Self-supervised Transfer Learning for Hand Mesh Recovery from Binocular Images

Zheng Chen Sihan Wang Yi Sun* Xiaohong Ma
Dalian University of Technology, China
{czheng,sylbia_w}@mail.dlut.edu.cn,{lslwf,maxh}@dlut.edu.cn

Abstract

Traditional methods for RGB hand mesh recovery usually need to train a separate model for each dataset with the corresponding ground truth and are hardly adapted to new scenarios without the ground truth for supervision. To address the problem, we propose a self-supervised framework for hand mesh estimation, where we pre-learn hand priors from existing hand datasets and transfer the priors to new scenarios without any landmark annotations. The proposed approach takes binocular images as input and mainly relies on left-right consistency constraints including appearance consensus and shape consistency to train the model to estimate the hand mesh in new scenarios. We conduct experiments on the widely used stereo hand dataset, and the experimental results verify that our model can get comparable performance compared with state-of-the-art methods even without the corresponding landmark annotations. To further evaluate our model, we collect a large real binocular dataset. The experimental results on the collected real dataset also verify the effectiveness of our model qualitatively.

1. Introduction

Hand mesh recovery from RGB images has always been an important research task in the field of computer vision. It has a wide range of applications, such as virtual reality, human computer interaction, robotics and so on. With the development of deep learning [6, 8] and the existence of some large hand datasets [19, 11], hand mesh recovery from RGB images has made great progress. However, a challenge of the hand mesh recovery is that many existing methods usually need a large labeled hand dataset [10, 5] for training. In the case of insufficient annotations or even in the most extreme case without any landmark annotations in unseen real scenarios, it is very difficult to obtain accurate hand mesh prediction results. In this paper, we are specifically interested in self-supervised hand mesh recovery to deal with the extreme case without any landmark annotations in unseen real scenarios.

It is very challenging to tackle this self-supervised mesh recovery task. Some previous methods generate synthetic datasets [5] for estimating the hand mesh coordinates. But the model trained on the synthetic hand dataset can hardly be adapted to the real data due to the domain gap, which limits their application in the real environment. Other methods more or less need landmark annotations for supervision. They propose several weakly-supervised methods and use 2D hand joints and/or depth map to train their network [1, 17, 4]. Since these approaches heavily rely on large-scale 2D pose annotations, they also have limited generalizability when applied to unseen images in new scenarios.

Unlike previous methods, our approach is based on the insight in cognitive science that human baby can adapt to a novel concept by correlating it with old concepts without receiving an explicit supervision. Therefore, we present a novel self-supervised learning approach that transfer the hand priors learned from past hand estimation tasks into new scenarios and regresses the network parameters of hand mesh prediction with no ground truth landmark available. In order to preserve certain past experience and adapt to a new unlabeled environment, our approach explores a form of self-supervised objective which learns hand mesh from easily accessible binocular images by left-right consistency without ground truth landmark.

The most crucial objective in this paper is to move away any kind of landmark supervision to improve generalizability in new scenarios. Since there are already some existing large labeled hand datasets [10, 5], our approach can make better use of these datasets to obtain the initial mesh estimate. For more accurate estimates in an unseen environment without any landmark annotations, we propose to learn from previous experience and regress the hand mesh vertices with binocular images by left-right consistency. Both the appearance consensus and shape consistency of self-supervised objectives are carefully designed. As will

*Corresponding author.

be shown in Section 4, our method can accurately recover dense meshes and achieves comparable results to weakly supervised and even some supervised methods. The main contributions of our paper can be summarized as follows:

1) We propose a self-supervised framework for hand mesh recovery. The model is pre-learned from an existing labeled dataset, and then is continually transferred to an unseen environment with binocular images by left-right consistency without any ground truth of 2D / 3D joint coordinates, depth maps, and mesh vertex coordinates.

2) We carefully design several self-supervised constraints to enforce the appearance consensus and the shape consistency. These constraints enable the model to dig into underlying spatial relations in binocular images, so as to generalize the model to new scenarios.

3) Taking binocular RGB images as input, our model can estimate the absolute hand mesh vertex coordinates which is especially useful in virtual reality and robot grasping where 3D absolute coordinates are required.

4) Our model achieves comparable performance with existing weakly supervised and even some supervised methods on the widely used stereo dataset. We also collect a large real binocular dataset and the experimental results on this dataset also verify the effectiveness of the proposed model.

2. Related work

There are already a large number of literatures on 3D hand pose estimation and shape recovery. In this paper, we only focus on the works related to weakly-supervised and self-supervised mesh estimation.

Weakly-supervised hand shape recovery. It is very difficult to manually label the 3D hand mesh vertex coordinates in the real dataset. Some works consider to estimate the hand mesh vertex coordinates with the supervision of hand joint coordinates and/or depth maps. Ge $et\ al.$ [4] use both depth map and 2D joint coordinates for estimating the hand mesh vertex coordinates in the real dataset. Boukhayma $et\ al.$ [1] use a ResNet-50 [6] to regress the shape and pose parameters of the MANO hand model [12]. Then, the 3D hand mesh is generated from the estimated parameters. Finally, they re-project the hand mesh to 2D hand joint coordinates for supervision. Zhang $et\ al.$ [17] also use MANO hand model for regressing the hand mesh vertex coordinates. They render the mesh to get hand mask and 3D joint coordinates and supervise them with ground truth. Zhou $et\ al.$ [18] capture the hand shape with multi-modal data. They train their model with the images from both synthetic and real datasets.

Self-supervision. There are few self-supervised approaches for estimating the hand mesh vertex coordinates. Chen $et\ al.$ [3] propose a temporal-aware self-supervised framework for estimating the 3D hand pose and mesh. They leverage the temporal consistency constraints for training their model. They also use 2D hand joint coordinates and hand mask for supervision to estimate the hand mesh vertex coordinates from the video. Wan $et\ al.$ [14] propose a self-supervised model for 3D hand pose estimation with hand depth map as input. They pre-define a hand model, which is approximated with 42 spheres, and estimate the spheres' center coordinates. Then they render the spheres with the estimations and fit these rendered spheres back to the depth map for fine-tuning the estimations.

Different from the above mentioned methods, we propose a self-supervised framework for hand mesh recovery without the supervision of 2D/3D hand joint coordinates, hand depth and mesh vertices. Our model can transfer the pre-learned hand priors to new scenarios by left-right consistency. Furthermore, our model can output absolute mesh coordinates, which has wider applications.

3. Method

Making better use of existing hand datasets and transferring the hand mesh prediction skill learned from these datasets to a new scenario can effectively alleviate the requirement of annotations in the new scenarios. Therefore, we present a novel self-supervised learning approach that regress network parameters of hand mesh prediction in the new scenarios with no ground truth landmark available by learning from the past hand estimation tasks. The diagram of the proposed approach is illustrated in Figure 1. We learn the hand priors from the existing large labeled hand dataset [19] with an encoder-decoder model (shown in Figure 2) and duplicate the model in our self-supervised framework when estimating hand mesh in new scenes with binocular images, as shown in the orange part of Figure 1. With the pre-learned model, we can obtain initial mesh estimates for the new scenarios. For more accurate estimates in an unseen environment without ground truth landmark, we propose to learn the mesh vertices by left-right consistency (especially photo-consistency constraint), shown in blue part of Figure 1. We introduce the learning pipeline of the hand prior on existing hand dataset in Section 3.1. Section 3.2 introduces how to transfer the learned hand prior to a new scenario by the proposed self-supervised framework and gives the details of the constraints used for self-supervised training.

3.1. Hand prior learning in previous dataset

There are already some existing large hand datasets, which contain abundant hand prior information. In this paper, we use the FreiHAND dataset [19] for hand prior learning, because it provides a large amount of hand data with sufficient viewpoint and hand pose variation. We design a simple encoder-decoder structure for hand prior learning as shown in Figure 2, which takes the RGB image as input

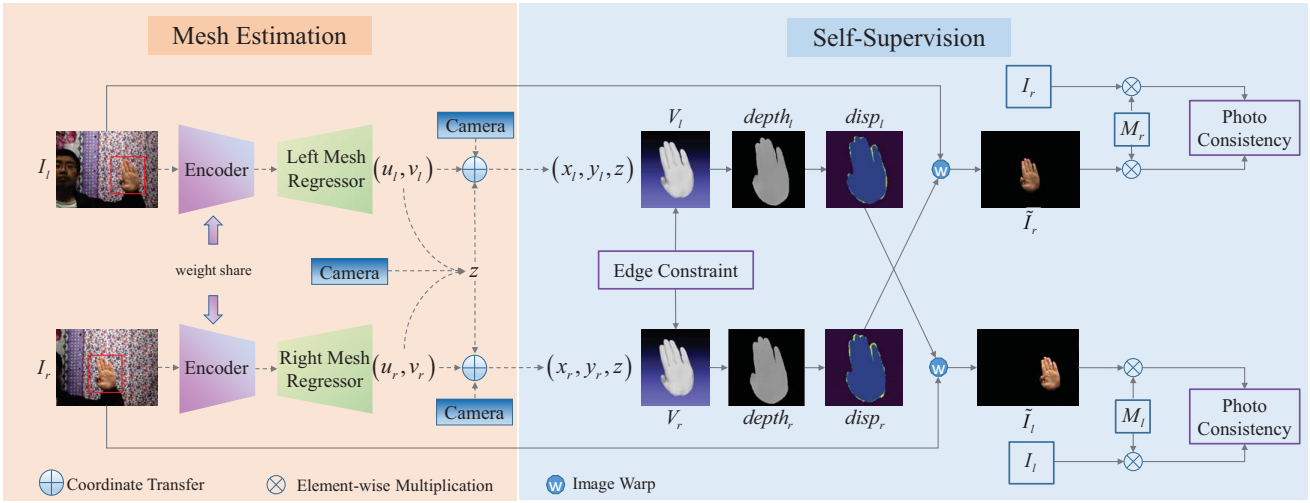

Figure 1. Our self-supervised framework.The left part is the mesh estimation module, which takes the binocular images as input and outputs the (u, v) coordinates of the hand mesh vertex. The encoder and the mesh regressor in the mesh estimation module are initialized by pre-training on the FreiHAND dataset [19]. The right part is the self-supervision module, where we mainly show the calculation process of the photo-consistency constraint.

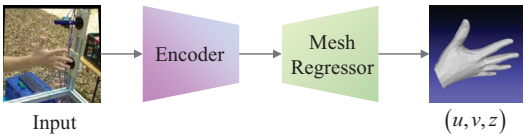

Figure 2. Encoder-decoder for hand prior learning.

and outputs the hand mesh vertex $V \in R^{N*3}$. N is the number of the hand mesh vertex. The detailed structure of the encoder-decoder can be seen in supplementary material. We use ground truth mesh vertex coordinates to supervise the learning process.

Directly applying the learned encoder-decoder to a new scenario (e.g. the STB dataset) will output inaccurate hand mesh estimations due to the domain gap. We give some examples of the FreiHAND dataset and the STB dataset in Figure 3 to illustrate the large domain gap between them. The first difference between the STB dataset and the FreiHAND dataset is the hand shape and pose. These two datasets are collected from different subjects performing various poses, thus the pre-learned model cannot give an accurate estimation for the pose unseen in the FreiHAND dataset, shown as the first two images in the third row of Figure 3. Another difference between these two datasets is the background and the illumination. The STB dataset is collected indoors and the samples usually have shaded regions on the hand due to the indoor lighting. The FreiHAND dataset is collected under 4 powerful LED lights, thus the illumination has little effect on hand. Therefore, the shaded hand regions on the STB dataset shown as the third and fourth images in the third row of Figure 3 can be easily misestimated. Moreover, the two datasets are collected with different sensors and their image styles are different. This also makes the pre-learned model output inaccurate mesh

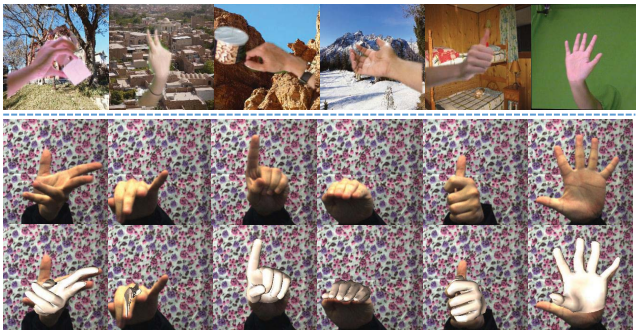

Figure 3. Results of direct prediction on the STB dataset [16] with the model pre-learned on the FreiHAND dataset [19]. The first row present the samples in FreiHAND dataset. The second row are the samples in the STB dataset. The third row present the mesh estimation results projected on the RGB image with the pre-learned model.

estimations on the STB dataset even with similar hand pose, shown as the last two columns of Figure 3. The aforementioned factors make it difficult for the learned model to get accurate prediction in a new scenario.

However, as shown in the third row of Figure 3, although the predicted results deviate from the real scene, they usually keep the hand shape undistorted, which indicates the pre-trained model has learned the hand priors from the FreiHAND dataset. Therefore, the keypoint is how to transfer the learned hand priors to new scenarios properly, especially without any ground truth landmark annotations in new scenarios. In this paper, we turn to binocular images and use the left-right consistency as supplementary information to adjust the pre-learned model in a self-supervised framework. The details of the self-supervision process is shown in the next subsection.

3.2. Self-supervised transfer learning in new scenarios

As discussed above, the model trained on the FreiHAND dataset shows poor performance when directly transferred to a new scene like the STB dataset. However, there are usually no ground truth landmark annotations in new scenarios for retraining the hand mesh estimation model. Therefore, we choose to add another view to provide more ground truth information for refining the pre-learned model in a new scenario. Since we use binocular images as input in this paper during self-supervised learning, we can calculate the z coordinates from disparities. The calculating process of z coordinate is as the following formula.

$$z = \frac{f_c * b}{|u_l - u_r|} \tag{1}$$

where f_c is the focal length and b is the baseline of the stereo camera. u_l and u_r denote the estimated u coordinates of the hand mesh in the left and right branch, respectively. Taking binocular images as input, the output hand mesh of the left and right branch have the same z coordinates.

As illustrated in Figure 1, we convert the predicted local coordinate (u, v, z) into the world coordinate (x, y, z) and train the self-supervised framework by enforcing photo-consistency between the left and right views [7]. The photo-consistency constraint in our model is shown in the blue part of Figure 1. Take the left branch as an example. The right branch is the same with the left one. We rendered the estimated right hand mesh V_r into depth map and then use it to calculate the disparity map $disp_r$ according to the following formula.

$$disp_r = \frac{f_c * b}{depth_r} \tag{2}$$

Finally, the left image I_l can be warped to the right image $\tilde{I}_r$ according to the estimated disparity $disp_r$. Note that we only warp the hand region and set the background to zero. After removing the background with ground truth mask M_r (obtained by existing skin detecting methods), the photo-consistency between the right image I_r and the warped right image $\tilde{I}_r$ can be used to train the network. Hence, the photo-consistency constraint in fact corrects the positions of all the frontal visible mesh vertices, which is a powerful cue to guide the prediction for better generalization in unseen scenarios. The photo-consistency loss can be formulated as follows:

$$\begin{aligned} L^l_{photo-cons} = \frac{\alpha}{2}\left(1 - SSIM\left(\left(\tilde{I}_r, I_r\right) * M_r\right)\right) \\ + (1-\alpha)\left\|\left(\tilde{I}_r, I_r\right) * M_r\right\| \end{aligned} \tag{3}$$

where, $\tilde{I}_r$ is the warped RGB image, I_r is the original right RGB image. α is weight coefficient. In this paper we use SSIM [15] to calculate the similarity between I_r and $\tilde{I}_r$. The SSIM is calculated with the following formula:

$$SSIM\left(\tilde{I}_r, I_r\right) = \frac{\sigma_{\tilde{I}_r, I_r} + c}{\sigma^2_{\tilde{I}_r} + \sigma^2_{I_r} + c} \tag{4}$$

where, $\sigma^2_{\tilde{I}_r}$ is the variance of the warped RGB image $\tilde{I}_r$. $\sigma^2_{I_r}$ is the variance of the right RGB image I_r. $\sigma_{\tilde{I}_r, I_r}$ is the covariance between $\tilde{I}_r$ and I_r. c is a constant.

However, the photo-consistency loss has no constraint on the occluded part of the hand mesh, and without a hand model or hand shape constraint, the output mesh may be distorted. By observing the pre-trained results in Figure 3, we can see that the results directly tested on the STB dataset, though not very accurate, preserve a reasonable hand shape. We use them as pseudo mesh labels to constraint hand shape, which is named as edge loss in Figure 1. Apart from it, we also add Laplacian loss to keep the local surface of the output mesh smooth. Here, we only give the left branch for illustration and the right branch is same with it. The edge loss is formulated as follows:

$$L^l_{edge} = \sum_{i=1}^{E} \left(\|e_i\| - \|\tilde{e}_i\|\right)^2 \tag{5}$$

where, e_i is the edge of the pseudo-ground truth hand mesh of the left branch.$\tilde{e}_i$ is the estimated hand mesh edge of the left branch in our self-supervised model during the training process. The Laplacian loss is formulated as follows:

$$L^l_{lap} = \frac{1}{V_l} \sum\nolimits_{v=1}^{V_l} \left(v - \frac{1}{\|N(v)\|} \sum\nolimits_{v' \in N(v)} v'\right) \tag{6}$$

where, $N(v)$ denotes the neighbor vertices of vertex v.

To help the model get accurate boundary of hands in new scenes, we further add the mask constraint in our model. The mask constraint is formulated as follows, where we only give the mask constraint used in the left branch for illustration and the right branch is same with it.

$$L^l_{mask} = \left\|M_l - \tilde{M}_l\right\|^2 \tag{7}$$

where, $\tilde{M}_l$ is the rendered hand mask with the estimated hand mesh and M_l is the ground truth.

Finally, we introduce the Chamfer Distance loss and the normal loss to ensure spatial structure consistency between the left and right branch. The Chamfer Distance loss is formulated as follows:

$$\begin{aligned} L_{chamfer} = \tfrac{1}{N}\Big(\sum_{v_l \in V_l} \min_{v_r \in V_r} \|v_l - v_r\|_2^2 \\ + \sum_{v_r \in V_r} \min_{v_l \in V_l} \|v_r - v_l\|_2^2\Big) \end{aligned} \tag{8}$$

where, v_l and v_r denote hand vertex coordinates of the left estimated hand mesh V_l and right estimated hand mesh V_r,

respectively. N is the number of the hand mesh vertex. The normal loss is formulated as follows:

$$L_{normal} = \sum_{f} \sum_{\{i,j\} \subset f} \left| \left\langle \frac{v_i^l - v_j^l}{\left\| v_i^l - v_j^l \right\|_2}, n_f^r \right\rangle \right| \quad (9)$$

where, f is the face of the hand mesh. n_f^r is the unit normal vector of face f of the right hand mesh. v_i^l and v_j^l denote the ith and the jth vertex coordinates of the left hand mesh, respectively.

Overall, the total loss for training our self-supervised framework is defined as follows:

$$\begin{aligned} L_{total} = \textstyle\sum_{k \in \{l,r\}} (&\lambda^k_{photo-cons} L^k_{photo-cons} \\ &+ \lambda^k_{edge} L^k_{edge} + \lambda^k_{lap} L^k_{lap} \\ &+ \lambda^k_{mask} L^k_{mask}) \\ &+ \lambda_{chamfer} L_{chamfer} + \lambda_{normal} L_{normal} \end{aligned} \quad (10)$$

where, l and r denote the left branch and right branch, respectively. $\lambda_{photo-cons}$, λ_{edge}, λ_{lap}, λ_{mask}, $\lambda_{chamfer}$ and λ_{normal} are the corresponding weight coefficients. In this paper, we set $\lambda_{photo-cons} = 100$, $\lambda_{edge} = 1$, $\lambda_{lap} = 10$, $\lambda_{mask} = 0.1$, $\lambda_{chamfer} = 0.001$ and $\lambda_{normal} = 0.001$.

4. Experiments

4.1. Dataset and evaluation metric

Dataset. We conduct experiments on two widely used hand datasets, including the FreiHAND dataset [19] and the Stereo Hand Pose Tracking Benchmark (STB) [16].

The FreiHAND dataset contains 130K training samples and 4K testing samples. We only use the training samples for pre-training the encoder-decoder model, which is used to initialize the weight of our self-supervised framework.

The STB hand dataset consists of six sequences with different backgrounds collected from a stereo sensor. It provides the ground truth of 21 3D hand joint coordinates. The baseline and focal length of the stereo sensor are 120.1mm and 822.8 pixels, respectively. Following [4, 2], we also use one sequence for testing and other sequences for training.

To further evaluate our self-supervised framework, we collect a large number of stereo images to evaluate the proposed model qualitatively. Details and the qualitative results on our dataset are shown in Section 4.4.

Metrics. We adapt two commonly used metrics to verify the validity of our method, including: 1) the average error between the estimated 3D hand joints and the ground truth. 2) the percentage of correct keypoints (PCK) which the Euclidean error is below a threshold.

4.2. Results of our self-supervised approach

Evaluations of self-supervised training. To evaluate the effectiveness of our framework, we compare the results on the STB dataset before and after self-supervised training. The quantitative results of our self-supervised training is shown in Table 1. *Test-directly* means training on the FreiHAND dataset [19] and directly testing on the STB dataset, which get 3D mean joint error of 44.52mm. After self-supervised training, denoted as *self-supervised (binocular)*, the 3D mean joint error significantly decreases from 44.52mm to 11.14mm.

Model	Mean Joint Error (mm)
Test-directly	44.54
Self-supervised (monocular)	12.66
Self-supervised (binocular)	**11.14**

Table 1. The experimental results of our self-supervised training on the STB dataset [16].

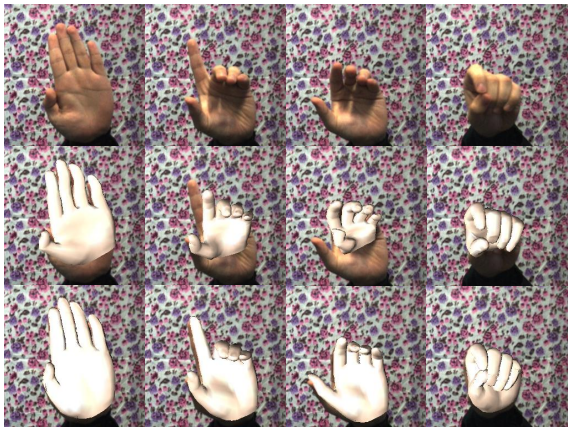

Figure 4. Comparison of the experimental results on the STB dataset [16] before and after self-supervised training with our model. The first row is the input RGB image. The second row is the results of directly test on STB dataset with pre-trained model on FreiHAND dataset. The third row is the results after our self-supervised training with the proposed model (binocular version).

In addition, in Figure 4 we compare the qualitative results before and after our self-supervised training. Comparing the second row with the third one, we can see that our self-supervised framework not only keeps a reasonable hand shape but also corrects the pose and location of the estimated mesh. We provide more qualitative results of the hand mesh estimations with our self-supervised model on the STB dataset in Figure 5. The first row and the third row give the original input images. The second row and the fourth row give the estimated hand meshes, which are rendered on the input RGB image. From the experimental results we can see that our model can overcome the uneven light condition and get accurate hand mesh estimation results for hands with different poses.

Apart from the binocular input which calculates z coordinates with the disparity, we can also regress z coordinates in each branch with the neural network, denoted as *self-supervised (monocular)* in Table 1. Our monocular version has similar structure as the binocular version, which

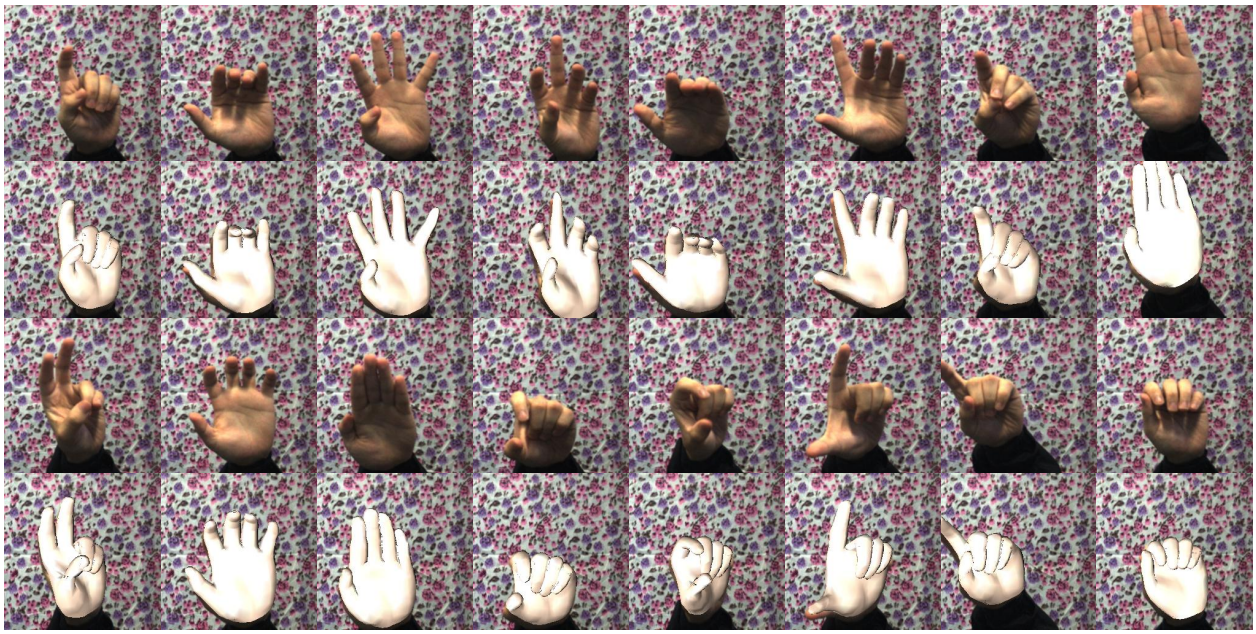

Figure 5. Qualitative results on the STB dataset [16]. The first row and the third row are the input images. The second row and the fourth row are the estimated hand mesh rendered on the input images.

also contains two branches and takes left and right images as input during training. The mesh regressor in each branch is built upon that of the binocular version by adding another network module for regressing the z coordinates. We also use the left-right consistency in training as the binocular version. During testing, each branch can directly output u, v and z coordinates of the hand mesh and gets 3D mean joint error of 12.66mm, which is slightly worse than the binocular input. Furthermore, the model with binocular input can directly estimate the absolute 3D hand joint coordinates, while the monocular version can only estimate the root relative coordinates.

Overall, the above experiments verify the effectiveness of our self-supervised framework.

Ablation study. We evaluate the effect of each loss function used in our self-supervised framework, and the quantitative results are shown in Table 2. Each row in Table 2 gives the result when training our self-supervised framework without the corresponding loss. We also visualize the estimated mesh results in Figure 6. From Table 2 we can see that the photo-consistency loss plays the most important role in hand mesh recovery and the 3D mean joint error increases from 11.14mm to 25.36mm when such loss is removed. This is because the photo-consistency loss exploits the implicit correspondence of the hand between binocular images and holds the most ground truth spatial information compared to other constraints. As illustrated in Figure 6, we convert the ground truth depth maps of the STB dataset into point clouds (the yellow points) and draw them together with the estimated mesh. In this way we can see that the hand mesh learned with full losses tend to fit the ground truth depth map well, while training without the photo-consistency loss could lead to inaccurate estimation. Secondly, when the edge loss is removed during self-supervised training, the 3D mean joint error increases from 11.14mm to 13.46mm, as shown in the third row in Table 2. The visualization results illustrated in the last two columns in Figure 6 show that training without the edge loss could lead to severe shape distortion, since there are no hand model used in our

Photo-consistency	Edge	Laplacian	Mask	Chamfer	Normal	Mean Joint Error(mm)
✓	✓	✓	✓	✓	✓	**11.14**
×	✓	✓	✓	✓	✓	25.36
✓	×	✓	✓	✓	✓	13.46
✓	✓	×	✓	✓	✓	11.36
✓	✓	✓	×	✓	✓	11.23
✓	✓	✓	✓	×	✓	11.37
✓	✓	✓	✓	✓	×	11.23

Table 2. Impact of each loss function on the STB dataset [16].

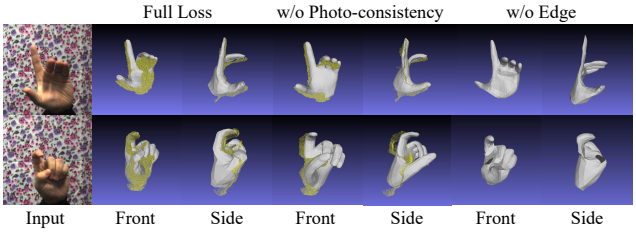

Figure 6. Qualitative results without the photo-consistency constraint and the edge constraint. The hand point cloud (the yellow points) converted from the ground truth hand depth map is illustrated for better comparison.

Model	Sensor	Absolute	Supervision	Mean Error(mm)
Ge [4]	monocular	no	2D joint & depth	10.57
Boukhayma [1]	monocular	no	3D joint	9.76
Spurr [13]	monocular	no	3D joint	8.56
Ge [4]	monocular	no	3D joint	6.37
Chen [3]	monocular(video)	no	2D joint	11.3
Yuncheng Li [9]	binocular	yes	2D joint	24.6
Zhang [16]	binocular	no	3D joint	-
self-supervised(ours)	**monocular**	**no**	**w/o 2D/3D joint & depth**	**12.66**
self-supervised(ours)	**binocular**	**yes**	**w/o 2D/3D joint & depth**	**11.14**

Table 3. Comparison of the 3D hand pose estimation results between our model and existing state-of-the-art methods on STB dataset [16].

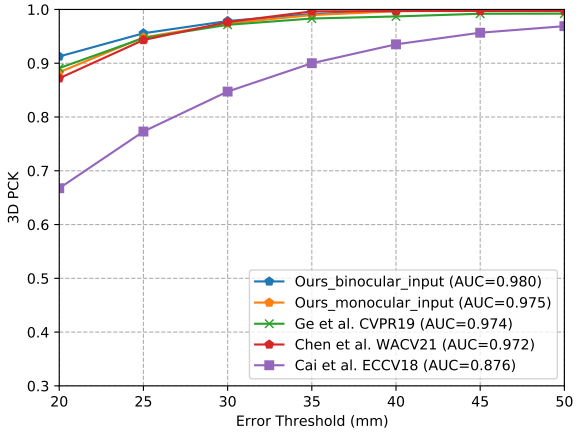

Figure 7. Comparison with state-of-the-art weakly supervised methods of the 3D PCK on STB dataset [16].

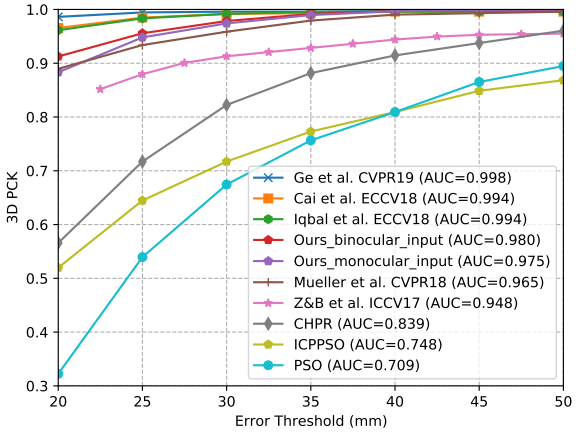

Figure 8. Comparison with state-of-the-art supervised methods of the 3D PCK on STB dataset [16].

framework, and other constraints do not preserve complete hand shape information as the edge loss does. The Laplacian loss, the mask loss, the Chamfer Distance loss and the normal loss also provide slight effect to our self-supervised model for 3D hand joint estimation, as shown in Table 2. But because the photo-consistency loss and the edge loss have already provide powerful pose and shape constraints, the effects of other losses are not obvious. In summary, our self-supervised model with full loss functions achieves the lowest 3D mean joint error.

4.3. Comparison with state-of-the-art methods

We only compare the 3D hand joint estimation result of our model with existing state-of-the-art methods, because the STB hand dataset only has the 3D joint ground truth. The comparison of the 3D hand pose estimation results between our model and existing state-of-the-art methods are shown in Table 3. From the experimental results we can see that our model achieves better performance than [9] when estimating the absolute 3D hand joint coordinates with binocular images as input. Compared with the weakly supervised method [4] with 2D hand joint coordinates and hand depth map for supervision, whose 3D mean joint error is 10.57mm, our self-supervised model can get comparable performance. Our model also presents comparable performance with [3], which uses the hand videos as model input and 2D hand joint coordinates for supervision. From the experimental results we can see our self-supervised framework can get comparable performance compared with existing state-of-the-art weakly supervised methods. It should be pointed out that our model doesn't need 2D/3D joint ground truth for supervision on STB hand dataset. Another advantage of our self-supervised framework is that our model can estimate the absolute hand joint coordinates, while existing methods usually can only estimate the root relative hand joint coordinates.

The PCK curve compared with state-of-the-art weakly supervised methods [4, 3, 2] is shown in Figure 7. It can be seen that our self-supervised framework with binocular images as input performs better than these weakly supervised methods. Our self-supervised model with monocular image as input also gets comparable performance with these methods. It should be noted that, [4, 2] use 2D hand joint coordinates and hand depth map for supervision. [3] use 2D hand joint coordinates for supervision with hand video as input. Our model doesn't need the supervision of the 2D hand joint coordinates and the hand depth on the STB hand dataset. Figure 8 gives the comparison results between our model and the state-of-the-art supervised methods, which use the 3D joint coordinates for supervision. From the experimental results we can see our model can also get comparable performance with these methods.

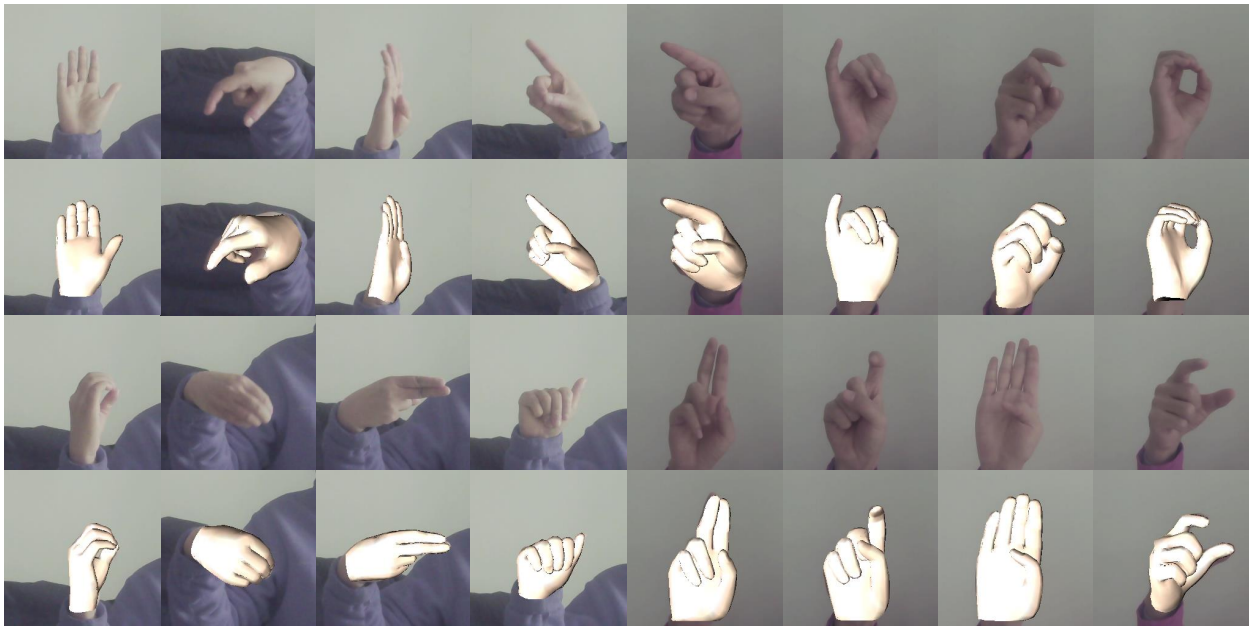

Figure 9. Qualitative results on our collected binocular images. The first row and the third row are the input images. The second row and the fourth row are the estimated hand mesh rendered on the input images.

4.4. Qualitative results on the collected real dataset

We use the D1000-IR-120 MYNT AI stereo sensor to collect the binocular images. The baseline of the binocular sensor is 120mm and the focal length is 762.7 pixels. The binocular images are collected from two subjects performing counting, random, and signs from the American sign language with size of 1280×720. The total number of the collected binocular images are 12150, where we randomly select 11000 binocular images for training the self-supervised model and the rest 1150 binocular images are used for testing the model qualitatively.

Figure 9 gives the experimental results on our collected binocular images. From the results we can see that our model can predict accurate and reasonable estimation results of the hand mesh for various hand poses without any landmark annotations. We provide more qualitative results on the STB dataset and our collected real images in the supplementary materials by videos.

Failure cases. Figure 10 gives some failure cases of the estimated hand mesh by our self-supervised model. The reason of the failure in the first two columns may be the insufficiency of pose in the FreiHAND dataset during hand prior learning. We will add more datasets for learning the hand prior in the future. The reason of the failure in the last two columns shown as the red circles is that we use binocular images as model input and the baseline of the stereo camera is short. The short baseline makes some parts of the hand region similar between the left and right image, which are difficult to learn. This will lead to inaccurate mesh estimation of these hand parts.

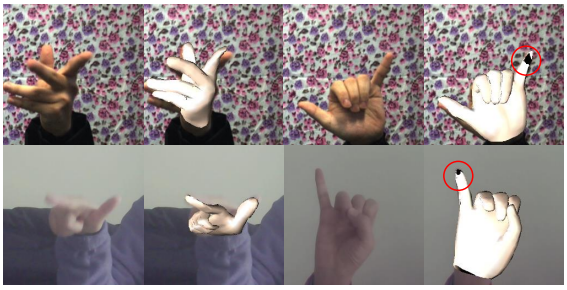

Figure 10. Failure cases on STB dataset and the collected real images. The first row gives the failure cases on STB dataset. The second row are the results on the collected real images.

5. Conclusion

In this paper, we propose a self-supervised transfer learning framework for hand mesh recovery. Different from existing models, which usually need the ground truth landmark for supervision in new scenarios, our self-supervised framework learns hand prior from existing hand datasets and transfer the priors to new scenarios with binocular images. We apply the left-right consistency constraints for training the proposed model without ground truth landmark and the experimental results on the stereo hand dataset show that our model can get comparable performance compared with state-of-the-art methods even without landmark annotations.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (No. U1708263, 61873046).

References

[1] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[2] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[3] 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1050–1059, 2021.

[4] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[9] Yuncheng Li, Zehao Xue, Yingying Wang, Liuhao Ge, Zhou Ren, and Jonathan Rodriguez. End-to-end 3d hand pose estimation from stereo cameras. In *BMVC*, page 161, 2019.

[10] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. *arXiv preprint arXiv:2008.03713*, 2020.

[11] 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. *arXiv preprint arXiv:2008.09309*, 2020.

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

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

[14] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2019.

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[16] 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.

[17] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019.

[18] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2020.

[19] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 813–822, 2019.