achieve this target among other concurrent works [6, 12].

3. Framework

The main goal of this paper is to present a unified framework for providing a detailed mesh representation $\mathcal{M} \in \mathbb{R}^{N\times 3}$ together with 2D key points $\Phi^{2D} \in \mathbb{R}^{K\times 2}$ and 3D joint locations $\Phi^{3D} \in \mathbb{R}^{K\times 3}$ of hand from a RGB image centered on hand, where N is the vertex count of $\mathcal{M}$ and K indicate the number of key points. We exploit a recent parametric generative model for generating hand mesh, then infer hand pose from the generated mesh. The overall architecture of our framework is illustrated as Fig. 2.

3.1. Hand Mesh Recovery

Hand Mesh Representation. We utilize MANO [36] as the generic 3D hand model in our framework. In particular, MANO factors hand mesh into *shape* - mainly models hand properties such as finger slenderness and palm thickness and *pose* - how the 3D surface deforms with articulation. Similar to SMPL [19], MANO parameterizes a triangulated mesh $\mathcal{M} \in \mathbb{R}^{N\times 3}$ with a set of parameters $\theta_{mesh} = \{\vec{\beta}, \vec{\theta}\}$, where $\vec{\beta} \in \mathbb{R}^{10}$ denote the shape parameters and $\vec{\theta} \in \mathbb{R}^{K\times 3}$ are the pose parameters. Technically speaking, $\vec{\beta}$ represents the coefficients of PCA components that sculpting the identity subject and $\vec{\theta}$ denotes the relative 3D rotation of K joints on Rodrigues Vector representation. The model is defined by a mean template mesh $\bar{\mathbf{T}} \in \mathbb{R}^{N\times 3}$ represented by concatenating N vertices in rest pose $\vec{\theta}^*$; a set of blend weights $\mathcal{J} \in \mathbb{R}^{N\times K}$ and $\mathcal{W} \in \mathbb{R}^{N\times K}$; a blend function $B_S : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{N\times 3}$ that takes $\vec{\beta}$ as input and outputs a blend shape to characterize the identity subject; a pose dependent blend function $B_P : \mathbb{R}^{|\vec{\theta}|} \mapsto \mathbb{R}^{N\times 3}$ that compensates for the deformation caused by pose variations.

Following standard skinning procedure, the shape and pose deformations $B_S(\vec{\beta})$, $B_P(\vec{\theta})$ are applied to the mean template $\bar{\mathbf{T}}$. To obtain the final mesh, the mesh is then posed by rotating each finger part around joints $J(\vec{\beta})$ using the blend skinning function $W(\cdot)$,

$$M(\vec{\beta}, \vec{\theta}) = W(T(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}), \quad (1)$$

$$T(\vec{\beta}, \vec{\theta}) = \bar{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta}). \quad (2)$$

With the mesh model, one can easily reconstruct a hand mesh by specifying certain values of the parameters $\vec{\beta}$ and

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

$\vec{\theta}$. It is worth mentioning that both $T(\cdot)$ and $W(\cdot)$ are differentiable functions of $\vec{\beta}$ and $\vec{\theta}$ [19, 36], which are favorable for composing the mesh model with other modules.

Derived Hand Pose Representations. Given the recovered mesh, we can compute the 3D joint locations Φ^{3D} via linear interpolations between the vertexes of the mesh, while obtain the 2D joint locations Φ^{2D} with a projection of the 3D joints. More specifically,

$$\Phi_k^{3D} = \sum_{i=1}^{N} \mathcal{J}_{ik}\mathcal{M}_i, \tag{3}$$

$$\Phi_k^{2D} = \prod \left(\Phi_k^{3D}|\theta_{cam}\right), \tag{4}$$

where Φ^{3D} is a set of 3D coordinates $\{(x,y,z)_k\}_{k=1}^K$, Φ^{2D} is a set of corresponding 2D coordinates $\{(u,v)_k\}_{k=1}^K$, and $\theta_{cam} = \{(s,t_x,t_y)\}$ denotes the camera parameter. The 3D-to-2D projection function $\prod(\cdot)$ is defined as below,

$$\prod((x,y,z)|(s,t_x,t_y)) = (s(x+t_x), s(y+t_y)), \tag{5}$$

where we used the weakly perspective camera model.

Silhouette Constraints. By impose supervision over 2D/3D hand pose only, the mesh may simply deform to best fit joint locations and ignoring hand surfaces, which lead to abnormal mesh representation. For a hand image, the projection of a well-defined hand mesh should be consistent with the corresponding silhouette. This constraint plays an indispensable role in single RGB image based 3D reconstruction [4, 5, 17, 31, 49]. In this work, we introduce the silhouette consistent loss to refine the hand mesh and camera parameter estimation. In particular, we utilize the Inverse Graphics technique [17, 21] to obtain a rendered mask image from the reconstructed hand mesh given by Eq. 1, then penalize misalignment between the rendered mask and the ground-truth silhouette to train the network.

Geometric Constraints. Upon the derived 3D hand pose representation, it is also helpful to apply several kinds of geometric constraints. Let P_a, P_b, P_c, P_d be the 4 joints of the fingers (except thumb) in turn (tip to palm order), where $P_i \in \mathbb{R}^3, i \in \{a,b,c,d\}$. For convenience, we define $\overrightarrow{V_{ab}}$ as $P_a - P_b$, which is similar to $\overrightarrow{V_{bc}}$ and $\overrightarrow{V_{cd}}$. A correct hand articulation should obey the following constraints. First, P_a, P_b, P_c, P_d are in the same plane, which implies:

$$(\overrightarrow{V_{ab}} \times \overrightarrow{V_{bc}}) \cdot \overrightarrow{V_{cd}} = 0. \tag{6}$$

Second, the rotate direction from $\overrightarrow{V_{ab}}$ to $\overrightarrow{V_{bc}}$ is consistent with that of $\overrightarrow{V_{bc}}$ to $\overrightarrow{V_{cd}}$, which implies:

$$(\overrightarrow{V_{ab}} \times \overrightarrow{V_{bc}}) \cdot (\overrightarrow{V_{bc}} \times \overrightarrow{V_{cd}}) \geq 0. \tag{7}$$

Note that these two types of geometric constraints can be reformulated as regularizers in a loss minimization problem by the Lagrangian multiplier method [35].

Loss Function. For our target, to recover the hand mesh from a single RGB image, we leverage deep convolutional networks to fit the mesh parameters θ_{mesh}. However, in real-world scenario, it is almost impossible to obtain the ground-truth mesh parameters when annotating from single RGB images. Fortunately, our HAMR framework can define derived 3D and 2D joint locations from the mesh. By doing so, we can train HAMR with widely-available 3D and 2D annotations thus enable the mesh reconstruction.

We shall point out there is an issue coming from the camera parameters θ_{cam}. To make our framework fully automatic and accurate, we direct another flow from the base deep convolutional networks to estimate the camera parameters. During training, the "ground-truth" camera parameters are computed from the ground-truth paired 3D and 2D annotations. More specifically, the "ground-truth" scale s is calculated as the ratio between the average 2D and 3D bone lengths. Then, the "ground-truth" offsets are computed as $(t_x, t_y) = \frac{1}{K}\sum_{k=1}^{K}((u_k,v_k)/s - (x_k,y_k))$.

Considering training the base deep convolutional networks, we resort to the intermediate supervision to make sure the earlier stage well captures the image semantics. In particular, the 2D pose estimation block is supervised by stacked heatmaps of 2D ground-truth labels. For each 2D joint, the ground-truth heatmap is defined as a Gaussian-blurred confidence image centered at that joint location.

To this end, we employ L2 loss between the derived 3D and 2D representations and ground-truth labels which result in $\mathcal{L}_{3D}$ and $\mathcal{L}_{2D}$ respectively. In addition, the geometric constraints are reformulated as regularizers leading to $\mathcal{L}_{geo}$ which defines over the predicted 3D poses. Equally important, we penalize the misalignment between the rendered mask and ground-truth silhouette via L1 loss leading to $\mathcal{L}_{seg}$. Furthermore, L2 loss is utilized to supervise the estimated camera parameters with ground-truth camera parameters leading to $\mathcal{L}_{cam}$. As for the intermediate supervision, we compute the pixel-wise distances between the predicted and ground-truth 2D heatmaps that yield $\mathcal{L}_{ht}$.

The whole process is fully differentiable with respect to all the learnable parameters, thus making our HAMR framework end-to-end trainable. The overall loss function is summarized as

$$\begin{aligned}\mathcal{L} = \lambda_{3D}\mathcal{L}_{3D} + \lambda_{2D}\mathcal{L}_{2D} + \lambda_{geo}\mathcal{L}_{geo} \\ + \lambda_{cam}\mathcal{L}_{cam} + \lambda_{ht}\mathcal{L}_{ht} + \lambda_{seg}\mathcal{L}_{seg},\end{aligned} \tag{8}$$

where $\{\lambda_{3D}, \lambda_{2D}, \lambda_{geo}, \lambda_{cam}, \lambda_{ht}, \lambda_{seg}\}$ are hyper-parameters to trade-off among different types of supervision over the whole framework.

3.2. Iterative Regression Module

We apply a regression module to fit the camera parameters θ_{cam} and the mesh parameters θ_{mesh}. However, the complicated domain gap makes it difficult to

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

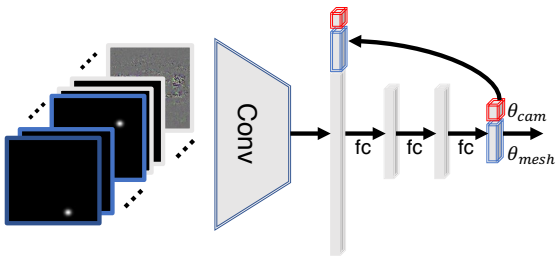

Figure 3: **Architecture of Iterative Regression Module**. This module takes the cross-level features as input and regresses the camera and mesh parameters in an iterative way.

produce reasonable estimates in one go. Several works [9, 10, 27, 52, 58] revealed that a cascade and coarse-to-fine method shall be more appropriate than a one-pass solution. Inspired by these studies, we implement an iterative regression module to fit the camera and mesh parameters from semantic features extracted from previous 2D pose module. Formally, we take the current parameters $(\theta^t_{cam}, \theta^t_{mesh})$ as additional inputs upon the image feature ϕ, and estimate the more accurate $(\theta^{t+1}_{cam}, \theta^{t+1}_{mesh})$. As illustrated in Fig. 3, the iterative regression module consists of a simple fully convolutional encoder and multiple fully connected layers.

To make the predicted θ_{cam} more accurate, we enforce a strong supervision by utilizing the ground-truth camera parameters computed from paired 3D and 2D annotations. The camera parameters θ_{cam} consist of three real numbers (s, t_x, t_y) for projecting 3D joint (x, y, z) to 2D key point (u, v), and the projection function follows Eq. 5.

3.3. 2D Pose Estimation

Similar to recent methods [22, 56, 59], we employ a cascade encoder-decoder style network to predict the 2D Gaussian like heat-maps $\Phi^{ht} \in \mathbb{R}^{K\times H\times W}$ for the 2D pose estimation task, where K indicates the number of joints and $\{H, W\}$ are the resolutions of the heat-maps. Each key point has a corresponding heat-map, and each pixel value on the heat-map indicates the confidence of the key point locating in that 2D position. Further, we share the similar view with [7, 24] that one can hardly resolve perspective ambiguity when applying a direct regression from heat-maps to 3D pose. Actually, different 3D joint locations may be projected to the same 2D key point. Therefore, aside from the K heat-maps for the 2D key point's locations, this block outputs an extra $M - K(M > K)$ heat-maps to handle the issue. In addition, we concatenate the intermediate-layer features with the M heat-maps, and feed them into the following iterative regression module.

As for the network architecture, we utilize the Stacked Hourglass network [26] for 2D hand pose estimation with several slight modifications. Similar to [25], we replace the residual block with simple 3×3 convolution, and use average-pooling to replace the max-pooling. Besides, a batch-normalization layer is added after each 3×3 convolution layer for faster training.

4. Experiments

To evaluate the efficacy of HAMR, we first present qualitative results on the recovered mesh from single RGB images. Secondly, since there exists no ground-truth mesh representation for comparison study, we quantitatively evaluate the superiority of HAMR on the task of 3D and 2D hand pose estimation instead. Finally, we perform ablation studies to better understand the impact of different design strategies on the 3D hand pose estimation task.[2]

4.1. Experiment Settings

Datasets. We mainly involve the Rendered Hand Dataset (RHD) [59], the Stereo Hand Pose Tracking Benchmark (STB) [55] and the Dexter Object (Dexter) [42] dataset. The RHD is a 3rd view based synthetic dataset that provides 41258 training and 2728 testing samples, and each sample contains an RGB image, a depth image, a segmentation mask image and both 2D and 3D annotations of the 21 standard key points. In this paper, we follow the standard training/testing division of RHD. Furthermore, for each sample, we only exploit the RGB images and the corresponding 2D and 3D annotations.

The STB dataset contains sequences with six different backgrounds, and each background has two sequences (1500 frame per-sequence). Similar to [7], we shift the root joint of STB to make it consistent with RHD, and we split the dataset into training and testing part following the rule given by [59]. Besides, we only exploit the RGB image and corresponding 3D annotations.

The Dexter dataset consists of six sequences with 2 actors (1 female), and various interactions with a simple object shape. Fingertip positions and cuboid corners are manually annotated for all sequences. Due to the incomplete hand annotation, therefore, similar to [59], we use this dataset for evaluating the cross-dataset generalization performance of our framework.

We shall point out that, for RHD dataset we obtain the hand mask by parsing the rendered depth map to train the network, Fig. 5 illustrate some paired hand images and corresponding masks.

Data Augmentations. Both RHD and STB share the same augmentation algorithm. We crop each image centered on hand and resize it to 256×256, then apply scaling (0.9 - 1.2), rotation (-/+60°). Finally, to resist the color variety, we perform color-jittering with the following configu-

[2]In practice, the MANO model only has 16 key points and lacks 5 fingertip endpoints. Thus, in our experiments, we manually select vertexes with the index of 734, 333, 443, 555, 678 from generated mesh as the locations of five fingertip endpoints.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

Figure 4: **Mesh Recovery Results.** The left four columns present five representative examples from the STB dataset, while the right four columns from the RHD dataset. For each sector, the first column shows the RGB images, the second to fourth columns are the rendered meshes from different views.

Figure 5: **Paired Mask Samples.** Each sample contains two columns. The first column shows the original RGB image; the second column presents the corresponding mask.

rations: brightness (0.2 - 1.8), contrast (0.3 - 1.7), saturation (0.4 - 1.6), hue (0.2 - 1.8), and add random noise governed by Gaussian distribution $\mathcal{N}(0, 0.01)$.

Network design. We use a two-stack Stacked Hourglass as the pose module, the hourglass module output 64 heat-maps which is numerically larger than the number of hand joints (21) for providing more features, the resolution of heat-maps is 64×64 and the σ^2 used for generating Φ^{ht} is set as 2.5. The iterative regression block consists of a fully convolutional encoder and a fully connected regressor. The encoder is built with 6 stride convolutional layers, for each convolutional layer, the kernel size is 3×3, and the stride is 2. The regressor is composed of 2 fully connected layers with 2048 neurons, followed by a final layer of 22 neurons. After each fully-connected layer, we insert a dropout layer with drop probability of 0.4 to prevent overfitting.

Training. We implement our framework with Pytorch [30], and we adopt the end-to-end training manner. We minimize the total loss in Eq. 8 to train the whole model, where the value of hyper-parameters λ_{3D}, λ_{2D}, λ_{geo}, λ_{cam}, λ_{ht}, λ_{seg} are empirically set to 1000, 1, 1, 0.1, 100 and 10 respectively, while one can also tune these hyper-parameters via the grid search method [3]. The RMSprop [48] is used to optimize the framework. We begin training the whole framework with a learning rate 2×10^{-4}, then decays the learning rate by a factor of 0.7 when validation loss plateaus.All the experiments are conducted on GeForce GTX TITAN Xp GPU with CUDA 8.0.

4.2. Visualization of Appealing Mesh

In order to verify the quality of the generated mesh and the robustness of the framework in various cases, we draw some representative samples from STB and RHD testing sets as illustrated in Fig. 4. Experiments demonstrate that our framework is capable of generating high-quality, appealing hand meshes.

Further, our framework is robust to reconstruct hand meshes accurately even in severe hard scenarios such as poor lighting, image truncation, self-occlusion, and exaggerated articulation. For illustration, we refer S_i to the ith

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

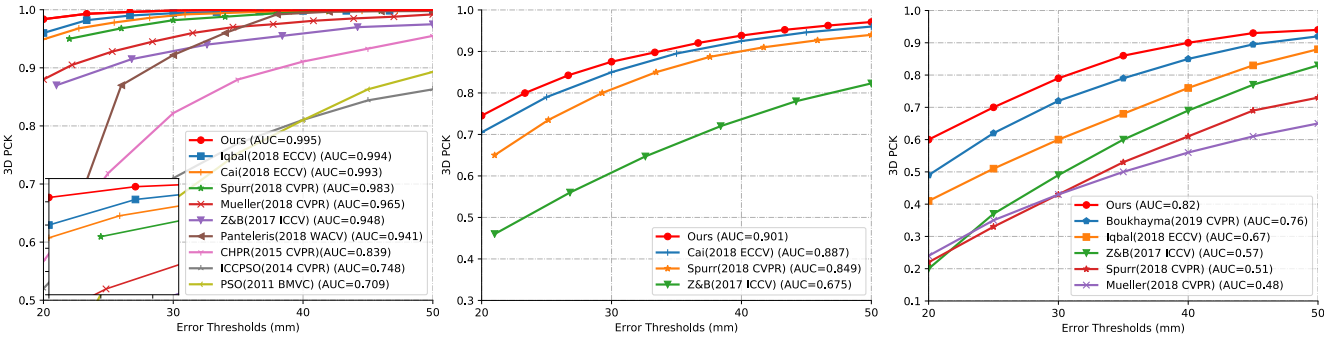

Figure 6: **Quantitative Evaluation.** The left plot presents the 3D PCK performance on the STB dataset, in which our approach is comparable with state-of-the-art methods. The middle plot shows the 3D PCK performance on the RHD dataset and our method achieves the best results compared with state-of-the-art methods. The right plot demonstrate the 3D PCK performance on the Dexter dataset, our framework outperform state-of-the-art methods by a large margin.

sample drawn from STB in Fig. 4, which is similar to R_i for RHD. The comparison among R_1, R_4 and R_5 reveals that our method can reconstruct the mesh correctly even with poor lighting. Beside, R_2 indicates our approach can estimate the mesh reasonably even in the presence of image truncation. By comparing S_1 to S_5 together with R_3, one can conclude that our framework is capable of recovering anthropometrically reasonable mesh representation even with heavily self-occlusion or complex exaggerated hand pose.

4.3. Quantitative Evaluation

To be consistent with [7, 24, 39, 59], we evaluate the performance of 3D hand pose estimation with the measure of the area under the curve (AUC) on the percentage of correct key points (PCK) score with different thresholds. All the comparison methods are evaluated on both RHD, STB and Dexter datasets, experimental results are shown by Fig. 6.

On STB dataset, the 3D PCK curves of [7, 15, 24] are intertwined to each other for the reason that STB dataset is relatively small, and lacks diversity. Our method performs competitively with all the comparison methods [7, 15, 24, 29, 39, 54, 59], which is reasonable considering the saturated performance on this dataset. In contrast, RHD dataset is relatively complex and more diverse, our method is superior to those proposed by [7, 39, 59] and achieves state-of-the-art result. Similarity, on Dexter Object dataset, our method largely outperform most state-of-the art methods [6, 15, 24, 39, 59]. It is consistent with our expectation that introducing the parametric hand model can greatly help solving the perspective ambiguity issue which in turn improves the accuracy of hand pose estimation.

4.4. Generalization Performance Evaluation

Similar to [15, 24, 59], we evaluate the generalization performance of our framework on Dexter dataset [42].

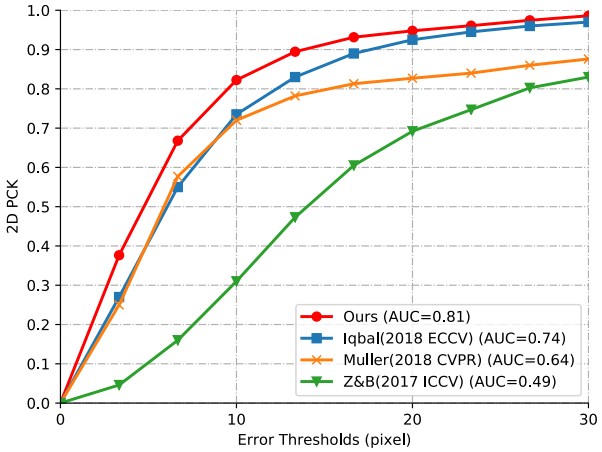

Figure 7: **Generalization Performance Evaluation.** The plot illustrates the results of 2D hand pose estimation on the Dexter Object dataset. Our method outperforms those proposed in [15, 24, 59].

Specifically, we train the model on RHD and STB datasets, then perform cross-dataset testing in two ways without fine-tuning. Firstly, following conventional practice, we take the AUC of 2D PCK as evaluation metric to evaluate the performance on 2D hand pose estimation. As shown in Fig. 7, our method achieves better result than those reported in [15, 24, 59] where the Dexter dataset contains certain ratio of occluded hand images. It is not astonishing that the mesh is capable of modeling inherent geometric prior of hand, which further solve uncertainty caused by occlusion.

Secondly, we examine the quality of estimated hand mesh. As illustrated by Fig. 9, our framework is capable of reconstructing high-quality mesh even the finger is occluded by some foreign matter or in the presence of high dynamic range lighting. The above experiments verified the generalization performance of our framework, proved the

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

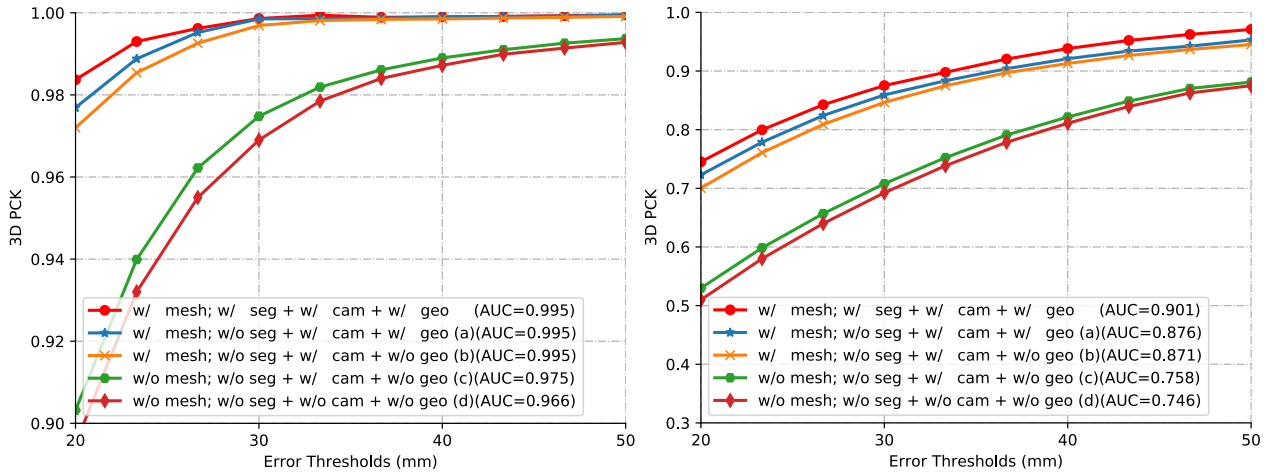

Figure 8: **Ablation Studies.** The left plot shows the 3D PCK performance on the STB dataset and the right plot on the RHD dataset. Note that **w/** and **w/o** abbreviates *with* and *without* respectively. Besides, **w/ mesh** means obtaining the 3D hand pose from the estimated mesh, **w/o mesh** denotes estimating the 3D joint locations with the iterative regression module, **w/ cam** applies supervision over the pose camera parameters, **w/ geo** denotes taking the geometric loss into consideration, **w/ seg** indicates considering the silhouette consistent constraint.

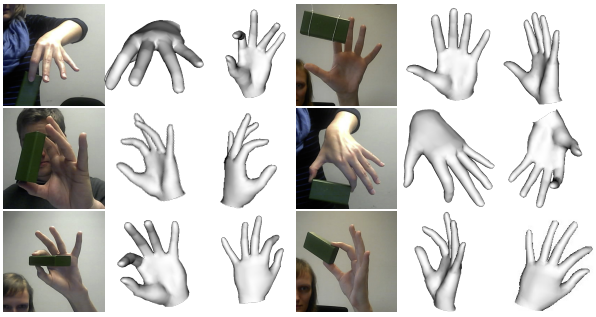

Figure 9: **Recovered Meshes from Dexter Dataset.** Each row presents two examples. For each example, the first column shows the RGB image, the second to third columns illustrate the recovered mesh from different views.

usefulness of mesh in solving the ambiguity caused by occlusion, and implied the essential importance of introducing the parametric model for hand pose estimation.

4.5. Ablation Studies

To better understand the impact of different design choices, we evaluate our framework in various settings. Four main design choices exist in this work: the effect of introducing parametric model on the task of pose estimation, the effect of the silhouette consistent constraint, the impact of pose supervision over camera parameters, and the influence of the geometric loss. We still take AUC of 3D PCK as the evaluation metric for RHD and STB datasets. Fig. 8 compares our full scheme **(w/ mesh + w/ cam + w/ geo + w/ seg)** with four alternatives, experiments a , b, c and d, under various settings.

We find that introducing parametric mesh model improve the performance of hand pose significantly, since the mesh model captures comprehensive properties of hand. Besides, adding specific geometric constraints over fingers yield marginal performance improvement because the mesh model has already covered the inherent geometric constraints of hand. Furthermore, pose supervision over camera parameters gained certain accuracy improvement because the $\mathcal{L}_{cam}$ term plays the role of calibrating the projection procedure from 3D to 2D. Equal important, penalize misalignment between the rendered mask and the ground-truth silhouette also gained performance improvement since the $\mathcal{L}_{seg}$ term can refine the hand shape and pose prediction.

5. Conclusion

Hand image understanding is a widely researched topic and has many real-world vision applications. While hand pose estimation has been well studied in the literature, there still exists little research on the problem of hand mesh recovery from single RGB images. The proposed HAMR enriches this field by leveraging from a generic 3D hand model to achieve a mesh representation. With the mesh model, the framework can produce 2D and 3D hand pose estimations as well. Qualitatively, HAMR successfully recovered reasonable hand meshes from single RGB images even under severe occlusions. Quantitatively, the superiority of HAMR was empirically confirmed on 2D and 3D hand pose estimation tasks compared with state-of-the-art methods. Along this direction, one can expect future advancements in both hand mesh recovery and pose estimation.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision (ECCV)*, 2012.

[3] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 2012.

[4] 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*, 1999.

[5] 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 (ECCV)*, 2016.

[6] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *European Conference on Computer Vision (ECCV)*, 2018.

[8] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2014.

[9] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[11] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[13] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. Large-scale multiview 3d hand pose dataset. *arXiv preprint arXiv:1707.03742*, 2017.

[15] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *European Conference on Computer Vision (ECCV)*, 2018.

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] Taehee Lee and Tobias Hollerer. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[19] 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)*, 2015.

[20] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. Vulture: a mid-air word-gesture keyboard. In *ACM Conference on Human Factors in Computing Systems*, 2014.

[21] Stephen Robert Marschner and Donald P Greenberg. *Inverse rendering for computer graphics*. Citeseer, 1998.

[22] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems (NPIS)*, 2017.

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2016.

[27] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[28] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *International Conference on 3D Vision (3DV)*, pages 484–494, 2018.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

[29] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[30] Adam Paszke, Soumith Chintala, Ronan Collobert, Koray Kavukcuoglu, Clement Farabet, Samy Bengio, Iain Melvin, Jason Weston, and Johnny Mariethoz. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, may 2017.

[31] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. IEEE, 2009.

[33] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, 2013.

[34] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[35] R Tyrrell Rockafellar. Lagrange multipliers and optimality. *SIAM review*, 35(2):183–238, 1993.

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

[37] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: learning shape, reflectance and illuminance of faces in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[39] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *ACM Conference on Human Factors in Computing Systems*, 2015.

[41] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[42] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision (ECCV)*, 2016.

[43] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[44] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, 2015.

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

[46] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[48] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

[49] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[50] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *European Conference on Computer Vision (ECCV)*, 2018.

[51] Robert Wang, Sylvain Paris, and Jovan Popović. 6d hands: markerless hand-tracking for computer aided design. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2011.

[52] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[55] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *IEEE International Conference on Image Processing (ICIP)*, 2017.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.

[56] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[58] Xiangyu Zhu, Zhen Lei, Stan Z Li, et al. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[59] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 05:25:08 UTC from IEEE Xplore. Restrictions apply.