Multi-Object Tracking with Hallucinated and Unlabeled Videos

Daniel McKee[1,2], Bing Shuai[2], Andrew Berneshawi[2], Manchen Wang[2],
Davide Modolo[2], Svetlana Lazebnik[1], Joseph Tighe[2]
[1]University of Illinois at Urbana-Champaign, [2]Amazon Web Services

Abstract

In this paper, we explore learning end-to-end deep neural trackers without tracking annotations. This is important as large-scale training data is essential for deep neural trackers, while tracking annotations are expensive to acquire. We first hallucinate videos from images with bounding box annotations using motion transformations along with simulated video effects to create a diverse tracking dataset. We then use a tracker trained from our hallucinated data to mine hard examples from a pool of unlabeled real videos. We propose an optimization-based connecting process to first identify and then rectify hard examples from the unlabeled videos. The output of this process is a set of mined hard examples with refined pseudo labels. We train jointly on hallucinated data and mined hard video examples, and our tracker achieves state-of-the-art performance on the MOT17 and TAO-person datasets.

1. Introduction

Recent progress on deep learning based trackers [1, 15, 14, 9] has elevated the performance of online multiple-object tracking to a level that is comparable to offline trackers. Training these online trackers requires both a large number of bounding box annotations for the detection component and a large number of instance-level correspondence annotations for the motion model. Unfortunately, large-scale instance correspondence annotations are extremely expensive to obtain, and current MOT datasets are usually restricted to a small number of videos (e.g. 7 training videos for MOT17 [10]). While these trackers can be trained with a large set of supplementary image based bounding box annotations, the small set of tracking annotations still limits performance and generalizability. In this work, we address these issues by exploring ways to auto-annotate large sets of videos for training a deep neural tracker. More specifically, we explore auto-annotation in a weakly-supervised setting, where we have bounding box annotations for images and a large pool of unlabeled videos. Our approach generates training videos from two sources. First we hallucinate short videos by applying transformations to an image and its corresponding bounding box annotations. Second, we use a tracker trained from these hallucinated videos to obtain pseudo-labels for unlabeled real videos, and we propose a video-level optimization process to identify and fix hard example failure cases in these pseudo-labels.

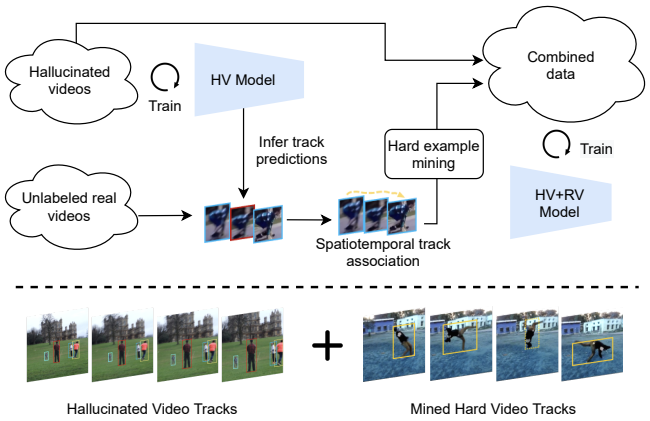

Figure 1: We first generate a dataset of hallucinated videos (HV) from images on which we train an HV tracking model. Next, we mine hard examples from unlabeled real videos (RV), and train a tracking model jointly on the HV and RV data. We hallucinate a short video from a single image with recurrent zoom-in/out motion transformations and random simulated motion effects. Hard examples refer to video clips where the HV-trained tracker fails, such as a video in which a person's pose deforms dramatically.

Our results include three key findings: 1) our weakly-trained tracker achieves state-of-the-art results on both the MOT17 [10] and TAO-person [4] datasets; 2) on the MOT17 [10] dataset, the tracker trained with large-scale self-generated data and annotations outperforms its counterpart trained on the provided MOT17 tracking annotation; and importantly 3) combining our self-generated annotations with the MOT17 training dataset significantly improves the performance.

2. Weakly-supervised DTracker

In this paper, we adopt a tracker architecture similar to the implicit model from SiamMOT [14], which achieves

Figure 2: Examples from our training data consisting of COCO[8]+CrowdHuman[13] hallucinated videos (top) and mined real Kinetics [3] videos (bottom) with their corresponding pseudo-label tracks. For the real video examples we visualize pseudo-labels of mined hard examples with broken tracks (dotted box) that were identified and corrected by our track rectification process.

state-of-the-art results on the MOT17 dataset. Hereafter, we refer to this specific deep neural tracker as **DTracker**. DTracker is a two-stage deep neural tracker based on Faster-RCNN [11] which includes three major functional sub-networks: region proposal network (RPN), detection branch, and track branch (motion model). Specifically, DTracker uses a single-object tracker – GOTURN [5] – to model an object instance's motion movement between two frames. Due to scarcity of video tracking annotations, the detection model in DTracker is trained with large-scale image datasets, while the motion model is trained with the limited annotated video data. As a result, the feature backbone of DTracker is heavily tuned for the detection task, and the motion model struggles to generalize in different scenes or motion patterns. To address these limitations, we explore training DTracker in a weakly-supervised setting by auto-annotating our own video sequences. We assume that we have a set of annotated images with bounding boxes for the objects of interest along with a pool of unlabeled videos. Below we describe our approach to create hallucinated videos with trajectories and generate a large dataset of auto-annotated unlabeled videos with mined hard examples.

Video Hallucination From Images. We first explore hallucinating short videos with corresponding tracking annotations from a single annotated image by applying successive transformations to the image and its bounding box annotations. We simulate zoom-in/out visual effects on images with cropping and scaling operations (crop_and_scale). Specifically, for an image $\mathbf{I}_1$, we generate a T frame video sequence $[\mathbf{I}_1, \ldots, \mathbf{I}_t, \ldots, \mathbf{I}_T]$, in which $\mathbf{I}_t = \text{crop\_and\_scale}(\mathbf{I}_{t-1}, r*(t-1))$ for $t \geq 2$. $r(\geq \frac{0.9}{T-1})$ denotes the relative size of image crop w.r.t that of original image $\mathbf{I}_1$. This zoom-in/out transformation is similar to the scaling and translation operation used in [15, 5]. To simulate more realistic sequences, we also inject visual effects such as motion blur, color/lighting changes and compression artifacts as illustrated in Figure 2.

Hard Example Mining from Unlabeled Videos. To generate training data from real videos, we start by running our hallucinated video trained DTracker over a pool of unlabeled videos. Our DTracker produces high confidence short tracks (tracklets) but fails to track objects through the challenging video-specific scenarios such as occlusion or object deformation. Inspired by other works which mine hard examples or rectify pseudo-labels to improve self-training for object detection [7, 6, 12], we propose a mining process that first locates these hard tracking examples and then rectifies them to output refined pseudo labels. We show visualization of such hard examples focused on broken object trajectories in the lower half of Figure 2.

To perform hard example mining and connect broken tracklets, we propose a pairwise matching cost between two tracklets in a video based on temporal and spatial coherence between the tracklets:

$$c(i,j) = tIoU(\tau_i, \tau_j) + (1 - \frac{t(\tau_i, \tau_j)}{\gamma}) + \chi(\tau_i, \tau_j) \quad (1)$$

where $tIoU(\tau_i, \tau_j)$ computes the spatial IoU between the end of track τ_i and the beginning of track τ_j while $t(\tau_i, \tau_j)$ computes the time gap between the two tracks. $\chi(\tau_i, \tau_j)$ is a characteristic function to impose hard constraints on matching between tracks. The characteristic function is defined as: $-\infty$ if $tIoU(\tau_i, \tau_j) < \mu$ or $t(\tau_i, \tau_j) > \gamma$, or $i \leq j$ and 0 otherwise.

In order to identify and rectify the broken tracklets, we propose an optimization process that connects the broken tracklets that belong to the same object instance. We define the optimization problem as follows:

$$\begin{aligned} &\max_x \sum_{i,j} c(i,j) x_{ij} \\ &\text{s.t.} \sum_i x_{i,j} \leq 1 \ \ \forall j, \quad \sum_j x_{i,j} \leq 1 \ \ \forall i, \\ &x_{ij} \in \{0,1\} \ \ \forall i,j \end{aligned} \quad (2)$$

in which we use a binary variable x_{ij} to denote the connectivity between two tracks (τ_i, τ_j) with 1 corresponding to joining the tracks. At each step, we impose constraints that a track may only be joined to one later track and one earlier track. This prevents cases where a single track might be split into multiple separate overlapping tracks. We optimize this process in an iterative manner: first running the optimization process to obtain a set of tracks to join, then

Method	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDsw ↓
Tracktor++v2 [1]	56.5	55.1	**8866**	235449	3763
NeuralSolver [2]	58.8	61.7	17413	213594	**1185**
CenterTrack [15]	61.5	59.6	14076	200672	2583
DTracker	60.7	58.8	15235	204373	2169
WDTracker	64.7	61.4	11970	185403	2023
WDTracker*	**65.8**	**63.5**	13076	**177893**	1883

Table 1: Comparison with state-of-the-art methods on MOT17 test set using **public detections**. All methods in the first block, including DTracker, are trained with MOT17 training dataset. WDTracker denotes weakly-supervised DTracker that is only trained with our self-generated dataset. WDTracker* represents DTracker that is trained with the combination of MOT17 and our self-generated dataset.

Model	AP@0.5	AP@0.75
Tracktor [1]	25.9	-
Tracktor++ [1]	36.7	-
WDTracker	40.7	22.0
WDTracker+ReID	**44.8**	**25.3**

Table 2: Comparison with state-of-the-art on TAO-Person validation set.

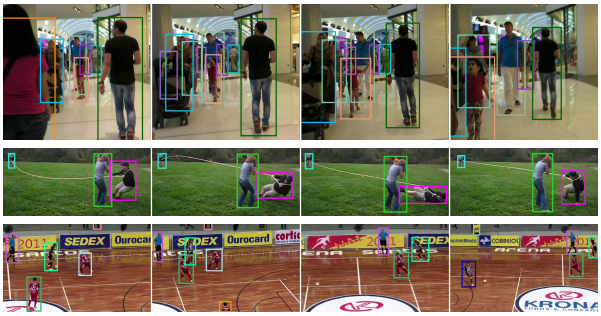

Figure 3: The tracker trained with our hallucinated and auto-annotated videos is able to track through very challenging scenarios including occlusion in crowded scenes, large pose deformations, and fast object motion. The example in the first row is from the MOT17 test set, and remaining examples are from the TAO-person validation set.

updating all tracks, and repeating until convergence when no additional tracks are matched. After this process we obtain a set of rectified pseudo labels along with hard examples which are the short sequences surrounding a timepoint where two tracks were joined. While our hard example mining process will inevitably have noise from either tracklet prediction errors or tracklet association errors in our optimization process, we mitigate these effects by joint training on the hallucinated video set which acts as a regularization.

3. Experiments

In this section, we present our Weakly-supervised DTracker (WDTracker) on public multi-person tracking datasets including MOT17 [10] and TAO-person [4].

Results on MOT17. We start by training a DTracker as other methods do on the 7 training videos in MOT17. For our WDTracker, as person bounding boxes in MOT17 are amodal, we use images from CrowdHuman dataset to generate our hallucinated videos as they include amodal bounding box annotations as well. That offers us 15,000 hallucinated videos with diverse scenes and settings compared to the limited 7 training videos available in MOT17. Our WDTracker is then trained with both hallucinated videos and hard mined self-generated annotations from real videos. We report the results on the MOT17 test set with standard metrics [10] including MOTA, IDF1, false positives, false negatives, and identity switches. In order to separate out the effects of detection, we generate the results with the public detections [10].

In Table 1 we present our MOT17 trained DTracker along with our WDTracker trained on both hallucinated and real mined video data. DTracker performs on par with other state-of-the-art trackers but our WDTracker achieves state-of-the-art results (64.7 MOTA / 61.4 IDF1), which outperforms existing best models (i.e. CenterTrack [15]) by a significant margin (3.2+ MOTA, 1.8+ IDF1). It's important to note that state-of-the-art models are all trained on the MOT17 dataset. This result suggests that our large-scale self-generated datasets are as valuable as small-scale manually-annotated datasets. To further vindicate this hypothesis, we present WDTracker* that trains the underlying DTracker jointly on our self-generated dataset and MOT17. Interestingly, we observe another non-trivial performance boost (1.1+ MOTA and 2.1+ IDF1), which indicates that our self-generated datasets are complementary to existing tracking manual annotations.

Results on TAO-Person. We train our WDTracker with hallucinated videos from the COCO-17 and CrowdHuman datasets together with hard examples mined from the Kinetics-700 dataset [3]. We adopt a larger feature backbone DLA-102, which is still a lighter network compared to ResNet-101 used for Tracktor++ [4]. We report the standard Federated Track AP [4] for TAO-Person. As shown in Table 2, our WDTracker significantly outperforms Tracktor and Tracktor++ by a substantial margin. When we also apply the person re-identification network and techniques as in [4] to link tracklets, our WDTracker is further improved to 44.8% AP@IOU=0.5, which sets a new state-of-the-art. These results show the clear advantage of our WDTracker as a generalizable tracker on the highly diverse and challenging examples in TAO. It's important to note that we are unable to present results of DTracker or WDTracker* as in Table 1, since TAO-person dataset does not have appropriate annotations for deep model training. We show qualitative tracking results in Figure 3.

Ablation on TAO-Person. We also include ablation studies on the TAO-Person dataset to validate usefulness

Model	Training data	AP@0.5	AP@0.75
Tracktor	HV	28.1	13.3
Tracktor + Flow	HV	32.5	16.4
WDTracker	HV^-	30.4	15.1
WDTracker	HV	32.0	15.4
WDTracker	HV+RV	35.4	17.0

Table 3: Ablation experiments on TAO-person dataset. HV and RV represents hallucinated videos and real videos respectively, while HV^- denotes hallucinated videos with only zoom-in/out motion effects.

of hallucinated video training and hard real video example mining in Table 3. As a first step, we train a DLA-34 based Tracktor on hallucinated videos (HV) generated from the COCO-17 and CrowdHuman datasets which achieves 28.1% Track AP @ IOU=0.5, significantly outperforming the Tracktor model reported in [4] (i.e. 25.9% in Table 2). Further, we adopt a second competitive baseline by estimating a person's movement with optical flow. In detail, we estimate the person's motion offset between two frames by taking the median motion vector of all constituent pixels within the person's bounding boxes. To extract optical flow, we use PWC-Net which is trained on standard supervised optical flow datasets. As expected, adding flow to Tracktor improves Track AP @IOU=0.5 by a substantial 4.4%.

Next, we train our WDTracker, which is effectively Tracktor + a motion model, on our hallucinated video (HV) dataset. As demonstrated in Table 3, it achieves 32.0% AP@IOU=0.5, which is 3.9% higher than Tracktor. This result shows that the motion model learned from hallucinated videos is able to generalize to real videos and improve tracking. Moreover, the model performs comparably to our strong baseline – Tracktor + Flow, which suggests that the learned motion model is as good as a fully-supervised flow model for predicting the person's movement. For comparison, we also train our WDTracker on a simpler set of hallucinated videos (HV^-) that don't have our advanced motion effect simulation (i.e. motion blur, light changes, video compression artifacts). This model under-performs that trained on HV by a clear 1.6% AP@IOU=0.5 margin, showing that advanced video augmentation is important. Lastly, we re-train WDTracker with hard examples mined from our real video (RV) dataset, and we observe a significant 3.4% AP improvement. This improvement demonstrates the significance of mined hard examples from real videos, which enables the learned model to track through occlusion, large pose deformation, and motion blur.

4. Conclusion

In this paper, we explored training a deep neural tracker in a weakly-supervised setting using only annotated detection images and unlabeled videos. In detail, we first hallucinated video sequences by applying zoom-in/out motion transformation and other simulated video effects. Next, we proposed to use this model to mine tracks from unlabeled video, and we presented an optimization-based connecting process to identify and rectify broken tracks while mining hard examples. Our weakly-supervised tracker achieves state-of-the-art tracking performance on the challenging MOT17 and TAO-Person datasets.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *CVPR*, 2019. 1, 3

[2] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020. 3

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv:1907.06987*, 2019. 2, 3

[4] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *ECCV*, 2020. 1, 3, 4

[5] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. 2

[6] SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *ECCV*, 2018. 2

[7] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. Improving object detection with selective self-supervised self-training. In *ECCV*, 2020. 2

[8] 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. 2

[9] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv:2101.02702*, 2021. 1

[10] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, Mar. 2016. 1, 3

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[12] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *CVPR*, 2019. 2

[13] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv:1805.00123*, 2018. 2

[14] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *CVPR*, 2021. 1

[15] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv:2004.01177*, 2020. 1, 2, 3