Peeking into occluded joints: A novel framework for crowd pose estimation

Lingteng Qiu[1,2†], Xuanye Zhang[1†], Yanran Li[3], Guanbin Li[4], Xiaojun Wu[2], Zixiang Xiong[5], Xiaoguang Han[1*], and Shuguang Cui[1]

[1]Shenzhen Research Institute of Big Data, The Chinese University of Hongkong, Shenzhen[2]Harbin Institute of Technology(Shenzhen)[3]Bournemouth University[4]Sun Yat-sen University[5]Texas A& M University

We provide details of our algorithms and more visualization results in this supplementary material. Two videos are attached for demonstrating the visual comparisons between our OPEC-Net and AlphaPose+ [2] in crowd and couple scenarios respectively. Our estimation displays as green skeletons and the red ones are from AlphaPose+. It is obvious to observe from the videos that our estimation results are more accurate than AlphaPose+, especially for occlusion cases. For the couple scenario, the comparison between OPEC-Net and OPEC-CG (with CoupleGraph extension) is also shown, where OPEC-Net is with red color while OPEC-CG is in green. As seen, with the help of CoupleGraph, the approach is more robust to severe occlusions.

1 Illustration of the CoupleGraph

CoupleGraph is proposed to capture the human interaction information. In the main paper, we gave out the formulation of the CoupleGraph. Here, we illustrate the structure of the couple graph in the following:

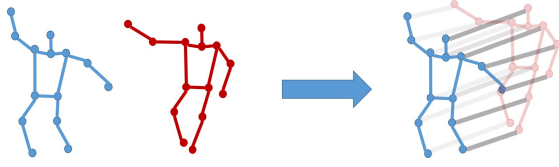

Fig. 1. The couple graph. The blue and red skeletons represent two different individuals. Other than in-skeleton bone edges, the graph also links the two corresponding joints between the two instances.

We extend the single human graph into a CoupleGraph that captures more human interactions and this is achieved by connecting the corresponding joints

†The first two authors should be considered as joint first authors.
* Corresponding Author, hanxiaoguang@cuhk.edu.cn

to capture human interaction information. The couple graph can be denoted as $G' = (V', E')$. The number of joints for a single person is N so that there is $2N$ joints in total in the couple graph. It can be formulated as $V' = \{v_i \mid i = 1, 2, ..., 2N\}$. There are two types of edges in E': the edges representing the human skeleton E_s and the edges connecting the two humans E_c. The human skeleton edges are noted as $E_s = \{v_i v_j \mid \text{if } i \text{ and } j \text{ are connected in the human body}\}$. The human interaction edges can be written as $E_c = \{v_i v_{i+N}\}$, where the v_i and v_{i+N} are corresponding to the same components of the two human skeletons. The CoupleGraph module is appended after the OPEC-Graph module to enhance the performance of estimation. Each pair of people is processed by CoupleGraph.

2 Heatmap representation to Coordinate representation.

First of all, we generate the initial pose for the GCN network from the heatmaps of the first two stages. An important factor to consider in obtaining the initial pose is that the translation from heatmap to coordinate representation needs to be differential for the end to end training purpose, so the initial pose cannot be grasped directly from the heatmap by searching max values as P. Finally, we found out that a coordinate initial pose $\hat{J}_i$ can be generated from the $Heatmap$ and estimated by an *integral regression* method [3]. Specifically, the heatmap is propagated into a Softmax layer which normalizes the values into likelihood values $[0, 1]$. After that, an integral operation is applied on the likelihood map to sum up the values and estimate joints positions.

$$\hat{J}_i^k = \int_{p \in A} p \cdot H_k(p), \tag{1}$$

where $\hat{J}_i^k$ is the position estimation of the kth joint. We use A to denote the region of likelihood and $H_k(p)$ to represent the likelihood value on point p. Therefore, every heatmap matrix contains the information to produce an initial pose P_{init}.

3 Details of the Network Architecture

We explain the detailed settings of the layers and parameters of our OPEC-Net in this section. Firstly, the proposals J_k are normalized into the range of $[-1, 1]$. There is a ReLu layer after each deep GCN layer [1] in our network. In the module of Cascaded Feature Adaption (CFA), each convolutional layer is of a 3×3 kernel and is followed by a ReLu function. In IGP-GCN, the first and second GCN layers are of 128 channels. The first ResGCN Attention Block (RAB) takes a 128-channel feature map as input and outputs a feature map with 256 channels. Both the input and output channels of the following two RABs are in 256. The last block takes a 256-channel feature map together with a 128-channel feature map as input and outputs a 256-channel feature map.

4 ResGCN Attention Block

As seen in Fig 2, each RAB consists of three GCN layers[1] and a self-attention layer. To ease the learning, we also add a residual link in the RAB to make the network being focusing on inferring residual value. For this goal, before conducting addition with the output of the lower two GCN layers, we also involve a GCN layer (the upper one in the Figure) to adjust the feature size of the input. Furthermore, we append a self-attention layer at the end to learn the dependency of joints and utilize this to capture more informative understanding of the inherent relationships in the human pose, for more accurate estimation.

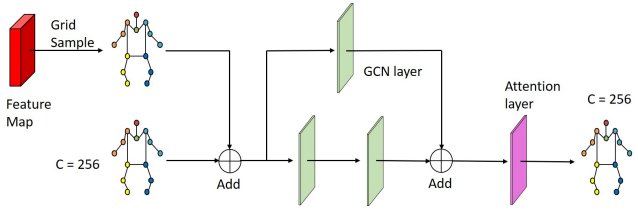

Fig. 2. The ResGCN Attention Block. C denotes the channel size of the feature.

5 Cascaded Feature Adaption (CFA) Module

As seen in Fig 3, the CFA module consists of two Fusion blocks (Fig 3) and three Conv blocks. In the experiments, we use one convolution layer in each Conv block. More specifically, the CFA module takes the three feature maps $\mathcal{F}_1, \mathcal{F}_2$, and $\mathcal{F}_3$ as input and outputs $\hat{\mathcal{F}}_1, \hat{\mathcal{F}}_2$, and $\hat{\mathcal{F}}_3$. Firstly, $\mathcal{F}_1$ is transformed to $\hat{\mathcal{F}}_1$ by a convolutional block. It can be formulated as

$$\hat{\mathcal{F}}_1 = Conv(\mathcal{F}_1; \theta). \tag{2}$$

We subsequently fuse two feature maps $\hat{\mathcal{F}}_1$ and $\mathcal{F}_2$ by a Fusion block, which gets a new feature map $\hat{\mathcal{F}}_2$ produced. This process will also be performed repeatedly again which generates $\hat{\mathcal{F}}_3$. We formulate this process as

$$\hat{\mathcal{F}}_{i+1} = Fusion(\hat{\mathcal{F}}_i, \mathcal{F}_{i+1}; \theta). \tag{3}$$

Thanks to such CFA module, the feature maps $\hat{\mathcal{F}}_1, \hat{\mathcal{F}}_2$, and $\hat{\mathcal{F}}_3$ are more informative and adaptive, than $\mathcal{F}_1, \mathcal{F}_2$, and $\mathcal{F}_3$, for the next stage.

$$Heatmap = Conv(\hat{\mathcal{F}}_3; \theta). \tag{4}$$

In particular, the Fusion Block takes $\hat{\mathcal{F}}_i$ and $\mathcal{F}_{i+1}$ as input and outputs $\hat{\mathcal{F}}_{i+1}$. The two feature maps are concatenated and are passed into an attention layer,

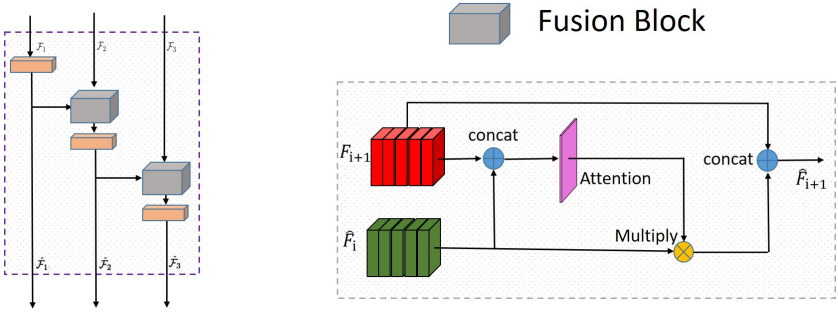

Fig. 3. The left figure shows the design of the Cascaded Feature Adaption module and the right one demonstrates the design of Fusion block.

which learns a weighting way for automatically combining information from both sides.

6 Result Gallery

In this section, we present more visually comparison results on the three datasets: OCPose, OCHuman [4], and CrowPose [2]. For each example, the left result is obtained from the AlphaPose+, and the right ones are estimated by our OPEC-Net.

Fig. 4. Result Gallery of the OCPose. For each example, the left result is from AlphaPose+ while the right is from ours OPEC-Net.

Fig. 5. Result Gallery of the CrowdPose. For each example, the left result is from AlphaPose+ while the right is from ours OPEC-Net.

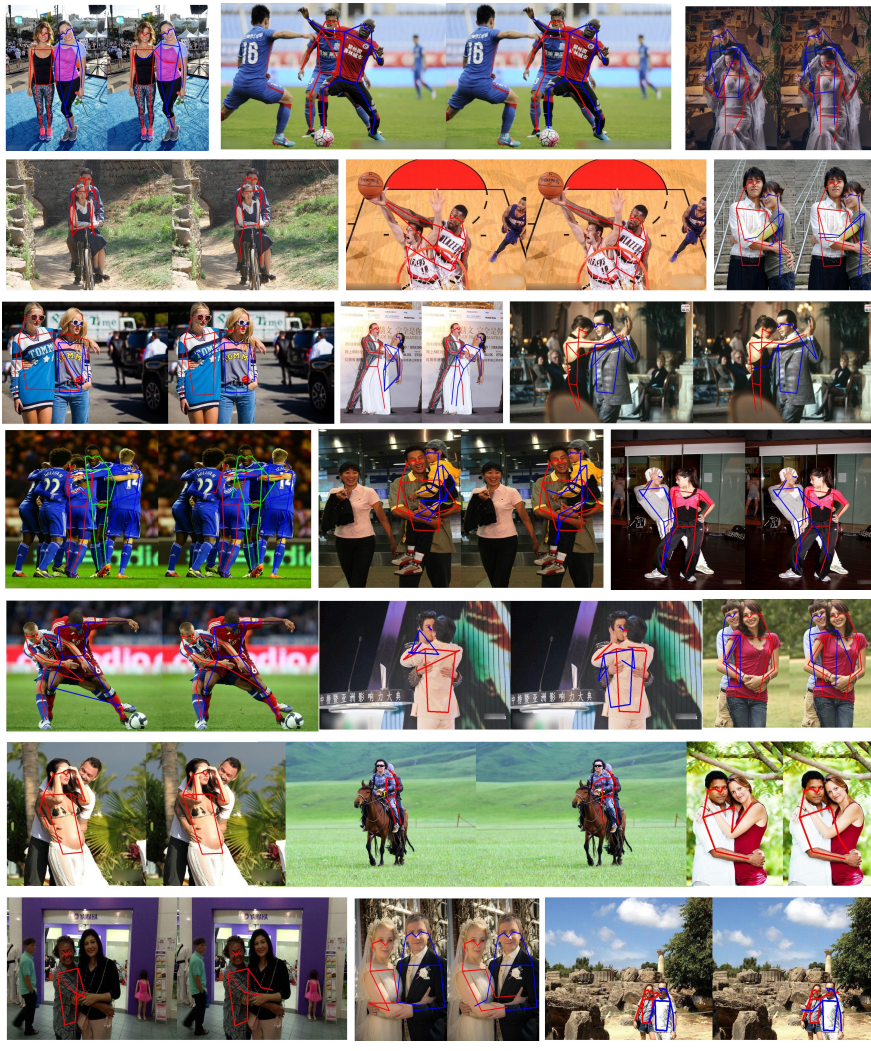

Fig. 6. Result Gallery of the OCHuman. For each example, the left result is from AlphaPose+ while the right is from ours OPEC-Net.

References

1. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9267–9276 (2019)
2. Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.S., Lu, C.: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10863–10872 (2019)
3. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 529–545 (2018)
4. Zhang, S.H., Li, R., Dong, X., Rosin, P., Cai, Z., Han, X., Yang, D., Huang, H., Hu, S.M.: Pose2seg: Detection free human instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 889–898 (2019)