Hand Pose Regression via A Classification-guided Approach

Hongwei Yang and Juyong Zhang(✉)

University of Science and Technology of China, Hefei, Anhui, China
juyong@ustc.edu.cn

Abstract. Hand pose estimation from single depth image has achieved great progress in recent years, however, up-to-data methods are still not satisfying the application requirements like in human-computer interaction. One possible reason is that existing methods try to learn a general regression function for all types of hand depth images. To handle this problem, we propose a novel "divide-and-conquer" method, which includes a classification step and a regression step. At first, a convolutional neural network classifier is used to classify the input hand depth image into different types. Then, an effective and efficient multiway cascaded random forest regressor is used to estimate the hand joints' 3D positions. Experiments demonstrate that the proposed method achieves state-of-the-art performance on challenging dataset. Moreover, the proposed method can be easily combined with other regression method.

1 Introduction

In recent years, the problem of pose estimation of 3D articulated objects such as human body [1–3] and hand [4–11] from markerless visual observations has been widely studied due to their wide applications on human-computer interaction, Augmented Reality (AR), motion sensing game, robotic control, etc. In the earlier period, researchers estimated gestures from the 2D RGB images [12–14] or videos [15]. Along with the development of hardware technology, in particular, low-cost commodity depth cameras like MicroSoft Kinect, PrimeSense and Intel RealSense have emerged in recent years. Human body and hand pose estimation from RGB-D data [7, 16–26] have noticeably progressed after the introduction of depth sensors.

Since human hand has large viewpoint variance, self-occlusion and similarity between fingers, hand pose estimation based on markerless visual observations is still a challenging task. Although it is an extremely difficult problem, a lot of literatures about hand pose estimation have been proposed in recent years. In the survey [27], vision-based markerless hand tracking algorithms were roughly classified into two types, model-based and appearance-based approaches.

Model-based methods often fit a hand template to the input data to estimate hand poses. Pose estimation can be formulated as a optimization problem[4, 16] or nearest-neighbor search problem [28]. Recently, Sridhar et al. [29, 30] proposed an accurate model-based hand tracking method in a multiple camera setup, with the purpose of resolving serious self-occlusions. Although multiple camera setup can achieve more precise pose recovery, the complex acquisition setup and manual calibration is less suitable for the consumer-level applications. In [4, 16, 31, 32], the 3D hand poses were reconstructed via inverse kinematics techniques, which optimize a nonlinear energy function that is extremely difficult to find global minimum. The numerical optimization algorithms of above methods usually are complex, time-consuming and easy to trap into local minima. Therefore, it limits their usages in real-time and accurate applications.

On the other hand, appearance-based approaches use direct mapping techniques which try to learn a direct mapping from the input image space to the output pose space. During the past few years, numerous appearance-based methods based on nearest neighbor search [20, 28], decision forest [5, 6, 8, 18, 33, 34] or convolutional networks [21, 23, 26, 35] have been developed for hand pose estimation. In [5], Keskin et al. introduced a multi-layered randomized decision forest framework. They divided a whole classification task into two classification stages, which only focus on different learning task and make the whole learning task more efficient and accurate. Recently, Sun et al. [33] presented a cascaded hierarchical regression approach with 3D pose-indexed features. Although the 3D pose-indexed features achieve approximatively strict 3D invariance and cascaded framework reinforces learning ability. It is extremely difficult to handle complicated hand poses estimation by only one regression model. Therefore, it is a good choice to improve the whole learning ability by introducing multiway regression models.

In this paper we propose a novel "divide-and-conquer" classification-guided regression learning framework to estimate hand pose from single depth image. At first, in order to reduce the search space of regression, a convolutional network based classifier is introduced to predict the hand gesture type. Cascaded random forest regressors for different hand gesture types are trained on disjoint part of training dataset. Then based on the predicted class of classifier, a corresponding regressor is selected to estimate the final hand pose. It means that the classifier divides the learning task and the regressors conquer their own task. The algorithm pipeline is shown in Fig. 1.

Our main contributions are as follow: a new classification-guided hand pose regression framework is developed. Based on the training dataset of regression, we train a classifier to partition a complex and difficult regression learning task into several more easier subproblems. Then each regressor only focuses on a part of training data so that it is more professional and accurate. The classification-guided regression approach outperforms the state-of-the-art methods. More broadly speaking, other discriminative hand pose regression model can be used as the regressor module in our framework and further improve the pose estimation accuracy.

This paper is organized as follows: Section 2 describes our proposed framework, which includes the new convolutional networks hand pose classifier and the cascaded random regression forest. After that, we introduce the experimen-

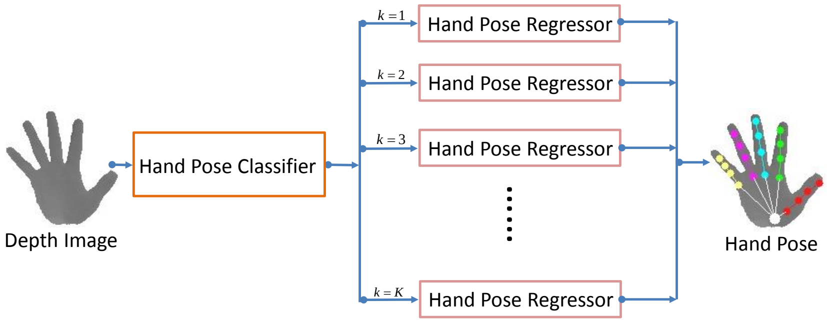


Fig. 1. Algorithm pipeline. We integrate a hand pose classifier and several hand pose estimation regressors into one framework. The hand pose classifier predicts the class of gesture from the depth image. Based on the predicted class, one of hand pose regressors is selected to estimate the final hand joints' 3D locations.

tal details and analyze quantitative and qualitative results of experiments in Section 3. Finally, we conclude this paper in Section 4.

2 Methodology

2.1 Hand Model and Method Overview

We use a hand skeleton model as illustrated in Fig. 2. The hand pose $\mathbf{\Theta} = \{\mathbf{p}_i\}_{i=1}^{21}$, where $\mathbf{p}_i = (x_i, y_i, z_i)$, represents 21 kinematic joints' 3D positions. We divide the hand pose $\mathbf{\Theta}$ into six parts including the palm $\mathbf{\Theta}^p$ (6 joints of the palm) and five fingers $\mathbf{\Theta}^f$ (each 3 joints of the maniphalanx), where $f \in F = \{1, 2, 3, 4, 5\}$.

An overview of our pipeline is showed in Fig. 1. We estimate the hand pose $\mathbf{\Theta}$ in the form of the 3D locations of its joints from a single depth image I. And we denoted a training dataset by $\{(I_i, \mathbf{\Theta}_i)\}_{i=1}^{N}$, each element of which is a depth image labeled with its corresponding ground truth joints' locations. Our proposed method integrates the hand pose classifier and hand pose regressor into one framework. In advance, the training dataset of regression is clustered into K subsets. And based on the clustering result, a hand pose classifier are trained on the entire dataset, but several hand pose regressors are respectively trained on each subset. At the testing stage, the hand pose classifier infers the hand pose class k from the depth image first. Then the hand pose regressor which is corresponding to k-class estimates the final hand pose $\mathbf{\Theta}$.

2.2 Clustering Training Data

To train the hand pose classifier, we utilize the existing dataset $\{(I_i, \mathbf{\Theta}_i)\}_{i=1}^{N}$ of hand pose regression to generate the training dataset $\{(I_i, L_i)\}_{i=1}^{N}$ of classifier. Thus, we cluster the hand joints' position vector $\mathbf{\Theta}_i$ to generate corresponding

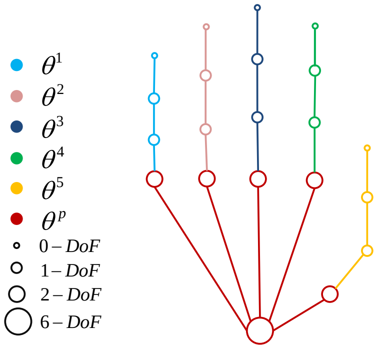


Fig. 2. A 21-joint representing of a canonical hand pose. The palm root (wrist joint) encodes 6 degrees of freedom (DoF) of the global rotation and translation. Each finger and its corresponding root point on the palm (4 joints in total) encode 4 degrees of freedom of finger articulation.

target label L_i for depth image I_i. The K-Means clustering algorithm with rigid alignment is used to cluster the hand poses. Since $\mathbf{\Theta}_i$ is related to camera viewpoint, the rigid registration is applied to the hand pose to remove the affects caused by the camera viewpoint. The rigid registration procedure is as follows:

$$\mathcal{T}_i = (\mathbf{R}_i, \mathbf{t}_i) = RigidAlig(\mathbf{\Theta}_C^p, \mathbf{\Theta}_i^p), \quad i = 1, 2, ..., N,$$

where N is the number of the training set, $\mathbf{\Theta}_C$ is a canonical hand pose, which is arbitrarily chosen from $\{(I_i, \mathbf{\Theta}_i)\}_{i=1}^N$. $RigidAlig$ is refered as rigid registration and it is achieved by Iterated Closest Point algorithm [36], which is used to compute the rigid transformation between the palm joints of canonical hand pose and each other hand pose. And $\mathbf{R}_i, \mathbf{t}_i$ respectively represent rotation and translation. The rigid transformation $\mathcal{T}_i$ aligns each hand pose $\mathbf{\Theta}_i$ of training dataset to a certain coordinate system that determined by canonical hand pose. Then we cluster the aligned training data $\{\mathbf{R}_i\mathbf{\Theta}_i + \mathbf{t}_i\}_{i=1}^N$ into K classes by K-Means clustering. Therefore, the target label L_i of depth image I_i equals its corresponding class of hand pose $\mathbf{\Theta}_i$ of K-Means cluster.

2.3 CNN Classifier

Hand pose classifier predicts the hand pose type k for each input depth image I. As the hand pose registration space and variation of camera viewpoints are very large and complex, it is difficult to directly classify hand poses based on depth images which are the only input information. Because of the excellent performance of Convolutional neural network(CNN) on complex and large-scale image classification task [37], we adopt CNN method in this work to classify the hand poses.

In this paper, the hand pose classifier is based on a standard CNN framework (Fig. 3). The CNN, similar to fully-connected neural networks, performs

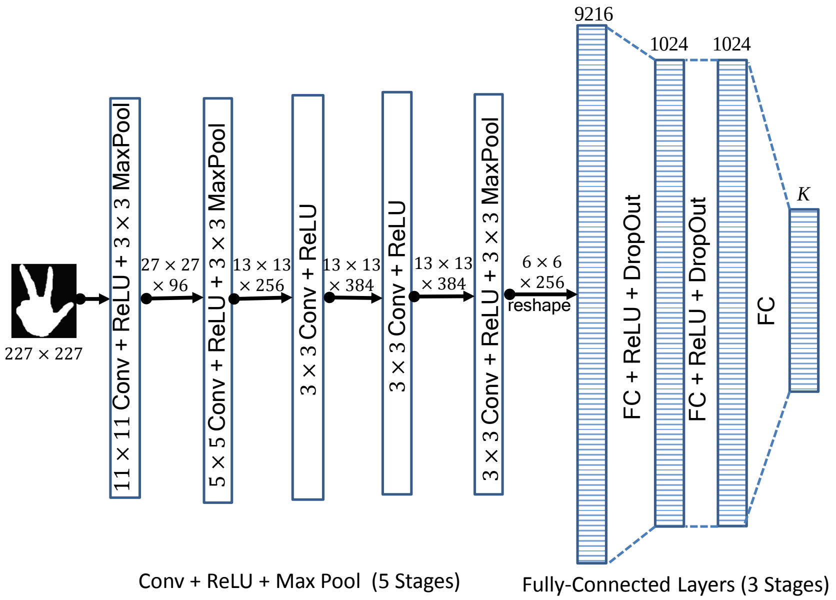

227 × 227
11 × 11 Conv + ReLU + 3 × 3 MaxPool
27 × 27 × 96
5 × 5 Conv + ReLU + 3 × 3 MaxPool
13 × 13 × 256
3 × 3 Conv + ReLU
13 × 13 × 384
3 × 3 Conv + ReLU
13 × 13 × 384
3 × 3 Conv + ReLU + 3 × 3 MaxPool
6 × 6 × 256
reshape
9216
FC + ReLU + DropOut
1024
FC + ReLU + DropOut
1024
FC
K
Conv + ReLU + Max Pool (5 Stages)
Fully-Connected Layers (3 Stages)

Fig. 3. The convolutional network architecture used in our paper. The network contains five convolutional layers and three fully-connected layers.

end-to-end feature learning and is trained with the back-propagation algorithm. However, they are different in many respects, most notably local connectivity, weight sharing, and local pooling. The first two properties significantly reduce the number of free parameters and the need to learn repeated feature detectors at different locations of the input. The third property makes the learned representation invariant to small translations of the input [38].

The CNN classifier is illustrated in Fig. 3. In the original depth image, the proportion of background pixels is much greater than hand pixels. Therefore, we crop the bounding box of the hand region from original depth image and resize it according to requirement of input data. The input is then processed by five stages of convolution and subsampling, which use rectified linear units (ReLUs) [39] and max-pooling. The convolution kernel stride of the first convolution layers is 4, and others are zero. The padding of the five convolution layers are respectively $0, 2, 1, 1$ and 1. Internal pooling layers contribute to reduce computational complexity and improve classification tolerance for small input image translations. Unfortunately, pooling also results in a loss of spatial precision. Since invariance to input translations can be learned with sufficient training exemplars, we only choose three stages of pooling where the stride is 2.

Following the five convolution and subsampling layers, the top-level pooled map is flattened to a vector and processed by three fully connected layers. Each of these output stages is composed of a linear matrix-vector multiplication with learned bias, followed by a point-wise non-linearity (ReLU). Dropout[40] is used on the input to each of fully-connected linear stages to reduce over-fitting for the restricted-size training set. There are two dropout layers with dropout ratio 0.5 that behind the first two fully connected layers. The output layer has a K-ways softmax unit which produces a distribution over K hand pose classes.

2.4 Hand Pose Regression

To estimate the hand pose $\mathbf{\Theta}$, we adopt the cascaded random regression forest method, which is a state-of-the-art hand pose estimation method presented in [33], as hand pose regressor. The final hand pose $\mathbf{\Theta}^T$ is progressively estimated via a series of sequent random forest regressors $\{\mathcal{R}^t\}, t = 1, 2, ..., T$, with pose indexed features, which depend on the estimated pose $\mathbf{\Theta}^{t-1}$ from the previous stage. In order to facilitate understanding, we will give a brief introduction of this method in the rest of this section.

Cascaded random regression forest needs a depth image I and an initial hand pose $\mathbf{\Theta}^0$ as input. In each stage t, it progressively updates the current pose estimation $\mathbf{\Theta}^t$ as

$$\mathbf{\Theta}^t = \mathbf{\Theta}^{t-1} + \mathcal{R}^t(I, \mathbf{\Theta}^{t-1}).$$

The above formula indicates that the hand pose is updated in the 3D camera coordinate system of its corresponding depth image I. When the 3D camera viewpoint is fixed, a specific pose hand model can be used to generate different depth images towards different 3D rigid transformations (corresponding to 3D camera coordinate systems). In the training stage, we compute the hand pose residual $\delta\mathbf{\Theta}$ which is irrelevant to 3D camera coordinate system. Therefore, it is necessary to align the pose $\mathbf{\Theta}$ to the canonical coordinate system which is determined by the canonical hand pose $\mathbf{\Theta}_C$. For a given hand pose $\mathbf{\Theta}$, we compute a 3D rigid transformation $\mathcal{T}_{\mathbf{\Theta}}$ between itself and the canonical hand pose $\mathbf{\Theta}_C$.

In the training stage, the stage regressor $\mathcal{R}^t$ is learnt to approximate the current pose residual $\delta\mathbf{\Theta}_i$, which is the difference between the ground truth pose and the previous pose estimation $\mathbf{\Theta}_i^{t-1}$, over all training samples $i(= 1, 2, ..., N)$. It's worth noting that the features of $\mathcal{R}^t$ depend on the estimated pose $\mathbf{\Theta}_i^{t-1}$ from the previous stage. Similar to previous random forest methods for image processing [1, 5, 6, 8, 34], the pixel difference features, i.e., the difference of two random pixels, are also used. The 3D pose indexed features are constructed as follow:

1. In the canonical coordinate system, randomly select a point pair $(\mathbf{p}_1, \mathbf{p}_2)$ within a 3D sphere whose centre is the centroid of hand point cloud and radius is R, which is related to the size of a real 3D hand model.
2. The point pair $(\mathbf{p}_1, \mathbf{p}_2)$ is transformed to camera coordinate system using the inversed rigid transformation $\overline{\mathcal{T}_{\mathbf{\Theta}}}$.
3. Transformed point pair is projected on depth image to get their corresponding pixels $(\mathbf{u}_1, \mathbf{u}_2)$, and then the pixel difference feature is computed.

The pose indexed feature is written as

$$I(\mathbf{u}_1) - I(\mathbf{u}_2),$$

where $\mathbf{u}_i = CamProj(\overline{\mathcal{T}_{\mathbf{\Theta}}}(\mathbf{p}_i)), i = 1, 2$.

In this paper, we use holistic regression algorithm as our hand pose regressor which regresses the entire hand pose $\mathbf{\Theta}$ at each stage. As for hierarchical regression algorithm of [33], it is completely feasible to directly replace holistic algorithm in our framework, and the accuracy can be further improved. Although we use the holistic regression algorithm in our framework, our approach also performs better than the hierarchical regression algorithm without classification step. The training algorithm for holistic cascaded regression is shown in Algorithm 1.

```
Input: depth image $I_i$, ground truth pose $\mathbf{\Theta}_i$, and initial pose $\mathbf{\Theta}_i^0$ for all
        training samples $i$
Output: regressors $\{\mathcal{R}^t\}_{t=1}^T$
for $t = 1$ to $T$ do
    $\delta\mathbf{\Theta}_i = \mathcal{T}^p_{\mathbf{\Theta}_i^{p,t-1}}(\mathbf{\Theta}_i) - \mathcal{T}^p_{\mathbf{\Theta}_i^{p,t-1}}(\mathbf{\Theta}_i^{t-1})$;
    learn $\mathcal{R}^t$ to approximate $\delta\mathbf{\Theta}_i$;
    $\mathbf{\Theta}_i^t = \mathbf{\Theta}_i^{t-1} + \overline{\mathcal{T}^p_{\mathbf{\Theta}_i^{p,t-1}}}(\mathcal{R}^t(I_i, \mathbf{\Theta}_i^{t-1}))$;
end
```

Algorithm 1: Training algorithm for holistic cascaded hand pose regression. Let $\overline{\mathcal{T}}$ represent the inverse of the rigid transformation $\mathcal{T}$.

3 Experiments

In this section we evaluate the proposed method on the MSRA Hand Pose Dataset [33] that is a real-world depth based dataset. We first describe the implementation details of the classifier and the regressors. Then we introduce the dataset and the evaluation metrics, and quantitatively and qualitatively evaluate the proposed method with the state-of-the art methods.

3.1 Implementation Details

The CNN classifier is implemented in CAFFE [41] framework and those parameters are optimized by using error back-propagation. We choose decay parameter as 0.2 and set batch size to 64, momentum to 0.9 and a weight decay to 0.0005. The learning rate decays over about 10 epochs and starts with 0.005, and the networks are trained for 50 epochs. We choose parameter K as 17 and the classifier is trained on the GPU mode.

The initial hand pose $\mathbf{\Theta}^0$ of hand pose regressor is similar to [33]. Each hand pose regressor consists of 6 cascaded stages and each random regression forest of hand pose regressor consists of 10 trees. Each split node of tree samples 540 random feature point pairs, and we pick one that gives rise to maximum variance reduction over all dimensions of the pose residual. The tree node splits until the node includes less than 10 samples.

In this paper, we propose a heuristic and effective left-right hand pose estimation method only based on the right hand training dataset. At first, a left-right hand binary classifier is used to predict the binary labels of hand. The binary classifier has a same architecture with the hand pose CNN classifier (in section 2.3), except that the output layer is a 2-way softmax unit. The training dataset for binary classifier is constructed by flipping the right hand image of regression training dataset to generate the left hand depth image. In the testing stage, a depth image I is put into the binary classifier to predict left or right hand. If the predicted class is left hand, the original depth image is flipped horizontally. This means that the point clouds of hand are projected symmetrically about YOZ plane, i.e.

$$flip(\mathbf{p}) = flip((x, y, z)) = (-x, y, z), \quad I' = flip(I),$$

where $\mathbf{p}(x, y, z)$ is a point in the original point clouds. Then the pseudo-right depth image I' is put into the right hand pose classifier to get predicted hand pose class k. The k-th cascaded random forest regressor estimates the right hand joints' 3D positions $\mathbf{\Theta}$. Finally, we can flip back to get the left hand pose, that is $\mathbf{\Theta}' = flip(\mathbf{\Theta})$.

3.2 Dataset and Evaluation Metric

There exist some public real-world depth based datasets for hand pose estimation. However, the depth images of the dataset [7] include the forearm that causes terrible initialization which usually produce large errors in pose estimation, and the dataset [34] has restricted range of viewpoints and large annotation errors of ground truth hand poses. The datasets [16, 17, 29] provide too little training data to train meaningful models. The above datasets are not suitable for our task. The MSRA Hand Pose Dataset [33] is a large-scale and challenging real-world benchmark for hand pose estimation. It consists of $76,500$ depth images with accurate ground truth hand poses. The depth images are captured from 9 subjects, and each subject contains 17 gestures. This dataset has larger viewpoint variations (yaw nearly spans the full $[-90, 90]$ range and pitch within $[-10, 90]$ degrees). Thus we select the MSRA Hand Pose Dataset to evaluate our proposed method.

Although the MSRA Hand Pose Dataset performs well for training hand pose regressor in [33], it is not sufficient to train a well-behaved CNN model to classify the complex hand poses that have large variation range of viewpoints. To avoid overfitting and improve the classification accuracy, data enhancement is applied to improve the diversity of dataset. In the MSRA Hand Pose Dataset, each depth image is rotated in $+10°$, $-10°$, $+20°$, $-20°$, $+30°$, $-30°$, $+40°$, $-40°$ to generate 8 depth images. Then the dataset is expanded 8 times from the previous one.

Similar to the previous work [33, 34], there are two accuracy metrics for hand pose estimation. The first one is the averaged Euclidean distance of entire

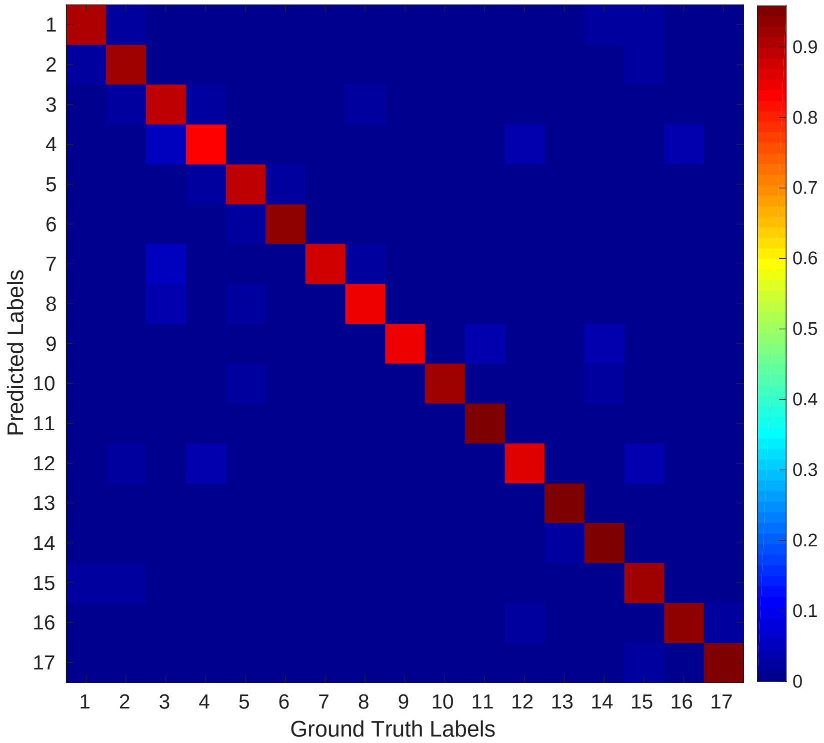


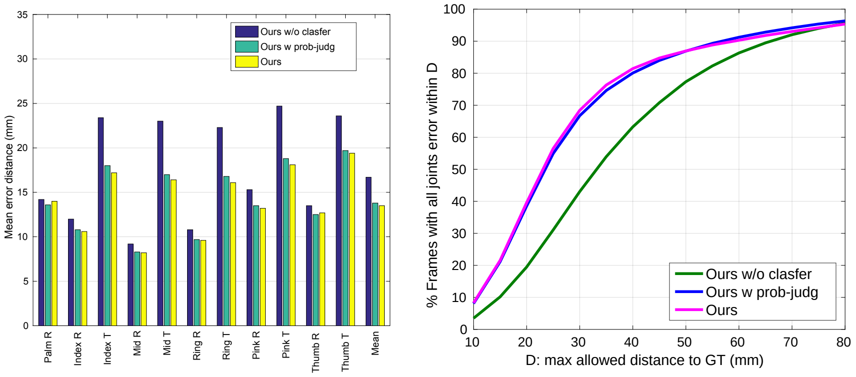
Fig. 4. The confusion matrix of hand pose classifier on the classification training dataset constructed in Sec 2.2. The average accuracy of leave-one-subject-out cross-validation is 91.2%.

predicted joints from the ground truth across all the test samples, that is $M_1 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{21}\|\hat{\mathbf{\Theta}}_{i,j} - \mathbf{\Theta}_{i,j}\|$, where $\hat{\mathbf{\Theta}}$ and $\mathbf{\Theta}$ are respectively the predicted joints and the ground truth joints. The second metric is the success rate, i.e., the percentage of frames where all joints are within a maximum distance threshold ε, that is $M_2 = \frac{1}{N}\sum_{i=1}^{N} I(\max_{1\leq j\leq 21}\|\hat{\mathbf{\Theta}}_{i,j} - \mathbf{\Theta}_{i,j}\| \leq \varepsilon) \times 100\%$, where $I(\cdot)$ is an indicator function. It is obvious that the second metric is more strict than the first one.

3.3 Quantitative Results

Our proposed method is evaluated by leave-one-subject-out cross-validation. For left-right hand classifier and hand pose classifier, the average classification accuracy is respectively 95.0% and 91.2%. The average confusion matrix of hand pose classifiers across all the subjects is shown in Fig. 4.

In order to demonstrate the efficiency of our pipeline, we implement two baselines. The first baseline directly estimates the hand pose without hand pose



(a) (b)

Fig. 5. Comparison our method with the two baselines. a) Mean error distance. b) Success rate with max allowed distance threshold ε. Compared to **ours w/o clasfer**, our method and the second baseline achieve a considerable improvement in the two error metrics.

classifier. We refer this baseline as **ours w/o clasfer**. Our proposed method largely outperforms the first baseline in both mean error distance and success rate metrics. But when the max distance threshold ε is larger than 75 mm, the success rate of our method is slightly less than the first baseline in Fig. 5 (b). The reason is that the hand pose classifier predicts an incorrect class and thus results in a poor estimation of the incorrect regressor.

It is well known that a classifier has a receiver operating characteristic curve. The probability which the classifier predicts a true-positive result is higher, when the probability of top-1 label is higher. In order to resolve that incorrect predicted hand pose class results in poor estimation, we propose the second baseline, a classification-guided regression pipeline with predicted probability judgement, which means a judgement of predicted probability for hand pose classifier is added to decide whether to trust the predicted label. We refer this baseline as **ours w prob-judg**. If predicted probability is greater than a given probability threshold ϵ, we trust the predicted label and use the k-label regressor. Otherwise, we don't trust the predicted label and use the regressor trained on entirety training dataset.

In all the experiments, we choose $\epsilon = 99\%$ as the threshold of predicted probability. The average true-positive ratio for our hand pose classifier is 97.4%. As shown in Fig. 5, the second baseline has a better performance than our proposed method on large distance threshold $\varepsilon \in [50, 80]$. This is because we only trust samples that have high predicted probability and do not coercively execute classification-guided regression method for which has low predicted probability.

We compare our pipeline with state-of-the-art methods [10, 33, 26] on MSRA Hand Pose Dataset. As shown in Fig. 6, our method entirely and substantially performs great better than **Cascaded Hierarchical Regression** [33]. This

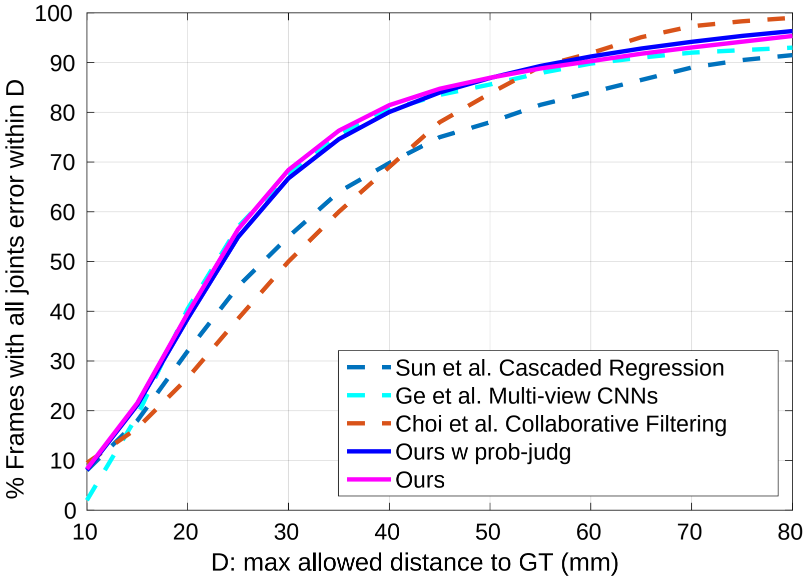


Fig. 6. Quantitative evaluation of hand pose estimation. The figure shows that the proportion of frames where all joints are within max allowed distance threshold ε. We compare our proposed approach to the second baseline and three state-of-the-art methods [33, 10, 26].

is because the hand pose classifier divides the overall complicated and difficult learning task into several relatively easy-to-learn tasks, which are suited to random forest regressor. For **Collaborative Filtering** [10], we achieve superior accuracy predictions on most threshold interval, especially when distance threshold ε is within 50 mm. When the distance threshold ε is greater than 55 mm, the performance of our method get worse than [10]. This is because that our method suffers from poor hand pose estimation caused by the incorrect predicted label. Compared with **Multi-view CNNs** [26], our method performs better in almost all distance threshold interval, especially in the highest and lowest threshold interval. We just use a holistic hand pose regressor to achieve the state-of-the-art performance on accuracy. Furthermore, it will achieve better performance than [26] by replacing regressor with **Multi-view CNNs** in our framework.

We also compare the mean error distance metric over different viewpoint angles of our proposed method and two methods [26, 33] in Fig. 7. Our proposed method has smaller average errors than those of **Cascaded Hierarchical Regression** over all yaw and pitch viewpoint angles, and performs better than **Multi-view CNNs** over most yaw viewpoint angles and partial pitch viewpoint angles.

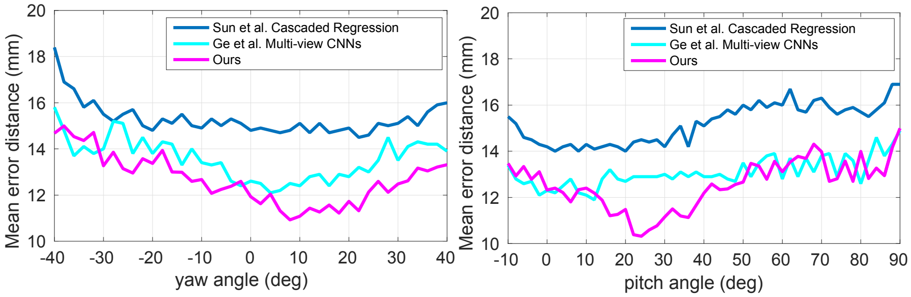


Fig. 7. Quantitative evaluation of hand pose estimation. We compare our approach and two state-of-the-art methods [26, 33] with respect to the mean error distance metric. Left: the mean joint errors distributed over all yaw viewpoint angles. Right: the mean joint errors distributed over all pitch viewpoint angles.

3.4 Qualitative Results

Some qualitative results of the proposed method and the two baselines on several challenging examples are shown in Fig. 8 to further illustrate the superiority of our method over the other two baselines. And more qualitative examples are demonstrated in the accompanying supplementary demo videos.

Our proposed algorithm is tested on Intel i5 3.3GHz with NVIDIA GTX980 GPU running Ubuntu 14.04. The overall hand pose estimation pipeline runs on a single thread on CPU, except that the left-right hand classifier and hand pose classifier are tested on the GPU mode. The left-right hand classifier and hand pose classifier cost 7.1 ms in all, and the hand pose estimation regressor costs 0.7 ms. Therefore, the overall computation time of our method is around 8 ms. Such high performance is sufficient for real-time applications. In terms of method efficiency, the proposed algorithm is faster than most existing methods [5, 7, 8, 10, 17, 26, 29].

4 Conclusions

In this paper, a classification-guided regression learning framework is presented to estimate hand joints' 3D locations from single depth image. In order to simplify the challenging task, a well-trained CNN classifier is applied to identify hand gesture types. Based on the predicted class of classifier, an accurate and efficient cascaded random forest regressor is used to estimate the final hand joints' positions. Our classifier reduces the search space of regression and speeds up hand pose estimation. Experiments demonstrate that the proposed method achieves state-of-the-art performance on challenging dataset. Our proposed method is effective and has strong extensibility that is easy to integrate classifier and regressor into single pipeline. More broadly speaking, any discriminative hand pose

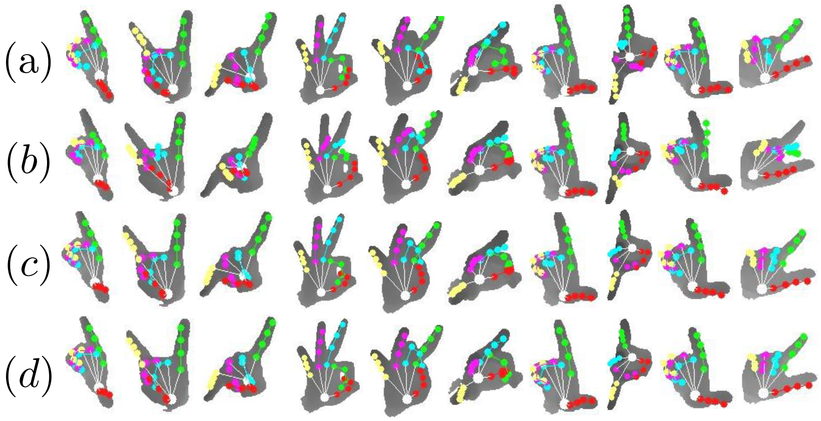


Fig. 8. Qualitative results for dataset in [33] of three approaches. a) The ground truth hand poses. b) Regression without classifier. c) Our classification-guided regression method. d) Classification-guided regression method with predicted probability judgement.

regression model can be used as the regressor module in our framework. Likewise, our classifier unit can be substituted by any kind of efficient classification models.

Acknowledgement. We thank Zishun Liu for his helpful suggestions on this paper. This work was supported by the National Key R&D Program of China (No.2016YFC0800501), NSF of China (Nos.61672481, 61303148), NSF of Anhui Province, China (No.1408085QF119), Specialized Research Fund for the Doctoral Program of Higher Education under contract (No.20133402120002).

References

1. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. In: CVPR. (2011)
2. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: ECCV. (2012)
3. Ye, M., Zhang, Q., Wang, L., Zhu, J., Yang, R., Gall, J.: A survey on human motion analysis from depth data. In: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. Springer (2013) 149–187
4. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC. (2011)
5. Keskin, C., Kıraç, F., Kara, Y.E., Akarun, L.: Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In: ECCV. (2012)
6. Sun, M., Kohli, P., Shotton, J.: Conditional regression forests for human pose estimation. In: CVPR. (2012)

7. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics **33** (2014) 169
8. Tang, D., Yu, T.H., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV. (2013)
9. Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.K., Shotton, J.: Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In: ICCV. (2015)
10. Choi, C., Sinha, A., Choi, J.H., Jang, S., Ramani, K.: A collaborative filtering approach to real-time hand pose estimation. In: ICCV. (2015)
11. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. IJCV (2015) 1–22
12. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. CVIU **108** (2007) 52–73
13. Puwein, J., Ballan, L., Ziegler, R., Pollefeys, M.: Joint camera pose estimation and 3d human pose estimation in a multi-camera setup. In: ACCV. (2014)
14. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Markerless and efficient 26-dof hand pose recovery. In: ACCV. (2010)
15. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-based 3d hand pose estimation from monocular video. PAMI **33** (2011) 1793–1805
16. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: CVPR. (2014)
17. Xu, C., Cheng, L.: Efficient hand pose estimation from a single depth image. In: ICCV. (2013)
18. Li, P., Ling, H., Li, X., Liao, C.: 3d hand pose estimation using randomized decision forest with segmentation index points. In: ICCV. (2015)
19. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., Izadi, S.: Accurate, robust, and flexible real-time hand tracking. In: CHI. (2015)
20. Supancic III, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: methods, data, and challenges. In: ICCV. (2015)
21. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. In: CVWW. (2015)
22. Poier, G., Roditakis, K., Schulter, S., Michel, D., Bischof, H., Argyros, A.A.: Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. In: BMVC. (2015)
23. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: ICCV. (2015)
24. Oberweger, M., Riegler, G., Wohlhart, P., Lepetit, V.: Efficiently Creating 3D Training Data for Fine Hand Pose Estimation. In: CVPR. (2016)
25. Taylor, J., Bordeaux, L., Cashman, T., Corish, B., Keskin, C., Soto, E., Sweeney, D., Valentin, J., Luff, B., Topalian, A., Wood, E., Khamis, S., Kohli, P., Sharp, T., Izadi, S., Banks, R., Fitzgibbon, A., Shotton, J.: Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In: ACM SIGGRAPH. (2016)
26. 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: CVPR. (2016)
27. Mohr, D., Zachmann, G.: A survey of vision-based markerless hand tracking approaches. CVIU (2013)

28. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ACM Transactions on Graphics **28** (2009)
29. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using rgb and depth data. In: ICCV. (2013)
30. Sridhar, S., Rhodin, H., Seidel, H.P., Oulasvirta, A., Theobalt, C.: Real-time hand tracking using a sum of anisotropic gaussians model. In: 3DV. (2014)
31. Sridhar, S., Mueller, F., Oulasvirta, A., Theobalt, C.: Fast and robust hand tracking using detection-guided optimization. In: CVPR. (2015)
32. Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., Pauly, M.: Robust articulated-icp for real-time hand tracking. Computer Graphics Forum **34** (2015) 101–114
33. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR. (2015)
34. Tang, D., Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: CVPR. (2014)
35. Neverova, N., Wolf, C., Nebout, F., Taylor, G.: Hand pose estimation through weakly-supervised learning of a rich intermediate representation. Computer Science (2015)
36. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: 3DIM. (2001)
37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
38. Jain, A., Tompson, J., Andriluka, M., Taylor, G.W., Bregler, C.: Learning human pose estimation features with convolutional networks. Computer Science (2013)
39. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: AISTATS. (2011)
40. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
41. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)