Estimating 3D hand poses from single RGB images for industrial robot teleoperation*

Digang Sun[a], Ping Zhang[a,*]

[a]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

ARTICLE INFO

Keywords:
Hand pose estimation
Extended heatmap attention
Prior knowledge
Mesh supervision

ABSTRACT

3D hand pose estimation from single RGB images is challenging because self-occlusion and the absence of depth make it difficult to regress relative depth between hand joints and to produce biomechanically feasible hand poses. To address these issues, we propose a Prior-knowledge Aware and Mesh-Supervised Network (PAMSNet) to integrate the knowledge implied in the hand's articulated structure and that contained in hand meshes. We explore and interpret the knowledge from a novel perspective inspired by cognitive psychology and forge it into implicit and explicit categories. The former is difficult to be formulated and should be learned from data while the latter can be embedded in loss functions. We estimate 3D poses by fusing the hand's 2D pose and texture features. Hand meshes produced by a parameterized hand model are employed as a regularizer to optimize feature extraction. Furthermore, an extended 128-joint hand skeleton model is proposed to generate denser heatmaps to provide approximately mask-aware spatial attention. Experimental results show that our method is competitive with the state-of-the-art on two public datasets and is superior in generalization ability, with a more efficient architecture. Finally, we apply 3D hand poses to control the moving direction and orientation of the robot end-effector (EE).

1. Introduction

The hand, which plays an important role in human communication, is widely used in human-computer interaction. Hand poses, represented by the positions of hand joints, can be utilized to express various intentions. As a result, hand pose estimation methods, especially based on vision, have witnessed significant progress in recent years. Particularly, the input of these methods has evolved from depth [1, 2, 3] to RGB images owing to the availability of RGB cameras. However, due to a lack of depth information, estimating 3D hand poses from single RGB images is more challenging than from depth ones. To address this problem, some approaches resort to images taken from multi-view [4], while some other methods take 3D hand pose estimation as a by-product of the hand shape reconstruction task [5]. In addition, to prevent producing infeasible 3D hand poses as much as possible, some geometric or biomechanical constraints are proposed [6, 7].

The methods mentioned above have significantly improved the performance of 3D hand pose estimation. However, a multi-view sensing system needs multiple cameras placed at different angles, which could damage the convenience and naturalness in some applications, e.g., human-machine interaction. Deriving 3D hand poses from hand shape reconstruction tasks means we have to reconstruct the hand shape and use it during training and inference stages, which might be heavy for hand pose estimation. On the other hand, the knowledge implied in the articulated structure of the hand, which could be beneficial to produce more accurate and feasible 3D hand poses, is still worth further exploration. Moreover, it is necessary and feasible to integrate various kinds of prior knowledge in an effective and efficient way.

The hand skeleton is of an articulated structure, implying some interesting prior knowledge. For example, if we decompose a 3D hand pose into a 2D hand pose and the relative depth between hand joints, we will get that, for any two adjacent hand joints, the larger the distance between them in the image plane, the smaller the distance in depth. It means regressing relative depth between hand joints from their 2D poses is possible. Generally, the prior knowledge can be forged into implicit and explicit categories. The former includes the part that varies from person to person and is hard to be formulated, while the latter contains some biomechanical constraints that apply to all ordinary people.

Although 2D coordinates of hand joints are requisites for 3D hand pose estimation, it is insufficient to accurately regress relative depth between hand joints using only them, since there are multiple 3D hand poses that can project to the same 2D pose. In this case, hand texture features could provide complementary information to deal with ambiguities. That is to say, it is necessary to combine 2D poses and texture features of the hand together. In addition, heat maps of 2D hand poses can provide pose- and even mask-aware spatial attention for extracting hand texture features of interest.

Hand meshes generated by MANO [8], a type of parameterized hand model, contain a higher magnitude of information than hand poses and are always biomechanically feasible. Thus, they can be employed as a higher level of supervision to optimize feature extraction for pose estimation.

Based on these ideas, we propose a Prior-knowledge Aware and Mesh-Supervised Network (PAMSNet) and design a set of loss functions related to prior knowledge and

*This document is the results of the research project funded by the Guangdong Major Project of Basic and Applied Basic Research (2023B0303000016).

*Corresponding author

cssundg@mail.scut.edu.cn (D. Sun); pzhang@scut.edu.cn (P. Zhang)

ORCID(s): 0000-0002-5336-9851 (D. Sun); 0000-0002-6238-7963 (P. Zhang)

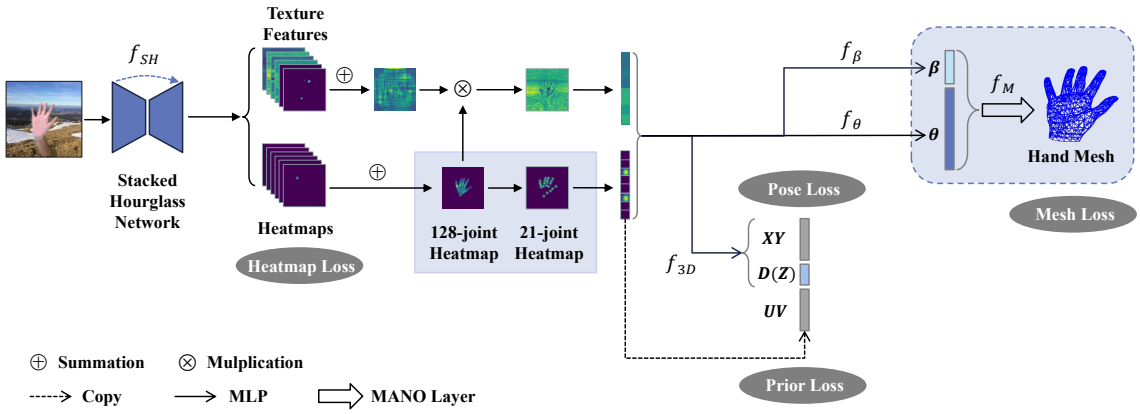


Figure 1: Architecture of the proposed Prior-knowledge Aware and Mesh-Supervised Network (PAMSNet). It mainly comprises a stacked hourglass network as the backbone, a relative depth regressor, and a MANO layer (in the dotted-line bounding box). The MANO layer will be removed when doing inference.

meshes, so that the network can be trained in an end-to-end fashion. The architecture of our proposed PAMSNet is shown in Fig. 1. It's worth noting that the mesh supervision mechanism can be removed when the network is put into practice so that the storage space and inference time can be reduced.

We test our method on the STB [9], RHD [10], and FreiHAND [11] datasets. We also test it on a custom hand gesture dataset for cross-dataset evaluation. We compare our method with the state-of-the-art in terms of evaluation accuracy and generalization ability.

Finally, we extract the pointing direction of the index finger and the orientation of the hand from 3D hand poses, and apply them to indicate the moving direction and the orientation of the robot end-effector, respectively, in a natural and efficient teleoperation manner.

The main contributions of this article can be summarized as follows.

- We explore and interpret the prior knowledge implied in the hand skeleton from a novel perspective inspired by cognitive psychology and propose to estimate 3D hand poses by fusing the 2D pose and texture features of the hand. We divide the prior knowledge into two categories; the first one is learned from data and the second can be embedded in loss functions.
- We extend the canonical 21-joint hand model to a 128-joint one and use corresponding denser 2D heatmaps as an approximately mask-aware spatial attention mechanism to help extract texture features.
- We utilize hand meshes as a higher level of supervision to optimize feature extraction without paying a price during network inference.

2. Related work

2.1. Hand pose and shape estimation from RGB images

Deep neural networks, being able to learn representations from data, are widely used for hand pose estimation [10, 12, 13, 14, 15]. Zimmermann *et al.* [10] propose a network to segment hand, extract 2D joint positions, and derive 3D hand poses, respectively. Spurr *et al.* [12] construct a unified latent space using multiple modalities to encourage similar poses with different modalities to be embedded close to each other. Iqbal *et al.* [13] propose 2.5D heatmaps for each key point for depth prediction. Intuitively, a straightforward way to mitigate depth ambiguity is using images taken from multiple views [4, 16]. In addition to pose annotations, Cai *et al.* [17] employ depth images as a weak supervision. As the hand skeleton is of a graph-like structure, some works [18, 19] construct the hand skeleton as an undirected graph.

Most hand shape estimation methods are based on MANO [8], a parameterized hand model. Boukhayma *et al.* [20] present an end-to-end method for hand shape and pose estimation from single RGB images in the wild. Zhang *et al.* [5] use a multi-task learning framework to estimate the 2D/3D hand pose, hand mask, and MANO hand mesh. More recently, personalized hand shape reconstruction from a single RGB image [21] or a short RGB video [22] is proposed to incorporate identity information. More challenging scenarios related to hand-object [23] and two-hand [24] interaction are attracting considerable attention. Different from the methods based on MANO, Ge *et al.* [25] propose an end-to-end trainable hand mesh generation approach using Graph CNN. These methods generally obtain hand poses as by-products of their shapes. In contrast, we employ hand shapes as a higher level of supervision for pose estimation.

2.2. Hand shape and pose estimation using geometric and biomechanical constraints

To produce biomechanically feasible hand poses, some works [6, 7, 26] adopt geometric or biomechanical constraints. Zhang *et al.* [6] introduce two geometric rules to restrict the joint position and bending direction of the finger. In [26], priors consist of the lower bound of the variational auto-encoder (VAE) as well as the bone lengths and self-occlusions of the hand. Spurr *et al.* [7] adopt palmar structure, bone lengths, and joint angles as biomechanical

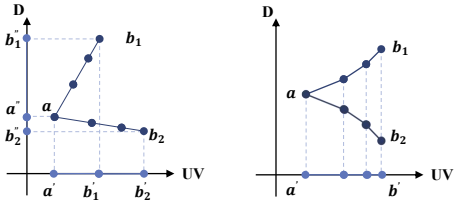


Figure 2: Illustration of the prior knowledge implied in the articulated structure of the hand. For the same finger, a smaller length in the image plane UV generally means a larger depth margin between joints (left). Meanwhile, there exist at least two 3D poses corresponding to the same 2D pose (right).

constraints. Both [26] and [7] use predefined ranges to restrict the bone's length; however, it is hard to accurately determine these ranges due to the diversity of hand bones. In contrast, we simply assume that the bone near the wrist is longer than the bone near the tip, giving the network more room to learn from data and fine-tune itself.

3. Methodology

3.1. Prior knowledge

The hand skeleton forms an articulated structure, the priors of which could provide opportunities for us to improve evaluation results. Fig. 2 presents a visualization of the prior geometric knowledge implied in the hand skeleton. Taking the bone of a stretched finger (denoted as vector ab) as an example, we can infer that a shorter projection of the finger on the image plane (UV) usually means a longer projection on the depth axis. Specifically, in the left of Fig. 2, the finger ab is presented in two different orientations, denoted as ab_1 and ab_2, i.e., $|ab_1| = |ab_2|$. $a'b_1'$ and $a'b_2'$ are their projections on the image plane and $a''b_1''$ and $a''b_2''$ are their projections on the depth axis. It can be observed that $|a'b_1'| < |a'b_2'|$ and $|a''b_1''| > |a''b_2''|$. From this, we can further get a rule that if $|ab_1| = |ab_2|$, then

$$|a'b_1'| > |a'b_2'| \Leftrightarrow |a''b_1''| < |a''b_2''|, \tag{1}$$

and

$$|a'b_1'| < |a'b_2'| \Leftrightarrow |a''b_1''| > |a''b_2''|. \tag{2}$$

This rule suggests that it is possible to infer relative depth between hand joints from their 2D positions. Considering there are some differences between the hand skeletons of individuals, it is necessary to learn a mapping between 2D coordinates and relative depth of hand joints from data. Unfortunately, however, at least two 3D poses can project to the same 2D pose. As shown in the right of Fig. 2, two poses (denoted as ab_1 and ab_2) formed by a finger bent towards opposite directions have the same projection ($a'b'$) on the image plane, which means we are unable to determine whether a joint is in front or back of its adjacent joints. Hence, additional information is needed to deal with this problem. Inspired by the human cognitive psychology, it can be realized that the two 3D poses that correspond to the same 2D pose can probably make the hand render different textures and show various effects with illuminations on the skin. These differences could make human beings able to distinguish between the two 3D poses. Based on these observations, we propose to extract 2D poses and texture features of the hand separately and combine them together to infer 3D poses.

On the other hand, geometric constraints on the joint position and the bending direction of an individual finger and between multiple fingers could be beneficial to produce biomechanically feasible poses. These constraints can be applied to all ordinary persons. As a result, the prior knowledge can be categorized into two aspects; the first one is implicitly implied in the hand skeleton and can be learned from data, while the second one can explicitly be defined as loss functions. To integrate these two types of prior knowledge appropriately, we use the deep neural network to learn a mapping f_p for implicit priors to regress relative depth d between hand joints from their 2D coordinates $\boldsymbol{J}^{2D}$. Taking into account hand texture features F_t as complementary information to 2D poses, we totally have

$$d = f_p(\boldsymbol{J}^{2D}; F_t), \tag{3}$$

where ";" is the concatenation operation.

3.2. Mesh supervision

MANO [8] is a popular parameterized hand model. It is mathematically defined as

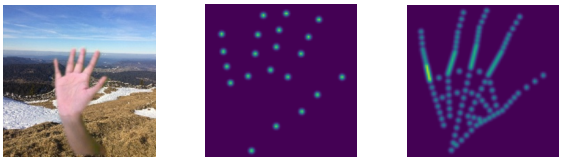
$$M(\beta, \theta) = W(T_P(\beta, \theta), J(\beta), \theta, \mathcal{W}), \tag{4}$$

$$T_P(\beta, \theta) = \overline{T} + B_S(\beta) + B_P(\theta), \tag{5}$$

where W is a skinning function applied to hand mesh with shape T_P, joint locations J defining a kinematic tree, pose $\theta \in \mathbb{R}^{45}$, shape $\beta \in \mathbb{R}^{10}$, and blend weights $\mathcal{W}$; $\overline{T}$ is the template mesh, $B_S(\beta)$ and $B_P(\theta)$ represent offsets from the template. To obtain MANO hand meshes, the shape β and pose θ parameters should be provided. Different from hand poses consisting of joint positions, the pose parameter of MANO is represented by the rotation angle between adjacent joints.

As a parameterized hand model, MANO can always produce feasible hand shapes and poses. However, when under difficult conditions, MANO will probably have to sacrifice accuracy for feasibility. Therefore, we propose to estimate MANO meshes and hand poses in parallel, which could encourage the network to learn representations for feasible poses, and thus complement direct pose estimation to achieve a better trade-off between accuracy and feasibility.

In this work, we adopt the hand mesh generated by a differentiable MANO layer [27] and mesh-related loss functions as a regularizer to optimize feature extraction for pose estimation. The pose and shape parameters of MANO are learned from the combination of 2D poses and texture features of the hand, respectively. The MANO layer is not needed when the network is put into practice so that the time and space complexities of it can be reduced.



(a) (b) (c)

Figure 3: Illustrations of a hand image (a) and its canonical 21-joint heatmap (b) and interpolated 128-joint heatmap (c).

3.3. Extended 128-joint hand model

As described earlier, we extract 2D poses and texture features of the hand separately and use the latter to complement the former to estimate 3D poses. In turn, we also propose to use hand joint heatmaps as an intuitive spatial attention mechanism to help extract hand texture features of interest.

In [15], the 21-joint heatmaps followed by convolutional and pooling layers are used as spatial attention to separate the hands for two-hand pose estimation. However, 21-joint heatmaps can only provide relatively sparse pose-aware attention. In contrast, heatmap attention is integrated into our proposed network without post-process layers. Furthermore, we interpolate the canonical 21-joint hand model to a 128-joint one (see Fig. 3), yielding an extended heatmap, which can provide approximately mask-aware attention without hand shape or mask annotations. It is worth noting that the interpolation can be performed automatically by a computer program (see Algorithm 1) and the network's complexity only increases slightly (less than 1% in model size).

We sum 128 joint heatmaps to get a single heatmap and sum texture feature maps to get a single feature map, respectively. These two maps are multiplied to get a new feature map. This process can be formulated as

$$F_{new} = \Sigma_{j=1}^{J} H_j \otimes \Sigma_{i=1}^{N} F_i, \tag{6}$$

where J and N are the numbers of hand joints and texture feature maps, H_j and F_i are heat map and texture feature map, respectively, and $\otimes$ denotes element-wise multiplication.

3.4. Network architecture

The architecture of our proposed deep neural network is shown in Fig. 1. It consists of (i) a two-stacked hourglass network as the backbone to produce hand joint heatmaps and texture feature maps, (ii) a multilayer perceptron (MLP) that regresses relative depth between hand joints from 2D joint coordinates and hand texture features, (iii) an MLP that generates MANO pose parameters from 2D joint coordinates and hand texture features, (iv) an MLP that produces MANO shape parameters from 2D joint coordinates and hand texture features, and (v) a differentiable MANO layer that generates hand meshes from pose and shape parameters.

Given an input image $I \in \mathbb{R}^{256\times256\times3}$, the stacked hour glass network [28] f_{SH} extract features $F = (F_{hm}; F_t) = f_{SH}(I)$, where $F_{hm} \in \mathbb{R}^{64\times64\times128}$ are the interpolated 128-joint heatmaps and $F_t \in \mathbb{R}^{64\times64\times512}$ are hand texture features, respectively. Heatmaps are then summed to form a single heatmap. Subsequently, the canonical 21-joint heatmaps are extracted from the 128-joint heatmaps, and flattened to a heatmap vector $\boldsymbol{v}_{hm} \in \mathbb{R}^{4096}$. Hand texture features are first multiplied by the 128-joint heatmap for spatial attention and converted to a texture vector $\boldsymbol{v}_t \in \mathbb{R}^{4096}$. An MLP f_{3D}, which consists of four fully-connected layers, takes the concatenation of $\boldsymbol{v}_{hm}$ and $\boldsymbol{v}_t$ as input and outputs 3D coordinates of hand joints $\boldsymbol{v}_{3D} = f_{3D}(\boldsymbol{v}_{hm}, \boldsymbol{v}_t)$, where $\boldsymbol{v}_{3D} \in \mathbb{R}^{63}$ is the x, y, and z coordinates. Meanwhile, 2D positions of hand joints in the image plane are obtained from heatmaps using $(u, v) = \text{argmax}_{(u,v)} H(u, v)$, where H denotes the 21-joint heatmap, and u, v is the pixel position in H. We here use (u, v) rather than (x, y) to integrate with z (also relative depth d) to generate final 3D hand joint positions. Accordingly, prior biomechanical constraints will be imposed on (u, v, d). The reason that, in addition to (u, v), we evaluate (x, y) simultaneously is to preserve the proportion of (x, y) to z and then replace (x, y) with (u, v) proportionally in scenarios where the scale of the hand is unknown. We can recover the scale of depth d by

$$S_d = S_{uv} S_z / S_{xy}, \tag{7}$$

where S_d, S_{uv}, S_{xy}, and S_z are the scales of d, uv, xy, and z, respectively.

The premise of using the MANO layer to generate hand meshes is obtaining the pose ($\theta \in \mathbb{R}^{45}$) and shape ($\beta \in \mathbb{R}^{10}$)

Algorithm 1 Interpolate joint positions

Input: uv /* 21 joint positions */
Output: uv_ext /* 128 joint positions */
Output: $joint\_indices$ /* new indices of original joints in the 128-joint model */

```
arr_n ← [4, 5, 3, 2, 6, 7, 3, 2, 10, 7, 3, 2, 6, 7, 3, 2, 6, 7, 3, 2]
/* Interpolate positions within the same finger */
for i ← 1 to len(arr_n) do
    n ← arr_n[i]
    p = FindParentJoint(i)
    /* Insert n new joints between joints i and p */
    InsertJoint(i, p, n)
end for
/* Get new indices of all five MCP joints */
mi = GetNewMCPIndices()
/* Interpolate positions between fingers in the palm */
src_dest_n ← [[mi[0], mi[1], 4], [mi[1], mi[2], 1],
      [mi[2], mi[3], 1], [mi[3], mi[4], 1],
      [mi[0] − 1, mi[1] − 3, 3], [mi[0] − 3, mi[1] − 6, 2],
      [mi[0] − 6, mi[1] − 6, 1], [mi[1] − 3, mi[2] − 3, 1],
      [mi[2] − 3, mi[3] − 3, 1], [mi[3] − 3, mi[4] − 3, 1]]
for i ← 1 to len(src_dest_n) do
    s, d, n ← src_dest_n[i]
    /* Insert n new joints between joints s and d */
    InsertJoint(s, d, n)
end for
```

parameters of MANO. We use an MLP f_θ consisting of four fully-connected layers to generate pose parameters from the concatenation of the 2D pose and texture features of the hand, i.e., $\theta = f_\theta(v_{hm}, v_t)$. Similarly, another MLP f_β containing four fully-connected layers is employed to extract shape parameters from the hand's 2D pose and texture features, i.e., $\beta = f_\beta(v_{hm}, v_t)$. A differentiable MANO layer f_M is then utilized to produce hand meshes $\mathcal{M} \in \mathbb{R}^{N\times 3}$ from the pose and shape parameters, i.e., $\mathcal{M} = f_M(\beta, \theta)$, where N is the number of vertices in the hand mesh and $N = 778$ for the MANO model.

3.5. Loss functions

We adopt a set of loss functions to supervise different parts of the network (see Fig. 1). They can be separated into (i) pose-related, (ii) prior-related, and (iii) mesh-related groups.

Pose-related loss: This part consists of 2D heatmap loss and 3D pose loss. The ground truth heatmap is defined as a 2D Gaussian with a standard deviation of 1 px centered on the ground truth 2D joint location. 2D heatmap loss is defined as:

$$\mathcal{L}_{hm} = \Sigma_{j=1}^{J} \| \hat{H}_j - H_j \|_2^2, \tag{8}$$

where $\hat{H}_j \in \mathbb{R}^{64\times 64}$ and $H_j \in \mathbb{R}^{64\times 64}$ denote the estimated and ground-truth heatmaps, respectively. This loss is applied to the canonical 21-joint and the extended 128-joint heatmaps. 3D pose loss is defined as:

$$\mathcal{L}_{jt} = \Sigma_{j=1}^{J} \| \hat{p}_j^{3D} - p_j^{3D} \|_2^2, \tag{9}$$

where $\hat{p}_j^{3D} \in \mathbb{R}^3$ and $p_j^{3D} \in \mathbb{R}^3$ are the estimated and ground-truth 3D joint positions, respectively.

Prior-related loss: Prior geometric constraints are introduced to encourage the network to produce as feasible poses as possible. We denote the wrist and a finger's joints (from the MCP joint to the tip) as r, a, b, c, and d, respectively, and divide them into three segments: (r, a, b), (a, b, c), and (b, c, d). First, these five joints should be approximately in the same plane. Accordingly, the loss function is formulated as

$$\mathcal{L}_p = \langle ra \times ab, bc \rangle + \langle ab \times bc, cd \rangle, \tag{10}$$

where ra, ab, bc, and cd represent the vector from joint r to a, from a to b, from b to c, and from c to d, respectively; "$\langle\rangle$" and "$\times$" are inner and cross product operators of two vectors.

Second, the three segments should bend in the same direction. We at first define the bending degree of a finger's segment, for example, (a, b, c), as

$$D_{abc} = (|ab| + |bc|)/|ac|. \tag{11}$$

The corresponding loss function of segment, e.g. (a, b, c) can be defined as

$$\mathcal{L}_d = \begin{cases} D_{abc} - 1 & \langle ra \times ab, ab \times bc \rangle < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Third, the length of a finger's bones should follow the rule that the bone near the wrist is longer than the one near the tip. The loss function is defined as

$$\begin{cases} \mathcal{L}_{len} = \frac{1}{2}[(|\delta_{rab}| - \delta_{rab}) + (|\delta_{abc}| - \delta_{abc}) \\ \qquad + (|\delta_{bcd}| - \delta_{bcd})], \\ \delta_{rab} = |ra| - |ab|, \\ \delta_{abc} = |ab| - |bc|, \\ \delta_{bcd} = |bc| - |cd|. \end{cases} \tag{13}$$

As a result, the prior loss in total is

$$\mathcal{L}_{prior} = \mathcal{L}_p + \mathcal{L}_d + \mathcal{L}_{len}. \tag{14}$$

Mesh-related loss: In works [29, 25] that adopt a non-MANO hand model, the mesh loss consists of vertex loss $\mathcal{L}_v$, normal loss $\mathcal{L}_n$, and edge loss $\mathcal{L}_e$, used to restrict 3D positions of mesh vertices, ensure surface normal consistency, and keep edge length, respectively. In this work, since we adopt the MANO model, which is self-consistent between vertices, surfaces, and edges, vertex loss is sufficient to ensure hand mesh consistency. Therefore, the mesh loss is defined as

$$\mathcal{L}_{mesh} = \mathcal{L}_v. \tag{15}$$

The overall loss function of our method is

$$\mathcal{L}_{total} = \lambda_{hm}\mathcal{L}_{hm} + \lambda_{jt}\mathcal{L}_{jt} + \lambda_{prior}\mathcal{L}_{prior} + \lambda_{mesh}\mathcal{L}_{mesh}, \tag{16}$$

where λ_{hm}, λ_{jt}, λ_{prior}, and λ_{mesh} are hyperparameters. In our experiment, we set $\lambda_{hm} = 5$, $\lambda_{jt} = 1$, $\lambda_{prior} = 0.1$, and $\lambda_{mesh} = 0.1$, respectively.

3.6. Implementation

Dataset: The Rendered Hand Dataset (RHD) [10] and the Stereo Hand Pose Tracking Benchmark (STB) [9] are two widely used datasets in 3D hand pose estimation. The former is a synthetic dataset while the latter is a real one. These two datasets both provide 3D hand pose annotations. InterHand2.6M [30] and FreiHAND [11] further offer hand shape annotations. InterHand2.6M [30] is a large-scale real RGB-based 3D hand pose and shape dataset, including both single and interacting hand sequences under various poses from multiple subjects. FreiHAND [11] additionally contains hand-object interaction scenarios.

For the STB dataset, we select images with background 3 as evaluation data and images with other backgrounds as training data. The RHD dataset is split into training and evaluation parts according to [10]. Although the original FreiHAND dataset contains a sub-dataset for evaluation, it lacks pose and mesh annotations. Therefore, we divide the original training dataset into four parts, each including 32560 images, and take the second part as real train data, the first 3000 images of the third part as real evaluation data. All hand images are cropped and resized to 256×256 pixels.

Data preprocess: Similar to [10], we estimate a scale-invariant and root-relative 3D hand pose. We select the wrist

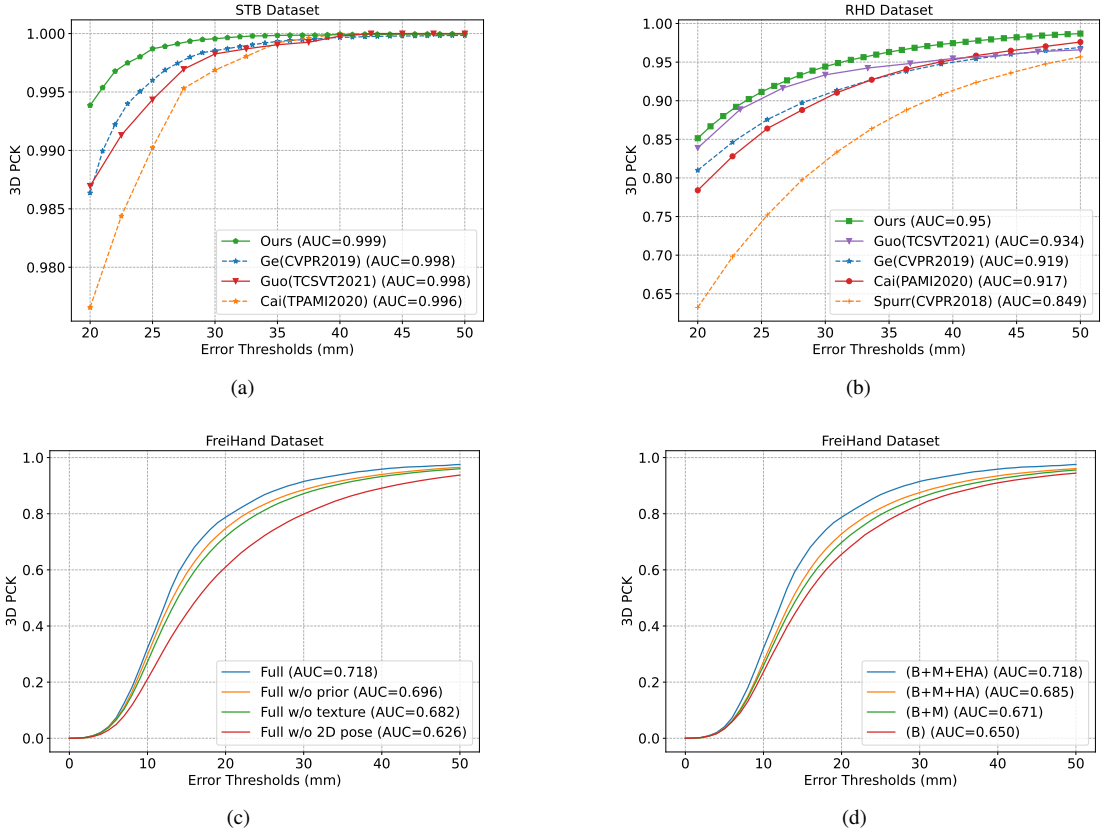


Figure 4: (1) Comparisons with other methods on the STB (a) and RHD (b) datasets. (2) Ablation study results on the FreiHand dataset (c)(d).

r as the root joint and the distance between the wrist and the MCP of the middle finger as the scale s of the hand. The relative and normalized 3D coordinates of the joints are given by

$$\widetilde{J}_i = (J_i - J_r)/s, \tag{17}$$

where $J_i \in \mathbb{R}^3$ is the original coordinates of hand joints. The mesh is also normalized in a similar way. We select vertices with the index of 33 and 370 as the wrist and the MCP of the middle finger, respectively. The relative and normalized 3D coordinates of mesh vertices are given by

$$\widetilde{\boldsymbol{v}}_i = (\boldsymbol{v}_i - \boldsymbol{v}_r)/s, \tag{18}$$

where $\boldsymbol{v}_i \in \mathbb{R}^3$ is the original coordinates of mesh vertices.

Data augmentation: For images in all training datasets, we perform scaling (0.9 - 1.1) and rotation with an angle in $\{0°, 90°, 180°, 270°\}$. The 2D and 3D coordinates of hand joints and mesh vertices are also rotated accordingly. Moreover, we randomly change an image by color-jittering with the following configurations: brightness (0.9 - 1.1), contrast (0.85 - 1.15), saturation (0.9 - 1.1), hue (0.9 - 1.1), and apply randomly chosen Gaussian blur on images.

Training: Our hand pose estimation method is implemented using the PyTorch framework. The network is trained using Adam optimizer with a batch size of 16 on a single GTX2080Ti GPU. The learning rate warmup strategy [31] is used in the first epoch to stabilize the training process. To train the neural network efficiently, we divide it into three main parts and train it in a gradually extended manner. At first, we train the two-stacked hourglass subnetwork using the heatmap loss with a learning rate of 0.001. We then include the 3D coordinate regressor using the 3D pose and prior-related loss with a learning rate of 0.0003. Finally, we train the whole network using all loss items with a learning rate of 0.0001.

4. Experiments

We evaluate our method on the STB, RHD, and FreiHAND datasets as well as a custom hand gesture dataset. Similar to [25], we report evaluation results with the following metrics: (i) 3D PCK: the percentage of correct key points of which the Euclidean error distance is below a threshold; (ii) AUC: the area under the 3D PCK curve; and (iii) EPE: the end position error(mm) between the predicted and ground-truth 3D hand pose after root joint alignment.

4.1. Comparison with other methods

First, we compare our method with methods [25, 18, 17] on the STB dataset. In work [25], the network is trained for hand shape reconstruction supervised by non-parameterized

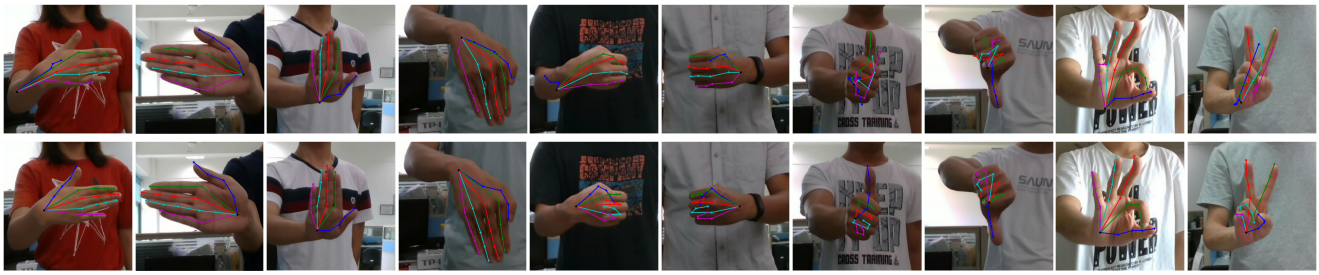


Figure 5: Qualitative comparison on a custom hand gesture dataset between method [25] (top) and ours (bottom). The thumb, index, middle, ring, and little fingers should be colored blue, green, red, cyan, and magenta, respectively.

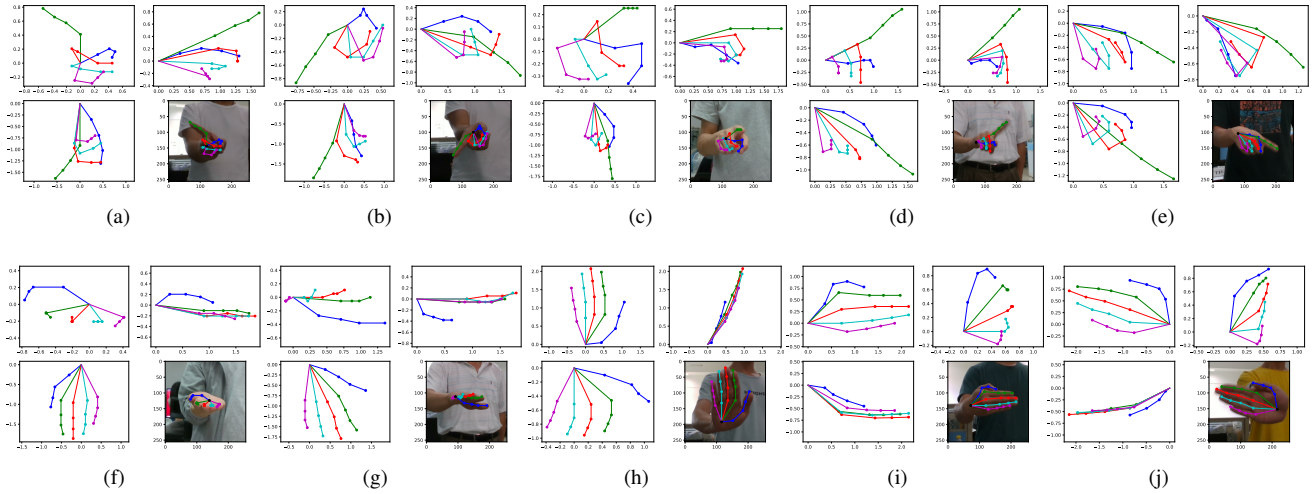


Figure 6: 3D hand pose estimation results of the pointing direction of the index finger (first row) and the orientation of the hand (second row). There are four images in each sub-figure; the bottom-right one displays the hand image with 2D pose annotation; the top-left, top-right, and bottom-left ones show the front-view, left-view, and top-view of the 3D hand pose, respectively. The thumb, index, middle, ring, and little fingers are colored blue, green, red, cyan, and magenta, respectively.

hand meshes, and fine-tuned on the STB dataset. Accordingly, we train our network on the InterHand dataset with mesh supervision followed by fine-tuning with the STB dataset without mesh supervision since it does not provide mesh annotations. Results are shown in Fig. 4(a).

Second, we compare our method with methods [25, 12, 18, 17] on the RHD dataset. Similarly, our network is trained on the InterHand dataset with mesh supervision and fine-tuned with the RHD dataset without mesh supervision since it does not provide mesh annotations. Results are shown in Fig. 4(b).

The cross-dataset generalization ability is more important for a method when put into practice. We qualitatively compare our method with method [25] in 2D hand pose estimation on a hand image dataset originally built for hand gesture classification. Some results are shown in Fig. 5. It can be seen that our method is superior to [25] by a considerable margin, especially in dealing with severe self-occlusion of hand joints, varied backgrounds, and different illuminations.

We also compare the EPE between method [25] and ours, shown in Table 2. The first and second rows are the results obtained with fine-tuning. As method [25] does not publish a model fine-tuned on the RHD dataset, we here omit the result. The third row is the EPE of cross-dataset test on the FreiHand dataset. These results indicate that our method is superior to method [25] in both inner- and cross-dataset test.

We demonstrate the 3D hand pose estimation results of our network, which is trained on the InterHand and FreiHAND datasets with mesh supervision, from three main views in Fig. 6. It indicates that our method can infer relative depth and produce biomechanically feasible 3D hand poses even under severe self-occlusion (Fig. 6(c), 6(f), 6(g)) and blur (Fig. 6(d), 6(e)).

Lastly, we compare the time and space complexities of our method with different configurations and method [25], shown in Table 1. It manifests that our proposed network is more efficient than [25] in terms of storage space, parameter count, and FLOPs at the inference stage.

4.2. Ablation study

In this work, we propose a framework that integrates 2D poses and texture features of the hand to infer 3D poses. To further improve the performance of the framework, we (i)

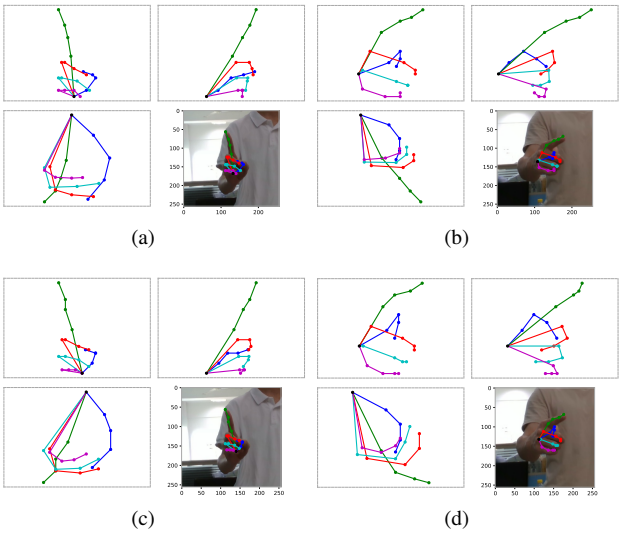


Figure 7: Illustrations of 3D hand poses estimated with (upper) and without (bottom) prior-related losses. It demonstrates the effectiveness of prior geometric constraints, especially for a single stretched finger.

Table 1
Comparisons of time and space complexities of models at the training and inference stages

Model	Size (MB)	Params (M)	FLOPs (G)
Ge et al.[25]	87.6	21.76	16.34
Ours (B)	85	17.34	13.69
Ours (B+M)	85	17.34	13.69
Ours (B+M+HA)	85	17.34	13.69
Ours (B+M+EHA)	85.3	17.42	14.03
Ours (B+M+EHA)*	120.5	25.84	14.07

B: Baseline; M: Mesh Supervision; HA: Heatmap Attention; EHA: Extended Heatmap Attention. The model with superscript * is in the training stage.

Table 2
Comparisons of EPE

Dataset	Ge et al. [25]	Ours
STB (fine-tuned)	14.04	5.30
RHD (fine-tuned)	–	14.16
FreiHand	121.80	72.25

utilize hand joint heatmaps as a spatial attention mechanism for hand texture feature extraction, (ii) interpolate the canonical 21-joint hand model to a 128-joint one to get denser heatmaps, and (iii) employ hand meshes as supervision to optimize feature extraction.

To demonstrate the contributions of these strategies, we evaluated our network with different configurations. For clarity, we denote the simplest model as the baseline (B), baseline with mesh supervision as "B+M", baseline with mesh supervision and 21-joint heatmap attention as "B+M+HA", and baseline with mesh supervision and extended 128-joint heatmap attention as "B+M+EHA" which is also denoted as the "Full" model. The network is trained on the InterHand and FreiHAND datasets since they both provide mesh annotations.

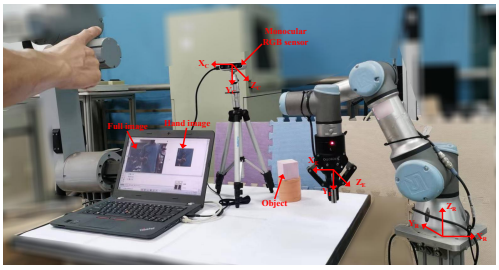


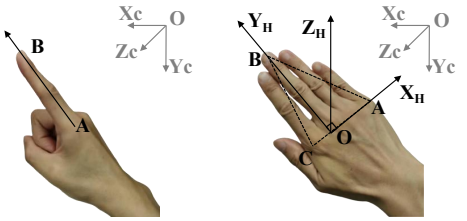
Figure 8: A scene of operating the robot using the pointing direction of the index finger. The operator's hand image is captured by a monocular RGB camera and cropped from the center for 3D hand pose estimation. Three coordinate systems are attached to the robotic base, EE, and the camera, respectively.

First, we evaluate the network's performance without using 2D poses and without using texture features of the hand, respectively. Results are shown in Fig. 4(c), indicating that (i) only using 2D hand poses can infer relatively accurate 3D poses and (ii) using hand texture features alone leads to a considerable drop in performance.

The effects of mesh supervision, hand joint heatmap attention and extended hand joint heatmap attention are shown in Fig. 4(d). It can be seen that (i) mesh supervision improves pose estimation since hand meshes can provide a higher magnitude of information, (ii) hand joint heatmap attention contributes to performance gains as joint position regression and feature map extraction can benefit each other, and (iii) extended 128-joint heatmap further enhances pose estimation considerably because it can provide approximately mask-aware attention. It can also be observed from Table 1 that (i) the 21-joint heatmap attention does not increase the network's complexity, (ii) the extended 128-joint heatmap attention only increases the network's complexity slightly, and (iii) the removable mesh supervision only increases the model's complexity at the training stage.

We also analyze the effect of prior-related loss items both quantitatively and qualitatively, shown in Fig. 4(c) and Fig. 7, respectively. It can be seen that prior geometric constraints are beneficial to generating more accurate and feasible hand poses.

As for mesh supervision, we tried three types of network architecture: (i) estimate 3D hand poses and shapes in cascade, (ii) extract the pose and shape parameters of MANO by the same MLP in parallel with pose estimation, and (iii) extract the parameters by two MLPs, respectively, in parallel with pose estimation. We find the second one is better than the first one and the last one is the best.



(a) (b)

Figure 9: Illustration of the pointing direction of the index finger (a) and the orientation of the hand (b) in the camera coordinate system.

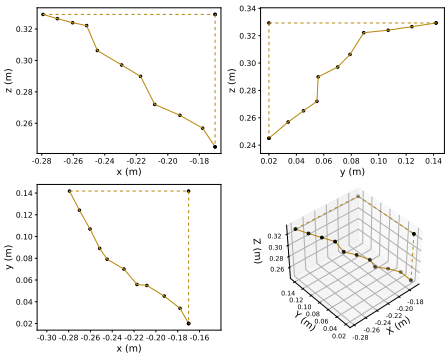


Figure 10: Comparison of the robotic EE's 3D trajectories generated by using the pointing direction of the index finger (solid line) with that produced by using the teaching board (dotted line).

5. Robot Teleoperation

A typical application of hand poses for robot teleoperation is mapping them to the robotic grasper to grasp objects [32, 33]. In this work, we extend it to control the moving direction and orientation of the robotic EE using the pointing direction of the index finger (see Fig. 8) and the orientation of the hand, respectively.

To extract the index finger's pointing direction in the camera coordinate system ($X_C Y_C Z_C$), we attach a vector AB on it, as shown in Fig. 9(a); point A represents the MCP joint of the finger, and point B the tip. We denote the coordinate of point A as $P_A = (x_A, y_A, z_A)$, B as $P_B = (x_B, y_B, z_B)$, and the unit vector parallel with AB as $s = AB/|AB|$.

Fig. 10 displays the robotic EE's trajectories produced by using the index finger (solid line) and that generated by using the teaching board (dotted line). It is obvious that using the index finger is not only more natural but can also reduce the moving distance (0.202m vs. 0.315m) of the robotic EE.

To calculate the hand's orientation, we attach a coordinate system $X_H Y_H Z_H$ to the hand based on the little finger's MCP joint (denoted as A), the middle finger's TIP joint (B), and the index finger's MCP joint (C) (see Fig. 9(b)). Subsequently, the roll angle α can be calculated by projecting vector Z_H onto plane $X_C O Y_C$, getting vector $Z_H^{'}$, and measuring the angle between Y_C and $Z_H^{'}$. Similarly, the pitch angle β can be obtained by projecting Z_H onto $Y_C O Z_C$, getting $Z_H^{''}$, and measuring the angle between Y_C and $Z_H^{''}$; the yaw angle γ can be obtained by projecting Y_H onto $X_C O Z_C$, getting $Y_H^{'}$, and measuring the angle between Z_C and $Y_H^{'}$.

To map an orientation from the hand to the robotic EE, we first define the hand's default orientation as the one when the three principal axes of $X_H Y_H Z_H$ are parallel with that of $X_C Y_C Z_C$ (e.g., $X_H = -X_C$, $Y_H = -Z_C$, and $Z_H = -Y_C$). Similarly, the robotic EE's default orientation is defined as the one when the three principal axes of $X_E Y_E Z_E$ are parallel with that of $X_R Y_R Z_R$ (e.g., $X_E = -X_R$, $Y_E = -Z_R$, and $Z_E = -Y_R$). Subsequently, orientation mapping can be formulated as

$$P_c^R = P_d^R + (P_c^H - P_d^H), \tag{19}$$

where superscript R and H represent the robot and the hand, respectively; subscript c and d indicate the current and default orientations, respectively.

6. Conclusion

This paper proposes a Prior-knowledge Aware and Mesh-Supervised Network (PAMSNet) to deal with depth ambiguity and improve feasibility for 3D hand pose estimation from single RGB images. We find fusing 2D poses and texture features of the hand can obtain competitive results with other methods, with better interpretability from a novel perspective inspired by cognitive psychology. Mesh supervision in parallel can complement direct pose estimation without paying a price at the inference stage. Extending the canonical 21-joint hand skeleton to a 128-joint one provides approximately mask-aware spatial attention to extract features of interest at little cost. Prior-related loss items are beneficial to generating more accurate and feasible hand poses. In total, PAMSNet is comparable to or outperforms other networks in terms of accuracy and generalization ability with an efficient architecture.

References

[1] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, J. Yuan, A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 793–802.

[2] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S. A. Ali, V. Golyanik, C. Theobalt, D. Stricker, Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7113–7122.

[3] M. Rezaei, R. Rastgoo, V. Athitsos, Trihorn-net: A model for accurate depth-based 3d hand pose estimation, Expert Systems with Applications 223 (2023) 119922.

[4] T. Simon, H. Joo, I. Matthews, Y. Sheikh, Hand keypoint detection in single images using multiview bootstrapping, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 1145–1153.

[5] X. Zhang, H. Huang, J. Tan, H. Xu, C. Yang, G. Peng, L. Wang, J. Liu, Hand image understanding via deep multi-task learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11281–11292.

[6] X. Zhang, Q. Li, H. Mo, W. Zhang, W. Zheng, End-to-end hand mesh recovery from a monocular rgb image, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2354–2364.

[7] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, J. Kautz, Weakly supervised 3d hand pose estimation via biomechanical constraints, in: European Conference on Computer Vision, Springer, 2020, pp. 211–228.

[8] J. Romero, D. Tzionas, M. J. Black, Embodied hands: Modeling and capturing hands and bodies together, arXiv preprint arXiv:2201.02610 (2022).

[9] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, Q. Yang, A hand pose tracking benchmark from stereo matching, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 982–986.

[10] C. Zimmermann, T. Brox, Learning to estimate 3d hand pose from single rgb images, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 4903–4911.

[11] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, T. Brox, Freihand: A dataset for markerless capture of hand pose and shape from single rgb images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 813–822.

[12] A. Spurr, J. Song, S. Park, O. Hilliges, Cross-modal deep variational hand pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 89–98.

[13] U. Iqbal, P. Molchanov, T. B. J. Gall, J. Kautz, Hand pose estimation via latent 2.5 d heatmap regression, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 118–134.

[14] Z. Zhao, X. Zhao, Y. Wang, Travelnet: Self-supervised physically plausible hand motion learning from monocular color images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11666–11676.

[15] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, H. Wang, Interacting two-hand 3d pose and shape reconstruction from single color image, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11354–11363.

[16] S. Sridhar, A. Oulasvirta, C. Theobalt, Interactive markerless articulated hand motion tracking using rgb and depth data, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 2456–2463.

[17] Y. Cai, L. Ge, J. Cai, N. M. Thalmann, J. Yuan, 3d hand pose estimation using synthetic data and weakly labeled rgb images, IEEE transactions on pattern analysis and machine intelligence 43 (2020) 3739–3753.

[18] S. Guo, E. Rigall, L. Qi, X. Dong, H. Li, J. Dong, Graph-based cnns with self-supervised module for 3d hand pose estimation from monocular rgb, IEEE Transactions on Circuits and Systems for Video Technology 31 (2020) 1514–1525.

[19] I. Kourbane, Y. Genc, A graph-based approach for absolute 3d hand pose estimation using a single rgb image, Applied Intelligence (2022) 1–16.

[20] A. Boukhayma, R. d. Bem, P. H. Torr, 3d hand shape and pose from images in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10843–10852.

[21] D. Kong, L. Zhang, L. Chen, H. Ma, X. Yan, S. Sun, X. Liu, K. Han, X. Xie, Identity-aware hand mesh estimation and personalization from rgb images, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V, Springer, 2022, pp. 536–553.

[22] K. Karunratanakul, S. Prokudin, O. Hilliges, S. Tang, Harp: Personalized hand reconstruction from a monocular rgb video, arXiv preprint arXiv:2212.09530 (2022).

[23] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, M. Pollefeys, H2o: Two hands manipulating objects for first person interaction recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10138–10148.

[24] C. Jiang, Y. Xiao, C. Wu, M. Zhang, J. Zheng, Z. Cao, J. T. Zhou, A2j-transformer: Anchor-to-joint transformer network for 3d interacting hand pose estimation from a single rgb image, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 8846–8855.

[25] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, J. Yuan, 3d hand shape and pose estimation from a single rgb image, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10833–10842.

[26] C. Wan, T. Probst, L. V. Gool, A. Yao, Self-supervised 3d hand pose estimation through training by fitting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10853–10862.

[27] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, C. Schmid, Learning joint reconstruction of hands and manipulated objects, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 11807–11816.

[28] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: European conference on computer vision, Springer, 2016, pp. 483–499.

[29] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, Y.-G. Jiang, Pixel2mesh: Generating 3d mesh models from single rgb images, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 52–67.

[30] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, K. M. Lee, Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image, in: European Conference on Computer Vision, Springer, 2020, pp. 548–564.

[31] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, large minibatch sgd: Training imagenet in 1 hour, arXiv preprint arXiv:1706.02677 (2017).

[32] F. Gomez-Donoso, S. Orts-Escolano, M. Cazorla, Accurate and efficient 3d hand pose regression for robot hand teleoperation using a monocular rgb camera, Expert Systems with Applications 136 (2019) 327–337.

[33] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, J. Zhang, Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 416–422.