1. Network Architecture of Hand Pose Regression Network

We present the detailed network architecture of the hand pose regression network in Figure 1, which is based on the architecture of hierarchical PointNet proposed in [7]. As can be seen, the hand pose regression network has three set abstraction levels. At the $l$-th level ($l = 1, 2$), $N_l$ points are sampled using iterative farthest point sampling and $k$ nearest neighboring points of each sampled point are grouped as a local region. Following [7], we search nearest points that are within a radius to the query point in the Euclidean space. After the process of sampling and grouping, we get $N_l$ local regions. Each local region containing $k$ points is fed into a basic PointNet [6] to extract a $C_l$-dim local feature. The output data size of this level is $N_l \times C_l$. The $C_l$-dim local features together with the $d$-dim coordinates of the $N_l$ sampled points are input to the next level. At the $3$rd level, all the input points of this level are fed into a basic PointNet to extract a $C_3$-dim global feature. The global feature is mapped to an $F$-dim output vector through a multi-layer perceptron (MLP) network.

Each MLP network in Figure 1 is composed of several fully connected layers. All fully connected layers are followed by batch normalization and ReLU except for the last layer of the last MLP network. We do not use dropout layer in our implementation.

2. Additional Experiments

2.1. Impact of Surface Normal

We evaluate the impact of surface normal on estimation errors. As shown in Figure 2, using surface normal as the input feature will improve the performance a little bit. We also experiment with approximating surface normal using different numbers ($k$) of nearest neighboring points. As can be seen, our method is not sensitive to the procedure of surface normal approximation.

Figure 2: Impact of surface normal on estimation errors.

2.2. Impact of Fingertip Refinement

As shown in Table 1, the improvement of fingertip refinement on overall mean error is small (0.1mm∼0.3mm), since the five fingertips only take a small part of all the joints. Moreover, without the fingertip refinement, our method can still achieve state-of-the-art performance on all the three datasets.

In addition, we use the estimation results of other methods as initial estimations for fingertip refinement. As shown
Table 1: The impact of fingertip refinement on NYU [10], MSRA [8] and ICVL [9] datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean Error on All Joints</th>
<th>Mean Error on 5 Fingertips</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Refine.</td>
<td>w/o Refine.</td>
<td>with Refine.</td>
</tr>
<tr>
<td>NYU [10]</td>
<td>10.8mm</td>
<td>10.9mm</td>
</tr>
<tr>
<td>MSRA [8]</td>
<td>8.6mm</td>
<td>8.5mm</td>
</tr>
<tr>
<td>ICVL [9]</td>
<td>7.2mm</td>
<td>6.9mm</td>
</tr>
</tbody>
</table>

Table 2: The impact of our fingertip refinement using estimation results of other methods as initial estimations (NYU [10] dataset).

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Error on All Joints</th>
<th>Mean Error on 5 Fingertips</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Refine.</td>
<td>w/o Refine.</td>
<td>with Refine.</td>
</tr>
<tr>
<td>3D CNN [2]</td>
<td>14.1mm</td>
<td>13.5mm</td>
</tr>
<tr>
<td>DeepPrior++ [4]</td>
<td>12.3mm</td>
<td>12.3mm</td>
</tr>
<tr>
<td>Pose-REN [1]</td>
<td>11.8mm</td>
<td>11.6mm</td>
</tr>
</tbody>
</table>

in Table 2, our fingertip refinement method can also improve fingertip estimations of other methods evidently, and slightly improve the mean error on all joints.

2.3. Additional Comparison with State-of-the-arts

In order to make a fair comparison with the spatial attention network-based hierarchical hybrid method proposed in [11], we evaluate the proportion of hand joints within different error thresholds on NYU hand pose dataset [10]. Following the experimental setting in [11], we calculate this evaluation metric on 11 hand joints (removing the palm joints except the root joint of thumb in the set of original 14 hand joints on NYU dataset).

Experimental results are shown in Figure 3. Apart from the spatial attention network-based method [11], we also compare our method with DeepPrior [5], 3D CNN [2], REN [3] and DeepPrior++ [4] methods by using this evaluation metric. As can be seen, our method is superior to all of these methods over all the error thresholds. For example, when the error threshold is 10mm, the proportion of joints within this error threshold of our method is about 30% more than that of the spatial attention network-based method [11], 20% more than that of the 3D CNN-based method [2] and 10% more than that of the DeepPrior++ method [4].

3. More Details on Runtime and Model Size

During the testing stage, the depth image is first converted to a set of 3D points, from which 1024 points are randomly sampled. Since we use the hierarchical PointNet [7], the farthest point sampling (FPS) algorithm is applied to evenly sample 512 and 256 points from the 1024 points for the first two set abstraction levels. This step including random sampling and FPS takes 1.7ms for one frame in average. The next step is surface normal approximation for the 1024 points, which takes 6.5ms for one frame in average.

These two steps are implemented using C++ with the Point Cloud Library (PCL). The 3D points attached with surface normals are then fed into the hand pose regression network to estimate 3D hand joint locations, which takes 9.2ms for one frame in average on a single Nvidia GTX1080 GPU with a batch size 1. For fingertip refinement, neighboring points search and surface normal approximation take 2.8ms for one frame in average, and the fingertip refinement network forward propagation takes 0.3ms for one frame in average on a single Nvidia GTX1080 GPU. The batch size is equal to the number of straightened fingers for each frame. Therefore, the total runtime of our method is 20.5ms for one test frame in average, and the average frame rate is about 48fps.
The number of parameters in the hand pose regression network is about $2.3 \times 10^6$ (2.3M). These parameters are stored in 32 bit float and the size of parameters is 9.2MB. The number of parameters in the fingertip refinement network is about $1.4 \times 10^5$ (142K). These parameters are stored in 64 bit double and the size of parameters is 1.1MB. In total, there are about $2.5 \times 10^6$ (2.5M) parameters in these two networks and the total size of network parameters is 10.3MB.

References


