Neural Voting Field for Camera-Space 3D Hand Pose Estimation

Lin Huang[1*] Chung-Ching Lin[2] Kevin Lin[2] Lin Liang[2]
Lijuan Wang[2] Junsong Yuan[1] Zicheng Liu[2]
[1]University at Buffalo [2]Microsoft
https://linhuang17.github.io/NVF

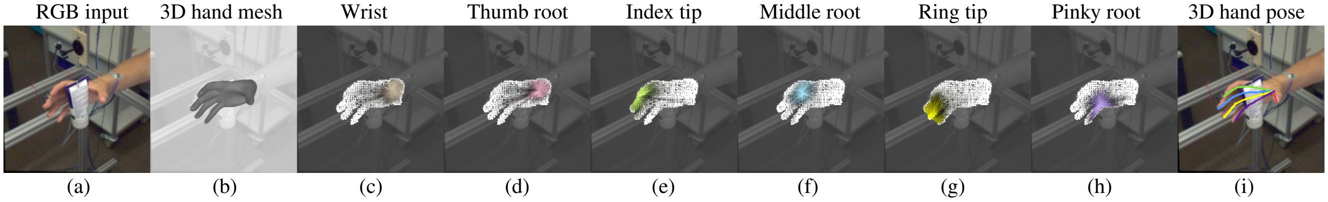

Figure 1. We present **Neural Voting Field (NVF)**, a 3D implicit function for camera-space 3D hand pose estimation from an RGB image. Given an RGB input (a), for a 3D point in camera space, NVF can predict its signed distance to hand surface and a set of 4D offset vectors (1D voting weight and 3D directional vector to each joint). 3D pose is then obtained via dense 3D point-wise voting. Rendered predictions: (b) Hand mesh generated by Marching Cubes from the predicted signed distances (c-h) Predicted 1D voting weight from each predicted near-surface 3D point to a joint. (Brighter line means larger weight, darker or no line means smaller weight) (i) Estimated 3D hand pose.

Abstract

We present a unified framework for camera-space 3D hand pose estimation from a single RGB image based on 3D implicit representation. As opposed to recent works, most of which first adopt holistic or pixel-level dense regression to obtain relative 3D hand pose and then follow with complex second-stage operations for 3D global root or scale recovery, we propose a novel unified 3D dense regression scheme to estimate camera-space 3D hand pose via dense 3D point-wise voting in camera frustum. Through direct dense modeling in 3D domain inspired by Pixel-aligned Implicit Functions for 3D detailed reconstruction, our proposed Neural Voting Field (NVF) fully models 3D dense local evidence and hand global geometry, helping to alleviate common 2D-to-3D ambiguities. Specifically, for a 3D query point in camera frustum and its pixel-aligned image feature, NVF, represented by a Multi-Layer Perceptron, regresses: (i) its signed distance to the hand surface; (ii) a set of 4D offset vectors (1D voting weight and 3D directional vector to each hand joint). Following a vote-casting scheme, 4D offset vectors from near-surface points are selected to calculate the 3D hand joint coordinates by a weighted average. Experiments demonstrate that NVF outperforms existing state-of-the-art algorithms on FreiHAND dataset for camera-space 3D hand pose estimation. We also adapt NVF to the classic task of root-relative 3D hand pose estimation, for which NVF also obtains state-of-the-art results on HO3D dataset.

1. Introduction

Monocular 3D hand pose estimation, which aims to recover 3D locations of hand joints from an RGB image, has attracted enormous attention and made remarkable progress in recent years. As a long-standing task in computer vision, it remains challenging due to its highly articulated structure, large variations in orientations, severe (self-)occlusion, and inherent 2D-to-3D scale and depth ambiguity.

Owing to the aforementioned difficulties, most existing works [4–7, 15, 19, 26, 32, 35, 36, 45, 52, 53, 59, 60, 63] focused on one aspect of this general problem, which is to estimate root-relative 3D hand pose (*i.e.*, 3D joint coordinates relative to a pre-defined root joint, such as hand wrist). While accurate 2D-to-3D root-relative pose estimation is essential for numerous applications in Virtual/Augmented Reality, there are various interactive tasks in which having root-relative hand joint coordinates alone is insufficient. For instance, being able to recover camera-space 3D hand joint coordinates in an AR view enables the user to directly use hands to manipulate virtual objects moving in 3D space.

To recover robust camera-space 3D hand pose, there are two key design elements: (1) the ability to exploit dense local evidence. Specifically, as demonstrated in previous works [14, 16, 25, 29, 34, 42, 43, 47, 49, 54–56], dense regression-based methods are more effective than holistic regression-based counterparts for handling highly articulated 3D pose structure, attributed to its ability to maintain the input data spatial structure and fully exploit local evidence; (2) the ability to reason 3D hand global geome-

[*] Work done during Lin Huang's internship with Microsoft.

Method	First Stage	Second Stage
Iqbal *et al.* [29]	2D-Dense	Scale Estimation
ObMan [22]	Holistic	Root Estimation
I2L-MeshNet [42]	1D-Dense	Root Depth Estimation
CMR [8]	2D-Dense+SpiralConv	Registration
Hasson *et al.* [21]	Holistic	Model Fitting
NVF (Ours)	Unified 3D-Dense	Weighted Average

Table 1. **Comparison of representative absolute 3D hand pose estimation schemes.** Please refer to Sec. 2.1 for more details.

try. As shown in previous literature [13, 29, 33], given 2D evidence and camera intrinsic parameters, reasonable understanding towards target object 3D structure/geometry is crucial to alleviate 2D-to-3D depth ambiguity, which is the key to accurately locate 3D hand pose in camera space.

To fully integrate both elements into our algorithm design in a unified manner, we connect with Pixel-aligned Implicit Function (PIFu) [24, 27, 49, 50, 62]. Through direct dense modeling in 3D domain with pixel-aligned local features, PIFu-based methods reconstruct highly detailed 3D human geometry from an RGB image in a unified way, showing its ability to model high-frequency local details such as clothing wrinkles while generating complete global geometry including largely occluded region such as the back of a person. Inspired by these results, we propose a novel unified 3D dense regression scheme based on a 3D implicit function for robust camera-space 3D hand pose estimation. Specifically, for each of the 3D query points densely sampled in camera frustum and its pixel-aligned image feature, unlike PIFu predicting occupancy value for each point, our proposed Neural Voting Field (NVF) regresses: (i) the signed distance between the point and the hand surface; (ii) a set of 4D offset vectors (1D voting weight and 3D directional vector from the point to each joint). Following a vote-casting scheme, 4D offset vectors from near-surface points (*i.e.*, points for which the predicted signed distance is below a threshold) are selected to calculate the 3D hand joint coordinates by a weighted average.

Most existing works for camera-space 3D hand pose estimation, as shown in Tab. 1, follow a two-stage estimation scheme. They first adopt holistic or pixel-level dense regression to obtain 2D and relative 3D hand poses and then follow with complex second-stage processing such as fitting, registration, using a separate network for 3D global root location or scale estimation. NVF instead provides a unified solution via direct dense modeling in 3D camera space followed by a simple weighted average operation, which enables reasoning about 3D dense local evidence and hand global geometry. As shown in Fig. 1, NVF makes solid 3D point-wise prediction and overall distribution of signed distance and voting weight even in highly occluded regions, leading to accurate camera-space pose estimation. In Sec. 4, we show that NVF noticeably outperforms two baselines based on holistic regression and 2D dense regression. Besides, NVF exhibits state-of-the-art performance for the task of camera-space 3D hand pose estimation on FreiHAND dataset. We also adapt NVF to the classic task of root-relative 3D hand pose estimation, for which NVF also achieves state-of-the-art results on HO3D dataset.

Since estimating absolute 3D pose from an RGB image is an ill-posed problem due to scale and depth ambiguity [13, 29], in Sec. 4.4, we also provide ablation analysis on hand scale based on results from NVF and the baselines.

This work makes the following contributions:

1. We propose Neural Voting Field (NVF), as the first 3D implicit representation-based unified solution to estimate camera-space 3D hand pose.
2. NVF follows a novel unified 3D dense regression scheme to estimate camera-space 3D hand pose via dense 3D point-wise voting in camera frustum.
3. NVF outperforms baseline methods based on holistic and 2D dense regression and achieves state-of-the-art results on absolute and relative hand pose estimation.

2. Related Work

2.1. Camera-Space 3D Hand Pose Estimation

As shown in Tab. 1, most existing works for monocular RGB input-based camera-space 3D hand pose estimation generally follow a two-stage estimation scheme. Iqbal *et al.* [29] recovers 3D hand pose in camera space up to a scaling factor by solving a quadratic equation given the camera intrinsic parameters and the estimated scale-normalized and root-relative 2.5D pose via 2D Dense regression. Extra global scale estimation is required for camera-space absolute 3D hand pose recovery. Another line of work [8, 21, 22] first obtains 2D pose or root-relative 3D hand pose estimations via 2D dense or holistic regression. Then, Hasson *et al.* [21] fits a 3D hand model to the initial estimations and recovers 3D hand pose in camera space from the input RGB video. Both ObMan [22] and CMR [8] work on a single RGB input and directly recover 3D global root from the initial 2D pose and root-relative 3D pose/mesh estimations. I2L-MeshNet [42] proposes a 1D dense regression scheme to predict per-lixel likelihood on 1D heat map for recovering root-relative 2.5D pose, and then uses a separate network (*i.e.*, RootNet [41]) for the absolute root depth estimation.

2.2. Root-relative 3D Hand Pose Estimation

For single RGB input-based root-relative 3D hand pose estimation, some early methods [52, 59, 63] adopt classic frameworks such as Convolutional Neural Network and Multi-Layer Perceptron in combination with holistic or dense regression scheme for scale-invariant and root-relative 3D hand joint positions recovery. After the introduction of the 3D parametric MANO [48] model, tremendous literature [3, 4, 9, 22, 45, 53, 60] attempts to integrate MANO into the learning pipeline and directly regress

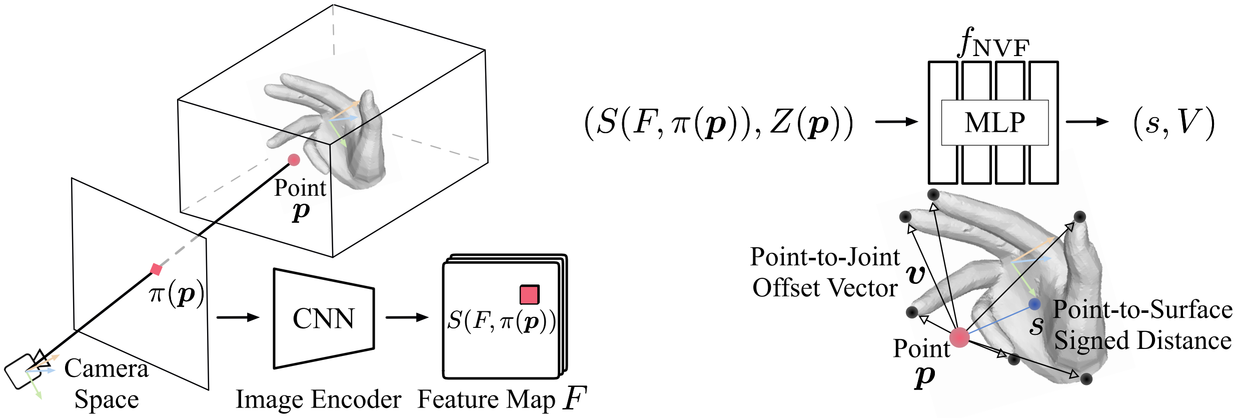

Figure 2. Overview of **Neural Voting Field (NVF)**: We first extract a feature map F from the input RGB image I by an image encoder. Then, for a 3D query point $\boldsymbol{p}$ sampled in camera space, we define an implicit function f_{NVF} realized by a MLP, which maps from pixel-aligned image feature $S(F,\pi(\boldsymbol{p}))$ and depth $Z(\boldsymbol{p})$ of $\boldsymbol{p}$ to its signed distance s and a set of 4D offset vectors V. Each 4D offset vector $\boldsymbol{v}$ contains a 1D voting weight and a 3D directional vector to each joint. The 3D hand pose is then obtained by dense 3D point-wise voting.

MANO parameters from the image. Meanwhile, MANO model-free approaches [6] continue to develop and more works also start to investigate the dependencies among hand joints as hands are inherently structured. Thus, geometric deep learning-based models (*e.g.*, Graph Convolution, Spiral Convolution) [5, 7, 8, 15, 32, 53] and Transformer-based models [19, 35, 36, 45] have also been proposed.

2.3. 3D Implicit Representation

Recent methods [1, 2, 10, 17, 37, 40, 46, 51] have shown that, as a continuous and differentiable function realized by Multi-Layer Perceptron, memory efficient deep implicit function can represent detailed 3D shapes with arbitrary topology and have no resolution limitations. It can also be extended to RGB image-based 3D reconstruction [40, 44, 46, 57]. Grasping Field [30] proposes to reconstruct hands grasping object via predicting signed distances for hand and object given the input point cloud or RGB image. Unlike previous implicit functions relying on a global feature vector, given an input RGB image with cleanly segmented foreground, Pixel-aligned Implicit Function [49] (PIFu) utilizes pixel-aligned image feature and depth value of each 3D point for occupancy prediction. PIFu-based methods have shown the ability to generate high-fidelity 3D human geometry with more clothes details [27, 31, 50, 62] and robust 6D pose estimation. For instance, Neural Correspondence Field [24] predicts signed distance and 3D object coordinate for each 3D point to establish dense 3D-3D correspondences for robust 6D object pose estimation.

3. The Proposed Method

3.1. Preliminaries

Pixel-Aligned Implicit Function (PIFu) [49]. PIFu, by learning an implicit function in 3D domain with pixel-aligned features, generates detailed 3D human geometry including largely occluded regions from a segmented RGB image. Specifically, given a segmented image $I:\mathbb{R}^2\mapsto\mathbb{R}^3$, PIFu first extracts a feature map $F:\mathbb{R}^2\mapsto\mathbb{R}^C$ with C channels by a fully convolutional image encoder. Then, for a 3D point $\boldsymbol{p}\in\mathbb{R}^3$ in the camera space and its 2D projection $\pi(\boldsymbol{p}):\mathbb{R}^3\mapsto\mathbb{R}^2$ on I, the implicit function, represented by a Multi-Layer Perceptron (MLP) $f_{\text{PIFu}}:\mathbb{R}^C\times\mathbb{R}\mapsto\mathbb{R}$ as $f_{\text{PIFu}}(S(F,\pi(\boldsymbol{p})),Z(\boldsymbol{p}))=y$ maps the pixel-aligned feature $S(F,\pi(\boldsymbol{p}))$ and the depth $Z(\boldsymbol{p})$ of $\boldsymbol{p}$ to the occupancy $y\in[0,1]$ (1 means $\boldsymbol{p}$ is inside the human surface and 0 means otherwise). S is the sampling function (*e.g.*, bilinear interpolation) to sample the value of F at pixel $\pi(\boldsymbol{p})$.

Signed Distance Function (SDF) [12, 46]. Given a 3D shape $M\in\mathcal{M}$, a signed distance function is a continuous function $\phi:\mathbb{R}^3\times\mathcal{M}\mapsto\mathbb{R}$ as $\phi(\boldsymbol{p},M)=s$, mapping a 3D point $\boldsymbol{p}$ to the signed distance s between $\boldsymbol{p}$ and the shape surface. The sign denotes if $\boldsymbol{p}$ is outside (positive) or inside (negative) of a watertight surface, with the iso-surface $\phi(\boldsymbol{p},M)=0$ implicitly representing the surface.

3.2. Dense Offset-based Pose Re-Parameterization

As opposed to holistic regression directly mapping the input data to sparse and absolute joint positions and 2D dense regression which mainly exploits the 2D input local evidence, our method aims to directly model in 3D domain and thus fully exploit the 3D dense local evidence and hand global structure. To achieve this goal, given a common sparse and explicit pose representation (*i.e.*, a set of 3D coordinates of T hand joints) as $J=\{\boldsymbol{j}_t\}_{t=1}^{T}\in\mathcal{J},\boldsymbol{j}_t\in\mathbb{R}^3$, we parameterize it as a set of dense 3D point-wise offset-based representation. As shown in previous works [16, 54], it is non-ideal to directly learn the 3D offset vector from each 3D point to a joint due to the large variance of offsets and far away joints are also beyond the scope of receptive field. Thus, given a camera-space 3D hand pose J and shape M, we define the pose parameterization as a function ψ, mapping from a 3D point $\boldsymbol{p}$ in the camera space to a set of 4D offset vectors $V = \{\boldsymbol{v}_t\}_{t=1}^{T}$, $\boldsymbol{v}_t \in \mathbb{R}^4$:

$$\psi : \mathbb{R}^3 \times \mathcal{J} \times \mathcal{M} \mapsto \mathbb{R}^{T \times 4} \quad \text{as} \quad \psi(\boldsymbol{p}, J, M) = V. \tag{1}$$

Each offset vector $\boldsymbol{v}_t = (w_t, \boldsymbol{d}_t)$ consists of a 1D voting weight $w_t \in \mathbb{R}$ as point-wise closeness to hand joint j_t:

$$w_t = \begin{cases} 1 - \frac{\|\boldsymbol{j}_t - \boldsymbol{p}\|_2}{r} & |s| < \delta \text{ and } \|\boldsymbol{j}_t - \boldsymbol{p}\|_2 \leq r \text{ and } \boldsymbol{p} \in B_t^K, \\ 0 & \text{otherwise;} \end{cases} \tag{2}$$

and a 3D unit directional vector $\boldsymbol{d}_t \in \mathbb{R}^3$ as point-wise direction to hand joint j_t:

$$\boldsymbol{d}_t = \begin{cases} \frac{\boldsymbol{j}_t - \boldsymbol{p}}{\|\boldsymbol{j}_t - \boldsymbol{p}\|_2} & |s| < \delta \text{ and } \|\boldsymbol{j}_t - \boldsymbol{p}\|_2 \leq r \text{ and } \boldsymbol{p} \in B_t^K, \\ 0 & \text{otherwise;} \end{cases} \tag{3}$$

where $s = \phi(\boldsymbol{p}, M)$ and $|s| < \delta$ means points that are within clamping distance δ from the hand surface. r denotes the radius of a 3D ball centered at each joint and B_t^K denotes the set of K-Nearest Neighbors (KNN) to joint $\boldsymbol{j}_t$ among all the near-surface points (*i.e*., points with $|s| < \delta$). Thus, among all the 3D points that are in the hand surface vicinity, we assign the non-zero 4D offset vector $\boldsymbol{v}_t$ to the K nearest points inside the 3D ball with radius r of joint $\boldsymbol{j}_t$.

3.3. Neural Voting Field

To fully exploit the 3D local evidence and hand global geometry for camera-space 3D hand pose estimation, based on the aforementioned signed distance function and pose parameterization, we unify both modalities using a pixel-aligned implicit function. Following the proposed pipeline in Fig. 2, we first extract a C-channel feature map F from the input RGB image I by a hourglass network $g(I; \boldsymbol{\eta})$ as used in [24, 49]. Then, for a 3D query point $\boldsymbol{p}$ sampled in camera space, we define a continuous implicit function f_{NVF} realized by a MLP, which maps from pixel-aligned image feature $S(F, \pi(\boldsymbol{p}))$ and depth $Z(\boldsymbol{p})$ of point $\boldsymbol{p}$ to its signed distance s and a set of 4D offset vectors V:

$$\begin{aligned} &f_{\text{NVF}} : \mathbb{R}^C \times \mathbb{R} \mapsto \mathbb{R} \times \mathbb{R}^{T \times 4} \quad \text{as} \\ &f_{\text{NVF}}\big(S(F, \pi(\boldsymbol{p})), Z(\boldsymbol{p}); \boldsymbol{\theta}\big) = \big(s, V\big), \end{aligned} \tag{4}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ are the parameters of the hourglass g and MLP f_{NVF}, respectively. We use a pinhole camera model for 3D-to-2D projection. For each 3D point, besides the set of 4D offset vectors, NVF also predicts its signed distance to the hand surface, helping to learn the global hand geometry and also to select near-surface 3D points by thresholding the signed distances. The near-surface 3D points are then considered as valid 3D voters to be used in the voting stage.

Optimization. During training, we sample N 3D points $P = \{\boldsymbol{p}_n\}_{n=1}^{N}$ in camera frustum for each image and apply direct supervision to both predicted signed distance s_n as L_s and the predicted set of 4D offset vectors V_n as L_V. Given the ground-truth signed distance $\hat{s}_n = \phi(\boldsymbol{p}_n, \hat{M})$ and set of offset vectors $\hat{V}_n = \psi(\boldsymbol{p}_n, \hat{J}, \hat{M})$, where $\hat{J}$ and $\hat{M}$ are the ground-truth pose and mesh, L_s and L_V are defined as:

$$L_s = \frac{1}{N} \sum_{n=1}^{N} \big| \text{clamp}\left(\hat{s}_n, \delta\right) - \text{clamp}\left(s_n, \delta\right) \big|, \tag{5}$$

$$L_V = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\left(|\hat{s}_n| < \delta\right) H\left(\hat{V}_n, V_n\right), \tag{6}$$

where $\text{clamp}(s, \delta) := \min(\delta, \max(-\delta, s))$ uses δ to control the distance from the hand surface over which we want to main a metric SDF, following [46]. H is the Huber loss [28] and $\mathbb{1}(\cdot)$ is an indicator function that selects near-surface points (*i.e*., points that are within clamping distance δ from the hand surface). Then, We jointly train the hourglass network g and MLP f_{NVF} end-to-end by solving the following optimization problem with λ as the balancing weight:

$$\boldsymbol{\eta}^{\star}, \boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\eta}, \boldsymbol{\theta}}{\arg\min}\, L_s + \lambda L_V. \tag{7}$$

Sampling of 3D Query Points. During training, unlike voxel-based methods which require discretization of ground-truth 3D meshes, we directly sample 3D points on the fly from the ground-truth mesh following [24, 49]. To make NVF fully exploit meaningful local evidence and understand the hand global geometry, it is crucial to focus on the hand by sampling the query points more densely around the hand surface during training. Thus, given the 3D camera-space hand model, we first sample 12500 points nearby the surface, 1000 points inside the bounding sphere of the model, and 1000 points inside the camera frustum. Among all these points, we then sample 2500 points inside the model and 2500 points outside to form a set of 5000 points in total for training. During inference, the points are directly sampled at centers of voxels that fill up the same camera frustum without any ground-truth information used.

3.4. Dense 3D Point-to-Joint Voting.

During inference, assuming N 3D points $P = \{\boldsymbol{p}_n\}_{n=1}^{N}$ sampled in camera frustum for each image, for each point $\boldsymbol{p}_n$ that is considered as a valid 3D voter (*i.e*., the predicted signed distance s_n is below the threshold δ), based on its predicted set of 4D offset vectors $V_n = \{\boldsymbol{v}_t^n\}_{t=1}^{T}$, we first convert each 4D offset vector $\boldsymbol{v}_t^n = (w_t^n, \boldsymbol{d}_t^n)$ into the actual 3D offset $\boldsymbol{o}_t^n$ from $\boldsymbol{p}_n$ to joint $\boldsymbol{j}_t$ based on Eq. 2 and Eq. 3:

$$\boldsymbol{o}_t^n = \mathbb{1}\left(|s_n| < \delta\right) \left[r(1 - w_t^n)\boldsymbol{d}_t^n\right]. \tag{8}$$

Then, each camera-space joint location $\boldsymbol{j}_t$ is calculated simply by a weighted average over each voter's 3D offset prediction:

$$\boldsymbol{j}_t = \sum_{n=1}^{N} \frac{\mathbb{1}\left(|s_n| < \delta\right) w_t^n (\boldsymbol{o}_t^n + \boldsymbol{p}_n)}{\sum_{n=1}^{N} \mathbb{1}\left(|s_n| < \delta\right) w_t^n}. \tag{9}$$

Finally, we obtain the final camera-space 3D hand pose as $J = \{\boldsymbol{j}_t\}_{t=1}^{T}$. Furthermore, instead of using all the near-surface points as valid voters, we can also only focus on using near-surface points with high predicted voting weight, as a higher voting weight means the point is more confident and closer to a hand joint. Therefore, we use voting fraction as a parameter to control the fraction of near-surface points with the highest voting weight for voting. Related ablation studies will be given in Sec. 4.7.

4. Experiments

4.1. Baseline Methods

To demonstrate the effectiveness of our proposed 3D dense regression scheme against classic holistic regression and 2D dense regression, we design two directly comparable baselines that share the same architecture of the hourglass network and the MLP as NVF described in Sec. 3. Specifically, similar to NVF, both baselines first extract C-channel feature map from the input RGB image I. Then, unlike our f_{NVF} which takes a sampled 3D query point's pixel-aligned feature $S(F, \pi(\boldsymbol{p}))$ and its depth $Z(\boldsymbol{p})$ to predict its signed distance s and the set of 4D offset vectors V, the two baselines take the following steps.

Baseline-Holistic. We directly apply a global average pooling α to its feature map F' extracted by the hourglass network $g(I; \boldsymbol{\eta'})$ and generate a C-dimension feature vector $\alpha(F')$. The feature vector is then passed through the MLP $f_{\text{HOL}} : \mathbb{R}^C \mapsto \mathbb{R}^{T\times 3}$ as $f_{\text{HOL}}(\alpha(F'); \boldsymbol{\theta'}) = J'$. f_{HOL} directly outputs T 3D hand joint coordinates $J' = \{\boldsymbol{j}'_t\}_{t=1}^{T}$ in camera space. Given the ground-truth pose $\hat{J}$, it is trained by solving the following optimization problem with a loss function $L_{J'} = H(\hat{J}, J')$:

$$\boldsymbol{\eta'}^{\star}, \boldsymbol{\theta'}^{\star} = \operatorname*{arg\,min}_{\boldsymbol{\eta'}, \boldsymbol{\theta'}} L_{J'}. \tag{10}$$

Baseline-2D-Dense. We adopt a 2D dense regression scheme to estimate the camera-space pose via dense 2D pixel-wise voting, which is different from NVF which is based on dense 3D point-wise voting. Given the feature map F'' extracted by the hourglass network $g(I; \boldsymbol{\eta''})$, we only pass the pixel-aligned feature $F''(\boldsymbol{u})$ at each 2D pixel location $\boldsymbol{u}$ through the MLP $f_{\text{DEN}} : \mathbb{R}^C \mapsto \mathbb{R} \times \mathbb{R}^{T\times 4}$ as $f_{\text{DEN}}(F''(\boldsymbol{u}); \boldsymbol{\theta''}) = (e, V'')$. f_{DEN} outputs the probability $e \in [0, 1]$ that the hand is present at $\boldsymbol{u}$ and also a set of 4D vectors $V'' = \{\boldsymbol{v}''_t\}_{t=1}^{T}$. Each 4D vector $\boldsymbol{v}''_t = (w''_t, \boldsymbol{j}''_t)$ contains a 1D voting weight w''_t and 3D joint coordinate $\boldsymbol{j}''_t$. For the 1D voting weight, it is defined similarly to the one in NVF. We first use the camera ray for each pixel to find its first 3D intersection point with the 3D hand model in the camera space. Then, we directly follow Eq. 2 to calculate the pixel's voting weight as the closeness to each joint. It is trained by solving the following optimization problem with λ'' as the balancing weight:

$$\boldsymbol{\eta''}^{\star}, \boldsymbol{\theta''}^{\star} = \operatorname*{arg\,min}_{\boldsymbol{\eta''}, \boldsymbol{\theta''}} L_e + \lambda'' L_{V''} \tag{11}$$

$$= \operatorname*{arg\,min}_{\boldsymbol{\eta''}, \boldsymbol{\theta''}} \frac{1}{L}\Big[\sum_{l=1}^{L} E\left(\hat{e}_l, e_l\right) + \lambda'' \sum_{l=1}^{L} \hat{e}_l H\left(\hat{V}''_l, V''_l\right)\Big], \tag{12}$$

where L is the number of pixels, E is the binary cross-entropy loss, $\hat{e}_l$ is given by the ground-truth hand mask, and $\hat{V}''_l$ is the ground-truth set of 4D vectors. During inference, for pixel $\boldsymbol{u}_l$ with predicted probability $e_l > 0.5$ (*i.e.*, predicted as the foreground pixel), its predicted voting weight and 3D joint location are used to calculate the camera-space pose $J'' = \{\boldsymbol{j}''_t\}_{t=1}^{T}$ by a weighted average, similar to Eq. 9.

4.2. Dataset and Evaluation Metrics

FreiHAND Dataset [64]. As a real 3D hand dataset, it provides 130,240 training and 3,960 evaluation samples. The result of the evaluation set is evaluated via the official server for online evaluation. Since FreiHAND provides hand scale during evaluation time, we provide an ablation on hand scale for analysis of 2D-to-3D scale and depth ambiguity.

Metrics. For FreiHAND, we evaluate for the task of camera-space 3D hand pose estimation. As in [8,64], we report the unaligned version of Mean Joint Error (mm) which is measured in camera space as **CS-MJE** and the Area-Under-the-Curve of PCK (Percentage of Correct Keypoints) vs. error thresholds as **CS-AUC** to evaluate the absolute 3D hand pose. We also define and report a 3D Translational Error (mm) **(TE)** [23] and Depth Error (mm) **(DE)** to evaluate the predicted and ground-truth pose's centroid and the depth-axis of the centroid, respectively. The results of **TE** and **DE** will be used in analysis for baseline studies, especially for the analysis of scale and depth ambiguity.

HO3D Dataset [18]. As a real 3D hand-object dataset, it contains 66,034 training samples and 11,524 evaluation samples. We use the official server for online evaluation as the annotations of the evaluation set are not available.

Metrics. For HO3D, we evaluate for the task of root-relative 3D hand pose estimation. As in [18, 19], we report the Mean Joint Error (mm) after scale-translation alignment of the root as **MJE** and corresponding **AUC** to evaluate the pose. We also follow [45] to add ground-truth root to the predicted root-relative pose and report error (mm) as **RS-MJE** as it is equivalent to measuring in root-relative space.

Complement (Comp) Dataset [7]. As a synthetic 3D hand dataset, it provides 1,520 poses and 71 viewpoints for research use. We only use it for pre-training when needed.

4.3. Implementation Details

For NVF and two baselines, the network architecture is adopted from PIFu [49]. Specifically, we use hourglass net-

Method	Extra Data	Hand Crop	Hand Scale	CS-MJE↓
ObMan [22]	-	✓	✗	85.2
MANO CNN [64]	-	✓	✗	71.3
I2L-MeshNet [42]	-	✓	✗	60.3
CMR-SG-RN18 [8]	-	✓	✗	49.7
CMR-SG-RN50 [8]	-	✓	✗	48.8
Baseline-Holisitc	-	✗	✗	54.5
Baseline-2D-Dense	-	✗	✗	53.2
CS-NVF (Ours)	-	✗	✗	**47.2**
Baseline-Holisitc	-	✗	✓	50.4
Baseline-2D-Dense	-	✗	✓	49.0
CS-NVF (Ours)	-	✗	✓	**42.4**
CS-NVF (Ours)	Comp*	✗	✗	**44.6**
CS-NVF (Ours)	Comp*	✗	✓	**39.3**

Table 2. **Comparison for absolute 3D hand pose on FreiHAND.** *: pre-training on Comp. Note that our CS-NVF and two baselines take the original image without hand detection or cropping.

work with the output stride of 4 and output channel of 256 as the image encoder. All the MLPs f_{NVF}, f_{HOL}, and f_{DEN} have four hidden fully-connected layers with the number of neurons as $(1024, 512, 256, 128)$. Unless stated otherwise, during training for NVF, the clamping distance δ is set to 5 mm. The KNN is set to 1024. The ball radius r is set to 80 mm. During inference, the sampling step size of 3D points is 16 mm in all three axes. The voting fraction is set to 50%. Unlike PIFu, no image segmentation is applied. Camera intrinsic parameters are assumed known. Base on the above setting, for camera-space 3D hand pose estimation, we directly sample 3D points in camera space, as described in Sec. 3.3. We call NVF working in camera space as CS-NVF. Unlike other methods in Tab. 2, CS-NVF and both baselines take the original image as input without hand detection or cropping. For the task of root-relative 3D hand pose estimation, we follow similar sampling strategy but sample in a hand root-relative 3D cube instead of the camera frustum. We then call NVF working in root-relative space as RS-NVF. RS-NVF takes cropped hand-centered image. Please refer to the supplementary for more details.

4.4. Baseline Studies

In this section, based on the results from CS-NVF and both baselines (as in Tab. 2, Tab. 3, Fig. 3, and Fig. 4), we investigate the impact of 2D-to-3D scale and depth ambiguity for the task of absolute 3D hand pose estimation and the effectiveness of using 3D dense regression over both 2D dense regression and holistic regression.

2D-to-3D Scale and Depth Ambiguity. Estimating absolute 3D pose from an RGB image is an ill-posed problem due to 2D-to-3D scale and depth ambiguity [13, 29]. In view of this mater, FreiHAND [64] provides hand scale during evaluation for participants to use. As most existing works do not use the provided hand scale and thus, no relevant ablation on hand scale exists, we provide ablation results on hand scale in Tab. 2 and Tab. 3 for CS-NVF and two baselines. First, based on results in Tab. 2 and Tab. 3, we observe that, for all three pipelines, the main cause of the error between the ground-truth and predicted hand joints is the global translation and especially the depth component as TE is close to CS-MJE and DE is close to TE. For instance, CS-NVF with hand scale and Comp pre-training achieves 39.3 mm CS-MJE, while TE and DE are 36.5 mm and 35.5 mm, respectively. This validates the fact that depth ambiguity is indeed one of the main challenges for estimating accurate absolute 3D hand pose from a single RGB image. Moreover, for CS-NVF and the two baselines, using hand scale leads to significant improvements in terms of all three metrics. It is also worth noting that, for all three pipelines, results only using hand scale outperform results using only extra data. For instance, CS-NVF with hand scale achieves 42.4 mm CS-MJE, while CS-NVF with Comp pre-training

Method	Extra Data	Hand Scale	TE↓	DE↓
Baseline-Holisitc	-	✗	50.6	49.1
Baseline-2D-Dense	-	✗	49.2	47.9
CS-NVF (Ours)	-	✗	**43.6**	**42.4**
Baseline-Holisitc	-	✓	46.9	45.5
Baseline-2D-Dense	-	✓	45.3	43.9
CS-NVF (Ours)	-	✓	**38.9**	**37.8**
Baseline-Holisitc	Comp*	✗	48.7	47.1
Baseline-2D-Dense	Comp*	✗	47.9	46.4
CS-NVF (Ours)	Comp*	✗	**41.5**	**40.4**
Baseline-Holisitc	Comp*	✓	41.7	40.1
Baseline-2D-Dense	Comp*	✓	40.5	38.8
CS-NVF (Ours)	Comp*	✓	**36.5**	**35.5**

Table 3. **Comparison of 3D Translational and Depth Error for absolute 3D hand pose on FreiHAND.** *: pre-training on Comp.

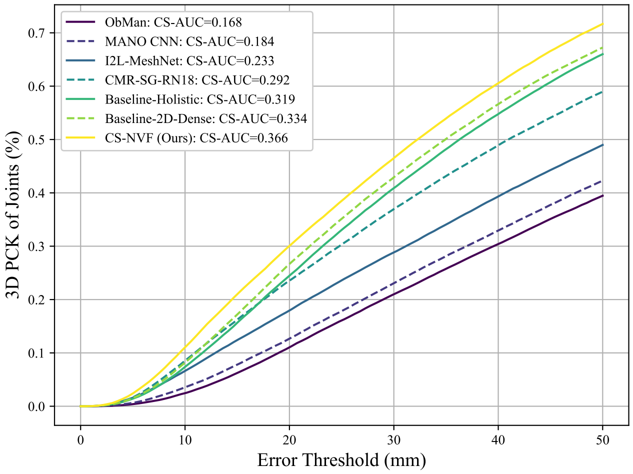

Figure 3. **3D PCK for absolute 3D hand pose on FreiHAND.**

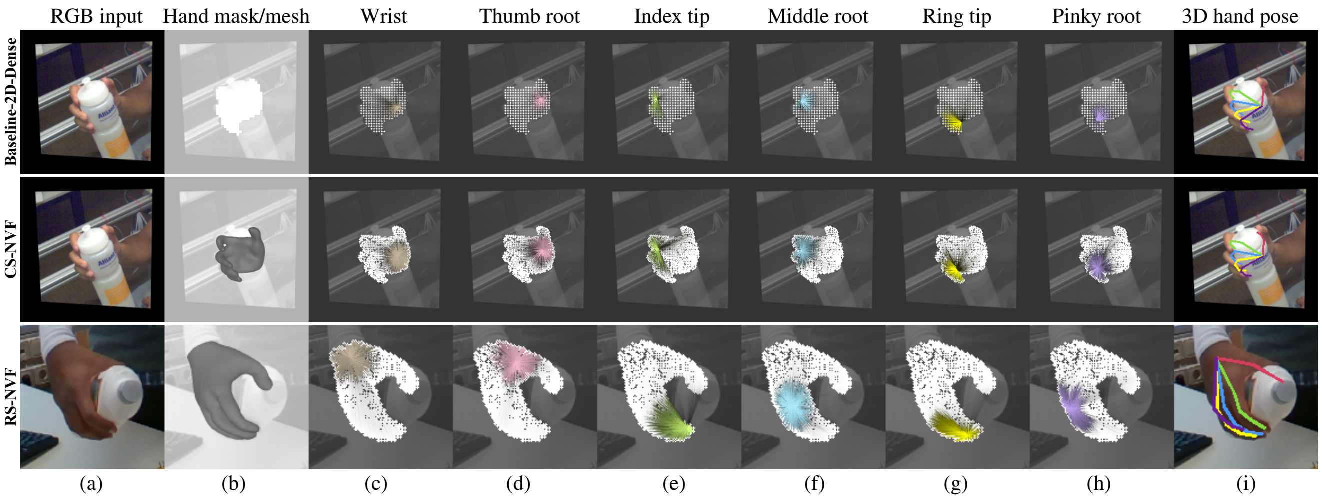

Figure 4. **Qualitative results for absolute and relative 3D pose:** (b) Predicted hand mask (Baseline) or mesh generated by Marching Cubes from the predicted signed distances (NVF). (c-h) Predicted 1D voting weight from each predicted foreground pixel (Baseline) or predicted near-surface 3D point (NVF) to a joint. (Brighter line means larger weight, darker or no line means smaller weight) (i) Estimated 3D pose. NVF makes solid point-wise prediction and overall distribution of signed distances and voting weights in highly occluded region.

Method	MJE↓	AUC↑	RS-MJE↓
Hasson et al. [20]	36.9	0.369	-
Pose2Mesh [11]	33.3	0.480	33.2
ObMan [22]	31.8	0.461	55.2
Liu et al. [38]	31.7	0.463	30.0
Hampali et al. [18]	30.4	0.494	-
METRO [35]	28.9	0.504	-
I2L-MeshNet [42]	26.0	0.529	26.8
Keypoint Trans. [19]	25.7	0.553	-
ArtiBoost [58]	25.3	0.532	-
Zheng et al. [61]	25.1	0.541	-
HandOccNet [45]	24.0	0.557	24.9
RS-NVF (Ours)	**21.8**	**0.610**	**23.2**

Table 4. **Comparison for relative 3D hand pose on HO3D.**

has 44.6 mm CJ-MJE. These verify that a reasonable understanding towards the hand scale would help for robust depth learning and alleviate depth ambiguity, which is also aligned with existing literature [13, 29, 33].

Dense Regression vs. Holistic Regression. In Tab. 2 and Fig. 3, we can observe that both CS-NVF and Baseline-2D-Dense consistently outperform Baseline-Holistic on CS-MJE and CS-AUC under different training settings. For instance, without using hand scale, Baseline-2D-Dense is 1.3 mm better than Baseline-Holistic in CS-MJE, and CS-NVF can be 7.3 mm better. Since three pipelines share the same network architecture, this supports our point that dense regression is more effective than holistic regression for handling highly articulated pose data.

3D Dense Regression vs. 2D Dense Regression. Quantitatively, in Tab. 2 and Fig. 3, it can be observed that CS-NVF consistently outperforms Baseline-2D-Dense by a large margin on CS-MJE and CS-AUC under different training settings. For instance, without using hand scale, CS-NVF can be 6 mm better on CS-MJE. Qualitatively, as indicated by the white circles in column (c-h) of Fig. 4, both methods are able to find valid voters (Baseline-2D-Dense: pixels with predicted probability $e > 0.5$. CS-NVF: 3D points with predicted signed distance $|s| < \delta$). However, CS-NVF generally provides not only more number of valid voters, but also more reasonable element-wise prediction and overall distribution of the voting weight than Baseline-2D-Dense. Specifically, for the number of voters, since CS-NVF models all the 3D points around the hand surface and Baseline-2D-Dense can only model foreground pixels, CS-NVF generally can find and use more valid voters. As for element-wise prediction and overall distribution of the voting weight, CS-NVF is generally able to predict large voting weight for points close to hand joint and small value for points far away, resulting in well-shaped symmetric distribution around the corresponding joint, even in occluded regions. Note that brighter line indicates larger voting weight, darker or no line indicates smaller voting weight. Moreover, as shown in column (b) for the mesh rendering of CS-NVF (obtained using Marching Cubes [39] from predicted signed distances), even in highly occluded region, CS-NVF provides solid signed distance distribution showing its ability to reason the global hand geometry. Overall, these demonstrate that CS-NVF, through direct dense modeling in 3D domain, can better model 3D dense local evidence, the relation between voters and 3D hand pose, and also the global hand geometry, compared with 2D dense regression-based counterpart, which leads to robust 3D hand pose estimation.

4.5. Main Results: Absolute 3D Hand Pose

In Tab. 2 and Fig. 3, we also compare CS-NVF with current state-of-the-art methods on FreiHAND for absolute 3D

Proposed	Evaluation	Voting Fraction (%)				Clamping Distance (mm)				KNN			
Method	Metric	25	50	75	100	1	5	10	20	256	512	1024	2048
CS-NVF	CS-MJE↓	39.9	39.3	39.5	40.2	39.3	39.3	40.0	40.1	40.3	39.5	39.3	39.3
(Ours)	CS-AUC↑	0.36	0.37	0.36	0.36	0.37	0.37	0.36	0.36	0.35	0.36	0.37	0.37
RS-NVF	MJE↓	22.2	21.8	21.9	22.0	21.9	21.8	21.8	21.9	23.6	22.6	21.8	21.8
(Ours)	AUC↑	0.60	0.61	0.61	0.60	0.60	0.61	0.61	0.61	0.59	0.60	0.61	0.61

Table 5. **Ablation study on parameter choices for absolute and relative 3D hand pose:** Voting fraction, clamping distance, and KNN

Step	CS-NVF (Ours)			
Size	#Points	CS-MJE↓	CS-AUC↑	FPS↑
4	1882k	39.2	0.37	1
8	230k	39.2	0.37	8
16	28k	39.3	0.37	34
24	8k	39.8	0.36	50
32	3k	42.4	0.34	56

Table 6. **Ablation study on sampling step size for absolute 3D hand pose on FreiHAND.** #points: number of sampled 3D points.

hand pose estimation. In Tab. 2, without the use of hand scale and extra data as a fair comparison, CS-NVF surpasses previous state-of-the-art methods, achieving 1.6 mm better than CMR [8]. Note that, unlike other methods, CS-NVF takes the original image as input without hand detection or cropping. Additional hand scale information used without extra data improves the CS-MJE of CS-NVF further to 42.4 mm. Additional Comp pre-training improves the CS-MJE of CS-NVF to 44.6 mm without hand scale and 39.3 mm with hand scale. Referring to Fig. 3, CS-NVF also achieves state-of-the-art performance on CS-AUC.

4.6. Main Results: Relative 3D Hand Pose

In Tab. 4, we compare RS-NVF with current state-of-the-art methods on HO3D for relative 3D hand pose estimation. RS-NVF significantly outperforms all these methods in all three metrics. Especially, RS-NVF outmatches HandOccNet [45] by 2.2 mm on MJE, 1.7 mm on RS-MJE, and also has 0.610 AUC agaist 0.557 AUC from HandOccNet.

4.7. Ablation Studies

Sampling Step Size. During inference, 3D query points are sampled at centers of voxels that fill up the 3D space. Since the choices of different sampling step sizes can influence both accuracy and efficiency, in Tab. 6, we show the ablation on sampling step size for CS-NVF. With 16 mm as the step size, CS-NVF achieves 34 FPS and 39.3 mm on CS-MJE given 28k points. Reducing the step size to 4 mm yields 1,882k points and 1 FPS with slightly improved CS-MJE of 39.2 mm. Increasing to 32 mm step size yields 3k points and improves the FPS to 56 with still competitive 42.4 mm on CS-MJE. For RS-NVF not shown in the table, 4 and 8 mm step size both yield MJE of 21.7 mm and AUC of 0.62. We can observe that, while step size affects FPS, NVF is relatively insensitive regarding the performance. In all other reported results, we use 16 mm as the default step size.

Voting Fraction. Since the predicted point-wise voting weight reflects the confidence and closeness of a given point towards a certain joint, besides using all the near-surface points as valid voters, we can use a specified fraction of near-surface points with the highest voting weight for voting during inference. In Tab. 5, we observe that using all the near-surface points (100%) yields competitive results while 50% fraction leads to the overall better performance for both CS-NVF and RS-NVF. The results are aligned with our algorithm design since points far from each joint are not expected to make meaningful contributions. In all other reported results, we use 50% as the default voting fraction.

Clamping Distance. During training, clamping distance δ is used to concentrate NVF's capacity on learning high-quality point-wise prediction (*i.e.*, signed distance and set of 4D offset vectors) near the hand surface. Ablation in Tab. 5 shows that 5 mm generates the overall better performance for both CS-NVF and RS-NVF. In all other reported results, we use 5 mm as the default clamping distance.

K-Nearest Neighbors (KNN). During training, among all the sampled points that are in the hand surface vicinity, non-zero 4D offset vector will only be assigned to the K nearest points inside a 3D ball of each hand joint. As shown in previous works [16, 54], estimating offsets for large KNN is unnecessary and far away joints are beyond the scope of receptive field. Small KNN might also cause poor performance due to the sparse learning scheme. Ablation in Tab. 5 shows that 1024 KNN generates the overall better performance for both CS-NVF and RS-NVF. In all other reported results, we use 1024 as the default KNN.

5. Conclusions

We propose Neural Voting Field (NVF), as the first 3D implicit representation-based unified solution to estimate camera-space 3D hand pose given a single RGB image. NVF follows a novel unified 3D dense regression scheme to estimate 3D hand pose in 3D camera space via dense 3D point-wise voting in camera frustum. Studies with both holistic regression baseline and 2D dense regression baseline verify NVF's ability to fully exploit 3D dense local evidence and model global hand structure/geometry, which are essential for precisely locating 3D hand joints in 3D camera space. NVF also demonstrates state-of-the-art results on both tasks of camera-space 3D hand pose estimation and root-relative 3D hand pose estimation.

References

[1] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *CVPR*, 2020. 3

[2] Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. In *ICLR*, 2021. 3

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

[4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 1, 2

[5] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, 2019. 1, 3

[6] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *ICCV*, 2021. 1, 3

[7] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, 2022. 1, 3, 5

[8] 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 *CVPR*, 2021. 2, 3, 5, 6, 8

[9] 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 *CVPR*, 2021. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 3

[11] 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 *ECCV*, 2020. 7

[12] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *SIGGRAPH*, 1996. 3

[13] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019. 2, 6, 7

[14] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *ECCV*, 2020. 1

[15] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019. 1, 3

[16] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, 2018. 1, 3, 8

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. 3

[18] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 5, 7

[19] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, 2022. 1, 3, 5, 7

[20] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 7

[21] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from rgb videos. In *3DV*, 2021. 2

[22] 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, 6, 7

[23] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *ECCV*, 2016. 5

[24] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher Twigg, Po-Chen Wu, Junsong Yuan, Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In *ECCV*, 2022. 2, 3, 4

[25] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: non-autoregressive structured modeling for 3d hand pose estimation. In *ECCV*, 2020. 1

[26] Lin Huang, Boshen Zhang, Zhilin Guo, Yang Xiao, Zhiguo Cao, and Junsong Yuan. Survey on depth and rgb image-based 3d hand shape and pose estimation. *VRIH*, 2021. 1

[27] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, 2020. 2, 3

[28] Peter J Huber. Robust estimation of a location parameter. *Breakthroughs in statistics*, 1992. 4

[29] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *ECCV*, 2018. 1, 2, 6, 7

[30] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *3DV*, 2020. 3

[31] Nilesh Kulkarni, Justin Johnson, and David F Fouhey. What's behind the couch? directed ray distance functions (drdf) for 3d scene reconstruction. In *arXiv e-prints*, 2021. 3

[32] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 1, 3

[33] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 2009. 2, 7

[34] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *CVPR*, 2019. 1

[35] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 3, 7

[36] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 3

[37] Feng Liu, Luan Tran, and Xiaoming Liu. Fully understanding generic objects: Modeling, segmentation, and reconstruction. In *CVPR*, 2021. 3

[38] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 2021. 7

[39] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 1987. 7

[40] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 3

[41] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019. 2

[42] 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. In *ECCV*, 2020. 1, 2, 6, 7

[43] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1

[44] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *ICCV*, 2019. 3

[45] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *CVPR*, 2022. 1, 2, 3, 5, 7, 8

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3, 4

[47] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017. 1

[48] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 2022. 2

[49] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 1, 2, 3, 4, 5

[50] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 2, 3

[51] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. In *NeurIPS*, 2020. 3

[52] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 1, 2

[53] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *CVPR*, 2022. 1, 2, 3

[54] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *CVPR*, 2018. 1, 3, 8

[55] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1

[56] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *ICCV*, 2019. 1

[57] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 3

[58] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *CVPR*, 2022. 7

[59] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *ICCV*, 2019. 1, 2

[60] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *ICCV*, 2019. 1, 2

[61] Xiaozheng Zheng, Pengfei Ren, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Joint-aware regression: Rethinking regression-based method for 3d hand pose estimation. In *BMVC*, 2021. 7

[62] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *TPAMI*, 2021. 2, 3

[63] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017. 1, 2

[64] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019. 5, 6