TEMPORAL FEATURE CORRELATION FOR HUMAN POSE ESTIMATION IN VIDEOS

Wentian Li ⋆ *Xiangyu Xu* ⋆ *Yu-Jin Zhang*

Department of Electronic Engineering, Tsinghua University, Beijing, China

ABSTRACT

Effectively utilizing temporal information is critical for human pose estimation in videos. Recent methods either neglect the displacements of keypoints in the video frames, or rely on time-consuming optical flow estimation when fusing temporal information. By contrast, we propose a flow-free and displacement-aware algorithm for pose estimation in videos. Our method is based on the observation that the appearance of the body keypoints remains almost unchanged throughout a video. This motivates us to exploit temporal visual consistency of keypoints via temporal feature correlation to establish sparse correspondences between the keypoints in neighboring frames. Specifically, we first extract keypoint features from the previous frame, which can be treated as exemplars to search on the intermediate feature map of the current frame. Then we conduct temporal feature correlation for the keypoint search, and the obtained correlation maps are combined with the convolutional features to further guide heatmap estimation. Extensive experiments demonstrate that the proposed method compares favorably against state-of-the-art approaches on both sub-JHMDB and Penn Action datasets. More importantly, our method is robust to large keypoint displacements and could be applied to videos under fast motion.

Index Terms— Human pose estimation, feature correlation, temporal consistency, CNN

1. INTRODUCTION

Human pose estimation has been extensively studied in recent years. Traditional approaches for this problem often rely on pictorial structure models [1, 2] to locate human keypoints in images. Most recently, convolutional neural networks (CNNs) [3, 4, 5, 6] have been effectively used for pose estimation. Rather than directly regressing human joint coordinates [5], it is advantageous to make predictions by estimating heatmaps [6, 7, 8, 9, 10] where every spatial location has a detection score for each keypoint.

More closely related to our work, these convolutional architectures have also been widely used for video pose estimation [11, 12], where the input consists of multiple frames. Different from the single image case, the usage of temporal information is critical for pose estimation in videos as the results of previous frames provide important clues about the keypoint locations in the current frame.

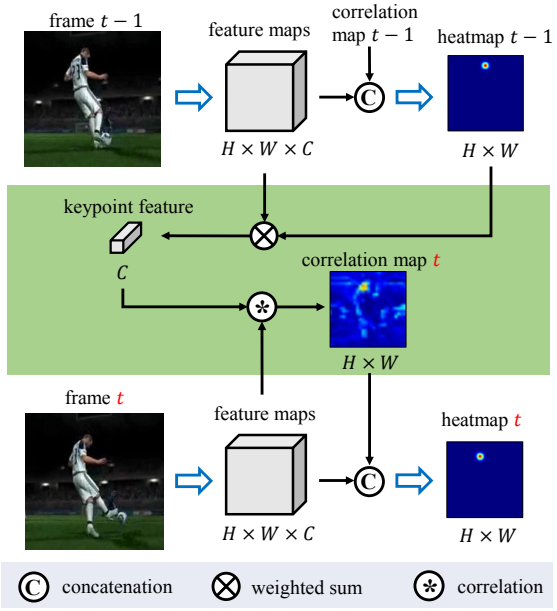

Fig. 1. Overview of the proposed algorithm. For simplicity, we only show the heatmap and correlation map for one of the keypoints. We utilize the features of both the current and previous frames to help the pose estimation of the current frame t. The middle part represents the proposed feature correlation layer which extracts keypoint features from frame $t-1$ and generates correlation map for the current frame t. The correlation map is concatenated with the convolutional features and then fed into the pose estimator. More detailed descriptions can be found in Section 2 and Section 3.

Following this principle, one line of works [7, 12, 13] apply inter-frame heatmap refinement to improve the estimation performance. To better learn temporal dependency, Gkioxari *et al*. [7] adopt a chained model which transforms the heatmap refinement process to the temporal axis. Similarly, by integrating an LSTM module into the CPM [6], the LSTM Pose Machine [12] propagates the hidden states as well as the output heatmaps through the temporal space, improving both accuracy and inference speed. The STAF [13] replaces five single-frame refinement stages of its baseline model with one inter-frame refinement stage, and observes significant speed-

⋆ The first two authors contribute equally to this work.

up with a minor drop in accuracy. While being effective, these approaches do not explicitly deal with keypoint displacements between frames, and therefore are prone to be affected by human and camera motions.

Another line of methods [11, 14] explicitly tackle keypoint displacements using dense optical flow, where the heatmaps from neighboring frames are warped to the current frame in order to merge the results and enforce temporal consistency. However, dense optical flow algorithms are computationally expensive [15, 16], which makes these flow-based methods non-practical for real use. Besides, recent work [17] shows that generic flow methods are insufficient to capture human motion in real scenes, and often miss small structures that are typical of humans (*e.g.* hands and legs).

In contrast to the above-mentioned approaches, we solve the displacement problem without dense spatial alignment, and establish only sparse correspondences of the human keypoints by exploiting their visual consistency. Our method is built on the observation that the appearance of the same person, such as clothing, accessories and hairstyle, remains almost unchanged throughout the video [18, 19].

An overview of our method is shown in Figure 1. We first extract features for each human keypoint from the previous frame. Then we use the extracted keypoint features as exemplars to search on the feature maps of the current frame via a correlation layer. The correlation maps obtained from this layer strongly imply the visually plausible keypoint candidates, which could effectively help the estimation of the current frame. Thus, we concatenate the correlation maps with the original CNN features and feed them into the subsequent estimators for final pose estimation. Note that our algorithm could achieve better temporal consistency without dense correspondence maps, and naturally solves the problem of large displacements.

The contributions of this work could be summarized as follows. First, we propose an effective temporal feature correlation method for estimating human pose in videos without dense optical flow estimation. Second, we integrate the correlation layer into the inter-frame refinement process, which achieves state-of-the-art performance on both sub-JHMDB and Penn Action datasets. Third, we show that the proposed method is robust to large keypoint displacements and thus could be applied to videos under fast motion.

2. TEMPORAL FEATURE CORRELATION

In this work, we propose a temporal feature correlation layer to efficiently exploit the visual consistency of keypoints. We first extract keypoint features and then compute the correlation maps via the correlation layer.

Keypoint feature extraction. The proposed keypoint feature is a weighted sum of feature maps over the spatial dimension. Given the intermediate feature maps $\mathbf{I} \in \mathbf{R}^{H\times W\times C}$ (H, W and C denotes height, width and number of channels respectively) and heatmaps $\mathbf{B} \in \mathbf{R}^{H\times W\times P}$ (P represents number of keypoints), we use $\mathbf{B}(h, w, p)$ as the weight of location (h, w) on $\mathbf{I}$ for keypoint p ($p \in 1, ..., P$). Thus, the pixels closer to the predicted location of keypoint p get larger weights, and all the pixels with non-zero detection scores are taken into account for keypoint feature extraction. Specifically, the feature for keypoint p is defined as:

$$\mathbf{J}_p(c) = \frac{1}{N_p} \sum_{h,w} \mathbf{B}(h, w, p)\mathbf{I}(h, w, c) \tag{1}$$

Here (h, w) enumerates all the locations on the feature map and the heatmap. The resulting keypoint feature $\mathbf{J}_p \in \mathbf{R}^C$ is normalized by a factor N_p so that it has an L_2 norm of 1, which eliminates the effects caused by the varying magnitude of both the heatmaps and the features.

Note that our design is not tied to any specific network architecture and can be applied to any pose estimator that has an intermediate feature map whose spatial dimension equals to that of the output heatmaps.

Feature correlation. We use inner-product to compute similarity between feature vector pairs, which is a special case of correlation and the computation is much simpler than that in [16, 20] where the features are 3D and the correlation is limited to the neighboring area of the exemplars. We compute the correlation between the keypoint feature $\mathbf{J}_p$ extracted from $\mathbf{I}_{t-1}$ (the feature maps of the previous frame), and the feature on each location of $\mathbf{I}_t$ (the feature maps of the current frame). The resulting correlation maps have a size of $H \times W \times P$, where every pixel has P scores each implying how likely the keypoint reappears at that location.

While the correlation maps are important cues for pose estimation, they are unsuitable for directly deriving the keypoint locations. The visual characteristics of the keypoints may shift due to the change of pose and background, and the correctness of keypoint detections are not guaranteed. Therefore, we concatenate the obtained correlation maps with the convolutional features $\mathbf{I}_t$ for final heatmap regression.

3. NETWORK ARCHITECTURE

In this section, we first introduce the architecture of our baseline model. Then we elaborate on how we integrate the correlation maps into the model to help heatmap regression.

Baseline model. We adopt part affinity fields (PAF) [21] as our baseline model for single-person pose estimation in videos. Affinity fields encode the orientation of predefined body parts (*e.g.* , arm and leg). Although originally designed for multi-person pose estimation task, affinity fields are helpful for filtering out implausible and redundant keypoint detections on the heatmaps. PAF is composed of a feature extractor and six subsequent heatmap refinement modules, namely

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 2. Examples of correlation maps. Input (a) and (b) are the previous frame and the current frame. The corresponding keypoints in the correlation maps (c)-(e) and in the heatmaps (f)-(h) are *head*, *right wrist* and *right ankle* respectively. The correlation maps and heatmaps are resized and overlaid with the current frame for visualization, and better viewed in color, with red indicating high response and blue indicating low response. The pose estimation is shown in (i).

Stage 1 to Stage 6. The output of the feature extractor and the heatmaps have a spatial size of $H \times W$ for an $8H \times 8W$ input.

Integrating correlation maps. As illustrated in Figure 1, the output of the feature extractor with the size of $H \times W \times C$ is selected as the intermediate feature maps. A scale layer with learnable weights and biases is added right after the correlation maps for better convergence. We feed the scaled correlation maps along with the current feature maps into Stage 1. This increases the input channel of Stage 1 from C to $(C+P)$, and therefore, more convolutional filters are added to handle the additional input channels. Other stages do not receive the correlation maps as input, because we find that feeding the correlation maps into only Stage 1 is enough for leveraging the temporal information from previous frames, while feeding into all modules could lead to overfitting issues.

Similar to the LSTM Pose Machine [12], our model runs in a chronological order. Since the keypoint feature is unavailable for the first frame of a video, the initial correlation maps are set to zeros.

4. EXPERIMENTS

We present our experimental setups and compare our method with previous single-person video pose estimation methods.

Datasets. We conduct experiments on two widely used datasets for single-person pose estimation. 1) JHMDB dataset. Following [12], experiments are conducted on a subset of JHMDB [22] called sub-JHMDB, which excludes incomplete human bodies. Sub-JHMDB has 316 video clips in total and provides 3 train/test splits. We report the average performance over three splits. The dataset was originally annotated using a puppet model, so no occlusion information was given. We treat all the keypoints as visible during training and testing. 2) Penn Action dataset. Penn Action Dataset [23] contains 2326 video clips. Among the 15 human keypoints that are annotated in MPII Dataset [24], annotations for *neck* and *belly* are absent in Penn Action Dataset. Therefore, we interpolate the locations of *neck* and *belly* for training, as they are indispensable for our baseline model to assemble keypoint pairs into person instances. The invisible keypoints do not contribute to the loss during training and are ignored during evaluation.

For data augmentation, we apply random resize with scale [0.7, 1.3], random rotation with degree $[-40°, 40°]$, and random horizontal flip. Images are cropped to a fixed size of 368×368 for training.

Evaluation protocol. Following previous works [12, 14], we use the PCK metric [25]. If the distance between the predicted keypoint and the ground truth one is less than 0.2 of the larger side of the bounding box of the person, the prediction is marked as correct. For sub-JHMDB, the bounding boxes are deduced from the puppet masks.

Handling multiple persons. A large number of videos in Penn Action Dataset have multiple persons present in the same frame with only one of them annotated. In such cases, the heatmaps could have multiple peaks, resulting in noisy keypoint features. Previous works [12, 14] require centermaps as input, which are generated from ground truth and indicate the central location of the person in interest. Analogously, for Penn Action Dataset we modify our baseline model and insert centermaps into Stage 1 to Stage 6, where additional convolutional filters are added to compensate the increased input channels. During testing, if the detected keypoints are assembled into more than one persons, the one that is closest to the ground truth person center is chosen as the final prediction. Centermaps are not necessary for sub-JHMDB.

Implementation details. Our baseline model is initialized using the weights provided by [21] and pretrained on MPI-

Table 1. Performance comparison on sub-JHMDB dataset.

Method	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
[26]	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5
[27]	80.3	63.5	32.5	21.6	76.3	62.7	53.1	55.7
[28]	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8
[14]	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1
LSTM [12]	**98.2**	96.5	89.6	86.0	**98.7**	**95.6**	90.9	93.6
PAF [21]	96.7	95.8	90.2	86.6	97.1	94.3	90.2	93.0
Ours	97.7	**96.6**	**91.5**	**88.1**	97.9	94.9	**91.1**	**94.0**

Table 2. Performance comparison on Penn Action Dataset.

Method	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
[26]	62.8	52.0	32.3	23.3	53.3	50.2	43.0	45.3
[27]	64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0
[28]	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1
[7]	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8
[14]	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.5
LSTM [12]	**98.9**	**98.6**	**96.6**	**96.6**	98.2	98.2	97.5	97.7
PAF [21]	98.4	98.2	96.1	95.9	98.5	97.9	97.6	97.5
Ours	98.8	98.5	96.6	96.6	**98.8**	**98.3**	**98.2**	**98.0**

I Dataset [24]. The added convolutional filters are initialized with a zero-mean and 0.01 std Gaussian distribution. Keypoint features are extracted randomly from one among ten frames that are temporally closest to the current frame. Ground truth heatmaps are used as weight to extract keypoint features so that the model would be more dependent on the input correlation maps. To make the model workable on the first frame of the video, we randomly set all correlation maps of some training samples to zero at the rate of 0.2.

We set batch size to 20 and use stochastic gradient descent with momentum of 0.9 and weight decay of 0.0005 as in [21]. The initial learning rate is 2×10^{-5} for the feature extractor and Stage 1, and 8×10^{-5} for other stages as well as the added convolutional filters. For sub-JHMDB, total iterations are 60k and the learning rate is decreased by a factor of 3 after 30k iterations. For Penn Action Dataset, the model is trained with centermaps and without correlation maps for 20k iterations, and then trained for another 60k iterations with correlation maps.

For evaluation, we test all images on a single scale. In sub-JHMDB, the persons are of smaller scales, so we enlarge the testing images with scale 1.2, which leads to slightly better results.

Comparison with our baseline and the state-of-the-arts. The results of previous methods and ours are listed in Table 1 for sub-JHMDB dataset and in Table 2 for Penn Action Dataset. Our model outperforms previous state-of-the-arts on both datasets.

For a fair comparison with our baseline model, we trained the baseline model alone under the same setting. By propagating temporal information, the accuracy of every keypoint category is improved on both datasets.

Comparison with inter-frame heatmap refinement. We also conduct experiments with the inter-frame heatmap refinement process similar to [7, 13]. Specifically, we simply replace the correlation maps with heatmaps from previous frames and train this new model. On sub-JHMDB dataset, its accuracy is 93.5, which is better than the baseline model but inferior to its counterpart with correlation maps. On Penn Action Dataset, there is no improvement over the baseline.

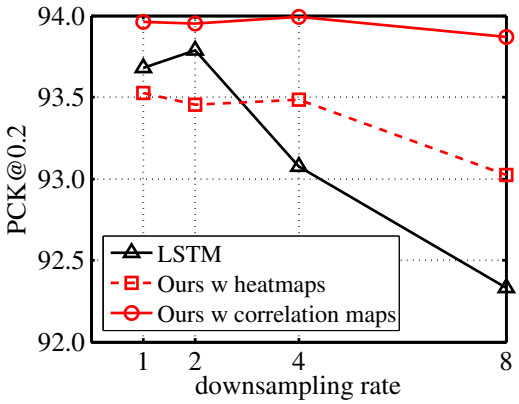

Fig. 3. Performance on sub-JHMDB dataset under different temporal downsampling rates. LSTM was tested with the source code provided in [12].

Results for videos with fast motion. For videos with fast motion, the keypoint displacements between frames are usually large, which can make it difficult to fuse multi-frame information and might degrade the performance of some previous video-based pose estimators. By contrast, our method relies on visual consistency of body parts, which is robust to large displacements.

To demonstrate the robustness of our proposed method, we temporally downsample the videos of sub-JHMDB dataset with different downsampling rates so that the neighboring frames have larger displacement. Figure 3 shows that our proposed method retains good performance on temporal downsampled input, whereas the accuracy of the LSTM Pose Machine drops significantly.

In Figure 2, we visualize some correlation maps for non-adjacent frame pairs. Despite large keypoint displacements, most keypoints are located correctly in the correlation maps. Since there are often some image patches that share similar visual characteristics with the keypoints, the correlation maps do not look clean.

5. CONCLUSION

In this paper, we exploit visual consistency of keypoints in video frames for human pose estimation. With keypoint features and correlation maps, we achieved substantial improvement for the pose estimator in the videos. We have experimentally shown the effectiveness and robustness of the proposed method.

6. REFERENCES

[1] M.Andriluka, S.Roth, and B.Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, 2009. 1

[2] M.Dantone, J.Gall, C.Leistner, and L.Van Gool, "Human pose estimation using body parts dependent joint regressors," in *CVPR*, 2013. 1

[3] J. J.Tompson, A.Jain, Y.LeCun, and C.Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NeurIPS*, 2014. 1

[4] X.Chen and A. L.Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *NeurIPS*, 2014. 1

[5] A.Toshev and C.Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*, 2014. 1

[6] S.-E.Wei, V.Ramakrishna, T.Kanade, and Y.Sheikh, "Convolutional pose machines," in *CVPR*, 2016. 1

[7] G.Gkioxari, A.Toshev, and N.Jaitly, "Chained predictions using convolutional neural networks," in *ECCV*, 2016. 1, 4

[8] W.Yang, S.Li, W.Ouyang, H.Li, and X.Wang, "Learning feature pyramids for human pose estimation," in *ICCV*, 2017. 1

[9] A.Newell, K.Yang, and J.Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016. 1

[10] K.He, G.Gkioxari, P.Dollár, and R.Girshick, "Mask r-cnn," in *ICCV*, 2017. 1

[11] T.Pfister, J.Charles, and A.Zisserman, "Flowing convnets for human pose estimation in videos," in *ICCV*, 2015. 1, 2

[12] Y.Luo, J.Ren, Z.Wang, W.Sun, J.Pan, J.Liu, J.Pang, and L.Lin, "Lstm pose machines," in *CVPR*, 2018. 1, 3, 4

[13] Y.Raaj, H.Idrees, G.Hidalgo, and Y.Sheikh, "Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields," *arXiv preprint arXiv:1811.11975*, 2018. 1, 4

[14] J.Song, L.Wang, L.Van Gool, and O.Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in *CVPR*, 2017. 2, 3, 4

[15] P.Weinzaepfel, J.Revaud, Z.Harchaoui, and C.Schmid, "Deepflow: Large displacement optical flow with deep matching," in *ICCV*, 2013. 2

[16] A.Dosovitskiy, P.Fischer, E.Ilg, P.Hausser, C.Hazirbas, V.Golkov, P.Van Der Smagt, D.Cremers, and T.Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015. 2

[17] A.Ranjan, J.Romero, and M. J.Black, "Learning human optical flow," in *BMVC*, 2018. 2

[18] K.Fragkiadaki, H.Hu, and J.Shi, "Pose from flow and flow from pose," in *CVPR*, 2013. 2

[19] J.Charles, T.Pfister, D.Magee, D.Hogg, and A.Zisserman, "Personalizing human video pose estimation," in *CVPR*, 2016. 2

[20] L.Bertinetto, J.Valmadre, J. F.Henriques, A.Vedaldi, and P. H.Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016. 2

[21] Z.Cao, T.Simon, S.-E.Wei, and Y.Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017. 2, 3, 4

[22] H.Jhuang, J.Gall, S.Zuffi, C.Schmid, and M. J.Black, "Towards understanding action recognition," in *ICCV*, 2013. 3

[23] W.Zhang, M.Zhu, and K. G.Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *ICCV*, 2013. 3

[24] M.Andriluka, L.Pishchulin, P.Gehler, and B.Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014. 3, 4

[25] Y.Yang and D.Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013. 3

[26] D.Park and D.Ramanan, "N-best maximal decoders for part models," in *ICCV*, 2011. 4

[27] B.Xiaohan Nie, C.Xiong, and S.-C.Zhu, "Joint action recognition and pose estimation from video," in *CVPR*, 2015. 4

[28] U.Iqbal, M.Garbade, and J.Gall, "Pose for action-action for pose," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2017, pp. 438–445. 4