Combining detection and tracking for human pose estimation in videos

Manchen Wang, Joseph Tighe, Davide Modolo
AWS Rekognition
manchenw,tighej,dmodolo@amazon.com

Abstract

We propose a novel top-down approach that tackles the problem of multi-person human pose estimation and tracking in videos. In contrast to existing top-down approaches, our method is not limited by the performance of its person detector and can predict the poses of person instances not localized. It achieves this capability by propagating known person locations forward and backward in time and searching for poses in those regions. Our approach consists of three components: (i) a Clip Tracking Network that performs body joint detection and tracking simultaneously on small video clips; (ii) a Video Tracking Pipeline that merges the fixed-length tracklets produced by the Clip Tracking Network to arbitrary length tracks; and (iii) a Spatial-Temporal Merging procedure that refines the joint locations based on spatial and temporal smoothing terms. Thanks to the precision of our Clip Tracking Network and our merging procedure, our approach produces very accurate joint predictions and can fix common mistakes on hard scenarios like heavily entangled people. Our approach achieves state-of-the-art results on both joint detection and tracking, on both the PoseTrack 2017 and 2018 datasets, and against all top-down and bottom-down approaches.

1. Introduction

Multi-person human pose tracking is the dual-task of detecting the body joints of all the people in all video frames and linking them correctly over time. The ability to detect body joints has improved considerably in the last several years [4, 5, 7, 14, 16, 21, 22, 24, 30, 33] thanks in part to the availability of large scale public image datasets like MPII [4] and MS COCO [19]. These approaches can be mostly classified into two categories, depending on how they operate: bottom-up approaches [4, 5, 16, 21, 24, 33] first detect individual body joints and then group them into people; while top-down approaches [7, 14, 30] first detect every person in an image and then predict each person's body joints within their bounding box location.

Largely thanks to advancements in object class detection [9, 14, 28], top-down approaches [30] have achieved better pose estimation performance on images than bottom-up methods. By taking advantage of robust person detectors, these approaches can focus on the task of joint detection within bounding box regions, and not have to deal with large scale variations and the problem of grouping joints into people that bottom-up methods do. Despite these positive results on image datasets, top-down methods do not perform as well on videos and were recently outperformed by a bottom-up approach [25]. We attribute this to the fact that detecting people bounding boxes in videos is a much harder task than on images. While images often capture people "posing", videos inherently contain atypical types of occlusion, viewpoints, motion blur and poses that make object detectors occasionally fail (e.g., in fig. 1a, the detector is not able to localize the highly occluded person instances in the first two frames).

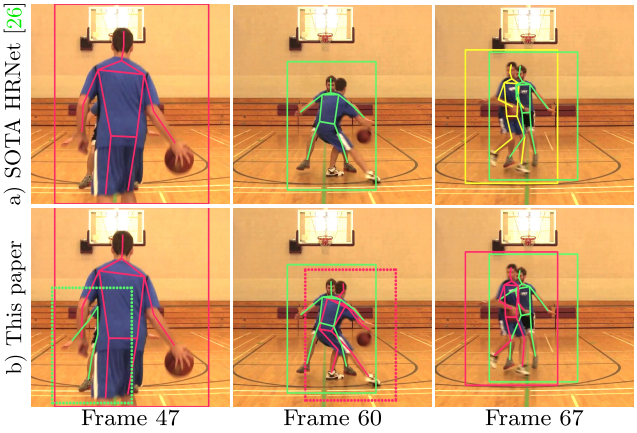


Figure 1: *Top-down approaches like HRNet rely heavily on the performance of their person detector, which sometimes fails on highly occluded people (frames 47, 60), and occasionally make mistakes on highly entangled people (frame 67). Our approach overcomes these limitations by propagating bounding boxes over time (drawn with dotted lines) and by predicting multiple pose hypothesis for each person and smartly selecting the best one.*

We propose a novel top-down approach that overcomes these problems and enables us to reap the benefits of top down methods for multi-person pose estimation in videos. We detect person bounding boxes on each frame and then

propagate these to their neighbours. Our intuition is that if a person is present at a specific location in a frame, they should still be at approximately that location in the neighbouring frames, even when the detector fails to find them. In detail, given a localized person bounding box, we crop a spatial-temporal tube from the video centered at that frame and location. Then, we feed this tube to a novel *Clip Tracking Network* that estimates the locations of all the body joints of that person in all the frames of the tube. To solve this task, our Clip Tracking Network performs body joint detection and tracking simultaneously. This has two benefits: (i) by solving these tasks jointly, our network can better deal with unique poses and occlusions, and (ii) it can compensate for missed detections by predicting joints in all frames of the spatial-temporal tube, even for frames where the person was not detected. To construct this Clip Tracking Network, we extend the state-of-the-art High-Resolution Network (HRNet) [30] architecture to the task of tracking, using 3D convolutions that are carefully designed to help learn the temporal correspondence between joints.

The Clip Tracking Network operates on fixed length video clips and produces multi-person pose tracklets. We combine these tracklets into pose tracks for arbitrary length videos in our *Video Tracking pipeline*, by first generating temporally overlapping tracklets and then associating and merging the pose detections in frames where the tracklets overlap. When merging tracklets into tracks, we use the multiple pose detections in each frame in a novel consensus-based *Spatio-temporal merging* procedure to estimate the optimal location of each joint. This procedure favours hypotheses that are spatially close to each other and that are temporally smooth. This combination is able to correct mistakes on highly entangled people, leading to more accurate predictions, as in frame 67 of fig. 1: while [30] wrongly selects the yellow player's left knee as the prediction for the green player's right knee (1a), our procedure is able to correct this mistake and predict the correct location (1b).

When compared to the literature, our approach achieves state-of-the-art results for both body joint detection and tracking, on the PoseTrack 2017 and 2018 video datasets [3], not only against top-down approaches, but also against bottom-up ones. The improvement is consistent and often significant; for example, error on body joint detection reduces by 28% PoseTrack 2017 and error on body joint tracking by 9% on PoseTrack 2018. Furthermore, we also present an extensive ablation study of our approach, where we validate its components and our hyperparameter choices.

The rest of the paper is organized as follows: in sec. 2 we present our related work; then, in sec. 3 we present our three contributions: (i) our novel clip tracking network (sec. 3.1), (ii) our tracking pipeline (sec. 3.2) and (iii) our spatial-temporal merging procedure (sec. 3.3). Finally, we present our experiments in sec. 4 and conclude in sec. 5.

2. Related Work

2.1. Human pose estimation in images

Recent human pose estimation methods can be classified into bottom-up and top-down approaches, depending on how they operate. *Bottom-up approaches* [5, 16, 21, 24] first detect individual body joints and then group them into people. On the other hand, *top-down approaches* [7, 14, 23, 30], first detect people bounding boxes and then predict their joint locations within each region. Top-down approaches have the advantage of not needing any joint grouping and because the input images they operate on are crops from detectors, they do not have to be robust to large scale variations. However, top-down approaches suffer from the limitations of the person detector: when it fails (i.e., a person is not localized), the joints on that person cannot be recovered. Bottom-up approaches do not have this reliance on a detector and they can predict any joint; however they suffer from the difficult tasks of joint detection across large scale variations and joints grouping. In this work we try to take the best of both words and propose a novel top-down approach for videos that recovers from the detector's misses by exploring and propagating information temporally.

We build upon the HRNet of Sun et al. [30]. This was originally proposed for human pose estimation, achieving state-of-the-art results in images. Recently, it was then modified to achieve state-of-the-art results on other vision tasks, like object detection [31] and semantic segmentation [32]. In this paper we show how to extend HRNet to human pose estimation and tracking in videos.

2.2. Human pose estimation and tracking in videos

Given the image approaches just introduced, it is natural to extend them to multi-person pose tracking in videos by running them on each frame independently and then linking these predictions over time. Along these lines, *bottom-up methods* [17, 25] build spatial-temporal graphs between the detected joints. Raaj et al. [25] did so by extending the spatial Affinity Field image work of Cao et al. [5] to Spatio-Temporal Affinity Fields (STAF), while Jin et al. [17] extended the spatial Associative Embedding image work of Newell et al. [21] to Spatio-Temporal Embedding.

On the other hand, *top-down methods* [13, 34] build temporal graphs between person bounding boxes, which are usually simpler to solve. SimpleBaseline [34] first run a person detector on each frame independently and then linked its detections in a graph, where the temporal similarity was defined using expensive optical flow. Detect-and-Track [13] instead used a 3D Mask R-CNN approach to detect the joints of a person in a short video clip and then used a lightweight tracker to link consecutive clips together by comparing the location of the detected bounding boxes. Like [13], our approach also runs inference on short clips in

a single forward pass, but it brings many advantages over it: (i) as most top-down approaches, [13] is limited by its detector's accuracy and it cannot recover from its misses; instead, we propose to propagate detected bounding boxes to neighbouring frames and look for missed people in those regions. (ii) [13] runs on non-overlapping clips and performs tracking based on person bounding boxes only; instead, we run on overlapping clips and use multiple joint hypothesis in a novel tracking system, that leads to more accurate predictions. (iii) [13] employs fully 3D convolutional networks, while we show that 3D filters on only part of a network is already sufficient to teach the network to track.

3. Methodology

At a high level, our method works by first detecting all candidate persons in the center frame of each video clip (i.e. the *keyframe*) and then estimating their poses forward and backward in time. Then, it merges poses from different clips in time and space, producing any arbitrary length tracks. More in details, our approach consist of three major components: *Cut, Sew* and *Polish*. Given a video, we first cut it into overlapping *clips* and then run a person detector on their keyframes. For each person bounding box detected in a keyframe, a spatial-temporal tube is *cut* out at the bounding box location over the corresponding clip. Given this tube as input, our **Clip Tracking Network** both estimates the pose of the central person in the keyframe, and tracks his pose across the whole video clip (sec. 3.1, fig. 2). We call these *tracklets*. Next, our **Video Tracking Pipeline** works as a tailor to *sew* these tracklets together based on poses in overlapping frames (sec. 3.2, fig. 3). We call these multiple poses for the same person in same frame *hypotheses*. Finally, **Spatial-Temporal merging** *polishes* these predictions using these hypotheses in an optimization algorithm that selects the more spatially and temporally consistent location for each joint (sec. 3.3, fig. 4). In the next three sections we present these three components in details.

3.1. Clip Tracking Network

Our Clip Tracking Network performs both pose estimation and tracking simultaneously, on a short video clip. Its architecture builds upon the successful HRNet architecture of Sun et al. [30]. In the next paragraph we summarize the original HRNet design and in the following one we explain how to extend it to tracking.

HRNet for human pose estimation in images. Given an image, this top-down approach runs a person detector on it, which outputs a list of axis-aligned bounding boxes, one for each localized person. Each of these boxes is independently cropped and fed into HRNet, which consists of four stages of four parallel subnetworks trained to localize all body joints of only the central person in the crop.

The output of HRNet is a set of heatmaps, one for each body joint. Each pixel of these heatmaps indicates the likelihood of "containing" a joint. As other approaches in the literature [5, 7, 14, 16, 21, 24], the network is trained using a mean squared error loss function, between the predicted heatmap H^{pred} and the ground-truth heatmap H^{gt}:

$$L = \frac{1}{KWH} \sum_{k}^{K} \sum_{i}^{W} \sum_{j}^{H} \left\| H_{ijk}^{pred} - H_{ijk}^{gt} \right\|_2^2, \qquad (1)$$

where K is the number of body joints (keypoints) and i, j the pixel coordinates. H^{gt} are generated by convolving a 2D Gaussian filter on the annotated location of each joint.

3D HRNet for video pose estimation and tracking. Our approach operates on short video clips: $\mathcal{C} = \{\mathcal{F}^{t-\delta}, ..., \mathcal{F}^{t}, ..., \mathcal{F}^{t+\delta}\}$. First, it runs a person detector on the center frame $\mathcal{F}^t$ and obtains a list of person bounding boxes $\mathcal{B}^t = \{\beta_1^t, ..., \beta_n^t\}$ (fig. 2a). Then, for each bounding box β_p^t, it creates a tube $\mathcal{T}_{\beta_p^t}$ by cropping the box region from all frames in the clip $\mathcal{C}$: $\mathcal{T}_{\beta_p^t} = \{\mathcal{F}_{\beta_p^t}^{t-\delta}, ..., \mathcal{F}_{\beta_p^t}^{t}, ..., \mathcal{F}_{\beta_p^t}^{t+\delta}\}$ (fig. 2b). Next, it feeds this tube to our video HRNet, which outputs a *tracklet* containing all the poses of person p in all the frames of the tube: $\mathcal{P}_{\beta_p^t} = \{\rho_{\beta_p^t}^{t-\delta}, ..., \rho_{\beta_p^t}^{t}, ..., \rho_{\beta_p^t}^{t+\delta}\}$ (fig. 2c). Importantly, all the poses in $\mathcal{P}_{\beta_p^t}$ need to belong to the same person, even when this becomes occluded or moves out of the tube frame (in which case the network should not output any prediction, even if other people are present). This is a difficult task, which requires the network to both learn to predict the location of the joints of the pose and track them through time.

In order to help the network tackle this challenge, we do two things: (i) to account for fast moving people, we enlarge each bounding box by 25% along both dimensions prior to creating a tube; and (ii) to allow the network to associate people between frames, we inflate the 2D convolutions in the first two stages of HRNet to 3D to help the network learn to track. Specifically, in the first stage we use $3\times1\times1$, $1\times3\times3$ and $1\times1\times1$ filters, while in the second stage we use $3\times3\times3$ filters. After this second stage the network has a receptive field that is temporally large enough to observe the whole tube, learn the person's appearance and his/her movements within it. Note how our method is similar in spirit to what Jin et al. [17] proposed with their temporal associative embedding, but it is learnt automatically by the network without the need of additional constraints. Finally, we train our video HRNet with the same mean squared loss of eq. 1, but now computed over all the frames in the clip $\mathcal{C}$:

$$L = \frac{1}{|\mathcal{C}|KWH} \sum_{f}^{|\mathcal{C}|} \sum_{k}^{K} \sum_{i}^{W} \sum_{j}^{H} \left\| H_{ijkf}^{pred} - H_{ijkf}^{gt} \right\|_2^2 \qquad (2)$$

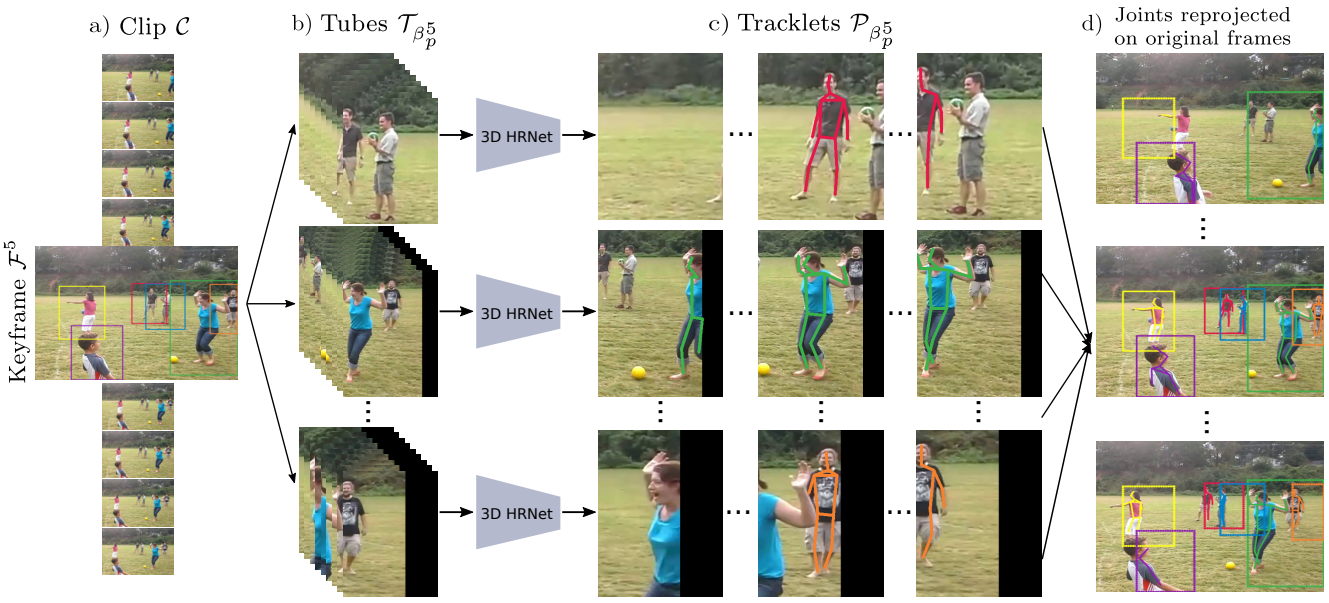


Figure 2: **Clip Tracking Network**. *First, (a) our approach runs a person detector on the keyframe of a short video clip. Then, (b) for each detected person it creates a tube by cropping the region within his/her bounding box from all the frames in the clip. Next, (c) each tube is independently fed into our Clip Tracking Network (3D HRNet), which outputs pose estimates for the same person (the one originally detected in the keyframe) in all the frames of the tube. Finally, (d) we reproject the predicted poses on the original images to show how the model can correctly predict poses in all the frames of the clip, by only detecting people in the keyframe.*

3.2. Video Tracking Pipeline

Our Clip Tracking Network outputs a tracklet $\mathcal{P}_{\beta_p^t}$ for each person p localized at β_p. However, p may exist beyond the length of $\mathcal{P}_{\beta_p^t}$ and the duty of our Video Tracking pipeline is to merge tracklets that belong to the same person, thus enabling pose estimation and tracking on any arbitrary length video (fig. 3). Our pipeline merges two fixed-length tracklets if their predicted poses on overlapping frames are similar (e.g., in fig. 3, $\mathcal{P}_{\beta_1^2}$ and $\mathcal{P}_{\beta_1^4}$ overlap on frames 2-4).

We generate these overlapping tracklets by running our Clip Tracking Network on clips of length $|\mathcal{C}|$ from keyframes sampled every S (*stepsize*) frames with $S < |\mathcal{C}|$.

We model the problem of merging tracklets that belong to the same person as a bipartite graph based energy minimization problem, which we solve using the Hungarian algorithm [18]. As a similarity function between two overlapping tracklets, we compute Object Keypoint Similarity (OKS) [19, 27] between their poses (reprojected on the original coordinate space, fig. 2d) on their overlapping frames. For example, in fig. 3 tracklets $\mathcal{P}_{\beta_3^6}$ and $\mathcal{P}_{\beta_1^8}$ are computed on tubes generated from keyframes 6 and 10 respectively and of length $|\mathcal{C}| = 5$. Under these settings, these tracklets both predict poses for frames 6, 7 and 8 and their similarity is computed as the average OKS over these three frames. On the other hand, tracklets $\mathcal{P}_{\beta_3^6}$ and $\mathcal{P}_{\beta_2^2}$ only overlap on frame 4 and as such their similarity is computed as the OKS on that single frame. Finally, we take the negative value of this OKS similarity for our minimization problem.

Note how this formulation is able to overcome the limitation that top-down approaches usually suffer from: missed bounding box detections. Thanks to our propagation of person detections from keyframes to their neighbouring frames (fig. 2b), we are able to obtain joints predictons even for those frames with missed detections. For example, in fig. 3 the person detector failed to localize the green person in keyframe 4, but by propagating the detections from keyframes 2 and 6 we are able to obtain a pose estimate for frame 4 as well. In addition, we are also able to link these correctly, thanks to the overlap between these two tracklets.

3.3. Spatial-Temporal merging of pose hypotheses

Our video tracking pipeline merges tracklets, but it does not deal with merging human poses. For example, in fig. 3 the approach correctly links all the yellow tracklets, but it does not address the question of what to do with the multiple pose estimates for frame 4 (i.e., $\rho^4_{\beta_1^2}$, $\rho^4_{\beta_1^4}$ and $\rho^4_{\beta_2^6}$). In this section we present our solution to this problem.

Given a set of merged, overlapping tracklets for person p, we define $\mathcal{H}_p^t = \{\rho^t_{\beta_p^{t-\delta}}, ..., \rho^t_{\beta_p^t}, ..., \rho^t_{\beta_p^{t+\delta}}\}$, as the *pose hypotheses* of p at time t. $\mathcal{H}_p^t$ represents the collection of poses for person p, generated by our Clip Tracking Network at time t by running on tube crops centered on different keyframes. The most straightforward procedure to obtain a single final pose for each person is to simply select, for each joint, the hypothesis $\mathcal{H}_p^t$ with the highest confidence score. We call this *Baseline Merge* and, as we show later in our

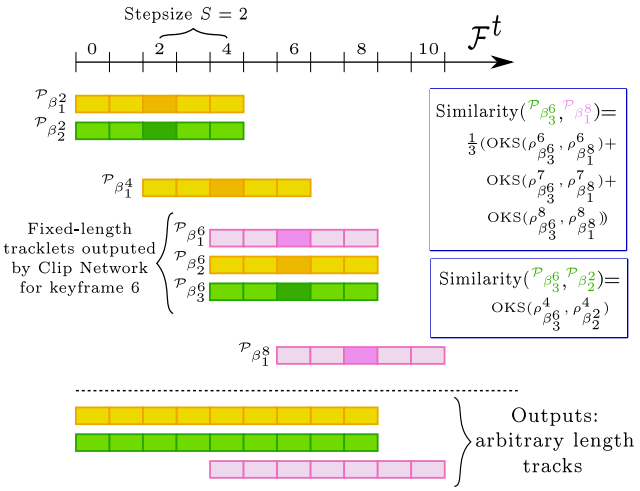


Figure 3: **Video Tracking Pipeline** *merges fixed-length tracklets into arbitrary length tracks by comparing the similarity of their detected poses in the frames the tracklets overlap on.*

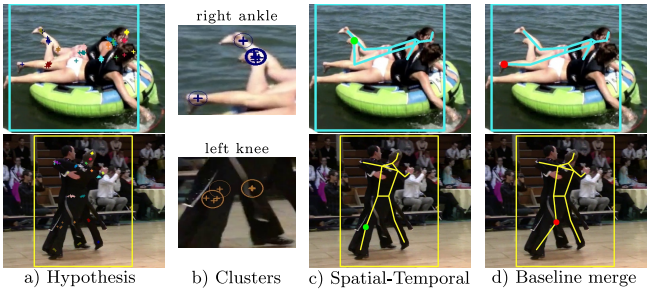


Figure 4: **Merging pose hypotheses.** *Our video tracking pipeline runs our Clip Tracking Network on multiple overlapping frames, producing multiple hypotheses for every joint of a person (a). We cluster these hypotheses (b) and solve a spatial-temporal optimization problem on these clusters to estimate the best location of each joint (c). This achieves better predictions than a simple baseline that always pick the hypothesis with the highest confidence score (d), especially on frames with highly entangled people.*

experiments, it achieves competitive performance, already highlighting the power of our Clip Tracking Network. Nevertheless, this procedure occasionally predicts the wrong location when the person of interest is entangled with or occluded by another person, as show in fig. 4d.

To overcome these limitations, we propose a novel method to merge these hypotheses (fig. 4b-c). Our intuition is that the optimal location for a joint should be the one that is both consistent across the multiple candidates within a frame (spatial constraint) and consistent over consecutive frames (temporal constraint). We model the problem of predicting the optimal location for each joint in each frame as a shortest path problem and we solve it using the Dijkstra's algorithm [10]. Instead of considering each joint detection as a node in the graph, we operate on clusters obtained by running a mean shift algorithm over joint hypotheses [8]. This clusters robustly smooth out noise in the individual hypotheses, while also reducing the graph size leading to faster optimization. As a similarity function ϕ between clusters c^t and c^{t+1} in consecutive frames, we compute a spatial-temporal weighting function that follows the aforementioned intuition: it favours clusters with more hypotheses and those that have smoother motion across time.

Formally,

$$\phi(c^t, c^{t+1}) = \underbrace{(|\mathcal{H}| - |c^t|) + (|\mathcal{H}| - |c^{t+1}|)}_{\textit{Spatial}} + \lambda \underbrace{\|\mu(c^t) - \mu(c^{t+1})\|_2^2}_{\textit{Temporal}}, \quad (3)$$

where $\mu(c^t)$, $\mu(c^{t+1})$ are the locations of the centers of the clusters, $|c^t|$, $|c^{t+1}|$ their magnitude and $|\mathcal{H}|$ the number of hypotheses. Finally, we balance these spatial and temporal constraints using λ.

4. Experiments

4.1. Datasets and Evaluation

We experiment with PoseTrack [3], which is a large-scale benchmark for human pose estimation and tracking in video. It contains challenging sequences of highly articulated people in dense crowds performing a wide range of activities. We experiment on both the 2017 and 2018 versions of this benchmark. **PoseTrack2017** contains 250 videos for training, 50 for validation and 214 for test. **PoseTrack2018** further increased the number of videos of the 2017 version to a total of 593 for training, 170 for validation and 375 for test. These datasets are annotated with 15 body joints, each one defined as a point and associated to a unique person id. Training videos are annotated with a single dense sequence of 30 frames, while validation videos also provide annotations for every forth frame, to enable the evaluation of longer range tracking.

We evaluate our models using the standard human pose estimation [19, 24, 27] and tracking [3, 20] metrics: joint detection performance is expressed in terms of **average precision (AP)**, while tracking performance in terms of **multi object tracking accuracy (MOTA)**. We compute these metrics independently on each body joint and then obtain our final performance by averaging over the joints. As done in the literature [13, 30, 34], when we evaluate on the validation sets of these datasets, we compute AP on all the localized body joints, but we threshold low confidence predictions prior to computing MOTA. For our experiments we learn a per-joint threshold on a hold out set of the training set. Moreover, we remove very short tracklets (< 5 frames) and tiny bounding boxes ($W * H < 3200$), as these often capture not annotated, small people in the background.

4.2. Implementation details

3D Video HRNet. Prior to inflating a 2D HRNet to our 3D version, we pre-train it for image pose estimation on the PoseTrack dataset (2017 or 2018, depending on what set we evaluate the models on). This step enables the network to learn the task of localizing body joints, so that during training on videos it can focus on learning to track. We inflate the first two stages of HRNet using "mean" initialization [6, 12, 13], which replicates the 2D filters and normalizes them accordingly. We use stepsize $S = 1$, as it produces the highest number of pose hypotheses, and clips of $|\mathcal{C}| = 9$ frames, so that the model can benefit from important temporal information. We use the same hyperameters of [30], but we train 3D HRNet for 20 epochs and decrease the learning rate two times after 10 and 15 epochs, respectively (1e-4 $\rightarrow$ 1e-5 $\rightarrow$ 1e-6). Finally, during inference we follow the procedure of [30, 34]: we run on both the original and the flipped image and average their heatmaps.

Person detector. We use a ResNet-101 SNIPER [28] detector to localize all the person instances. We train it on the MS COCO 2017 dataset [19] and achieve an AP of 57.9 on the "person" class on COCO *minival*, which is similar to that of other top-down approaches [34, 36].

Merging pose hypotheses. We follow the PoseTrack evaluation procedure to determine a good size estimate for our clusters. This procedure considers a prediction correctly, if the L_2 distance between that prediction and the closest ground truth is within a radius defined as 50% of the head size of the person. We use the same radius for our clusters. Moreover, we set $\lambda = 0.1$ to give equal importance to the spatial and temporal components, as the latter has approximately $10\times$ the magnitude of the former.

4.3. Comparisons with the state-of-the-art

We compare our approach with the state-of-the-art (SOTA) methods in the literature on body joints detection and tracking, on the validation sets of PoseTrack2017 (tables 1 and 2) and PoseTrack2018 (tables 3 and 4). Our approach achieves SOTA results on both metrics, on both datasets and against both top-down and bottom-up approaches. In some cases, the improvement over the SOTA is substantial: +6.5 mAP on PoseTrack2017 (which corresponds to 28% in error reduction), and +3.0 MOTA on PoseTrack2018 (9% in error reduction). When compared to only top-down approaches, which is the category this approach belongs to, the improvement in MOTA is even more significant, up to +6.2 on PoseTrack2017 (18% in error reduction) over the winner of the last PoseTrack challenge (FlowTrack, 65.4 vs 71.6), showing the importance of performing joint detection and tracking simultaneously.

Next, we evaluate our approach on the test sets of PoseTrack 2017 (table 5) and PoseTrack 2018 (table 6). The annotations for these sets are private and we obtained our results by submitting our predictions to the evaluation server [1]. Again, our approach achieves the best tracking results on both test sets (+3 MOTA) and on par to SOTA results on joint detection, even though our model is actually trained on less data than the competitors on PoseTrack2018.

	Method	Head	Sho	Elb	Wri	Hip	Kne	Ank	**Avg**
Bottom-up	JointFlow [11]	-	-	-	-	-	-	-	69.3
	TML++ [15]	-	-	-	-	-	-	-	71.5
	STAF [25]	-	-	-	65.0	-	-	62.7	72.6
	STEmbedding [17]	83.8	81.6	77.1	70.0	77.4	74.5	70.8	77.0
Top-down	Detect&Track [13]	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
	PoseFlow [35]	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
	FastPose [37]	80.0	80.3	69.5	59.1	71.4	67.5	59.4	70.3
	FlowTrack [34]	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.7
	HRNet [30]	82.1	83.6	80.4	73.3	75.5	75.3	68.5	77.3
	Our approach	**89.4**	**89.7**	**85.5**	**79.5**	**82.4**	**80.8**	**76.4**	**83.8**

Table 1: *Joint detection (***AP***) on* **PoseTrack2017 val**.

	Method	Head	Sho	Elb	Wri	Hip	Kne	Ank	**Avg**
Bottom-up	JointFlow [11]	-	-	-	-	-	-	-	59.8
	TML++ [15]	75.5	75.1	62.9	50.7	60.0	53.4	44.5	61.3
	STAF [25]	-	-	-	-	-	-	-	62.7
	STEmbedding [17]	78.7	79.2	71.2	61.1	**74.5**	**69.7**	**64.5**	**71.8**
Top-down	Detect&Track [13]	61.7	65.5	57.3	45.7	54.3	53.1	45.7	55.2
	PoseFlow [35]	59.8	67.0	59.8	51.6	60.0	58.4	50.5	58.3
	FastPose [37]	-	-	-	-	-	-	-	63.2
	FlowTrack [34]	73.9	75.9	63.7	56.1	65.5	65.1	53.5	65.4
	Our approach	**80.5**	**80.9**	**71.6**	**63.8**	70.1	68.2	62.0	**71.6**

Table 2: *Joint tracking (***MOTA***) on* **PoseTrack2017 val**.

	Method	Head	Sho	Elb	Wri	Hip	Kne	Ank	**Avg**
B-U	STAF [25]	-	-	-	64.7	-	-	62.0	70.4
	TML++ [15]	-	-	-	-	-	-	-	74.6
T-D	PT_CPN++ [36]	82.4	**88.8**	86.2	**79.4**	72.0	**80.6**	**76.2**	80.9
	Our approach	**84.9**	87.4	**84.8**	**79.2**	**77.6**	79.7	75.3	**81.5**

Table 3: *Joint detection (***AP***) on* **PoseTrack2018 val**.

	Method	Head	Sho	Elb	Wri	Hip	Kne	Ank	**Avg**
B-U	STAF [25]	-	-	-	-	-	-	-	60.9
	TML++ [15]	**76.0**	**76.9**	66.1	56.4	**65.1**	61.6	52.4	65.7
T-D	PT_CPN++ [36]	68.8	73.5	65.6	61.2	54.9	64.6	56.7	64.0
	Our approach	74.2	76.4	**71.2**	**64.1**	64.5	**65.8**	**61.9**	**68.7**

Table 4: *Joint tracking (***MOTA***) on* **PoseTrack2018 val**.

Method	Additional Data	wrists AP	ankles AP	**Total AP**	**Total MOTA**
JointFlow [11]	COCO	53.1	50.4	63.4	53.1
TML++ [15]	COCO	60.9	56.0	67.8	54.5
FlowTrack [34]	COCO	71.5	65.7	74.6	57.8
HRNet [30]	COCO	**72.0**	**67.0**	**75.0**	57.9
POINet [26]	COCO	69.5	67.2	72.5	58.4
KeyTrack [29]	COCO	71.9	65.0	74.0	61.2
Our approach	COCO	69.8	65.9	**74.1**	**64.1**

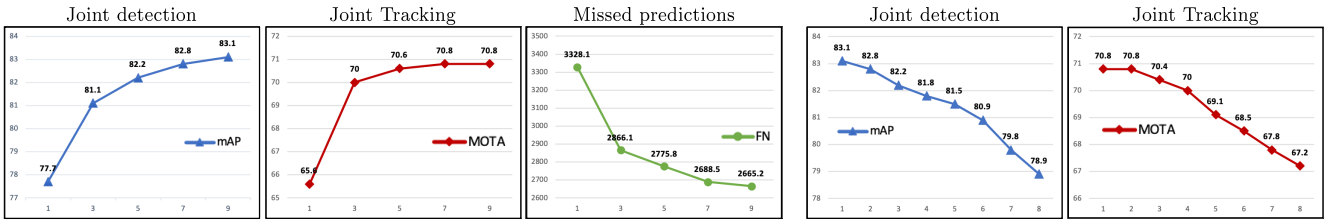
Table 5: *Results from the* **PoseTrack2017 test** *leaderboard [1].*

Method	Additional Data	wrists AP	ankles AP	**Total AP**	**Total MOTA**
TML++ [15]	COCO	60.2	56.8	67.8	54.9
PT_CPN++ [36]	COCO + Other	68.2	66.1	70.9	57.4
FlowTrack [34]	COCO + Other	**73.0**	**69.0**	**74.0**	61.4
Our approach	COCO	69.8	67.1	**73.5**	**64.3**

Table 6: *Results from the* **PoseTrack2018 test** *leaderboard [2].*

4.4. Analysis of our approach

We now analyze our approach and our hyper-parameter choices. For simplicity, we run our experiments only on the validation set of PoseTrack2017, using the settings described in sec. 4.2. Unless specified, we do not employ our spatial-temporal merging procedure (sec. 3.3) to keep our analysis transparent, as this corrects some mistakes.



a) Clip length $|\mathcal{C}|$ b) Stepsize S

Figure 5: *Results for different values of clips lengths $|\mathcal{C}|$ (a) and stepsize S.*

Backbone: HRNet	Linking	Spatial Merge (sec. 3.3)	Temporal Merge (sec. 3.3)	Detection mAP	Tracking MOTA
Base 2D	oks-gbm			77.7	65.6
Our 3D (sec. 3.1)	none			82.3	-
	sec. 3.2			83.1	70.8
	sec. 3.2	✓		83.5	71.4
	sec. 3.2		✓	83.4	71.1
	sec. 3.2	✓	✓	**83.8**	**71.6**

Table 7: *Ablation study on the components of our approach. In line 3, we test our Video Tracking pipeline paired with a Baseline merge that always selects the hypothesis with the highest score.*

Ablation study. Here we evaluate the different components of our approach and quantify how much each of them contributes to the model's final performance (table 7). First, we compare against a baseline 2D HRNet model [30] that runs on each frame independently. This baseline model achieves a mAP of 77.7; this is substantially lower compared to our most basic 3D HRNet (82.3 mAP), which does not perform any tracking and just uses OKS-based NMS over the hypotheses. This big improvement is due to our model being able to predict joints in frames where the person detector failed to localized the person.

When our 3D HRNet is paired with our video tracking pipeline (sec. 3.2) and the baseline merge, it improves MOTA considerably compared to the same 2D HRNet baseline paired with the popular OKS-based greedy bipartite matching (*oks-gbm*) algorithm that links pose predictions over time [13, 34]. Interestingly, this also improves mAP over our 3D HRNet with no tracking (+0.8 mAP). Finally, when we substitute the baseline merge with our procedure (sec. 3.3), the results further improve: both spatial and temporal merges are beneficial and complementary, bringing our full model performance to 83.8 mAP and 71.6 MOTA, almost a 10% improvement over the strong baseline.

Clip length $|\mathcal{C}|$. Our 3D HRNet operates on spatial-temporal tubes of length $|\mathcal{C}|$. In sec. 4.2, we set this value to 9, so that both our Clip Tracking Network and our Video Tracking pipeline can greatly benefit from rich temporal information. Here we examine how performance changes as we change this hyperparameter (fig. 5a). Setting $|\mathcal{C}| = 1$ is equivalent to running the baseline 2D HRNet presented in the previous section and it achieves the lowest performance among all variations. Interestingly, the largest improvement is brought by moving from 1 to 3, which indicates that little temporal information is already sufficient to compensate for many failures of the person detector. Further increasing $|\mathcal{C}|$ leads to a slow, but steady improvement in both mAP and MOTA, as the model can recover from even more mistakes. We quantitatively show this recovery in fig. 5a, where the number of false negatives decreases as $|\mathcal{C}|$ increases.

HRNet: 3D filters	None	Early (ours)	Last	All
mAP	77.7	**81.1**	80.6	79.3
MOTA	65.6	**70.0**	69.2	68.0

Table 8: *Results from different HRNet architecture as Clip Tracking Network, which differ in where they have 3D temporal filters.*

Step size S. In sec. 4.2, we set this to 1, so that our approach can use every frame of a video as keyframe and collect the largest set of pose hypotheses. This procedure, however, may be too expensive for some applications and here we evaluate how the performance changes as we improve the runtime by reducing the number of keyframes (i.e., increase the stepsize). Increasing the value of S leads to a linear speed up by a factor S, as the two most expensive components of our approach (person detector and 3D HRNet) now run only every S frames. As expected, results (fig. 5b) for both joint detection and tracking decrease as we increase S, as the model looses its temporal benefits. Nevertheless, they decreases slowly and even when we run our fastest inference with the largest step size, the model still achieves competitive performance (mAP 78.9 and MOTA 67.2), on par with that of many state-of-the-art models (table 1). Furthermore, note out how these results are better than those of our baseline 2D HRNet (mAP 77.7 and MOTA 65.6, fig. 5a, $|\mathcal{C}| = 1$), yet this 3D model is effectively faster, as it runs its person detector only once every 8 frames, as opposed to all frames, as done by 2D HRNet.

Network design. Our 3D HRNet architecture uses 3D convolutions in its early 2 stages (sec. 3.1), as these are the best suited to learn the low-level correspondence needed to correctly link the joints of the same person within a tube. In this section we evaluate different network designs: our design (Early), a 3D HRNet architecture with 3D filters in its last stage (Last), which learn to smooth joint predictions over small temporal windows, and a fully 3D HRNet architecture (All), that balances learning good temporal corre-

Figure 6: *Visualization of the output of our approach on five videos from the PoseTrack dataset. Bounding boxes and poses are color coded using the track id predicted by our model. Solid bounding boxes indicate that the instance was localized by the person detector, while dotted bounding boxes were originally missed by the detector, but recovered by our approach.*

spondences and spatially smooth joint predictions. As training a full 3D HRNet requires a considerable amount of GPU memory, we experiment here with a lightweight setup with $|\mathcal{C}| = 3$. Results are presented in table 8. For reference, we report the mAP performance of a standard 2D HRnet without any 3D filter. Adding 3D filters, no matter the location, always improves over the simple 2D architecture. Among the different choices, "Early" achieves the best performance for both detection and tracking, validating our design.

Dependency on person detector. Like all top-down methods, our approach is also limited by the accuracy of the employed person detector. However, we believe that our approach is significantly less sensitive than others in the literature, as it can recover missed predictions using its temporal reasoning. To validate this, we evaluate how well the propagation of detection boxes to neighboring frames allows the model to improve recall. We experiment on the validation set of PoseTrack2018, as the 2017 set does not have bounding box annotations. We compare our 3D approach against its 2D counterpart, using two different backbones (table 9). Results show that: (i) our 3D approach can indeed recover a substantial number of missed predictions (+4-7% recall) and (ii) it can even raise the recall of a weaker detector (3D MobileNet-V2, recall 83) on par with that of a much stronger model (2D ResNet-101, recall 82.9).

Person detector	Base 2D	Our 3D
Strong ResNet-101	82.9	86.5
Weaker MobileNet-V2	77.6	83.0

Table 9: *Person bounding box recall on PoseTrack 2018.*

5. Conclusion

We have presented a novel top-down approach for multi-person pose estimation and tracking in videos. Our approach can recover from failures of its person detector by propagating known person locations through time and by searching for poses in them. Our approach consists of three components. Clip Tracking Network was used to jointly perform joint pose estimation and tracking on small video clips. Then, Video Tracking Pipeline was used to merge tracklets predicted by Clip Tracking Network, when these belonged to the same person. Finally, Spatial-Temporal Merging was used to refine the joint locations based on a spatial-temporal consensus procedure over multiple detections for the same person. We showed that this approach is capable of correctly predicting people poses, even on very hard scenes containing severe occlusion and entanglements (fig. 6). Finally, we showed the straight of our approach by achieving state-of-the-art results on both joint detection and tracking, on both the PoseTrack 2017 and 2018 datasets, and against all top-down and bottom-down approaches.

References

[1] Posetrack 2017: Leader board. https://posetrack.net/leaderboard.php, 2017.

[2] Posetrack 2018: Leader board. https://posetrack.net/workshops/eccv2018/posetrack_eccv_2018_results.html, 2018.

[3] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[8] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 5:603–619, 2002.

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[10] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[11] Andreas Doering, Umar Iqbal, and Juergen Gall. Joint flow: Temporal flow fields for multi person tracking. In *BMVC*, 2018.

[12] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.

[13] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-Track: Efficient Pose Estimation in Videos. In *CVPR*, 2018.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[15] Jihye Hwang, Jieun Lee, Sungheon Park, and Nojun Kwak. Pose estimator and tracker using temporal flow maps for limbs. *IJCNN*, pages 1–8, 2019.

[16] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[17] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *CVPR*, 2019.

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[20] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[21] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, 2017.

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[23] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, 2017.

[24] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[25] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, and Yaser Sheikh. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In *CVPR*, 2019.

[26] Weijian Ruan, Wu Liu, Qian Bao, Jun Chen, Yuhao Cheng, and Tao Mei. Poinet: pose-guided ovonic insight network for multi-person pose tracking. In *ACM Multimedia*, 2019.

[27] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, 2017.

[28] Bharat Singh, Mahyar Najibi, and Larry S. Davis. SNIPER: Efficient multi-scale training. In *NIPS*, 2018.

[29] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. *arXiv preprint arXiv:1912.02323*, 2019.

[30] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[31] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

[32] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *arXiv preprint arXiv:1908.07919*, 2019.

[33] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[34] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[35] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.

[36] Dongdong Yu, Kai Su, Jia Sun, and Changhu Wang. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In *ECCVW*, 2018.

[37] Jiabin Zhang, Zheng Zhu, Wei Zou, Peng Li, Yanwei Li, Hu Su, and Guan Huang. Fastpose: Towards real-time pose estimation and tracking via scale-normalized multi-task networks. *arXiv preprint arXiv:1908.05593*, 2019.