Robust Hand Pose Estimation during the Interaction with an Unknown Object - Supplementary Material

Chiho Choi Sang Ho Yoon Chin-Ning Chen Karthik Ramani
Purdue University
West Lafayette, IN 47907, USA
{chihochoi, yoon87, chen2300, ramani}@purdue.edu

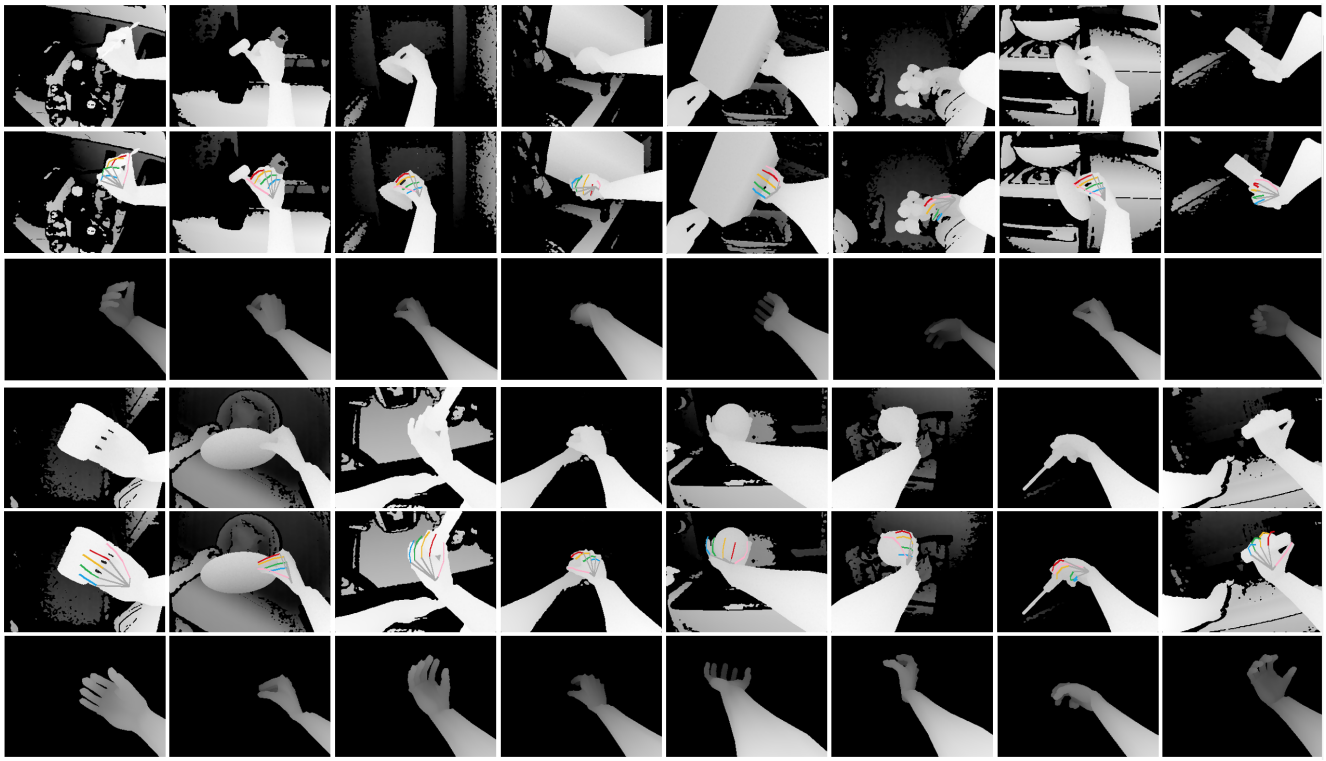

Figure 1: Additional qualitative evaluations using a synthetic dataset. The first (fourth) row shows the input depth image, and estimated hand skeletons are presented in the second (fifth) row. The third (sixth) row shows the reconstructed hand mesh model from skeleton estimation.

This document serves as supplementary materials to our paper *Robust Hand Pose Estimation during the Interaction with an Unknown Object*. We present additional qualitative evaluations, details of our network design, and the system specifications.

1. Qualitative analysis

We conduct additional qualitative evaluations of our approach using the synthetic dataset. The first and fourth row of Figure 1 shows the input depth images, and the corresponding hand estimates are presented in the second and fifth row. Then we reconstruct hand models based on our estimation in the third and sixth row.

2. Details of Network Design

2.1. Architecture of localization ConvNet

We visualize neuron activations of our localization ConvNet in Figure 2 to validate our network structure. Our network is mainly comprised of six convolutional layers.

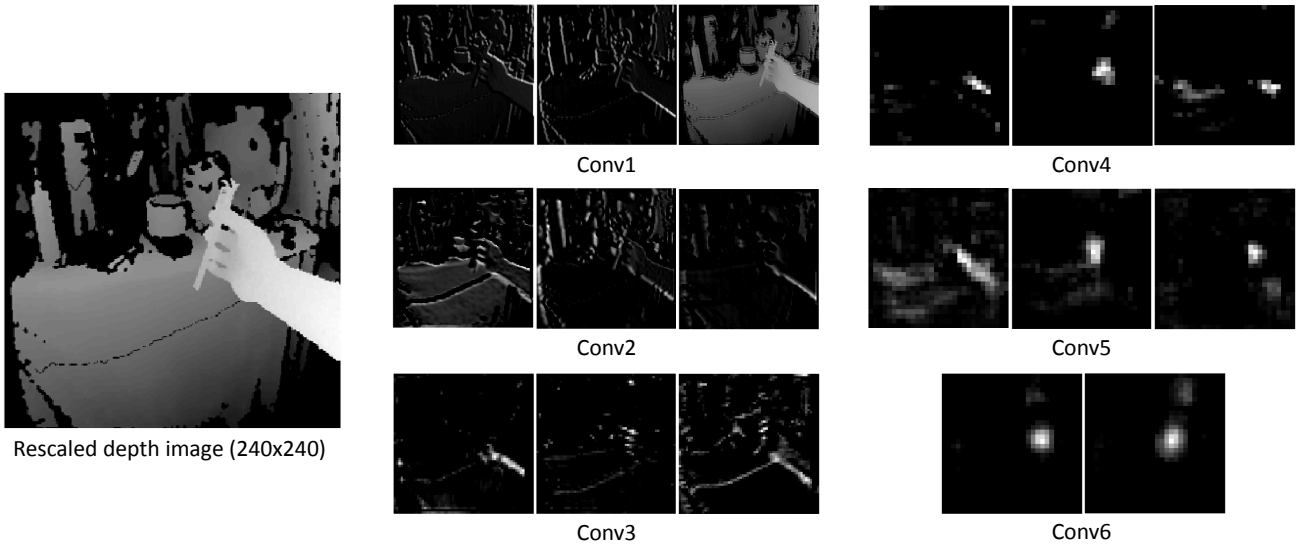

Figure 2: Neuron activations are presented for each convolutional layer. We randomly select three feature maps and resized for only visualization purpose. The outputs of Conv6 are two-channel 30×30 heatmaps that represents a confidence of the hand/object center position.

Through Conv1 to Conv6, neurons are activated nearby edges of foreground (*i.e*. the hand and object). As shown in Figure 2, the network shows higher confidences to the center of the hand and object (see Conv4-conv5). Then, it outputs two clear heamaps corresponding to their centers after Conv6. Note that the left and right image of Conv6 shows the center position of the hand and the object, respectively.

2.2. Architectures for grasp classification

To find a best ConvNet configuration for classifying global orientations and grasp types, we evaluate different ConvNet architectures as outlined in Table 1. Our grasp classification network is named as ConvNet A, and its structure consists of five convolutional layers followed by a max pooling and nonlinear (ReLU) layer and two fully connected (FC) layers with a nonlinear layer at the end.

We test three distinctive configurations (ConvNet B-D) which differ in the order of feature concatenation and the number of FC layers from ConvNet A. Figure 3 shows the effect of the proposed variations in terms of loss and accuracy. ConvNet B concatenates the feature maps after the fifth convolutional layer. As a result, both the hand-oriented network and object-oriented network independently process the data by convolving 128 4×4 kernels with the output feature map of the fourth convolutional layer. However, this step makes our decision function less discriminative. In ConvNet C, we expected better performance in grasp classification with additional FC-32 layer for the grasp type decision network. However, we observed slower loss convergence and lower accuracy comparing to ConvNet A (Figure 3c and 3d). Still, ConvNet C exhibits higher performance than ConvNet B and D because the convolutional layer after concatenation locally extracts more better representations. For ConvNet D, although we put an additional FC-64 layer for more expressive feature extraction in the decision network, the performance drops due to reduced dimensionality of learned features.

3. Implementation Specifications

Data processing As an output of the localization network, we get the centroid of the hand $d_m^h = \{u_h, v_h, d_h\}$ and the object $d_m^o = \{u_o, v_o, d_o\}$. Then we first convert d_m^h and d_m^o to $\{x_h, y_h, d_h\}$ and $\{x_o, y_o, d_h\}$ in world coordinates:

$$\begin{aligned} x &= \frac{(u - \frac{\text{width}}{2})}{f} \times d \\ y &= \frac{(v - \frac{\text{height}}{2})}{f} \times -d, \end{aligned} \tag{1}$$

where f denotes the focal length (241.42 for Intel's RealSense F200). Then, we create a bounding box around the center point with 120 mm offset in world coordinates and transform these 3D points into image coordinates. Finally, we crop and resize D_m to obtain 64×64 depth maps D_m^h and D_m^o. We again carry out depth normalization so that the data reproduction network can take as input a pair of depth images D_i^h and D_i^o.

Training We simultaneously trained neural networks with an aid of GPUs (NVIDIA's Geforce GTX 1070) using the

ConvNet Architecture

ConvNet A		ConvNet B		ConvNet C		ConvNet D	
Hand	Object	Hand	Object	Hand	Object	Hand	Object
conv5-16	conv5-16	conv5-16	conv5-16	conv5-16	conv5-16	conv5-16	conv5-16
conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32
conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32	conv5-32
conv5-64	conv5-64	conv5-64	conv5-64	conv5-64	conv5-64	conv5-64	conv5-64
		conv4-128	conv4-128			conv4-128	conv4-128
Concatenation							
conv4-128	conv4-128			conv4-128	conv4-128	FC-64	
FC-64	FC-64	FC-64	FC-32	FC-64	FC-32	FC-32	FC-32
FC-48	FC-33	FC-48	FC-33	FC-48	FC-32	FC-48	FC-33
					FC-33		
Softmax							
Orientation	Grasp	Orientation	Grasp	Orientation	Grasp	Orientation	Grasp

Table 1: Details of ConvNet architectures. We present four different configurations which differ in the order of feature concatenation (ConvNet B and D) and the number of FC layers (ConvNet C and D) from our grasp classification network (ConvNet A). Hand: hand-oriented network, Object: object-oriented network, Orientation: orientation decision network, Grasp: grasp decision network.

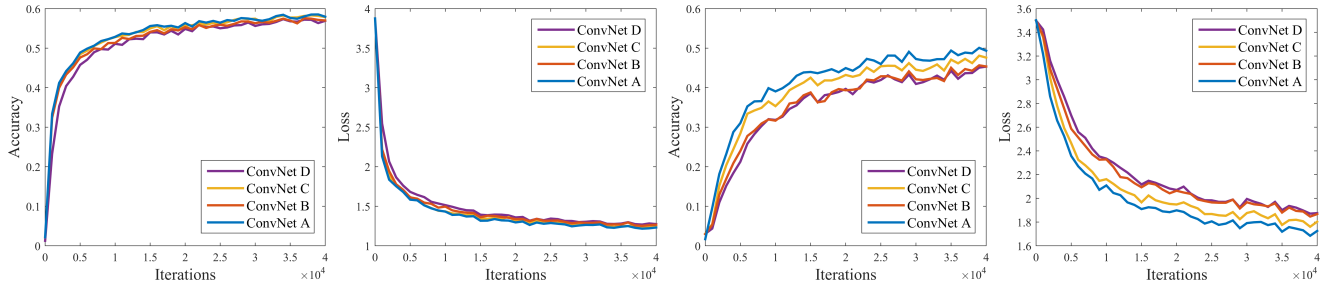

Figure 3: Loss and accuracy while training the network models. (a) and (b): Accuracy and loss of the orientation decision network. (c) and (d): Accuracy and loss of the grasp decision network.

Caffe framework [1]. Training of localization ConvNet took about 3 hours by setting the learning rate to 10^{-6} and the number of iterations to 50K. In addition, our data reproduction network was converged after 20 epochs (25K iterations) by setting the learning rate 10^{-4}, and it took about 40 mins for training the parameters. Finally, training of our grasp classification network took about another 40 mins and it converged after 40K iterations with the learning rate 10^{-2}. At runtime, the computation time for each frame is split as 3 ms for processing input data (*i.e.* depth normalization, resizing, rescaling, and cropping the depth maps/images), 4 ms for localization, 0.5 ms for data synthesis (for both the hand and object depth images), 0.2 ms to classify orientation and grasp type, and 4 ms to find a set of nearest neighbors and regress the pose angle parameters.

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 2