# MobileHand: Real-Time 3D Hand Shape and Pose Estimation from Color Image

Guan Ming Lim[1](✉), Prayook Jatesiktat[2], and Wei Tech Ang[1,2]

[1] School of Mechanical and Aerospace Engineering,
Nanyang Technological University, 50 Nanyang Avenue,
Singapore 639798, Singapore
guanming001@e.ntu.edu.sg, wtang@ntu.edu.sg

[2] Rehabilitation Research Institute of Singapore, Nanyang Technological University,
11 Mandalay Road, Singapore 308232, Singapore
prayook001@e.ntu.edu.sg

**Abstract.** We present an approach for real-time estimation of 3D hand shape and pose from a single RGB image. To achieve real-time performance, we utilize an efficient Convolutional Neural Network (CNN): MobileNetV3-Small to extract key features from an input image. The extracted features are then sent to an iterative 3D regression module to infer camera parameters, hand shapes and joint angles for projecting and articulating a 3D hand model. By combining the deep neural network with the differentiable hand model, we can train the network with supervision from 2D and 3D annotations in an end-to-end manner. Experiments on two publicly available datasets demonstrate that our approach matches the accuracy of most existing methods while running at over 110 Hz on a GPU or 75 Hz on a CPU.

**Keywords:** End-to-end learning · 3D hand tracking · Efficient CNN

## 1 Introduction

Our hands play an important role in our interaction with the environment. Therefore, the ability to understand the hand shape and motion from color images is useful for a myriad of practical applications such as hand sign recognition, virtual/augmented reality, human-computer interaction, hand rehabilitation assessment and many more. New opportunities could also be realized if the hand tracking algorithm could run efficiently on mobile devices to take advantage of its portability and ubiquitous nature.

Although some methods are capable of tracking 2D or 3D hand joints on mobile devices [2,7], 3D hand shape and pose estimation is still restricted to devices with GPU hardware. As compared to sparse prediction of hand joint

Supported by Agency for Science, Technology and Research (A*STAR), Nanyang Technological University (NTU) and the National Healthcare Group (NHG). Project code: RFP/19003.

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 450–459, 2020.
https://doi.org/10.1007/978-3-030-63820-7_52

**Table 1.** List of recent works on hand shape and pose estimation from color image

| Authors (Publication) | Type of CNN used for feature extraction | Type/generation of hand model | Runtime |
|---|---|---|---|
| Baek *et al.* (CVPR'19) [1] | ResNet-50 | MANO | Nil |
| Boukhayma *et al.* (CVPR'19) [3] | ResNet-50 | MANO | Nil |
| Ge *et al.* (CVPR'19) [6] | Stacked hourglass, residual network | Graph CNN | 50 Hz (GPU GTX 1080) |
| Hasson *et al.* (CVPR'19) [10] | ResNet-18 | MANO | 20 Hz (GPU Titan X) |
| Zhang *et al.* (ICCV'19) [20] | Stacked hourglass | MANO | Nil |
| Kulon *et al.* (CVPR'20) [14] | ResNet-50 | Spatial mesh conv. decoder | 60 Hz (GPU RTX 2080 Ti) |
| Zhou *et al.* (CVPR'20) [22] | ResNet-50 | MANO | 100 Hz (GPU GTX 1080 Ti) |
| This work | MobileNetV3-Small | MANO | 110 Hz (GPU RTX 2080 Ti)<br>75 Hz (CPU 8-Core) |

positions, dense recovery of 3D hand mesh is considerably more useful as it offers a richer amount of information. Therefore, the design of an efficient method for estimating 3D hand shape and pose remains an open and challenging problem.

In this work, we present an approach for real-time estimation of hand shape and pose, by using a lightweight Convolutional Neural Network (CNN) to reduce computation time. Although some tradeoff between speed and accuracy is unavoidable, our experiments on two datasets demonstrate that while the accuracy is comparable to most of the existing methods, the runtime of the proposed network is the fastest among all competitive approaches. We also proposed a simple joint angle representation to articulate a commonly used 3D hand model, which helps to improve accuracy. The video demonstrations and software codes are made available for research purposes at https://gmntu.github.io/mobilehand/.

## 2 Related Work

The advance in deep learning and ease of using a monocular RGB camera to capture hand motion, have motivated many previous works to use deep neural networks to estimate 3D hand pose from a single RGB image [4,12,16,18,24].

But recent works as listed in Table 1, are moving towards the estimation of hand shape together with pose because a 3D hand mesh is much more expressive. For example, it allows the computation of contact loss from mesh vertices during hand-object interaction [10], and also enables the rendering of 2D hand silhouette to refine hand shape and pose prediction [1,3,20]. As shown in Table 1, while two

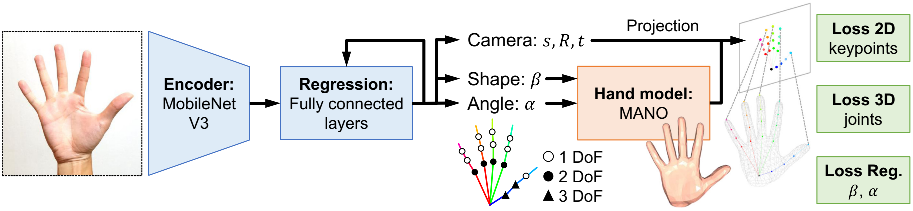

**Fig. 1.** Framework overview: a cropped image of a hand is passed through a CNN encoder to extract key features which are sent to an iterative regression module to infer a set of parameters. The shape and joint angles are used by the hand model to generate a 3D hand mesh which is projected to a 2D image plane using the camera parameters. By incorporating the generative and differentiable hand model as part of the deep learning architecture, the network can be trained end-to-end using both 2D keypoints and 3D joints supervision.

of the methods generate a 3D hand mesh using Graph CNN [6] or spatial mesh convolutional decoder [14], most of the methods employ a common parametric mesh model (MANO [17]) to exploit the inherent geometric priors encoded in the 3D hand model.

Although the runtimes of existing methods have achieved real-time rates on a GPU, we show that it is possible to further improve the computational performance and achieve real-time rates on CPU as well, making it suitable to be extended for mobile phone applications.

## 3 Method

The proposed method as illustrated in Fig. 1, is inspired by the work on end-to-end recovery of human body shape and pose [13]. To further improve the speed and accuracy of the method for hand shape and pose estimation, we proposed two key modifications: 1) an efficient CNN for image encoder and 2) a direct joint angle representation to articulate a 3D hand model. More details on the proposed framework are provided in the following sections.

### 3.1 Neural Network Architecture

The neural network architecture consists of two main parts: an image encoder and an iterative 3D regression module with feedback.

**Image Encoder:** We utilize MobileNetV3-Small [11] to extract image features as it is one of the latest generations of efficient and lightweight CNN targeted for mobile devices. The encoder takes in an RGB image (224 by 224 pixels) and the structure of MobileNetV3-Small is used up to the average pooling layer to output a feature vector $\phi \in \mathbb{R}^{576}$.

**Iterative Regression:** With the extracted feature vector $\phi \in \mathbb{R}^{576}$, it is possible to use a fully connected layer to directly regress the camera, hand shape and joint angle parameters $\Theta = \{s, R, t, \beta, \alpha\} \in \mathbb{R}^{39}$ [3]. However, it is challenging to regress $\Theta$ in one forward pass, due to large semantic gap [20] and especially when $\Theta$ includes rotation parameters $R$ and $\alpha$ [13].

Thus, we use an iterative regression module [13,20] to make progressive changes to an initial estimate. This helps to simplify the learning problem as the module only needs to predict the change to move the parameters closer to the ground truth [5]. More specifically, the feature vector $\phi$ and current parameter vector $\Theta_t$ are concatenated and fed into a fully connected network that outputs the residual $\Delta\Theta_t$. The residual is then added to the current parameter to obtain a more accurate estimate $\Theta_{t+1} = \Theta_t + \Delta\Theta_t$. The initial estimate $\Theta_0$ is set as a zero vector $\mathbf{0} \in \mathbb{R}^{39}$, and the number of iterations is kept at three as additional iteration has little effect on accuracy.

In this work, the regression block consists of an input layer with 615 nodes (576 features and 39 pose parameters), followed by two hidden layers with 288 neurons in each layer, and an output layer with 39 neurons. It is also important to insert dropout layers with a probability of 0.5 after the first and second layers to prevent overfitting.

### 3.2 3D Hand Model

The output parameters from the neural network are used by the 3D hand model MANO [17], to generate a triangulated hand mesh $M(\beta, \theta) \in \mathbb{R}^{3\times N}$, with $N = 778$ vertices. The underlying 3D joints $J(\beta, \theta) \in \mathbb{R}^{3\times K}$, where $K = 15$ joints, are obtained by linear regression from mesh vertices $M$.

MANO has been used in the majority of recent works on hand mesh recovery [1,3,10,20,22], as it offers simple control of the hand shape (finger length, palm thickness, etc.) and pose (3D rotation of the joints in axis-angle representation) with $\beta \in \mathbb{R}^{10}$ and $\theta \in \mathbb{R}^{3K}$ respectively.

However, pose $\theta \in \mathbb{R}^{45}$ contains redundant dimensions, resulting in infeasible hand pose (such as twisting of finger joint) if it is not constrained during the optimization process. This issue can be partially addressed by reducing the dimensionality of pose $\theta$ to $\theta_{PCA} \in \mathbb{R}^{10}$ [3] which is based on the Principal Component Analysis (PCA) of the pose database used to build MANO [17]. Nevertheless, $\theta_{PCA}$ may not be expressive enough and some works prefer the original pose representation, but manually define the pose limits [9] or impose geometric constraints [20].

**Joint Angle Representation:** Contrary to other methods, we propose a simple and effective joint angle representation $\alpha \in \mathbb{R}^{23}$ with a total of 23 degrees of freedom (DoF): four DoF for each finger and seven DoF for the thumb. In fact, joint angles have been used in other types of 3D hand model [15,21], where the rotation angles are bounded within a feasible range based on anatomical studies.

In order to maintain compatibility with MANO, we compute rotation matrices that transform our local joint angles to match MANO pose. By combining all the rotation matrices to form a sparse matrix $\mathbf{T} \in \mathbb{R}^{45\times 23}$, all the joint angles $\alpha \in \mathbb{R}^{23}$ can be mapped to MANO pose $\theta \in \mathbb{R}^{45}$ in a single step:

$$\theta = \mathbf{T}\alpha \tag{1}$$

The advantage of using joint angle representation is further discussed in Sect. 4.1 which compares the results of using $\alpha$, $\theta$ and $\theta_{PCA}$ representations.

**Camera Model:** The camera parameters $\{s, R, t\}$ represent the scaling $s \in \mathbb{R}^+$ in image plane, global rotation matrix $R \in SO(3)$ in axis-angle representation, and translation $t \in \mathbb{R}^2$ in image plane. A weak perspective camera model is used to project the 3D joints into the 2D image plane:

$$x = s\varPi(RJ(\beta, \theta)) + t, \tag{2}$$

where $\varPi$ is simply an orthographic projection to remove the dependency on camera intrinsics for supervising with 2D keypoint annotations.

### 3.3 Loss Functions

The loss function consists of three main terms:

$$\mathcal{L} = \lambda_{2D}\mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} + \lambda_{reg}\mathcal{L}_{reg} \tag{3}$$

where the hyperparameters $\lambda_{2D}$, $\lambda_{3D}$, and $\lambda_{reg}$, are empirically set to $10^2$, $10^2$ and $10^3$ respectively.

The first and second terms share a similar formulation to minimize the mean squared difference between the estimated 2D keypoints/3D joints and ground truth 2D/3D annotations:

$$\mathcal{L}_{2D/3D} = \frac{1}{n}\sum_{i=1}^{n} \|Estimated_i - Groundtruth_i\|_2^2 \tag{4}$$

where $n = 21$ includes the 15 hand joints $J$, with the addition of a wrist joint and five fingertips extracted from the mesh vertices $M$ [3].

The last term acts as a regularizer to prevent mesh distortion by reducing the magnitude of shape $\beta$, where $\beta = \mathbf{0} \in \mathbb{R}^{10}$ is the average shape. The joint angle $\alpha$ is also constraint within a feasible range of upper $U \in \mathbb{R}^{23}$ and lower $L \in \mathbb{R}^{23}$ joint angle boundaries [15]:

$$\mathcal{L}_{reg} = \|\beta\|_2^2 + \sum_{i=1}^{23} [max(0, L_i - \alpha_i) + max(0, \alpha_i - U_i)] \tag{5}$$

## 4 Experiments

**Datasets:** We evaluate our method on two publicly available real-world datasets: Stereo Hand Pose Tracking Benchmark (STB) [19] and FreiHAND dataset [23].

The STB dataset is commonly used to benchmark performance on 3D hand joint estimation from a single RGB image. Since the ground truth annotations are obtained manually, it only features a single subject posing in a frontal pose with different backgrounds and without object. Thus, this basic dataset serves as a useful starting point to test different variations of our proposed model. Following previous works [3,16,24], we split the dataset captured with a RealSense camera into 15k/3k for training/testing. To match the palm center annotation used in the STB dataset, we take the midpoint of MANO's wrist joint and middle finger metacarpophalangeal (MCP) joint as the palm center.

The FreiHAND dataset is the first dataset that includes both 3D hand pose and shape annotations based on MANO, which allows the evaluation of hand mesh reconstruction. It contains challenging hand poses with varied viewpoints, 32 different subjects, and hand-object interactions. There are a total of 130,240 and 3,960 images for training and testing respectively. To ensure consistent reporting of results, the evaluation is performed online through a centralized server where the test set annotations are withheld.

**Metrics:** To evaluate the accuracy of 3D hand pose estimation, we report two metrics: (i) 3D PCK: plots the percentage of correct keypoints below different threshold values; (ii) AUC: measures the area under the PCK curve. For evaluation on the STB dataset, the thresholds for PCK curve range 20 mm to 50 mm to allow comparison with previous works, but for the FreiHAND dataset, the threshold starts from 0 mm.

To evaluate the prediction of hand shape, we report two metrics: (i) Mesh error: measures the average endpoint error between corresponding mesh vertices; (ii) F-score: the harmonic mean of recall and precision at two distance thresholds 5 mm (fine scale) and 15 mm (coarse scale) [23].

Similar to previous works [13,23,24], a common protocol of aligning the prediction with ground truth is performed using the Procrustes transformation [8] to remove global misalignment and evaluate the local hand pose and shape.

**Implementation Details:** We implemented our models using PyTorch framework and the experiments are performed on a computer with a Ryzen 7 3700X CPU, 32GB of RAM, and an Nvidia RTX 2080 Ti GPU. The network is trained using an Adam optimizer with a learning rate of $10^{-3}$ and reduced to $10^{-4}$ after 50 epochs when training with the FreiHAND dataset. Using a batch size of 20, the training on the STB dataset takes around 1.5 h for 120 epochs and the training on the FreiHAND dataset takes around 30 h for 400 epochs. We also augment the data with random scaling (±10%) and translation (±20 pixels) for better generalization to unseen data.

**Table 2.** Self-comparison on joint angle, pose and PCA pose representations

| Method | Joint angle (23 DoF) | PCA pose (10 DoF) | PCA pose (23 DoF) | PCA pose (45 DoF) | Pose (45 DoF) |
|---|---|---|---|---|---|
| AUC ↑ | **0.994** | 0.991 | 0.992 | 0.982 | 0.972 |

**Table 3.** Quantitative result of 3D mesh reconstruction on FreiHAND dataset

| Authors (Publication) | Mesh Error (cm) ↓ | F-score at 5 mm ↑ | F-score at 15 mm ↑ |
|---|---|---|---|
| Kulon *et al.* (CVPR'20) [14] | **0.86** | **0.614** | **0.966** |
| Zimmermann *et al.* MANO CNN (ICCV'19) [23] | 1.09 | 0.516 | 0.934 |
| Ours | 1.31 | 0.439 | 0.902 |
| Boukhayma *et al.* (CVPR'19) [3] | 1.32 | 0.427 | 0.894 |
| Hasson *et al.* (CVPR'19) [10] | 1.33 | 0.429 | 0.907 |
| Zimmermann *et al.* MANO fit (ICCV'19) [23] | 1.37 | 0.439 | 0.892 |
| Zimmermann *et al.* Mean shape (ICCV'19) [23] | 1.64 | 0.336 | 0.837 |

### 4.1 Evaluation on STB Dataset

We compare our results with deep learning-based methods [3,4,12,16,18,22,24] and our PCK curve and AUC score (0.994) are comparable with most of the existing results as shown in Fig. 2. Although a few other works reported higher AUC score of 0.995 [1,20] and 0.998 [6], the results on STB dataset are saturated due to its relatively small size with a large number of similar frames [20,22].

Therefore, a recent method [22] did not include the STB dataset in training and to maintain a fair comparison with the method, we also provide an additional result that was trained on only FreiHAND dataset and evaluated on STB dataset. Our AUC score of 0.908 is also slightly better than 0.898 [22].

**Comparison of Joint Angle Representation:** We used the STB dataset to conduct a self-comparison on the use of joint angle, pose and PCA pose representations. As shown in Table 2, our proposed joint angle representation yields the highest AUC score, whereas over-parametrizing the hand pose with 45 DoF has a negative impact on the AUC score.

### 4.2 Evaluation on FreiHAND Dataset

Figure 2 and Table 3 show the result of 3D hand pose and shape estimation respectively. Our PCK curve and AUC score are also comparable with most of the existing results except for MANO CNN [23]. Furthermore, Kulon *et al.* [14] achieved the best results with the lowest mesh error as they proposed an additional dataset of hand action obtained from YouTube videos to train their network, whereas we only train the network using the FreiHAND dataset.

Additional qualitative results on the STB and the FreiHAND datasets are also provided in Fig. 3.

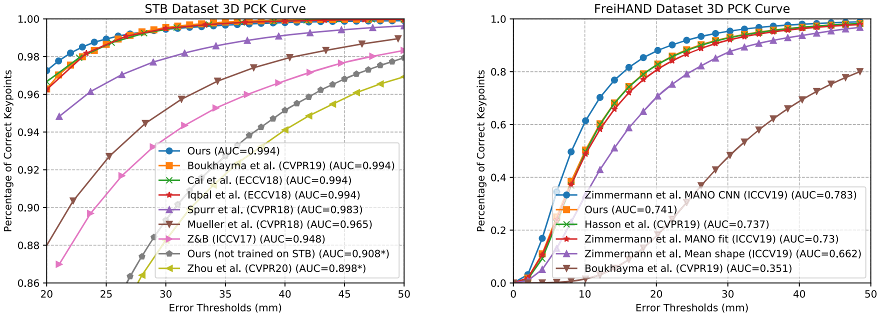

**Fig. 2.** (Left) Quantitative results on STB dataset, note that all the methods used the STB dataset for training, except for the last two methods with an "*" on the AUC score. (Right) Quantitative results on FreiHAND dataset.

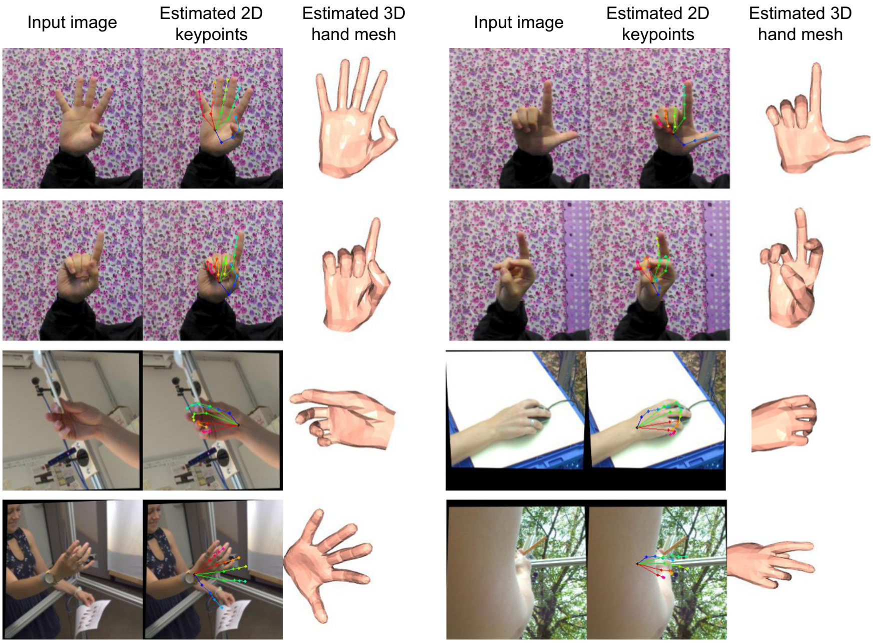

**Fig. 3.** Qualitative results: First two rows show the results on the STB dataset and the last two rows show the results on the FreiHAND dataset which contain hand-object interactions. The last row shows failure cases for challenging hand poses with the presence of another hand (bottom left) and extreme viewpoint where the hand is heavily occluded (bottom right).

## 5 Conclusion

In this paper, we present an efficient method to estimate 3D hand shape and pose that can achieve comparable accuracy against most of the existing methods, while the runtime of our method is the fastest on a GPU as well as a CPU. The proposed joint angle representation to articulate the hand model also helps to improve accuracy. Future works include increasing the robustness of the predictions and extending the method to run on mobile devices.

**Acknowledgments.** The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

## References

1. Baek, S., Kim, K.I., Kim, T.: Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In: CVPR, pp. 1067–1076 (2019)
2. Bazarevsky, V., Zhang, F.: On-device, real-time hand tracking with mediapipe. Google AI Blog, August 2019
3. Boukhayma, A., de Bem, R., Torr, P.H.S.: 3D hand shape and pose from images in the wild. In: CVPR, pp. 10835–10844 (2019)
4. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3D hand pose estimation from monocular RGB images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11210, pp. 678–694. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_41
5. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: CVPR, pp. 4733–4742 (2016)
6. Ge, L., et al.: 3D hand shape and pose estimation from a single RGB image. In: CVPR, pp. 10825–10834 (2019)
7. Gouidis, F., Panteleris, P., Oikonomidis, I., Argyros, A.A.: Accurate hand keypoint localization on mobile devices. In: MVA, pp. 1–6 (2019)
8. Gower, J.: Generalized procrustes analysis. Psychometrika **40**(1), 33–51 (1975)
9. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: HOnnotate: a method for 3D annotation of hand and object poses. In: CVPR, pp. 3193–3203 (2020)
10. Hasson, Y., et al.: Learning joint reconstruction of hands and manipulated objects. In: CVPR, pp. 11799–11808 (2019)
11. Howard, A., et al.: Searching for mobilenetv3. In: ICCV, pp. 1314–1324 (2019)
12. Iqbal, U., Molchanov, P., Breuel, T., Gall, J., Kautz, J.: Hand pose estimation via latent 2.5D heatmap regression. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 125–143. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_8
13. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR, pp. 7122–7131 (2018)
14. Kulon, D., Güler, R.A., Kokkinos, I., Bronstein, M., Zafeiriou, S.: Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: CVPR (2020)
15. Lim, G.M., Jatesiktat, P., Kuah, C.W.K., Ang, W.T.: Camera-based hand tracking using a mirror-based multi-view setup. In: EMBC, pp. 5789–5793 (2020)
16. Mueller, F., et al.: Ganerated hands for real-time 3D hand tracking from monocular RGB. In: CVPR, pp. 49–59 (2018)

17. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: modeling and capturing hands and bodies together. ACM TOG **36**(6) (2017)
18. Spurr, A., Song, J., Park, S., Hilliges, O.: Cross-modal deep variational hand pose estimation. In: CVPR, pp. 89–98 (2018)
19. Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., Yang, Q.: A hand pose tracking benchmark from stereo matching. In: ICIP, pp. 982–986 (2017)
20. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: ICCV, pp. 2354–2364 (2019)
21. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: IJCAI, pp. 2421–2427 (2016)
22. Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., Xu, F.: Monocular real-time hand shape and motion capture using multi-modal data. In: CVPR (2020)
23. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M.J., Brox, T.: Freihand: a dataset for markerless capture of hand pose and shape from single RGB images. In: ICCV, pp. 813–822 (2019)
24. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: ICCV, pp. 4913–4921 (2017)