Pose Guided Structured Region Ensemble Network for Cascaded Hand Pose Estimation

Xinghao Chen[a], Guijin Wang[a,*], Hengkai Guo[b], Cairong Zhang[a]

[a]*Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*
[b]*AI Lab, Bytedance Inc., Beijing, China*

Abstract

Hand pose estimation from single depth images is an essential topic in computer vision and human computer interaction. Despite recent advancements in this area promoted by convolutional neural networks, accurate hand pose estimation is still a challenging problem. In this paper we propose a novel approach named as Pose guided structured Region Ensemble Network (Pose-REN) to boost the performance of hand pose estimation. Under the guidance of an initially estimated pose, the proposed method extracts regions from the feature maps of convolutional neural network and generates more optimal and representative features for hand pose estimation. The extracted feature regions are then integrated hierarchically according to the topology of hand joints by tree-structured fully connections to regress the refined hand pose. The final hand pose is obtained by an iterative cascaded method. Comprehensive experiments on public hand pose datasets demonstrate that our proposed method outperforms state-of-the-art algorithms.

Keywords: Hand Pose Estimation, Convolutional Neural Network, Human Computer Interaction, Depth Images

1. Introduction

Accurate 3D hand pose estimation is one of the most important techniques in human computer interaction and virtual reality [1], since it can provide fundamental information for interacting with objects and performing gestures [2, 3]. Hand pose estimation from single depth images has attracted broad research interests in recent years [4, 5, 6, 7, 8, 9, 10, 11] thanks to the availability of depth cameras [12, 13, 14, 15], such as Microsoft Kinect, Intel Realsense Camera etc. However, hand pose estimation is an extremely challenging problem due to the severe self-occlusion, high complexity of hand articulation, noises and holes in depth image, large variation of viewpoints and self-similarity of fingers etc.

Hand pose estimation has achieved great advancements by convolutional neural networks (CNNs). CNN-based data-driven methods either predict heatmaps of hand joints [5, 16] and infer hand pose from heatmaps, or directly regress the 3D coordinates of hand joints [17, 18, 7, 19, 10, 20]. In either ways, features are critical for the performance of hand pose estimation. Prior works mainly focused on incorporating prior knowledge into CNN [17, 20] or using error feedback [18] and spatial attention design [7]. However, few of prior works have paid attentions to extracting more optimal and representative features of CNN. Ye et al. [7] used spatial attention module to select and transform features to a canonical space. Guo et al. [9, 51] proposed the region ensemble network (REN) that divides the feature maps of last convolutional layer into several spatial regions and integrates them in fully connected layers. All aforementioned works haven't fully exploit optimal features of CNN for hand pose estimation.

In this paper, we propose a novel method called pose guided structured region ensemble network (Pose-REN) to boost the performance of hand pose estimation, as shown in Figure 1. Upon an iterative refinement procedure, our proposed method takes a previously estimated pose as input and predicts a more accurate result in each iteration. We present a novel feature extraction method under the guidance of previous predicted hand pose to get optimal and representative features for hand pose estimation. Furthermore, inspired by hierarchical recurrent neural network [21], we present a hierarchical method to fuse features of different joints according to the topology of hand. Features from joints that belong to the same finger are integrated in the first layer and features from all fingers are fused in the following layers to predict the final hand pose.

We evaluate our proposed method on three public hand pose benchmarks [22, 5, 23]. Compared with state-of-the-art methods, our method has achieved the best performance. Extensive ablation analyses illustrate the contributions of different components of the framework and robustness of our proposed method.

The remainder of this paper is organized as follows. In Section 2, we review prior works that are highly related

☆Email addresses: chen-xh13@mails.tsinghua.edu.cn (X. Chen), guohengkaighk@gmail.com (H. Guo), zcr17@mails.tsinghua.edu.cn (C. Zhang). Work was done when H. Guo was with Dept. of EE, Tsinghua University.

*Corresponding author: wangguijin@tsinghua.edu.cn (G. Wang)

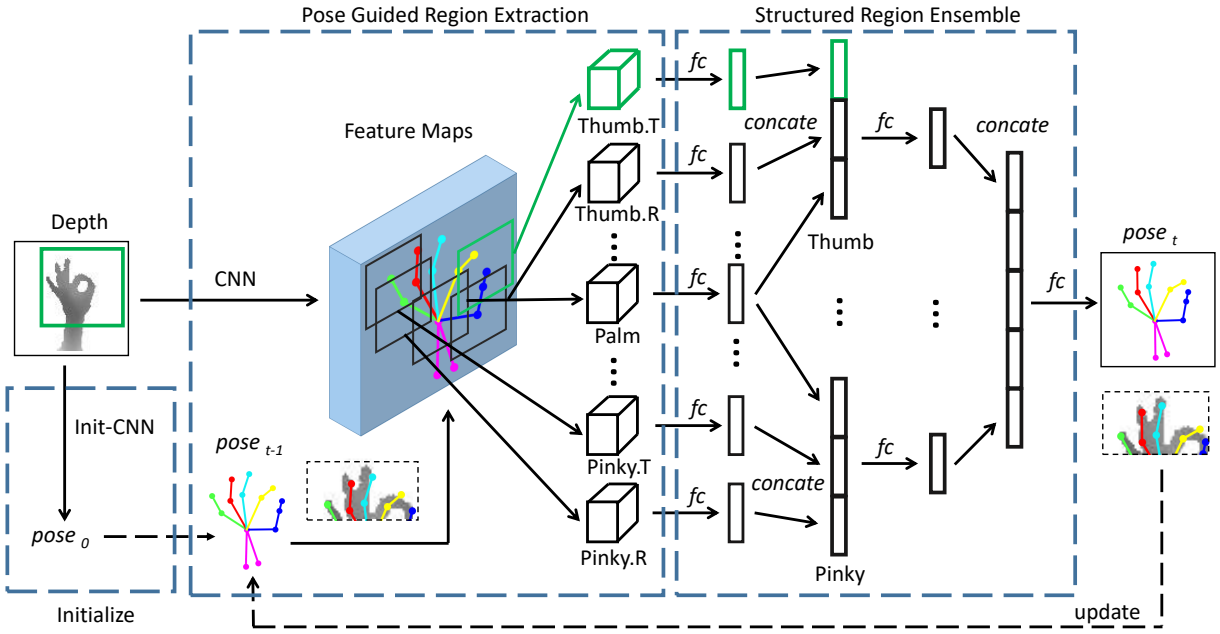


Figure 1: The framework of our proposed pose guided structured region ensemble network (Pose-REN). A simple CNN (Init-CNN) predicts $pose_0$ as the initialization of the cascaded framework. Feature regions are extracted from the feature maps generated by a CNN under the guidance of $pose_{t-1}$ and hierarchically fused using a tree-like structure. $pose_t$ is the refined hand pose obtained by our proposed Pose-REN and will be used as the guidance in next stage.

to our proposed method. In Section 3, we present details about our proposed pose guided structured region ensemble network. Evaluations on public datasets and ablation studies are provided in Section 4. Section 5 gives a brief conclusion of this paper.

2. Related Work

In this section we briefly review related works of our proposed method. Firstly we will review recent algorithms for depth based hand pose estimation. Since our method basically builds upon cascaded framework, we will introduce the cascaded methods for hand pose estimation. Finally, we will review related works about the hierarchical structure of neural network, as the hierarchical structured connections are utilized in our method.

2.1. Depth-based Hand Pose Estimation

Recent approaches of hand pose estimation are generally categorized into three classes: discriminative methods [5, 22, 17, 23, 24, 10, 16, 25, 26, 11, 9], generative methods [27, 28, 29, 30] and hybrid methods [31, 6, 32, 33, 34, 7, 20]. Comprehensive review and analysis on depth based 3D hand pose estimation can be found in [4].

Generative methods fit a predefined hand model to the input data using optimization algorithms to obtain the optimized hand pose, such as PSO (particle swarm optimization) [34], ICP (Iterative Closest Point) [27] and their combination (PSO-ICP) [35]. Hand-crafted energy functions that describe the distance between the hand model and input image are utilized in prior works, such as golden energy [34] and silver energy [6]. Several kinds of hand model have been adopted, including sphere model [35], sphere-meshes model [28], cylinder model [27] and mesh model [34]. Generative methods are robust for self-occlusive areas or missing areas and ensure to output plausible hand pose. However, they need a complex and time-consuming optimizing procedure and are likely to trap into local optimizations.

Discriminative methods directly learn a predictor from the labelled training data. The predictor either predicts the probability maps (heatmaps) of each hand joints [5, 16] or directly predicts the 3D hand joint coordinates [17, 9]. The most frequently used methods for predictor are random forest [36, 22, 37, 6, 23] and convolutional neural network [5, 17, 9, 10, 11]. Discriminative methods do not require any complex hand model and are totally data-driven, which are fast and appropriate for real-time applications. Guo et al. [9, 51] proposed a region ensemble network (REN) that greatly promoted the performance of hand pose estimation based on a single network. Region ensemble network divides the feature maps of last convolutional layer into several spatial regions and integrates them in fully connected layers. However, REN extracts the feature regions using a uniform grid and all features are treated equally, which is not optimal to fully incorporate the spatial information of feature maps and obtain highly representative features.

Hybrid methods try to combine the discriminative and generative methods to achieve better hand pose estimation

performance. Some works adopted the generative methods after obtaining initial results by discriminative methods [31, 33, 34]. Zhou et al. [20] proposed to incorporate a hand model into the CNN, which exploits the constraints of the hand and ensures the geometric validity of the estimated pose. However, hybrid methods have to predefine the properties of the hand model, such as the length of bones. Oberweger et al. [18] proposed a data-driven hybrid method, which learns to generate a depth image from hand pose. However, the generation of depth images is likely affected by the errors of annotations.

Our proposed method basically falls into the category of discriminative method and does not rely on any predefined hand model. Compared with prior CNN-based discriminative methods, our proposed method directly predicts the 3D locations of hand pose using a cascaded framework without any postprocessing procedure. What's more, our proposed pose guided structured region ensemble network (Pose-REN) can learn better features for hand pose estimation by incorporating guided information of previously estimated hand pose into the feature maps and improve the performance of our method.

Although our proposed Pose-REN follows the idea of feature region ensemble as REN [9], there are several essential differences between Pose-REN and REN [9]: 1) Different from REN that uses grid region feature extraction, the proposed Pose-REN fully exploits an initially estimated hand pose as the guided information to extract more representative features from CNN, which is shown to have a large impact for hand pose estimation problem, as discussed in Section 4.4.2. 2) Instead of simple feature fusion as adopted in REN, our Pose-REN presents a structured region ensemble strategy that better models the connections and constraints between different joints in the hand. 3) The Pose-REN is a common framework that can easily be compatible with any existing methods (for example, Feedback [18], DeepModel [20] etc.) by using them to produce initial estimations for Pose-REN.

2.2. Cascaded Method

The cascaded framework has been widely used in face alignment [38, 39, 40], human pose estimation [41, 42] and has also shown good performances in the problem of hand pose estimation [23, 18, 7].

Sun et al. [23] proposed a method to iteratively refine the hand pose using hand-crafted 3D pose index features that are invariant to viewpoint transformation. Oberweger et al. [17] proposed a post-refinement method to refine each joint independently using multiscale input regions centered on the initially estimated hand joints. These works have to train multi models for refinement and independently predict different parts of hand joints while our proposed needs only one model to iteratively improve the estimated hand pose.

Oberweger et al. [18] presented a feedback loop framework for hand pose estimation. One discriminative network is used to produce initial hand pose. A depth image is then generated from the initial hand pose using a generative CNN and an updater network improves the hand pose by comparing the synthetic depth image and input depth image. However, the depth synthetic network is highly sensitive to the annotation errors of hand poses.

Ye et al. [7] integrated cascaded and hierarchical regression into a CNN framework using spatial attention mechanism. The partial hand joints are iteratively refined using transformed features generated by spatial attention module. In their method, the features in cascaded framework are generated by a initial CNN and remain unchanged in each refinement stage except for the spatial transformation. In our proposed method, feature maps are updated in each cascaded stage using an end-to-end framework, which will help to learn more effective features for hand pose estimation.

Our Pose-REN also adopts the cascaded framework. Different from the above prior methods, we present a novel feature extraction method under the guidance of previous predicted hand pose to get optimal and representative features from CNN. What's more, Pose-REN explicitly models the constraints and relations between different hand joints using structured region ensemble strategy, which is a novel method to improve the robustness and performance of hand pose estimation.

2.3. Hierarchical Structure of Neural Network

Du et al. [21] proposed a hierarchical recurrent neural network (RNN) for skeleton-based human action recognition. The whole skeleton is divided into five parts and fed into different branches of the RNN. Different parts of skeleton are hierarchically fused to generated higher-level representations. Madadi et al. [19] proposed a tree-shape structure of CNN which regresses local poses at different branches and fuses all features in the last layer. In their structure, features of different partial poses are learned independently except for sharing features in very early layers. In contrast, our method shares features in the convolutional layers for all joints and hierarchically fuses different regions from feature maps to finally estimate the hand pose. The shared features enables better representation of hand pose and the hierarchical structure of feature fusion can better model the correlation of different hand joints.

3. Pose Guided Structured Region Ensemble Network

In this section, we first give an overview of Pose-REN in Section 3.1. After that we will provide detailed elaboration about extracting regions from the feature maps under the guidance of a hand pose in Section 3.2. In Section 3.3 we present the details of fusing feature regions using hierarchically structured connection. Finally, the training strategy and implementation details are given in Section 3.4 and Section 3.5.

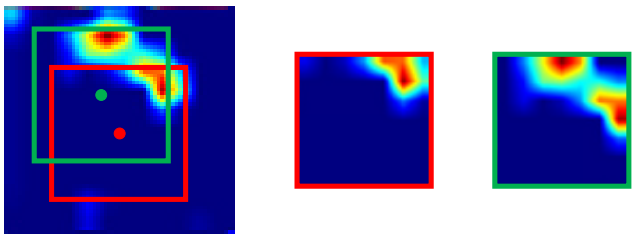


Figure 2: The scheme of our proposed pose guided region extraction. The green and red dots represent two hand joints from previously estimated hand pose. The rectangles of different colors are the corresponding feature regions extracted from the feature maps.

3.1. Overview

The framework of our proposed method is depicted in Figure 1. A simple CNN (denoted as Init-CNN) predicts an initial hand pose $pose_0$, which is used as the initialization of the cascaded framework. The proposed framework takes a previously estimated hand pose $pose_{t-1}$ and the depth image as input. The depth image is fed into a CNN to generate feature maps. Feature regions are extracted from these feature maps under the guidance of the input hand pose $pose_{t-1}$. The insight of our proposed method is that features around the location of a joint contribute more while other features like corner regions are less important. Afterwards, features from different joints are hierarchically integrated using the structured connection to regress the refined hand pose $pose_t$. The images in dash rectangles show the close-up results of $pose_{t-1}$ and $pose_t$. It can be seen that the network refines the hand pose gradually.

Our method aims to estimate the 3D hand pose from a single depth image in a cascaded framework. Specifically, given a depth image $\mathcal{D}$, the 3D locations $\mathcal{P} = \{p_i = (p_{xi}, p_{yi}, p_{zi})\}_{i=1}^{J}$ of J hand joints are inferred. Given a previously estimated hand pose result $\mathcal{P}^{t-1}$ in stage $t-1$, our method uses the learned regression model $\mathcal{R}$ to refine the hand pose in stage t.

$$\mathcal{P}^t = \mathcal{R}(\mathcal{P}^{t-1}, \mathcal{D}) \tag{1}$$

After T stages, we get the final estimated hand pose $\mathcal{P}^T$ for the input depth image $\mathcal{D}$.

$$\mathcal{P}^T = \mathcal{R}(\mathcal{P}^{T-1}, \mathcal{D}) \tag{2}$$

It should be noted that only one same model $\mathcal{R}$ is used in every stage of refinement in the inference phase, see Section 3.4 for details.

3.2. Pose Guided Region Extraction

We first use a standard convolutional neural network (CNN) with residual connections to generate feature maps. The backbone architecture of CNN for generating feature maps used in our method is the same as the baseline network in [9], with 6 convolutional layers and 2 residual connections. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) [43] as the activation function and every 2 convolutional layers are followed by a max pooling layer. The residual connections are added between max pooling layers.

Denote feature maps from the last convolutional layer as $\mathcal{F}$ and the estimated hand pose from previous stage as $\mathcal{P}^{t-1} = \{(p_{xi}^{t-1}, p_{yi}^{t-1}, p_{zi}^{t-1})\}_{i=1}^{J}$. We use $\mathcal{P}^{t-1}$ as the guidance to extract feature regions from $\mathcal{F}$. Specifically, for the i^{th} hand joint, We first project the real-world coordinates into the image pixel coordinates using the intrinsic parameters of the depth camera, as shown in Eq. 3.

$$(p_{ui}^{t-1}, p_{vi}^{t-1}, p_{di}^{t-1}) = proj(p_{xi}^{t-1}, p_{yi}^{t-1}, p_{zi}^{t-1}) \tag{3}$$

The feature region for this joint is then cropped using a rectangular window which can be defined by a tuple $(b_{ui}^t, b_{vi}^t, w, h)$, where b_{ui}^t and b_{vi}^t is the coordinates of top-left corner, w and h is the width and height of the cropped feature region. The coordinates of the rectangular window are calculated by normalizing and converting the original coordinates $(p_{ui}^{t-1}, p_{vi}^{t-1}, p_{di}^{t-1})$ into coordinates in feature maps.

The extracted feature region for hand joint i is then obtained by cropping the feature maps within the rectangular window:

$$\mathcal{F}_i^t = crop(\mathcal{F}; b_{ui}^t, b_{vi}^t, w, h) \tag{4}$$

where the function $crop(\mathcal{F}; b_u, b_v, w, h)$ means extracting the region specified by a rectangular window (b_u, b_v, w, h) from $\mathcal{F}$.

Figure 2 gives an example of pose guided region extraction. The left image is a feature map from the last convolutional layer of the CNN. It should be noted the feature maps usually contains multiple channels, we only use one channel of them to depict how to crop a region guided by a joint. The green dot and red dot indicate two joints (palm center joint and Metacarpophalangeal joint for middle finger respectively) from the previously estimated hand pose. The green and red rectangles are the corresponding cropped windows. The images in middle and right columns show the extracted feature regions for these two joints.

3.3. Structured Region Ensemble

In the previous section we have described how to extract feature regions from the feature maps for each joint using the guidance of previously estimated hand pose. One intuitional way to fuse these feature regions is to connect each region with fully connected (fc) layers respectively and then fuse these layers to regress the final hand pose, which is adopted in REN [9].

Human hand is a highly complex articulated object. Therefore, there are many constraints and correlations between different joints [44, 45]. Independently connecting

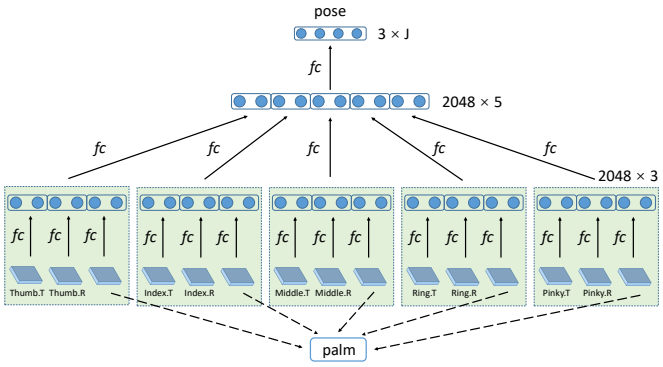


Figure 3: The architecture of the proposed structured region ensemble method. Features from the joints of the same finger (including the palm joint) are fused first. Afterwards, features of different fingers are fused to regress the final hand pose.

feature regions with fc layers and fusing them in the last layer can not fully adopt these constraints. Inspired by hierarchical recurrent neural network [21], in this paper we adopt hierarchically structured region ensemble strategy to better model the constraints of hand joints, as shown in Figure 3. First a set of feature regions $\{\mathcal{F}_j^t\}_{j=1}^M$ are fed into fc layers respectively.

$$h_j^{l_1} = fc(\mathcal{F}_j^t), \quad j = 1, \ldots, M \tag{5}$$

Where M is the number of regions extracted from the feature maps.

Next, $\{h_j^{l_1}\}_{j=1}^M$ are integrated hierarchically according the topology structure of hand. Specifically, denote the indices of joints that belong to the i^{th} finger as $\{I_j^i\}_{j=1}^{M_i}$, where M_i is the number of joints that belong to the i^{th} finger. All joints that belong to the same finger are concatenated (denote as *concate*) and then fed into a fc layer, as shown in Eq. 6 and Eq. 7.

$$\bar{h}_i^{l_1} = concate(\{h_{I_j^i}^{l_1}\}_{j=1}^{M_i}), \quad i = 1, \ldots, 5 \tag{6}$$

$$h_i^{l_2} = fc(\bar{h}_i^{l_1}), \quad i = 1, \ldots, 5 \tag{7}$$

Afterwards, features from different fingers $\{h_i^{l_2}\}_{i=1}^5$ are concatenated and fed into a fc layer to regress the final hand pose $\mathcal{P}^t \in \mathbb{R}^{3\times J}$.

$$\bar{h}^{l_2} = concate(\{h_i^{l_2}\}_{i=1}^5) \tag{8}$$

$$\mathcal{P}^t = fc(\bar{h}^{l_2}) \tag{9}$$

Each fc layer in Eq. 5 and Eq. 7 has a dimension of 2048 nodes. They are followed by ReLU layers and dropout layers with dropout rate of 0.5. The last fc layer output a $3 \times J$ vector $\mathcal{P}^t$ which represents the 3D locations of hand pose.

3.4. Training

Denote the original training set as

$$\mathcal{T}^0 = \{(\mathcal{D}_i, \mathcal{P}_i^0, \mathcal{P}_i^{gt})\}_{i=1}^{N_\mathcal{T}} \tag{10}$$

where $N_\mathcal{T}$ is the number of training samples, $\mathcal{D}_i$ is the depth image, $\mathcal{P}_i^0$ is the initially estimated hand pose and $\mathcal{P}_i^{gt}$ is the corresponding ground truth of hand pose.

In stage t, a regression model $\mathcal{R}^t$ is trained using $\mathcal{T}^{t-1}$. Using this model, we can obtain the refined hand pose for each sample in training set.

$$\mathcal{P}_i^t = \mathcal{R}^t(\mathcal{P}_i^{t-1}, \mathcal{D}) \tag{11}$$

we add the refined samples $\overline{\mathcal{T}^t} = \{(\mathcal{D}_i, \mathcal{P}_i^t, \mathcal{P}_i^{gt})\}_{i=1}^{N_\mathcal{T}}$ to the training set, generating an augmented training set $\mathcal{T}^t$.

$$\mathcal{T}^t = \mathcal{T}^{t-1} \bigcup \overline{\mathcal{T}^t} \tag{12}$$

Again, we train a model $\mathcal{R}^{t+1}$ in stage $t+1$ using $\mathcal{T}^t$ and iteratively repeat this process until reaching the maximum iteration T. The trained model $\mathcal{R}^T$ is the final model used in the inference phase to refine the initial hand pose iteratively, as described in Eq. 1 and Eq. 2.

3.5. Implementation Details

We implemented our proposed method using Caffe [46]. RoI Pooling layer [47] was used to facilitate the implementation of pose guided region extraction.

We used the baseline network in [9] as the Init-CNN to produce initial poses for our method. Generally speaking, any existing hand pose estimation algorithms can be adopted as the initialization method of Pose-REN. We will further discuss the effect of different initializations in Section 4.4.4, including generalization of our pre-trained model to other initializations in inference phase and robustness of Pose-REN to other initializations.

Preprocessing. Similar to previous methods [18, 9], we extracted a fix-sized cube from the input depth image. The center of the cube was determined by calculating the centroid of mass of the hand region. The extracted cube was then resized into a patch with size of 96×96 and the depth values within it were normalized into $[-1, 1]$. Besides, depth values that were outside the cube were truncated according to the size of cube, providing robustness to invalid depth values. The idea of extracting a fix-sized cube is to ensure invariance of the hand size to the distance to the camera.

Training. We first trained the Init-CNN to obtain initial hand pose. After that, we used the weights of trained Init-CNN to initialize Pose-REN and train the network. The whole network was trained using stochastic gradient descent (SGD) with a batch size of 128 and a momentum of 0.9. A weight decay of 0.0005 was also adopted for the network. The learning rate was set to 0.001 and divided by

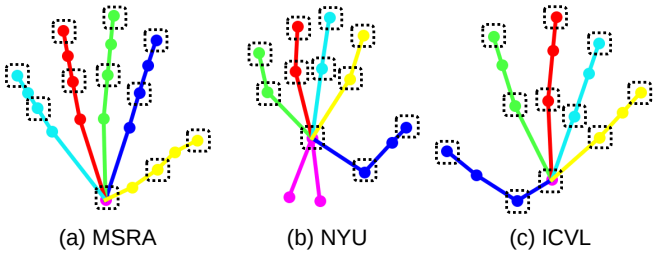


Figure 4: The subset of joints used in pose guided region extraction. The joints circled by dash rectangles are used when extracting feature regions under the guidance of previous joints. Totally $M = 11$ joints are used, including a joint for the palm, two joints for the root and tip of each finger.

10 after every 25 epochs. The model was trained for 100 epochs for each stage and totally trained for two stages.

We followed several good practices that have been proved to be quite effective for hand pose estimation [9], including random data augmentation, smooth L_1 loss. For data augmentation, we applied random scaling of $[0.9, 1.1]$, random translation of $[-10, 10]$ pixels and random rotation of $[-180, 180]$ degrees to the depth image. We used smooth L_1 loss to achieve less sensitivity to the outliers.

Parameter settings. Different datasets have the different number of hand joints, e.g. 21 joints in MSRA dataset and 16 joints in ICVL dataset. To balance the complexity of model and accuracy, we only used part of joints as the guidance to extract feature regions. Specifically, 11 out of all joints were used, as shown in Figure 4. The joints circled by dash rectangles were used, with $M_i = 3$ for each finger, including a joint for the palm, a joint for the root of finger and a joint the tip of finger. It should be noted that despite part of joints ($M = 11$) are utilized as the guidance to extract features, the network still predicts the locations of all joints. The insights behind are that the overlaps of different feature regions make sure the covering of almost all important features even only a part of the joints is used.

In our experiments, the size of extracted region was set to $(w, h) = (7, 7)$. In inference phase, the number of iterations was set to $T = 3$, which will be further discussed in Section 4.4.1.

4. Experiments

In this section, we will first introduce the datasets and evaluation metrics in the experiments. Afterwards we will evaluation our proposed method on three challenging public datasets: ICVL Hand Posture Dataset [22], NYU Hand Pose Dataset [5] and MSRA Hand Pose Dataset [23]. Finally we conduct extensive experiments for ablation study to discuss the effectiveness and robustness of different components of our proposed method.

4.1. Datasets

ICVL Hand Posture Dataset *[22].* This dataset was collected from 10 different subjects using Intel's Creative Interactive Gesture Camera [48]. In-plane rotations are applied to the collected samples and the final dataset contains $330k$ samples for training. There are totally 1596 samples in the testset, including 702 samples for test sequence A and 894 samples for test sequence B. The annotation of hand pose contains 16 joints, including 3 joints for each finger and 1 joint for the palm.

NYU Hand Pose Dataset [5]. The NYU hand pose dataset was collected using three Kinects from different views. The training set contains 72757 frames from 1 subject and the testing set contains 8252 frames from 2 subjects, while one of the subjects in testing set doesn't appear in training set. The annotation of hand pose contains 36 joints. Following the protocol of previous works [5, 17, 18, 20, 9], we only use frames from the frontal view and 14 out of 36 joints in evaluation.

MSRA Hand Pose Dataset [23]. The MSRA hand pose dataset contains 76500 frames from 9 different subjects captured by Intel's Creative Interactive Camera. The leave one subject out cross validation strategy is utilized for evaluation. The annotation of hand pose consists of 21 joints, with 4 joints for each finger and 1 joint for the palm. This dataset has large viewpoint variation, which makes it a rather challenging dataset.

4.2. Evaluation Metric

There are two evaluation metrics widely used in hand pose estimation: per-joint errors and success rate. Denote $\{p_{ij}\}$ as the predicted joint locations of test frames, where i is the index of frame and j is the index of joint. $\{p_{ij}^{gt}\}$ is the corresponding groundtruth label. N is the number of test frames and J is the number of joints in a frame.

Per-joint Errors. Average euclidean distance between predicted joint location and groundtruth for each joint over all test frames. The error for the j^{th} joint is calculated by:

$$err_j = \frac{\sum_i (\|p_{ij} - p_{ij}^{gt}\|)}{N} \tag{13}$$

Average joint error $err = \frac{\sum_j err_j}{J}$ is also used to evaluate the overall performance of hand pose estimation.

Success Rate. The fraction of good frames. A frame is considered as good if the maximum joint error of this frame is within a distance threshold τ. The success rate for distance threshold τ is calculated as Eq. 14.

$$rate_\tau = \frac{\sum_i \mathbb{1}(\max_j(\|p_{ij} - p_{ij}^{gt}\|) \leq \tau)}{N} \tag{14}$$

where $\mathbb{1}(cond)$ is an indicate function that equals to one if $cond$ is true and equals to zero otherwise.

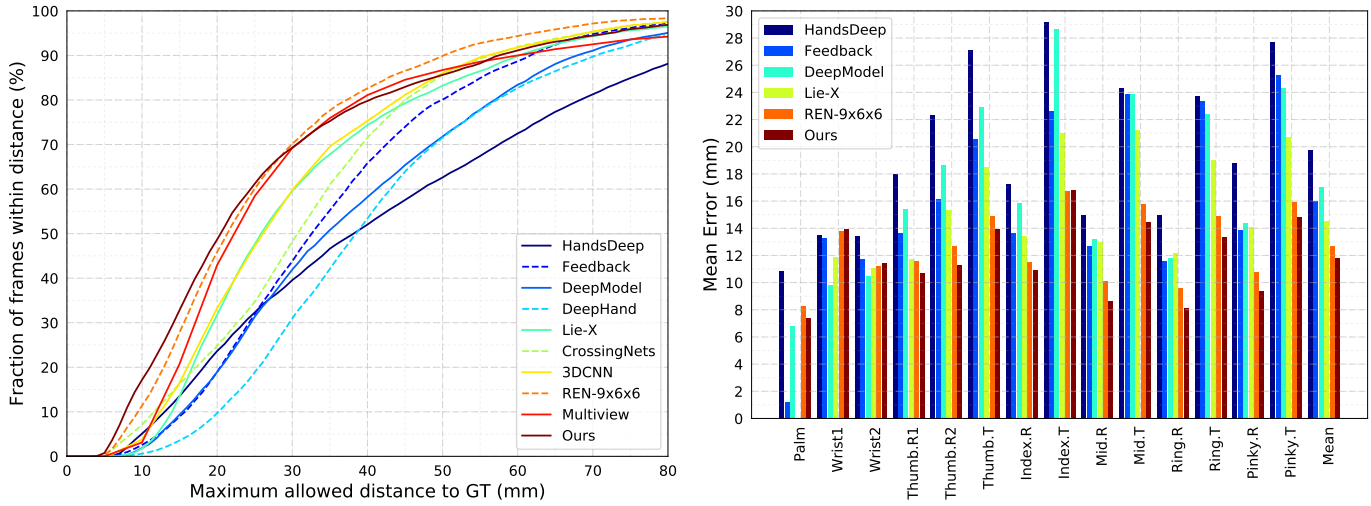


Figure 5: Comparison of our approach with state-of-the-art methods on NYU dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

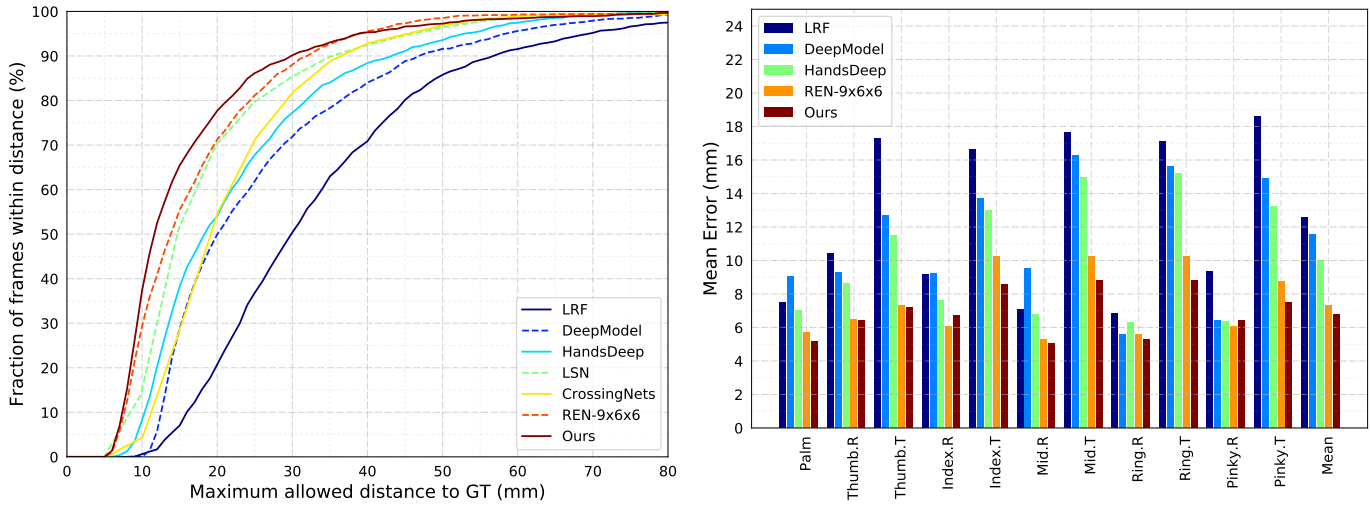


Figure 6: Comparison of our approach with state-of-the-art methods on ICVL dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

4.3. Comparison with State-of-the-Arts

To demonstrate the effectiveness of our proposed method, we compare it against several state-of-the-art methods, including latent random forest (LRF) [22], DeepPrior with refinements (HandsDeep) [17], cascaded hand pose regression (Cascaded) [23], feedback loop (Feedback) [18], deep hand model (DeepModel) [20], Lie group based method (Lie-X) [8], multi-view CNN (Multiview) [16], 3D-CNN based method (3DCNN) [11] , CrossingNets [10], local surface normals (LSN) [24], occlusion aware method (Occlusion) [52], JTSC [50], global to local CNN (Madadi *et al.*) [19] and region ensemble network with $9 \times 6 \times 6$ region setting (REN-9x6x6) [51].

It should be noted that some reported results of state-of-the-art methods are calculated using the predicted labels that are available online [22, 18, 20, 9, 8, 51] and others are estimated from the figures and tables of the original papers [23, 24, 16, 11, 10, 52, 19].

We also compare our method with Mask R-CNN [49] due to its impressive performance on RGB human pose estimation. For fair comparison, we first crop the depth images and resize them into 96×96, which is the same preprocessing as our proposed method. We use similar setting with human pose estimation task in [49] that exploits ResNet-50-FPN as the backbone network. To adopt Mask R-CNN for depth-based hand pose estimation, we first use Mask R-CNN to detect 2D hand pose in image coordinates and then infer depth values from the original depth images to recover 3D hand pose. To alleviate the impact of noises and holes in depth images, the inferred depth values are constrained within the 3D cube of hand and valid depth values from 9-neighbours are averaged to

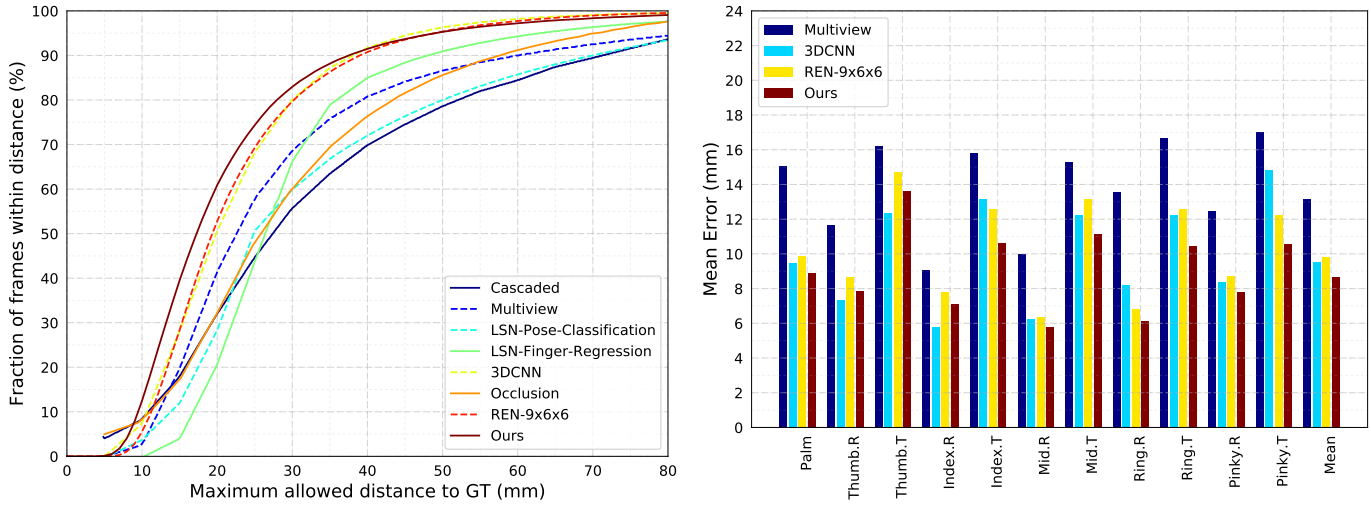


Figure 7: Comparison of our approach with state-of-the-art methods on MSRA dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

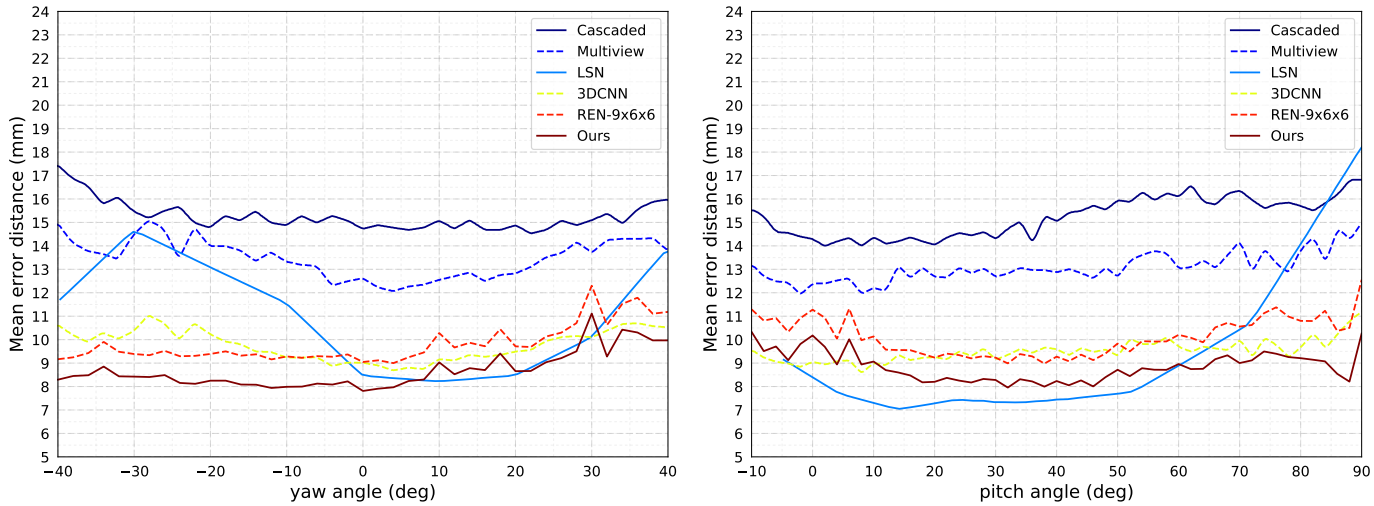


Figure 8: Comparison of mean error distance over different yaw (left) and pitch (right) viewpoint angles on MSRA dataset.

get the final depth coordinate.

On NYU dataset, we compare our proposed method with [17, 18, 20, 53, 8, 10, 16, 11, 51, 49]. The success rate with respect to the worse case criteria and per-joint errors are given in Figure 5. As shown in the figure, our proposed outperforms all state-of-the-art methods. We further compare the overall 2D and 3D mean joint error in Table 1. Our method obtain $0.88mm$ 3D error decrease compared with existing best performance by REN [51]. Mask R-CNN performs 2D keypoint detection and the post-processing is used to lift 2D pose to 3D pose. It achieves comparable 2D error with prior methods. Nevertheless, our Pose-REN outperforms Mask R-CNN and reduces the 2D error by 2.7 pixels.

On ICVL dataset, we compare our proposed method against [22, 20, 17, 10, 24, 51]. Results in Figure 6 demonstrate that our proposed method outperforms all other methods with a large margin. Compared with REN [51], our method reduces the mean error by $0.514mm$, which is a 7.04% relative improvement.

On MSRA dataset, we compare with several state-of-the-art methods [23, 16, 11, 24, 52, 51]. The success rate with respect to maximum allowed threshold and per-joint errors are shown in Figure 7. Our method achieves the best performance among all evaluated methods. Following the protocol of previous works [23], we also report the mean joint errors distributed over yaw and pitch viewpoint angles, as shown in Figure 8. Our method achieves the smallest errors in almost all angles. It should be noted that the LSN [24] get slightly smaller errors when the yaw or pitch angle is relatively small. However, the performance of LSN decreases rapidly when the viewpoint be-

Table 1: Quantitative evaluation of different methods on the benchmark NYU dataset for hand pose estimation task. We report 2D average pixel errors and 3D average joint errors in mm.

Methods	3D error (mm)	2D error (pixels)
HandsDeep [17]	19.73	9.81
Feedback [18]	15.97	8.20
DeepModel [20]	16.90	8.76
Mask R-CNN [49]	27.61	8.25
JTSC [50]	16.80	8.02
Madadi *et al.* [19]	15.60	-
Lie-X [8]	14.51	7.48
REN (4x6x6) [9]	13.39	6.78
REN (9x6x6) [51]	12.69	6.32
Ours	**11.81**	**5.53**

comes larger. These results demonstrate that our method is much more robust to viewpoint changes, which is a quite challenging problem in hand pose estimation.

The fraction of good frames of our method decreases slightly compared with REN [51] when the errors are larger than around $30mm$. This is mainly due to worse initial pose for these challenging samples. When regarding to the per-joint errors, our method achieves the best performance among all compared methods.

4.4. Ablation Study

In this section we will provide extensive experiments to discuss the contributions of different components of our method and the effect of some parameters.

4.4.1. Effect of the Number of Iteration T

First we will discuss how the number of iteration T affects the performance. The average joint errors on NYU dataset with using the different number of iterations are shown in Figure 10. The error for iteration 0 is the result of the initialization. After one iteration, the error drops rapidly. As the iteration increases, the error becomes stable and finally converges. To better balance the computation complexity and performance, we choose the number of iteration as $T = 3$.

4.4.2. Effect of Pose Guided Region Extraction

One of the contributions of our proposed method is to extract feature regions under the guidance of hand pose from previous stage. We will show whether this strategy helps to improve the performance of hand pose estimation. In REN [9], feature regions are extracted using a uniformly distributed grid. We report the performances of our method that only adopts one iteration and sets the number of regions and the size of regions the same as REN-4x6x6 [9] and REN-9x6x6 [51]. Under such experimental settings, the number of parameters of our method and REN are the same, which ensures fair comparison.

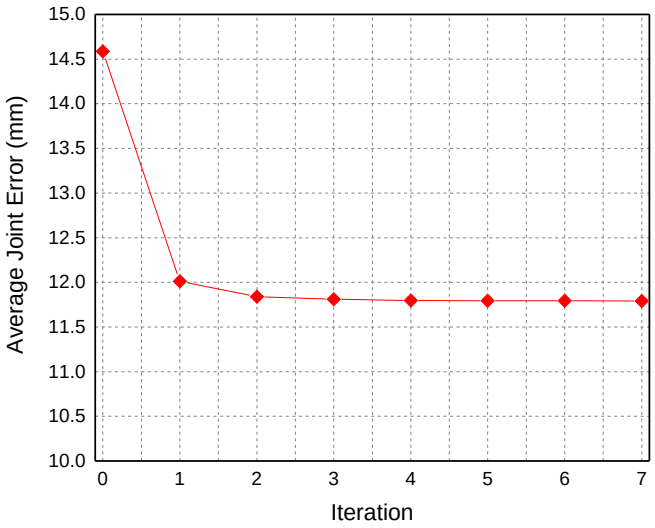


Figure 10: Effect of the number of iteration on NYU dataset.

The first number in the suffix indicates the number of regions and the last two numbers represent the size of regions. Specifically, we use the palm joint, the root joint of thumb, middle, pinky finger in 4x6x6 setting (denoted as Our-4x6x6) and use all joints except for two joints in thumb finger and the tip joint of pinky finger in 9x6x6 setting (denoted as Our-9x6x6). The success rate curve and per-joint errors on NYU dataset are shown in Figure 9. With different region settings, our method both performs better than REN that adopts grid region ensemble, indicating the contributions of pose guided region extraction strategy.

4.4.3. Effect of Structured Region Ensemble

We will demonstrate the effectiveness of another component of our proposed method: the hierarchically structured region ensemble. We compare our method with a network (denoted as $Ours_w/o_structure$) that use two simple fc layers as is adopted in REN [9] instead of hierarchical fc layers. For fair comparison, we set the dimensions of the two fc layers as 2304 and 2048 respectively to ensure the similar number of parameters between our method and $Ours_w/o_structure$. The mean joint errors on NYU, ICVL and MSRA dataset are shown in Table 2. It can be seen that our method performs better than $Ours_w/o_structure$, which illustrates the effectiveness of the hierarchically structured region ensemble strategy.

4.4.4. Effect of the Initialization

In this section we will demonstrate the robustness of our proposed method over different initializations. Our proposed method builds upon the cascaded framework, which takes an initial hand pose as input and iteratively refine the results. To explore the impact of initialization for our methods, we conduct several experiments on NYU dataset with different initializations.

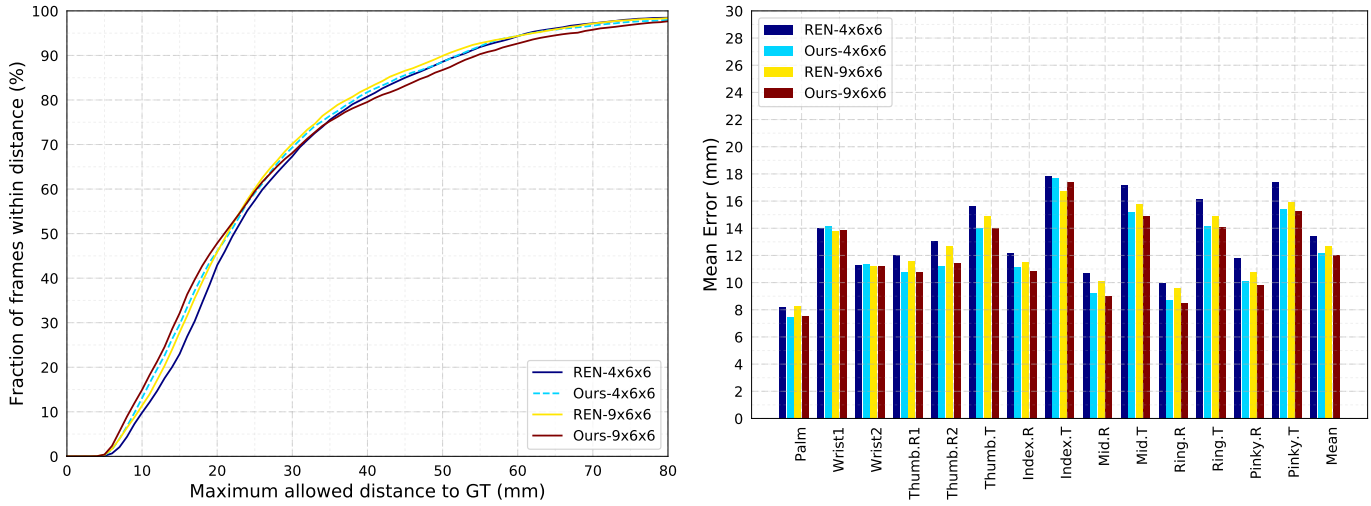


Figure 9: Effect of pose guided region ensemble by comparing our method against grid region ensemble (REN [9]). Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

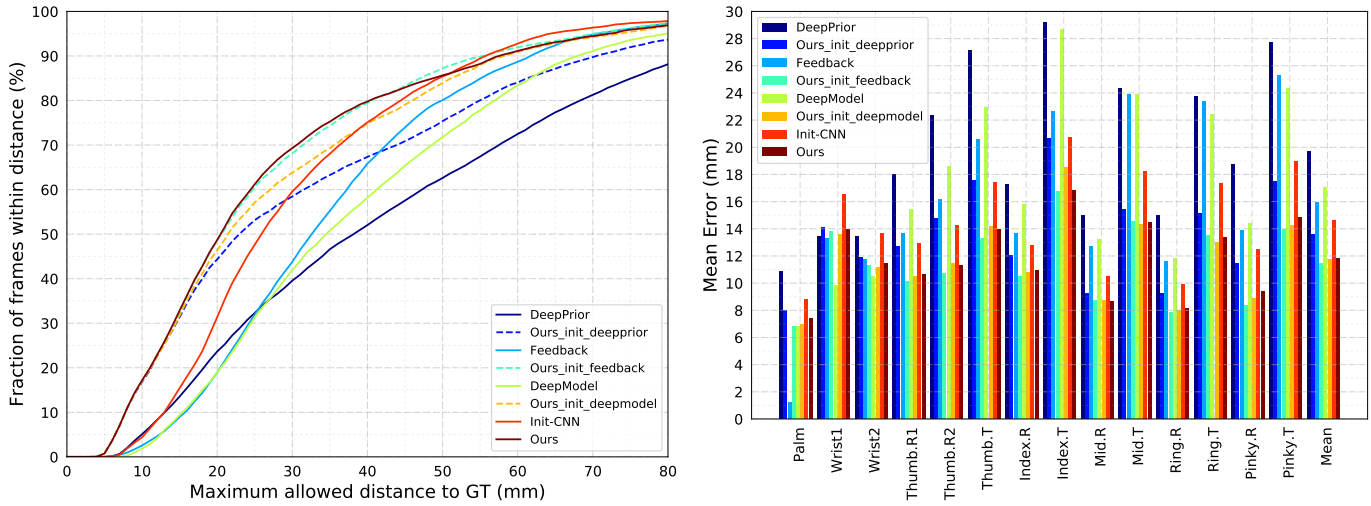


Figure 11: Performance of our model with different initial hand pose used in inference phase on NYU dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

Table 2: Comparing average joint errors of our method with and without structured region ensemble strategy on three datasets. The numbers in the brackets indicate the percentages of error reduction.

Dataset	Ours w/o structure (mm)	Ours (mm)
NYU [5]	11.869	**11.811**(−0.5%)
ICVL [22]	6.932	**6.793**(−2.0%)
MSRA [23]	8.728	**8.649**(−0.9%)

Firstly we will discuss the impact of initialization in inference phase. Specifically, we chose four methods as initialization: Init-CNN (which is proposed in [9] as a baseline network and also adopted as the initialization of our method), DeepPrior [17], Feedback [18], DeepModel [20]. The results of different initializations and refined results (denoted as, e.g. *Ours_init_deepprior*) are shown in Figure 11. We can observe that our method can considerably boost the performances of the initializations. Even with some rather rough initialization (e.g. DeepPrior), the refined results boosted by our Pose-REN are quite competitive. With other better initializations (Feedback, DeepModel), the final results are similar to our method, even if their initializations are slightly worse than ours. These results indicate the robustness over initializations of our method. It should be noted that the model used above were trained using the samples with our initialization (Init-CNN). We used different initializations in inference to get the results above. Therefore, the results above also demonstrate the generalization of our model.

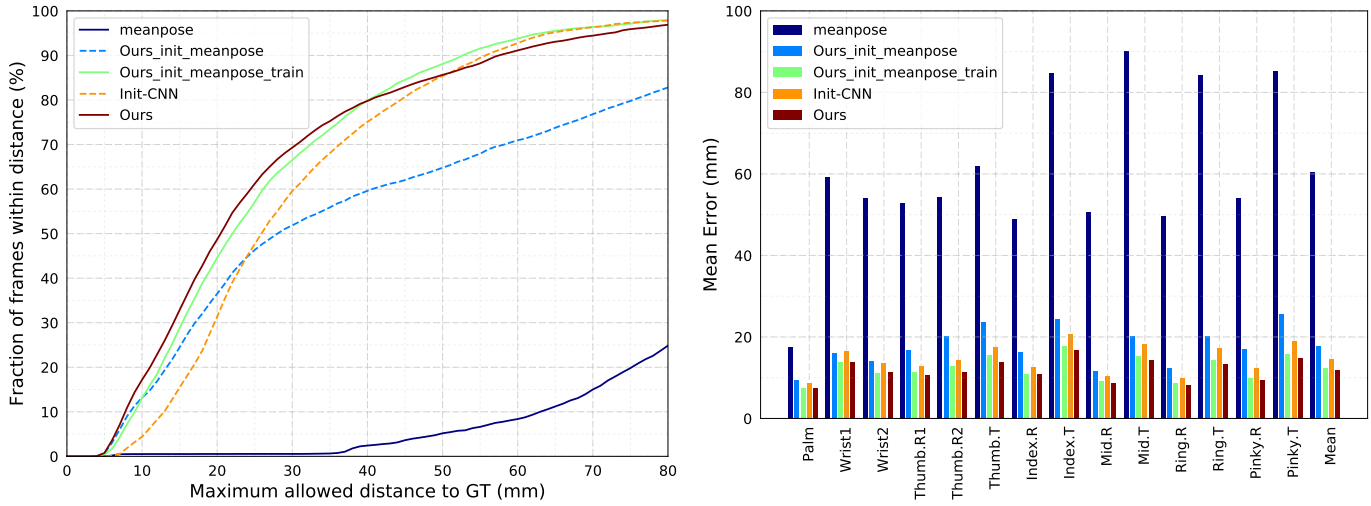


Figure 12: Performance of our method when using mean pose as the initialization on NYU dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

Furthermore, we consider the case that uses a natural pose (denoted as *meanpose*) as initialization and discuss which performance can be expected. We used the model that was trained on our initialization to refine the hand pose with *meanpose* as the initialization. As shown in Figure 12, the initial meanpose is very poor and the results are boosted by adopting our method. We empirically find that the performance converges after 10 stages (*Ours_init_meanpose*), resulting the average joint error of 17.708*mm*, which is comparable with some state-of-the-are methods, as shown in Table 1. We further trained a model using the *meanpose* as initialization and report the refined results (*Ours_init_meanpose_train*) in Figure 12. It can be seen that the results are quite close to those of our method that uses a better initialization, which indicates that our proposed method is robust to different initializations.

As discussed above, the model trained on our initialization greatly generalize to other initializations. Furthermore, for a very poor initialization, our proposed method can still obtain satisfying results by training a model using this initialization.

4.5. Qualitative Results

Figure 13 shows some examples of the iterative process on NYU dataset. The first column shows the results of the initialize hand pose, the second to fourth columns show the refined results on stage $1-3$. The rightmost column is the groundtruth annotation. Our method gradually improves the estimated hand pose and obtains accurate results after several iterations.

Some qualitative results on three datasets can be seen in Figure 14. For each dataset, the first row represents the results of REN-9x6x6 [51], the second row shows the results of our proposed method and the third row is the groundtruth. It can be seen that our method performs better than REN even in some challenging samples.

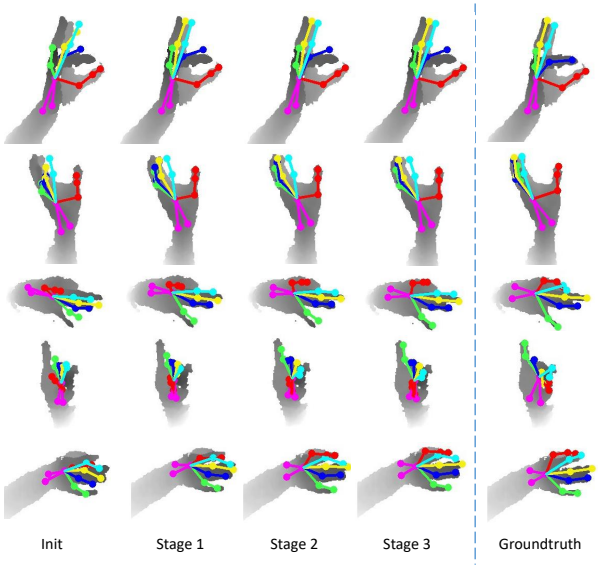


Figure 13: Qualitative results on NYU dataset of different stages.

5. Conclusion

In this paper we propose a novel method called pose guided structured region ensemble network (Pose-REN) for accurate 3D hand pose estimation from a single depth image. Our method extracts regions from the feature maps under the guidance of an initially estimated hand pose to attain more optimal and representative features. Feature regions are then integrated hierarchically by adopting a tree-like structured connection that models the topology of hand joints. Our method iteratively refines the hand

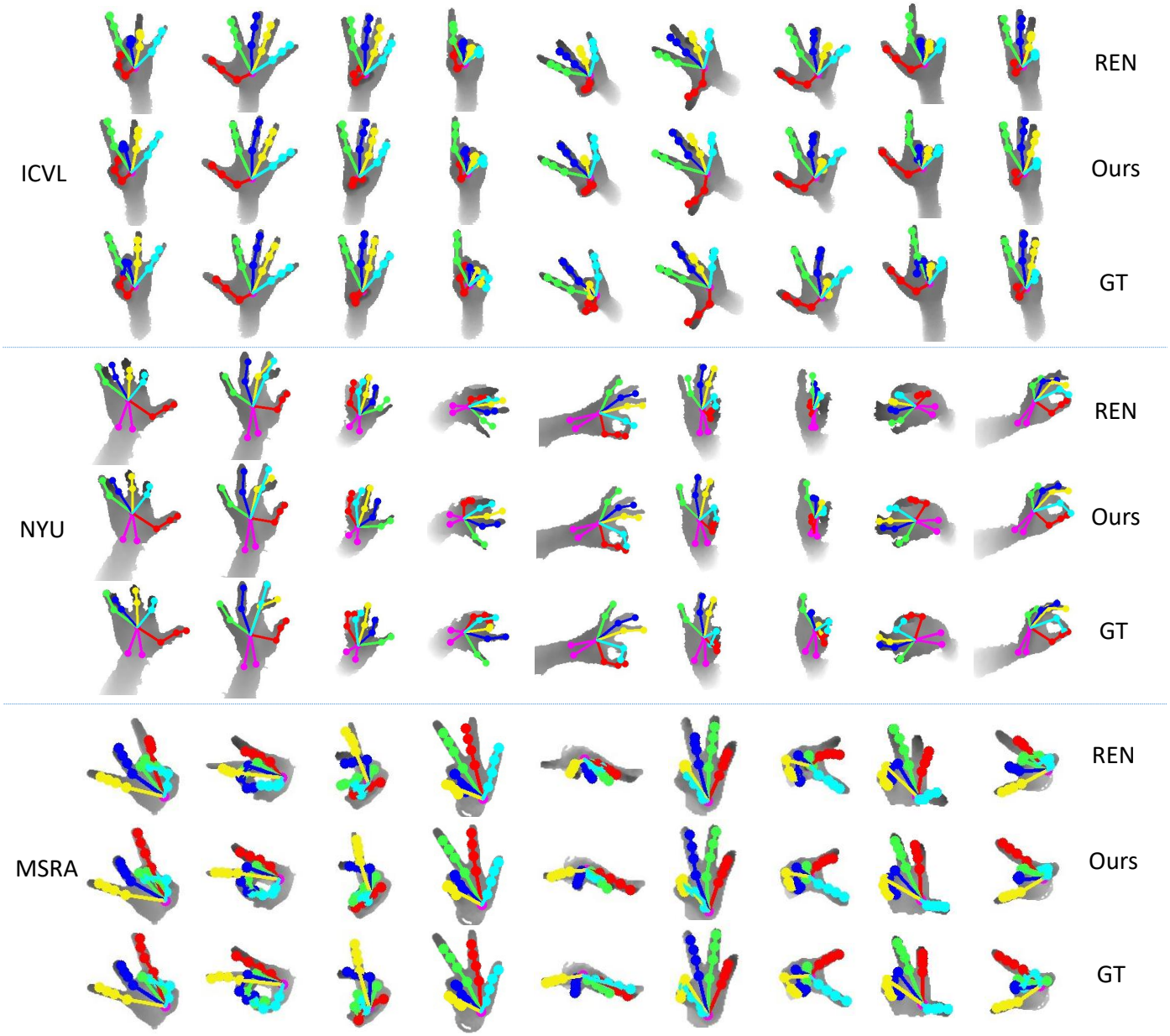


Figure 14: Qualitative results. For each dataset, three rows show the results from region ensemble network (REN-9x6x6) [51], our method (Ours) and groundtruth (GT) respectively.

pose to obtain the final estimated results. Experiments on public hand pose datasets demonstrate that our proposed method outperforms all state-of-the-art methods. In our future work, we intend to further improve our method for robust and accurate 3D hand pose estimation when hands are interacting with other hands or objects. We would like to research on integrating hand detection and hand pose estimation into a unified framework, based on Faster R-CNN[54] or Mask R-CNN [49] etc. It will also be interesting to apply our proposed method for more articulated pose estimation tasks, like human pose estimation and face alignment.

Acknowledgments

This work was partially supported by State High-Tech Research and Development Program of China (863 Program) [grant number 2015AA016304].

References

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, X. Twombly, Vision-based hand pose estimation: A review, Computer Vision and Image Understanding 108 (1) (2007) 52–73.

[2] X. Chen, H. Guo, G. Wang, L. Zhang, Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition, in: Image Processing (ICIP), 2017 24th IEEE International Conference on, IEEE, 2017, pp. 2881–2885.

[3] Q. De Smedt, H. Wannous, J.-P. Vandeborre, Skeleton-based dynamic hand gesture recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–9.

[4] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, D. Ramanan, Depth-based hand pose estimation: data, methods, and challenges, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1868–1876.

[5] J. Tompson, M. Stein, Y. Lecun, K. Perlin, Real-time continuous pose recovery of human hands using convolutional networks, ACM Transactions on Graphics (TOG) 33 (5) (2014) 169.

[6] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, J. Shotton, Opening the black box: Hierarchical sampling optimization for estimating human hand pose, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3325–3333.

[7] Q. Ye, S. Yuan, T.-K. Kim, Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation, in: The European Conference on Computer Vision (ECCV), 2016, pp. 346–361.

[8] C. Xu, L. N. Govindarajan, Y. Zhang, L. Cheng, Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups, International Journal of Computer Vision (2017) 1–25doi:10.1007/s11263-017-0998-6.

[9] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, H. Yang, Region ensemble network: Improving convolutional network for hand pose estimation, in: Image Processing (ICIP), 2017 IEEE International Conference on, IEEE, 2017.

[10] C. Wan, T. Probst, L. Van Gool, A. Yao, Crossing nets: Dual generative models with a shared latent space for hand pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[11] L. Ge, H. Liang, J. Yuan, D. Thalmann, 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1991–2000.

[12] Z. Zhang, Microsoft kinect sensor and its effect, IEEE multimedia 19 (2) (2012) 4–10.

[13] G. Wang, X. Yin, X. Pei, C. Shi, Depth estimation for speckle projection system using progressive reliable points growing matching, Applied optics 52 (3) (2013) 516–524.

[14] C. Shi, G. Wang, X. Yin, X. Pei, B. He, X. Lin, High-accuracy stereo matching based on adaptive ground control points, IEEE Transactions on Image Processing 24 (4) (2015) 1412–1423.

[15] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, A. Bhowmik, Intel realsense stereoscopic depth cameras, arXiv preprint arXiv:1705.05548.

[16] L. Ge, H. Liang, J. Yuan, D. Thalmann, Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3593–3601.

[17] M. Oberweger, P. Wohlhart, V. Lepetit, Hands deep in deep learning for hand pose estimation, in: Proceedings of Computer Vision Winter Workshop, 2015, 2015, pp. 21–30.

[18] M. Oberweger, P. Wohlhart, V. Lepetit, Training a feedback loop for hand pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3316–3324.

[19] M. Madadi, S. Escalera, X. Baro, J. Gonzalez, End-to-end global to local cnn learning for hand pose recovery in depth data, arXiv preprint arXiv:1705.09606.

[20] X. Zhou, Q. Wan, W. Zhang, X. Xue, Y. Wei, Model-based deep hand pose estimation, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2016, pp. 2421–2427.

[21] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110–1118.

[22] D. Tang, H. Jin Chang, A. Tejani, T.-K. Kim, Latent regression forest: Structured estimation of 3d articulated hand posture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3786–3793.

[23] X. Sun, Y. Wei, S. Liang, X. Tang, J. Sun, Cascaded hand pose regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 824–832.

[24] C. Wan, A. Yao, L. Van Gool, Hand pose estimation from local surface normals, in: European Conference on Computer Vision, Springer, 2016, pp. 554–569.

[25] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, C. Keskin, Learning to navigate the energy landscape, in: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp. 323–332.

[26] Y. Zhang, C. Xu, L. Cheng, Learning to search on manifolds for 3d pose estimation of articulated objects, arXiv preprint arXiv:1612.00596.

[27] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, M. Pauly, Robust articulated-icp for real-time hand tracking, in: Computer Graphics Forum, Vol. 34, Wiley Online Library, 2015, pp. 101–114.

[28] A. Tkach, M. Pauly, A. Tagliasacchi, Sphere-meshes for real-time hand modeling and tracking, ACM Transactions on Graphics (TOG) 35 (6) (2016) 222.

[29] D. Joseph Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, J. Shotton, Fits like a glove: Rapid and reliable hand shape personalization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5610–5619.

[30] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, et al., Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences, ACM Transactions on Graphics (TOG) 35 (4) (2016) 143.

[31] P. Krejov, A. Gilbert, R. Bowden, Combining discriminative and model based approaches for hand pose estimation, in: Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 1, IEEE, 2015, pp. 1–7.

[32] C. Choi, A. Sinha, J. Hee Choi, S. Jang, K. Ramani, A collaborative filtering approach to real-time hand pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2336–2344.

[33] S. Sridhar, F. Mueller, A. Oulasvirta, C. Theobalt, Fast and robust hand tracking using detection-guided optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3213–3221.

[34] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al., Accurate, robust, and flexible real-time hand tracking, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2015, pp. 3633–3642.

[35] C. Qian, X. Sun, Y. Wei, X. Tang, J. Sun, Realtime and robust hand tracking from depth, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1106–1113.

[36] D. Tang, T.-H. Yu, T.-K. Kim, Real-time articulated hand pose estimation using semi-supervised transductive regression forests, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 3224–3231.

[37] H. Liang, J. Yuan, D. Thalmann, Parsing the hand in depth images, IEEE Transactions on Multimedia 16 (5) (2014) 1241–1253.

[38] S. Zhu, C. Li, C. Change Loy, X. Tang, Face alignment by coarse-to-fine shape searching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4998–5006.

[39] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: European Conference on Computer Vision, Springer, 2014, pp. 109–122.

[40] M. Kowalski, J. Naruniec, T. Trzcinski, Deep alignment net-

work: A convolutional neural network for robust face alignment, arXiv preprint arXiv:1706.01789.
[41] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660.
[42] J. Carreira, P. Agrawal, K. Fragkiadaki, J. Malik, Human pose estimation with iterative error feedback, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4733–4742.
[43] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Citeseer, 2013.
[44] J. Lin, Y. Wu, T. S. Huang, Modeling the constraints of human hand motion, in: Human Motion, 2000. Proceedings. Workshop on, IEEE, 2000, pp. 121–126.
[45] Y. Wu, T. S. Huang, Hand modeling, analysis and recognition, IEEE Signal Processing Magazine 18 (3) (2001) 51–60.
[46] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093.
[47] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[48] S. Melax, L. Keselman, S. Orsten, Dynamics based 3d skeletal hand tracking, in: Proceedings of Graphics Interface 2013, Canadian Information Processing Society, 2013, pp. 63–70.
[49] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017.
[50] D. Fourure, R. Emonet, E. Fromont, D. Muselet, N. Neverova, A. Trémeau, C. Wolf, Multi-task, multi-domain learning: application to semantic segmentation and pose regression, Neurocomputing 251 (2017) 68–80.
[51] G. Wang, X. Chen, H. Guo, C. Zhang, Region ensemble network: Towards good practices for deep 3d hand pose estimation, Journal of Visual Communication and Image Representationdoi:https://doi.org/10.1016/j.jvcir.2018.04.005.
[52] M. Madadi, S. Escalera, A. Carruesco, C. Andujar, X. Baró, J. Gonzàlez, Occlusion aware hand pose recovery from sequences of depth images, in: Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on, IEEE, 2017, pp. 230–237.
[53] A. Sinha, C. Choi, K. Ramani, Deephand: Robust hand pose estimation by completing a matrix imputed with deep features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4150–4158.
[54] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.