RealisticHands: A Hybrid Model for 3D Hand Reconstruction

Michael Seeber[1] Roi Poranne[2] Marc Polleyfeys[1,4] Martin R. Oswald[1,3]
[1]ETH Zurich [2]University of Haifa [3]University of Amsterdam [4]Microsoft

Abstract

Estimating 3D hand meshes from RGB images robustly is a highly desirable task, made challenging due to the numerous degrees of freedom, and issues such as self-similarity and occlusions. Previous methods generally either use parametric 3D hand models or follow a model-free approach. While the former can be considered more robust, e.g. to occlusions, they are less expressive. We propose a hybrid approach, utilizing a deep neural network and differential rendering based optimization to demonstrably achieve the best of both worlds. In addition, we explore Virtual Reality (VR) as an application. Most VR headsets are nowadays equipped with multiple cameras, which we can leverage by extending our method to the egocentric stereo domain. This extension proves to be more resilient to the above mentioned issues. Finally, as a use-case, we show that the improved image-model alignment can be used to acquire the user's hand texture, which leads to a more realistic virtual hand representation.

1. Introduction

Hand pose and shape estimation from images are long standing problems in computer vision. Both are fundamental components in making mixed reality devices more immersive and accessible. As mixed reality headsets are becoming ubiquitous, so are hands as a primary input device, replacing the old and clumsy controllers of the past. However, in order to create the illusion of mixed reality, high degrees of performance and fidelity are necessary.

Our goal is to enable fast and accurate hand pose and shape estimation, including the hand texture too, in order to ultimately create a better sense of body ownership in virtual reality. However, while both accuracy and processing speed are desirable in general, they are notoriously difficult to attain together. In addition, small reconstruction errors are visibly amplified when textures are involved.

Previous work can be divided into model-based and model-free approaches. While model-based methods are generally more robust to e.g. occlusions, they tend to be slower and sensitive to initialization. Additionally, they are limited by the representation power of the underlying model, and cannot faithfully represent the whole diversity of hand shapes. In contrast, model-free methods require vast amounts of data and many one-shot predictions often lead to poor image-model alignments. Some methods rely on depth images, but with the emergence of deep learning, RGB images as a single input modality are becoming more popular, as such methods do not require special hardware. Generally, this task is much more challenging, due to self-occlusions, finger self-similarity, and depth ambiguities.

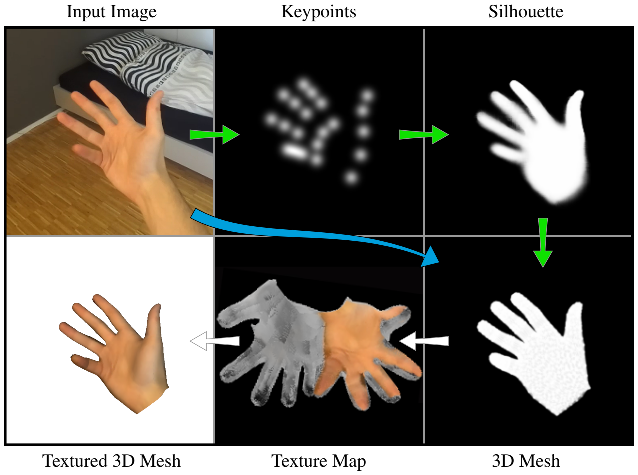

Figure 1. Given a single RGB image of a hand, our methods produces a faithful, textured 3D hand mesh. Our approach is hybrid between a model-based approach (green path) and a model-free approach (blue path).

We propose a hybrid approach that combines the robustness of model-based methods with the expressiveness of model-free methods. We use a deep learning approach to quickly and robustly obtain predictions, and a test-time optimization to further refine the result. This enables easy texture extraction based on a straightforward projection. In addition, we provide a stereo extension, making the method more robust, particularly for egocentric viewpoints.

To summarize, our *contributions* are: **(1)** Accurate 3D mesh estimation, fusing input modalities using a novel training strategy and improving resiliency by leveraging stereo views. **(2)** Hand segmentation and fine grained fit optimization of the predicted MANO mesh via differential rendering. **(3)** Personalized hand mesh rig including camera captured hand texture.

2. Related Work

3D Hand Pose Estimation is the process of recovering the joints of a 3D hand skeleton, mainly from RGB and/or depth cameras. Early methods [1] used low-level visual cues such as edge maps and Chamfer matching [2] to rank the likelihood of possible predefined poses. The advent of low-cost depth cameras paved the way for unconstrained 3D hand pose estimation and research focused on using depth data as input, with earlier methods based on optimization [30, 34, 35]. Later, works such as [39, 31] were able to replace depth by stereo cameras to a certain extent.

Convolutional Neural Networks (CNN) are a natural choice for pose estimation. A popular approach is to predict keypoint joint locations. Earlier methods, e.g. [29], used depth images as input and processed them with 2D CNNs. Others argue that 2D CNNs are too limited, due to their lack of 3D spatial information [10, 11, 26, 9]. Normal hand motions exhibits a large range of viewpoints for a given articulation, which makes it difficult to define a canonical viewpoint. To overcome this, various intermediate 3D representations have been studied (see e.g. [10]). Other methods employed D-TSDF [11], voxels [26] or point sets [9].

More recent approaches can recover hand poses from RGB images only. This is a significantly harder challenge due to the problem ill-posedness without depth information, but that has become feasible nonetheless thanks to learning based formulations e.g. [41]. One of the limiting factors however, is the massive amount of annotated data required in order to resolve the inevitable depth ambiguities. Mueller *et al*. [28] used generative methods to translate synthetic training data into realistic training images. Due to the large discrepancy between factors of variation ranging from image background and pose to camera viewpoint, Yang *et al*. [37] introduced disentangled variational autoencoders to allow specific sampling and inference of these factors. Another approach to deal with the lack of annotated training data is by utilizing a depth camera for weak supervision of the training procedure [5].

3D Hand Reconstruction. While pose estimation yields a skeletal representation of the hand, hand reconstruction aims to recover the surface geometry of the hand. For the more general full body reconstruction problem, multiple approaches have been proposed, such as 3D supervised learning methods (e.g. [13]), or model-based deformable surface methods, like SMPL [25, 16, 20]. Inspired by the success of SMPL, a similar parameterized model for hands known as MANO was proposed in [32], and was followed up and used in many publications, such as [3, 40, 14, 6].

Although MANO and such are able to represent a large variety of hand shapes, as a model-based approach, they are still limited in their expressiveness. Model-free methods like [12, 8] , use a graph CNNs to directly regress vertex positions in a coarse-to-fine manner. [21] relied on a spiral operator to construct spatial neighborhoods. Another approach to obtain vertices directly was recently proposed by Moon *et al*. [27], where they introduce a novel Image-to-Lixel (I2L) prediction network which retains the spatial relationship between pixels in the input image. Additionally, to cope with the limited annotated data available also self-supervised methods [7] have been studied.

3. Method

Overview. Our method estimates a textured 3D hand mesh from monocular or stereo inputs. We go beyond simply estimating the hand model parameters only, by having an additional differential rendering-based optimization step for fine grained adjustments. This makes our approach more expressive compared to solely relying on the MANO [32] hand model and also leads to more accurate image-model alignments compared to other one-shot prediction methods. This further enables us to rely on projections of the camera image to yield accompanying textures for a more personalized digital hand replication. Furthermore we extend our method towards a stereo setting to cope better with occlusions an ambiguities and thus improve robustness.

3.1. Architecture

Our proposed network architecture for the monocular setting is illustrated in Fig. 2. In short, inputs are fed into our deep encoder-decoder network to regress MANO and camera parameters and to predict a segmentation mask. Then, the obtained hand mesh is optimized via differential rendering, such that the error between the segmentation mask and the rendered silhouette is minimized.

Input. The inputs consist of an RGB hand crop and hand keypoints encoded in a 21 channel heatmap, where each channel represents a joint. We obtain them using MediaPipe Hands [38], a lightweight hand joint detection library. The 2D joint locations are used to compute the bounding box for the hand crop *and* the keypoint heatmap by encoding each joint location by a 2D Gaussian distribution.

Encoder. We require an architecture with high representation capability, which is also computationally cheap enough for our targeted application. Thus, we base our encoder on the ResNet50 architecture [15]. Since we have an RGB image *and* keypoint heatmaps, we extend the ResNet to support both modalities and included a learned implicit modality fusion. We do this by dividing the network at the third ResNet layer into a front section and a tail section. Then, we duplicate the head section for both inputs and adapt them to the corresponding input channel sizes. To fuse the modalities in a sensible way, we add a fusion block that is inspired by the self-supervised model adaption block introduced in [36]. The idea behind the fusion block is to adap-

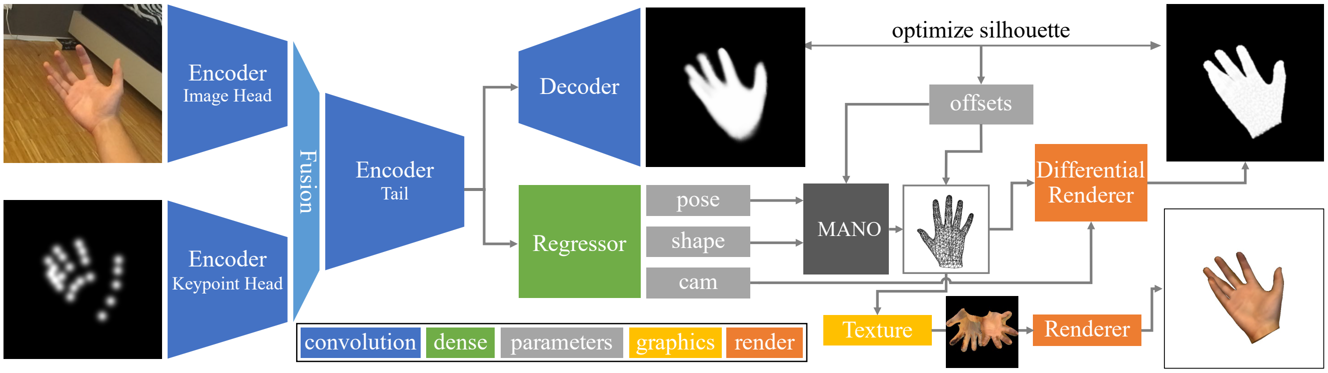

Figure 2. **Overview of the architecture for a monocular camera setting.**

tively recalibrate and fuse encoded feature maps according to spatial location. In other words, it learns to explicitly model the correlation between two feature maps, to selectively emphasize more informative features from a respective modality while suppressing the others. This is achieved by concatenating the feature maps of each modality: image $\boldsymbol{X}^{img} \in \mathbb{R}^{C\times H\times W}$ and keypoints $\boldsymbol{X}^{kp} \in \mathbb{R}^{C\times H\times W}$ to $\boldsymbol{X}^{img|kp} \in \mathbb{R}^{2C\times H\times W}$. Afterwards, $\boldsymbol{X}^{img|kp}$ is passed through a bottleneck, where the first ReLU activated convolution reduces the channel dimensionality by a ratio of 8 to $2C/8 = 128$. Subsequently, the dimensionality of the features is increased back to their original size by the second convolutional layer. This time, a sigmoid activation is used to scale the values to the $[0, 1]$ range, so that they represent weights. The resulting output w is used to emphasize and de-emphasize different regions in the originally concatenated features $\boldsymbol{X}^{img|kp}$ by taking the Hadamard product, i.e. performing element wise multiplication of the obtained weights and the concatenated features, which corresponds to $\hat{\boldsymbol{X}}^{img|kp} = w \circ \boldsymbol{X}^{img|kp}$. As a last step, the adaptive recalibrated feature maps $\hat{\boldsymbol{X}}^{img|kp}$ are passed through a final convolution layer in order to reduce the channel depth and yield the fused output with the correct dimensionality, so that it can be further processed by the tail of the encoder to obtain the base features, of which the decoding components of our method make use.

Decoder. The decoder part of our architecture is composed out of two components, a MANO regressor and a segmentation decoder. The structure of the MANO regressor is straightforward and consists of dense layers only. The obtained base features $\boldsymbol{X}^{base} \in \mathbb{R}^{2048}$ from the encoder are flattened and processed by two fully connected layers with respective sizes 2048 and 512, from which the outputs are regressed. We predict the paramerters of the MANO hand model, which consist of pose $p \in \mathbb{R}^{48}$ and shape $s \in \mathbb{R}^{10}$. The pose parameters correspond to the global rotation and joint rotations in angle axis representation. The shape is determined through 10 PCA components that encode the captured hand shape variance of the MANO hand scans [32]. By feeding p and s into the MANO model we obtain an articulated and shape adapted hand mesh $\in \mathbb{R}^{778\times 3}$. Lastly, we obtain weak-perspective camera parameters $c = (t, \delta) \in \mathbb{R}^3$, where $t \in \mathbb{R}^2$ is the 2D translation on the image plane and $\delta \in \mathbb{R}$ is a scale factor. This allows to project a posed 3D hand model back onto the input image and obtain absolute camera coordinates.

The segmentation decoder is tasked with attributing every pixel in the input hand crop to either 'hand' or 'background'. Like in [24, 33], we use an adapted U-Net with lateral connections between the contracting encoder and the successive expanding convolutional layers of the decoder. The first skip connections use the RGB branch of the encoder, as it contains more expressive segmentation features, but the segmentation decoder can benefit from the fused keypoint modality through deeper levels.

3.2. Test-time Refinement

Previous work show that one-shot MANO parameter regression methods [3, 40, 14], although fast and robust, often have poor image-mesh alignment. This is attributed to the pose being defined as relative joint rotations. Therefore, minor rotation errors accumulated along the kinematic chain can result in noticeable drifts from 2D image features. To mitigate this problem, we propose a test time optimization strategy. The idea is to iteratively refine the hand mesh to better align with the input image during test-time, by adding offsets $\Delta p, \Delta s$ and Δv for the pose and shape and vertices respectively. Let $\boldsymbol{M}$ be the MANO model, the optimized mesh $\boldsymbol{H}$ at iteration step $t + 1$ is then

$$\boldsymbol{H}_{t+1} = \boldsymbol{M}\left(p + \Delta p_t, s + \Delta s_t\right) + \Delta v_t \qquad (1)$$

Similarly, we update c by $c_{t+1} = c + \Delta c_t$. All offsets $\Delta P_t = (\Delta p_t, \Delta s_t, \Delta v_t, \Delta c_t)$ are obtained from the gradients provided by a differential renderer, aiming to reduce the discrepancy between the predicted segmentation and the rendered silhouette of $\boldsymbol{H}_t$, i.e. make the rendered mesh silhouette S_{mesh} more similar to the target silhouette image S_{target} obtained from the segmentation decoder.

We use SoftRas [23] as differential renderer, which aggregates probabilistic contributions of all mesh triangles with respect to the rendered pixels. This formulation allows to propagate gradients to occluded and far-range vertices too. We use stochastic gradient descent with the learning rate $\eta = 0.002$ and momentum $\alpha = 0.9$, and update the offsets by $\Delta P_{t+1} = \eta \nabla \mathcal{L}(H_t) + \alpha \Delta P_{t+1}$, where $\mathcal{L}(H)$ is described in the following.

Refinement Loss. The loss $\mathcal{L}$ is a weighted sum of an *image silhouette loss* and several regularization terms that are commonly used and thoroughly studied in literature. First, the image silhouette loss is computed as the squared L_2 distance between the predicted silhouette and the target. Namely $\mathcal{L}_{\text{sil}} = \|S_{\text{mesh}} - S_{\text{target}}\|_2^2$ where S_{mesh} is the rendered silhouette and S_{target} the predicted segmentation mask. Additionally, we minimize the vertex offsets to jointly optimize the pose and the shape parameters. As mentioned the other regularizers are commonly used and minimize the mesh normal variance and Laplacian to encourage a smooth surface, as well as an edge loss to encourage uniform distribution of the mesh vertices. These are defined below,

$$\mathcal{L}_{\text{v}} = \|\Delta v\|_2^2, \quad \mathcal{L}_{\text{e}} = \sum_{v} \sum_{k \in \mathcal{N}(v)} \|v - k\|_2^2 \tag{2}$$

$$\mathcal{L}_{\text{lap}} = \sum_{v} \|\delta_v\|_2^2 \text{ , where } \delta_v = v - \sum_{k \in \mathcal{N}(v)} \frac{k}{\|\mathcal{N}(v)\|} \tag{3}$$

$$\mathcal{L}_n = 1 - \frac{n_l \cdot n_r}{||n_l|| \cdot ||n_r||} \tag{4}$$

where v corresponds to a vertex, $\mathcal{N}(v)$ are the edges connected to v, and n_l, n_r are normals of neighboring faces. Finally $\mathcal{L}$ is formally defined by

$$\mathcal{L} = \underbrace{\lambda_1 \mathcal{L}_{\text{sil}}}_{\text{image loss}} + \underbrace{\lambda_2 \mathcal{L}_{\text{v}} + \lambda_3 \mathcal{L}_{\text{n}} + \lambda_4 \mathcal{L}_{\text{lap}} + \lambda_5 \mathcal{L}_{\text{edge}}}_{\text{regularize /smooth loss}} \tag{5}$$

where we used $\lambda_1, \lambda_2, \lambda_3, \lambda_4 = 1$ and $\lambda_5 = 0.1$.

Texturing. To obtain a texture, we unwrap the MANO hand mesh to find UV coordinates. During test time, we rasterize the UVs over the mesh to obtain UV coordinates per fragment, which can be used to map colors from the image to the texture map. This is done in every frame, to adapt to changes in the texture caused by e.g. wrinkling of the skin. To fight outliers we use exponential smoothing when updating the texture, i.e. we average the current texture with the previous.

3.3. Stereo Extension

Estimating a 3D hand pose from a single RGB image is an ill-posed problem due to depth and scale ambiguities. Stereo images have the ability to resolve these, but stereo training data is lacking. Thus we propose a network architecture that allows the use of monocular datasets like FreiHAND to learn the correlations between poses and real hand image data. Limited stereo data such as our proposed synthetic hand dataset, introduced in Sec. 3.4 can then be utilized to learn a sensible stereo fusion.

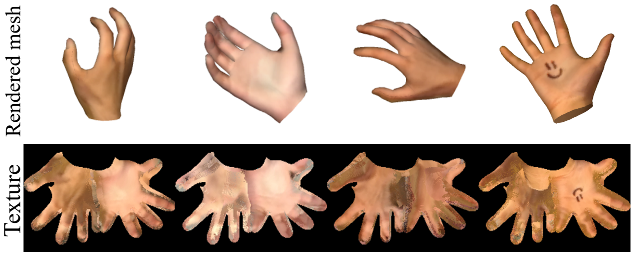

Figure 3. **Visualization of texture maps and rendered meshes.**

We extend our model to support stereo images, mostly by duplicating the mono architecture. An illustration of the full architecture can be found in the supplementary material. However, in addition to regressing p, s and c for each view from the base features $\boldsymbol{X} \in \mathbb{R}^{512}$ of the MANO regressor, we also concatenate them from both views into $\boldsymbol{X}_{stereo} \in \mathbb{R}^{1024} = \boldsymbol{X}_{right}|\boldsymbol{X}_{left}$. From this, we regress additional stereo weights $w \in \mathbb{R}^{48}$ using a fully connected layer. The obtained weights are used to combine the predicted poses from the left and right views by computing the new stereo fused right hand pose p_{right} as

$$p_{\text{right}} = w \; p_{\text{right}} + (1 - w) \; p_{\text{left}}. \tag{6}$$

In other words, we ideally want the network to be able to differentiate visible and occluded joints of different views, so that they can be merged in a meaningful way. For example, if a joint is self-occluded by the palm in the right view, the exact location is not recoverable and we have to rely on predicting likely distributions of possible locations. However, if the specific joint is visible in the left view, the network now has the power to compensate for the lack of information by utilizing the left view joint predictions over the right view predictions.

The predicted shape parameters from both views are fused by averaging, the same way we refine the left and right camera parameters while taking their geometric constraints into account.We opted to not learn weights for fusing these parameters, because of the small variance exhibited between the views. With ξ transforming from left to right view, we obtain

$$s_{\text{stereo}} = \frac{s_{\text{right}} + s_{\text{left}}}{2} \qquad c_{\text{right}} = \frac{c_{\text{right}} + \xi(c_{\text{left}})}{2}. \tag{7}$$

We can also leverage stereo during the optimization stage, by computing the silhouette image loss as the average over the two views. Letting S^L, S^R be the left and right silhouette images, the stereo loss term is simply

$$\mathcal{L}_{\text{sil}} = \frac{\|S^L_{\text{mesh}} - S^L_{\text{target}}\|_2^2 + \|S^R_{\text{mesh}} - S^R_{\text{target}}\|_2^2}{2} \tag{8}$$

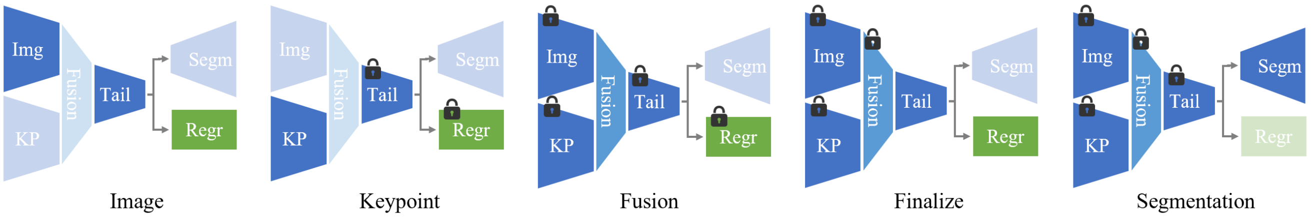

Figure 4. **Training framework.** Visualization of the five monocular training stages. The active components of each stage are highlighted. Components with the lock symbol have their parameters fixed, i.e. they are not updated during training.

3.4. Datasets

FreiHAND Dataset. The FreiHAND [42] dataset consists of 32,560 unique RGB images of right hands with the corresponding MANO annotation, intrinsic camera parameters and 2D joint locations. The FreiHAND dataset is unique in that the annotations include not only pose but also shape parameters for the MANO model.

Synthetic Stereo Hands Dataset. Since training data for a stereo setting with focus on egocentric viewpoints is currently lacking, we generated a large-scale synthetic dataset. The primary use is to learn the proposed stereo fusion, but the pixel-accurate segmentation masks are also utilized for training the segmentation decoder. We adapt ObMan [14] to our needs by modeling our stereo camera and focusing on egocentric viewpoints. We render left and right RGB images, corresponding segmentation masks and output annotations that include hand vertices, 2D joints, 3D joints along with the camera parameters. In total, the dataset contains 15082 samples, corresponding to 30164 RGB images, annotations and segmentation masks.

3.5. Training Framework

Training is performed in multiple stages, where different components of the model are frozen or removed components from the information flow. This has proven to greatly increase performance, as discussed in Table 2, at a marginal implementation overhead. For the monocular model we used five training stages, as illustrated in Fig. 4. The 'Image', 'Keypoint', 'Fusion' and 'Finalize' stage use the FreiHAND dataset with data augmentation (Sec. 3.6). The 'Segmentation' stage use both our synthetic stereo hands dataset, which is pixel-perfect but synthetic, and FreiHAND, which is real but less accurate). This is done in order to mitigate the drawback of each respective dataset.

The stereo setting includes two additional stages, which are carried out using the synthetic stereo hand dataset (Fig. 5). The first 'synthetic adaption' stage retrains the image encoder to adapt towards the synthetic image data. We conclude by training the 'Stereo' weights regressor to learn a sensible fusion. During test time on real data, the original non synthetically adapted image encoder is used.

Additionally, the training strategy does not only outperform a joint training that tends to emphasize the dominant branch, but also exhibits better adaptability and flexibility, as only the image head could be retrained to adapt to a new setting.

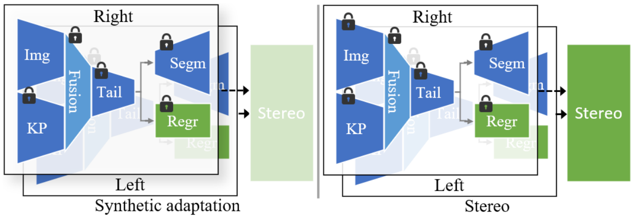

Figure 5. **Visualization of the two stereo training stages.** In the synthetic adaption stage the image head is retrained, whereas in the stereo stage the fusion weights are learned.

3.6. Training Losses

MANO Loss. We use the following weighted loss to train the network that regresses the MANO parameters:

$$\mathcal{L}_{\text{MANO}} = \underbrace{\lambda_1 \mathcal{L}_{\text{pose}} + \lambda_2 \mathcal{L}_{\text{shape}} + \lambda_3 \mathcal{L}^{\text{2D}}_{\text{joints}}}_{\text{mano losses}} + \underbrace{\lambda_4 \mathcal{L}_{\text{shape}} + \lambda_5 \mathcal{L}_{\text{cam}}}_{\text{regularization}} \quad (9)$$

where $\mathcal{L}_{\text{pose}}, \mathcal{L}_{\text{shape}}, \mathcal{L}^{2D}_{\text{joints}}, \mathcal{L}_{\text{cam}}$ are the MSE of the pose, shape, 2D joints and scale camera parameters, and $\mathcal{L}_{\text{shape}}$ is the norm of the shape PCA. While $\mathcal{L}_{\text{pose}}$ and $\mathcal{L}_{\text{shape}}$ infer MANO parameters, $\mathcal{L}^{2D}_{\text{joints}}$ learns the weak camera parameters for the root joint, so that the mesh can be projected back onto the image. As the MANO model encodes the mean shape by the zero vector, we introduce an additional regularization term $\mathcal{L}_{\text{shape}}$ to prevent blow ups. Finally, we add $\mathcal{L}_{\text{cam}}$ to avoid placing the root joint behind the camera. We used $\lambda_1 = 10$, $\lambda_2 = 1$ $\lambda_3 = 100$, $\lambda_4 = 0.5$ and $\lambda_5 = 10^{-5}$.

Segmentation Loss. For training the segmentation decoder, we rely on the binary cross entropy.

$$\mathcal{L}_{\text{seg}} = \text{BCE}(\hat{y}, y) = -(y \log(\hat{y}) + (1-y) \log(1-\hat{y})) \quad (10)$$

Stereo Loss. We use a weighted sum similar to Eq. (9), but we directly supervise the vertices and the 3D joint locations by minimizing their corresponding MSE's $\mathcal{L}_{\text{vert}}$ and $\mathcal{L}^{3D}_{\text{joints}}$:

$$\mathcal{L}_{\text{stereo}} = \lambda_1 \mathcal{L}_{\text{vert}} + \lambda_2 \mathcal{L}^{3D}_{\text{joints}} \quad (11)$$

Data Augmentation. We use common techniques such as rescaling, rotating, cropping and blurring during training.

4. Results

All our experiments were carried out on a workstation with an AMD Ryzen 9 5900x, 64 GB of RAM and an Nvidia GTX TITAN X. Our approach was implemented in PyTorch and the code, models and also our synthetic stereo dataset will be released upon publication.

Metrics. The hand vertex prediction is evaluated by the Mean Per Vertex Position error (MPVE), which measures the Euclidean distance in millimeters between the estimated and groundtruth vertex coordinates. We also report the Area Under Curve (AUC) [42] of the Percentage of Correct Keypoints (PCK) by computing the percentage of predicted vertices lying within a spherical threshold around the target vertex position. Further, we report F-scores which - given a distance threshold - define the harmonic mean between recall and precision between two point sets [18]. To evaluate the segmentation performance we use pixel accuracy and the more meaningful mean intersection over union metric. Also binary masks, obtained by the projection of the hand meshes are evaluated using the mean IoU.

4.1. MANO Regression

We evaluate the HandNet component which regresses MANO parameters from monocular RGB images. The HandNet was trained for a total of 500 epochs, distributed with 120 + 100 + 100 + 180 epochs on the respective training stages (Sec. 3.5). During the last 20 epochs of each stage the learning rate was linearly decayed. The Adam optimizer [17] was used with an initial learning rate of 0.0002 at each stage and $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

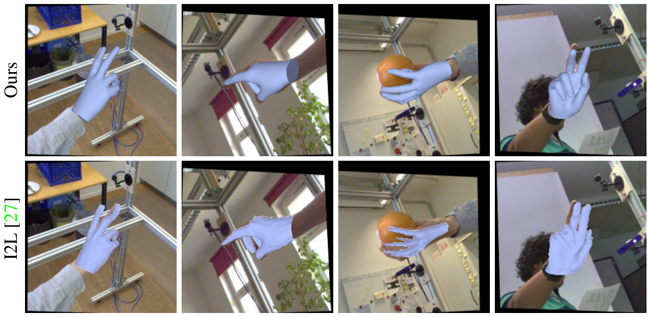

Figure 6. **Qualitative results on the FreiHAND test set.** Visualization of predictions on the FreiHAND test set, in comparision to I2L [27]. We used 3 optimization iterations for our method.

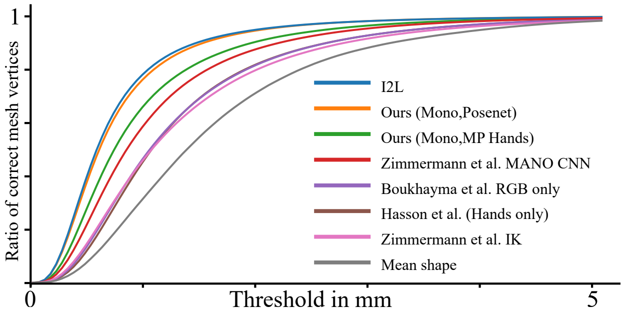

Figure 7. **PCK curves for the FreiHAND benchmark test.** Based on vertex positions using PA. We outperform all other model based approaches and show similar quantitative performance than the state of the art model free approach, while being more robust.

State-of-the-art Comparison. We compared the monocular HandNet component to the state-of-the-art using the FreiHAND test set, which provides no groundtruth, via an online challenge[1]. As some methods predict only relative coordinates, the benchmark reports through Procrustes analysis (PA) aligned metrics. Table 1 shows that our approach outperforms baselines, such as the mean shape and inverse kinematic fits of the MANO model, and also ranks higher than other model based approaches such as [14, 3, 42, 21]. Further, it performs quantitative similarly to I2L [27], a model-free approach that occassionaly produces collapsed results (see Fig. 6). The qualitative superiority of our methods over I2L is also well demonstrated in the supplementary video. Note that I2L uses a 3D joint prediction network (PoseNet) trained on the FreiHand dataset, which is geared towards the test data more than our general MP Hands keypoint detector [38] . For a fair comparison, we use projected PoseNet 3D joint predictions as our keypoint input. The PCK curves of the methods are visible in Fig. 7.

Methods	MPVE (PA)↓	F@5mm (PA)↑	F@5mm (PA)↑	Model based	GT scale
Mean shape	1.64	0.336	0.837	✓	
Inverse Kinematics [42]	1.37	0.439	0.892	✓	
Hasson *et al.* [14]	1.33	0.429	0.907	✓	✓
Boukhayma *et al.* [3]	1.32	0.427	0.894	✓	✓
Mano CNN [42]	1.09	0.516	0.934	✓	✓
Kulon *et al.* [21]	0.86	0.614	0.966	✗	✗
I2L [27]	0.76	0.681	0.973	✗	✗
Ours (Mono, MP Hands)	0.97	0.575	0.949	✓	✗
Ours (Mono, PoseNet(I2L))	0.78	0.662	0.971	✓	✗

Table 1. **Quantitative results on the FreiHAND benchmark test set.** Comparison of our approach with other methods on the task of monocular hand pose and shape estimation. Our method outperforms the other model based methods in all three evaluation metrics and shows similar, but qualitatively more robust performance than model free approaches.

Modality Fusion. For the MANO Regression experiments we use a local validation set that was created by randomly splitting 30% of the FreiHand samples into a validation dataset using the remaining 70% for training the network. We made sure to base the training and test splits on unique samples of the FreiHAND dataset and include the different color-hallucinated versions in the respective set.

To test the robustness of our method, when no keypoints

[1] https://competitions.codalab.org/competitions/21238

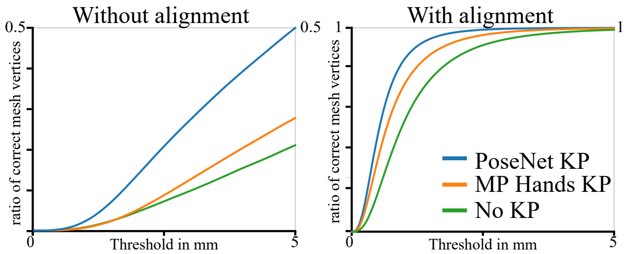

Figure 8. **Modality Fusion.** PCK plots comparing the performance of our method, when no keypoints are provided against using MP Hands [38] or PoseNet [27] as keypoint detector on the validation dataset.

	MPVE↓		F@5mm↑		F@15mm↑	
	no KP	with KP	no KP	with KP	no KP	with KP
single-stage	1.79	1.16	0.31	0.49	0.82	0.92
multi-stage	**0.85**	**0.78**	**0.45**	**0.66**	**0.96**	**0.97**

Table 2. **Training stages.** The multi-stage training yields the best results when the keypoints are provided, but also outperforms the directly trained version with no keypoints. Especially the adaptive fusion seems to profit from training framework, as the network better learns to emphasize the respective modality.

are available, we evaluated the impact of the modality as well as the performance of the fusion component, in such cases. Therefore no test-time refinement was performed for this experiment. The results in Fig. 8 with mesh alignment through PA, show that good keypoint prediction boost the performance and are a useful modality. When no meaningful information is provided by this input branch our fusion component successfully learned to suppress the keypoint modality with slightly lower performance. In contrast, when we inspect the non-aligned results in Fig. 8, it becomes apparent that the keypoint heatmap provides especially valuable information for the global image-model alignment with larger difference between the versions. This proves, that the additional modality and fusion are indeed helpful, because good initial global alignments are crucial for the fine grained optimization to succeed.

Training Framework. To evaluate the usefulness of the multi-stage training procedure we compare the performance to a version of the same network without the introduced training stages on the validation set. The results are shown in Table 2. The network trained using our training stages significantly outperforms a directly trained network and the ablation on the keypoint modality shows, that the fusion component is better capable at adapting to the provided inputs when trained with the proposed stages.

4.2. Optimization

To evaluate the benefit of the optimization stage, which refines the image-model alignment, we project the obtained mesh onto the image plane and compare it with the

	IoU↑	
	MP Hands	PoseNet
no refinement	0.677	0.722
3 iterations	0.706	0.758
10 iterations	0.735	0.793
15 iterations	**0.748**	**0.806**
I2L [27]		0.737

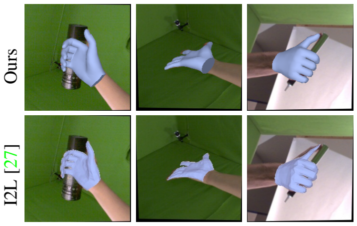

Figure 9. **Projection error.** The table lists the IoU scores for the projection error of the hand mesh with the GT mask, when no optimization, 3, 10 and 15 optimization iterations are used. We further compare against I2L [27], where use the same PoseNet keypoint predictions. Additionally we visualize samples, where the projection error between our method (3 iterations) and I2L is comparable, but our method outputs qualitatively more robust hand meshes, compared to the partly collapsed I2L outputs.

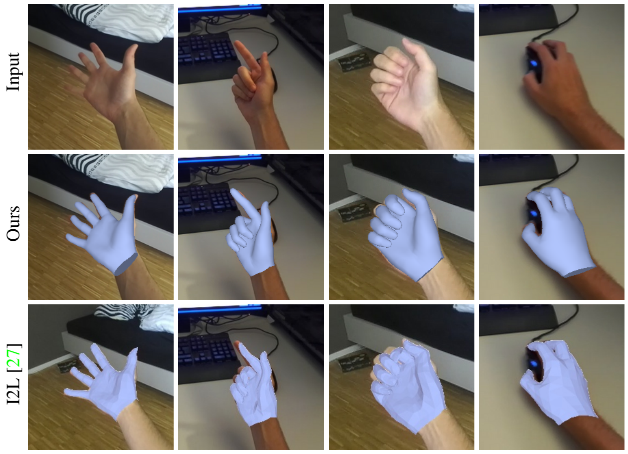

Figure 10. **Qualitative results on real captured data.** Visualization of predictions from real captured data, different from the FreiHand training data for both our method and I2L [27] as comparison. We used 3 optimization iterations for our method. The required root joint input for I2L is extracted from our predictions.

groundtruth segmentation mask. The validation results are reported in Fig. 9, where we also compare against I2L [27]. Additionally, we visualize samples with similar projection losses for both methods, but with qualitative more robust results produced by our approach.

Qualitative results. To compare our method qualitative with I2L [27] on real data different from the FreiHAND [42] dataset, we captured video sequences with various hand poses. Sample hand crops and the respective hand mesh predictions are visualized in Fig. 10. Our method is more robust, especially with fingers located on top of the palm.

4.3. Segmentation

The segmentation decoder was trained for 10 epochs, linearly decaying the learning rate during the last 6 epochs. The ADAM optimizer [17] was used with an initial learning rate of 0.0002 and $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

The segmentation experiments used both FreiHAND and

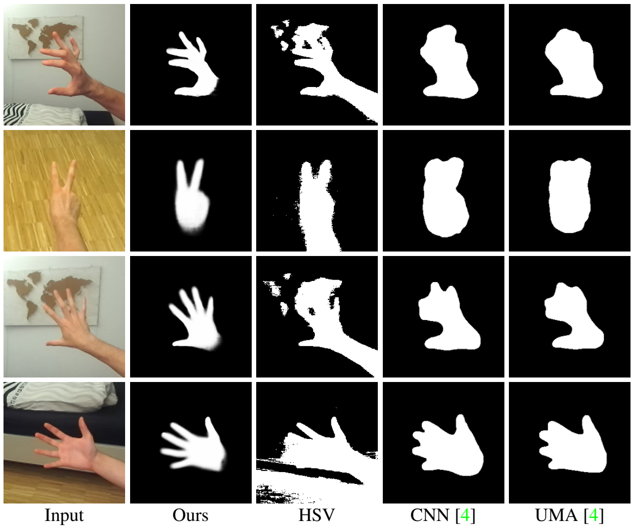

Figure 11. **Qualitative segmentation comparison.** Visualization of obtained segmentations by different methods. HSV corresponds to thresholding on skin color in the HSV color space. CNN [4] corresponds to the Bayesian CNN that serves as starting point for the uncertainty-guided model adaptation (UMA [4]).

	Combined		FreiHAND		Synth	
	IoU↑	Accuracy↑	IoU↑	Accuracy↑	IoU↑	Accuracy↑
HSV	0.26	0.81	0.33	0.90	0.18	0.72
CNN [4]	0.85	0.98	0.77	0.97	0.93	0.99
UMA [4]	0.86	0.98	0.78	0.98	0.93	0.99
Ours	**0.88**	**0.99**	**0.80**	**0.98**	**0.95**	**0.99**

Table 3. **Segmentation Results.** Our method performs best.

our synthetic dataset. Specifically, we use 70% of the data from each dataset for training and 30% for validation. Due to the chosen size of the synthetic dataset, the corresponding number of samples from the two datasets are also balanced.

We compared the segmentation performance to the following three baselines, abbreviated by HSV, CNN, UMA. **HSV:** We applied simple thresholding in the HSV color space for sensible values for skin color [19]. **CNN [4]:** This refers to the bayesian CNN in Cai *et al.* [4] that is based on RefineNet [22] and a starting point for their proposed self-supervised uncertainty-guided domain model adaption (UMA). **UMA [4]:** We retrained the bayesian CNN using our training dataset and further ran the uncertainty-guided model adaption (UMA) on the validation dataset. The results are recorded in Table 3. The performance of the CNN and the UMA is rather similar, because the training and validation dataset are from the same data distribution and the domain adaption is limited. To make correct use of the UMA method, we further evaluated the segmentation qualitatively on captured real data, where the UMA is based on the obtained video frames. Fig. 11 visualizes input image crops and the respective predicted segmentation masks of each method. Contrasting with the color thresholding, all data driven methods correctly learned to differentiate between hand and arm. Furthermore, they are not impacted by similarly colored objects like the world map or floor. Our method stands out at segmenting fingers, as both the CNN and UMA method fail to segment them in detail.

Version	MPVE (PA)↓	F@5mm (PA)↑	F@15mm (PA)↑
Monocular	1.65	0.352	0.860
Stereo	**0.94**	**0.578**	**0.967**

Table 4. **Stereo case.** The stereo version greatly outperforms the monocular, showing that multiple views can resolve ambiguities.

4.4. Stereo extension

For the stereo case, the training consists of 2 epochs of synthetic adaption and 8 epochs of learning the stereo weights, where we linearly decay the learning rate over the last 4 epochs. We used Adam [17] with initial learning rate of 0.0002 and $\beta_1 = 0.9, \beta_2 = 0.99$. For the experiment in Tab. 4 we created a 70/30 split of our synthetic stereo dataset and only used the right image for the monocular validation (the same synthetic adapted HandNet was used for comparability).

5. Discussion and Conlusion

We argue that our hybrid approach is more expressive and personalized than work relying on the MANO hand model alone and also more efficient and robust than methods that directly try to infer a mesh without an underlying hand model. The experiments show that our approach outperforms state-of-the-art methods in monocular RGB hand pose and shape estimation. The integrated hand segmentation network exhibits state-of-the-art performance, which enables qualitative improvements via the proposed fine grained test time optimization. The accurate mesh-image alignments allows for texturing, which yields visually pleasing personalized hand meshes. Additionally, we demonstrated the usefulness of the stereo extension for increased robustness. Due to low computational requirements, our method can be used in real-time applications.

For future work, we believe that the monocular and stereo hand mesh estimation could be further improved. This includes the creation of new and better real datasets, as they appear to be a limiting factor especially for the challenging egocentric perspective. In addition, the realistic appearance component could be further studied. Due to the fine grained optimization, we were able to project the texture map, but generative approaches for the texture map creation could also be studied and lead to fruitful results. Further, the intrinsic decomposition of texture maps for adaptive relighting could be explored.

Acknowledgments. This research was partly supported by Innosuisse funding (Grant No. 34475.1 IP-ICT) and a research grant by FIFA.

References

[1] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–432. IEEE, 2003. 2

[2] Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 10(6):849–865, 1988. 2

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 2, 3, 6

[4] Minjie Cai, Feng Lu, and Yoichi Sato. Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14392–14401, 2020. 8

[5] 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. 2

[6] Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13283, 2021. 2

[7] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021. 2

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020. 2

[9] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018. 2

[10] 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. 2

[11] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–2000, 2017. 2

[12] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019. 2

[13] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. 2

[14] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 2, 3, 5, 6

[15] 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. 2

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 7, 8

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 6

[19] Seema Kolkur, D Kalbande, P Shimpi, C Bapat, and Janvi Jatakia. Human skin detection using rgb, hsv and ycbcr color models. *arXiv preprint arXiv:1708.02694*, 2017. 8

[20] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019. 2

[21] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020. 2, 6

[22] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 8

[23] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 4

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 3

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[26] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 2

[27] 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. 2, 6, 7

[28] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 2

[29] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3316–3324, 2015. 2

[30] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011. 2

[31] Paschalis Panteleris and Antonis Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 575–584, 2017. 2

[32] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 2, 3

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[34] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3633–3642, 2015. 2

[35] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114. Wiley Online Library, 2015. 2

[36] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, pages 1–47, 2019. 2

[37] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9877–9886, 2019. 2

[38] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 2, 6, 7

[39] 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. 2

[40] 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/CVF International Conference on Computer Vision*, pages 2354–2364, 2019. 2, 3

[41] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017. 2

[42] 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/CVF International Conference on Computer Vision*, pages 813–822, 2019. 5, 6, 7