3D Hand Shape and Pose Estimation from a Single RGB Image

Liuhao Ge[1,*] Zhou Ren[2], Yuncheng Li[3], Zehao Xue[3], Yingying Wang[3], Jianfei Cai[1], Junsong Yuan[4]
[1]Nanyang Technological University [2]Wormpex AI Research
[3]Snap Inc. [4]State University of New York at Buffalo
ge0001ao@e.ntu.edu.sg, zhou.ren@bianlifeng.com, yuncheng.li@snap.com,
zehao.xue@snap.com, ywang@snap.com, asjfcai@ntu.edu.sg, jsyuan@buffalo.edu

Task

- estimating the full 3D hand shape and pose from a single RGB image

- predefine the topology of a triangle mesh representing the hand surface

A 3D mesh is represented as an undirected graph $M = (V, \mathcal{E}, W)$,
where $V = \{v_i\}_{i=1}^{N}$ is the N vertices in the mesh, $\mathcal{E} = \{e_i\}_{i=1}^{E}$ is the E edges in the mesh, and
$W = (w_{ij})_{N*N}$ is the adjacency matrix.

2 Challenges

- high dimensionality of the output space for 3D hand mesh generation.

- lack of ground truth 3D hand mesh training data for real-world images

Training: 2 stages

- On a large-scale synthetic dataset containing both ground truth 3D meshes and 3D poses; with full supervision
- 2D heat-map loss, 3D mesh loss, 3D pose loss

- On real-world datasets without 3D ground truth; fine-tune the network using weakly-supervised approach by leveraging the depth map as a weak supervision in training.

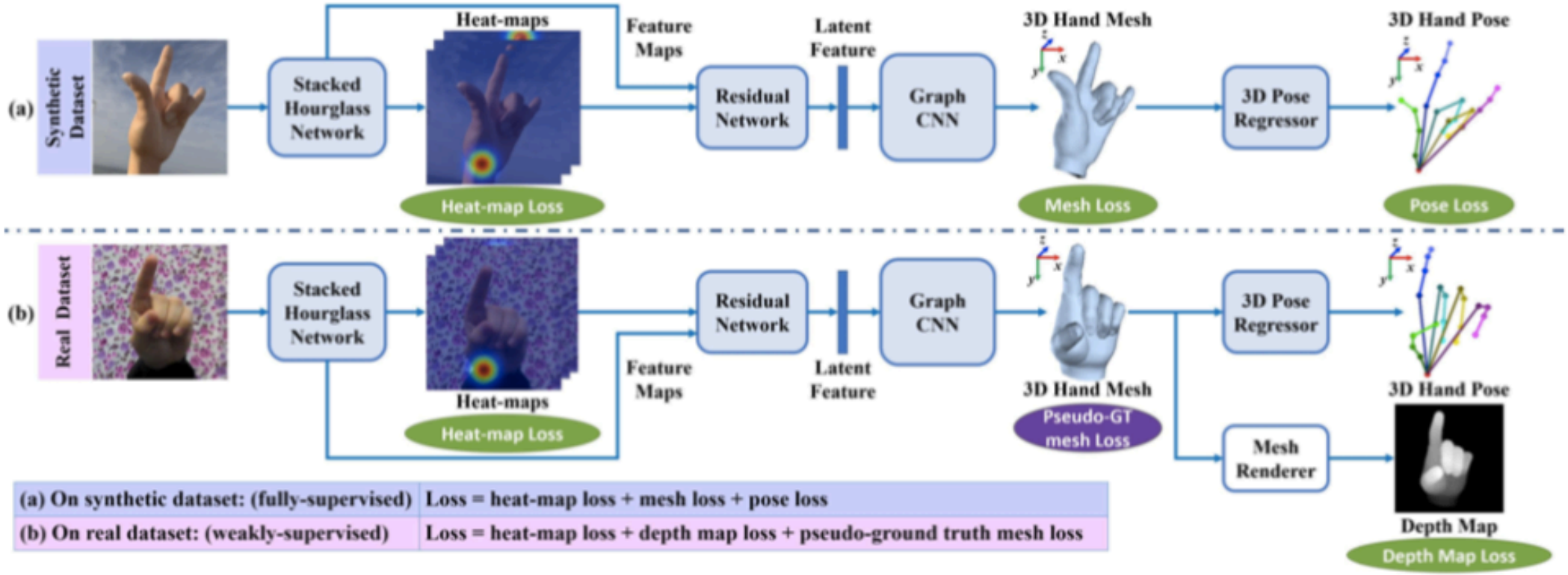


Figure 3: Overview of our method for 3D hand shape and pose estimation from a single RGB image. Our network model is first trained on a synthetic dataset in a fully supervised manner with heat-map loss, 3D mesh loss, and 3D pose loss, as shown in (a); and then fine-tuned on a real-world dataset without 3D mesh or 3D pose ground truth in a weakly-supervised manner by innovatively introducing a pseudo-ground truth mesh loss and a depth map loss, as shown in (b). For both (a) and (b), the input RGB image is first passed through a two-stacked hourglass network [34] for extracting feature maps and 2D heat-maps, which are then combined and encoded as a latent feature vector by a residual network [18]. The latent feature is fed into a Graph CNN [8] to infer the 3D coordinates of mesh vertices. Finally, the 3D hand pose is linearly regressed from the 3D hand mesh. During training on the real-world dataset, as shown in (b), the generated 3D hand mesh is rendered to a depth map to compute the depth map loss against the reference depth map. Note that this step is not involved in testing.

Graph CNN

Chebyshev Spectral Graph CNN

$$L = I_N - D^{-1/2} W D^{-1/2}$$

$$\mathbf{f}_{\text{out}} = \sum_{k=0}^{K-1} T_k\left(\tilde{L}\right) \cdot \mathbf{f}_{\text{in}} \cdot \theta_k, \qquad (1)$$

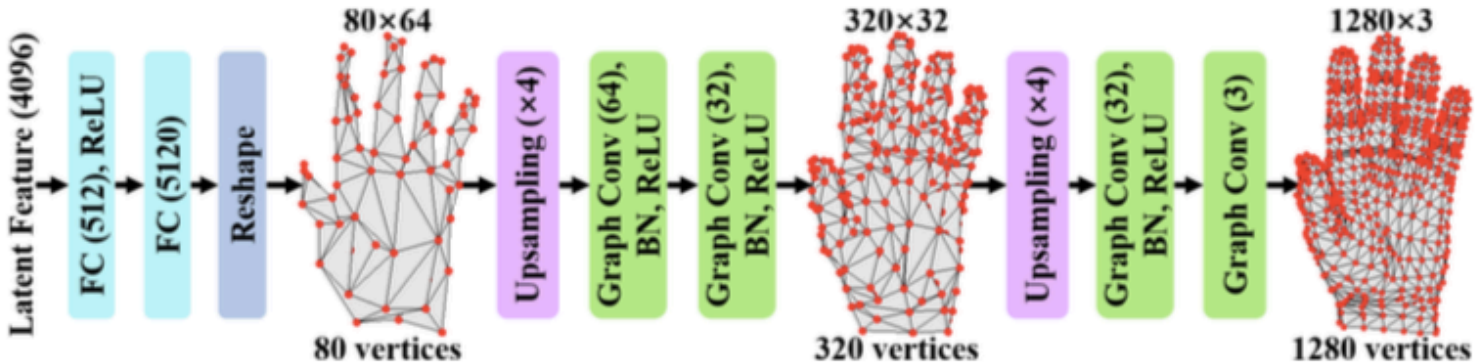


Figure 4: Architecture of the Graph CNN for mesh generation. The input is a latent feature vector extracted from the input RGB image. Passing through two fully-connected (FC) layers, the feature vector is transformed into 80 vertices with 64-dim features in a coarse graph. The features are upsampled and allocated to a finer graph. With two upsampling layers and four graph convolutional layers, the network outputs 3D coordinates of the 1280 mesh vertices. The numbers in parentheses of FC layers and graph convolutions represent the dimensions of output features.

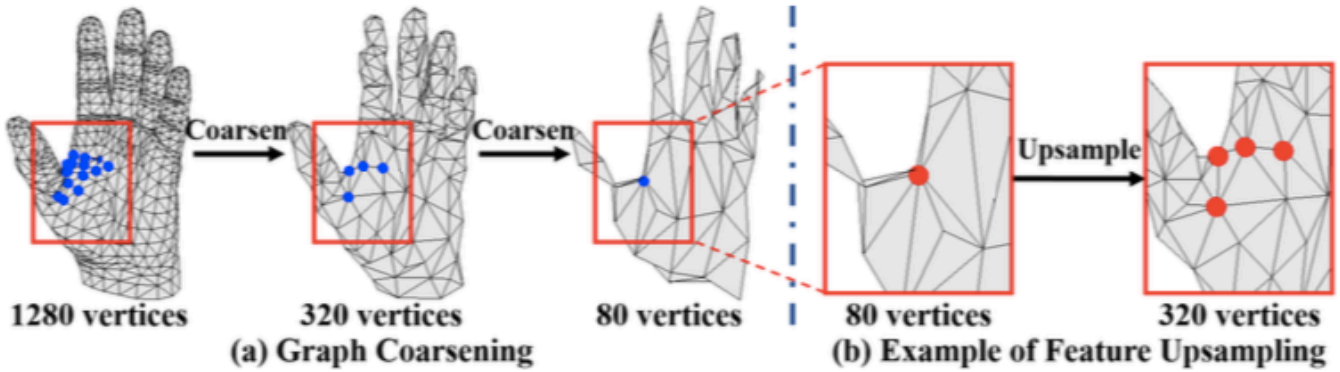


Figure 5: (a) Given our predefined mesh topology, we first perform graph coarsening [8] to cluster meaningful neighborhoods on graphs and create a tree structure to store correspondences of vertices in graphs at adjacent coarsening levels. (b) During the forward propagation, we perform feature upsampling. The feature of a vertex in the coarse graph is allocated to its children vertices in the finer graph.

1 $$\mathcal{L}_{\mathcal{H}} = \sum_{j=1}^{J} \left\| \mathcal{H}_j - \hat{\mathcal{H}}_j \right\|_2^2 \qquad \mathcal{L}_{\mathcal{J}} = \sum_{j=1}^{J} \left\| \boldsymbol{\phi}_j^{3D} - \hat{\boldsymbol{\phi}}_j^{3D} \right\|_2^2$$

Mesh Loss. Similar to [56], $\mathcal{L}_{\mathcal{M}} = \lambda_v \mathcal{L}_v + \lambda_n \mathcal{L}_n + \lambda_e \mathcal{L}_e + \lambda_l \mathcal{L}_l$ is composed of vertex loss $\mathcal{L}_v$, normal loss $\mathcal{L}_n$, edge loss $\mathcal{L}_e$, and Laplacian loss $\mathcal{L}_l$. The vertex loss

$$\mathcal{L}_v = \sum_{i=1}^{N} \left\| \boldsymbol{v}_i^{3D} - \hat{\boldsymbol{v}}_i^{3D} \right\|_2^2 + \left\| \boldsymbol{v}_i^{2D} - \hat{\boldsymbol{v}}_i^{2D} \right\|_2^2, \qquad (2)$$

$$\mathcal{L}_n = \sum_t \sum_{(i,j) \in t} \left\| \left\langle \hat{\boldsymbol{v}}_i^{3D} - \hat{\boldsymbol{v}}_j^{3D}, \boldsymbol{n}_t \right\rangle \right\|_2^2, \qquad (3)$$

$$\mathcal{L}_e = \sum_{i=1}^{E} \left(\|\boldsymbol{e}_i\|_2^2 - \|\hat{\boldsymbol{e}}_i\|_2^2 \right)^2, \qquad (4)$$

$$\mathcal{L}_l = \sum_{i=1}^{N} \left\| \boldsymbol{\delta}_i - \sum_{\boldsymbol{v}_k \in \mathcal{N}(\boldsymbol{v}_i)} \boldsymbol{\delta}_k \Big/ B_i \right\|_2^2, \qquad (5)$$

2

$$\mathcal{L}_{\mathcal{D}} = smooth_{L1}\left(D, \hat{D}\right),\ \hat{\mathcal{D}} = \mathcal{R}\left(\hat{\mathcal{M}}\right), \qquad (7)$$

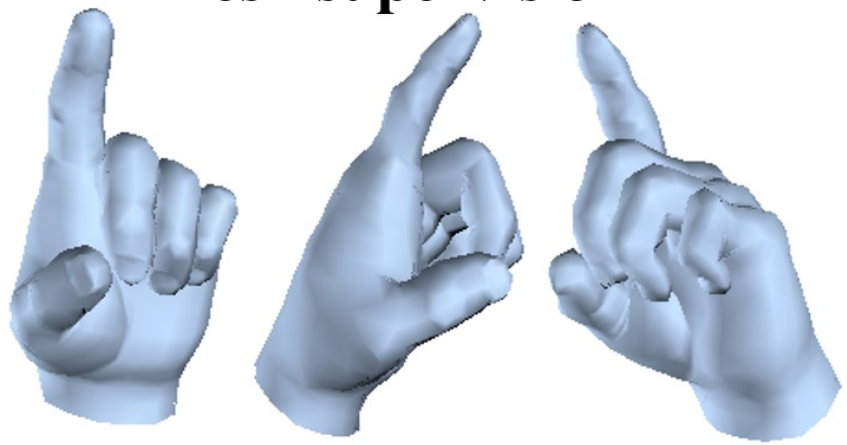
where $\mathcal{D}$ and $\hat{\mathcal{D}}$ denote the ground truth and rendered depth maps, respectively; $\mathcal{R}(\cdot)$ is the depth rendering function; $\hat{\mathcal{M}}$ is the estimated 3D hand mesh. We set the resolution of a depth map as 32×32 px.

$$\mathcal{L}_{p\mathcal{M}} = \lambda_e \mathcal{L}_e + \lambda_l \mathcal{L}_l$$

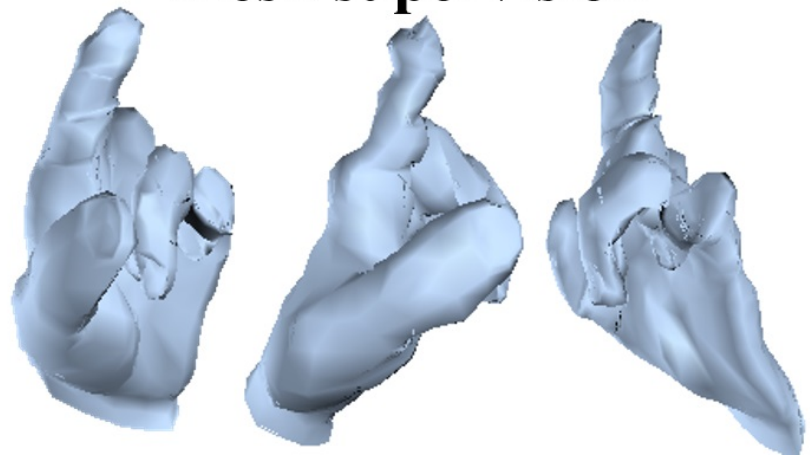
$$\mathcal{L}_{weakly} = \lambda_{\mathcal{H}} \mathcal{L}_{\mathcal{H}} + \lambda_{\mathcal{D}} \mathcal{L}_{\mathcal{D}} + \lambda_{p\mathcal{M}} \mathcal{L}_{p\mathcal{M}}, \qquad (8)$$

Input

With pseudo-ground truth mesh supervision



Without pseudo-ground truth mesh supervision



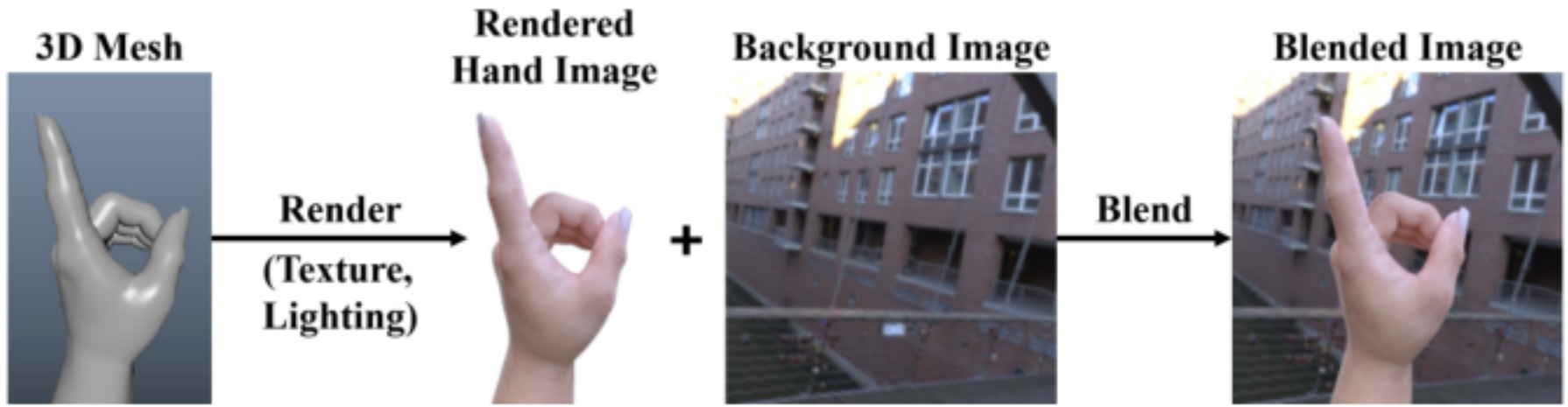


Figure 2: Illustration of our synthetic hand shape and pose dataset creation as well as background image augmentation during training.

Mesh error (mm)	MANO-based	Direct LBS	Ours
Our synthetic dataset	12.12	10.32	**8.01**
Our real-world dataset	20.86	13.33	**12.72**

Table 2: Average mesh errors tested on the validation set of our synthetic dataset and our real-world dataset. We compare our method with two baseline methods. Note that the mesh errors in this table are measured on the aligned mesh defined by MANO [42] for fair comparison.

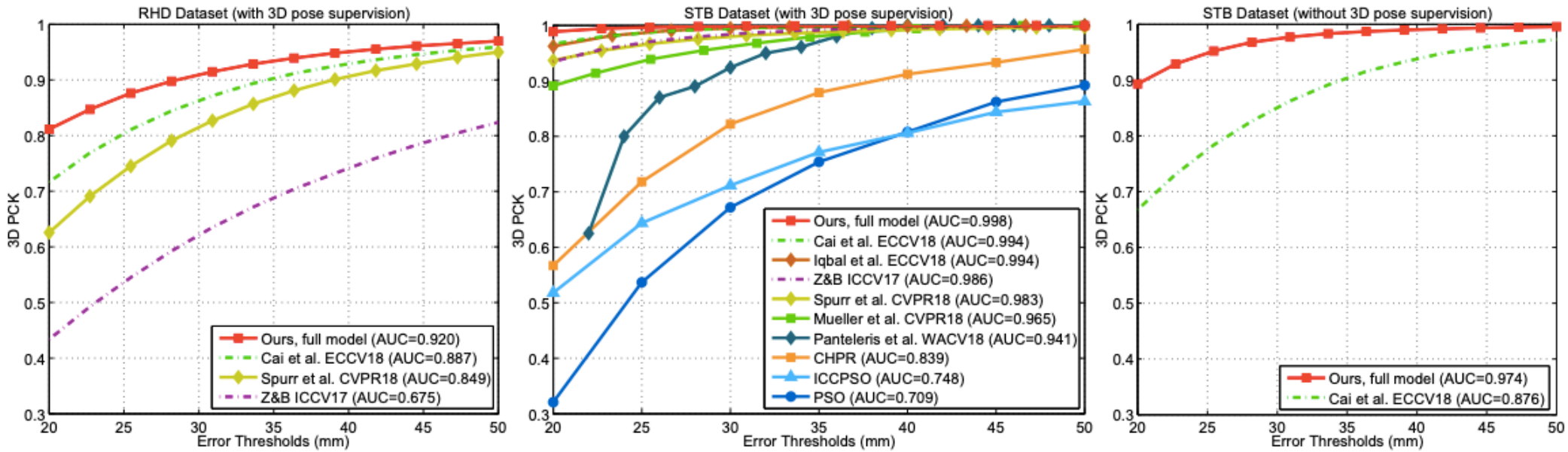


Figure 9: Comparisons with state-of-the-art methods on RHD [63] and STB [62] dataset. **Left**: 3D PCK on RHD dataset [63] with 3D hand pose supervision. **Middle**: 3D PCK on STB dataset [62] with 3D hand pose supervision. **Right**: 3D PCK on STB dataset [62] without 3D hand pose supervision. The AUC values are shown in parentheses.

Contribution

- propose a novel end-to-end trainable hand mesh generation approach based on Graph CNN
- propose a weakly-supervised training pipeline on real-world dataset, by rendering the generated 3D mesh to a depth map on the image plane and leveraging the reference depth map as a weak supervision, without requiring any annotations of 3D hand mesh or 3D hand pose for real-world images
- introduce the first large-scale synthetic RGB-based 3D hand shape and pose dataset as well as a small-scale real-world dataset, which contain the annotation of both 3D hand joint locations and the full 3D meshes of hand surface. We will share our datasets publicly upon the acceptance of this work.