Structure-aware 3D Hand Pose Regression from a Single Depth Image

Jameel Malik[1,2], Ahmed Elhayek[1], and Didier Stricker[1]

[1] Department Augmented Vision, DFKI Kaiserslautern, Germany
[2] NUST-SEECS, Pakistan
{jameel.malik,ahmed.elhayek,didier.stricker}@dfki.de

Abstract. Hand pose tracking in 3D is an essential task for many virtual reality (VR) applications such as games and manipulating virtual objects with bare hands. CNN-based learning methods achieve the state-of-the-art accuracy by directly regressing 3D pose from a single depth image. However, the 3D pose estimated by these methods is coarse and kinematically unstable due to independent learning of sparse joint positions. In this paper, we propose a novel structure-aware CNN-based algorithm which learns to automatically segment the hand from a raw depth image and estimate 3D hand pose jointly with new structural constraints. The constraints include fingers lengths, distances of joints along the kinematic chain and fingers inter-distances. Learning these constraints help to maintain a structural relation between the estimated joint keypoints. Also, we convert sparse representation of hand skeleton to dense by performing n-points interpolation between the pairs of parent and child joints. By comprehensive evaluation, we show the effectiveness of our approach and demonstrate competitive performance to the state-of-the-art methods on the public NYU hand pose dataset.

Keywords: Hand pose · Depth image · Convolutional Neural Network (CNN).

1 Introduction

Markerless 3D hand pose estimation is a fundamental challenge for many interesting applications of virtual reality (VR) and augmented reality (AR) such as handling of objects in VR environment, games and interactive control. This task has been extensively studied in the past few years and great progress has been achieved. This is primarily due to the arrival of low cost depth sensors and rapid advancements in deep learning. However, estimating 3D hand pose from a single depth image is still challenging due to self similarities, occlusions, wide range of articulations and varying hand shapes.

Hand pose estimation methods are classified into three main catagories namely learning based methods (discriminative), model-based methods (generative) and combination of the discriminative and generative methods (hybrid). Among these methods, CNN-based discriminative methods have shown the highest accuracy on the public benchmarks. Despite of the fact that these methods achieve higher accuracy, they do not well exploit the structural information of hands during the learning process [35, 34, 11]. Specifically, independent learning of sparse joint positions with no consideration to joint connection structure and hand skeleton constraints leads to coarse predictions.

This is the main reason these methods still generalize poorly on unseen hand shapes [34] and consequently, not directly usable in practical VR applications.

Therefore, our main contribution for this paper is a novel structure-aware CNN-based discriminative approach which incorporates the structural constraints of hand skeleton and enhances the loss function for better learning of 3D hand pose. Our main idea is to jointly learn the 3D joint keypoints and the hand structure parameters. Thereby, facilitating the CNN to maintain a structural relation between the estimated joint keypoints. Our method is simple, efficient and effective. It optimizes a combined loss function of 3D joint positions and simple structural constraints of the hand skeleton. The constraints comprise of fingers lengths, fingers inter-distances and distances of joints in the kinematic chain of the hand skeleton (kinematic distances). These constraints are easy to learn and guide the optimization process to estimate more refined and accurate 3D hand pose. Another contribution which helps to improve the accuracy is to convert the sparse joints keypoints to dense representation. To this end, we perform n-points interpolation between the pairs of parent and child ground truth joint positions along the kinematic chain of hand skeleton. These simple strategies can be easily used to improve the accuracy of any CNN-based discriminative method without additional cost.

Existing hand pose estimation methods assume already segmented hand region from a raw depth image as input to their algorithms. The hand segmentation approaches are mainly based on heuristics or ground truth annotation which make them difficult to use in practical applications. The problem of hand segmentation is not well addressed in the existing works. Hence, our second contribution is a new CNN-based hand segmentation method to extract the hand region from a raw depth frame. For training over images with varying backgrounds and camera noise, we combine several existing hand pose datasets including a new dataset which we capture to include more variation in hand shapes. The combined dataset will be public.

By performing exhaustive evaluation of our algorithm, we show the effectiveness of our hand segmentation algorithm, n-points interpolation strategy and learning the structural constraints jointly with the 3D hand pose. Experiments show that our method performs better than several state-of-the-art hand pose estimation on the NYU public benchmark. The main contributions for this paper are:

1. A novel structure-aware CNN-based algorithm for 3D hand pose estimation including the structural constraints of hand skeleton; see Section 4.2.
2. A novel CNN-based algorithm to effectively segment the hand region from a raw depth image; see Section 4.1.
3. A simple and effective interpolation strategy for improved hand pose estimation; see Section 4.2.

2 Related Work

3D hand pose estimation using a depth sensor has been widely studied in the past few years. For detailed overview, we refer the reader to the survey papers [24, 34]. Here we limit our discussion to the most related works.

Depth-based hand segmentation methods: Tompson et al. [27] introduce a per-pixel classification of the hand region using random decision forest (RDF) based method.

However, the per-pixel manual labeling of large number of training frames is cumbersome. Oberweger et al. [16] apply depth-thresholding thereby, computing the center of mass of hand region. Then, crop the hand using the center of mass. Recently, [15] propose a CNN-based refinement network to further refine the segmented hand depth image by [16] to achieve better localization. In contrast, we convert the raw depth image to RGB by applying simple JET colormap and use a CNN to predict the 2D position of the hand palm center. Then, using the predicted palm center, depth value can easily be obtained from input depth frame. The proposed approach is simple and effective.

Discriminative methods: RDF-based discriminative works [20, 30, 32, 10, 23] are lagging behind recent CNN-based methods such as [12, 1, 31, 6, 5, 19] in accuracy of the estimated hand pose. Some works have employed either RGB or RGB-D data to estimate 3D joint positions [36, 18, 21, 13]. In [5], Ge et al. effectively regress 3D pose using a single 3D-CNN. Recently, [12] propose a voxel-to-voxel pose predictor which takes voxelised input depth image and outputs 3D joint heatmaps. [6, 31] introduce a region ensemble (REN) strategy which concatenates features from multiple networks to regress the 3D pose. Chen et al. [1] extend [31] by an iterative pose-guided REN strategy. All of the above methods optimize only for the 3D pose without incorporating any structural relations between the joint positions. In contrast, we extend the loss function defined on the joint positions only by including several hand structural constraints. Thereby, improving the accuracy of the estimated pose.

Hybrid methods: [27] predict 2D heatmaps using a single CNN. After that they use inverse kinematics to recover the 3D pose. Ge et al. [4] use a 3D-CNN for 2D heatmaps estimation and then recover 3D joint positions. Oberweger et al. [17] train a complex feedback loop to regress 3D joint positions. Wan et al. [28] learn a shared latent space, between an encoder and a decoder, to reconstruct the depth image using generative adversarial network(GAN) and refine the 3D pose. The above mentioned works optimize only for the joints positions and do not explicitly account for the hand geometric constraints. Dibra et al. [3] propose a complex end-to-end framework to indirectly recover the 3D pose from reconstructed depth image. Zhou et al. [35] implement a forward kinematics layer inside the CNN and train an end-to-end pipeline. Malik et al. [11] extend this work to generalize over varying hand shapes. However, these methods suffer from low accuracy because regressing joint angles (for rotation matrices) is cumbersome.

3 Method Overview

The goal of our pipeline is to estimate more stable and accurate 3D joint positions J, given a raw depth input D_o. To this end, we simultaneously optimize for J, fingers lengths FL, fingers inter-distance FD and kinematics distances KD to facilitate the learning of 3D joint positions in a structured manner. Our pipeline is shown in Figure 1. D_o is resized and then colorized (using the JET colormap) by a function g. The output RGB image D_i is of size 227 x 227 x 3. D_i is passed as input to the PalmCNN to directly regress hand palm center (u,v) in image coordinates. Then, a cropping function f is applied to segment the 3D hand region D_s from the raw depth frame D_o. The colorization step is simple and helps to improve the accuracy; see Section 5.2. Finally, the PoseCNN takes D_s as input and estimates 3D joint positions J, fingers lengths FL,

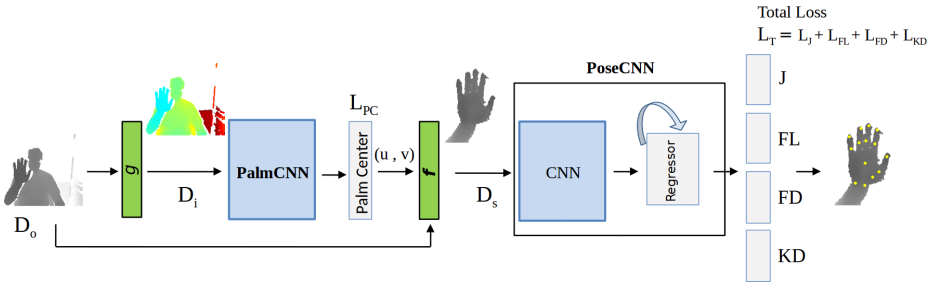

Fig. 1: Our pipeline for hand segmentation and pose estimation. The raw depth frame D_o is given as input to a function g which resizes D_o to 227 x 227 x 3 dimension and colorizes it using the JET colormap. The output of g (D_i) is fed to the PalmCNN to regress 2D hand palm center (u,v). L_{PC} is the loss for the PalmCNN. The function f crops the hand region D_s given (u,v). D_s is fed to PoseCNN which outputs 3D joint positions J, fingers lengths FL, fingers inter-distances FD, and kinematic distances KD.

fingers inter-distance FD and kinematics distances KD. The PoseCNN comprises of a CNN and a regressor; see Section 4.2 for details. The PalmCNN and the PoseCNN are trained separately.

4 Hand Segmentation and Pose Estimation

In this section, we explain the individual components of the pipeline shown in Figure 1. The function g, the PalmCNN and the crop function f are described in Section 4.1. In Section 4.2, we explain the main component of our pipeline i.e. the PoseCNN.

4.1 CNN-based Hand Segmentation

The function g simply resizes and colorizes D_o to be fed as input to the PalmCNN. The output D_i of g is an RGB image of size 227 x 227 x 3. The task of the PalmCNN is to estimate the pixel coordinates of the center of the hand region i.e. palm center (u,v). The CNN architecture of the PalmCNN is similar to the AlexNet [9] except that the final fully connected layer regresses the palm center. The softmax loss layer is replaced by euclidean loss layer. The euclidean 2D palm center loss is given as:

$$L_{PC} = \frac{1}{2} \left\| PC - PC_{GT} \right\|^2 \tag{1}$$

Where L_{PC} is the palm center loss and PC_{GT} is the ground truth palm center. To train the PalmCNN, we combine four of the publicly available hand pose datasets (i.e NYU [27], ICVL [26], MSRA-2015 [23] and Dexter-1 [22]) with a new dataset which we captured using creative senz3D camera [2]. This additional small scale dataset is captured because the public datasets lack in hand shape variation [11]. To obtain the ground truth palm center, we employ the generative method proposed by [25]. We captured depth images from five different subjects. Our dataset contains 8000 original depth images. Notably, the variation in hand position should cover the whole image space. Therefore,

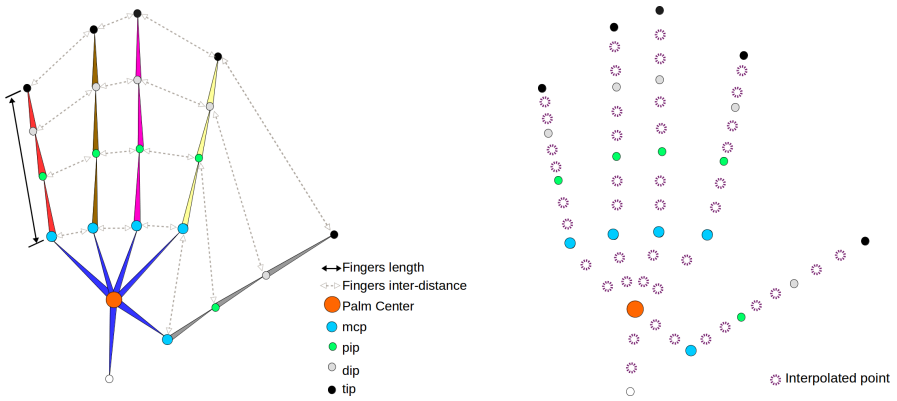

Fig. 2: The left figure shows the graphical representation of two of the structural constraints i.e. Fingers lengths and Fingers inter-distances. The hand skeleton on the right shows the interpolated points ($n = 2$) between the sparse ground truth joint positions.

we create around 10 augmented copies of every depth frame in the combined dataset by translating it around the whole image using the ground truth hand palm center position. The total number of training and testing frames are $4.55M$ and $200K$ respectively. We fine-tune the AlexNet (pre-trained on ImageNet dataset) with the combined dataset. The crop function f takes the estimated (u,v) and D_o as inputs and segments the 3D hand region; see Section 5.1 for details about f. The resultant image D_s is of size 224 x 224.

4.2 Structure-aware 3D Hand Pose Estimation

In our pipeline, the PoseCNN aims to jointly estimate the hand joint keypoints J and additional constraints (i.e. fingers lengths *FL*, fingers inter-distance *FD*, kinematic distances *FD*). During training, these constraints help to maintain a structural relation between the joints positions. The ground truth for the constraints can easily be obtained from the ground truth joint positions. The euclidean 3D joint positions loss L_J is given as:

$$L_J = \frac{1}{2} \left\| J - J_{GT} \right\|^2 \tag{2}$$

Where $J_{GT} \in \mathbb{R}^{P\text{x}3}$ is a vector of 3D ground truth joint positions. P is the number of joint keypoints. The constraints are explained as follows:
Fingers lengths: We first calculate J-1 hand bone-lengths from the ground truth joint positions using the standard 3D euclidean distance formula. To obtain a finger's length fl, we add the bone-lengths from the base joint (*mcp*) to the finger-tip joint (*tip*) as shown in Figure 2. The equation for fl can be written as:

$$fl = bl_{mcp-pip} + bl_{pip-dip} + bl_{dip-tip} \tag{3}$$

Where bl_{x-y} is the bone-length from a parent joint x to a child joint y. Therefore, a set FL_{GT} is represented as:

$$FL_{GT} = \{fl_{pinky}, fl_{ring}, fl_{middle}, fl_{index}, fl_{thumb}\} \tag{4}$$

The euclidean fingers lengths loss L_{FL} is:

$$L_{FL} = \frac{1}{2} \left\| \boldsymbol{FL} - \boldsymbol{FL}_{GT} \right\|^2 \tag{5}$$

Where *FL* is the vector of estimated fingers lengths.

Fingers inter-distances: The distances between the *mcp* joints of consecutive fingers for a particular hand mostly remain fixed. However, the distances between *pip*, *dip* and *tip* joints between fingers can vary depending on the pose of the hand. The inter-distances between neighboring fingers can easily be obtained by calculating 3D euclidean distances between respective joints of the fingers; see Figure 2. For example, the inter-distances between *index* and *middle* fingers are evaluated as:

$$\begin{aligned} fd(index, middle) = \{ & d(mcp_{index}, mcp_{middle}), d(pip_{index}, pip_{middle}), \\ & d(dip_{index}, dip_{middle}), d(tip_{index}, tip_{middle}) \} \end{aligned} \tag{6}$$

Where $fd(.)$ is a set of inter-distances between the joints of two adjacent fingers and $d(.)$ represents 3D euclidean distance between two joints. Likewise, inter-distances for remaining finger pairs i.e. $(middle, ring)$, $(ring, pinky)$ and $(thumb, index)$ can be obtained using Equation 6. Hence, a set $\boldsymbol{FD}_{GT}$ can be expressed as:

$$\begin{aligned} FD_{GT} = \{ & fd(index, middle), fd(middle, ring), \\ & fd(ring, pinky), fd(thumb, index) \} \end{aligned} \tag{7}$$

The fingers inter-distances loss L_{FD} can be written as:

$$L_{FD} = \frac{1}{2} \left\| \boldsymbol{FD} - \boldsymbol{FD}_{GT} \right\|^2 \tag{8}$$

Where *FD* is the vector of estimated fingers inter-distances.

Kinematic distances: Hand skeleton bears an inherent kinematic structure which should not be ignored in the pose estimation task. Otherwise, the resultant pose could be kinematically unstable [35, 11]. In this work, we add a much needed loss function which incorporates kinematic distances of all the joints in the hand skeleton. Given the set of parents joints S_{p_j} of a joint p_j in J_{GT}, the kinematic distance kd_j from the root joint to p_j can be calculated as:

$$kd_j = \sum_{i=0}^{M-1} d(J_{GT_i}, J_{GT_{i+1}}) \tag{9}$$

Where $i \in S_{p_j}$ and M is the size of the set S_{p_j}. Using Equation 9, the kinematic distances of each joint in J_{GT} can be obtained. Hence, the loss L_{KD} can be written as:

$$L_{KD} = \frac{1}{2} \left\| \boldsymbol{KD} - \boldsymbol{KD}_{GT} \right\|^2 \tag{10}$$

Where KD and KD_{GT} are the vectors of estimated and ground truth kinematic distances.

Total loss: Including the additional constraints (mentioned above) help to improve the accuracy of hand pose estimation task and maintain the structure of the hand skeleton; see Section 5.2. The final loss equation for the PoseCNN can be written as:

$$L_T = L_J + L_{FL} + L_{FD} + L_{KD}. \tag{11}$$

Interpolation: In order to get a dense representation of hand skeleton, we linearly interpolate n joints between each pair of parent and child joints in the kinematic hierarchy of the hand skeleton; see Figure 2. We try different number of interpolated points n and study their effects on the accuracy of the estimated pose; see Section 5.2. As an example, the formulas for interpolating two 3D points $P1$ and $P2$ between two 3D points P_a and P_b are:

$$P1 = 0.7 * P_a + 0.3 * P_b \quad , \quad P2 = 0.3 * P_a + 0.7 * P_b \tag{12}$$

Architecture and iterative regression: The architecture of CNN in the PoseCNN is similar to the ResNet-50 [7] except that final fully connected (FC) layer which outputs the features $\varphi \in \mathbb{R}^{1024}$. The features φ are concatenated with an initial estimate of $E = \{J, FD, FL \text{ and } KD\}$ i.e. $\phi = \{\varphi, E\}$. Initial estimate of E is obtained using the mean values of $\{J, FD, FL \text{ and } KD\}$ from the NYU ground truth annotations. This estimate is kept fixed during the training and the testing. ϕ is fed to a regressor which comprises of two FC layers with 1024 neurons each. Both the FC layers use dropout layers with ratio of 0.3. The last FC layer contains M neurons. Where $M = 2P(n+1)+10n+21$. The regressor aims to refine E in an iterative feedback manner i.e. $E_{t+1} = E_t + \delta E_t$. In our implementation, we use at least three iterations. Directly regressing E is challenging therefore, we observe that inclusion of the regressor is beneficial.

5 Experiments

In this section, we provide the implementation details, evaluation of our framework and comparison with the state-of-the-art hand pose estimation methods. The evaluation metrics are 3D joint location error and number of frames within certain thresholds. All the error metrics are reported in mm.

5.1 Implementation Details

We use Caffe [8], an open-source deep learning framework, to train the PalmCNN and the PoseCNN in our pipeline (see Figure 1). The networks run on a desktop using Nvidia Geforce GTX 1080 Ti GPU. The PalmCNN is trained on the combined dataset; see Section 4.1. The learning rate is set to 0.0001 with a batch size of 256 and 0.9 SGD momentum. One forward pass in the PalmCNN takes $4.5ms$. We train the PoseCNN on the NYU hand pose dataset [27]. The NYU dataset has $72,757$ images for training and 8252 frames for testing. In order to segment the hand region from the raw depth input D_o, we use the estimated palm center from the PalmCNN. Given (u,v) and D_o, the hand region is cropped in 3D using a bounding box of size 300 and the camera focal length. The pre-processed image is of size 224 x 224 and the depth values are normalized to $[-1, 1]$. The 3D joints annotations J_{GT} in camera coordinates are also normalized to range $[-1, 1]$. We obtain FL_{GT}, FD_{GT} and KD_{GT} from the normalized J_{GT}. For training the PoseCNN, we use 0.001 learning rate with 0.9 SGD momentum and a batch size of 128. The forward pass for the PoseCNN takes $35ms$.

Method Implementations	3D Joint Location Error
	J
PoseCNN(J)	15.2mm
PoseCNN($J \cup FL$)	14.7mm
PoseCNN($J \cup FD$)	13.6mm
PoseCNN($J \cup KD$)	13.9mm
PoseCNN($J \cup FL \cup FD \cup KD$)	**12.9mm**

Table 1: We evaluate five different implementations of our PoseCNN on the NYU hand pose dataset. The PoseCNN(J) is the *baseline* which is trained for estimating joint positions only. The PoseCNN($J \cup FL \cup FD \cup KD$) performs the best and shows an error improvement of 15.13% on the estimated J over the *baseline*.

n-points Interpolation	3D Joint Location Error
	J
PoseCNN(1-point Interp.)	12.80mm
PoseCNN(2-point Interp.)	12.63mm
PoseCNN(3-point Interp.)	12.38mm
PoseCNN(4-point Interp.)	12.17mm
PoseCNN(5-point Interp.)	**11.9mm**

Table 2: We observe the effects of n-points interpolation between the pairs of parent and child joints in the kinematic hierarchy of the hand skeleton. The value of n varies from 1 to 5. 5-point interpolation shows 5.5% improvement in accuracy. For $n > 5$, we do not observe notable error improvement.

5.2 Method Evaluation

In this subsection, we comprehensively evaluate the PoseCNN and the PalmCNN. We first observe the effects of the proposed structural constraints on the accuracy of the estimated joint positions J. Second is to study the effects of interpolating n-points between the sparse joint positions.

Structural constraints: To this end, we train the following implementations of the PoseCNN on the NYU hand pose dataset which learns:

1. Joint positions J only.
2. Fingers lengths FL with J (i.e. $J \cup FL$).
3. Fingers inter-distances FD with J (i.e. $J \cup FD$).
4. Kinematic distances KD with J (i.e. $J \cup KD$).
5. KD, FD and FL with J (i.e. $J \cup FL \cup FD \cup KD$).

Table 1 shows the quantitative results of the these implementations. In simplest form, the PoseCNN is trained to estimate 3D joint keypoints J only, we call this implementation as our *baseline* (PoseCNN(J)). On top of the *baseline*, we include the structural constraints one by one to observe the effects on the accuracy of estimated joints J. By including fingers lengths FL with J (i.e. PoseCNN($J \cup FL$)), we observe a small increase (3.28%) in accuracy of J. Inclusion of fingers inter-distances FD (PoseCNN($J \cup FD$)) and kinematic distances KD (PoseCNN($J \cup KD$)) improves

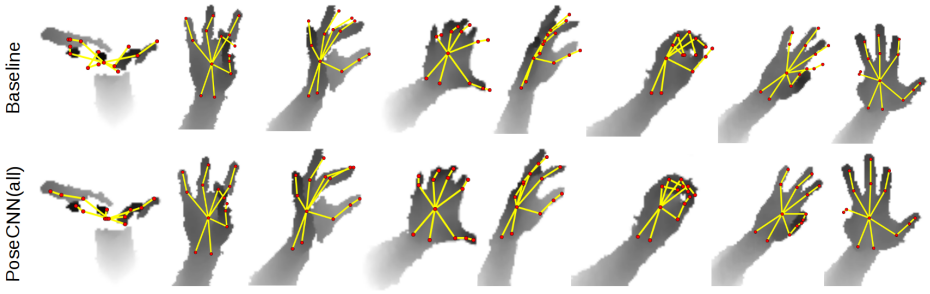

Fig. 3: **Qualitative evaluation of our PoseCNN**. The top row shows the predicted hand joint positions overlaid on the preprocessed NYU depth images from our *baseline* implementation (i.e. PoseCNN(J)). The bottom row shows the corresponding images with corrected joint positions from our PoseCNN(all) implementation.

Methods	3D Joint Loc. Error	3D Palm Center Loc. Error
CoM	14.83mm	28.1mm
Ours (wo/colorization)	13.05mm	15.1mm
Ours (w/colorization)	**11.9mm**	**10.2mm**

Table 3: **Influence of hand segmentation**: Our hand segmentation method without colorization (wo/colorization) improves the joints prediction error by more than $1mm$ over center of hand mass (CoM) calculation method. Our method with colorization (w/colorization) further improves the accuracy by 19.75% over CoM.

the accuracy of the estimated J by 10.5% and 8.55% over the *baseline*, respectively. The best accuracy is achieved by the architecture which includes all the constraints (PoseCNN($J \cup FL \cup FD \cup KD$). It shows 15.13% improvement over the *baseline*.

Dense hand pose representation: We further experiment on the PoseCNN($J \cup FL \cup FD \cup KD$) by interpolating n-points between the pairs of parent and child joints in the kinematic hierarchy of the hand skeleton. Thereby, converting the sparse hand skeleton to dense representation. This leads to increase in number of joint positions depending on the value of n. Consequently, the size of the vectors FD and KD also increases. The quantitative results are summarized in Table 2. Our model (PoseCNN($J \cup FL \cup FD \cup KD$)) with 5-points interpolation performs the best among the others. The results show improvement in accuracy of the estimated J using the interpolation strategy. Therefore, dense hand skeleton representation is useful for improved hand pose regression. For notational simplicity, we call this model as PoseCNN(all). This model improves the accuracy over the *baseline* by 21.71%.

The qualitative comparison of our *baseline* and PoseCNN(all) on the NYU dataset is shown in Figure 3. The estimated joint positions J are displayed on the sample preprocessed depth images. The predicted hand skeleton from our *baseline* architecture (PoseCNN(J)) can be of incorrect size (i.e. shorter or longer) due to independent learning of joint keypoints. Whereas, PoseCNN(all) which incorporates all the constraints along-with interpolated points produces more stable and reliable results. These results

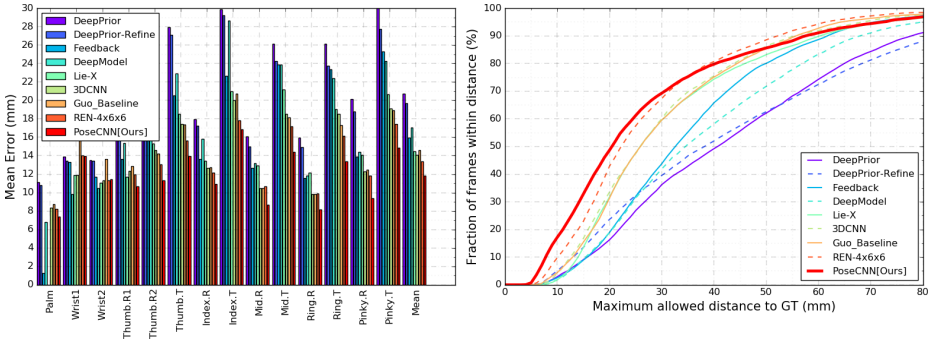

Fig. 4: **Quantitative comparison on the NYU test set** [27]. The right figure shows the fraction of frames within thresholds in mm. The left one shows the mean errors (mm) on individual joints of the NYU hand pose dataset. Our method PoseCNN(all) shows the average error of $11.9mm$ which is better than several state-of-the-art methods.

Methods	3D Joint Location Error
DeepPrior [16]	20.75mm
DeepPrior-Refine [16]	19.72mm
Crossing Nets [28]	15.5mm
Neverova et al. [14]	14.9mm
Feedback [17]	15.9mm
DeepModel [35]	17.0mm
Lie-X [32]	14.5mm
GuoBaseline [6]	14.6mm
3DCNN [5]	14.11mm
REN [6]	13.3mm
DeepPrior++ [15]	12.3mm
PoseCNN(all) [**Ours**]	**11.9mm**

Table 4: **Comparison with the state-of-the-art on the NYU test set** [27]: Our proposed model (PoseCNN(all)) exceeds in accuracy over the state-of-the-art hand pose estimation methods.

clearly show the effectiveness of our novel strategies, namely, structural constraints and the dense hand pose representation.

Hand segmentation: We evaluate our hand segmentation method (see Section 4.1) on the NYU dataset by studying the impact of colorization and comparing with the depth-thresholding followed by center of mass (CoM) computation method. The goal is to observe the effects of hand segmentation on the final 3D pose estimation accuracy. We train two different implementations of the PalmCNN. First, with colorized depth input (Ours(w/colorization)) and second, without colorization (Ours(wo/colorization)). Therefore, we get two different 3D palm centers for cropping the NYU depth images. Also, we obtain 3D palm centers from center of hand mass (CoM) calculation method; see Section 2. Using these three different palm centers, we obtain three distinct sets of pre-processed NYU training and testing frames. The PoseCNN(all) is trained for each of the three training sets. The effects on the accuracy of estimated J from the

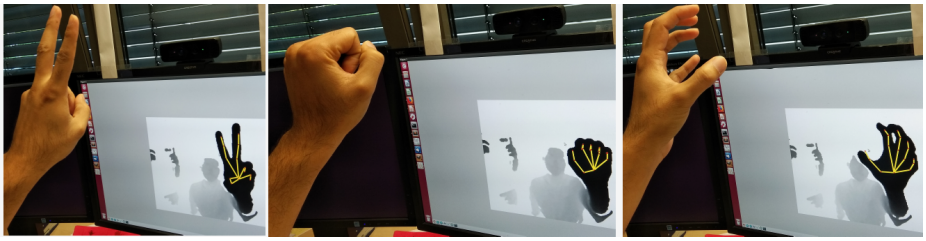

Fig. 5: **Real-time demonstration**: We test our complete pipeline in real-time using the creative Senz3D depth camera [2]. The camera is mounted on top of the display screen. The predicted hand skeleton (yellow) is overlaid on the depth image. Our system successfully tracks various challenging hand poses from frontal camera view.

three PoseCNN(all) models are reported in Table 3. The best results are achieved by Ours(w/colorization) model. It shows an error improvement of 19.75% and 8.81% over the CoM and Ours(wo/colorization) methods; respectively.

Real-time demonstration: We test our complete framework in real-time using a single creative Senz3D depth camera [2]. The camera is placed on top of the display screen. Our framework tracks the hand movements with challenging poses as shown in Figure 5. For better generalization, we train our PoseCNN(all) architecture on the *HandSet* dataset [11]. This dataset combines several public hand pose datasets (e.g. ICVL, NYU and MSRA-2015) in a single unified format. The PalmCNN successfully estimates the hand palm center. Thereafter, the PoseCNN reliably estimates the joint positions. The predicted hand skeleton is displayed on the input depth frame. The run-time of the pipeline is $42ms$.

5.3 Comparison with the State-of-the-art

The state-of-the-art methods use either the ground truth palm center or the CoM localization approach to segment the hand region from a raw depth image. However, these approaches are not feasible for practical applications. In contrast, our CNN-based hand segmentation method automatically segments the hand region from a raw depth image and outperforms the commonly used CoM method (see Table 3). We compare our best performing model, PoseCNN(all), with the state-of-the-art hand pose estimation methods i.e. DeepModel [35], DeepPrior [16], DeepPriorRefine [16], Crossing Nets[28], Feedback [17], LieX[32], GuoBaseline [6] , 3DCNN [5] and REN [6]. The quantitative results are shown in Table 4 and Figure 4. Our algorithm exceeds in accuracy over these methods. The results clearly indicate the benefits of our hand segmentation approach, the interpolation strategy and simultaneous learning of the hand structural constraints with the joint positions.

6 Conclusion

In this paper, we present a novel structure-aware 3D hand pose regression pipeline from a single raw depth image. We propose two strategies which can be easily used to improve the hand pose estimation accuracy of any CNN-based discriminative method. To

this end, a novel CNN-based hand segmentation method regresses the hand palm center which is used to segment the hand region from a raw depth image. Thereafter, a new CNN-based regression network simultaneously estimates the 3D hand pose and its structural constraints. Thereby, enforcing the hand pose structure during the training process. The proposed constraints help to maintain a structural relation between the estimated joint positions. Moreover, we study the effects of n-points interpolation between the pairs of parent and child joints in the kinematic chain of the hand skeleton. By performing extensive evaluations, we show the effectiveness of our approach. Experiments demonstrate competitive performance to the state-of-the-art hand pose estimation methods.

Acknowledgements

This work has been partially funded by the Federal Ministry of Education and Research of the Federal Republic of Germany as part of the research projects DYNAMICS (Grant number 01IW15003) and VIDETE (Grant number 01IW18002).

References

1. Chen, X., Wang, G., Guo, H., Zhang, C.: Pose guided structured region ensemble network for cascaded hand pose estimation. arXiv preprint arXiv:1708.03416 (2017)
2. Creative: Senz3d interactive gesture camera. https://us.creative.com/p/web-cameras/creative-senz3d (March 2018)
3. Dibra, E., Wolf, T., Oztireli, C., Gross, M.: How to refine 3d hand pose estimation from unlabelled depth data? In 3DV (2017)
4. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3593–3601 (2016)
5. Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
6. Guo, H., Wang, G., Chen, X., Zhang, C., Qiao, F., Yang, H.: Region ensemble network: Improving convolutional network for hand pose estimation. In ICIP (2017)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. Li, P., Ling, H., Li, X., Liao, C.: 3d hand pose estimation using randomized decision forest with segmentation index points. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 819–827 (2015)
11. Malik, J., Elhayek, A., Stricker, D.: Simultaneous hand pose and skeleton bone-lengths estimation from a single depth image. In 3DV (2017)

12. Moon, G., Chang, J.Y., Lee, K.M.: V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. arXiv preprint arXiv:1711.07399 (2017)
13. 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: Proceedings of International Conference on Computer Vision (ICCV). vol. 10 (2017)
14. Neverova, N., Wolf, C., Nebout, F., Taylor, G.W.: Hand pose estimation through semi-supervised and weakly-supervised learning. Computer Vision and Image Understanding **164**, 56–67 (2017)
15. Oberweger, M., Lepetit, V.: Deepprior++: Improving fast and accurate 3d hand pose estimation. In: ICCV workshop. vol. 840, p. 2 (2017)
16. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. In CVWW (2015)
17. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3316–3324 (2015)
18. Panteleris, P., Oikonomidis, I., Argyros, A.: Using a single rgb frame for real time 3d hand pose estimation in the wild. arXiv preprint arXiv:1712.03866 (2017)
19. Rad, M., Oberweger, M., Lepetit, V.: Feature mapping for learning fast and accurate 3d pose inference from synthetic images. arXiv preprint arXiv:1712.03904 (2017)
20. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 3633–3642. ACM (2015)
21. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2 (2017)
22. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using rgb and depth data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2456–2463 (2013)
23. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 824–832 (2015)
24. Supancic, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: data, methods, and challenges. In: IEEE international conference on computer vision. pp. 1868–1876 (2015)
25. Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., Pauly, M.: Robust articulated-icp for real-time hand tracking. In: Computer Graphics Forum. vol. 34, pp. 101–114. Wiley Online Library (2015)
26. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3786–3793 (2014)
27. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics (ToG) **33**(5), 169 (2014)
28. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
29. Wan, C., Probst, T., Van Gool, L., Yao, A.: Dense 3d regression for hand pose estimation. arXiv preprint arXiv:1711.08996 (2017)
30. Wan, C., Yao, A., Van Gool, L.: Hand pose estimation from local surface normals. In: European Conference on Computer Vision. pp. 554–569. Springer (2016)

31. Wang, G., Chen, X., Guo, H., Zhang, C.: Region ensemble network: Towards good practices for deep 3d hand pose estimation. Journal of Visual Communication and Image Representation (2018)
32. Xu, C., Govindarajan, L.N., Zhang, Y., Cheng, L.: Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. International Journal of Computer Vision pp. 1–25 (2017)
33. Ye, Q., Yuan, S., Kim, T.K.: Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In: European Conference on Computer Vision. pp. 346–361. Springer (2016)
34. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J.Y., Lee, K.M., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al.: Depth-based 3d hand pose estimation: From current achievements to future goals. In: IEEE CVPR (2018)
35. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In IJCAI (2016)
36. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: International Conference on Computer Vision (2017)