Light Weight Computation and Compact Network for 3D Hand Pose Estimation from Depth

Li Li Liu
Shanghai Institute of Microsystem and Information Technology
liulili@mail.sim.ac.cn

Abstract

3D hand pose estimation methods from depth have made significant progress recently. However, the heavy computation of current models are not practical for real world applications on mobile devices. Therefore, We propose a light-weight network to estimate the hand pose from depth. The network is consisted of linear attention modules, a compact feature representation module and light-weight convolution computation modules. In comparison with the state-of-the-art methods, the proposed network achieves 0.27G flops which is about 2% of the state-of-the-art method. Besides, the reconstruction error increases about 6% on ICVL Dataset and 8% on NYU Dataset, respectively.

1. Introduction

Hands are important for allowing humans to interact with the world around them. There are many applications in human-computer interaction (HCI), augmented reality (AR), virtual reality (VR), and gesture recognition [13] [12] that require accurate hand pose estimation. Significant progress has been made in depth-based 3D hand pose estimation [7] and hand segmentation [16][9] as commodity depth cameras become more accurate and affordable and more widely available. Existing methods are limited in their reduction of inference time and memory consumption, especially for resource-constrained devices. Therefore, we propose a network for 3D hand pose estimation from depth.

2. Related Work

2.1. Linear attention

Improving the efficiency of MHA(multi-headed self-attention) in transformers is an active area of research. The first line of research introduces locality to address the computational bottleneck in MHA [2]. To improve the efficiency of MHA, the second line of research uses similarity measures to group tokens[8] [18]. The third line of research improves the efficiency of MHA via low-rank approximation[17] [3]. Even though these methods speed-up the self-attention operation significantly, they still use expensive operations for computing attention, which may hinder the deployment of these models on resource-constrained devices. However, we propose a light-weight network for the estimation of hand pose from depth.

2.2. Mobile convolution network

A lot of studies are made for the development of the efficient networks. ShuffleNet[21] utilizes group convolution and channel shuffle operations to further reduce the MAdds. CondenseNet [6] learns group convolutions at the training stage to keep useful dense connections between layers for feature re-use. ShiftNet [19]proposes the shift operation interleaved with point-wise convolutions to replace expensive spatial convolutions. GhostNet applies a series of linear transformations to generate many ghost feature maps. [5]. Therefore, we apply similar light-weight modules to reduce the flops of the network for the estimation of hand pose.

3. Methodology

The task of 3D hand pose estimation is defined as follows: given an input depth image $D_I \in \mathbb{R}^{H \times W}$, the task is to estimate the 3D location of a set of pre-defined hand joints $P \in \mathbb{R}^{J \times 3}$ in the camera coordinate system. As illustrated in Figure 1, the proposed network consists of two stages. In the first stage, the input depth image is run through the encoder network *f*. The component1(com1), computes a per-joint attention map. The component2(com2), also computes a per-joint attention map. The fused attention map is then used as guidance for pooling features from the depth feature map computed by the component3(com3). Then, the linear self-attention is applied for the grid computation as the prior to the proposed network. Finally, a weight-sharing linear layer is used to estimate the joint depth values from the feature vectors computed for each joint.

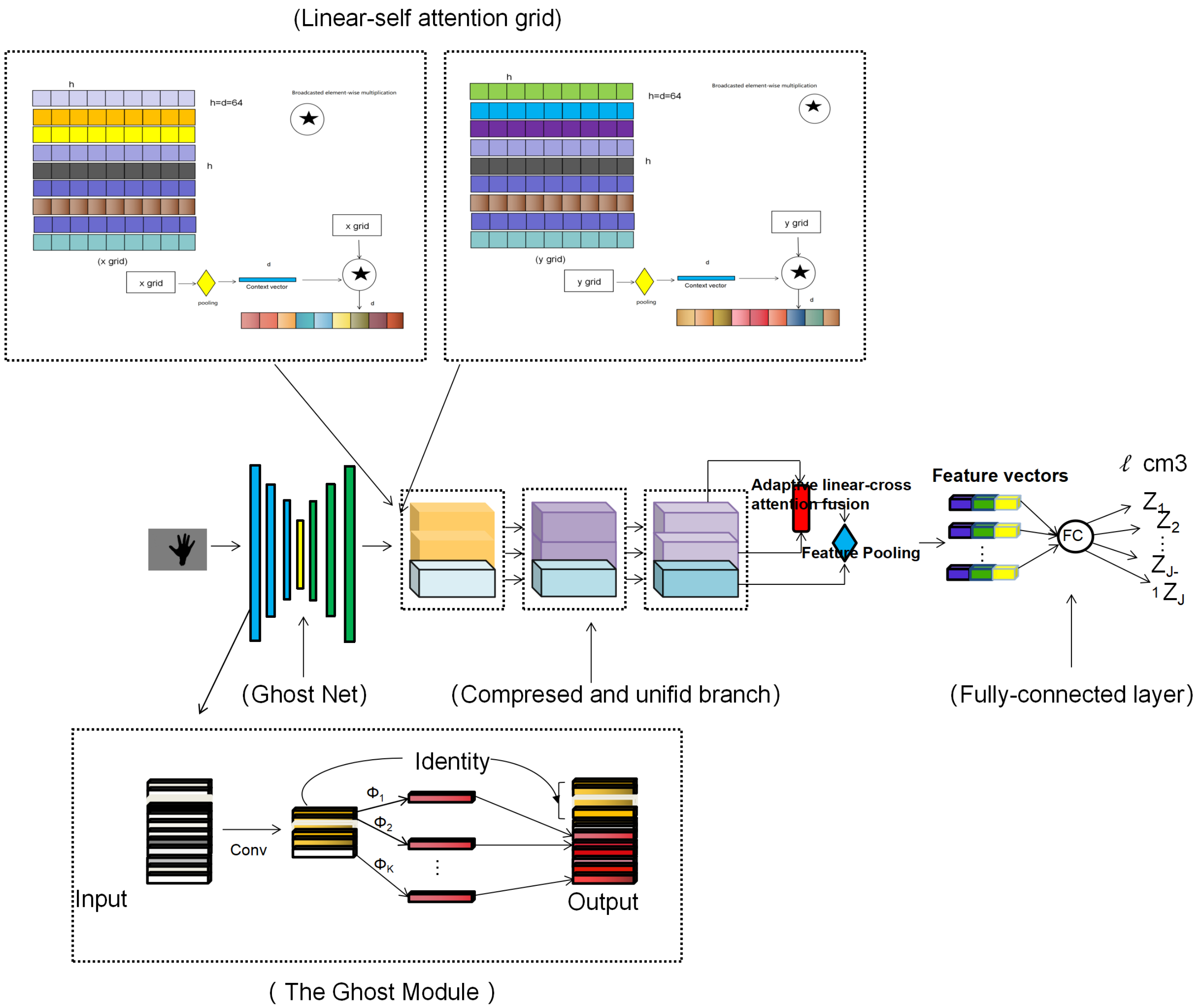

Figure 1. An overview of the proposed network. the proposed network consists of two stages. The first stage is the encoder, the second stage consists of Compressed and unified branch and attention fusion and Linear-self attention grid computation.

3.1. Ghost module

The encoder is defined as a non-linear mapping from the input depth image to the output feature volume f : $\mathbb{R}^{H\times W} \rightarrow \mathbb{R}^{c\times h\times w}$, where h, w and c denote the height, width and the number of the channels of the output feature volume respectively. We use Lighter Stacked Hourglass network[4].

m intrinsic feature maps $\boldsymbol{Y}^{'} \in \mathbb{R}^{h^{'}\times w^{'}\times m}$ are generated using a primary convolution:

$$\boldsymbol{Y}^{'} = \boldsymbol{X} \times f^{'} \tag{1}$$

where $f^{'} \in \mathbb{R}^{c\times k\times k\times m}$ is the utilized filters, $m \leq$ n and the bias term is omitted for simplicity. To further obtain the desired n feature maps, we adopt to apply a series of cheap linear operations on each intrinsic feature in $\boldsymbol{Y}^{'}$ to generate s ghost features according to the following function:

$$y_{ij} = \Phi_{i,j}(y_i^{'}), \qquad \forall i = 1,\cdots,m, \quad j = 1,\cdots,s, \tag{2}$$

where $y_i^{'}$ is the i-th intrinsic feature map in $\boldsymbol{Y}^{'}$, $\Phi_{i,j}$in the above function is the j-th (except the last one) linear operation for generating the j-th ghost feature map y_{ij}.

3.2. Compressed and unified branch

This branch takes as input the output feature volume from the Ghost module and computes attention map $Att \in \mathbb{R}^{(2\times J+D)\times h\times w}$. We use $Att^{j}_{com1} \in \mathbb{R}^{J\times h\times w}$ to refer to the attention map corresponding to the j^{th} joint. Att^{j}_{com2} $\in \mathbb{R}^{J\times h\times w}$ denotes the attention map corresponding to the j^{th} joint. $Att^{j}_{com3} \in \mathbb{R}^{D\times h\times w}$ denotes a dense depth feature map, D represents the depth feature vector dimension.

The attention map Att^{j}_{com1} is first normalized by a spatial softmax layer to obtain the corresponding heatmap $C^{2D}_{j} = \sigma(Att^{j}_{com1})$ as follows:

$$C^{2D}_{j}(x,y) = \frac{exp(Att^{j}_{com1}(x,y))}{\sum\limits_{k_i,l_i\in\Omega} exp(Att^{j}_{com1}(k_i,l_i))} \tag{3}$$

In the above, σ denotes the spatial softmax layer. The heatmap C^{2D}_{j} represents the likelihood of the j^{th} joint occurring at each pixel location. Ω represents the spacial domain of the attention map Att^{j}_{com1}. The 2D location of the j^{th} joint is computed through an integration operation similar to, as follows:

$$(\overline{K}^{j}, \overline{L}^{j}) = \sum_{k_i}\sum_{l_i}(k_i,l_i)C^{2D}_{j}(k_i,l_i) \tag{4}$$

The attention map Att^{j}_{com2} is first normalized by a spatial softmax layer to obtain the corresponding heatmap $E^{2D}_{j} = \sigma(Att^{j}_{com2})$ as follows:

$$E^{2D}_{j}(x,y) = \frac{exp(Att^{j}_{com2}(x,y))}{\sum\limits_{k_i,l_i\in\Omega} exp(Att^{j}_{com2}(k_i,l_i))} \tag{5}$$

The depth value for the j^{th} joint is estimated from the feature vector obtained by pooling features from the pixels that contain the most relevant information about its depth, which is guided by $Att^{j}_{wise-fused}$ as follows:

$$\begin{aligned} H_j = Att^{j}_{wise-fused} \circ Att^{j}_{com3} = \\ \sum_{x}\sum_{y} Att^{j}_{wise-fused}(x,y)\Gamma(x,y) \end{aligned} \tag{6}$$

The depth value for the j^{th} joint, denoted by $\overline{Z}_j$, is then estimated using a single linear layer as follows:

$$\overline{Z}_j = H_j\mathbf{W} + \mathbf{b} \tag{7}$$

The depth value estimation for the joints is supervised by the following loss term:

$$\ell_{cm3} = \frac{1}{J}\sum_{j=1}^{J}|\overline{Z}^{j} - Z^{j}| \tag{8}$$

Where Z_j refers to the ground-truth depth value for the j^{th} joint.

3.3. Attention fusion

The two attention maps $Att_{com1}$and Att_{com2} are complementary in the sense. These two attention maps are fused as follows:

$$Att^{j}_{wise-fused} = \sigma(\beta_j Att^{j}_{com1} + (1-\beta_j)Att^{j}_{com2}) \tag{9}$$

3.4. Linear-self attention and grid computation

As shown in Figure 1, we adopt a separable self-attention method with linear complexity. It uses element-wise operations for computing self-attention, making it a good choice for resource-constrained devices.

The context scores $\boldsymbol{c_s}$ are used to compute a context vector $\boldsymbol{c_v}$. Specifically, the input x is linearly projected to a d-dimensional space using key branch K with weights $\boldsymbol{W_k} \in \mathbb{R}^{d\times d}$ to produce an output $\boldsymbol{X_K} \in \mathbb{R}^{d\times d}$. The context vector $\boldsymbol{c_v} \in \mathbb{R}\ ^{d}$ is then computed as a weighted sum of $\boldsymbol{X_K}$ as:

$$\boldsymbol{c}_v = \sum_{1}^{k}\boldsymbol{c_s}(i)\boldsymbol{X_K}(i) \tag{10}$$

The context vector $\boldsymbol{c_v}$ is analogous to the attention matrix a in (Eq.10) in a sense that it also encodes the information from all tokens in the input $\boldsymbol{x}$, but is cheap to compute.

Mathematically, separable self-attention can be defined as:

$$y = \left(\underbrace{\sum \left(\overbrace{\sigma(xW_I)}^{c_s\in\mathbb{R}^k} * xW_K \right)}_{c_v\in\mathbb{R}^d} * ReLU(xW_v) \right) W_o \tag{11}$$

where $*$ and $\sum$ are broadcastable element-wise multiplication and summation operations, respectively.

3.5. Loss

$(\overline{K}^{j}, \overline{L}^{j})$ represents the estimated coordinates of the j^{th} joint in the depth image space. The mean L1distance defined as:

$$\ell_{com1} = \frac{1}{2J}\sum_{j=1}^{J}|\overline{K}^{j} - K^{j}|+|\overline{L}^{j} - L^{j}| \tag{12}$$

In the above, (K^{j}, L^{j}) represents the ground-truth 2D location of the j^{th}joint. The proposed model is end-to-end differentiable and is trained by minimizing the loss function, which is formulated as:

$$\mathcal{L} = \ell_{com1} + \ell_{cm3} \tag{13}$$

4. Experiments

4.1. Implementation Details

The pre-processing method for preparing the input depth image includes first cropping the hand area from a depth image similar to[11], and then resizing it to a fixed size of 128 ×128.

4.2. Datasets and Evaluation Metrics

ICVL Dataset. The ICVL dataset [15] provides $22K$ and $1.6K$ depth frames for training and testing, respectively.

NYU Dataset. The NYU dataset [16] is captured from three different views with Microsoft Kinect sensor.

Evaluation metrics. We use the commonly used metrics for the evaluation of 3D hand pose estimation: the mean distance error (in mm).

4.3. Ablation Study

Impact of using Ghost module. We study the impact of Ghost bottleneck in the Hourglass model.

Effectiveness of Different Approaches for channel-wise computation. We study the effectiveness of channel-wise computation in our model.

Impact of the linear-self attention module. A linear attention is applied for the grid computation.

In the above, as can be seen in Table 3.

Table 1,2 summarizes the performance based on the mean distance error on the two datasets. As shown in Figure 2, we visualized some of the estimation results.

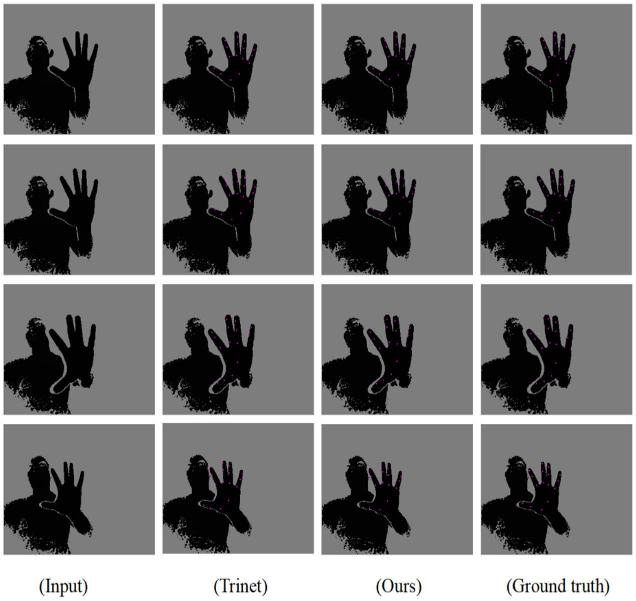

Figure 2. Input is the input depth image. Trinet is the result of TRIHORN-NET module. ours is the result of the proposed network. The ground truth is the ICVL Dataset proved.

Methods	Error(mm)	FLOPs(G)	params(M)
A2J [20]	6.46	6.21	82.01
V2V-PoseNet [10]	6.28	38.453	3.41
HandFoldingNet [1]	5.95	35.788	1.283
trinet[14]	5.73	13.47	7.81
ours	6.077	0.27	0.18

Table 1. Comparison with the state-of-the-art method on ICVL

Methods	Error(mm)	FLOPs(G)	params(M)
A2J [20]	8.61	6.21	82.01
V2V-PoseNet [10]	8.42	38.453	3.41
HandFoldingNet [1]	8.58	35.788	1.283
trinet[14]	7.68	13.47	7.81
ours	8.32	0.27	0.18

Table 2. Comparison with the state-of-the-art method on NYU

Approaches	Error(mm)	FLOPs(G)
Compressed and unifid branch	6.049	0.398
Ghost module	5.683	11.097
linear-self attention grid compution	5.835	13.47
ours	6.077	0.27

Table 3. Comparison of different approaches on ICVL

5. Conclusion

In this paper, we proposed a light-weight network for 3D hand pose estimation from a single depth image. In comparison with the state-of-the-art models, the proposed network achieves the Flops 0.27G which is 2 % of the state-of-the-art models. Besides, the reconstruction error increases about 6 % on ICVL Dataset and 8% on NYU Dataset, respectively.

References

[1] Wencan Cheng, Jae Hyun Park, and Jong Hwan Ko. Handfoldingnet: A 3d hand pose estimation network using multiscale-feature guided folding of a 2d hand skeleton, 2021.

[2] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

[3] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2022.

[4] Ahmed Elhagry, Mohamed Saeed, and Musie Araia. Lighter stacked hourglass human pose estimation, 2021.

[5] K. Han, Y. Wang, Q. Tian, J. Guo, and C. Xu. Ghostnet: More features from cheap operations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. *CoRR*, abs/1711.09224, 2017.

[7] Umar Iqbal, Pavlo Molchanov, Thomas M. Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. *CoRR*, abs/1804.09534, 2018.

[8] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer, 2020.

[9] Sri Raghu Malireddi, Franziska Mueller, Markus Oberweger, Abhishake Kumar Bojja, Vincent Lepetit, Christian Theobalt, and Andrea Tagliasacchi. Handseg: A dataset for hand segmentation from depth images. *CoRR*, abs/1711.05944, 2017.

[10] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map, 2018.

[11] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. *CoRR*, abs/1708.08325, 2017.

[12] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794, 2021.

[13] Razieh Rastgoo, Kourosh Kiani, Sergio Escalera, and Mohammad Sabokrou. Sign language production: A review. *CoRR*, abs/2103.15910, 2021.

[14] Mohammad Rezaei, Razieh Rastgoo, and Vassilis Athitsos. Trihorn-net: A model for accurate depth-based 3d hand pose estimation, 2022.

[15] Tang, DH, Chang, HJ, Tejani, A, Kim, and TK. Latent regression forest: Structured estimation of 3d articulated hand posture. *Proc Cvpr IEEE*, 2014.

[16] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. In *ACM*, pages 1–10, 2014.

[17] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.

[18] S. Wang, L. Zhou, Z. Gan, Y. C. Chen, Y. Fang, S. Sun, Y. Cheng, and J. Liu. Cluster-former: Clustering-based sparse transformer for question answering. In *Meeting of the Association for Computational Linguistics*, 2021.

[19] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter H. Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. *CoRR*, abs/1711.08141, 2017.

[20] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image, 2019.

[21] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.