PathTrack: Fast Trajectory Annotation with Path Supervision

Santiago Manen[1] Michael Gygli[1] Dengxin Dai[1] Luc Van Gool[1,2]

[1]Computer Vision Laboratory
ETH Zurich

[2]ESAT - PSI / IBBT
K.U. Leuven

{smanenfr, gygli, daid, vangool}@vision.ee.ethz.ch

Abstract

Progress in Multiple Object Tracking (MOT) has been historically limited by the size of the available datasets. We present an efficient framework to annotate trajectories and use it to produce a MOT dataset of unprecedented size. In our novel path supervision *the annotator loosely follows the object with the cursor while watching the video, providing a* path annotation *for each object in the sequence. Our approach is able to turn such weak annotations into dense box trajectories. Our experiments on existing datasets prove that our framework produces more accurate annotations than the state of the art, in a fraction of the time. We further validate our approach by crowdsourcing the* PathTrack *dataset, with more than 15,000 person trajectories in 720 sequences*[1]*. Tracking approaches can benefit training on such large-scale datasets, as did object recognition. We prove this by re-training an* off-the-shelf *person matching network, originally trained on the MOT15 dataset, almost halving the misclassification rate. Additionally, training on our data consistently improves tracking results, both on our dataset and on MOT15. On the latter, we improve the top-performing tracker (NOMT) dropping the number of ID Switches by* 18% *and fragments by* 5%*.*

1. Introduction

Progress in vision has been fueled by the emergence of datasets of ever-increasing scale. An example is the surge of Deep Learning thanks to ImageNet [26, 44]. The scaling up of datasets for Multiple Object Tracking (MOT) however has been limited due to the difficulty and cost to annotate complex video scenes with many objects. As a consequence, MOT datasets consist of only a couple dozens of sequences [18, 29, 35] or are restricted to the surveillance scenario [53]. This has hindered the development of fully learned MOT systems that can generalize to any scenario. In this paper, we tackle these issues by introducing a fast and intuitive way to annotate trajectories in videos and use it to create a large-scale MOT dataset.

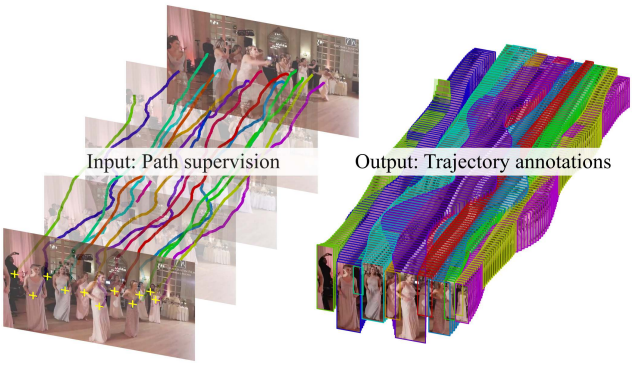


Figure 1: This sequence is heavily crowded with similarly-looking people. Annotating such sequences is typically time-consuming and tedious. In our *path supervision*, the user effortlessly follows the object while watching the video, collecting *path annotations*. Our approach produces dense box trajectory annotations from such path annotations.

Objects can be annotated at different levels of detail. The cheapest way is to provide video-level object labels [39] or action labels [4]. On the other end of the spectrum, sophisticated methods [38, 30, 3, 46, 2] produce pixel-accurate segmentations of objects. Per-frame bounding box annotations lie in between these extremes. We call this the *trajectory annotation* task. The common approach to it is to annotate a sparse set of boxes and interpolate between them linearly [55] or with shortest-paths [47]. This is expensive, *e.g.* it cost *tens of thousands of dollars* to annotate the VIRAT dataset [49].

The typical annotation pipeline involves the user idly watching the video in-between manual annotations. This is arguably a waste of time. In this paper, we present *path supervision* as a more productive alternative. In it, the annotator follows the object with the cursor while playing the video, collecting a *path annotation*, *c.f.* Fig. 1. Hence, *watching time* is efficiently turned into *annotation time*. Our experiments show that these paths are fast to annotate, almost in real time.

[1]We provide our dataset at http://www.vision.ee.ethz.ch/~smanenfr/pathtrack.

Table 1: Comparison of PathTrack with other popular MOT datasets.

Dataset	Train			Test			Total			Classes (P = Person, C = Car)	Camera (S=Static M=Moving)	Scene-type label	Camera-movement label
	# seqs	Duration (mins)	# tracks	# seqs	Duration (mins)	# tracks	# seqs	Duration (mins)	# tracks				
Virtual KITTI [17]	-	-	-	-	-	-	5*	4*	261*	C*	car-mounted		
KITTI [18]	21	13		29	18	-	50	30	-	C + P	car-mounted		
MOT15 [29]	11	6	500	11	10	721	22	16	1221	P	S+M		
MOT16 [35]	7	4	512	7	4	830	14	8	1342	C+P**	S+M		
PathTrack (ours)	**640**	**161**	**15,380**	**80**	**11**	**907**	**720**	**172**	**16,287**	**P**	**S+M**	✓	✓

* [17] provides 10 different conditions (e.g. different angles, lighting conditions) for each of the 5 sequences. Sequences are virtually rendered.
** [35] provides a rich set of labels, such as whether an object is an occluder or a target is riding a vehicle.

Path annotations are approximate and do not provide the scale of the object. So recovering full box trajectories from them is far from trivial. We alleviate these problems by using object detections, since our goal is to generate large MOT datasets, for which we know the class of interest. Our optimization produces an accurate box trajectory for each path annotation, by linking detections in a global optimization. Our approach is presently the fastest way to annotate MOT trajectories for any annotation quality.

Since our annotation approach is intuitive, we could crowd source a large-scale dataset with Amazon Mechanical Turk (AMT) [1]. This *PathTrack* dataset is our second major contribution: a large MOT dataset of more than 15,000 person trajectories in 720 sequences, 30 times more than currently available ones [29]. Its focus lies on a large-scale and diverse training set, aimed to initiate a new generation of fully data-driven MOT systems. We show its potential by learning better detection-association models for MOT, which substantially improves the top-performing tracker in MOT15, i.e. NOMT [9]. In summary, our contributions are:

- A novel approach to produce full box trajectories from path annotations. It is currently the fastest way to annotate trajectories for any annotation quality and it specially shines for quick quantity-over-quality data collection strategies, ideal for training data.
- The novel PathTrack MOT dataset, which includes the collection and annotation of 720 challenging sequences. It focuses on providing abundant training data to learn data-driven trackers. We show its potential by improving the top tracker on MOT15 [29].
- Insights into collection of training data for MOT. Our experiments show that the MOT community can still benefit from more training data and a saturation point has not yet been reached. Furthermore, quantity seems to be more important than quality when learning to link detections into trajectories.

2. Related work

There is quite some work on multimedia annotation [11]. The most related works annotate objects in videos and can generate datasets for MOT training and evaluation.

Trajectory annotation in videos We focus on frameworks aimed at annotating persons with the purpose of generating tracking datasets. Of less relevance to us are those that work on videos with only a few people, such as [50, 36]. The naive way to annotate trajectories is to indicate the object location in every frame. This is inefficient as objects tend to move little between frames. Hence, VIPER-GT [34] and LabelMe video [55] propose to linearly interpolate boxes between annotated keyframes. There is also a family of methods that learn an appearance model from a sparse set of box annotations. VATIC [49] uses this appearance model to define a graph on which it performs a shortest-path interpolation between manual annotations with Dynamic Programming [6]. The shortest-path interpolation allows for larger time gaps without manual annotations, assuming that the object is clearly visible, and it can be efficient [51]. A VATIC improvement [48] incorporated active learning to decide which frames to annotate, to maximize the gain coming with such frames [40]. [10] built on top of shortest-path interpolation by updating the optimization weights with each extra annotation. Recently, [19] reconstructed annotated boxes and interpolated the final trajectories in 3D space. Based on the aforementioned approaches, multiple annotation tools have been developed [23, 8, 37]. Some gamify the annotation process [12]. As an alternative to trajectory supervision, some works aim to automatically discover and track objects in video collections, e.g. [27].

Compared to previous approaches, we annotate large quantities of videos with the minimum effort possible and prefer quantity over quality in our training data, which have shown success in other tasks.

Path supervision Pointing at objects comes very natural and has often been used in human-computer interaction [21, 22], yet it only recently gained popularity in Computer Vision. In parallel with our work, [33] found path annotations promising for action localization in videos. Compared to [33, 52], we annotate dozens of people in highly-crowded sequences, ideal for MOT purposes. Also recently, [5] and [22] used point supervision to segment objects in images and videos, resp. [22] uses *multiple* points to segment, by iteratively re-ranking a collection of thousands of object proposals, called *Click Carving*.

We are the first to propose a trajectory annotation framework based on linking detections with path supervision and use it to generate a large MOT dataset.

Tracking datasets There is a corpus of video datasets that provide frame-level [15, 20] or pixel-level annotations [38]. [25] and [41] are the largest datasets for single object tracking. Most large-scale MOT datasets are restricted to surveillance videos [13, 45, 53, 43], since they depict smooth and quasi-linear trajectories that are easy to annotate. More related to ours, KITTI [18] is collected from a car-mounted camera and focuses on pedestrians and vehicles. Parts of this dataset have been reproduced and rendered virtually, to show the potential of virtual datasets [17]. [29, 35] have become the standard benchmarks for MOT, containing complex pedestrian scenes with static or moving cameras. Compared to these datasets, ours exhibits more diverse scenes and camera movement and is 33 times larger. Tab. 1 shows a quantitative comparison.

3. Trajectory annotation with path supervision

In this section, we describe our annotation framework: we formalize path supervision in Sec. 3.1 and then detail how we leverage it to infer accurate trajectories in Sec. 3.2. In Sec. 3.3 we show how to incorporate box supervision.

3.1. Path supervision

A path annotation of an object i consists of an (x, y)-coordinate $\mathbf{p_i}(t)$ that *lies* inside its bounding box at frame id t. Path annotations are intuitive and efficient to obtain by watching each object independently while following its location with the mouse cursor, *c.f*. Fig. 1. Our results show that annotating paths is only 33% slower than watching the video in real time. We say that a video has path supervision if a human annotator has provided a *path annotation* for the objects of interest. The following section explains how we use these annotations to obtain accurate box trajectories.

3.2. From path supervision to full box trajectories

While path supervision is intuitive and efficient, it comes with its own set of challenges: a) It offers no information about the spatial extent of the object. b) The relative position of the path annotation inside the object is unknown. We partially solve these two problems by drawing on the success of *object detection*, since our final goal is to generate large MOT datasets and we know what kind of objects we want to annotate. Object detection is gaining maturity for objects of primary interest, so it is natural to use it as an established technique. Each detection is represented with a box and a confidence score at a given frame.

Our goal is to infer the trajectories $\mathcal{T}$ of the objects in the sequence, given the set of input path annotations $\mathcal{P}$ and object detections $\mathcal{D}$. This problem is similar to the tracking-by-detection data association problem, but with additional information from path supervision: the number of objects, their time span and their rough location are given. Our optimization considers the following intuitive forces:

1. **Path potential:** Detections should be assigned to trajectories with compatible path annotations.
2. **Video-content potential:** *Confident detections* should be used and *affine detections* should be encouraged to have the same label. We say that two detections are affine if they are likely to belong to the same object in different frames, according to the content of the video.
3. **Trajectory constraint:** Trajectories have a single location per frame. Therefore, at most one detection can be assigned to one trajectory at any given frame.

We include these conditions in a two-step optimization. We first relax the trajectory constraint and label each detection with a provisional trajectory. This clusters the detections according to their corresponding trajectory, *c.f*. Fig. 2b. We can assume that a final trajectory can be constructed with detections from its cluster, and will not contain detections from another cluster. This *detection pre-labeling step* is detailed in Sec. 3.2.1. At this point each cluster is might include false positives, which violate the trajectory constraint. So, in a second step, we find the most probable trajectory in each cluster in a *detection linkage* step, see Fig. 2c. We describe this step in Sec. 3.2.2.

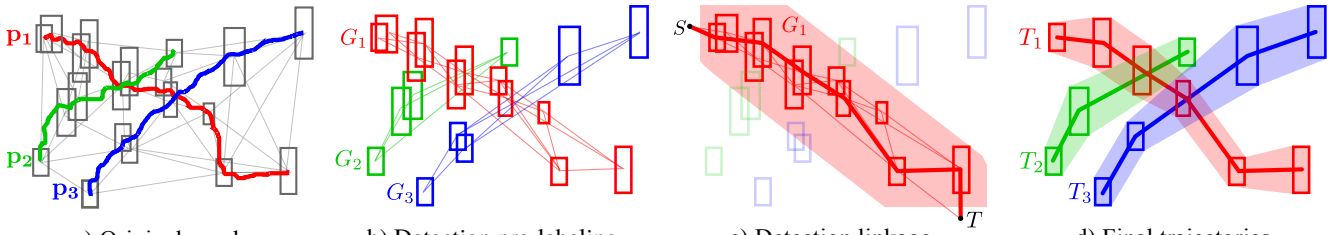
3.2.1 Detection pre-labeling

The goal of this step is to assign a path annotation label y_i to each detection d_i. Dropping the trajectory constraint allows us encode the path and video-content potentials in a global discrete energy minimization framework. Intuitively, we will assign path annotations to compatible object detections, assigning affine detections to the same cluster. The optimal label assignment $\mathcal{Y}^*$ is that which minimizes:

$$\underset{\mathcal{Y}}{\text{minimize}} \quad \sum_{i \in \mathcal{D}} U(y_i) + \sum_{(i,j) \in \mathcal{E}} W(y_i, y_j) \tag{1}$$

where the unary potential $U(y_i)$ is the cost of assigning label y_i to detection i and the pairwise potential $W(y_i, y_j)$ the cost of assigning different labels to detections i and j according to their affinity. For computational reasons, we limited to a temporal window of 4 seconds, which did not worsen the empirical results. Fig. 2b illustrates a typical pre-labeling result. We now describe the potentials we use.

As aforementioned, we do not assume the path annotations to be pixel-accurate center annotations. Instead we assume that they frequently lie in the bounding box of the object, a much weaker restriction. Therefore, our unary potential encourages assigning a label y to a detection d_i if the



a) Original graph b) Detection pre-labeling c) Detection linkage d) Final trajectories

Figure 2: Overview of our pipeline. a) We take path annotations ($\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}$) and object detections as input. b) We pre-label each detection with a potential path candidate, creating detection clusters (G_1, G_2, G_3). c) For each cluster, we compute the most likely trajectory via ST shortest paths. Finally, we output full bounding-box trajectories (T_1, T_2, T_3) for each path annotation.

corresponding path annotation $\mathbf{p}_y(t_i)$ falls inside the detection for the corresponding frame t_i:

$$U(y_i) = \begin{cases} 0, & \text{if } \mathbf{p}_y(t_i) \in d_i, \\ \infty, & \text{otherwise.} \end{cases} \quad (2)$$

Indeed our unary only requires a rough location of the path annotation somewhere inside the bounding box of the object. Note that this requirement does not need to be satisfied in every frame: the path supervision occasionally falling inside the object is usually enough to annotate it accurately. We prune detections which do not contain path annotations.

While the unary potential is based on the path supervision, the pairwise encodes video content. It discourages affine detections being assigned to different clusters:

$$W(y_i, y_j) = \begin{cases} -\log a_{ij}, & \text{if } y_i \neq y_j, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where a_{ij} represents the affinity between detections i and j and must be decimal number between 0 and 1. This pairwise potential is submodular, so the energy function Eq. (1) can be solved with Graph Cuts [24] efficiently. We now describe the affinity measure we used.

OF-trajectory affinity measure In our work, we use an affinity measure based on *optical-flow trajectories* (OF trajectory). These are obtained by linking pixels through time using frame-to-frame optical flow and forward-backward consistency checks [16]. These trajectories are represented with an (x, y)-position for each frame in their time span. Intuitively, two detections that share many OF trajectories are very likely to belong to the same object. Thus, we define the affinity between two detections as the intersection-over-union of their OF-trajectories, in the spirit of [9]. More details follow in the supplementary material.

So far we have discussed how we pre-assign object detections to path annotations *c.f.* Fig. 2b. In the following section, we describe how to obtain the most likely trajectory for each detection cluster via shortest-paths, *c.f.* Fig. 2c.

3.2.2 Detection linkage

In this second step, the goal is to infer the final object trajectories. Finding the most probable detection-paths in a set of detections has been well studied in the MOT literature [31]. We assume that the detection pre-labeling step has labeled the set of detections appropriately. So each detection can either be part of its assigned trajectory or a false positive, but it can not belong to another trajectory. Thus, we process each detection-cluster independently Fig. 2c and find the most probable detection-path in the cluster Fig. 2d.

Let T_i be the final trajectory corresponding to detection-cluster i. It will be composed of a set of time-sorted detections x_1 to x_K. We find the most likely trajectory by minimizing the sum of detection-confidence costs and between-detections transition costs [56]. Fig. 3 shows how this can be intuitively interpreted as finding the shortest-path in a directed ST-graph where detections are represented by detection-confidence edges. The optimal detection-path will have the lowest cost:

$$\underset{T}{\text{minimize}} \sum_{i=1}^{K} C(x_i) + \sum_{i=1}^{K-1} W(x_i, x_{i+1}) \quad (4)$$

where C is the detection-confidence cost. C_i follows the expression $\log((1 - s_i)/s_i)$, where s_i is the 0-to-1 score-confidence of the detection. Importantly, we use the same transition costs W when linking detections as we used in step one Eq. (1) for pre-labeling detections. Reusing pairwise costs makes the method more efficient. The detection-confidence costs become negative for confident detections, encouraging the optimization to include them in the final position, while the transition costs penalize the association of detections which are unlikely to be connected. We refer the reader to [56] for details. The entry and exit nodes, S and T, are connected only to the earliest and latest detections in the cluster, respectively, ensuring that the trajectory has the same time span as its corresponding path annotation.

As result of the optimization we have a sparse detection-path. Empirically we find the gap between detections to be small, 0.2 on average in our data. Thus we opt to linearly interpolate between detections to obtain the final trajectory, as per standard practice [55].

Until now we have presented our annotation approach using path supervision. It is useful for quickly annotating many sequences, particularly interesting for training data collection. We propose next an extension to incorpo-

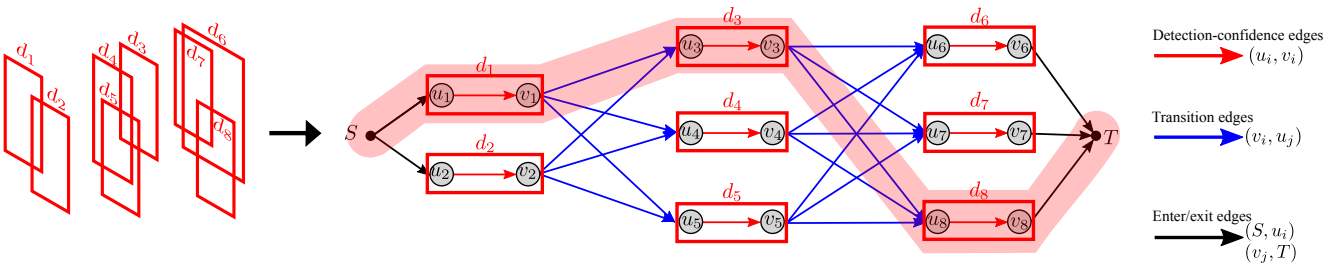


Figure 3: Over a set of prelabeled detections a min-cost flow network is defined. Each detection is represented by an *observation edge* with the confidence cost. ST-shortest paths are computed over this graph, red shadow.

rate additional box annotations, improving trajectories up to ground-truth quality.

3.3. Incorporating box supervision

We propose a simple yet effective way to extend our method with box annotations, to achieve ground-truth quality. Consider the detection-path we used to interpolate a trajectory. To include a box annotation, we simply add it to the path and then remove temporally close detections, those less than half a second away. Interpolating the updated detection-path produces the final trajectory. These fast updates progressively improve trajectory annotations as more box annotations are included. Our method is more accurate than the state of the art for any number of updates, as we show in the experiments.

4. The PathTrack dataset

We use our annotation approach to collect a MOT dataset of unprecedented size. This *PathTrack* dataset is an important part of our contribution. We first provide an overview of the dataset in Sec. 4.1. Then, we describe how we crowdsourced the annotations in Sec. 4.2. We generate the final trajectory annotations with our approach, which associates R-CNN detections [42] with the help of path supervision. Importantly, we focus on training data in order to encourage research in fully data-driven trackers.

4.1. Dataset overview

The PathTrack dataset consists of 720 sequences with a total of 16,287 trajectories of *humans*. Focusing on tracking humans allows us to collect more data for this specific class, which is of great interest both in the MOT community and in practical applications. The sequences are partitioned in a training set of 640 sequences with 15,380 trajectories and a test set of 80 sequences with 907 trajectories. Importantly, we allow a certain amount of *noise* in the training set annotations. This noise stems from inaccuracies in the path supervision and full-trajectory inference and has allowed us to annotate more sequences for a given time budget. Our experiments show that we can learn strong appearance models from large quantities of data even if the annotations are not perfectly clean (Sec. 5). Indeed, favoring quantity over quality when collecting training data has also been found to be beneficial for other tasks [54, 20]. Additional effort has been made for test annotations to be clean for evaluation purposes. Tab. 1 compares PathTrack with other popular MOT datasets. Compared to MOT15 [29], our dataset contains 33 times more sequences and 26 times more trajectory annotations available. We hope that the large scale of PathTrack encourages research in more data-driven tracking algorithms.

Dataset diversity MOT datasets typically focus on surveillance [53], street-scenes [29, 35] or car-mounted cameras [18, 17]. With PathTrack, we aim to explore tracking in new types of sequences. We have thus collected a diverse set of sequences and we have labeled each one according to two criteria: a *camera-movement label* and one out of 7 *scene labels*, *c.f*. Fig. 4. There is a clear emphasis in street scenes and moving cameras, due to their challenge, ubiquity and general interest. But our dataset also allows focusing on static cameras or sequences with a lot of motion, such as *sports* and *dancing*. These fine-grained categories can also help to evaluate tracking under different conditions. Additional statistics that show the diversity of our data are presented in Fig. 4c. In the following sections, we describe how we crowdsourced the path annotations and detail in the supplementary material how we collected the videos.

4.2. Crowdsourcing path annotations

A critical aspect of any annotation framework is whether it is *easy to use*. This is an often-overlooked factor that is vital if we want to crowdsource annotations. Path annotation is intuitive and straightforward. This has allowed us to crowdsource 16,287 path annotations of PathTrack using AMT. We now describe our interface and the measures we took to ensure the quality of our annotations.

Interface Our interface features a video player with browsing capabilities and a list of the current annotations. The key difference with other interfaces is that ours records the path of the object by following the cursor. Additionally, the user easily can speed up and slow down the video, according to the speed of the object. In our measurements, path annotation was only 30% slower than watching the

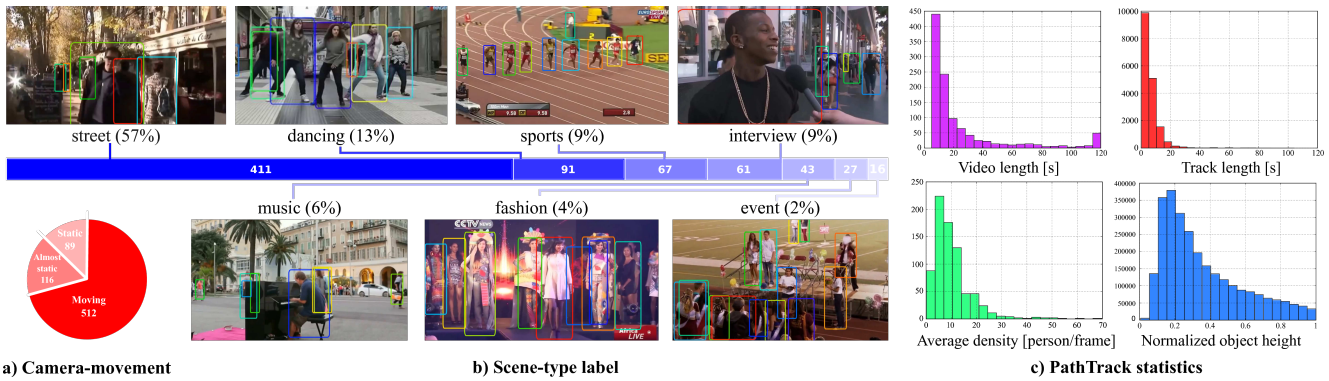


Figure 4: Scene-label distribution in PathTrack. We show in a) the distribution of camera-movement labels. Almost three quarters of the sequences have been recorded with a moving camera. We show in b) the distribution of scene-type labels and corresponding examples. More than half of the sequences are street scenes. c) Statistics of PathTrack. Videos are up to 2 minutes long.

video in real time. To further improve our final trajectories, we also asked the workers to provide a bounding box for the first and last appearance of each object and a third one in between. Since some sequences are very long and can contain dozens of people, the workers were allowed to partially annotate sequences. This also means that some workers received partially annotated videos and had to continue annotating them. This was not much of a challenge with our annotation framework. We have received very encouraging feedback from our workers, validating the ease of use of our interface and suggesting a potential for gamification. Here are some examples:

> *"System was very easy to use and the normal speed was perfect for tracking each subject."*
>
> *"I really enjoyed your hit. I like to do a lot of annotation work on mechanical turk and thought your interface was, once I got used to it, one of the best I have worked with."*

Qualification process After a short training video, *c.f*. supplementary material, each worker was asked to qualify by annotating the *TUD-Stadtmitte* sequence. The qualification certificate was only provided if the path and box annotations were similar to the ground truth up to a certain threshold. This was checked automatically.

Reviewing process If the users are not trained properly or the interface is cumbersome to use, crowdsourced annotations can be erroneous [49]. So we have made an extensive effort to review *every single video* and remove bad annotations. Videos with missing annotations were sent back to the annotation pool. We revoked the qualification of workers who continuously provided faulty annotations. Interestingly, only 3 out of our 81 workers were revoked, while previous work had difficulties collecting annotations of sufficient quality [47]. This further confirms that path annotation is an engaging and natural way to annotate trajectories.

5. Experiments

We present our experiments in three parts. First we evaluate our annotation framework in Sec. 5.1. We then demonstrate in Sec. 5.2 its impact on training data collection for matching detections, which is a key problem of MOT [9] that is shared by most trackers. We finalize by evaluating the impact of our data on the Multi Object Tracking task.

5.1. Trajectory annotation efficiency

In this section we evaluate the effectiveness and efficiency of path supervision and compare it to other trajectory annotation approaches.

Dataset description We evaluate our method on the MOT15 dataset [29] since it is most similar to our final goal, the generation of a massive MOT dataset. This dataset consists of 22 sequences, 11 of which belong to the training set. The sequences are challenging. Pedestrians are frequently occluded and some sequences have been recorded with a moving camera. We evaluate on the 521 trajectories of the training set, for which the ground truth is provided.

State of the art We compare to other existing trajectory annotation approaches. LabelMe [55] is an effective framework based on linear interpolation between box annotations. The more sophisticated VATIC [49] learns an appearance model from the box annotation, which it uses for a shortest-paths interpolation. An additional extension of VATIC uses active learning to propose to the user which frame to annotate [48].

Effectiveness of path supervision We first follow the standard evaluation of trajectory annotation frameworks [47]. In Fig. 5, we compare the annotation accuracy for different amounts of box annotations. Except for the active learning version of VATIC [48], box annotations are distributed uniformly in time, *e.g.*, every 10, 5, 1 seconds. The performance of each framework is measured in terms of how many ground truth boxes are recalled,

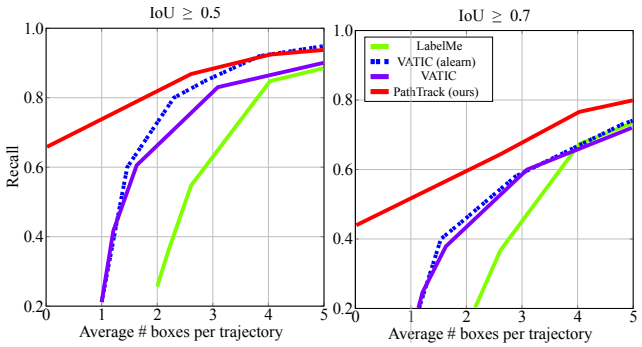


Figure 5: Performance comparison of 3 state-of-the-art trajectory annotation frameworks and ours. We plot the annotation accuracy for different box-annotation budgets.

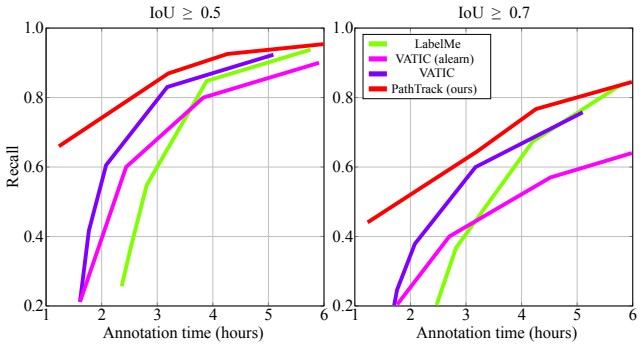


Figure 6: We compare the efficiency of our method with the state of the art for 0.5 and 0.7 IoU thresholds. The time measurements derive from a user study with 91 subjects.

for different Intersection-over-Union (IOU) [14] thresholds. Fig. 5 demonstrates the *effectiveness* of our path supervision: our cheap path supervision improves performance for any amount of box annotations. Interestingly, the annotation frameworks seem to converge in performance for large annotation budgets. A problem of this classical comparison is that it does not take into account the effort required to annotate path trajectories, *i.e.*, it assumes that path annotations can be produced in real time, which is not always the case. We address evaluate time performance in the next section.

Annotation efficiency We compare the *efficiency* of path supervision with previous approaches using a common unit to measure effort: the *annotation time*. Our time measurements are based on a user study of 78 AMT workers and 13 vision-expert annotators. We consider three time-consuming components: 1) watching the video at least once to identify the objects, 2) following each trajectory individually while annotating its boxes or path (for ours) and 3) the time required to annotate the bounding boxes. Our measurements revealed that box annotations take 5.2 seconds on average and that path annotations require slowing down the video by 33% on average. We provide a detailed explanation in the supplementary material. We use these time measurements to produce Fig. 6, where we compare the efficiency of our framework with the state of the art. Our method is efficient, as VATIC [49] and LabelMe [55] respectively require almost twice and three times more time to obtain our accuracy with only path supervision. We observe again how all methods converge to the same performance for a larger annotation budget, but ours is much more accurate for very small annotation-budgets.

Overall, our framework is ideal for fast video annotation, which is desirable for generating large training sets, as we demonstrate in the next section.

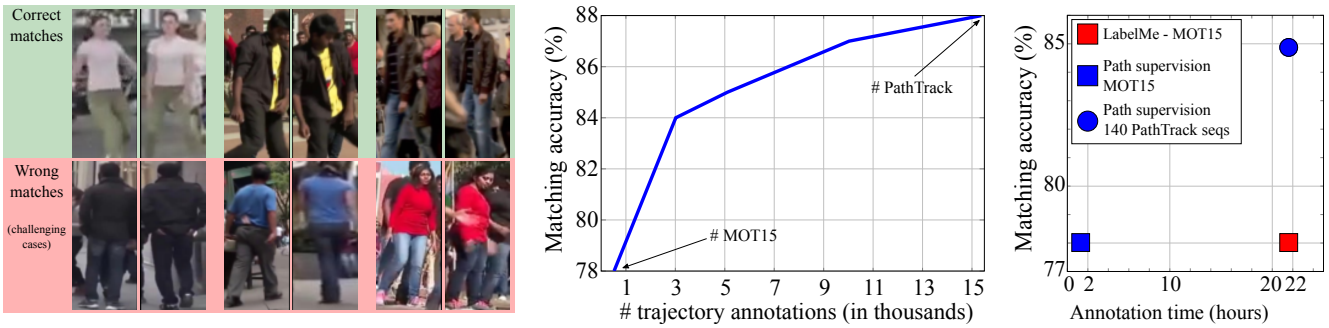
5.2. Person matching

We demonstrated in the previous section that path supervision is an efficient way to obtain accurate annotations in a short amount of time. We now explore the implications for a key task in MOT applications: *person matching*. This key problem consists of determining the likelihood that two detections belong to the same object in different frames Fig. 7a. There is a long tradition of handcrafted matching functions in the literature, with Convolutional Neural Networks (CNN) becoming more popular in the last few years. These models require extensive training data [26, 54], which we can provide with PathTrack. Learning tracker-specific components (e.g. entry/exit costs, mixing coefficients) is outside of the scope of this paper, but should be possible with our data.

We aim to answer the questions: i) does the tracking community benefit from more training data?, ii) for a limited budget, should we prioritize data quantity or quality?

Experimental protocol We base our conclusions on a person matching network similar to SiameseCNN [28]. The network takes as input the crops of the two detections, resized to 121x53, and outputs a confidence score that they belong to the same object. These input crops are stacked, so the input volume is of 121x53x6. The network has a simple AlexNet style architecture of 3 convolutional and 2 fully connected layers [28]. In our evaluations, we sample 2 million training and test samples. Positives are randomly sampled pairs of detections that belong to the same object up to 6 seconds away. For each positive we sample a negative pair belonging to another trajectory in the same video. We use a learning rate of 0.001. In our experiments, we train this network with different data sources and compare their test accuracies. Accuracy refers to the percentage of properly classified pairs. We evaluate on the test set of PathTrack, for which the ground truth annotation is clean.

Impact of training data In Fig. 7b we evaluate how the accuracy evolves as more training data becomes available. The left extreme corresponds to training on the 521 trajectories of the MOT15, which yields an accuracy of 78%. Training on the full 15,380 trajectories of PathTrack we improve the accuracy by 10%, *almost halving* the misclassification rate. This clearly shows the potential of PathTrack. More-



a) Qualitative results of person matching b) Impact of training data c) Annotation strategy

Figure 7: We show in a) qualitative results of our person matcher on PathTrack. False positives are even challenging for humans. b) Evolution of matching accuracy for different amounts of training trajectories. Training on the 15,380 trajectories of PathTracks results in an accuracy of 88%, reducing the misclassification rate by 45%, compared to MOT15. c) Person-matching accuracy for different annotation times using path supervision (blue) or exhaustive LabelMe annotation (red). A high-quantity annotation strategy with our path supervision provides the best accuracy for the same annotation-time budget.

over, we observe a certain effect of diminishing returns, but have not reached a plateau. If we use *context features* (e.g. relative distance, size) [28] in the network, we also see an improvement when using our data, from 84% to 90%. This shows that our data is useful to learn data-driven MOT.

Quantity-over-quality annotation When collecting and annotating data for training purposes, a vital question is whether we should coarsely annotate a large amount of data or precisely annotate a small amount of data. That is, whether we should follow a *quantity* or a *quality* strategy. We estimate that it would take 22h to perfectly annotate the 11 videos in the training set of the MOT Challenge with LabelMe [55]. We reach this number by counting only the number of windows necessary to obtain an accuracy larger than 0.95 IoU. This represents the *high-quality* strategy. We compare this with a *high-quantity* strategy, in which, for the same annotation time, we annotate 140 videos of PathTrack with our framework, with path supervision and 3 boxes per trajectory. We show the results in Fig. 7b. A *high quantity* approach boosts the final accuracy from 78% to 85%. Interestingly, we can also use our method to quickly annotate the MOT 15 training set and train a model with *exactly* the same accuracy *c.f*. Fig. 7b. These results further showcase the benefit of our framework, which is ideal for fast annotation of large datasets. Other works [54, 20] have also found a quantity strategy to be advantageous to train deep models.

5.3. Multi Object Tracking

In the previous section we demonstrated how we can train strong person-matching models with PathTrack. We now evaluate what impact this improvement has on MOT performance. We first use a standard tracker based on Linear Programming (LP) [56, 32] and evaluate it on the test set of PathTrack with the standard CLEAR MOT metrics [7]. In Tab. 2a We compare the performance of this tracker with two different person-matching models: one trained on MOT15 and the other on our data. Training on PathTrack substantially improves all the metrics. We further show the potential of PathTrack by improving the top-performing tracker in MOT15 [9] with our person-matching model *c.f*. Tab. 2b. More specifically, we use our discriminative person matcher to further link their trajectory results through occlusions, improving the number of ID Switches by 18% with 5% less fragments. Low-level details about the trackers are presented in the supplementary material.

Table 2: We show in a) how training on PathTrack improves all metrics compared to training on MOT15. We use in b) our person-matcher (TRID) to improve the top method in MOT15.

(a) Tracking results on PathTrack

LP Tracker trained on	MOTA ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Switch ↓
MOT15 [29]	24.5	81.4	44.2%	19.2%	42,502	37,720	1,827
PathTrack (ours)	**27.6**	**81.5**	**47.3%**	**18.2%**	**40,614**	**36,508**	**1,576**

(b) Tracking results on MOT15

Tracker	MOTA ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Switch ↓	Frag ↓
NOMTwSDP [9]	55.5	**76.6**	39.0%	**25.8%**	**5,594**	21,322	427	701
+ TRID (ours)	**55.7**	76.5	**40.6%**	**25.8%**	6,273	**20,611**	**351**	**667**

6. Conclusion

In this work, we propose a new framework to annotate trajectories in videos using *path supervision*, with the goal of generating massive MOT datasets. In the path supervision paradigm, the user annotates the position of the objects of interest with the cursor while watching the video. Our user study shows that this operation is efficient. We show in our experiments that we can quickly generate large datasets with our path supervision. We use our approach to annotate PathTrack, a crowdsourced MOT dataset 33 times larger than currently available ones. We improve current person-matching deep models using our data and that this has an impact on MOT accuracy. We release PathTrack to promote research in richer tracking models.

Acknowledgments This work was supported by the project VarCity (ERC #273940) and TRACE-Zurich by Toyota.

References

[1] Amazon Mechanical Turk. http://www.mturk.com. Accessed: November 2016.

[2] X. R. Alireza Fathi, Maria Florina Balcan and J. M. Rehg. Combining Self Training and Active Learning for Video Segmentation. In *BMVC*, 2011.

[3] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label Propagation in Video Sequences. In *CVPR*, 2010.

[4] S. Bandla and K. Grauman. Active Learning of an Action Detector from Untrimmed Videos. In *ICCV*, 2013.

[5] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016.

[6] R. Bellman. The theory of dynamic programming. *BAMS*, 1954.

[7] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008.

[8] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini. An Interactive Tool for Manual, Semi-automatic and Automatic Video Annotation. *CVIU*, 2015.

[9] W. Choi. Near-Online Multi-Target Tracking With Aggregated Local Flow Descriptor. In *ICCV*, December 2015.

[10] A. Ciptadi and J. M. Rehg. Minimizing human effort in interactive tracking by incremental learning of model parameters. In *ICCV*, 2015.

[11] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. Knowledge-driven Multimedia Information Extraction and Ontology Evolution. chapter A Survey of Semantic Image and Video Annotation Tools, pages 196–239. Springer-Verlag, Berlin, Heidelberg, 2011.

[12] R. Di Salvo, D. Giordano, and I. Kavasidis. A Crowdsourcing Approach to Support Video Annotation. In *VIGTA*, VIGTA '13, pages 8:1–8:6, New York, NY, USA, 2013. ACM.

[13] A. Ellis and J. Ferryman. PETS2010: Dataset and Challenge. *AVSS*, 00(undefined):143–150, 2010.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[15] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, pages 961–970, 2015.

[16] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. *Two-Granularity Tracking: Mediating Trajectory and Detection Graphs for Tracking under Occlusions*, pages 552–565. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[17] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, June 2016.

[18] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[19] P. Gil-Jiménez, H. Gómez-Moreno, R. J. López-Sastre, and S. Maldonado-Bascón. Geometric bounding box interpolation: an alternative for efficient video annotation. *EURASIP J. Image and Video Processing*, 2016:8, 2016.

[20] M. Gygli, Y. Song, and L. Cao. Video2GIF: Automatic Generation of Animated GIFs from Video. In *CVPR*, 2015.

[21] E. Hosoya, H. Sato, M. Kitabata, I. Harada, H. Nojima, and A. Onozawa. *Arm-Pointer: 3D Pointing Interface for Real-World Interaction*, pages 72–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[22] S. D. Jain and K. Grauman. Click Carving: Segmenting Objects in Video with Point Clicks. *CoRR*, abs/1607.01115, 2016.

[23] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. A Semi-automatic Tool for Detection and Tracking Ground Truth Generation in Videos. In *VIGTA*, VIGTA '12, pages 6:1–6:5, New York, NY, USA, 2012. ACM.

[24] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? In *ECCV*, London, UK, UK, 2002. Springer-Verlag.

[25] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The Visual Object Tracking VOT2015 Challenge Results. In *ICCV Workshops*, December 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[27] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised Object Discovery and Tracking in Video Collections. In *ICCV*, 2015.

[28] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. In *CVPR Workshop*, 2016.

[29] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv:1504.01942 [cs]*, Apr. 2015.

[30] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.

[31] W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim. Multiple Object Tracking: A Literature Review. *arXiv preprint arXiv:1409.7618*, 2014.

[32] S. Manen, R. Timofte, D. Dai, and L. V. Gool. Leveraging single for multi-target tracking using a novel trajectory overlap affinity measure. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016.

[33] P. Mettes, J. C. van Gemert, and C. G. M. Snoek. Spot On: Action Localization from Pointly-Supervised Proposals. In *ECCV*. 2016.

[34] D. Mihalcik and D. Doermann. The Design and Implementation of ViPER, 2003.

[35] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A Benchmark for Multi-Object Tracking. *arXiv:1603.00831 [cs]*, Mar. 2016.

[36] J. Niño-Castañeda, A. Frías-Velázquez, N. B. Bo, M. Slembrouck, J. Guan, G. Debard, B. Vanrumste, T. Tuytelaars, and W. Philips. Scalable Semi-Automatic Annotation for Multi-Camera Person Tracking. *IEEE Transactions on Image Processing*, 2016.

[37] O. S. P. Schallauer and H. Neuschmied. Efficient semantic video annotation by object and shot re-detection. In *SAMT*, 2008.

[38] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *CVPR*, 2016.

[39] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[40] M. Prince. Does active learning work? A review of the research. *JEE*, pages 223–231, 2011.

[41] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video, Feb. 2017.

[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[43] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[45] D. Shao, M. Rauter, and C. Beleznai. AVSS2011 demo session: Real-time human detection using fast contour template matching for visual surveillance. *AVSS*, 00:514, 2011.

[46] S. Vijayanarasimhan and K. Grauman. *Active Frame Selection for Label Propagation in Videos*, pages 496–509. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[47] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently Scaling Up Crowdsourced Video Annotation. *IJCV*, 101(1):184–204, Jan. 2013.

[48] C. Vondrick and D. Ramanan. Video Annotation and Tracking with Active Learning. In *NIPS*, 2011.

[49] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces. In *ECCV*, pages 610–623, Berlin, Heidelberg, 2010. Springer-Verlag.

[50] N. Wang and D. yan Yeung. Ensemble-Based Tracking: Aggregating Crowdsourced Structured Time Series Data. In T. Jebara and E. P. Xing, editors, *ICML*, pages 1107–1115. JMLR Workshop and Conference Proceedings, 2014.

[51] Y. Wei, J. Sun, X. Tang, and H.-Y. Shum. Interactive Offline Tracking for Color Objects. *ICCV*, 2007.

[52] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV 2015 - IEEE International Conference on Computer Vision*, pages 3164–3172, Santiago, Chile, Dec. 2015. IEEE.

[53] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *arXiv CoRR*, abs/1511.04136, 2015.

[54] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning From Massive Noisy Labeled Data for Image Classification. In *CVPR*, June 2015.

[55] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. LabelMe video: Building a video database with human annotations. In *ICCV*, pages 1451–1458. IEEE Computer Society, 2009.

[56] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, pages 1–8, June 2008.