HandDGP: Camera-Space Hand Mesh Prediction with Differentiable Global Positioning

Eugene Valassakis*, Guillermo Garcia-Hernando

Niantic
https://nianticlabs.github.io/handdgp/

Abstract. Predicting camera-space hand meshes from single RGB images is crucial for enabling realistic hand interactions in 3D virtual and augmented worlds. Previous work typically divided the task into two stages: given a cropped image of the hand, predict meshes in relative coordinates, followed by lifting these predictions into camera space in a separate and independent stage, often resulting in the loss of valuable contextual and scale information. To prevent the loss of these cues, we propose unifying these two stages into an end-to-end solution that addresses the 2D-3D correspondence problem. This solution enables back-propagation from camera space outputs to the rest of the network through a new differentiable global positioning module. We also introduce an image rectification step that harmonizes both the training dataset and the input image as if they were acquired with the same camera, helping to alleviate the inherent scale-depth ambiguity of the problem. We validate the effectiveness of our framework in evaluations against several baselines and state-of-the-art approaches across three public benchmarks.

Keywords: camera-space hand mesh estimation · hand and body pose shape from RGB images · 3D-to-2D scale ambiguity · differentiable solver

1 Introduction

Predicting 3D hand meshes from single-view RGB images has become an increasingly popular research area due to its potential in augmented and virtual reality applications, such as virtual try-on experiences [49], human digitization [16], gaming [22], and teleoperation [1,18]. Despite recent progress, challenges remain [3,52] due to the hand's articulated structure, self-occlusions, annotation difficulty, and 2D-to-3D scale and depth ambiguity.

Because of these challenges, most previous work have focused on predicting quality root-relative hand meshes, *i.e.* 3D hand meshes in coordinates relative to a pre-defined root joint, such as the wrist [14], as opposed to predicting in the global camera space. Root-relative predictions with a camera projection model [4, 8, 29, 54] can be sufficient in applications that end up displayed on 2D images, such as virtual try-on experiences. However, camera-space predictions are critical for interactions in 3D virtual and augmented worlds, *e.g.* in applications such as gaming, and office work when using mixed-reality headsets [2, 37].

* Now at Synthesia. Work done while at Niantic.

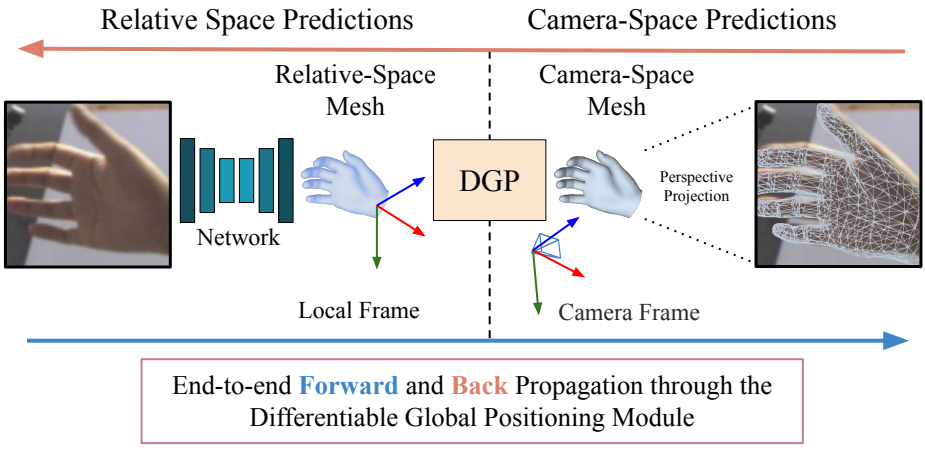

Fig. 1: Method overview: Through our Differentiable Global Positioning module (DGP) which predicts the root translation of the hand, our method is able to back-propagate through the root-finding operation, enabling an end-to-end solution.

For tasks that need 3D camera-space hand meshes, the dominant approach is to take those root-relative hand meshes and then lift those predictions to the camera-space 3D coordinates in a separate, independent process. For example, Iqbal *et al.* [28] and Zhang *et al.* [55] output predictions in a relative 2.5D space and infer the global coordinates analytically up to scale. With a similar relative representation, Moon *et al.* [40] and Tang *et al.* [49] predict camera-space coordinates with the aid of an independent network known as RootNet [39]. Exploiting 2D-to-3D correspondences, CMR [15] and MobRecon [14] first predict both 2D keypoints and 3D root-relative meshes, and then find the 3D rigid transformation that best explains the mesh projection via an independent test-time optimization process. Similarly, HandOccNet [41] also uses a test-time optimization to predict scale and root translation by minimizing a 2D projection loss. All these methods confine the learning stage to the relative space, yielding state-of-the-art relative meshes with high efficiency, but falling short when it comes to placing the hand in the camera space, as shown in our experiments.

We propose to achieve the best of both worlds by simultaneously learning root-relative meshes and the 3D lifting function in an end-to-end manner. To this end, we propose a Differentiable Global Positioning (DGP) module, a modern take on the classical Direct Linear Transform algorithm [23]. DGP enables the backpropagation of gradients directly from camera space outputs to the 2D-3D correspondences defined by the 2D keypoint predictions and the root-relative 3D hand meshes. Thanks to being differentiable, DGP could potentially also be included in the learning of any hand mesh prediction neural network that predicts 2D hand keypoints and root-relative 3D hand meshes such as CMR [15] and MobRecon [14]. In our experiments, we show that allowing gradients to flow from the camera-space through the root-relative network results in better global hand mesh predictions compared to disjoint two-step approaches, *e.g.* using RootNet [39] or test-time optimizations, as well as an end-to-end regression

baseline that predicts both camera-space translation and relative predictions. Further, inspired by recent metric 3D scene geometry prediction work [51], we found in our experiments that a simple image and camera parameter rectification step helps to alleviate the inherent 2D-to-3D depth and scale ambiguity problem. The core idea is to conduct all learning as though the images were captured using the same camera model, thereby reducing the ambiguity that the network must resolve. While this approach leads to improved camera-space predictions, it incurs a slight decline in performance for relative-space predictions. We thoroughly examine the impact of this rectification step on both our method and on baseline approaches. In summary, our **contributions** are as follows:

- We propose HandDGP, a flexible framework that unifies the learning of both root-relative and camera-space hand meshes in an end-to-end manner.
- We identify and highlight the importance of performing image rectification in alleviating some of the 2D-to-3D depth and scale ambiguity in the camera-space hand mesh prediction problem.
- We conduct an extensive experimental evaluation of various design choices for addressing the problem of camera-space hand mesh prediction, aiming to encourage research in this direction. This area has been somewhat overlooked in previous work in favor of root-relative predictions.

2 Related Work

Camera-Space 3D Hand Mesh Prediction. Most previous works in monocular RGB-based camera-space 3D hand mesh and pose estimation follow a two-stage approach: (1) a first stage for predicting hand mesh/pose in root-relative coordinates, and (2) a lifting stage to recover camera-space coordinates from root-relative ones. For monocular 3D hand pose estimation, Iqbal *et al*. [28] predicts in 2.5D root-relative space, and then lifts those predictions to 3D camera space predictions using an analytical solution. This result is up to a scaling factor, however, and to resolve the scale ambiguity an extra scale parameter is required, which is assumed to be provided [48] or globally estimated from data. I2L-MeshNet [40] proposes a regression approach to recover root-relative 2.5D meshes and subsequently lifts them to camera-space using a separate network RootNet [39] which uses prior anthropometric knowledge to reduce scale ambiguity [34]. NFV [26] proposes a neural voting approach with a 3D implicit function that directly regresses 3D hand poses in camera-space from full images. NFV uses a Marching Cubes post processing to predict meshes, which degrades efficiency and is at the cost of not having semantic mappings for their predicted vertices, which are crucial for some applications. Hasson *et al*. [24] predicts both object and hand camera-space translations using both hand and object cues, but makes assumptions on the geometry of the object which facilitate the scale recovery. Closest to our work, CMR [15], MobRecon [14] and HandOccNet [41] first predict both 2D keypoints and 3D root-relative meshes, while 3D camera-space coordinates are obtained with a test-time registration function that estimates the root position. This function typically aims to find the

3D rigid transformation for the hand mesh that best projects into the 2D input image given correspondences between 2D hand keypoints and 3D mesh keypoints. Our method builds on top of such 2D-3D paradigm with the key difference of leveraging a differentiable registration function, enabling us to directly learn our mesh network directly in the camera-space in an end-to-end manner. We compare our work with [14, 15, 24, 26, 39–41] in Section 4.3 and find that our method outperforms these state-of-the-art methods in camera-space predictions.

Root-Relative 3D Hand Mesh Prediction. Different methods have been proposed for RGB-based monocular hand mesh reconstruction, with various hand output representations, such as parametric models, voxels, vertices, implicit functions or UV maps [13]. Parametric models [4, 5, 8, 10, 15, 17, 21, 25, 53] typically regress hand shape and pose coefficients of the MANO parametric hand model [46]. These methods are intrinsically limited by the expressiveness of the model used. Model-free methods circumvent these limitations by working on high dimensional representations. This can be true in voxel-based methods, such as I2L-MeshNet [40], which represents volumetric hand data in a 2.5D manner at the cost of efficiency. Also typically constrained by efficiency but at the benefit of high resolution are implicit function methods [26, 30, 31], which inherit from the trend started by human body digitization [6, 26, 38, 43, 47]. Vertex-based methods [14, 15, 19, 32, 35, 36, 41] aim to directly predict 3D vertex coordinates. In this work we build upon MobRecon's framework [14] by adopting their pipeline: predicting 2D keypoints following an encoder-decoder approach then leveraging those keypoints to grid-sample features that are lifted to root-relative 3D by a graph neural decoder. We extend this framework by instead learning in camera-space owing to our proposed differentiable global positioning function. In Section 4.1 we show that our method improves [14]'s camera-space predictions significantly.

Differentiable Correspondence Solvers. Solving for unknown geometric quantities using 2D-2D or 2D-3D correspondences has long been a central subject in computer vision [23, 44]. With the emergence of deep learning, several studies have attempted to integrate these well known geometric and algebraic solutions to deep learning pipelines [7, 9, 11, 12, 45, 50]. Notable examples include (1) Chen *et al.* [11] which propose a differentiable perspective-n-point (PnP) solver and validate it in various problems such as pose estimation, or camera calibration, and Remelli *et al.* [45] that use a differentiable Direct Linear Transform (DLT) implementation to perform multi-view body pose estimation. We derive a DLT solution to the root finding problem in the context of hand-mesh inference, and show that it can be implemented differentiably and integrated to an end-to-end pipeline for camera-space hand-mesh prediction.

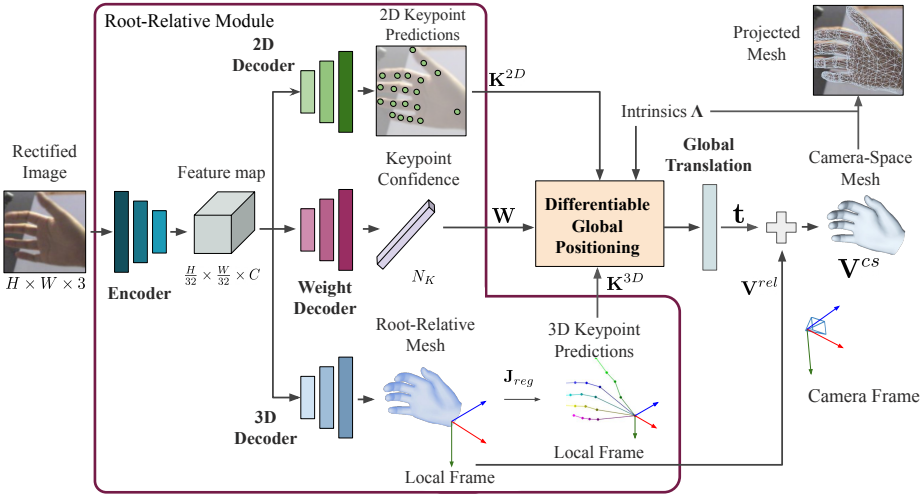

Fig. 2: HandDGP Framework Overview. Rectified images are passed through our framework, which predicts camera-space coordinates using our proposed DGP module.

3 Proposed Framework

Overview. Shown in Figure 2, the core idea behind our approach is to exploit the geometry of the problem to integrate hand root finding in a differentiable pipeline that can predict a hand mesh directly in 3D camera-space coordinates.

Predicting root-relative hand meshes. Starting from one RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we first predict a set of 2D keypoints $\mathbf{K}^{2D} = \{k_i^{2D}\}_{i=1}^{N_K}$, that can be joints or other landmarks, a set of root-relative 3D vertices $\mathbf{V}^{rel} = \{v_i^{rel}\}_{i=1}^{N_V}$, and a set of weights $\mathbf{W} = \{w_i\}_{i=1}^{N_K}$, that represent the confidence in the predictions of each keypoint. We then obtain $\mathbf{K}^{3D} = \{k_i^{3D}\}_{i=1}^{N_K}$, a set of root-relative $3D$ keypoints on the hand model that correspond to the 2D keypoints $\mathbf{K}^{2D}$. To obtain $\mathbf{K}^{3D}$, we assume having access to a 3D keypoint regressor $\mathbf{J}_{reg} : \mathbf{V}^{rel} \rightarrow \mathbf{K}^{3D}$. $\mathbf{J}_{reg}$ usually comes in the form of a matrix, which defines keypoints on the hand as a linear combination of mesh vertices, and is typically provided with popular mesh models such as MANO [46] for hands and SMPL [42] for full body meshes.

Finding the hand root with HandDGP. Our key innovation is our differentiable global positioning (DGP or positioning module, for brevity), which uses the root-relative 3D keypoints $\mathbf{K}^{3D}$, the 2D keypoints $\mathbf{K}^{2D}$ and the weights $\mathbf{W}$ in order to predict a global translation $\mathbf{t} \in \mathbb{R}^3$ in camera-space which we can use to obtain the camera-space vertex predictions $\mathbf{V}^{cs} = \{v_i^{cs}\}_{i=1}^{N_V}$, with

$$v_i^{cs} = v_i^{rel} + \mathbf{t}. \tag{1}$$

The camera-space vertex predictions can finally be used to project the mesh into 2D using a pinhole camera perspective projection. This pipeline allows us to

include the global root translation and the resulting mesh projections as part of our network training. Incorporating root prediction as part of the training this way has the benefit of avoiding the accumulation of errors that can occur when using two independent processes for root-relative predictions and root-finding. We finally note that (1) our method is agnostic to the particular method used to obtain the predictions for $\mathbf{V}^{rel}$, $\mathbf{K}$ and $\mathbf{W}$ and (2) the task of recovering the root of the hand is equivalent to finding the hand's global translation in camera space. Therefore, we will use both terms interchangeably throughout.

3.1 Differentiable Global Positioning

At the core of our method lies the differentiable global positioning module, which takes in $\mathbf{K}^{3D}$, $\mathbf{K}^{2D}$, $\mathbf{J}_{reg}$, and the camera intrinsics $\mathbf{\Lambda}$ as input, and outputs the global camera-space translation $\mathbf{t}$ in a differentiable manner. Although our full approach also considers the keypoint confidences $\mathbf{W}$, for clarity, in this section we describe how we obtain the global translation assuming equally confident keypoints. We explain how we incorporate keypoint confidences in Section 3.2.

To obtain the global translation $\mathbf{t} = [\tau_x, \tau_y, \tau_z]^T$ in a differentiable way, we derive a solution based on the Direct Linear Transform (DLT) [23, 44], adapted to our specific problem. Firstly, by design $\mathbf{K^{3D}}$ and $\mathbf{K^{2D}}$ give us a set of 2D-3D correspondences $\mathcal{M} = \{(k_i^{3D}, k_i^{2D})\}_{i=1}^{N_K}$, with $k_i^{3D} = [x_i, y_i, z_i]^T$ and $k_i^{2D} = [u_i, v_i]^T$. Additionally, it is important to note that the 3D keypoints $\mathbf{K}^{3D}$ are expressed in a frame that shares the same orientation as the camera frame, with only the global root translation missing to map root-relative keypoint coordinates to camera-space coordinates. Assuming a pinhole camera model with intrinsic parameters $\mathbf{\Lambda}$, we can express the projection equation as:

$$d_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{\Lambda} \left[\begin{array}{ccc|c} 1 & 0 & 0 & \tau_x \\ 0 & 1 & 0 & \tau_y \\ 0 & 0 & 1 & \tau_z \end{array}\right] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \tag{2}$$

with d_i the depth value of keypoint i. Expanding and re-arranging, this gives a system of linear equations that can be written in the following form:

$$\begin{bmatrix} -1 & 0 & u_i' \\ 0 & -1 & v_i' \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} x_i - z_i u_i' \\ y_i - z_i v_i' \end{bmatrix}, \tag{3}$$

where

$$\begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} = \mathbf{\Lambda}^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}. \tag{4}$$

Since Equation 3 is obtained considering a single keypoint correspondence, and since we have three unknowns, it is under-constrained. Using all the keypoints in

our correspondence set, Equation 3 can be re-written as

$$\begin{bmatrix} -1 & 0 & u'_1 \\ 0 & -1 & v'_1 \\ & \vdots & \\ 0 & -1 & v'_{N_K} \end{bmatrix} \mathbf{t} = \begin{bmatrix} x_1 - z_1 u'_1 \\ y_1 - z_1 v'_1 \\ \vdots \\ y_{N_K} - z_{N_K} v'_{N_K} \end{bmatrix}, \tag{5}$$

which has the form $\mathbf{At} = \mathbf{B}$. To solve for $\mathbf{t}$, we consider the least-squares solutions $\mathbf{t}^* = \arg\min_{\mathbf{t}} ||\mathbf{At} - \mathbf{B}||^2$ [44], which can be obtained in closed-form:

$$\mathbf{t}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}. \tag{6}$$

The DGP module first uses the network outputs to build the matrices $\mathbf{A}$ and $\mathbf{B}$, and then uses the linear least-squares solution to solve for the translation. Since all the operations involved are differentiable, we can use this translation to obtain and backpropagate through camera-space vertex predictions, fully incorporating the root-finding task in an end-to-end training pipeline.

3.2 Keypoint Selection

While the solution presented in Section 3.1 allows us to incorporate root finding into an end-to-end differentiable pipeline, it does not provide for any outlier filtering or keypoint selection mechanism that could help filter out more uncertain correspondences. This can be problematic in cases such as occlusion or self-occlusion of parts of the hand. In those instances, occluded parts would presumably result in more uncertain keypoint placements, they would be considered equally to visible keypoints when computing the global translation. To address this issue, we consider a weighted variant to our approach. Assuming each keypoint correspondence has a confidence score w_i associated with it it – in practice obtained from our weight decoder – we first construct a weight matrix by duplicating each weight once and placing them in a diagonal matrix $\mathbf{W} = \text{diag}([w_1, w_1, w_2, w_2 \dots w_{N_K}, w_{N_K}])$. Then, we consider a weighted least-squares minimisation $\mathbf{t}^* = \arg\min_{\mathbf{t}} ||\mathbf{W}(\mathbf{At} - \mathbf{B})||^2$, with closed-form solution:

$$\mathbf{t}^* = (\mathbf{A}^T\mathbf{W}^2\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{B}. \tag{7}$$

3.3 Input Image Rectification

Inspired by the recent work on monocular 3D geometry estimation from Yin *et al*. [51], the main idea is to establish a canonical camera space and transform all the training data to that space. During inference, the image is rectified just before entering the network, and the predictions are then mapped back to the original camera space. The original set of camera parameters is defined by $\{f, u_0, v_0\}$ where f represents the focal length (we assume $f_x = f_y = f$) and u_0 and v_0 be the principal points. We resize the input image $\mathbf{I}$ with the ratio $\omega_r = \frac{f^c}{f}$ which converts the camera parameters to $\{f^c, \omega_r u_0, \omega_r v_0\}$. Different to [51] and

motivated by the hand being the object of interest, we further rectify the principal point to be the center of the hand crop, resulting in the final canonical intrinsics matrix $\mathbf{\Lambda}$ defined by $\{f^c, H/2, W/2\}$ where f^c is the canonical focal length and the rectified principal point is the center of the input image. It is important to note that this does not affect the 3D geometry, thus root $\mathbf{t}$ remains unchanged.

3.4 Architecture and Training Details

Network Overview. In practice, our network first takes an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ as an input to a convolutional encoder to produce a feature map $\mathbf{F} \in \mathbb{R}^{H/32 \times W/32 \times C}$. The feature map $\mathbf{F}$ is then input to three separate decoder heads: a 2D decoder outputting a set of N_K 2D keypoints $\mathbf{K}^{2D}$, a 3D decoder outputting the root-relative vertices $\mathbf{V}^{rel}$, and a weights decoder outputting a set of confidence weights $\mathbf{W}$. Using $\mathbf{J}_{reg}$, we obtain the root-relative 3D keypoints $\mathbf{K}^{3D}$, forming a set of 2D-3D correspondences $\mathcal{M} = \{(u_i, v_i, x_i, y_i, z_i)\}_{i=1}^{N_K}$. Using $\mathcal{M}$, we then construct the matrices $\mathbf{A}$ and $\mathbf{B}$, and obtain $\mathbf{t}$ using Equations 6 and 7. Finally, we use $\mathbf{t}$, $\mathbf{V}^{rel}$ and $\mathbf{K}^{3D}$ to obtain the camera-space vertices $\mathbf{V}^{cs}$ and camera-space keypoints, following Equation 1, and use all of the mentioned network outputs to construct our training losses.

Weights decoder. While our 2D and 3D decoders follow MobRecon [14], our weights decoder is illustrated in Fig. 3. Starting from the feature map $\mathbf{F} \in \mathbb{R}^{H/32 \times W/32 \times C}$, we perform a series of 1×1 convolutions to obtain a new feature map $\mathbf{F_W} \in \mathbb{R}^{H/32 \times W/32 \times D}$. We then use the 2D positions provided by $\mathbf{K}^{2D}$ in order to grid sample a set of N_K, D-dimentional features, which we concatenate in $D \times N_K$ dimensional latent vector $\mathbf{Z_w}$ which is then processed through a set of dense layers with leaky ReLU activations, and the final output is processed through a sigmoid function, forcing the confidence weights to be in $[0, 1]$.

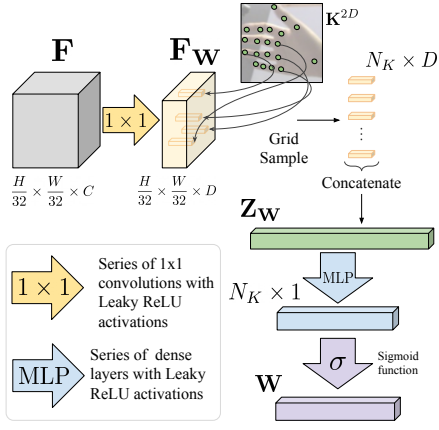

Fig. 3: Weight decoder head.

Losses. We distinguish between two sets of losses: *Relative-space* losses that are applied to any outputs that precede our global positioning module, and *camera-space* losses that follow the global positioning module. *Relative-space*: we follow MobRecon [14], and use (1) a root-relative 3D Vertex L_1 Loss between the root-relative ground truth vertices and the outputs of the 3D decoder, (2) a 2D keypoint L_1 loss on the outputs of the 2D decoder, (3) a consistency loss that enforces consistency of outputs between inputs with different visual augmentations, (4) a L_1 edge loss on the length of the predicted mesh edges, and (5) a normal loss on the predicted mesh normals. For our *camera-space* losses, we use (1) a

root mean squared error loss on the translation, (2) a keypoint consistency loss, which is a L_1 loss between the outputs of our 2D decoder and the projection of the predicted camera-space 3D keypoints, and (3) a 2D vertex L_1 loss on the 2D projection of the 3D camera-space vertices.

Implementation details and additional method information, including losses and network architecture details, are available in the supplementary material.

4 Experiments

Datasets. We report our experiments on the following datasets:

- **FreiHAND** [56]. We follow the evaluation protocol by [15, 26] using the dataset for our experiments. The dataset consists of images, 3D hand poses and MANO [46] fittings. It provides 130,240 training and 3,960 test images.
- **HO3D-v2** [20]. This dataset comprises real images capturing 3D hand-object interactions, with 66,034 images in the training set and 11,524 in the test set with MANO [46] model hand mesh annotations. Hands in this dataset suffer from severe occlusions caused by the manipulated object. The test set is not publicly available, and the evaluation is conducted on a public server. The server provides results in camera-space, root-relative, and aligned formats. However, participants are given ground truth camera-space hand translation values, and previous work typically reports results using this ground truth.
- **Human3.6M** [27] This dataset is a large-scale 3D body pose benchmark containing 3.6 million frames with annotations of 3D joint coordinates and SMPL [42] meshes. We follow existing evaluation protocols [15], but do not use common aligned metrics and measure errors in camera-space. We adapt our framework to predict body meshes by just swapping MANO by SMPL.

Metrics. We report the following metrics:

- **CS-MJE** / **CS-MVE**: Measures the error, in terms of Euclidean distance, between the predicted joints (MJE) / vertices (MVE) and the ground truth in camera-space (CS) coordinates. Both are average errors over the test set in mm. In some experiments, we compute the Area-Under-the-Curve (**AUC**) of Percentage of Correct Keypoints (**PCK**) vs. error thresholds.
- **RS-MJE** / **RS-MVE**: This measures the error between Procrustes-aligned predicted and ground truth joints and vertices. This metric serves as a measure of root-relative reconstruction quality.

4.1 Baselines and Method Ablations: Descriptions

Baselines. Our root-relative module is combined with different methods for predicting camera-space root translation. Methods labeled as 'Baseline + {*root prediction method*}' incorporate the root-relative module (see Fig. 2) without any global positioning system during training. Therefore, only the relative-space losses are applied and the positioning of the hand mesh in camera space occurs only at

Method	Image Rectification	End-to-End Training	Root-Relative RS-MJE↓	Root-Relative RS-MVE↓	Camera-Space CS-MJE↓	Camera-Space CS-MVE↓
B1. Baseline + PnP	✗	✗	6.8	6.9	50.1	50.2
B2. Baseline + DLT	✗	✗	6.8	6.9	50.0	50.0
B3. Baseline + RootNet [39]	✗	✗	6.8	6.9	62.6	62.5
B4. Baseline + Optimization [14]	✗	✗	6.8	6.8	50.2	50.3
B5. Root regression	✗	✓	7.2	7.6	81.3	81.8
BR1. Baseline + DLT + Rect.	✓	✗	7.4	7.5	48.4	48.8
BR2. Baseline + Optimization [14] + Rect.	✓	✗	7.4	7.5	48.9	49.0
BR3. Root regression + Rect.	✓	✓	7.8	7.7	52.6	55.4
A4. Ours - w/o Rect.	✗	✓	6.8	6.9	49.4	49.4
Ours (Full framework)	✓	✓	7.4	7.6	46.3	46.3

Table 1: Baseline and ablation experiments on FreiHAND dataset [56]. The *'Image Rectification'* column indicates whether the training images are rectified with our proposed approach. *'End-to-end Training'* denotes whether gradients flow through the global positioning function during training. The task we care about for 3D interactions is quality in the camera space where we outperform all the baselines and variants.

test time. In '**B1. Baseline + PnP**', the global translation is obtained by solving the perspective-n-point (PnP) problem [44], using the outputs of our 2D decoder and the 3D root-relative keypoints as 2D-3D correspondences. '**B2. Baseline + DLT**' refers to our baseline wherein our formulation of the DLT module is applied at test time only during a forward pass, *i.e.*, without gradient propagation through the network. In '**B3. Baseline + RootNet**', the global translation is determined using RootNet [39]. '**B4. Baseline + Optimization**' corresponds to our reimplementation of MobRecon [14], where the global translation is identified via an optimization-based process that minimizes the re-projection errors of predicted keypoints. Finally, '**B5. Root Regression**' is the only end-to-end baseline that employs a decoder, similar to our weight decoder shown in Fig.3, but for regressing the camera-space root value $\mathbf{t}$ and incorporating a translation loss. We also implemented some of these baselines with our image rectification step, denoted as **BR1-3**. Results are summarized in Table 1.

Ablations. We conduct several ablation studies on our full pipeline, as shown in Table 2. **Ours (Full Framework)** is our proposed method, which includes performing image rectification on the input image and conducting differentiable global positioning during training, with gradients back-propagating through the DGP module. '**A1. Ours - w/o DGP - w/o Rectification**' represents our framework without the proposed DGP and rectification steps, following the setup of B4. '**A2. Ours - w/o DGP**' evaluates the impact of our image rectification step in comparison to **A1**. In the '**A3. Ours - w/o Keypoint Weights W**' experiment, we utilize our full framework but, instead of predicting keypoint confidences using our weight decoder, manually set all the weights to 1.0. Lastly, in **A4. No Rectification + DGP** as detailed in Table 1, we implement the full DGP module but exclude the image rectification from the pipeline.

4.2 Baselines and Method Ablations: Results Discussion

Learning the 3D Global Positioning Function. In this evaluation, we assess the effect of back-propagating gradients from the DGP to the network, as pro-

Method	FreiHAND		HO3D-v2		Human 3.6M	
	CS-MJE↓	CS-MVE↓	CS-MJE↓	CS-MVE↓	CS-MJE↓	CS-MVE↓
A1. Ours - w/o DGP - w/o rectification	50.2	50.3	121.7	121.6	336.9	342.8
A2. Ours - w/o DGP	48.9	49.0	85.3	85.4	164.7	179.3
A3. Ours - w/o keypoint weights **W**	46.6	46.6	50.4	50.4	159.9	174.0
Ours (Full framework)	46.3	46.3	50.3	50.3	147.6	162.0

Table 2: Method ablation experiments. We outperform all ablations on camera-space predictions across all datasets, validating our design choices and use of HandDGP.

posed in Section 3.1. In Table 1 we observe that DGP outperforms all the other root-finding mechanisms. Non-learning approaches such as PnP, DLT and the optimization from Chen *et al*. [14] (**B1, B2, B4**) offer better predictions than learning-based methods such as RootNet [39] and our regression baseline (**B5**). These results still hold when image rectification is applied (**BR1-3**). Notably, the DGP formulation, even without training (forward pass), achieves comparable results to more sophisticated optimization methods, validating our design choice. In Table 2 we observe a significant reduction in error in all the datasets when learning the global positioning function with the proposed DGP (**A2**). For example, the camera-space vertex error is reduced by 2.7 mm, 35.1 mm and 17.3 mm in FreiHAND, HO3D-v2 and Human3.6M datasets respectively. This last result shows that HandDGP also transfers well to full body mesh predictions.

Image Rectification Effect. We observe that rectifying input images enhances the learning of scale-sensitive features, as we remove one source of ambiguity by keeping the camera intrinsics constant during training. Interestingly, we note that while rectifying images aids in camera space predictions, it impacts root-relative predictions (**Ours** vs **A4**). We hypothesize that the primary reason for this degradation is that rectifying training images *de facto* reduces the amount of data augmentations that the network encounters during training, affecting the prediction of both 2D keypoints and 3D mesh geometry. This suggests a potential trade-off between root-relative and absolute prediction quality, depending on how the training data is processed, exacerbated by the 2D-to-3D depth and scale ambiguity. The network may generate incorrect hand shapes to compensate for errors in translation prediction and vice versa. In all cases, better camera-space positioning is achieved when the input image is rectified.

Fig. 4: Keypoint selection. Effect of keypoint selection with our weight decoder. Test-set images on FreiHAND and HO3D-v2 with the 2D keypoints overlaid: The brighter the keypoint, the higher the weight.

Keypoint Selection. In Table 2 we observe that learning the keypoint selection weights **W** introduced in Eq. 7, instead of leaving them fixed further helps reducing the prediction error (**Ours** vs. **A3**). We illustrate this effect qualitatively

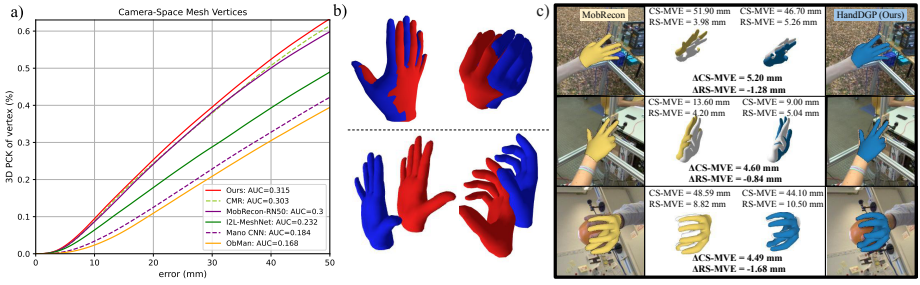

Fig. 5: (a) 3D PCK for camera-space hand mesh prediction on FreiHAND. **(b) Camera-space hand mesh predictions** rotated for illustration purposes. All meshes project correctly in the image, however some predictions display a 3D error offset. **(c) Root-relative vs camera-space errors**. Selected FreiHAND images with average camera-space (CS) and root-relative (RS) errors and ground truth (mesh in white).

in Fig. 4. For images in the test set, we draw the keypoints predicted by our 2D decoder onto the input image, color-coding each keypoint using its associated weight, as output by our weight decoder. The brighter the keypoint, the higher the weight associated with it. We also illustrate the weights themselves associated with each image in the bottom row of the figure. The first observation is that the weights associated with each keypoint clearly vary from image to image, indicating that our weight decoder actively contributes to our overall pipeline. Next, we notice that occluded keypoints generally tend to be associated with lower weights, suggesting that our method tends to focus on higher confidence keypoints.

Qualitative Results. In Fig. 6 we present qualitative results using our full framework on test images from different datasets. In most cases, the hand projections appear accurate, as also quantitatively demonstrated in our relative results in Table 1. Additionally, we showcase some rare failure cases where high ambiguity arises due to blurring, changes in viewpoint, and self-occlusion. Merely displaying projected 2D meshes might not convey the complete truth, as predictions are made in the actual 3D camera space. To address this, we include predicted 3D meshes from rotated viewpoints to illustrate the actual translation gap between the predictions and the ground truth in Fig. 5 (b). While the hand posture is generally correct, we observe instances of translation errors, sometimes overcompensating with larger or smaller hand shapes.

Root-relative results discussion. While not the focus of this work, Table 1 shows our method is outperformed by some baselines in the root-relative task by an average of 0.6 mm (RS-MVE). However, our improvements on camera-space errors (CS-MVE) are about 4 mm. To assess if this RS-MVE increase is an acceptable trade-off, Fig. 5 (c) compares our baseline and HandDGP on the FreiHAND test set. We selected images with a CS-MVE decrease of about 4 mm and an RS-MVE increase of at least 0.6 mm for MobRecon over HandDGP. The CS-MVE improvement is more evident when rotating the 3D viewpoint (columns 2-3). Our meshes are visibly closer to the ground truth, crucial for applications

requiring interaction with real and digital objects. Higher RS-MVE (0.84-1.68 mm) are harder to visualize. Notably, a better RS error doesn't always result in a better image projection (last row) and can hide scale errors due to Procrustes alignment. We believe the 4 mm CS-MVE improvement is significant, and our visualizations suggest a 0.6 mm increase in RS errors is an acceptable trade-off.

Framework Generalizability. In the supp. material we show the benefits of applying our framework to a different state-of-the-art root-relative method [15].

4.3 Comparison with State of the Art

FreiHAND: In Table 3 and Fig. 5 (a) we compare our proposed framework with state-of-the-art camera-space methods. Notably, our method achieves the lowest camera-space errors among all methods. We achieve a 2.6 mm error improvement over CMR [15], despite also using segmentation masks in their root finding. We also compare with the ResNet50 variant of MobRecon [14] which is the same as our baseline B4 from the previous section. For MobRecon, we report our results based on our implementation, which closely aligns with the one provided by the authors, with slight changes in the data processing that are also shared with our method. A similar trend is observed in Figure 5 where our method achieves the highest AUC of vertex PCK's among all methods closely followed by both CMR and MobRecon-RN-50. We also compare favorably with NVF [26], the only available method that, similar to us, predicts directly in camera space. Note that NVF does not directly predict hand meshes, making them not easily comparable due to the loss of the MANO topology as their meshes are generated using Marching Cubes. We observe that the root-relative + 2D-3D global positioning paradigm (ourselves, CMR and MobRecon) performs significantly better than other methods, followed by I2L-MeshNet [40] that uses [39] for root positioning, similar to our baseline B3.

Method	CS-MJE↓	CS-MVE↓
ObMan [25]	85.2	85.2
MANO CNN [56]	71.3	71.5
I2L-MeshNet [40]	60.3	60.4
NVF [26]	<u>47.2</u>	n/a†
CMR-SG-RN18 [15]	49.7	49.8
CMR-SG-RN50 [15]	48.8	<u>48.9</u>
MobRecon-RN50 [14]	50.2	50.3
Ours	**46.3**	**46.3**

Table 3: State of the art comparison on FreiHAND. †can only be evaluated for keypoints.

Method	CS-MJE↓	CS-MVE↓
HandOccNet [41]	156.4	156.2
MobRecon-RN50 [14]	121.7	121.6
Hasson *et al.* [24]	<u>55.2</u>	<u>55.1</u>
Ours	**50.3**	**50.3**

Table 4: State of the art comparison on HO3D-v2.

HO3D-v2: In Table 4 we present quantitative results of our method compared to state-of-the-art methods in camera-space coordinates using the public submission server. It is important to note that previous work typically reports their results in relative coordinates after an aligning step and often uses provided ground-truth root values. We do not have a way to know which participants on the leaderboard used this ground-truth; because of this, we had to recompute the scores. We report results of the available subset of methods that: i) have publicly available code, ii) provide a trained model, **and** iii) include a global coordinate prediction stage. This dataset is more challenging than FreiHAND

Fig. 6: Qualitative visualizations of our camera-space mesh prediction framework on FreiHAND, HO3D-v2, Human3.6M and in-the-wild web images [33]. Meshes are projected into the image plane using perspective projection, including failure cases. Further qualitative results and comparisons are available in the supplementary material.

as it is object-focused, and several occlusions are present. We observe that our method compares favorably to HandOccNet [41], which is currently the state of the art in relative pose predictions. To compute these predictions, we run their code that performs test-time optimization to predict the root translation. Despite their mesh projections looking good, the 3D predictions are often incorrect, with errors in the order of 15 cm. We also compare favorably to the work of Hasson *et al*. [24] that predicts both object and hand global translations. Given that object size is constant—compared to hands—predicting object roots is likely to help with the scale/depth ambiguity problem. We show qualitative comparisons both in the supplementary material and in the video presentation.

5 Conclusion

We presented a framework for camera space hand mesh prediction, enabling learning directly in camera space. Our baseline and ablation studies validated our design choices, showing our method surpasses state-of-the-art approaches that predict hand meshes in camera-space coordinates. Estimating absolute 3D geometry from a single RGB image is inherently ill-posed. Rectifying images and predicting in camera space help reduce errors.

Our experiments show that while root-relative error is in the low single-digit millimeters, likely lower than annotation error, camera space error is 6 to 7 times larger. This is visually illustrated in Fig. 5 (b) and (c), suggesting a significant portion of total errors stems from 2D-to-3D depth ambiguity. We conjecture that isolating the hand from its context will soon reach a performance ceiling. Further research in new datasets and context-aware approaches, such as using scene geometry or objects, is needed to advance camera-space mesh inference.

Acknowledgements

We would like to thank Filippo Aleotti for his help with baseline experiments and infrastructure; Jamie Watson, Zawar Qureshi, and Jakub Powierza for their help with infrastructure; Axel Laguna for his insightful discussions on minimal solvers and network architectures; Daniyar Turmukhambetov for valuable technical discussions; and Gabriel Brostow, Sara Vicente, Jessica Van Brummelen, and Michael Firman for their valuable feedback on different versions of the manuscript.

References

1. Antotsiou, D., Garcia-Hernando, G., Kim, T.K.: Task-oriented hand motion retargeting for dexterous manipulation imitation. In: ECCV Workshop (2018)
2. Apple: Vision Pro. `https://www.apple.com/apple-vision-pro/`, [Online; accessed 7-March-2024]
3. Armagan, A., Garcia-Hernando, G., Baek, S., Hampali, S., Rad, M., Zhang, Z., Xie, S., Chen, M., Zhang, B., Xiong, F., et al.: Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction. In: ECCV (2020)
4. Baek, S., Kim, K.I., Kim, T.K.: Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In: CVPR (2019)
5. Baek, S., Kim, K.I., Kim, T.K.: Weakly-supervised domain adaptation via gan and mesh model for estimating 3D hand poses interacting objects. In: CVPR (2020)
6. Bhatnagar, B.L., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In: NeurIPS (2020)
7. Bhowmik, A., Gumhold, S., Rother, C., Brachmann, E.: Reinforced feature points: Optimizing feature detection and description for a high-level task. In: CVPR (2020)
8. Boukhayma, A., Bem, R.d., Torr, P.H.: 3d hand shape and pose from images in the wild. In: CVPR (2019)
9. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: CVPR (2017)
10. Cao, Z., Radosavovic, I., Kanazawa, A., Malik, J.: Reconstructing hand-object interactions in the wild. In: CVPR (2021)
11. Chen, B., Parra, A., Cao, J., Li, N., Chin, T.J.: End-to-end learnable geometric vision by backpropagating pnp optimization. In: CVPR (2020)
12. Chen, H., Wang, P., Wang, F., Tian, W., Xiong, L., Li, H.: Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In: CVPR (2022)
13. Chen, P., Chen, Y., Yang, D., Wu, F., Li, Q., Xia, Q., Tan, Y.: I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In: ICCV (2021)
14. Chen, X., Liu, Y., Dong, Y., Zhang, X., Ma, C., Xiong, Y., Zhang, Y., Guo, X.: Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In: CVPR (2022)
15. Chen, X., Liu, Y., Ma, C., Chang, J., Wang, H., Chen, T., Guo, X., Wan, P., Zheng, W.: Camera-space hand mesh recovery via semantic aggregation and adaptive 2D-1D registration. In: CVPR (2021)

16. Chen, X., Wang, B., Shum, H.Y.: Hand avatar: Free-pose hand animation and rendering from monocular video. In: CVPR (2023)
17. Chen, Y., Tu, Z., Kang, D., Bao, L., Zhang, Y., Zhe, X., Chen, R., Yuan, J.: Model-based 3d hand reconstruction via self-supervised learning. In: CVPR (2021)
18. Garcia-Hernando, G., Johns, E., Kim, T.K.: Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. In: IROS (2020)
19. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3d hand shape and pose estimation from a single rgb image. In: CVPR (2019)
20. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3D annotation of hand and object poses. In: CVPR (2020)
21. Hampali, S., Sarkar, S.D., Rad, M., Lepetit, V.: Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3D pose estimation. In: CVPR (2022)
22. Han, S., Liu, B., Cabezas, R., Twigg, C.D., Zhang, P., Petkau, J., Yu, T.H., Tai, C.J., Akbay, M., Wang, Z., et al.: Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. ACM TOG (2020)
23. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
24. Hasson, Y., Tekin, B., Bogo, F., Laptev, I., Pollefeys, M., Schmid, C.: Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In: CVPR (2020)
25. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C.: Learning joint reconstruction of hands and manipulated objects. In: CVPR (2019)
26. Huang, L., Lin, C.C., Lin, K., Liang, L., Wang, L., Yuan, J., Liu, Z.: Neural voting field for camera-space 3D hand pose estimation. In: CVPR (2023)
27. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. TPAMI (2013)
28. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5D heatmap regression. In: ECCV (2018)
29. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)
30. Karunratanakul, K., Spurr, A., Fan, Z., Hilliges, O., Tang, S.: A skeleton-driven neural occupancy representation for articulated hands. In: 3DV (2021)
31. Karunratanakul, K., Yang, J., Zhang, Y., Black, M.J., Muandet, K., Tang, S.: Grasping field: Learning implicit representations for human grasps. In: 3DV (2020)
32. Kulon, D., Guler, R.A., Kokkinos, I., Bronstein, M.M., Zafeiriou, S.: Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: CVPR (2020)
33. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)
34. Li, Z., Liu, J., Zhang, Z., Xu, S., Yan, Y.: Cliff: Carrying location information in full frames into human pose and shape estimation. In: ECCV (2022)
35. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: CVPR (2021)
36. Lin, K., Wang, L., Liu, Z.: Mesh graphormer. In: ICCV (2021)
37. Meta: Quest 3. `https://www.meta.com/us/quest/quest-3/`, [Online; accessed 7-March-2024]

38. Mihajlovic, M., Zhang, Y., Black, M.J., Tang, S.: Leap: Learning articulated occupancy of people. In: CVPR (2021)
39. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image. In: ICCV (2019)
40. Moon, G., Lee, K.M.: I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In: ECCV (2020)
41. Park, J., Oh, Y., Moon, G., Choi, H., Lee, K.M.: Handoccnet: Occlusion-robust 3d hand mesh estimation network. In: CVPR (2022)
42. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR (2019)
43. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: CVPR (2021)
44. Prince, S.J.: Computer vision: models, learning, and inference. Cambridge University Press (2012)
45. Remelli, E., Han, S., Honari, S., Fua, P., Wang, R.: Lightweight multi-view 3D pose estimation through camera-disentangled representation. In: CVPR (2020)
46. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM TOG (2017)
47. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: CVPR (2019)
48. Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., Kautz, J.: Weakly supervised 3d hand pose estimation via biomechanical constraints. In: ECCV (2020)
49. Tang, X., Wang, T., Fu, C.W.: Towards accurate alignment in real-time 3D hand-mesh reconstruction. In: ICCV (2021)
50. Wei, T., Patel, Y., Shekhovtsov, A., Matas, J., Barath, D.: Generalized differentiable ransac. In: ICCV (2023)
51. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3D: Towards zero-shot metric 3D prediction from a single image. In: ICCV (2023)
52. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J.Y., Lee, K.M., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al.: Depth-based 3D hand pose estimation: From current achievements to future goals. In: CVPR (2018)
53. Zhang, X., Huang, H., Tan, J., Xu, H., Yang, C., Peng, G., Wang, L., Liu, J.: Hand image understanding via deep multi-task learning. In: ICCV (2021)
54. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: ICCV (2019)
55. 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)
56. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., Brox, T.: FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In: ICCV (2019)