Hand Pose Estimation via Latent 2.5D Heatmap Regression

Umar Iqbal[1,2](✉), Pavlo Molchanov[1], Thomas Breuel[1], Juergen Gall[2], and Jan Kautz[1]

[1] NVIDIA, Santa Clara, USA
uiqbal@iai.uni-bonn.de
[2] University of Bonn, Bonn, Germany

Abstract. Estimating the 3D pose of a hand is an essential part of human-computer interaction. Estimating 3D pose using depth or multi-view sensors has become easier with recent advances in computer vision, however, regressing pose from a single RGB image is much less straightforward. The main difficulty arises from the fact that 3D pose requires some form of depth estimates, which are ambiguous given only an RGB image. In this paper we propose a new method for 3D hand pose estimation from a monocular image through a novel 2.5D pose representation. Our new representation estimates pose up to a scaling factor, which can be estimated additionally if a prior of the hand size is given. We implicitly learn depth maps and heatmap distributions with a novel CNN architecture. Our system achieves state-of-the-art accuracy for 2D and 3D hand pose estimation on several challenging datasets in presence of severe occlusions.

Keywords: Hand pose · 2D to 3D · 3D reconstruction
2.5D heatmaps

1 Introduction

Hand pose estimation from touch-less sensors enables advanced human machine interaction to increase comfort and safety. Estimating the pose accurately is a difficult task due to the large amounts of appearance variation, self occlusions and complexity of the articulated hand poses. 3D hand pose estimation escalates the difficulties even further since the depth of the hand keypoints also has to be estimated. To alleviate these challenges, many proposed solutions simplify the problem by using calibrated multi-view camera systems [1–9], depth sensors [10–24], or color markers/gloves [25]. These approaches are, however, not very desirable due to their inapplicability in unconstrained environments. Therefore, in this work, we address the problem of 3D hand pose estimation from RGB images taken from the wild.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01252-6_8) contains supplementary material, which is available to authorized users.

© Springer Nature Switzerland AG 2018
V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11215, pp. 125–143, 2018.
https://doi.org/10.1007/978-3-030-01252-6_8

Given an RGB image of the hand, our goal is to estimate the 3D coordinates of hand keypoints relative to the camera. Estimating the 3D pose from a monocular hand image is an ill-posed problem due to scale and depth ambiguities. Attempting to do so will either not work at all, or results in over-fitting to a very specific environment and subjects. We address these challenges by decomposing the problem into two subproblems both of which can be solved with less ambiguities. To this end, we propose a novel 2.5D pose representation and then provide a solution to reconstruct the 3D pose from 2.5D. The proposed 2.5D representation is scale and translation invariant and can be easier estimated from RGB images. It consists of 2D coordinates of the hand keypoints in the input image and scale normalized depth for each keypoint relative to the root (palm). We perform scale normalization of the depth values such that one of the bones always has a fixed length in 3D space. Such a constrained normalization allows us to directly reconstruct the scale normalized absolute 3D pose with less ambiguity compared to full depth recovery from the image crop. Our solution is still ill-posed because of relative normalized depth estimation, but it is better defined compared to relative or absolute depth estimation.

As a second contribution, we propose a novel CNN architecture to estimate the 2.5D pose from images. In the literature, there exists two main learning paradigms, namely heatmap regression [26,27] and holistic pose regression [28,29]. Heatmap regression is now a standard approach for 2D pose estimation since it allows to accurately localize the keypoints in the image via per-pixel predictions. Creating volumetric heatmaps for 3D pose estimation [30], however, results in very high computational overhead. Therefore, holistic regression is a standard approach for 3D pose estimation, but it suffers from accurate 2D keypoint localization. Since the 2.5D pose representation requires the prediction of both the 2D pose and depth values, we propose a new heatmap representation that we refer to as 2.5D heatmaps. It consists of 2D heatmaps for 2D keypoint localization and a depth map for each keypoint for depth prediction. We design the proposed CNN architecture such that the 2.5D heatmaps do not have to be designed by hand, but are learned in a latent way. We do this by a softargmax operation which converts the 2.5D heatmaps to 2.5D coordinates in a differentiable manner. The obtained 2.5D heatmaps are compact, invariant to scale and translation, and have the potential to localize keypoints with sub-pixel accuracy.

We evaluate our approach on five challenging datasets with severe occlusions, hand object interactions and in-the-wild images. We demonstrate its effectiveness for both 2D and 3D hand pose estimation. The proposed approach outperforms state-of-the-art approaches by a large margin.

2 Related Work

Very few works in the literature have addressed the problem of 3D hand pose estimation from a single 2D image. The problem, however, shares several properties with human body pose estimation and many approaches proposed for the human body can be easily adapted for hand pose estimation. Hence, in the following, we discuss the related works for 3D articulated pose estimation in general.

Model-Based Methods. These methods represent the articulated 3D pose using a deformable 3D shape model. This is often formulated as an optimization problem, whose objective is to find the model's deformation parameters such that its projection is in correspondence with the observed image data [31–37].

Search-Based Methods. These methods follow a non-parametric approach and formulate 3D pose estimation as a nearest neighbor search problem in large databases of 3D poses, where the matching is performed based on some low [38, 39] or high [40,41] level features extracted from the image.

From 2D Pose to 3D. Earlier methods in this direction learn probabilistic 3D pose models from MoCap data and recover 3D pose by lifting the 2D keypoints [42–45]. More recent approaches, on the other hand, use deep neural networks to learn a mapping from 2D pose to 3D [46–48]. Instead of 2D keypoint locations, [48,49] use 2D heatmaps [26,27] as input and learn convolutional neural networks for 3D pose regression.

The aforementioned methods have the advantage that they do not necessarily require images with ground-truth 3D pose annotations for training, but their major drawback is that they cannot handle re-projection ambiguities *i.e.*, a joint with positive or negative depth will have the same 2D projections. Moreover, they are sensitive to errors in 2D image measurements and the required optimization methods are often prone to local minima due to incorrect initializations.

3D Pose from Images. These approaches aim to learn a direct mapping from RGB images to 3D pose [50–53]. While these methods can better handle 2D projection ambiguities, their main downside is that they are prone to over-fitting to the views only present in training data. Thus, they require a large amount of training data with accurate 3D pose annotations. Collecting large amounts of training data in unconstrained environments is, however, infeasible. To this end, [52] proposes to use Generative Adversarial Networks [54] to convert synthetically generated hand images to look realistic. Other approaches formulate the problem in a multi-task setup to jointly estimate both 2D keypoint locations and 3D pose [29,30,55–58]. Our method also follows this paradigm. The closest work to ours are the approaches of [29,30,56,58] in that they also perform 2.5D coordinate regression. While the approach in [29] performs holistic pose regression with a fully connected output layer, [56] follows a hybrid approach and combines heatmap regression with holistic regression. Holistic regressions is shown to perform well for human body but fails in cases where very precise localization is required, *e.g.*, fingertips in case of hands. In order to deal with this, the approach in [30] performs dense volumetric regression. This, however, substantially increases the model size, which in turn forces to work at a lower spatial resolution.

Our approach, on the other hand, retains the input spatial resolution and allows one to localize hand keypoints with sub-pixel accuracy. It enjoys the differentiability and compactness of holistic regression-based methods, translation invariance of volumetric representations, while also providing high spatial output resolution. Moreover, in contrast to existing methods such as VNect [58], it does

not require hand-designed target heatmaps, which can arguably be sub-optimal for a particular problem, but rather implicitly learns a latent 2.5D heatmap representation and converts them to 2.5D coordinates in a differentiable way.

Finally, note that given the 2.5D coordinates, the 3D pose has to be recovered. The existing approaches either make very strong assumptions such as the ground-truth location of the root [29] and the global scale of the hand in 3D is known [56], or resort to an approximate solution [30]. The approach [57] tries to directly regress the absolute depth from the cropped and scaled image regions which is a very ambiguous task. The approach VNect [58] regresses both 2D and 3D coordinates simultaneously which is ill-posed without explicit modeling of the camera parameters matrix and requires training a specific network for all unique camera matrices. In contrast, our approach does not make any assumptions. Instead, we propose a scale and translation invariant 2.5D pose representation, which can be easily obtained using CNNs, and then provide an exact solution to obtain the absolute 3D pose up to a scaling factor and only approximate the global scale of the hand.

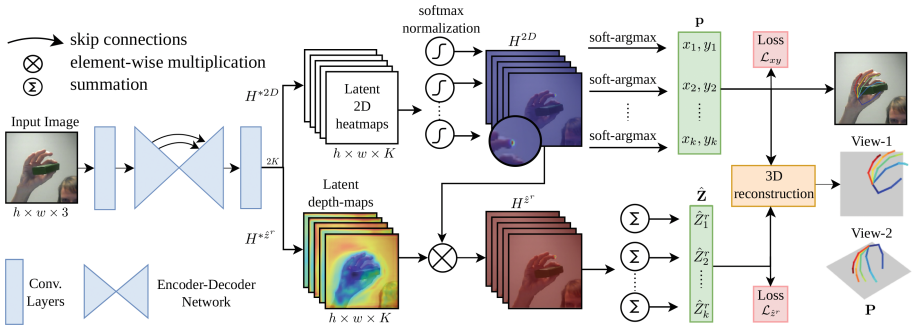


Fig. 1. Overview of the proposed approach. Given an image of a hand, the proposed CNN architecture produces latent 2.5D heatmaps containing the latent 2D heatmaps H^{*2D} and latent depth maps $H^{*\hat{z}}$. The latent 2D heatmaps are converted to probability maps H^{2D} using softmax normalization. The depth maps $H^{\hat{z}}$ are obtained by multiplying the latent depth maps $H^{*\hat{z}}$ with the 2D heatmaps. The 2D pose $\mathbf{p}$ is obtained by applying spatial soft-argmax on the 2D heatmaps, whereas the normalized depth values $\hat{\mathbf{Z}}^r$ are obtained by the summation of depth maps. The final 3D pose is then estimated by the proposed approach for reconstructing 3D pose from 2.5D.

3 Hand Pose Estimation

An overview of the proposed approach can be seen in Fig. 1. Given an RGB image $\mathbf{I}$ of a hand, our goal is to estimate the 2D and 3D positions of all the $K = 21$ keypoints of the hand. We define the 2D hand pose as $\mathbf{p} = \{p_k\}_{k \in K}$ and 3D pose as $\mathbf{P} = \{P_k\}_{k \in K}$, where $p_k = (x_k, y_k) \in \mathbb{R}^2$ represents the 2D pixel coordinates of the keypoint k in image $\mathbf{I}$ and $P_k = (X_k, Y_k, Z_k) \in \mathbb{R}^3$ denotes the location of the keypoint in the 3D camera coordinate frame measured in millimeters.

The Z-axis corresponds to the optical axis. Given the intrinsic camera parameters $\mathcal{K}$, the relationship between the 3D location P_k and corresponding 2D projection p_k can be written as follows under a perspective projection:

$$Z_k \begin{pmatrix} x_k \\ y_k \\ 1 \end{pmatrix} = \mathcal{K} \begin{pmatrix} X_k \\ Y_k \\ Z_k \\ 1 \end{pmatrix} = \mathcal{K} \begin{pmatrix} X_k \\ Y_k \\ Z_{root} + Z_k^r \\ 1 \end{pmatrix} \quad k \in 1, \dots K \tag{1}$$

where $k \in 1, \dots K$, Z_{root} is the depth of the root keypoint, and $Z_k^r = Z_k - Z_{root}$ corresponds to the depth of the kth keypoint relative to the root. In this work we use the palm of the hand as the root keypoint.

3.1 2.5D Pose Representation

Given an image $\mathbf{I}$, we need to have a function $\mathcal{F}$, such that $\mathcal{F} : \mathbf{I} \rightarrow \mathbf{P}$, and the estimated 3D hand pose $\mathbf{P}$ can be projected to 2D with the camera parameters $\mathcal{K}$. However, predicting the absolute 3D hand pose in camera coordinates is infeasible due to irreversible geometry and scale ambiguities. We, therefore, choose a 2.5D pose representation, which is much easier to be recovered from a 2D image, and provide a solution to recover the 3D pose from the 2.5D representation. We define the 2.5D pose as $\mathbf{P}_k^{2.5D} = \{P_k^{2.5D}\}_{k \in K}$, where $P_k^{2.5D} = (x_k, y_k, Z_k^r)$. The coordinates x_k and y_k are the image pixel coordinates of the kth keypoint and Z_k^r is its metric depth relative to the root keypoint. Moreover, in order to remove the scale ambiguities, we scale-normalize the 3D pose as follows:

$$\hat{\mathbf{P}} = \frac{C}{s} \cdot \mathbf{P}, \tag{2}$$

where $s = \|P_n - P_{parent(n)}\|_2$ is computed for each 3D pose independently. This results in a normalized 3D pose $\hat{\mathbf{P}}$ with a constant distance C between a specific pair of keypoints $(n, parent(n))$. Subsequently, our normalized 2.5D representation for keypoint k becomes $\hat{P}_k^{2.5D} = (x_k, y_k, \hat{Z}_k^r)$. Note that the 2D pose does not change due to the normalization, since the projection of the 3D pose remains the same. Such a normalized 2.5D representation has several advantages: it allows to effectively exploit image information; it enables dense pixel-wise prediction (Sect. 4); it allows us to perform multi-task learning so that multiple sources of training data can be used; and finally it allows us to devise an approach to exactly recover the absolute 3D pose up to a scaling factor. We describe the proposed solution to obtain the function $\mathcal{F}$ in Sect. 4, while the 3D pose reconstruction from 2.5D pose is explained in the next section.

3.2 3D Pose Reconstruction from 2.5D

Given a 2.5D pose $\hat{\mathbf{P}}^{2.5D} = \mathcal{F}(\mathbf{I})$, we need to find the depth $\hat{Z}_{root}$ of the root keypoint to reconstruct the scale normalized 3D pose $\hat{\mathbf{P}}$ using Eq. (1). While there exist many 3D poses that can have the same 2D projection, given the

2.5D pose and intrinsic camera parameters, there exists a unique 3D pose that satisfies

$$(\hat{X}_n - \hat{X}_m)^2 + (\hat{Y}_n - \hat{Y}_m)^2 + (\hat{Z}_n - \hat{Z}_m)^2 = C^2, \tag{3}$$

where $(n,\ m = parent(n))$ is the pair of keypoints used for normalization in Eq. (2). The equation above can be rewritten in terms of the 2D projections (x_n, y_n) and (x_m, y_m) as follows:

$$(x_n\hat{Z}_n - x_m\hat{Z}_m)^2 + (y_n\hat{Z}_n - y_m\hat{Z}_m)^2 + (\hat{Z}_n - \hat{Z}_m)^2 = C^2. \tag{4}$$

Subsequently, replacing $\hat{Z}_n$ and $\hat{Z}_m$ with $(\hat{Z}_{root} + \hat{Z}^r_n)$ and $(\hat{Z}_{root} + \hat{Z}^r_m)$, respectively, yields:

$$\begin{aligned}(x_n(\hat{Z}_{root} + \hat{Z}^r_n) - x_m(\hat{Z}_{root} + \hat{Z}^r_m))^2 + (y_n(\hat{Z}_{root} + \hat{Z}^r_n) - y_m(\hat{Z}_{root} + \hat{Z}^r_m))^2 \\ +((\hat{Z}_{root} + \hat{Z}^r_n) - (\hat{Z}_{root} + \hat{Z}^r_m))^2 = C^2.\end{aligned} \tag{5}$$

Given the 2.5D coordinates of both keypoints n and m, Z_{root} is the only unknown in the equation above. Simplifying the equation further leads to a quadratic equation with the following coefficients

$$\begin{aligned}a &= (x_n - x_m)^2 + (y_n - y_m)^2 \\ b &= \hat{Z}^r_n(x_n^2 + y_n^2 - x_nx_m - y_ny_m) + \hat{Z}^r_m(x_m^2 + y_m^2 - x_nx_m - y_ny_m) \\ c &= (x_n\hat{Z}^r_n - x_m\hat{Z}^r_m)^2 + (y_n\hat{Z}^r_n - y_m\hat{Z}^r_m)^2 + (\hat{Z}^r_n - \hat{Z}^r_m)^2 - C^2.\end{aligned} \tag{6}$$

This results in two values for the unknown variable Z_{root}, one in front of the camera and one behind the camera. We choose the solution in front of the camera *i.e.*, $\hat{Z}_{root} = 0.5(-b + \sqrt{b^2 - 4ac})/a$. Given the value of Z_{root}, $\hat{\mathbf{P}}^{2.5D}$, and the intrinsic camera parameters $\mathcal{K}$, the scale normalized 3D pose can be reconstructed by back-projecting the 2D pose $\mathbf{p}$ using Eq. (1). In this paper, we use $C = 1$, and use the distance between the first joint (metacarpophalangeal - MCP) of the index finger and the palm (root) to calculate the scaling factor s. We choose these keypoints since they are the most stable in terms of 2D pose estimation.

3.3 Scale Recovery

Up to this point, we have obtained the 2D and scale normalized 3D pose $\hat{\mathbf{P}}$ of the hand. In order to recover the absolute 3D pose $\mathbf{P}$, we need to know the global scale of the hand. In many scenarios this can be known a priori, however, in case it is not available, we estimate the scale $\hat{s}$ by

$$\hat{s} = \underset{s}{\operatorname{argmin}} \sum_{k,l \in \mathcal{E}} (s \cdot \|\hat{P}_k - \hat{P}_l\| - \mu_{kl})^2, \tag{7}$$

where μ_{kl} is the mean length of the bone between keypoints k and l in the training data, and $\mathcal{E}$ defines the kinematic structure of the hand.

4 2.5D Pose Regression

In order to regress the 2.5D pose $\hat{\mathbf{P}}^{2.5D}$ from an RGB image of the hand, we learn the function $\mathcal{F}$ using a CNN. In this section, we first describe an alternative formulation of the CNN (Sect. 4.1) and then describe our proposed solution for regressing latent 2.5D heatmaps in Sect. 4.2. In all formulations, we train the CNNs using a loss function $\mathcal{L}$ which consists of two parts $\mathcal{L}_{xy}$ and $\mathcal{L}_{\hat{Z}^r}$, each responsible for the regression of 2D pose and root-relative depths for the hand keypoints, respectively. Formally, the loss can be written as follows:

$$\mathcal{L}(\hat{\mathbf{P}}^{2.5D}) = \mathcal{L}_{xy}(\mathbf{p}, \mathbf{p}_{gt}) + \alpha \mathcal{L}_{\hat{Z}^r}(\hat{\mathbf{Z}}^r, \hat{\mathbf{Z}}^{r,gt}), \tag{8}$$

where $\hat{\mathbf{Z}}^r = \{\hat{Z}_k^r\}_{r \in K}$ and $\hat{\mathbf{Z}}^{r,gt} = \{\hat{Z}_k^{r,gt}\}_{r \in K}$ and gt refers to ground-truth annotations. This loss function has the advantage that it allows to utilize multiple sources of training, *i.e.*, in-the-wild images with only 2D pose annotations and constrained or synthetic images with accurate 3D pose annotations. While $\mathcal{L}_{xy}$ is valid for all training samples, $\mathcal{L}_{\hat{Z}^r}$ is enforced only when the 3D pose annotations are available, otherwise it is not considered.

4.1 Direct 2.5D Heatmap Regression

Heatmap regression is the de-facto approach for 2D pose estimation [26,27,59,60]. In contrast to holistic regression, heatmaps have the advantage of providing higher output resolution, which helps in accurately localizing the keypoints. However, they are scarcely used for 3D pose estimation since a 3D volumetric heatmap representation [30] results in a high computational and storage cost.

We, thus, propose a novel and compact heatmap representation, which we refer to as 2.5D heatmaps. It consists of 2D heatmaps H^{2D} for keypoint localization and depth maps $H^{\hat{z}^r}$ for depth predictions. While the 2D heatmap H_k^{2D} represents the likelihood of the kth keypoint at each pixel location, the depth map $H_k^{\hat{z}^r}$ provides the scale normalized and root-relative depth prediction for the corresponding pixels. This representation of depth maps is scale and translation invariant and remains consistent across similar hand poses, therefore, it is significantly easier to be learned using CNNs. The CNN provides a $2K$ channel output with K channels for 2D localization heatmaps H^{2D} and K channels for depth maps $H^{\hat{z}^r}$. The target heatmap $H_k^{2D,gt}$ for the kth keypoint is defined as

$$H_k^{2D,gt}(p) = \exp\left(-\frac{\|p - p_k^{gt}\|}{\sigma^2}\right), \quad p \in \Omega \tag{9}$$

where p_k^{gt} is the ground-truth location of the kth keypoint, σ controls the standard deviation of the heatmaps and Ω is the set of all pixel locations in image $\mathbf{I}$. Since the ground-truth depth maps are not available, we define them by

$$H_k^{\hat{z}^r} = \hat{Z}_k^{r,gt} \cdot H_k^{2D,gt} \tag{10}$$

where $\hat{Z}_k^{r,gt}$ is the ground-truth normalized root-relative depth value of the kth keypoint. During inference, the 2D keypoint position is obtained as the pixel with the maximum likelihood

$$p_k = \underset{p}{\operatorname{argmax}}\ H_k^{2D}(p), \tag{11}$$

and the corresponding depth value is obtained as,

$$\hat{Z}_k^r = H_k^{\hat{z}^r}(p_k). \tag{12}$$

4.2 Latent 2.5D Heatmap Regression

The 2.5D heatmap representation as described in the previous section is, arguably, not the most optimal representation. First, the ground-truth heatmaps are hand designed and are not ideal, *i.e.*, σ remains fixed for all keypoints and cannot be learned due to indifferentiability of Eq. (11). Ideally, it should be adapted for each keypoint, *e.g.*, heatmaps should be very peaky for fingertips while relatively wide for the palm. Secondly, the Gaussian distribution is a natural choice for 2D keypoint localization, but it is not very intuitive for depth prediction, *i.e.*, the depth stays roughly the same throughout the palm but is modeled as Gaussians. Therefore, we alleviate these problems by proposing a latent representation of 2.5D heatmaps, *i.e.*, the CNN learns the optimal representation by minimizing a loss function in a differentiable way.

To this end, we consider the $2K$ channel output of the CNN as latent variables H_k^{*2D} and $H_k^{*\hat{z}^r}$ for 2D heatmaps and depth maps, respectively. We then apply spatial softmax normalization to the 2D heatmap H_k^{*2D} of each keypoint k to convert it to a probability map

$$H_k^{2D}(p) = \frac{\exp(\beta_k H_k^{*2D}(p))}{\sum_{p' \in \Omega} \exp(\beta_k H_k^{*2D}(p'))}, \tag{13}$$

where Ω is the set of all pixel locations in the input map H_k^{*2D}, and β_k is the learnable parameter that controls the spread of the output heatmaps H^{2D}. Finally, the 2D keypoint position of the kth keypoint is obtained as the weighted average of the 2D pixel coordinates as,

$$p_k = \sum_{p \in \Omega} H_k^{2D}(p) \cdot p, \tag{14}$$

while the corresponding depth value is obtained as the summation of the Hadamard product of $H_k^{2D}(p)$ and $H_k^{*\hat{z}^r}(p)$ as follows

$$\hat{Z}_k^r = \sum_{p \in \Omega} H_k^{2D}(p) \circ H_k^{*\hat{z}^r}(p). \tag{15}$$

A pictorial representation of this process can be seen in Fig. 1. The operation in Eq. (14) is known as soft-argmax in the literature [61]. Note that the computation

of both the 2D keypoint location and the corresponding depth value is fully differentiable. Hence the network can be trained end-to-end, while generating a latent 2.5D heatmap representation. In contrast to the heatmaps with fixed standard deviation in Sect. 4.1, the spread of the latent heatmaps can be adapted for each keypoint by learning the parameter β_k, while the depth maps are also learned implicitly without any ad-hoc design choices. A comparison between heatmaps obtained by direct heatmap regression and the ones implicitly learned by latent heatmap regression can be seen in Fig. 2.

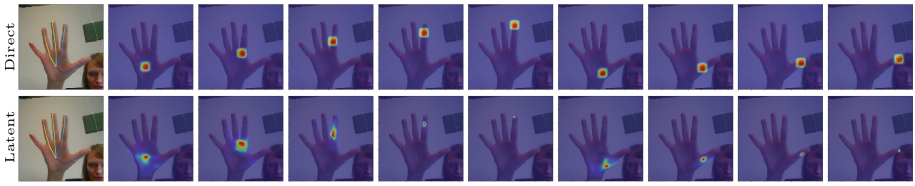


Fig. 2. Comparison between the heatmaps obtained using direct heatmap regression (Sect. 4.1) and the proposed latent heatmap regression approach (Sect. 4.2). We can see how the proposed method automatically learns the spread separately for each keypoint, *i.e.*, very peaky for fingertips while a bit wider for the palm.

5 Experiments

In this section, we evaluate the performance of the proposed approach in detail and also compare it with the state-of-the-art. For this, we use five challenging datasets as follows.

The **Dexter+Object (D+O)** dataset [22] provides 6 test video sequences (3145 frames) recorded using a static camera with a single hand interacting with an object. The dataset provides annotations for the fingertips only.

The **EgoDexter (ED)** dataset [62] consists of 4 test sequences (3190 frames) recorded with a body-mounted camera from egocentric viewpoints and contains cluttered backgrounds, fast camera motion, and complex interactions with various objects. In addition, [62] also provides the so called "SynthHands" dataset of synthetic images of hands from ego-centric views. The images are provided with chroma-keyed background, that we replace with random backgrounds [63] and use them as additional training data for testing on the ED dataset.

The **Stereo Hand Pose (SHP)** dataset [64] provides 3D pose annotations of 21 keypoints for 6 pairs of stereo sequences (18000 frame pairs) recording a person while performing various gestures.

The **Rendered Hand Pose (RHP)** dataset [48] provides 41258 and 2728 synthetic images for training and testing, respectively. The dataset contains 20 different characters performing 39 actions in different settings.

The **MPII+NZSL** [60] dataset provides 2D hand pose annotations for 2800 in-the-wild images split into 2000 and 800 images for training and testing, respectively. In addition, [60] also provides additional training data that contains 14261 synthetic images and 14817 real images. The annotations for real images are generated automatically using multi-view bootstrapping. We refer to these images as MVBS in the rest of this paper.

5.1 Evaluation Metrics

For our evaluation on the D+O, ED, SHP, and RHP datasets, we use average End-Point-Error (EPE) and the Area Under the Curve (AUC) on the Percentage of Correct Keypoints (PCK). We report the performance for both 2D and 3D hand pose where the performance metrics are computed in pixels (px) and millimeters (mm), respectively. We use the publicly available implementation of evaluation metrics from [48]. For the D+O and ED datasets, we follow the evaluation protocol proposed by [52], which requires estimating the absolute 3D pose with global scale. For SHP and RHP, we follow the protocol proposed by [48], where the root keypoints of the ground-truth and estimated poses are aligned before calculating the metrics. For the MPII+NZSL dataset, we follow [60] and report head-normalized PCK (PCKh) in our evaluation.

5.2 Implementation Details

For 2.5D heatmap regression we use an Encoder-Decoder network architecture with skip connections [27,65] and fixed number of channels (256) in each convolutional layer. The input to our model is a 128×128 image, which produces 2.5D heatmaps as output with the same resolution as the input image. Further details about the network architecture and training can be found in the appendix. For all the video datasets, *i.e.*, D+O, ED, SHP we use the YOLO detector [66] to detect the hand in the first frame of the video, and generate the bounding box in the subsequent frames using the estimated pose of the previous frame. We trained the hand detector using the training sets of all aforementioned datasets.

5.3 Ablation Studies

We evaluate the proposed method under different settings to better understand the impact of different design choices. We chose the D+O dataset for all ablation studies, mainly because it does not have any training data. Thus, it allows us to evaluate the generalizability of the proposed method. Finally, since the palm (root) joint is not annotated, it makes it compulsory to estimate the absolute 3D pose in contrast to the commonly used root-relative 3D pose. We use Eq. (7) to estimate the global scale of each 3D pose using the mean bone lengths from the SHP dataset.

The ablative studies are summarized in Table 1. We first examine the impact of different choices of CNN architectures for 2.5D pose regression. For holistic

Table 1. Ablation studies. The arrows specify whether a higher or lower value for each metric is better. The first block compares the proposed approach of latent 2.5D heatmap regression with two baseline approaches. The second block shows the impact of different training data and the last block shows the impact due to differences in the annotations.

Method	2D pose estimation			3D pose estimation		
	AUC ↑	EPE (px)		AUC ↑	EPE (mm)	
		median ↓	mean ↓		median ↓	mean ↓
Comparison with baselines						
Holistic 2.5D reg.	0.41	17.34	22.21	0.54	42.76	47.80
Direct 2.5D heatmap reg.	0.57	10.33	21.63	0.55	36.97	52.33
Latent 2.5D heatmap reg. (Ours)	0.59	9.91	16.67	0.57	39.62	45.54
Impact of training data						
Latent 2.5D heatmap regression trained with						
SHP [64] + RHP [48]	0.59	9.91	16.67	0.57	39.62	45.54
+ MPII + NZSL [60]	0.67	9.07	10.65	0.68	28.11	32.78
+ MVBS [60]	0.68	8.84	10.45	0.68	27.27	32.75
Comparisons with the baselines with additional training data trained with SHP+RHP (3D pose) and MPII+NZSL+MVBS (2d pose) datasets						
Holistic reg.	0.53	12.98	16.17	0.66	31.71	34.86
Direct heatmap reg.	0.65	9.60	12.06	0.68	25.92	35.56
Latent heatmap reg.	0.68	8.84	10.45	0.68	27.27	32.75
Performance after removing labeling discrepancy						
Holistic regression	0.59	10.66	14.10	0.67	30.69	33.80
Direct heatmap reg.	0.72	7.05	9.66	0.68	25.37	34.88
Latent heatmap reg.	0.76	5.95	7.97	0.69	26.56	31.86
Latent heatmap reg. - fast	0.71	6.44	10.67	0.68	28.08	33.35

2.5D pose regression, we use the commonly adopted [29] ResNet-50 [67] model. The details can be found in the appendix. We use the SHP and RHP datasets to train the models. Using a holistic regression approach results in an AUC of 0.41 and 0.54 for 2D and 3D pose, respectively. Directly regressing the 2.5D heatmaps significantly improves the performance of 2D pose estimation (0.41 vs. 0.57), while also raising the 3D pose estimation accuracy from 0.54 AUC to 0.55. Using latent heatmap regression improves the performance even further to 0.59 AUC and 0.57 AUC for 2D and 3D pose estimation, respectively. While the holistic regression approach achieves a competitive accuracy for 3D pose estimation, the accuracy for 2D pose estimation is inferior to the heatmap regression due to its limited spatial output resolution.

We also evaluate the impact of training the network in a multi-task setup. For this, we train the model with additional training data from [60] which provides annotations for 2D keypoints only. First, we only use the 2000 manually anno-

tated real images from the training set of the MPII+NZSL dataset. Using additional 2D pose annotations significantly improves the performance. Adding additional 15,000 annotations of real images from the MVBS dataset [60] improves the performance only slightly. Hence, only 2000 real images are sufficient to generalize the model trained on synthetic data to a realistic scenario. We also evaluate the impact of additional training data on all CNN architectures for 2.5D regression. We can see that the performance improves for all architectures, but importantly, the ordering in terms of performance stays the same.

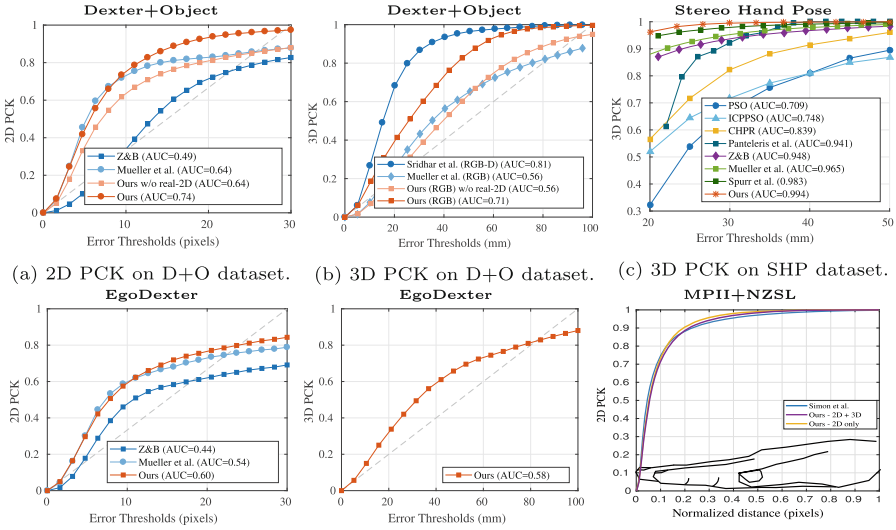
The annotations of the fingertips in the D+O dataset are slightly different than in the other datasets *i.e.*, they are annotated at the middle of the tips whereas other datasets annotate them at the edge of the nails. To remove this discrepancy, we shorten the last bone of the fingertip by 0.9. Fixing the annotation differences results in further improvements, revealing the true performance of the proposed approach.

We also evaluate the runtime of the used models on an NVIDIA TitanX Pascal GPU. While the holistic 2.5D regression model runs at 145 FPS, direct and latent 2.5D heatmap regression networks run at 20 FPS. We trained a smaller 1-stage model with 128 feature maps (base layers) and replaced 7×7 convolutions in the last 3 layers with 1×1, 3×3 and 1×1 convolutions. The simplifications resulted in 150 FPS and parameter reduction by 3.8x while remaining competitive to direct heatmap regression with the full model. This model is marked with label "fast" in Table 1.

Finally, we also evaluate the impact of using multiple stages in the network, where each stage produces latent 2.5D heatmaps as output. The complete 2-stage network is trained from scratch with no weight-sharing. While the first stage only uses the features extracted from the input image using the initial block of convolutional layers, each subsequent stage also utilizes the output of the preceding stage as input. This provides additional contextual information to the subsequent stages and helps in incrementally refining the predictions. Similar to [26,27] we provide local supervision to the network by enforcing the loss at the output of each stage (see appendix for more details). Adding one extra stage to the network increases the 3D pose estimation accuracy from AUC 0.69 to 0.71, but decreases the 2D pose estimation accuracy from AUC 0.76 to 0.74. The decrease in 2D pose estimation accuracy is most likely due to over-fitting to the training datasets. Remember that we do not use any training data from the D+O dataset. In the rest of this paper, we always use networks with two stages unless stated otherwise.

5.4 Comparison to State-of-the-Art

We provide a comparison of the proposed approach with state-of-the-art methods on all aforementioned datasets. Note that different approaches use different training data. We thus replicate the training setup of the corresponding approaches for a fair comparison.



(a) 2D PCK on D+O dataset. (b) 3D PCK on D+O dataset. (c) 3D PCK on SHP dataset.

(d) 2D PCK on ED dataset. (e) 3D PCK on ED dataset. (f) 2D PCKh on MPII+NZSL.

Fig. 3. Comparison with the state-of-the-art on the DO, ED, SHP and MPII+NZSL datasets.

Figure 3a and b compare the proposed approach with other methods on the D+O dataset for 2D and 3D pose estimation, respectively. In particular, we compare with the state-of-the-art approaches by Zimmerman and Brox (Z&B) [48] and Mueller *et al.* [52]. We use the same training data (SHP+RHP) for comparison with [48] (AUC 0.64 vs 0.49), and only use additional 2D pose annotations (MPII+NZSL+MVBS) provided by [60] for comparison with [52](AUC 0.74 vs 0.64). For the 3D pose estimation accuracy (Fig. 3b), the approach [48] is not included since it only estimates scale normalized root-relative 3D pose. Our approach clearly outperforms the state-of-the-art RGB based method by Mueller *et al.* [52] by a large margin. The approach [52] utilizes the video information to perform temporal smoothening and also performs subject specific adaptation under the assumption that the users hold their hand parallel to the camera image plane. In contrast, we only perform frame-wise predictions without temporal filtering or user assumptions. Additionally, we report the results of the depth based approach by Sridhar *et al.* [22], which are obtained from [52]. While the RGB-D sensor based approach [22] still works better, our approach takes a giant leap forward as compared to the existing RGB based approaches.

Figure 3c compares the proposed method with existing approaches on the SHP dataset. We use the same training data (SHP+RHP) as in [48] and outperform all existing methods despite the already saturated accuracy on the dataset and the additional training data and temporal information used in [52].

Figure 3d compares the 2D pose estimation accuracy on the EgoDexter dataset. While we outperform all existing methods for 2D pose estimation, none

Table 2. Comparison with the state-of-the-art on the RHP dataset. *uses noisy ground-truth 2D poses for 3D pose estimation.

Method	2D pose estimation			3D pose estimation		
	AUC ↑	EPE (px)		AUC ↑	EPE (mm)	
		median ↓	mean ↓		median ↓	mean ↓
Z & B [48]	0.72	5.00	9.14	-	18.8*	-
Spurr *et al.* [53]	-	-	-	0.85	19.73	-
Ours	0.89	2.20	3.57	0.91	13.82	15.77
Ours w. GT $\hat{Z}_{root}$ and $\hat{s}$	0.89	2.20	3.57	0.94	11.33	13.41

of the existing approaches report their performance for 3D pose estimation on this dataset. We, however, also report our performance in Fig. 3e.

The results on the RHP dataset are reported in Table 2. Our approach significantly outperforms state-of-the-art approaches [48,53]. Since the dataset provides 3D pose annotations for complete hand skeleton, we also report the performance of the proposed approach when the ground-truth depth of the root joint and the global scale of the hand is known (w. GT $\hat{Z}_{root}$ and $\hat{s}$). We can see that our approach of 3D pose reconstruction and scale recovery is very close to the ground-truth.

Finally, for completeness, in Fig. 3f we compare our approach with [60] which is a state-of-the-art approach for 2D pose estimation. The evaluation is performed on the test set of the MPII+NZSL dataset. We follow [60] and use the provided center location of the hand and the size of the head of the person to obtain the hand bounding box. We define a square bounding box with height and width equal to $0.7 \times$ *head-length*. We report two variants of our method; (1) the model trained for both 2D and 3D pose estimation using the datasets for both tasks, and (2) a model trained for only 2D pose estimation using the same training data as in [60]. In both cases we use the models trained with 2-stages. Our approach performs similar or better than [60], even though we use a smaller backbone network as compared to the 6-stage Convolutional Pose Machines (CPM) network [26] used in [60]. The CPM model with 6-stages has $51M$ parameters, while our 1 and 2-stage models have only $17M$ and $35M$ parameters, respectively. Additionally, our approach also infers the 3D hand pose.

Some qualitative results for 3D hand pose estimation for in-the-wild images can be seen in Fig. 4.

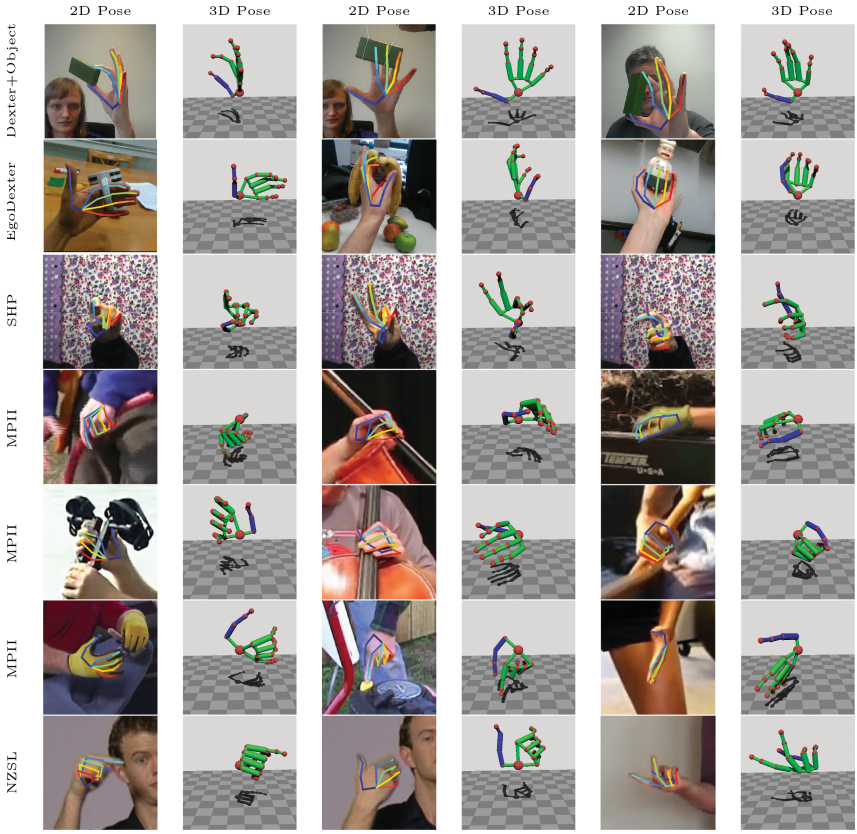


Fig. 4. Qualitative Results. The proposed approach can handle severe occlusions, complex hand articulations, and unconstrained images taken from the wild.

6 Conclusion

We have presented a method for 3D hand pose estimation from a single RGB image. We demonstrated that the absolute 3D hand pose can be reconstructed efficiently from a single image up to a scaling factor. We presented a novel 2.5D pose representation which can be recovered easily from RGB images since it is invariant to absolute depth and scale ambiguities. It can be represented as 2.5D heatmaps, therefore, allows keypoint localization with sub-pixel accuracy. We also proposed a CNN architecture to learn 2.5D heatmaps in a latent way using a differentiable loss function. Finally, we proposed an approach to reconstruct the 3D hand pose from 2.5D pose representation. The proposed approach demonstrated state-of-the-art results on five challenging datasets with severe occlusions, object interactions and images taken from the wild.

Acknowledgements. JG was supported by the ERC Starting Grant ARCA.

References

1. Rehg, J.M., Kanade, T.: Visual tracking of high DOF articulated structures: an application to human hand tracking. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 35–46. Springer, Heidelberg (1994). https://doi.org/10.1007/BFb0028333
2. de Campos, T.E., Murray, D.W.: Regression-based hand pose estimation from multiple cameras. In: CVPR (2006)
3. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Markerless and efficient 26-DOF hand pose recovery. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6494, pp. 744–757. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19318-7_58
4. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3D hand pose reconstruction using specialized mappings. In: ICCV (2001)
5. Ballan, L., Taneja, A., Gall, J., Van Gool, L., Pollefeys, M.: Motion capture of hands in action using discriminative salient points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 640–653. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_46
6. Sridhar, S., Rhodin, H., Seidel, H.P., Oulasvirta, A., Theobalt, C.: Real-time hand tracking using a sum of anisotropic Gaussians model. In: 3DV (2014)
7. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. IJCV **118**, 172–193 (2016)
8. Panteleris, P., Argyros, A.: Back to RGB: 3D tracking of hands and hand-object interactions based on short-baseline stereo. arXiv preprint arXiv:1705.05301 (2017)
9. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. In: SIGGRAPH Asia (2017)
10. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: ICCV (2011)
11. Xu, C., Cheng, L.: Efficient hand pose estimation from a single depth image. In: ICCV (2013)
12. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: CVPR (2014)
13. Taylor, J., et al.: User-specific hand modeling from monocular depth sequences. In: CVPR (2014)
14. Tang, D., Chang, H.J., Tejani, A., Kim, T.K.: Latent regression forest: structured estimation of 3D articulated hand posture. In: CVPR (2014)
15. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ToG **33**, 169 (2014)
16. Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.K., Shotton, J.: Opening the black box: hierarchical sampling optimization for estimating human hand pose. In: ICCV 2015)
17. Makris, A., Kyriazis, N., Argyros, A.A.: Hierarchical particle filtering for 3D hand tracking. In: CVPR (2015)
18. Sridhar, S., Mueller, F., Oulasvirta, A., Theobalt, C.: Fast and robust hand tracking using detection-guided optimization. In: CVPR (2015)
19. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR (2015)

20. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: ICCV (2015)
21. Oberweger, M., Riegler, G., Wohlhart, P., Lepetit, V.: Efficiently creating 3D training data for fine hand pose estimation. In: CVPR (2016)
22. Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D., Oulasvirta, A., Theobalt, C.: Real-time joint tracking of a hand manipulating an object from RGB-D input. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 294–310. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_19
23. Yuan, S., et al.: Depth-based 3d hand pose estimation: from current achievements to future goals. In: IEEE CVPR (2018)
24. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC, vol. 1, p. 3 (2011)
25. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ToG **28**, 63 (2009)
26. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
27. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
28. Toshev, A., Szegedy, C.: Deeppose: human pose estimation via deep neural networks. In: CVPR (2014)
29. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: ICCV (2017)
30. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. In: CVPR (2017)
31. Heap, T., Hogg, D.: Towards 3D hand tracking using a deformable model. In: FG (1996)
32. Wu, Y., Lin, J.Y., Huang, T.S.: Capturing natural hand articulation. In: ICCV (2001)
33. Sigal, L., Balan, A.O., Black, M.J.: HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. IJCV **87**(1), 4–27 (2010)
34. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-based 3D hand pose estimation from monocular video. TPAMI **33**, 1793–1805 (2011)
35. Lu, S., Metaxas, D., Samaras, D., Oliensis, J.: Using multiple cues for hand tracking and model refinement. In: CVPR (2003)
36. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep It SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34
37. Panteleris, P., Oikonomidis, I., Argyros, A.: Using a single RGB frame for real time 3D hand pose estimation in the wild. In: WACV (2018)
38. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. In: CVPR (2003)
39. Romero, J., Kjellström, H., Kragic, D.: Hands in action: real-time 3D reconstruction of hands in interaction with objects. In: ICRA (2010)
40. Chen, C., Ramanan, D.: 3D human pose estimation = 2D pose estimation + matching. In: CVPR (2017)
41. Iqbal, U., Doering, A., Yasin, H., Krüger, B., Weber, A., Gall, J.: A dual-source approach for 3D pose estimation in single images. CVIU (2018, in Press)

42. Ramakrishna, V., Kanade, T., Sheikh, Y.: Reconstructing 3D human pose from 2D image landmarks. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 573–586. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_41
43. Simo-Serra, E., Quattoni, A., Torras, C., Moreno-Noguer, F.: A joint model for 2D and 3D pose estimation from a single image. In: CVPR (2013)
44. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3D human pose reconstruction. In: CVPR (2015)
45. Tome, D., Russell, C., Agapito, L.: Lifting from the deep: convolutional 3D pose estimation from a single image. In: CVPR (2017)
46. Moreno-Noguer, F.: 3D human pose estimation from a single image via distance matrix regression. In: CVPR (2017)
47. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. In: ICCV (2017)
48. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: ICCV (2017)
49. Tekin, B., Marquez-Neila, P., Salzmann, M., Fua, P.: Learning to fuse 2D and 3D image cues for monocular body pose estimation. In: ICCV (2017)
50. Li, S., Chan, A.B.: 3D human pose estimation from monocular images with deep convolutional neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9004, pp. 332–347. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16808-1_23
51. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 186–201. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_17
52. Mueller, F., et al.: GANerated hands for real-time 3D hand tracking from monocular RGB. In: CVPR (2018)
53. Spurr, A., Song, J., Park, S., Hilliges, O.: Cross-modal deep variational hand pose estimation. In: CVPR (2018)
54. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
55. Popa, A., Zanfir, M., Sminchisescu, C.: Deep multitask architecture for integrated 2D and 3D human sensing. In: CVPR (2017)
56. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Weakly-supervised transfer for 3D human pose estimation in the wild. In: ICCV (2017)
57. Nie, B.X., Wei, P., Zhu, S.C.: Monocular 3D human pose estimation by predicting depth on joints. In: ICCV (2017)
58. Mehta, D., et al.: VNect: real-time 3D human pose estimation with a single RGB camera. In: SIGGRAPH (2017)
59. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: CVPR (2015)
60. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: CVPR (2017)
61. Chapelle, O., Wu, M.: Gradient descent optimization of smoothed information retrieval metrics. Inf. Retr. **13**, 216–235 (2010)
62. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: ICCV (2017)
63. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54

64. Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., Yang, Q.: 3D hand pose tracking and estimation using stereo matching. arXiv preprint arXiv:1610.07214 (2016)
65. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
66. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR (2017)
67. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)