Efficient Online Multi-Person 2D Pose Tracking with Recurrent Spatio-Temporal Affinity Fields (Supplementary Material)

Yaadhav Raaj Haroon Idrees Gines Hidalgo Yaser Sheikh
The Robotics Institute, Carnegie Mellon University
{raaj@cmu.edu, hidrees@cs.cmu.edu, gines@cmu.edu, yaser@cs.cmu.edu}

1. Model Training

During training, we unroll each model so that it can handle multiple frames at once. Each model is first pre-trained in **Image Mode** where we present a single image or frame at each time instant to the model. This implies multiple applications of PAF/KP stages to the same frame. We train with COCO, MPII and PoseTrack datasets with a batch distribution of 0.7, 0.2 and 0.1, respectively, matching dataset sizes, where each batch consists of images or frames from one dataset exclusively. We use the head bounding box information in MPII and Posetrack datasets to mask out the eyes/nose/ears in the background heatmap channel when considering the MPII and PoseTrack batches, and mask out the neck and top head positions using the annotated eyes/nose/ears keypoints for COCO batches. The net is input images of 368 $\times$ 368 dimensions and has scaling, rotation and translation augmentations, where regions not annotated are masked out. Heatmaps are computed with an ℓ_2 loss with a stride of 8 resulting in 46 $\times$ 46 dimensional heatmaps. In topology (b) and (c), we initialize the TAF with PAF, and zeros for (a). We train the net for max of 400k iterations.

Param	**Image Mode**	**Video Mode**
Input Resolution	368x368	368x368
Heatmap Resolution	46x46	46x46
Data Dist. (COCO, MPII, PT)	0.7,0.2,0.1	0.4,0.1,0.5
Frame Skip Augmentation	-	3
Scaling	$[0.7x, 1.3x]$	$[0.7x, 1.3x]$
Rotation	$[-30°, 30°]$	$[-20°, 30°]$
Translation	$[-30, 30]$	$[-50, 50]$
VGG Learning Rate	0.00004	0.0
PAF/TAF/KP Learning Rate	0.00008	0.00004
Solver	Adam	Adam
Momentum and Decay β_1, β_2	0.9, 0.999	0.9, 0.999
Decay	0.0005	0.0005
Step	[150k, 250k, 360k]	[100k, 200k, 250k]
Step Size	0.5	0.5
Total Epochs	400k	300k

Table 1: Training Parameters for both Image Mode and Video Mode

Next, we proceed training in the **Video Mode** where we expose the network to video sequences. For static image datasets including COCO and MPII, we augment data with video motion sequences by synthesizing motion with scaling, rotation and translation over the unroll count. We train COCO, MPII and PoseTrack in Video Mode with a batch distribution of of 0.4, 0.1 and 0.5, respectively. Moreover, we also use skip-frame augmentation for video-based PoseTrack dataset, where some of the randomly selected sequences skip up to 3 frames. We lock the weights of VGG module in Video Mode and only train the STAFs and keypoints blocks. For Model I, we only trained the TAFs block when training on videos. For Model II, we trained PAFs, keypoints and TAFs for 5000 epochs, then locked PAFs and keypoints before training TAFs only. In Model III, both STAFs and keypoints were kept unlocked and were trained for 300k iterations.

1.1. Inference and Tracking

The method described till now predicts heatmaps of keypoints and STAFs at every frame by running CNNs associated with each module while passing required data computed from previous frame. Next, we present the framework to perform pose inference as well as tracking across frames given the output heatmaps. Let the inferred poses at time t and $t-1$ be given by:

$$\begin{aligned}\mathbf{P}^{t} &= \{\mathbf{P}^{t,1}, \mathbf{P}^{t,2}, \ldots, \mathbf{P}^{t,N}\},\\ \mathbf{P}^{t-1} &= \{\mathbf{P}^{t-1,1}, \mathbf{P}^{t-1,2}, \ldots, \mathbf{P}^{t-1,M}\},\end{aligned} \tag{1}$$

where the second superscript indexes over people in each frame. Each pose at a particular time $\mathbf{P}^{t,n}$ consists of up to K keypoints post inference, i.e., $\mathbf{P}^{t,n}$ includes only those keypoints that become part of a pose:

$$\mathbf{P}^{t,n} = \{\overline{\mathbf{K}}_1^{t,n}, \overline{\mathbf{K}}_2^{t,n}, \ldots, \overline{\mathbf{K}}_K^{t,n}\}. \tag{2}$$

The detection and tracking procedure begins with localization of keypoints at time t. The inferred keypoints $\overline{\mathbf{K}}^t$ are obtained by rescaling the heatmaps to match the original image resolution followed by non-maximal suppression.

Algorithm 1 : Estimation and tracking of keypoints and STAFs

Input: $\mathbf{K}^t, \mathbf{L}^t, \mathbf{R}^t$, and $\mathbf{P}^{t-1}$ with unique ids
Output: $\mathbf{P}^t$ with ids

procedure INFERPOSES()
  Compute $\overline{\mathbf{L}}^t$ given $\overline{\mathbf{K}}^t$, $\mathbf{L}^t$ and $\ddot{\mathbf{L}}$
  Compute $\overline{\mathbf{R}}^t$ given $\overline{\mathbf{K}}^t$, $\mathbf{R}^t$, $\mathbf{P}^{t-1}$ and $\ddot{\mathbf{R}}$
  Sort $\overline{\mathbf{L}}^t$ and $\overline{\mathbf{R}}^t$ by score
  Initialize empty map of people $\mathbf{P}^t$
  for every $\overline{\mathbf{L}}^{t,n}_{k\to k'}$ in $\overline{\mathbf{L}}^t$ **do**
    If k and k' unassigned; add new $\mathbf{P}^{t,n}$
    If k or k' assigned; add to existing $\mathbf{P}^{t,n}$
    If k and k' assigned; update score of $\mathbf{P}^{t,n}$
    If k assigned to $\mathbf{P}^{t,n}$ and k' assigned to $\mathbf{P}^{t,n'}$
    & $\mathbf{P}^{t,n}$ and $\mathbf{P}^{t,n'}$ lack opposing points;
    merge $\mathbf{P}^{t,n}$ and $\mathbf{P}^{t,n'}$.
  end for
  for $\mathbf{P}^{t,n}$ in $\mathbf{P}^t$ **do**
    for $\overline{\mathbf{K}}^{t,n}$ in $\mathbf{P}^{t,n}$ **do**
      Find $\overline{\mathbf{R}}^t$ with the highest score; copy id
    end for
    Update $\mathbf{P}^{t,n}$ with the most frequent id
  end for
  If insufficient keypoint matches for $\mathbf{P}^{t,n}$;
  initialize tracklet
  Remove $\mathbf{P}^{t-1,n}$ if no association made
end procedure

Then, we infer PAF and TAF weights between all possible pairs of keypoints in each frame defined by the given topology, i.e.,

$$\overline{\mathbf{L}}^t_{k\to k'}\omega(\overline{\mathbf{K}}^t_k, \overline{\mathbf{K}}^t_{k'}), \quad \overline{\mathbf{R}}^t_{k\to k'} \tag{3}$$

where the function $\omega(\cdot)$ samples points between the two argument keypoints, computes the dot product between directional vector among the two points and the mean vector of the sampled points. The inference of PAF, $\overline{\mathbf{L}}^t$, and TAF, $\overline{\mathbf{R}}^t$, weights is constrained by the spatio-temporal topology, where the spatial and temporal constraints are encoded in tables $\ddot{\mathbf{L}}$ and $\ddot{\mathbf{R}}$, respectively.

$$max\left(\sum_t^N\sum_k^K\left(\overline{\mathbf{L}}^{t,n}_{k\to k'} + \overline{\mathbf{R}}^t_{k\to k'}\right)\right) \tag{4}$$

Overall, we wish to generate a set of people $\mathbf{P}^t$ with ids that maximizes the connection scores between their keypoints pairs $(\overline{\mathbf{K}}^{t-1,n}_k, \overline{\mathbf{K}}^{t,n}_{k'})$, and temporal connections given previous keypoints $\overline{\mathbf{K}}^{t-1,n}$ from tracklets $\mathbf{P}^{t-1,n}$ given an association.

To do this, both the inferred PAF and TAF weights are computed then sorted by their scores before inferring the complete poses, $\mathbf{P}^t$, and associating them across frames with unique ids. We perform this in a bottom-up style as described in Algorithm 1 where we utilize $\overline{\mathbf{R}}^t$ and $\mathbf{P}^{t-1}$ to determine the update, addition or deletion of tracklets. Going through each PAF in the sorted list, (i) we initialize a new pose if both keypoints in the PAF are unassigned, (ii) add to existing pose if one of the keypoints is assigned, (iii) update score of PAF in pose if both are assigned to the same pose, and (iv) merge two poses if keypoints belong to different poses with opposing keypoints unassigned. Finally, we assign id to each pose in the current frame with the most frequent id of keypoints from the previous frame. This is done over all tracklets and people very quickly as it is done on the GPU.

Furthermore, we make use of past poses $\mathbf{P}^{t-1}$ and TAFs $\overline{\mathbf{R}}^t$ to reweigh PAFs. For cases where we have an ambiguous PAFs (Alg. 1, 7:) as seen in Figure 1, we use transitivity that reweighs PAFs to disambiguate between them. In this figure, keypoint $\{A\}$ - an elbow - is under consideration, with wrists $\{B\}/\{E\}$ as two possibilities. We select the strongest TAFs where $\{A, B, C, D, A\}$ has a higher weight than $\{A, E, F, G, A\}$.

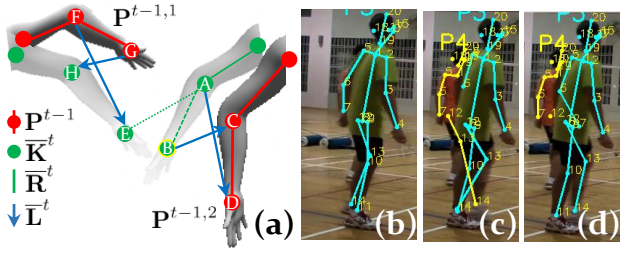

Figure 1: **(a)** This figure shows how the transitivity property can be used to improve scores of ambiguous PAFs. In this case, there are two possible wrist locations: one correct on the right and an incorrect on the left from an adjacent person. After reweighing PAFs through TAFs, the PAF pointing to the correct wrist receives a higher score. **(b)-(d):** After using transitivity, the incorrect PAFs from the leg of the occluded person in yellow are down-weighted resulting in correct pose estimation for the person in cyan.

$$\overline{\mathbf{L}}^{t,n}_{k\to k'} = (1-\alpha)\omega(\overline{\mathbf{K}}^{t-1,n}_k, \overline{\mathbf{K}}^{t,n}_{k'}) + \alpha * \omega(\overline{\mathbf{K}}^{t,n}_k, \overline{\mathbf{K}}^{t,n}_{k'}). \tag{5}$$

2. Additional Experiments

We present some experiments that were otherwise not displayed in detail in the main paper.

For the sake of completion, we first report results on the COCO dataset in Table 2. Despite using single set of weights for all the stages, we were able to get close results.

Our network is designed to be lightweight and work in a recurrent fashion, so our main reference point is still the Posetrack datasets.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Top-Down Approaches					
Megvii [2]	73.0	91.7	80.9	69.5	78.1
G-RMI [6]	71.0	87.9	77.7	69.0	75.2
Mask R-CNN [3]	69.2	90.4	76.0	64.9	76.3
Bottom-Up Approaches					
PersonLab [5]	68.7	89.0	75.4	64.1	75.5
Associative Emb. [4]	65.5	86.8	72.3	60.6	72.6
OpenPose 2018 [1]	64.4	86.5	70.2	61.5	68.8
Proposed	61.5	82.2	67.1	58.5	66.7

Table 2: Results on the COCO test-dev dataset. Top: top-down results. Bottom: bottom-up results (top methods only). AP^{50} is for OKS $= 0.5$, AP^{L} is for large scale persons.

Filter Sizes: We observed that having each 7×7 filter replaced with a three 3×3 filter resulted in better accuracies, especially for knees and ankles. The results are shown in Table 3. We run single frame inference on Model I and find the 3×3 to be 2% more accurate than 7×7, with significant boosts in average precision of knee and ankle keypoints.

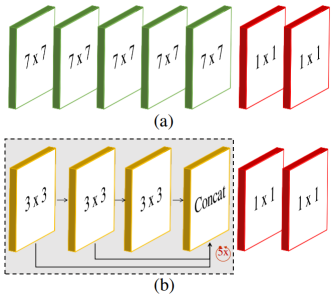

Figure 2: Types of filters usued

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP	fps
Model I - 3x3	75.7	73.9	67.8	56.3	66.8	62.3	56.9	66.3	14
Model I - 7x7	76.0	73.3	66.4	54.0	63.4	59.2	52.2	64.3	10

Table 3: This table shows results for experiments with the two filter sizes on PoseTrack 2017 validation set.

Recurrence of Keypoints Module: To verify that the network was indeed benefiting from ingesting previous frame heatmaps, we explicitly train a model where the keypoint module was connected to current PAF module in an *auxiliary* fashion and did not receive previous heatmaps (first row of Table 4). The second row shows the case where keypoint module was *connected* to ingest output heatmaps from previous frames. The result is 2.1% improvement at single scale on the PoseTrack 2017 validation set using Model II with 7×7 filter size.

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP	fps
KP Auxiliary	72.3	71.2	63.9	51.4	60.1	56.3	50.0	61.5	28
KP Connected	76.2	71.6	64.5	51.9	62.6	59.3	52.5	63.6	27

Table 4: Performance using Model II where the keypoint module does not take feedback from previous heatmaps (auxiliary), and when it does ingest previous heatmaps (connected).

STAF Topology: We experimented with Topology A, B and C. Topology B proved to be better than A due to it's ability to preserve information even during minimal motion, lending itself better to the recurrent structure of our network. It especially performed better during jittery camera motion, or during crowded scenes with several people. Topology C, which does not consist of any current frame spatial information, was difficult to train and resulted in an MAP that was about 8% lower. This was mainly because we had to construct a person during the first frame, or during a new person appearance, and simply propagate it using the TAF, which proved to be less reliable than extracting poses on every frame with PAF/TAF, then propagating it with TAF.

3. Implementation

Training: We train on 4 Titan XP in Caffe. Testing: We test on a single 1080 Ti, and i7 7800X. We write our own code in C++ and CUDA.

References

[1] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2D pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. 3

[2] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. *arXiv:1711.07319*, 2017. 3

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 3

[4] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, volume 2017., 2017. 3

[5] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy. PersonLab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. *arXiv:1803.08225*, 2018. 3

[6] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, 2017. 3