LightTrack: A Generic Framework for Online Top-Down Human Pose Tracking

Guanghan Ning[1], Jian Pei[2], Heng Huang[1, 3]
[1]JD Finance America Corporation [2]Simon Fraser University [3]University of Pittsburgh

Abstract

In this paper, we propose a simple yet effective framework, named LightTrack, for online human pose tracking. Existing methods usually perform human detection, pose estimation and tracking in sequential stages, where pose tracking is regarded as an offline bipartite matching problem. Our proposed framework is designed to be generic, efficient and truly online for top-down approaches. For efficiency, Single-Person Pose Tracking (SPT) and Visual Object Tracking (VOT) are incorporated as a unified online functioning entity, easily implemented by a replaceable single-person pose estimator. To mitigate offline optimization costs, the framework also unifies SPT with online identity association and sheds first light upon bridging multi-person keypoint tracking with Multi-Target Object Tracking (MOT). Specifically, we propose a Siamese Graph Convolution Network (SGCN) for human pose matching as a Re-ID module. In contrary to other Re-ID modules, we use a graphical representation of human joints for matching. The skeleton-based representation effectively captures human pose similarity and is computationally inexpensive. It is robust to sudden camera shifts that introduce human drifting. The proposed framework is general enough to fit other pose estimators and candidate matching mechanisms. Extensive experiments show that our method outperforms other online methods and is very competitive with offline state-of-the-art methods while maintaining higher frame rates. Code and models are publicly available at https://github.com/Guanghan/lighttrack.

1. Introduction

Pose tracking is the task of estimating multi-person human poses in videos and assigning unique instance IDs for each keypoint across frames. Accurate estimation of human keypoint-trajectories is useful for human action recognition, human interaction understanding, motion capture and animation, *etc*. Recently, the publicly available PoseTrack dataset [1, 2] and MPII Video Pose dataset [3] have pushed the research on human motion analysis one step further to its real-world scenario. Two PoseTrack challenges have been held. However, most existing methods are offline hence lacking the potential to be real-time. More emphasis has been put on the *Multi-Object Tracking Accuracy* (MOTA) criterion while neglecting the *Frame Per Second* (FPS) criterion. Existing offline methods divide the tasks of human detection, candidate pose estimation, and identity association into sequential stages. In the procedure, multi-person poses are estimated across frames within a video. Based on the pose estimation results, pose tracking outputs are derived by solving an offline optimization problem. It requires the poses across frames to be pre-computed, or at least for the frames within some range.

In this paper, we propose a simple yet effective framework for pose tracking. It is designed to be generic, top-down (*i.e.*, pose estimation is performed after candidates are detected), and truly online. To efficiently perform pose tracking, we incorporate *Single-Person Pose Tracking* (SPT) and *Visual Object Tracking* (VOT) as a unified online functioning entity, easily implemented by a replaceable single-person pose estimatior. Therefore, object detection can be performed scarcely (in key frames). In order to mitigate the offline optimization cost, we unify single-person pose tracking with intervallic person re-identification, namely, key-frame pose matching. This problem conversion bridges multi-person keypoint tracking with multi-target object tracking. Since the proposed framework is general enough to fit other pose estimators and candidate matching mechanisms, advances in pose estimation, person re-identification and multi-target object tracking can be conveniently utilized for pose tracking in the future.

Specifically, in contrast to VOT methods, in which the visual features are implicitly represented by kernels or CNN feature maps, we track each human pose by recursively updating the bounding box and its corresponding pose in an explicit manner. The bounding box region of a target is inferred from the explicit features, *i.e.*, the human keypoints. Human keypoints can be considered as a series of special visual features. The advantages of using pose as explicit features include: (1) The features are human-related, interpretable, and have strong, stable correlation with the bounding box position. Human pose enforces direct constraint on the bounding box region. (2) The task of pose estimation

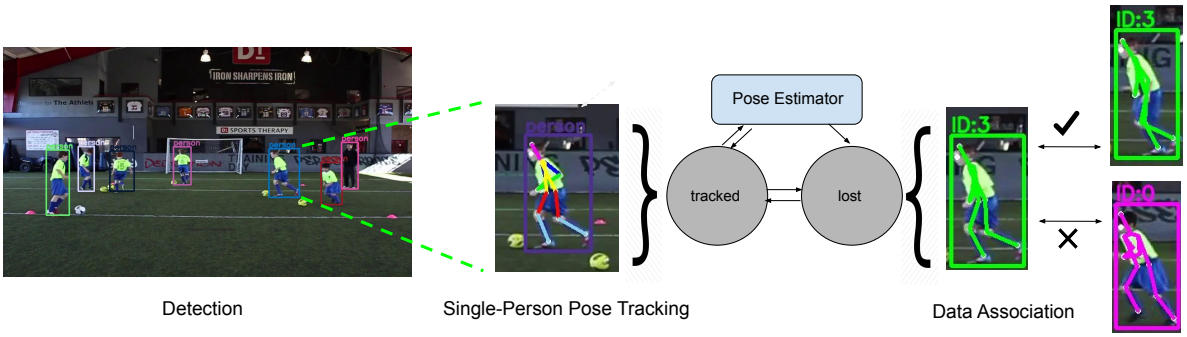

Figure 1. Overview of the proposed online pose tracking framework. We detect human candidates in the first frame, then track each candidate's position and pose by a single-person pose estimator. When a target is lost, we perform detection for this frame and data association with a graph convolution network for skeleton-based pose matching. We use skeleton-based pose matching because visually similar candidates with different identities may confuse visual classifiers. Extracting visual features can also be computationally expensive in an online tracking system. Pose matching is considered because we observe that in two adjacent frames, the location of a person may drift away due to sudden camera shift, but the human pose will stay almost the same as people usually cannot act that fast.

and tracking requires human keypoints to be predicted in the first place. Taking advantage of the predicted keypoints is efficient in tracking the ROI region, which is almost free. This mechanism makes the online tracking possible. (3) It naturally keeps the identity of the candidates, which greatly alleviates the burden of data association in the system. Even when data association is necessary, we can re-use the pose features for skeleton-based pose matching. In this way, SPT and VOT are unified with a replaceable single-person human pose estimation module.

Our contributions are three-fold: (1) We propose a general online pose tracking framework that is suitable for top-down approaches of human pose estimation. Both human pose estimator and Re-ID module are replaceable. In contrast to *Multi-Object Tracking* (MOT) frameworks, our framework is specially designed for the task of pose tracking. (2) We propose a *Siamese Graph Convolution Network* (SGCN) for human pose matching as a Re-ID module in our pose tracking system. Unlike existing Re-ID methods, we use a graphical representation of human joints for matching. The skeleton-based representation effectively captures human pose similarity and is computationally inexpensive. It is robust to sudden camera shift that introduces human drifting. (3) We conduct extensive experiments with various settings and ablation studies. Our proposed online pose tracking approach outperforms existing online methods and is competitive to the offline state-of-the-art methods, with much higher frame rates. We make the code publicly available to facilitate future research.

2. Related Work

Human Pose Estimation and Tracking: *Human Pose Estimation* (HPE) has seen rapid progress with the emergence of CNN-based methods [4, 5, 6, 7]. The most widely used datasets, *e.g.*, MPII [8] and LSP [9], are saturated with methods that achieve 90% and higher accuracy. Multi-person human pose estimation is more realistic and challenging, and has received increasing attentions with the hosting of COCO keypoints challenges [10] since 2017. Existing methods can be classified into top-down and bottom-up approaches. The top-down approaches [11, 12, 13] rely on the detection module to obtain human candidates and then applying single-person pose estimation to locate human keypoints. The bottom-up methods [14, 15, 16, 17] detect human keypoints from all potential candidates and then assemble these keypoints into human limbs for each individual based on various data association techniques. The advantage of bottom-up approaches is their excellent trade-off between estimation accuracy and computational cost because the cost is nearly invariant to the number of human candidates in the image. In contrast, the advantage of top-down approaches is their capability in disassembling the task into multiple comparatively easier tasks, *i.e.*, object detection and single-person pose estimation. The object detector is expert in detecting hard (usually small) candidates, so that the pose estimator will perform better with a focused regression space. Pose tracking is a new topic that is primarily introduced by the PoseTrack dataset [1, 2] and MPII Video Pose dataset [3]. A typical top-down but offline method was introduced in [3], where pose tracking is transformed into a minimum cost multi-cut problem with a graph partitioning formulation. Existing methods [18, 19, 20] are either offline or theoretically online but requires heavy overhead across frames pre-computed before performing an actual batch process. [21] propsosed to use box propagation to refine detection. In our approach, we also employs box propagation, but we incorporate the box propagation scheme with a pose estimator to form a single-object tracker. In their approach, detection is performed for every frame, while we only perform detection at keyframes.

Object Detection vs. Human Pose Estimation: Earlier works in object detection regress visual features into bounding box coordinates. HPE, on the other hand, usually re-

gresses visual features into heatmaps, each channel representing a human joint. Recently, research in HPE has inspired many works on object detection [22, 23, 24]. These works predict heatmaps for a set of special keypoints to infer detection results (bounding boxes). Based on this motivation, we propose to predict human keypoints to infer bounding box regions. Human keypoints are a special set of keypoints to represent detection of the human class only.

Multi-Object Tracking: MOT aims to estimate trajectories of multiple objects by finding target locations while maintaining their identities across frames. Offline methods use both past and future frames to generate trajectories while online methods are performed on the go. An online MOT pipeline [25] was presented where a single object tracker keeps tracking targets given their detections across frames. The target state is set as "tracked" until the tracking result turns unreliable. The target is then considered lost, and data association is performed to compute the similarity between the track-let and detections. Our proposed framework also tracks each target (with corresponding keypoints) individually while keeping their identities, and performs data association when target is lost. However, our framework is distinct in two aspects: (a) the detection is generated by object detector only at keyframes. It can be provided scarcely; (b) the single object tracker is actually a pose estimator that predicts keypoints based on an enlarged region.

Graphical Representation for Human Pose: It is recently studied in [26] on how to effectively model dynamic skeletons with a specially tailored graph convolution operation, which turns human skeletons into spatio-temporal representation of human actions. Inspired by this work, we propose to employ GCN to encode spatial interdependencies among human joints into a latent representation of human pose. The representation aims to robustly encode the pose, which is invariant to human location or view angle. We measure similarities of such encodings for the pose matching problem.

3. Proposed Method

3.1. Top-Down Pose Tracking Framework

We propose a novel top-down pose tracking framework. It has been proved that human pose can be employed for better inference of human locations [27]. We observe that, in a top-down approach, accurate human locations also ease the estimation of human poses. We further study the relationship between these two levels of information: (1) Coarse person location can be distilled into body keypoints by a single-person pose estimator. (2) The position of human joints can be straightforwardly used to indicate rough locations of human candidates. (3) Thus, recurrently estimating one from the other is a feasible strategy for SPT.

However, it is unreliable to merely consider the *Multi-target Pose Tracking* (MPT) problem as a repeated SPT problem for multiple individuals. Because certain constraints need to be met, *e.g.*, in a certain frame, two different IDs cannot belong to the same person, nor should two candidates share the same identity. A better way is to track multiple individuals simultaneously and preserve/update their identities occasionally with an additional Re-ID module. The Re-ID module is essential because it is usually hard to maintain correct identities all the way. It is unlikely to track the individual poses effectively across frames of the entire video. For instance, under the following scenarios, identities have to be updated: (1) human candidates disappear from the camera view or get occluded; (2) new candidates enter the scene or previous ones re-appear; (3) people walk across each other (two identities may merge into one if not treated carefully); (4) tracking fails due to fast camera shifting or zooming.

In our method, we first treat each human candidate separately such that their corresponding identity is kept across the frames. In this way, we circumvent the time-consuming offline optimization procedure. In case the tracked candidate is lost due to occlusion or camera shift, we then call the detection module to revive candidates and associate them to the tracked targets from the previous frame via pose matching. In this way, we accomplish multi-target pose tracking with an SPT module and a pose matching module.

Specifically, the bounding box of the person in the upcoming frame is inferred from the joints estimated by the pose module from the current frame. We find the minimum and maximum coordinates and enlarge this ROI region by 20% on each side. The enlarged bounding box is treated as the localized region for this person in the next frame. If the average confidence score $\bar{s}$ from the estimated joints is lower than the standard τ_s, it reflects that the target is lost since the joints are not likely to appear in the bounding box region. The state of the target is defined as:

$$\text{state} = \begin{cases} \text{tracked}, & \text{if } \bar{s} > \tau_s, \\ \text{lost}, & \text{otherwise.} \end{cases} \tag{1}$$

If the target is lost, we have two modes: (1) **Fixed Keyframe Interval (FKI)** mode. Neglect this target until the scheduled next key-frame, where the detection module re-generate the candidates and then associate their IDs to the tracking history. (2) **Adaptive Keyframe Interval (AKI)** mode. Immediately revive the missing target by candidate detection and identity association. The advantage of FKI mode is that the frame rate of pose tracking is stable due to the fixed interval of keyframes. The advantage of AKI mode is that the average frame rate can be higher for non-complex videos. In our experiments, we incorporate them by taking keyframes with fixed intervals while also calling detection module once a target is lost before the arrival of

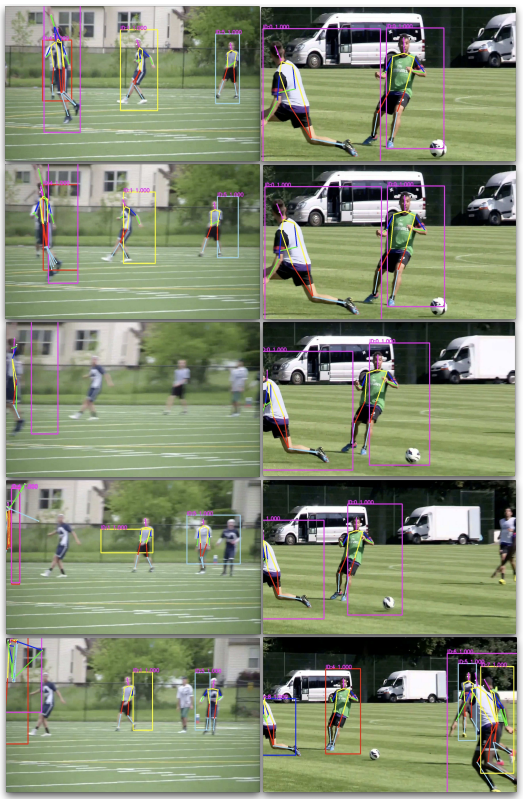

Figure 2. Sequentially adjacent frames with sudden camera shift (left frames), and sudden zooming (right frames). Each bounding box in the current frame indicates the corresponding region inferred from the human keypoints from the previous frame. The human pose in the current frame is estimated by the pose estimator. The ROI for the pose estimator is the expanded bounding box.

the next arranged keyframe. The tracking accuracy is higher because when a target is lost, it is handled immediately.

For identity association, we propose to consider two complementary pieces of information: spatial consistency and pose consistency. We prioritize spatial consistency, *i.e.*, if two bounding boxes from the current and the previous frames are adjacent, or their *Intersection Over Union* (IOU) is above a certain threshold, we consider them to belong to the same target. Specifically, we set the matching flag $m(t_k, d_k)$ to 1 if the maximum IOU overlap ratio $o(t_k, \mathcal{D}_{i,k})$ between the tracked target $t_k \in \mathcal{T}_k$ and the corresponding detection $d_k \in \mathcal{D}_k$ for key-frame k is above a threshold τ_o. Otherwise, $m(t_k, d_k)$ is set to 0:

$$m(t_k, d_k) = \begin{cases} 1, & \text{if } o(t_k, \mathcal{D}_{i,k}) > \tau_o, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The above criterion is based on the assumption that the tracked target from the previous frame and the actual location of the target in the current frame have significant overlap, which is true in most cases. However, such assumption is not always reliable, especially when the camera shifts swiftly. In such cases, we need to match the new observation to the tracked candidates. In Re-ID problems, this is usually accomplished by a visual feature classifier. However, visually similar candidates with different identities may confuse such classifiers. Extracting visual features can also be computationally expensive in an online tracking system. Therefore, we design a *Graph Convolution Network* (GCN) to leverage the graphical representation of the human joints. We observe that in two adjacent frames, the location of a person may drift away due to sudden camera shift, but the human pose will stay almost the same as people usually cannot act that fast, as illustrated in Fig. 2. Consequently, the graph representation of human skeletons can be a strong cue for candidate matching, which we refer to as pose matching in the following text.

3.2. Siamese Graph Convolutional Networks

Siamese Network: Given the sequences of body joints in the form of 2D coordinates, we construct a spatial graph with the joints as graph nodes and connections in human body structures as graph edges. The input to our graph convolutional network is the joint coordinate vectors on the graph nodes. It is analogous to image-based CNNs where the input is formed by pixel intensity vectors residing on the 2D image grid [26]. Multiple graph convolutions are performed on the input to generate a feature representation vector as a conceptual summary of the human pose. It inherently encodes the spatial interdependencies among the human joints. The input to the Siamese networks, therefore, is a pair of inputs to the GCN network. The distance between two output features indicate the similarity of the corresponding poses. Two poses are called a match if they are conceptually similar. The network is illustrated in Fig. 3. The Siamese network consists of 2 GCN layers and 1 fully convolutional layer. We take normalized keypoint coordinates as input; the output is a 128 dimensional feature vector. The network is optimized with contrastive loss $\mathcal{L}$ because we want the network to generate feature vectors, that are close by enough for positive pairs, whereas they are far away at least by a minimum for negative pairs. we employ the margin contrastive loss:

$$\mathcal{L}(p_j, p_k, y_{jk}) = \frac{1}{2} y_{jk} D^2 + \frac{1}{2}(1 - y_{jk}) \max(0, \epsilon - D^2), \tag{3}$$

where $D = \|f(p_j) - f(p_k)\|_2$ is the Euclidean distance of two ℓ_2-norm normalized latent representations, $y_{jk} \in \{0, 1\}$ indicates whether p_j and p_k are the same pose, and ϵ is the minimum distance margin that pairs depicting different poses should satisfy.

Graph Convolution for Skeleton: For standard 2D convolution on natural images, the output feature maps can have the same size as the input feature maps with stride 1 and appropriate padding. Similarly, the graph convolution operation is designed to output graphs with the same number

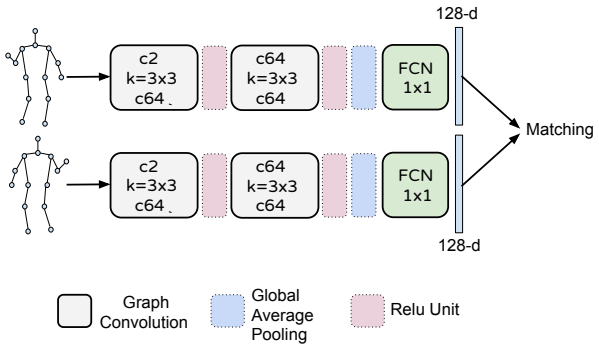

Figure 3. The Siamese graph convolution network for pose matching. We extract two feature vectors from the input graph pair with shared network weight. The feature vectors inherently encode the spatial interdependencies among the human joints.

of nodes. The dimensionality of attributes of these nodes, which is analogous to the number of feature map channels in standard convolution, may change after the graph convolution operation.

The standard convolution operation is defined as follows: given a convolution operator with the kernel size of $K \times K$, and an input feature map f_{in} with the number of channels c, the output value of a single channel at the spatial location $\mathbf{x}$ can be written as:

$$f_{out}(\mathbf{x}) = \sum_{h=1}^{K} \sum_{w=1}^{K} f_{in}(\mathbf{s}(\mathbf{x}, h, w)) \cdot \mathbf{w}(h, w), \quad (4)$$

where the **sampling function** $\mathbf{s} : Z^2 \times Z^2 \rightarrow Z^2$ enumerates the neighbors of location $\mathbf{x}$. The **weight function** $\mathbf{w} : Z^2 \rightarrow \mathbb{R}^c$ provides a weight vector in c-dimension real space for computing the inner product with the sampled input feature vectors of dimension c.

The convolution operation on graphs is defined by extending the above formulation to the cases where the input features map resides on a spatial graph V_t, *i.e.* the feature map $f_{in}^t : V_t \rightarrow R^c$ has a vector on each node of the graph. The next step of the extension is to re-define the sampling function $\mathbf{p}$ and the weight function $\mathbf{w}$. We follow the method proposed in [26]. For each node, only its adjacent nodes are sampled. The neighbor set for node v_i is $B(v_i) = \{v_j | d(v_j, v_i) \leq 1\}$. The sampling function $\mathbf{p} : B(v_i) \rightarrow V$ can be written as $\mathbf{p}(v_i, v_j) = v_j$. In this way, the number of adjacent nodes is not fixed, nor is the weighting order. In order to have a fixed number of samples and a fixed order of weighting them, we label the neighbor nodes around the root node with fixed number of partitions, and then weight these nodes based on their partition class. The specific partitioning method is illustrated in Fig. 4.

Therefore, Eq. (4) for graph convolution is re-written as:

$$f_{out}(v_i) = \sum_{v_j \in B(v_i)} \frac{1}{Z_i(v_j)} f_{in}(\mathbf{p}(v_i, v_j)) \cdot \mathbf{w}(v_i, v_j), \quad (5)$$

where the normalization term $Z_i(v_j) = |\{v_k | l_i(v_k) = l_i(v_j)\}|$ is to balance the contributions of different subsets

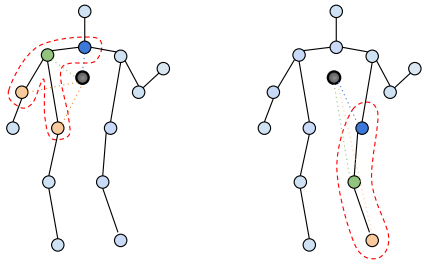

Figure 4. The spatial configuration partitioning strategy proposed in [26] for graph sampling and weighting to construct graph convolution operations. The nodes are labeled according to their distances to the skeleton gravity center (black circle) compared with that of the root node (green). Centripetal nodes have shorter distances (blue), while centrifugal nodes have longer distances (yellow) than the root node.

to the output. According to the partition method mentioned above, we have:

$$l_i(v_j) = \begin{cases} 0 & \text{if } r_j = r_i \\ 1 & \text{if } r_j < r_i \\ 2 & \text{if } r_j > r_i \end{cases} \quad (6)$$

where r_i is the average distance from gravity center to joint i over all frames in the training set.

4. Experiments

In this section, we present quantitative results of our experiments. Some qualitative results are shown in Fig. 5.

4.1. Dataset

PoseTrack [2] is a large-scale benchmark for human pose estimation and articulated tracking in videos. It provides publicly available training and validation sets as well as an evaluation server for benchmarking on a held-out test set. The benchmark is a basis for the challenge competitions at ICCV'17 [28] and ECCV'18 [29] workshops. The dataset consisted of over $68,000$ frames for the ICCV'17 challenge and is extended to twice as many frames for the ECCV'18 challenge. It now includes 593 training videos, 74 validation videos and 375 testing videos. For held-out test set, at most four submissions per task can be made for the same approach. Evaluation on validation set has no submission limit. Therefore, ablation studies in Section 4.4 are performed on the validation set. Since PoseTrack'18 test set is not open yet, we compare our results with other approaches in Section 4.5 on PoseTrack'17 test set.

4.2. Evaluation Metrics

The evaluation includes pose estimation accuracy and pose tracking accuracy. Pose estimation accuracy is evaluated using the standard **mAP** metric, whereas the evaluation of pose tracking is according to the **clear MOT** [30] metrics that are the standard for evaluation of multi-target tracking.

Figure 5. Qualitative evaluation results. Each person is visualized with a different color. Same color indicates identical IDs.

4.3. Implementation Details

To hightlight the generality of our framework, we adopt out-of-the-shelf object detectors trained with ImageNet and COCO datasets. Specifically, we use pre-trained models from deformable ConvNets [31]. We conduct experiments on validation sets to choose the object detector with better recall rates. We compare the deformable convolution versions of the R-FCN network [32] and of the FPN network [33], both with ResNet101 backbone [34]. The FPN feature extractor is attached to the Fast R-CNN [35] head for detection. We also compare the detection results with the ground truth based on the precision and recall rate on PoseTrack'17 validation set. In order to eliminate redundant candidates, we drop candidates with lower likelihood. Table 2 shows the precision and recall of the detectors given various drop thresholds. Since the FPN network performs better, we choose it as our detector. During training, we infer ground truth bounding boxes of candidates from the annotated keypoints, because in PoseTrack'17 dataset, the bounding box positions are not provided in the annotations. Specifically, we locate a bounding box from the minimum and maximum coordinates of the 15 keypoints, and then enlarge this box by 20% both horizontally and vertically.

For the single-person human pose estimator, we adopt CPN101 [36] and MSRA152 [21] with slight modifications. We first train the networks with the merged dataset of PoseTrack'17 and COCO for 260 epochs. Then we finetune the network solely on PoseTrack'17 for 40 epochs in order to mitigate the inaccurate regression on head and neck. For COCO, bottom-head and top-head positions are not given.

-	Train	Validation
Positive Pairs	56908	9731
Hard Negative Pairs	25064	7020
Other Negative Pairs	241450	91228

Table 1. Pose pairs collected from PoseTrack'18 dataset.

We infer these keypoints by interpolation on the annotated keypoints. We find that by finetuning on the PoseTrack dataset, the prediction on head keypoints will be refined. During finetuning, we use the technique of online hard keypoint mining, only focusing on losses from the 7 hardest keypoints out of the total 15 keypoints. Pose inference is performed online with single thread.

For the pose matching module, we train a Siamese graph convolutional network with 2 GCN layers and 1 convolutional layer using contrastive loss. We take normalized keypoint coordinates as input; the output is a 128 dimensional feature vector. Following [26], we use spatial configuration partitioning as the sampling method for graph convolution and use learnable edge importance weighting. To train the Siamese network, we generate training data from the PoseTrack dataset. Specifically, we extract people with same IDs within adjacent frames as positive pairs, and extract people with different IDs within the same frame and across frames as negative pairs. Hard negative pairs only include spatially overlapped poses. The number of collected pairs are illustrated in Table 1. We train the model with batch size of 32 for a total of 200 epochs with SGD optimizer. Initial learning rate is set to 0.001 and is decayed by 0.1 at epochs of $40, 60, 80, 100$. Weight decay is 10^{-4}.

4.4. Ablation Study

We conducted a series of ablation studies to analyze the contribution of each component on the overall performance.

-	Method / Thresh	0.1	0.2	0.3	**0.4**	0.5
Prec	Deformable FPN	17.9	27.5	32.2	**34.2**	35.7
	Deformable R-FCN	15.4	21.1	25.9	30.3	34.5
Recall	Deformable FPN	87.7	86.0	84.5	**83.0**	80.8
	Deformable R-FCN	87.7	86.5	85.0	82.6	80.1

Table 2. Comparison of detectors: Precision-Recall on PoseTrack 2017 validation set. A bounding box is correct if its IoU with GT is above certain threshold, which is set to 0.4 for all experiments.

-	**Estimation (mAP)**			**Tracking (MOTA)**		
Method	Wri	Ankl	Total	Wri	Ankl	Total
GT Detections	74.7	75.4	**81.7**	56.3	56.2	**67.0**
Deformable FPN-101	70.2	64.7	**74.6**	54.6	48.7	**61.3**
Deformable RFCN-101	69.0	64.3	**73.7**	52.2	47.4	**59.0**

Table 3. Comparison of offline pose tracking results using various detectors on PoseTrack'17 validation set.

Detectors: We experiment with several detectors and decide to use Deformable ConvNets with ResNet101 as backbone, *Feature Pyramid Networks* (FPN) for feature extraction, and fast R-CNN scheme as detection head. As shown in Table 2, this detector outperforms Deformable R-FCN with the same backbone. It is no surprise that better detectors result in improved performance on both pose estimation and pose tracking tasks, as shown in Table 3.

Offline vs. Online: We study the effect of keyframe intervals of our online method and compare with the offline method. For fair comparison, we use identical human candidate detector and pose estimator for both methods. For the offline method, we pre-compute human candidate detection and estimate the pose for each candidate, then we adopt a flow-based pose tracker [19], where pose flows are prebuilt by associating keypoints that indicate the same person across frames. For online method, we perform truly online pose tracking. Since human detection is performed only at keyframes, the online performance varies with different intervals. In Table 4, we illustrate the performance of the offline method, compared with the online method that is given various keyframe intervals. Online methods perform competitively with offline methods. The upper-bound of detections (DET) at keyframes is achieved with ground truth (GT) detections. As expected, the performance is positively correlated with the keyframe frequency. Note that the online methods only use spatial consistency for data association at keyframes. We report ablation experiments on the pose matching module in the following text.

GCN vs. Spatial Consistency (SC): Next, we report results when pose matching is performed during the data association stage, compared with only employing spatial consistency. It can be shown in Table 5 that the tracking performance improves with GCN-based pose matching. However, in some situations, different people may have near-duplicate poses, as shown in Fig. 6. To mitigate such ambiguities, spatial consistency is considered prior to pose similarity.

-	**Estimation (mAP)**			**Tracking (MOTA)**		
Method	Wri	Ankl	Total	Wri	Ankl	Total
Offline-CPN101	72.6	68.9	**76.4**	56.1	55.3	**62.4**
Offline-MSRA152	73.6	70.5	**77.3**	58.5	58.5	**64.9**
Online-DET-CPN101-8F	70.5	68.3	**74.0**	52.4	50.3	**58.1**
Online-DET-CPN101-5F	71.7	68.9	**75.1**	53.3	51.0	**59.0**
Online-DET-CPN101-2F	72.4	69.1	**76.0**	54.2	51.5	**60.0**
Online-DET-MSRA152-8F	71.1	69.5	**75.0**	54.6	54.6	**61.0**
Online-DET-MSRA152-5F	72.1	70.4	**76.1**	55.2	55.5	**61.9**
Online-DET-MSRA152-2F	73.3	70.9	**77.2**	56.5	56.6	**63.3**

Table 4. Comparison of offline and online pose tracking results with various keyframe intervals on PoseTrack'18 validation set.

Method	**Detect**	**Keyframe**	**MOTA**	
			CPN101	MSRA152
SC	GT	8F	68.2	72.0
SC+GCN			68.9	72.6
SC		5F	68.7	73.0
SC+GCN			69.2	73.5
SC		2F	72.0	76.7
SC+GCN			73.5	78.0
SC	DET	8F	58.1	61.0
SC+GCN			59.0	62.1
SC		5F	59.0	61.9
SC+GCN			60.1	63.1
SC		2F	60.0	63.3
SC+GCN			61.3	64.6

Table 5. Performance comparison of LightTrack with GCN and SC on PoseTrack'18 validation set.

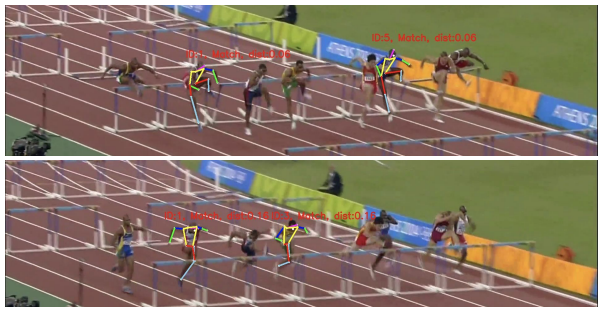

Figure 6. In some situations, different people indeed have very similar poses. Therefore, spatial consistency is considered first.

GCN vs. Euclidean Distance (ED): We studied whether the GCN network outperforms a naive pose matching scheme. With same normalization on the keypoints, ED as the dissimilarity metric for pose matching renders 85% accuracy on validation pairs generated from PoseTrack dataset, while GCN renders 92% accuracy. We validate on positive pairs and hard negative pairs.

4.5. Performance Comparison

Since PoseTrack'18 test set is not open yet, we compare our methods with other approaches, both online and offline, on PoseTrack'17 test set. For fair comparison, we only use PoseTrack'17 training set and COCO train+val set to train the pose estimators. No auxiliary data is used. We performed ablation studies on validation sets with CPN-101 [36] as the pose estimator. During testing, in addition to CPN-101, we conduct experiments using MSRA-152 [21].

	Method	Wrist-AP	Ankles-AP	mAP	MOTA	fps
	Posetrack 2017 Test Set					
Offline	PoseTrack, CVPR'18 [2]	54.3	49.2	59.4	48.4	-
Offline	BUTD, ICCV'17 [37]	52.9	42.6	59.1	50.6	-
Offline	Detect-and-track, CVPR'18 [18]	-	-	59.6	51.8	-
Offline	Flowtrack-152, ECCV'18 [21]	71.5	65.7	74.6	57.8	-
Offline	HRNet, CVPR'19[20]	72.0	67.0	74.9	57.9	-
Offline	**Ours-CPN101 (offline)**	**68.0 / 59.7**	**62.6 / 56.3**	**70.7 / 63.9**	**55.1**	-
Offline	**Ours-MSRA152 (offline)**	**68.9 / 61.8**	**63.2 / 58.4**	**71.5 / 65.7**	**57.0**	-
Offline	**Ours-manifold (offline)**	**- / 64.6**	**- / 58.4**	**- / 66.7**	**58.0**	-
Online	PoseFlow, BMVC'18 [19]	59.0	57.9	63.0	51.0	10*
Online	JointFlow, BMVC'18 [38]	53.1	50.4	63.3	53.1	0.2
Online	STAF, CVPR'19 [17]	65.0	60.0	70.3	53.8	2
Online	**Ours-CPN101-LightTrack-3F**	**61.2**	**57.6**	**63.8**	**52.3**	**47* / 0.8**
Online	**Ours-MSRA152-LightTrack-3F**	**63.8**	**59.1**	**66.5**	**55.1**	**48* / 0.7**
	Posetrack 2018 Validation Set					
	Ours-CPN101 (offline)	72.6 / 63.9	68.9 / 62.6	76.4 / 69.7	62.4	-
	Ours-MSRA152 (offline)	73.6 / 65.6	70.5 / 64.9	77.3 / 71.2	64.9	-
	Ours-YoloMD-LightTrack-2F	62.9 / 56.2	57.8 / 53.3	70.4 / 66.0	55.7	59* / 1.9
	Ours-CPN101-LightTrack-2F	72.4 / 66.3	69.1 / 64.2	76.0 / 70.3	61.3	47* / 0.8
	Ours-MSRA152-LightTrack-2F	73.3 / 66.4	70.9 / 66.1	77.2 / 72.4	64.6	48* / 0.7

Table 6. Performance comparison on Posetrack dataset. The last column shows the speed in frames per second (* means excluding pose inference time). For our online methods, mAP are provided after keypoints dropping. For offline methods, mAP are provided both before (left) and after (right) keypoints dropping.

Accuracy: As shown in Table 6, LightTrack outperforms other online methods while maintaining higher frame rates among top-down approaches, and is competitive with offline state-of-the-art methods. For the offline approach, we use same detector and pose estimators, except that we replace LightTrack with the official release of PoseFlow [19] for performance comparison. Although this algorithm is conceptually online, the actual process is performed in multiple stages, and requires full-image keypoint extraction and matching to be pre-computed between all adjacent frames, which is computationally expensive (time not reflected in Table 6). In contrast, LightTrack is truly processed online.

Speed: Testing on single Tesla P40 GPU, pose matching costs an average of 2.9 ms. Since pose matching only occurs at key-frames, its occurrence frequency depends on candidate number and keyframe intervals. Therefore, we test the average process time on PoseTrack'18 val set, which consists of 74 videos with a total of $8,857$ frames. It takes CPN101-LightTrack $11,638$ seconds to process, $11,450$ secs of which spent on pose estimation. The frame rate of the whole system is 0.76 fps. Excluding pose inference time, the framework runs at 47.11 fps. In total, $57,928$ persons are encountered. An average of 6.54 people are tracked per frame. It takes CPN101 140 ms to process each candidate, including 109 ms pose inference and 31 ms for pre-processing and post-processing. There is potential room to improve the frame rate and tracking performance with other choices of pose estimators and parallel inference optimization (20+ fps camera demo available). We see an improved performance with MSRA152-LightTrack but slightly slower frame rate due to its 133 ms inference time.

4.6. Discussion

Accuracy: Since the components in our framework are easily replaceable and extendable, methods employing this framework can potentially become faster, more accurate, or possibly both. Note that the pose estimator mentioned in Section 4.3 can be replaced by a more accurate [39] or a much faster counterpart. The performance boost in the general object detector, or methods that focus on detecting people (*e.g.*, using auxiliary dataset [40]), should also improve the pose tracking performance. Ablation study in Section 4.4 has shown that better detection increases the MOTA score, regardless of which detectors to use.

Speed: The pose estimation network can be prioritized for speed while sacrificing some accuracy. For instance, we use YOLOv3 and MobileNetv1-deconv (YoloMD) as detector and pose estimator, respectively. It achieves an average of 2 FPS with 70.4 mAP and MOTA score 55.7% on PoseTrack'18 validation set. Aside from network structure design, a faster network could also refine heatmaps from previous frame(s). Recently, refinement-based networks [41, 42] have drawn enormous attention.

Flexibility: The advantage of our top-down approach in pose tracking is that we can conveniently track specific targets and do not necessarily track all candidates. It can be achieved simply by choosing the target(s) at the first frame and providing target locations at key-frames. As a side effect, this further reduces computational complexity. If the target has specific visual appearance, the framework can be conveniently extended to ensure only the target can be matched at key-frames and tracked at remaining frames.

5. Conclusions

In this paper, we propose an simple yet effective framework for online top-down human pose tracking. We also provide a baseline employing this framework, and propose a Siamese graph convolution network for human pose matching as a Re-ID module in our pose tracking system. The skeleton-based representation effectively captures human pose similarity and is computationally inexpensive. Our method outperforms other online methods, stays competitive with offline state-of-the-art methods with higher frame rates. We believe the proposed framework is worthy to be widely used due to its superior performance, generality, and extensibility.

References

[1] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *CVPR*, 2017. 1, 2

[2] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *CVPR*, 2018. 1, 2, 5, 8

[3] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "Arttrack:articulated multiperson tracking in the wild," in *CVPR*, 2017. 1, 2

[4] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016. 2

[5] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016. 2

[6] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *ICCV*, 2017. 2

[7] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," *ECCV*, 2018. 2

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

[9] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *BMVC*, 2010. 2

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 2

[11] Y.-W. T. Hao-Shu Fang, Shuqin Xie and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017. 2

[12] 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. 2

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017. 2

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

[15] F. Xia, P. Wang, X. Chen, and A. L. Yuille, "Joint multi-person pose estimation and semantic part segmentation.," in *CVPR*, 2017. 2

[16] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *NIPS*, 2017. 2

[17] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh, "Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields," in *CVPR*, 2019. 2, 8

[18] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *CVPR*, 2018. 2, 8

[19] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *BMVC*, 2018. 2, 7, 8

[20] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *CVPR*, 2019. 2, 8

[21] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," *ECCV*, 2018. 2, 6, 8

[22] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *CVPR*, 2019. 3

[23] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018. 3

[24] K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *CVPR*, 2018. 3

[25] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *ECCV*, 2018. 3

[26] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018. 3, 4, 5, 6

[27] Z. Liu, B. Pan, Y. Xiu, and C. Lu, "Posehd: Boosting human detectors using human pose information," in *AAAI*, 2018. 3

[28] "Posetrack challenge 2017." https://posetrack.net/workshops/iccv2017/. Accessed: 2019-02-10. 5

[29] "Posetrack challenge 2018." https://posetrack.net/workshops/eccv2018/. Accessed: 2019-02-10. 5

[30] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008. 5

[31] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, p. 3, 2017. 6

[32] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016. 6

[33] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection.," in *CVPR*, 2017. 6

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 6

[35] R. Girshick, "Fast r-cnn," in *ICCV*, 2015. 6

[36] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded Pyramid Network for Multi-Person Pose Estimation," in *CVPR*, 2018. 6, 8

[37] S. Jin, X. Ma, Z. Han, Y. Wu, W. Yang, W. Liu, C. Qian, and W. Ouyang, "Towards multi-person pose tracking: Bottom-up and top-down methods," in *ICCV Workshop*, 2017. 8

[38] A. Doering, U. Iqbal, and J. Gall, "Joint flow: Temporal flow fields for multi person tracking," *BMVC*, 2018. 8

[39] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *arXiv preprint arXiv:1901.00148*, 2019. 8

[40] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," *CVPR*, 2019. 8

[41] G. Moon, J. Y. Chang, and K. M. Lee, "Posefix: Model-agnostic general human pose refinement network," *CVPR*, 2019. 8

[42] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, "Learning to refine human pose estimation," in *CVPR Workshops*, 2018. 8