2021 IEEE Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-0477-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/WACV48630.2021.00088

MVHM: A Large-Scale Multi-View Hand Mesh Benchmark for Accurate 3D Hand Pose Estimation

Liangjian Chen[1], Shih-Yao Lin[2], Yusheng Xie*[3], Yen-Yu Lin[4], and Xiaohui Xie[1]

[1]University of California, Irvine , [2]Tencent America , [3]Amazon , [4]National Chiao Tung University ,
{*liangjc2,xhx*}*@ics.uci.edu* , *mike.lin@ieee.org* , *yushx@amazon.com* , *lin@cs.nctu.edu.tw*

Abstract

Estimating 3D hand poses from a single RGB image is challenging because depth ambiguity leads the problem ill-posed. Training hand pose estimators with 3D hand mesh annotations and multi-view images often results in significant performance gains. However, existing multi-view datasets are relatively small with hand joints annotated by off-the-shelf trackers or automated through model predictions, both of which may be inaccurate and can introduce biases. Collecting a large-scale multi-view 3D hand pose images with accurate mesh and joint annotations is valuable but strenuous. In this paper, we design a spin match algorithm that enables a rigid mesh model matching with any target mesh ground truth. Based on the match algorithm, we propose an efficient pipeline to generate a large-scale multi-view hand mesh (MVHM) dataset with accurate 3D hand mesh and joint labels. We further present a multi-view hand pose estimation approach to verify that training a hand pose estimator with our generated dataset greatly enhances the performance. Experimental results show that our approach achieves the performance of 0.990 in $AUC_{20\text{-}50}$ on the MHP dataset compared to the previous state-of-the-art of 0.939 on this dataset. Our datasset is available at https://github.com/Kuzphi/MVHM.

1. Introduction

Estimating 3D hand poses from images has attracted increasing attention because it is essential to a wide range of applications such as human-computer interaction (HCI) [1, 27, 26], virtual reality (VR) [12, 18], augmented reality (AR) [7], medical diagnosis [20], and sign language understanding [43]. Although extensive research efforts have been made on this research topic for decades, there are still several unsolved challenges. One of the most crucial challenges is to handle the issue of depth ambiguity present in single view 3D hand pose estimation.

Conventional studies mainly focus on inferring 3D hand poses from either depth or RGB images directly. To address the problems caused by depth ambiguity, Some previous studies [5, 19, 8, 9] want to address this problem by leveraging depthmap information. These works come up with several ways to introduce depthmap into the training procedure, such as making depthmap as intermediate supervision [19] or using depth regularizer [5]. On the other hand, recent studies [22, 50] point out that imposing 3D hand shape (mesh) supervision can boost both the performance of 3D hand pose and shape estimators. It is clear that 3D hand shape brings richer hand structure information than hand keypoints. Furthermore, a preset mesh serves as a strong prior to reduce the freedom of the hand, therefore mitigating depth ambiguity. Along this line, several methods such as [3, 2, 45, 25, 40, 39, 48, 47, 46] are proposed. Despite the potential, the aforementioned methods highly rely on a preset hand model learned with a large number of accurate 3D mesh annotations. Hence, a large-scale dataset with accurate annotations of mesh vertices is in great demand.

Accurate mesh ground truth is hard to be manually an-

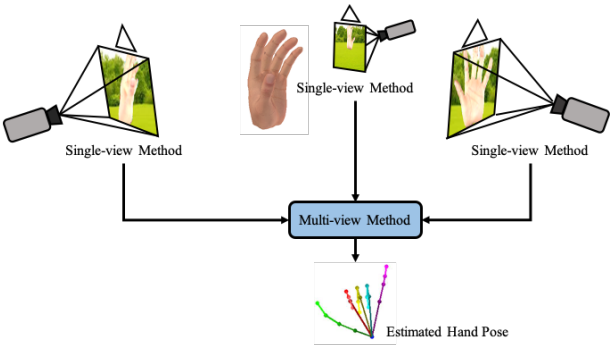


Figure 1. Our core idea. We build a synthetic dataset from a multi-view perspective, *e.g.*, rendering hand images from different a angles. With the aid of this dataset, a single-view method takes the image from each view and generates a possible hand pose candidate. We proposed a multi-view method takes different single-view predictions as input and predicts the final result.

*Work done outside of Amazon.

978-1-6654-0477-8/21/$31.00 ©2021 IEEE
DOI 10.1109/WACV48630.2021.00088

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

notated in general. The hand mesh annotations in most existing datasets are often annotated by hand shape estimator which can be inaccurate because hand mesh estimation itself is an even more challenging task. Most existing methods leveraging mesh information for 3D hand pose estimation are derived based on a single view. However, mere mesh information is insufficient to address depth ambiguity. Thus, 3D pose estimation still remains ill-posed in these methods.

The issue of depth ambiguity can be tackled by multi-view vision according to epipolar geometry. Multi-view sensing systems can capture hand images from cameras in different angles and therefore depth information of hands can be accurately inferred as long as camera parameters are known. Inspired by the above observations, we aim to build a large-scale multi-view hand mesh dataset that provides hand meshes and multi-view hand images simultaneously for training pose estimators.

In this work, we present an effective mechanism to synthesize 3D hand joint and mesh annotations, and establish a large-scale multi-view hand mesh (MVHM) dataset. We acquired a hand mesh model with a rigging system, and 3D hand ground truth from existing datasets, and a rigged hand model to match the given ground truth to perform various gestures. We render the hand model from different angles to collect multi-view images, as well as the 3D keypoints and mesh annotations to built MVHM. Then, we determine if the generated MVHM dataset can be used to improve 3D hand pose estimators. To this end, a multi-view based approach is developed for inferring 3D hand poses. The experimental results show that the resultant pose estimator can be greatly boosted by leveraging the generated MVHM dataset, and performs favorably against existing methods. This work makes three major contributions, which are summarized as follows:

1. We propose an effective mechanism for compiling a large-scale multi-view hand mesh (MVHM) dataset for 3D hand pose estimator training. To the best of our knowledge, this is the first large-scale hand dataset with multi-view hand images, accurate mesh annotations, hand joint keypoints labels.
2. We present a multi-view hand pose estimation approach based on an end-to-end trainable graph convolutional neural network where information from multi-view images is utilized to predict 3D hand poses.
3. Our proposed approach achieves the state-of-the-art performance on the benchmark, the MHP dataset, in both single-view and multi-view settings.

2. Related Work

2.1. RGB based 3D Hand Pose Estimation

RGB cameras are much more widely used than depth sensors. Estimating 3D hand poses merely from monocular RGB images are more practical and active in the literature [5, 10, 19, 29, 37, 41, 49, 24, 23, 11]. The pioneering work by Zimmermann and Brox [49] utilizes convolutional neural networks (CNN) to extract image feature, and feed camera parameters with these features to a 3D lift network where depth information is then estimated. Based on [49], Iqbal *et al*. [19] leverage depth maps as intermediate supervision. Meanwhile, Cai *et al*. [5] propose a weakly supervised approach that reconstructs the depth map and uses it as a regularizer during model training.

2.2. 3D Hand Mesh Estimation

3D hand pose estimation provides sparse joint locations. However, many computer vision applications would benefit more from hand shape information than sparse joints. Therefore, 3D hand mesh estimation, an effective shape representation, has emerged as an increasingly popular topic [16, 3, 2, 21, 45]. Most methods [3, 2, 45, 25, 40] are developed around a pre-defined deformable hand mesh model called MANO [32]. Because of the high degree of freedom and complexity of the hand gesture, searching for the right hand mesh in such a high dimensional space is quite challenging. Using this MANO model often relies on strong prior to constrain the model to only regress low-dimensional model parameters, and may ignore the high-dimensional information. Ge *et al*. [16] argue that mesh is a graph-structure data, and propose to directly regress 3D mesh vertices through graph convolutional neural network (GCN) with a pre-defined mesh graph.

2.3. Multi-View Hand Pose Estimation

Unlike single-view pose estimation, few research efforts focus on 3D hand pose estimation from multi-view data. Ge *et al*. [15] first introduce multi-view CNN to formulate it as an estimation problem. Their method assumes that hand joint locations independently follow 3D Gaussian distributions, and uses CNN to estimate the mean and variance of the location distribution of each joint. The main drawbacks of their method include 1) its inability to train in an end-to-end manner and 2) its impractical assumption about the independence among different joint locations. Simon *et al*. [33] propose a multi-view system which is trained to progressively improve hand keypoints detection. Their method would work well on fine-tuning a well pre-trained estimator but could not train a 3D hand pose estimator from scratch.

2.4. 3D Hand Pose Benchmark

There exist extensive research efforts such as [36, 38, 35, 42, 44, 49, 34, 30, 28, 25, 50] on building hand datasets for 3D hand pose estimation. We summarize the publicly available hand datasets and our dataset in Table 1. Most existing datasets do not contain mesh information, since labeling

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

Table 1. Comparison between our dataset with publicly available datasets. `Auto` in field `Annotation` represents that the annotation is made by some algorithms and therefore may not be accurate.`Mano` means the emsh annotaion is fitted by Mano Model

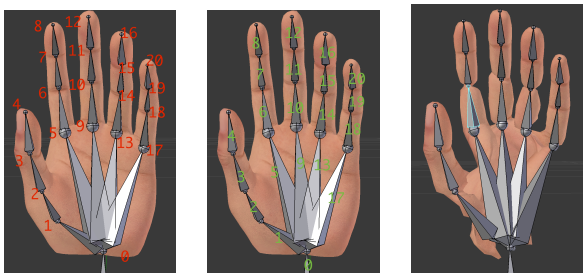
Dataset	RGB	Depth	Image Type	Resolution	Annotation	Dataset Size	Multi-View	Mesh
ICVL [36]	✗	✓	real	320 × 320	tracking	18K	✗	✗
NYU [38]	✗	✓	real	648 × 480	tracking	243K	✗	✗
MSRA [35]	✗	✓	real	1920 × 1080	tracking	76K	✗	✗
BigHand2.2M [42]	✗	✓	real	640 × 480	marker	2.2M	✗	✗
STB [44]	✓	✓	real	640 × 480	manual	36K	✗	✗
RHP [49]	✓	✓	synthetic	640 × 480	synthetic	44K	✗	✗
Dexter+Object [34]	✓	✓	real	640 × 480	manual	3K	✗	✗
EgoDexter [30]	✓	✓	real	640 × 480	manual	3K	✗	✗
MHP [17]	✓	✗	real	480 × 480	auto	80K	✓	✗
FreiHand [50]	✓	✗	real	224 × 224	auto	134K	✗	Mano
InterHand [28]	✓	✗	real	512 × 334	auto	2.2M	✓	Mano
Youtube Hand [25]	✓	✗	real	256×256	auto	47K	✗	Mano
Ours	✓	✓	synthetic	256×256	synthetic	320K	✓	✓

hand meshes manually is almost infeasible for human annotators.

To address the issue of labor-intensive annotations, recent studies [50, 25, 9] propose semi-automatic ways to label RGB images. FreiHand (Zimmermann *et al*. [50]) use an iterative process where the trained models first make predictions on the images and then the annotators are asked to make necessary adjustments. YoutubeHand (Kulon *et al*. [25]) run OpenPose [6] to get 2D annotations, upon which the parameters of the MANO model are regressed. Thresholding according to confidence scores is applied to remove those with low confidence, and hence ensures annotation quality. Despite the progress on efficiency and efficacy of labeling RGB images, the accuracy of annotation relies heavily on the pre-trained models used in the process. In addition, these methods rely on the MANO model as the ground-truth mesh generator, which could lose high-dimensional information of hands, as mentioned in Section 2.2. Compared to existing datasets, our dataset consists of large-scale RGB images and includes a variety of sequences. In addition, synthetic nature provides 100% accurate annotation for both hand joints and mesh. We make the first attempt to collect the dataset that provides large-scale, multi-view training images, thereby enhancing pose estimator training with a multi-view perspective.

3. Generating Muti-View Dataset

Currently, there exists no dataset providing *both* large-scale mesh and multi-view annotations of 3D hands, although many potential applications can benefit from such a dataset. Therefore, we create a new dataset called Multi-View Hand Mesh (MVHM) dataset for training multi-view hand pose estimators. To get accurate mesh and joint annotation, we use a well-made hand model from TurboSquid[1]



(a) (b) (c)

Figure 2. An example of hand joint and bone labels and their orders adopted in this work. (a) Joint labels. (b) Bone labels, which are used by Algorithm 1 during spin matching. (c) A failure case when directly rigging the hand mesh based on the bone coordinates without using Algorithm 1.

which provides around 2000 mesh vertices as well as an armature system to form various hand gestures. We render the images in the open-source software Blender[2].

In order to generate images and meshes from different gestures, we deploy the NYU dataset [38] which provides various hand pose and accurate keypoints annotation and rigged hand bone in our model to match with the given groundtruth. The hand bone in the rigged system consists of 7 degrees of freedom, 3 for its bone head, 3 for its bone tail, and 1 for its spin. The first 6 degrees of freedom determine the location of the bone and the last one represents its orientation. In order to rig the hand joints and the mesh surface correctly, we need to consider both location and orientation. Figure 2(c) shows an example of a distorted mesh obtained when we do not perform orientation match but simply move the bone location.

For this purpose, we define a bone vector as the difference between the bone tail and head. Assume u as the bone vector of the current bone we are working on. We take its

[1] `https://www.turbosquid.com`

[2] `https://www.blender.org`

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

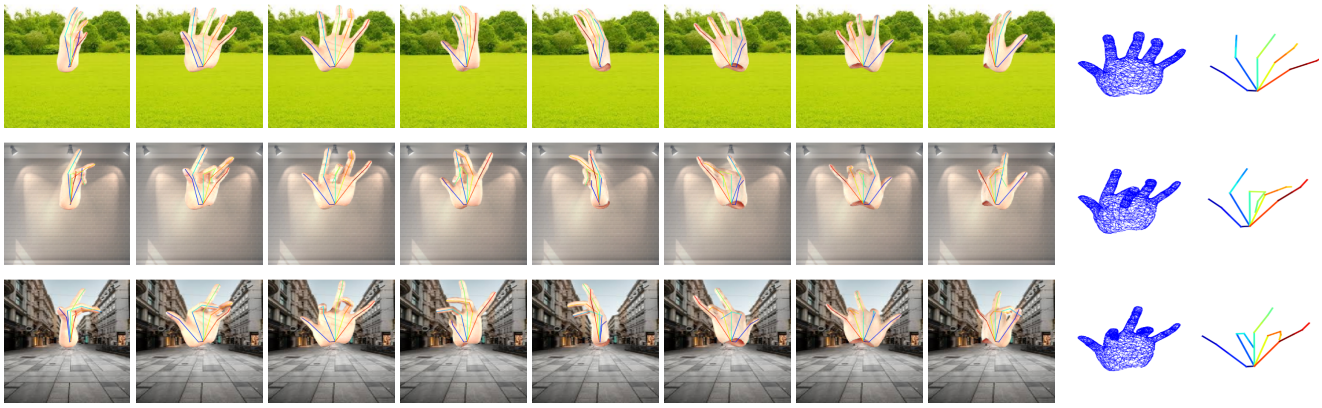


Figure 3. Some examples of the MVHM dataset. Column 1 to Column 8 shows the image with 2D annotation from view 1 to view 8, Column 9 shows the mesh. Column 10 shows the 3D annotation

adjacent bone's vector as the spin reference ref. We define spin sign vector as $u \times ref \times u$, and make sure this vector does not change after matching bone with groundturth. The detailed algorithm is summarized in Algorithm 1.

For each ground-truth gesture, we set 8 different camera positions that are evenly located on a circle within the plane perpendicular to the palm. All 8 cameras point to the center of the palm to ensure that the hand locates at the center of each rendered image. Figure 4 shows the scene when we render hand images.

To increase the diversity of the collected MVHM, we randomly change the intensity of the light and global illumination in the blender. In addition, we select some background scenes from online sources, and randomly use them as background during our rendering. We render $320,000$ images of resolution 256×256 for MVHM construction.

We emphasize that each sample in MVHM comes with full annotations of 21 hand joint and 2651 mesh vertices. Following the setting in [49], each finger is fully represented by 4 keypoints (Metacarpophalangeal, Proximal interphalangeal, Distal interphalangeal and fingertip), additionally. Carpometacarpal joints are also labeled in MVHM. Figure 2(a) shows a sample of the hand joint labels. In addition, we also release the hand segmentation mask, the camera intrinsic matrix, and the optical flow for each sample. Nevertheless, we in this paper only use the multi-view and mesh information from the collected MVHM dataset for hand pose estimator training.

4. Methodology

4.1. Overview

Given an RGB image of a hand $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$, our goal is to estimate the 3D joint locations of the hand $\mathbf{P}_j \in \mathbb{R}^{k \times 3}$, where W and H denote the image height and width respectively, and K is the number of the hand joints. Recent studies [16, 3] have demonstrated that using the mesh distance loss as an intermediate supervision during training can boost the performance of the learned hand pose estimator. Inspired by the approach [16], we define a hand mesh as a bidirectional graph $\mathbf{G}(\mathbf{V}, \mathbf{\Lambda})$, where $\mathbf{V}$ is the vertex set and $\mathbf{\Lambda}$ is the adjacency matrix. We also assume that $\mathbf{V}$ contains N different elements (*i.e.*, points on the mesh) and our mesh estimator would predict the 3D locations $\mathbf{P}_m \in \mathbb{R}^{N \times 3}$ for all vertices in $\mathbf{V}$.

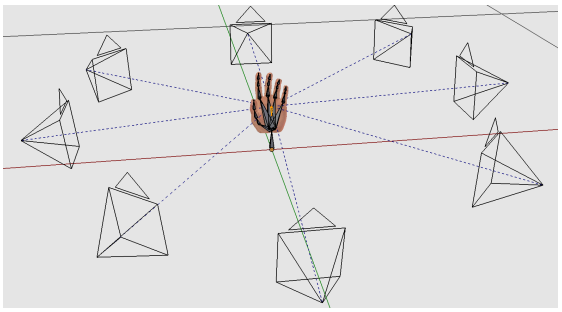


Figure 4. Synthetic scene when rendering hands in MVHM.

In our single-view approach, we use the stacked hourglass [31] as the CNN backbone to extract hand features from an image. The graph convolution network (GCN) is applied to estimate the 3D pose and mesh. Figure 5 shows the architecture of our single-view network, which consists of three major components: the 2D evidence network, mesh evidence network, and 3D pose estimator. These components are elaborated in the following subsections.

4.2. 2D Evidence Network

The 2D evidence network offers two main functionalities. First, it estimates hand keypoint heatmaps to obtain the 2D hand joint locations. Second, it extracts image features that then serve as the input to the mesh evidence network. We denote the estimated heatmaps as $\boldsymbol{H} \in \mathbb{R}^{K \times H \times W}$. As shown in Figure 5, the hourglass backbone gives two outputs, including the estimated hand joint heatmaps and the

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Spin matching Algorithm for rigging hand mesh base on 3D hand pose ground truth

Input:
C is the array of 3D keypoints ground truth that we want our mesh to match with.
B is the array of hand bones in the rig system. Each bone has two attributes, head and tail, which represent the beginning and end location of the bone.
B and C is stored in an array whose orders are shown in Figure 2(a) and 2(b)

```
begin
    Move B[0].tail to location C[0] ;
    // spin the bone inside the palm
    for i ∈ {1, 5, 9, 13, 17} do
        // bone vector
        u ← B[i].tail - B[i].head;
        v ← C[i] - C[0] ;
        adj ← i + 4;
        if i = 17 then
            adj ← i − 4;
        end
        // reference vector
        ref_ori ←B[adj].tail - B[adj].head;
        ref_aft ←C[adj] - C[0] ;
        Move B[i] to match groundturth;
        // sign vector
        e_1 = u × ref_ori × u;
        e_2 = v × ref_aft × v ;
        Spin B[i] with the angle between e_1 & e_2;
    end
    for i ← 1 to 21 do
        if i mod 4 ≠ 1 then
            Move B[i] to match groundturth;
            B[i] performs the same spin as B[i-1] ;
        end
    end
end
```

extracted features. The ground-truth heatmaps $\bar{\boldsymbol{H}}_s$ are obtained by smoothing the keypoint location k^{th} with Gaussian blur. To train the 2D evidence network, we apply the heatmap loss $\boldsymbol{\mathcal{L}}_h$ to each hourglass block as supervision. The heatmap loss is defined by

$$\boldsymbol{\mathcal{L}}_h = \frac{1}{S * K} \sum_{s=1}^{S} \sum_{k=1}^{K} ||\boldsymbol{H}_k^s - \bar{\boldsymbol{H}}_k||_F^2, \tag{1}$$

where S and K denote the number of the hourglass blocks and keypoints, respectively.

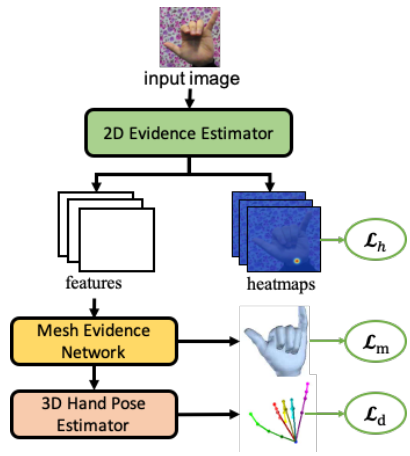


Figure 5. Overview of our single-view method. Given a single-view RGB image, the 2D evidence network predicts its heatmap and outputs the encoded image features. The mesh evidence network takes image features as input and outputs the hand mesh. Based on the estimated mesh, the 3D pose estimator gives final hand pose prediction.

4.2.1 Mesh Evidence Network

Our mesh evidence network is built on the basis of spectral GCN [4]. Given the image features extracted by the 2D evidence network, our mesh evidence network estimates the 3D hand mesh. A 3D hand mesh is represented by a set of vertex coordinates $\mathbf{P}_m \in \mathbb{R}^{N\times 3}$ where N is the number of the vertices in the hand mesh. We represent a hand mesh as a graph $\mathbf{G}(\mathbf{V}, \mathbf{\Lambda})$, where $\mathbf{V}$ is the vertices set, and $\mathbf{\Lambda}$ is the adjacency matrix. $\mathbf{\Lambda}_{i,j}$ is 1 if there is an edge between vertex i and vertex j, otherwise it is 0.

Specifically, we first normalize the adjacency matrix $\mathbf{\Lambda}$ via graph Laplacian operation and obtain a normalized adjacency matrix $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\Lambda\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is the degree matrix of graph $\mathbf{G}$ and $\boldsymbol{I}$ is an identity matrix. Graph spectral decomposition is then used to decompose the normalized adjacency matrix $\boldsymbol{L}$ as $\boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^T$, where $\boldsymbol{A} = diag(\lambda_1, \lambda_2, ..., \lambda_N)$ consists of the eigenvalues of the $\boldsymbol{L}$, where λ_{max} is the largest eigenvalue of $\boldsymbol{L}$.

Following [13], we define the convolution kernel $\hat{\boldsymbol{A}}$ in GCN as

$$\hat{\boldsymbol{A}} = diag(\sum_{i=0}^{S} \alpha_i \lambda_1^i, ..., \sum_{i=0}^{S} \alpha_i \lambda_C^i), \tag{2}$$

where α is the kernel parameter and S is a pre-set hyper-parameter used to control the receptive field.

Thus, the GCN convolutional operation is defined by

$$\boldsymbol{F}' = \boldsymbol{U}\hat{\boldsymbol{\Lambda}}\boldsymbol{U}^T\boldsymbol{F}\boldsymbol{\theta}_i = \sum_{i=0}^{S} \alpha_i \boldsymbol{L}^i \boldsymbol{F}\boldsymbol{\theta}_i, \tag{3}$$

where $\boldsymbol{F} \in \mathbb{R}^{N\times F_{\text{in}}}$ and $\boldsymbol{F}' \in \mathbb{R}^{N\times F_{\text{out}}}$ indicate the input and output features respectively, and $\boldsymbol{\theta}_i \in \mathbb{R}^{F_{\text{in}}\times F_{\text{out}}}$ is train-

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

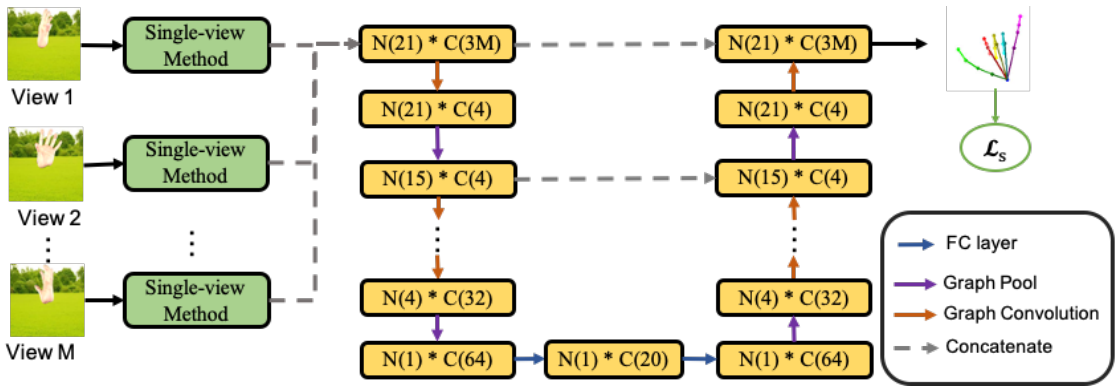


Figure 6. Overview of the multi-view method. The single-view method first predicts the hand pose for each view independently. A graph U-Net takes the concatenation of these single-view predictions as input, and estimates the final pose estimation. $N(\cdot)$ and $C(\cdot)$ represent the number of nodes in the graph and feature size of each node, respectively.

able parameter used to refine the input feature and control the output channel size.

Since our hand mesh surface is composed of 2561 vertices, it takes a huge computational cost to apply the above operation to each vertex because the time complexity of matrix multiplication for $\boldsymbol{L}^i$ is $O(N^3)$. Therefore, we utilize the Chebyshev polynomial approximation to reduce the complexity. The convolutional operation is then defined by

$$\boldsymbol{F}' = \sum_{i=0}^{S} \alpha_i T_i(\hat{\boldsymbol{L}})\boldsymbol{\theta}_i, \tag{4}$$

where $T_k(x)$ is the kth Chebyshev polynomial and $\hat{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - \boldsymbol{I}$ is used to normalize the input features.

To enable our model to learn both local and global features, we adopt a scheme that is used in [13, 16] for generating hand meshes from coarse to fine. We leverage the heavy-edge matching algorithm to coarsen the graph by three different coarsening levels, and record the mapping between graph nodes in every two consecutive levels. In the forward pass, our model first constructs the most coarse hand mesh and then up-samples more nodes from the coarse graph to the fine graph based on the stored mappings.

At the last layer of the GCN, we set F_{out} to 3 to represent the 3D coordinate vertices. Also, we apply the l_2 loss between the ground-truth mesh and prediction map as the mesh loss function:

$$\mathcal{L}_m = \frac{1}{N}||\boldsymbol{P_m} - \bar{\boldsymbol{P}}_{\boldsymbol{m}}||_F^2. \tag{5}$$

4.3. 3D Depth Evidence Network

The proposed 3D evidence network infers the depth of 3D hand keypoints P_d from the predicted hand mesh P_m by the mesh evidence network. Taking P_m as the input, we adopt a two layers GCN with a similar structure of the mesh evidence network to predict the pose features. These pose features are then fed to two fully connected layers to regress the depth of 3D hand keypoint locations. The corresponding loss is defined by

$$\mathcal{L}_d = \frac{1}{K}||\boldsymbol{D} - \bar{\boldsymbol{D}}||_F^2, \tag{6}$$

where $\boldsymbol{D} \in \mathbb{R}^K$ and $\bar{\boldsymbol{D}} \in \mathbb{R}^K$ represent the predicted and the ground-truth joint depths, respectively.

To infer the 3D hand keypoints, we use non-maximum suppression to get the 2D coordinates from the estimated heatmaps. With the estimated 2D coordinates and the depth map calculated by the 3D depth evidence network, we then obtain 3D coordinates in the camera coordinate system. Since the camera parameters are known, we are then able to infer hand keypoints in the world system.

4.4. Multi-view Method

Based on our single-view method, we propose a simple yet effective multi-view approach to hand pose estimation. Figure 6 illustrates the core idea of our approach.

Our single-view method predicts the 3D hand pose for each view independently. We concatenate these view-specific predictions on their coordinate channel. The concatenated prediction serves as the input features to a graph U-Nets[14] and predicts the final 3D hand keypoints. We utilize the L_2 distance as the loss function in our multi-view network, *i.e.*,

$$\mathcal{L}_s = \frac{1}{K}||\boldsymbol{P}_j - \bar{\boldsymbol{P}}_j||_F^2, \tag{7}$$

where $\boldsymbol{P}_j$ and $\bar{\boldsymbol{P}}_j$ represent the predicted and the ground-truth joint depth, respectively.

5. Experiment Setting

5.1. Datasets for Evaluation

We evaluate our single-view approach on two benchmark hand pose datasets, including the Stereo Tracking Benchmark (STB) Dataset [44] and the Multi-view 3D Hand Pose

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

Figure 7. Examples of the two hand pose datasets used for evaluation. The first row shows images with the annotated hand poses from the STB dataset [44] while the second row shows those from the MHP dataset [17].

(MHP) dataset [17]. The proposed multi-view approach is evaluated on the MHP dataset. Both the MHP and STB datasets provide real hand video sequences performed by different people in various backgrounds. The hand joint annotations of the STB dataset are manually labeled while the annotations of the MHP dataset are obtained by using the Leap Motion sensor. The MVHM dataset we build is used in all of our experiments. We aim at determining if training the hand pose estimators with the MVHM dataset can be effectively improved in different experimental settings.

For the STB dataset, we use its SK subset, which contains 6 different hand videos, to evaluate our approach. Following the train-validation split setting in [16], we take the first video as the validation set while the rest videos serve as the training set.

The MHP dataset includes 21 different hand motion videos. Each hand motion video provides hand RGB images and multiple types of annotations in each sample, including bounding boxes and 2D/3D hand joint locations. Figure 7 displays some examples of the STB and MHP datasets. We follow [5, 49] and apply the standard data pre-processing for both of the STB and MHP datasets. During data pre-processing, we firstly crop the images to remove the irrelevant background and make sure the hands are located at the center of the images. All the cropped images are then resized to resolution 256×256. Secondly, we follow the mechanism used in [5] to change the hand center from the palm center to the joint of the wrist for data in both of the STB and MHP datasets.

5.2. Metrics

We follow the settings from previous researches [49, 16] and adopt the *average end-point-error* (EPE_m), and the *area under the curve* (AUC) on the *percentage of correct keypoints* (PCK) within a threshold range as the metrics to evaluate model effectiveness. We report the performance in both AUC on PCK between 0mm and 50mm as well as between 20mm and 50mm.

Table 2. Ablation studies of 3D hand pose estimation on the STB and MHP datasets. ↑: higher is better; ↓: lower is better; The measuring unit of EPE is millimeter(mm). SV stands for the single-view method and MV represents the multi-view method.

	$\text{AUC}_{0\text{-}50}$ ↑	$\text{AUC}_{20\text{-}50}$ ↑	EPE_m ↓
MHP Dataset			
SV w/o MVHM	0.604	0.802	22.13
SV w/ MVHM	0.660	0.857	18.09
MV w/o MVHM	0.832	0.985	8.43
MV w/ MVHM	**0.895**	**0.990**	**5.20**
STB Dataset			
SV w/o MVHM	0.820	0.987	8.95
SV w/ MVHM	**0.832**	**0.991**	**8.38**

Table 3. Results on the MHP dataset. ↑: higher is better.

	$\text{AUC}_{20\text{-}50}$ ↑
Zimmermann *et al.* [50]	0.717
Cai *et al.* [5]	0.928
Chen *et al.* [8]	0.939
Our multi-view method	**0.991**

5.3. Implementation Details

We implement our single-view and multi-view approaches in Python with PyTorch. In the training phase, we set the batch size as 8, and use the Adam solver with an initial learning rate 0.01. Both models are trained on a server with four GeForce GTX 1080-Ti GPUs.

When training the single-view network, we use a multi-stage training strategy. In the first stage, we train our 2D evidence network with the heatmap loss $\boldsymbol{\mathcal{L}}_h$. In the second stage, we fix the weights of the 2D evidence network and train the mesh network with mesh loss $\boldsymbol{\mathcal{L}}_m$. In the third stage, we fix the weights of both of the 2D evidence network and mesh network, and focus on training the joint depth network with loss $\boldsymbol{\mathcal{L}}_d$. In the final stage, the whole network is optimized end-to-end.

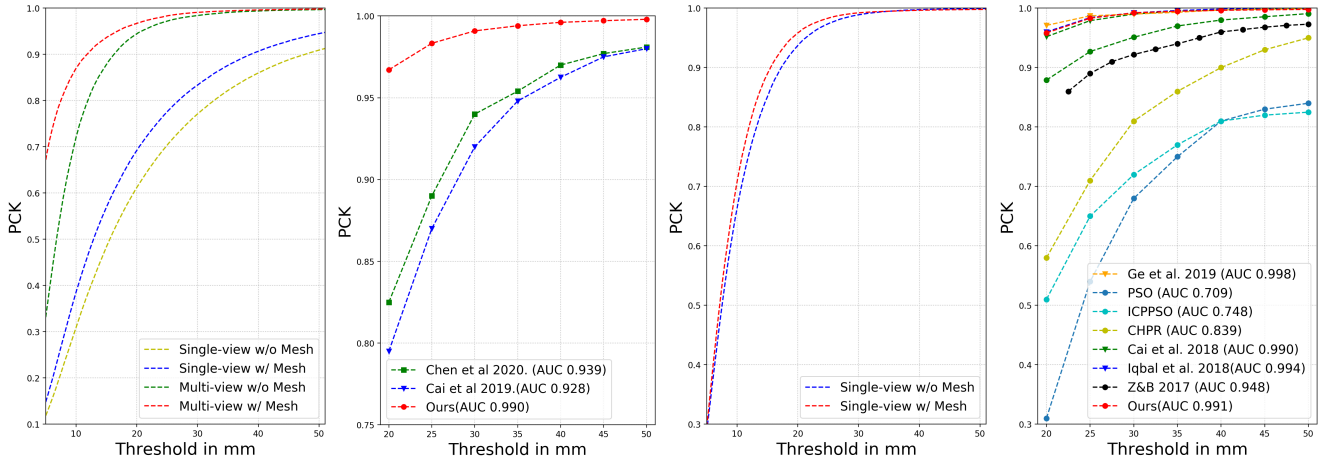
For training the multi-view network, we apply the same multi-stage training strategy. In the first stage, we use the pre-trained weights from the single-view network for initializing the 3D hand single-view network, and keep this part fixed for training the 3D hand fusion network. In the second stage, we activate both networks and fine-tune the whole network architecture in an end-to-end manner.

6. Experimental Results

6.1. Multi-view task

To evaluate the effectiveness of the proposed multi-view method, we compare our single-view method with our multi-view method on the MHP dataset under the setting of with or without using data from the MVHM dataset. Table 2 and Figure 8(a) show that utilizing the multi-view information from the MHP dataset itself boosts the testing perfor-

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Figure 8. Ablation studies and comparison of the state-of-the-art methods for single-view pose estimation. (a) PCK results of different settings on the STB dataset. (b) Comparison results in PCK for the state-of-the-art methods on the STB dataset. (c) PCK results under different settings on the MHP dataset. (d) Comparison results in PCK for the state-of-the-art methods on the MHP dataset.

mance in $AUC_{0\text{-}50}$, $AUC_{20\text{-}50}$, and EPE_m by large margins, *i.e.*, 0.218, 0.183, and 13.80mm respectively. When additional data from the MVHM dataset are used, substantial performance gains are achieved, which reveals the effectiveness of using the collected MVHM dataset for training.

Three current state-of-the-art methods are chosen for comparing with our method on the MHP dataset, including Zimmermann *et al*. [50] (0.717 in $AUC_{20\text{-}50}$), Cai *et al*. [5] (0.928 in $AUC_{20\text{-}50}$)[3], and Chen *et al*. [8] (0.939 in $AUC_{20\text{-}50}$). Zimmermann *et al*. [50] just report the numerical result so we include their result in Table 3 and does not show it in Figure 8(b). Our multi-view method achieves the performance of 0.990 in $AUC_{20\text{-}50}$, outperforming these competing methods by a large margin. This experiment shows that both the proposed multi-view method and the established MVHM dataset are beneficial and can work together to get the new state-of-the-art performance on the MHP dataset.

6.2. Single-view task

To further validate the effectiveness of the generated mesh dataset MVHM in addition to multi-view methods, we also conduct the following experiments for comparison on single-view methods. We compare the results when models are trained solely on the MHP/STB datasets and trained on the MHP/STB datasets together with the MVHM dataset. Table 2, Figure 8(a) and Figure 8(c) show, on both MHP and STB datasets, adding the mesh data greatly enhances the performance by granting a model the ability to capture the mesh-level features, therefore leading to better results.

We select seven powerful and recently published methods for comparison with the proposed method, including PSO [3], ICPPSO [10], CHPR [44], Iqbal *et al*. [19], Cai *et al*. [5], Zimmermann and Brox [49], and Ge *et al*. [16]. The AUC curves are plotted in Figure 8(d). Ge *et al*. [16] also utilize an additional dataset to train their model and got the STOA result, which demonstrates the effectiveness of their mesh dataset. Besides, they introduce more complicated mesh metrics like the surface norm loss. Iqbal *et al*. [19] and Cai *et al*. [5] leverage additional depth-map information to derive their models, and achieve good results. As a multi-view approach without complicated components, our method is on par with methods by Ge *et al*. [16] and Iqbal *et al*. [19] while outperforms most of them on single-view tasks.

7. Conclusions

Estimating 3D hand poses from monocular images is an ill-posed problem due to its depth ambiguity. Nevertheless, multi-view images could make up the deficiency. To this end, we build a multi-view mesh hand dataset, MVHM, to enable training 3D pose estimators with mesh supervision. We present a multi-view method that effectively fuses single-view predictions. When testing on the real-world multi-view dataset MHP, our multi-view method with the aid of the MVHM dataset achieves the state-of-the-art performance.

Acknowledgement. The authors thank Kun Han for helping generate the dataset images used in this work. This work was supported in part by the Ministry of Science and Technology (MOST) under grants MOST 107-2628-E-009-007-MY3, MOST 109-2634-F-007-013, and MOST 109-2221-E-009-113-MY3, and by Qualcomm through a Taiwan University Research Collaboration Project.

[3]Cai *et al*. [5] do not report the results in their paper. Here we report the re-implementaition results by Chen *et al*. [8].

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

References

[1] Shamama Anwar, Subham Kumar Sinha, Snehanshu Vivek, and Vishal Ashank. Hand gesture recognition: a survey. In *Nanoelectronics, Circuits and Communication Systems*. 2019.

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, 2019.

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

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[5] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *TPAMI*, 2018.

[7] Tejo Chalasani and Aljosa Smolic. Simultaneous segmentation and recognition: Towards more accurate ego gesture recognition. In *ICCVW*, 2019.

[8] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, Wei Fan, and Xiaohui Xie. Dggan: Depth-image guided generative adversarial networks fordisentangling rgb and depth images in 3d hand pose estimation. In *WACV*, 2020.

[9] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In *WACV*, 2021.

[10] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Hui Tang, Yufan Xue, Yen-Yu Lin, Xiaohui Xie, and Wei Fan. Tagan: Tonality-alignment generative adversarial networks for realistic hand pose synthesis. In *BMVC*, 2019.

[11] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Hui Tang, Yufan Xue, Xiaohui Xie, Yen-Yu Lin, and Wei Fan. Generating realistic training images based on tonality-alignment generative adversarial networks for hand pose estimation. *arXiv preprint arXiv:1811.09916*, 2018.

[12] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Trans Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.

[13] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016.

[14] Hongyang Gao and Shuiwang Ji. Graph u-nets. *arXiv preprint arXiv:1905.05178*, 2019.

[15] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016.

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

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

[18] Yi-Ping Hung and Shih-Yao Lin. Re-anchorable virtual panel in three-dimensional space, Dec. 27 2016. US Patent 9,529,446.

[19] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, 2018.

[20] MR Mohamad Ismail, CK Lam, K Sundaraj, and MHF Rahiman. Hand motion pattern recognition analysis of forearm muscle using mmg signals. *Bulletin of Electrical Engineering and Informatics*, 8(2):533–540, 2019.

[21] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[22] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning to estimate pose and shape of hand-held objects from rgb images. *arXiv preprint arXiv:1903.03340*, 2019.

[23] Deying Kong, Yifei Chen, Haoyu Ma, Xiangyi Yan, and Xiaohui Xie. Adaptive graphical model network for 2d handpose estimation. *arXiv preprint arXiv:1909.08205*, 2019.

[24] Deying Kong, Haoyu Ma, and Xiaohui Xie. Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. *arXiv preprint arXiv:2009.12473*, 2020.

[25] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, pages 4990–5000, 2020.

[26] Shih-Yao Lin, Yun-Chien Lai, Li-Wei Chan, and Yi-Ping Hung. Real-time 3d model-based gesture tracking for multimedia control. In *ICPR*, 2010.

[27] Shih-Yao Lin, Chuen-Kai Shie, Shen-Chi Chen, and Yi-Ping Hung. Airtouch panel: a re-anchorable virtual touch panel. In *MM*, 2013.

[28] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image.

[29] 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 *CVPR*, 2018.

[30] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *ICCVW*, 2017.

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016.

[32] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245, 2017.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.

[33] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[34] 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 *ECCV*, 2016.

[35] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[36] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*, 2014.

[37] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. *CVPR*, 2019.

[38] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, 2014.

[39] Zhenyu Wu, Duc Hoang, Shih-Yao Lin, Yusheng Xie, Liangjian Chen, Yen-Yu Lin, Zhangyang Wang, and Wei Fan. Mm-hand: 3d-aware multi-modal guided hand generative network for 3d hand pose synthesis. In *MM*, 2020.

[40] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. *arXiv preprint arXiv:2007.05168*, 2020.

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

[42] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017.

[43] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American sign language recognition with the kinect. In *ICMI*, 2011.

[44] 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.

[45] Xiong Zhang, Qiang Li, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. *arXiv preprint arXiv:1902.09305*, 2019.

[46] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. *arXiv preprint arXiv:2002.04724*, 2020.

[47] Zhengli Zhao, Samarth Sinha, Anirudh Goyal, Colin Raffel, and Augustus Odena. Top-k training of gans: Improving gan performance by throwing away bad samples. *arXiv preprint arXiv:2002.06224*, 2020.

[48] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

[49] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *CVPR*, 2017.

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

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 09,2026 at 13:08:15 UTC from IEEE Xplore. Restrictions apply.