WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection

Tatjana Chavdarova[1], Pierre Baqué[2], Stéphane Bouquet[2],
Andrii Maksai[2], Cijo Jose[1], Timur Bagautdinov[2], Louis Lettry[3],
Pascal Fua[2], Luc Van Gool[3], and François Fleuret[1]

[1]Machine Learning group, Idiap Research Institute & École Polytechnique Fédérale de Lausanne
[2]CVLab, École Polytechnique Fédérale de Lausanne
[3]Computer Vision Lab, ETH Zurich
{*firstname.lastname*}@{*idiap*[1],*epfl*[2],*vision.ee.ethz*[3]}.*ch*

Abstract

People detection methods are highly sensitive to occlusions between pedestrians, which are extremely frequent in many situations where cameras have to be mounted at a limited height. The reduction of camera prices allows for the generalization of static multi-camera set-ups. Using joint visual information from multiple synchronized cameras gives the opportunity to improve detection performance.

In this paper, we present a new large-scale and high-resolution dataset. It has been captured with seven static cameras in a public open area, and unscripted dense groups of pedestrians standing and walking. Together with the camera frames, we provide an accurate joint (extrinsic and intrinsic) calibration, as well as 7 *series of* 400 *annotated frames for detection at a rate of* 2 *frames per second. This results in over* 40 000 *bounding boxes delimiting every person present in the area of interest, for a total of more than* 300 *individuals.*

We provide a series of benchmark results using baseline algorithms published over the recent months for multi-view detection with deep neural networks, and trajectory estimation using a non-Markovian model.

1. Introduction

Pedestrian detection is an important computer vision problem with numerous applications in security, surveillance, robotics, autonomous driving, and crowdsourcing. The variation of pedestrians appearance greatly increases the difficulty of this problem. With the availability of large-scale monocular datasets of annotated pedestrians and the advances in detection algorithms, the accuracy of the pedestrian detectors has improved significantly in the past few years. Moreover, modern detection algorithms using deep learning allow us to learn discriminative features which are transferable across datasets. Impressively, recently developed deep learning based monocular detectors are approaching human-level performance [49] on common benchmark datasets [17].

However, many situations of practical interest require detection in highly crowded and cluttered scenes. Severe occlusions make monocular pedestrian detection insufficient in these scenarios. Luckily, in real-world applications, image feeds from multiple cameras with overlapping fields of view are often available. Most commonly, the cameras are positioned slightly above the average human height. Hence designing pedestrian detectors by exploiting multiple views and the geometry of the scene will provide reliable detection estimates in crowded scenes.

Surprisingly, to nowadays standards, there is no large scale and good quality public dataset that replicate this setup. The most frequently used one, PETS 2009 [21] is only in the order of several hundreds of frames long. The provided camera calibration is inaccurate which makes it difficult to exploit geometrical constraints jointly. Moreover, it is recorded in a so-called *actor set-up*, meaning that throughout the sequence, a very limited total number of different individuals actually appear. All other standard datasets are either much shorter and scarcely crowded [22, 48, 7], have a very constrained scenario [3] or use non-overlapping cameras [40].

The lack of such a dataset seriously limits the development of multi-camera detection methods. Recent improvements made by the community call for a more realistic and challenging benchmark that could be used to compare multi-camera detection methods. For example, [12] shows that utilizing the multi-camera input for deep learning based detectors improves both the accuracy and the classification confidence. However, these methods are severely limited by the lack of a large-scale dataset to train on without overfitting. As a direct consequence, most of the existing joint methods use *ad-hoc* techniques to combine information extracted using pre-trained monocular detectors.

To help in accelerating the research on methods taking advantage of the multi-camera set-ups, we introduce a *large-scale person dataset acquired with seven static cameras, with overlapping fields of view*. It captures a realistic unscripted scenario where pedestrians often occlude each other. We provide a very precise joint (extrinsic and intrinsic) calibration and synchronization of sequences from the 7 views as well as 7 series of 400 annotated frames for detection at a rate of 2 frames per second. This results in over 40 000 bounding boxes delimiting every person present in the area of interest, for a total of more than 300 individuals. The annotations of the individual tracks are provided both as 3D locations on the ground plane and 2D bounding-boxes projected in each of the 7 views. Although our dataset is designed to benchmark 3D multi-camera detection, it can also be used for monocular detection. In the monocular case, the size of our dataset increases seven-fold which makes it comparable to the widely used Caltech-USA dataset [15]. Compared to the latter, our dataset has *much higher image resolution*.

We make the source code of our web-based annotation platform public in order to encourage other researchers to collect and annotate other multi-camera datasets. Finally, we also provide annotations for evaluating camera calibration methods.

This paper is organized as follows: in Section 2 we make a review of multi-camera person datasets and related methods. Section 3 enumerates details of the new dataset, including our camera calibration procedure. We benchmark several state-of-the-art multi-camera detection methods in Section 4 and finally in Section 5 we present potential future research directions.

2. Related work

2.1. Datasets

In Table 1 we list the commonly used pedestrian datasets with a focus on the multi-camera ones. For a more exhaustive listing of the monocular datasets, we refer the reader to [16, chap. 2.4]. As *overlapping* we refer to multi-camera datasets whose camera's fields of view strictly overlap. The DukeMTMC [40] dataset does not belong to this category, as only 2 of its camera's fields of view slightly overlap.

The most widely used dataset with an overlapping camera set-up is the PETS 2009 S2.L1 [21] sequence. In part due to the presence of a slope in the scene, the provided calibration poses large homography mapping deterioration and inconsistencies when projecting $3D$ points across the views (as noted also in [38, p. 10], [23, p. 10], [12, p. 3]). Besides being a small scale dataset, the PETS 2009 S2.L1 is acquired in an actor set-up. Hence it does not allow for good generalization and fair benchmarking of appearance-based methods.

The three sequences shot at the EPFL campus [22]: Laboratory, Terrace and Passageway, as well as SALSA [3] and Campus [48] are overlapping multi-camera datasets as well. However, they have a small number of total identities and are relatively sparsely crowded. As we can see from Table 1, Laboratory, Terrace and Passageway are of small size and low image quality. In SALSA [3], a cocktail party of 30 minutes is recorded, where the people are static most of the time, making this dataset less challenging for tracking. Finally, Campus does not provide the annotations of the $3D$ locations of the people.

The EPFL-RLC [12] dataset demonstrates improved joint-calibration accuracy and synchronization compared to PETS. However, rather than providing a complete ground-truth, this dataset represents a collection of a balanced set of positive and negative multi-view annotations and is used for classification of a position as occupied by a pedestrian or not. Full ground-truth annotations are provided solely for a small subset of the last 300 of the total 8000 frames, originally used for testing [12]. Moreover, in comparison to ours, it is acquired with only three cameras which have a much more limited field of view. This results in a ∼10-fold smaller number of detections per frame on average.

Hence, WILDTRACK improves upon other multi-camera person datasets thanks to: (i) the high precision calibration and synchronisation between the cameras (see § 3.2), (ii) the large number of annotations that allows for developing deep learning based multi-view detectors. With regard to the state-of-the-art monocular datasets [15]: (i) it exceeds the total number of annotations; and (ii) the regions of interest (ROIs) are of significantly larger resolution.

2.2. Methods

Probabilistic Occupancy Map method (POM) [22] is a generative model which estimates the probabilities of occupancy on the ground plane by exploiting the geometrical constraints from multiple views. POM is formulated in the framework of mean-field inference which naturally handles the occlusions. To leverage time consistency it can be combined with a convex max-cost flow optimization [6] for tracking.

In [2], multi-view detection is re-casted as a linear in-

Table 1. Commonly used person datasets with a focus on the multi-camera ones. *FPS*, *pos* and *imp* stand for frames per second, positive images and image pairs, respectively. Where applicable, with '+' we denote the pre-defined splits to train and test data. See § 2.1.

Dataset	Resolution	Cameras	FPS	Mobile/Static	Overlapping	Video	IDs	Annotations	Size/Duration
INRIA [13]	high	n/a	n/a	n/a	n/a	No	-	1200+566	614+288 pos.
ETH [19, 20] (5 seq.)	640×480	$1^{\ddagger}$	15	M	n/a	Yes	-	2853 (∼4fps) & 10398 (15fps)	4203 frames

TUD-Brussels [46]	7720×576;
	640×480;
		1	1	M	n/a	Yes	-	1776+1326 (pos.)	1092+508 pos.imp.

Daimler [18]	640×480	1^{*}	n/a	M	n/a	No	n/a	2400+1600	-
Daimler-stereo [29]	640×480	$1^{*\ddagger}$	15	M	n/a	Yes	-	3915	15600+56500
Caltech-USA [15]	640×480	1	30	M	n/a	Yes	-	$\sim 2\cdot 10^4$(10fps)+$1\cdot 10^3$(1fps)	∼10 hours
KITTI [24]	1392×512	$4^{\dagger\ddagger}$	10	M	n/a	Yes	-	194+195 imp.	7 min.
APIDIS [14]	1600×1200	7	22	S	Yes	Yes	12	86870 (25fps)	1 min.

PETS 2009 [21]	768×576;
	720×576;
		7	7	S	Yes	Yes	19	4650 (7fps)	795 frames

DukeMTMC [40]	1920×1080	8	60	S	No	Yes	∼2000	4077132 (60fps); 9668 trajectories.	85 min.
Laboratory [22]	320×240	4	25	S	Yes	Yes	6	476 (1fps)	2.5 min.
Terrace [22]	320×240	4	25	S	Yes	Yes	9	1023 (1fps)	3.5 min.
Passageway [22]	320×240	4	25	S	Yes	Yes	13	226 (1fps)	20 min.
Campus [48]	1920×1080	4	25	S	Yes	Yes	25	240 (1fps)	4×4 min.
SALSA [3]	1024×768	4	15	S	Yes	Yes	18	1200 (0.3fps)	60 min.
EPFL-RLC [12]	1920×1080	3	60	S	Yes	Yes	-	∼3×2044a.+300frames	8000 frames
WILDTRACK (ours)	1920×1080	7	60	S	Yes	Yes	313	∼7×9518 (2fps)	∼60 min.

[*] No color channels.

[†] 2 color and 2 grayscale cameras.

[‡] Stereo camera(s).

verse problem. Their model is regularized by enforcing a sparsity constraint on the occupancy vector. It uses a dictionary whose atoms approximate multi-view silhouettes. To alleviate the need for the time demanding Lasso-based [11] computations, a regression model is derived by [25]. This model solely comprises of Boolean arithmetic and sustains the sparsity assumption of [2]. In addition, the iterative method of [2] is replaced with a greedy algorithm based on set covering.

In [38] the occlusions are modeled explicitly per view by a separate Bayesian Network. A multi-view network is then constructed by combining them, based on the ground locations and the geometrical constraints.

Considering crowd analysis, in [23] the multi-view image generation is modeled with a stochastic generative process of random crowd configurations and then maximum a posteriori (MAP) estimate is used to find the best fit with respect to the image observations.

The Deep Multi-camera Detection (DeepMCD) [12] method which integrates CNN features demonstrated state-of-the-art results and showed that the accuracy and the confidence of a CNN classifier increase as more views are used. To mitigate the data requirement problem and improve generalization, the authors first used the larger monocular dataset Caltech [15] to train a base network. The multi-view CNN architecture is adapted so as to use the weights from the base network and produces joint estimates by processing the multi-view streams in parallel. To increase the sharpness and the localization accuracy, [12] also proposes two particular schemes of multi-view hard negative mining to train such a model.

The Deep-Occlusion Reasoning method [5] uses a joint CNN-CRF architecture and Mean-Fields inference to produce a Probabilistic Occupancy Map (POM) as in [22] while leveraging discriminative features extracted by a CNN. It introduces a Higher Order CRF, where unary potentials are produced by ROI pooling CNN [39]. Higher-order potentials are computed as a measure of the consistency between pixel labels produced by a fully convolutional network and a generative model which accounts for geometry and occlusions.

Tracking multiple objects in multiple overlapping cameras, as well as tracking in a single view, mostly follows the tracking-by-detection paradigm [4]. However, due to the scarcity of datasets with multiple overlapping cameras, tracking multiple objects in multiple overlapping cameras has received relatively little attention compared to tracking from a single view. For example, 3DMOT2015 benchmark [31] for 3D tracking lists only a couple methods that exceed a simple linear programming baseline that was proposed with the benchmark. [32] models the motion of people using social forces, and solves a linear program to es-

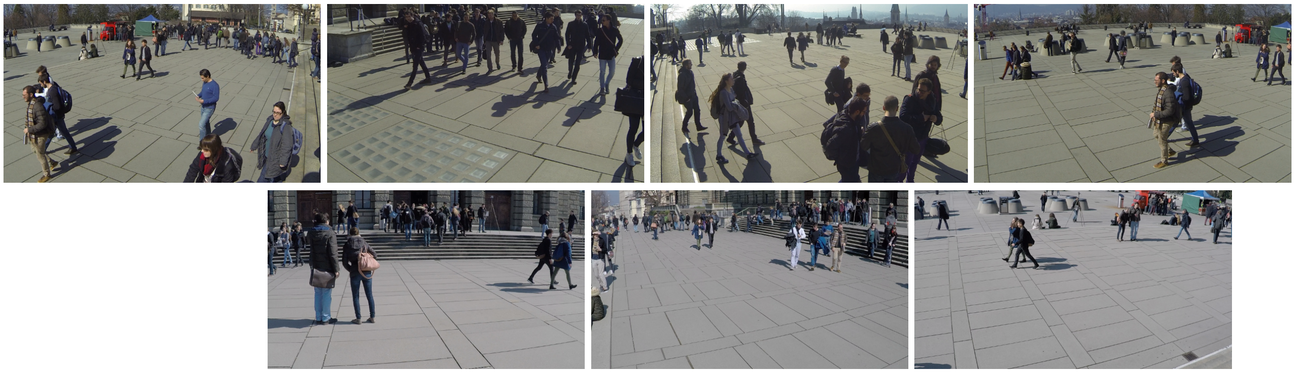


Figure 1. Synchronized corresponding frames from the seven views. Four *GoPro Hero 3* and three *GoPro Hero 4* were used, frames of which are shown in the top and bottom row, respectively.

timate the tracks over the whole sequences. [30] uses joint probabilistic data association for online tracking. The approach of [7] can be used both for regularizing detections and tracking. It formulates tracking as a problem of finding K-Shortest paths in a graph of detections.

Some of the recently proposed approaches, such as [35] or [42] can be applied for multiple object tracking in 3D. [35] learns weights of the Recurrent Neural Network (RNN) to predict the motion of objects for data association. [42] learns several RNNs to combine motion, appearance, and social information throughout time for data association. While both of these approaches could be trained for 3D scenes, combining the appearance information from multiple cameras for [42] is non-trivial. Training the model of [35] requires a lot more annotated data than what is currently available for 3D tracking. A recently proposed approach of [34] can be used as a post-processing step to improve the results of other tracking approaches by learning the patterns of human motion in specific scenes and modifying the tracks to follow such patterns.

3. The dataset

3.1. Hardware and data acquisition

Hardware. The dataset was recorded using seven statically positioned HD cameras. In particular, we used four *GoPro Hero 3* and three *GoPro Hero 4* cameras (Fig. 1). All the seven sequences are of resolution 1920×1080 pixels and were recorded with a frame rate of 60 frames per second (fps). The synchronization between the seven sequences was obtained with $\sim$50 ms accuracy (the precision of which can be observed in Fig. 4).

Camera placement and layout. The camera layout is highly overlapping as shown in Fig. 1, and the cameras are positioned above the humans' average height. In Fig. 2 we illustrate from a top view perspective the amount of overlap between the fields of view of the cameras, where the darker the shading, the higher the number of cameras that capture that area. To obtain this illustration, we considered points on the ground plane and counted the number of cameras for which a given point is in their field of view. The circles in Fig. 2 denote the position of the cameras.

Data acquisition. The data acquisition took place in front of the main building of the university ETH Zurich in Switzerland, during nice weather conditions.

3.2. Calibration of the cameras

To calibrate the cameras we used the Pinhole camera model [45], due to its widespread usage and support in multiple libraries, including OpenCV [9]. Primarily, we obtained the intrinsic and the extrinsic parameters, and for the latter, we used points on the ground whose distance was measured by hand.

We put a major focus on providing jointly optimal $3D$ reconstruction between the cameras. To this end, we manually annotated precisely $|\mathcal{D}| = 1398$ $3D$ points by marking corresponding $2D$ points across the seven views and throughout multiple frames; which we used to perform Bundle adjustment [44] as we explain below.

Let $\mathbf{I}$ and $\mathbf{E}$ denote the intrinsic and the extrinsic parameters of all of the cameras, respectively. Given a dataset $\mathcal{D}$ of corresponding projections across the views, more precisely: $\mathcal{D} = \{p_i\}$, where $p_i = \{p_i^1 \dots p_i^C\}$, with $p_i^c \in \mathbb{R}^2$ and C denoting the number of cameras, the goal is to find projection matrices P_c whose parameters are contained in $\{\mathbf{I}, \mathbf{E}\}$ and the $3D$ points $M = \{m_i\}$, $m_i \in \mathbb{R}^3$, $i = 1, \dots, |\mathcal{D}|$, such that:

$$\mathbf{I}^*, \mathbf{E}^* = \underset{\mathbf{I},\mathbf{E},M}{\operatorname{argmin}} \sum_{i=1}^{|\mathcal{D}|} \sum_{c=1}^{C} w_i^c \, \|p_i - P_c\, m_i\|^2, \qquad (1)$$

where $\|\cdot\|$ denotes the Euclidean image distance, and w_i^c is the indicator variable equal to 1 when the point p_i is visible in view c, and equal to 0 otherwise. In other words,

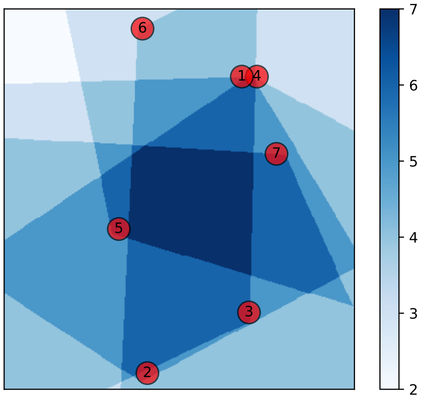


Figure 2. The overlap between the cameras' fields of view (top view). See § 3.1.

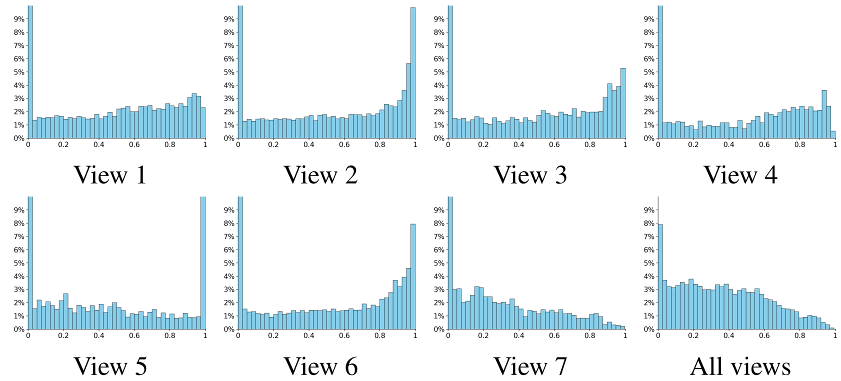


Figure 3. Percentage of examples that are occluded within certain range: x-axis–range of normalized occlusion, y-axis percentage of examples within that range. See § 3.4.

we formulated the calibration as a non-linear least squares problem, where the error is the squared L_2 norm of the difference between the observed feature location and the projection of the corresponding 3D point on the image plane of the camera.

In accordance with the selected model, the set $\{\mathbf{I}, \mathbf{E}\}$ in our implementation consists of 7×15 parameters for each camera: 3 for rotation, 3 for translation, 2 for focal length (x and y), 2 for the principal point, 3 for radial distortion and 2 for tangential distortion. To optimize Eq. 1, we used the open source C++ library *Ceres* [1] (see App. 3 for further details on the implementation).

We experimented with fixing one of the sets of parameters and sequentially optimizing each. Nonetheless, jointly optimizing **I** and **E** as formalised in Eq. 1 provided best results and this final step significantly improved the joint projection accuracy.

Fig. 4 depicts cropped regions taken from synchronized images and of a different view. To illustrate the precision of the camera calibration, we first manually marked a point projections in two of the views, shown in blue color in the left column of Fig. 4. Using the provided camera calibration, we then compute the $3D$ location of the point as an intersection between the rays defined by those projections. Finally, the resulting $3D$ point is projected in the remaining views, displayed as red points in Fig. 4.

3.3. Annotation process

To make sure that our annotations are as precise as possible, we made use of the accurate camera calibration. Namely, we formulated the annotation task as adjusting the position of a $3D$ cylinder on the ground, such that its projections (rectangles) across *all* of the views overlap the person being marked by the annotator. In summary, we considered this approach as: (i) more time-effective, as one has to perform one adjustment instead of marking bounding boxes and adjusting each separately; as well as that it allows for obtaining (ii) higher precision, since the position is jointly adjusted. The latter follows from the fact that the best position of a bounding box when annotating in one view is likely to be ambiguous, and in cases of more severe occlusions, it could also be infeasible to guess it.

To this end, we designed a new multi-camera annotation tool, whose Graphical User Interface (GUI) is illustrated in Fig. 5. It was written using Python, Django, and Javascript. The tool was deployed through Amazon Mechanical Turk [10] so that external people would be remunerated to help us carry out this time-consuming procedure. To ensure that profit is not prioritized over the accuracy and the precision of the annotations, we were highly involved in the process, and particular annotators were carefully selected. For details regarding how the annotation process was carried out, and the file format of the annotations, please refer to App. 1 and App. 2, respectively.

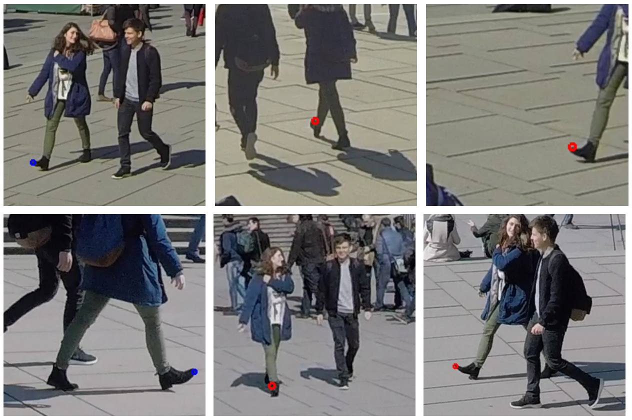
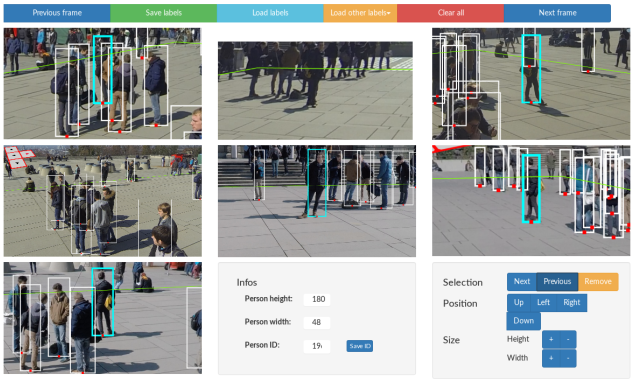


Figure 4. Illustration of the camera calibration precision (best seen in color). See § 3.2.



Infos

Person height: 180

Person width: 48

Person ID: 19

Figure 5. GUI of our multi-camera annotation tool (best seen in color). The bounding boxes being adjusted (displayed in cyan color) are zoomed-in in those views where the person is visible.

3.4. Statistics

Annotated frames. Following practices amassed in monocular pedestrian detection with CNNs, we used frame rate of 10 fps to extract the frames. Herein, as a frame I_t we refer to a set of synchronized images from the seven views (C=7), *i.e.* $\{I_t^1, \dots, I_t^C\}$. We provide annotations for the first 2000 frames. However, in our observations annotating every 5th suffices the speed of movement of the persons, and hence interpolation can be used to further enlarge the dataset. To summarize, this amounts to total of 400 frames at 2 fps, or alternatively 7×400 manually annotated images. Bellow we always list total number of annotations at 2 fps.

Multi-view annotations. There are 9518 multi-view single person annotations in total. In Fig. 6, we illustrate an example of a multi-view annotation of our dataset, visible in all of the seven views at the same time. On average, each frame I_t captures 23.8 people. Considering a frame rate of 2 fps, each person is seen in $30.41(47.87)$ frames, with a mode of 22 frames (see App. 4).

Monocular annotations. Since a multi-view detection may not be visible in each of the cameras' fields of view, the number of monocular examples is precisely 8731, 7875, 6703, 2239, 3920, 9408, 3731, respectively for each view. This amounts to a total of 42607 detections at 2 fps.

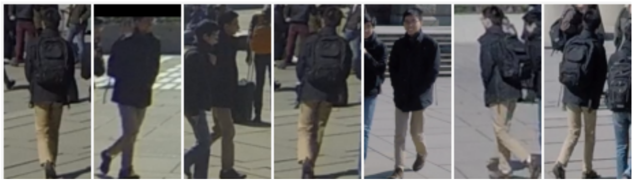


Figure 6. An example of one positive multi view annotation.

Occlusion levels. As the annotations are obtained in $3D$, the occlusion of each pedestrian in each view can be automatically obtained. Following the work of [15] in Fig. 3 we illustrate the level of occlusions per each view separately, by calculating the number of occluded pixels over the number of total pixels for each detection. Similarly, we calculate such normalized occlusion levels for the multi-view examples across all of the views. As can be noticed from Fig. 3, if multi-view samples are used, the probability that a person will be completely occluded in all of the views simultaneously significantly goes down.

4. Benchmarks

In the sequel, we evaluate detection and tracking methods. In our experiments, we used a frame rate of 2 fps.

4.1. Evaluation protocol

We compute false positives (FP), false negatives (FN) and true positives (TP) by assigning detections to ground truth using Hungarian matching. Since we operate on the ground plane, we impose that a detection can be assigned to a ground truth annotation if and only if it is closer than a distance r. Given FP, FN and TP, we calculate:

- **Multiple Object Detection Accuracy (MODA)** accounting for the normalized missed detections and false positives, as well as the **Multiple Object Detection Precision (MODP)** metric which assesses the localization precision [28].
- **Precision & Recall**. We estimate the empirical precision and recall, calculated by P=$TP/(TP$+$FP)$ and R=$TP/(TP$+$FN)$ respectively.

We report MODP, Precision, and Recall for radius r=0.5m, which roughly corresponds to the width of a human body.

Unless otherwise emphasized, the used metrics are always calculated in terms of the Euclidean distance between the detection and the annotation on the ground plane in world coordinates, or alternatively from top-view. Thus note that such metrics are unforgiving in terms of projection errors as we measure distances on the ground plane, which would not be the case if we evaluated overlap in the image plane as is often done in the monocular case.

Regarding multiple object tracking, we report metrics which are also reported for MOTChallenge benchmark [31]: a set of CLEAR MOT [8] metrics, as well as identity-aware metrics of [41]. For evaluation we used the devkit provided with [31], and we similarly report metrics for radius r=1m. Below we briefly review some of the reported tracking metrics:

- **Multiple Object Tracking Accuracy (MOTA)** accounts for missed detections (**False Negatives (FN)**),

False Positives (FP), and **Identity Switches (IDs)** between current and next frame of tracking. We also report **Multiple Object Tracking Precision (MOTP)**.

- **Mostly Tracked (MT), Partially Tracked (PT), Mostly Lost (MT)** trajectories, as well as the number of **Fragmentations (FM)**.

- **IDF1** accounts for missed detections, false positives, and identity switches throughout tracking, after finding a global one-to-one assignment between ground truth and predicted trajectories. It is calculated similarly to F1 score, using the **Identity Precision (IDP)** and **Identity Recall (IDR)** as proxies for precision and recall.

4.2. Evaluated methods

We evaluated the following detection methods:

- **POM-CNN.** The multi-camera detector [22] described in § 2.2, uses background subtraction pre-processing and takes such segmented images as input. In its original implementation the input is obtained using traditional algorithms [50, 36]. Hence, for a fair comparison reflecting the progress that has occurred since then, we use a CNN-based background subtraction [33].

- **DeepMCD** (described in § 2.2), which is an end-to-end deep learning method. We used its implementation that uses GoogLeNet [43] by: (i) testing on the WILDTRACK dataset with the provided pre–trained model on the PETS dataset–Pre-DeepMCD; as well as (ii) training solely the top classifier on the WILDTRACK dataset–TopDeepMCD. Its implementation uses monocular Non Maxima Suppression (NMS) and the performance measures are calculated using the first view, rather then the world coordinates.

- **ResNet-DeepMCD & DenseNet-DeepMCD** are our implementations of DeepMCD in PyTorch [37], which use ResNet-18 [26] and DenseNet-121 [27], respectively. As WILDTRACK is of larger size, we omitted the step of pre-training on the Caltech dataset. Our implementation uses top-view NMS to select the final detections of the candidates while prioritizing the detections with higher probability estimates, as opposite to [12]. We use NMS threshold of 0.4 and r=0.5m. We used 90% and 10% of the frames for training and testing, respectively and grid density of 60×180. Analogously, we also train a monocular detector ResNet-18 [26] while using samples from the training frames from all of the views, denoted as **ResNet-Monocular**.

- **Deep-Occlusion** (see § 2.2), which is a hybrid CNN-CRF method to use information about geometry and calibration while leveraging on the discriminative power of a pre-trained monocular CNNs.

- **RCNN-projected.** The recent work of [47] proposes a MCMT tracking framework that relies on a powerful CNN for detection purposes [39]. Since the code of [47] is not publicly available, we reimplemented their detection methodology *without* the tracking component for a fair comparison with the detection methods that operate on images acquired at the same time. More precisely, we first run the 2D detector proposed by [39] on each image. We then project the bottom of the 2D bounding box onto the ground reference frame as in [47] to get 3D ground coordinates. Finally, we cluster all the detections from all the cameras using 3D proximity to produce the final set of detections.

For multiple object tracking, we provide results of the following methods:

- **KSP** is a simple baseline approach of [7] which finds the most likely sequence of occupancies of ground floor locations of POM given probabilities of occupancies of each location given by a detector. To do so it solves the problem of finding the most likely trajectories given the detections as finding the K-Shortest Paths in a graph. This method does not use appearance information for tracking, but is a suitable baseline as it can be applied out of the box in multi-camera scenario.

- **ptrack** is a recent approach of [34] which improves the results of other tracking approaches by learning the motion patterns of the scene and modifying the tracks of people so as to follow those patterns. This is suitable for our scenario both because in our scene there are several clearly identifiable patterns of motion, as well as because it can handle tracking in the ground plane. We therefore use this approach on top of the trajectories found by **KSP**.

4.3. Benchmark results

Detection. The results of the enumerated methods in § 4.2 are given in Tab. 3. We observe that joint utilization of the multiple views largely improves detection performance. More explicit occlusion reasoning further improves the MODA metric, even for small values of r, Fig. 7. In Tab. 4 we list the classification results obtained when training a monocular ResNet-18 using samples extracted from all of the seven views, as well as multi-view training. The results indicate that even though the size of the dataset is seven-fold larger for the monocular case, using the multiple views increases both the accuracy and the confidence of the classifier.

Table 2. Benchmark tracking results on the WILDTRACK dataset.

Method	IDF1	IDP	IDR	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
DeepOcclusion+KSP	73.2	83.8	65.0	49	79	43	1095	7503	85	92	69.6	61.5
DeepOcclusion+KSP+ptrack	78.4	84.4	73.1	72	74	25	2007	5830	103	95	72.2	60.3

Table 3. Benchmark detection results on WILDTRACK.

Method	MODA	MODP	Precision	Recall
Deep-Occlusion+KSP	0.752	-	-	-
Deep-Occlusion	0.741	0.538	0.95	0.80
ResNet-DeepMCD	0.678	0.642	0.85	0.82
DenseNet-DeepMCD	0.635	0.666	0.87	0.74
POM-CNN	0.232	0.305	0.75	0.55
RCNN-projected	0.113	0.184	0.68	0.43
Pre-DeepMCD	0.334*	0.528*	0.93	0.36
Top-DeepMCD	0.601*	0.642*	0.80	0.79
ResNet-View 1	-1.823	0.598	0.26	1.00
ResNet-View 2	-1.050	0.607	0.32	0.99
ResNet-View 3	-1.036	0.583	0.32	0.98
ResNet-View 4	-0.251	0.723	0.42	0.71
ResNet-View 5	0.466	0.623	0.67	0.91
ResNet-View 6	-1.841	0.591	0.26	1.00
ResNet-View 7	-0.122	0.701	0.47	0.97

* Monocular calculation of the metric, using the first view.

Table 4. Classification results (accuracy and area under ROC curve) on the test frames of the WILDTRACK dataset using ResNet-18 [26]. The results obtained for monocular classification are averaged over all of the views.

Training	Accuracy (%)	AUC
Monocular	84.57 (1.718)	0.91 (0.028)
Multi-view	95	0.95

Tracking. Tab. 2 lists benchmark results of the tracking methods described in § 4.2 (see App. 6 for additional results). Parameters of the methods were optimized for IDF1 metric on the same training data as the appropriate detector was trained on. As shown, the dataset presents a significant tracking challenge, with IDF1 metric results lower than those seen on the [31] benchmark on [40] dataset, where each individual camera mostly observes a separate, and somewhat simpler scene.

5. Discussion

The development of new multi-view people tracking methods is hampered by the surprising lack of appropriate datasets. We provide a new large scale, high-resolution, and highly accurately calibrated multi-camera pedestrian dataset, which is more realistic than any of the previously published ones.

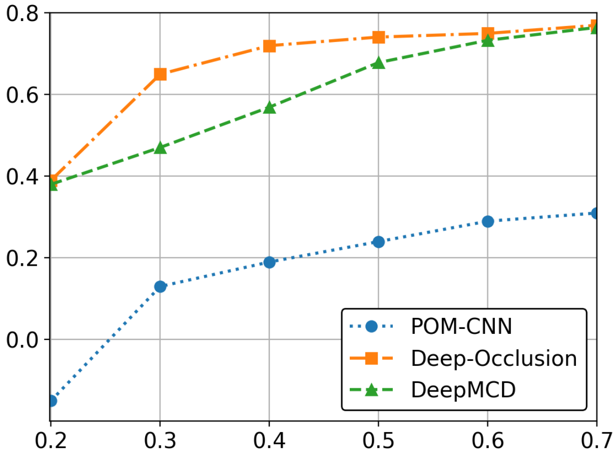


Figure 7. Benchmark of the multi-view detectors using the MODA metric (y-axis) for different radius r (x-axis) on WILDTRACK.

Our initial benchmarks show that deep learning person detection indeed largely benefits from a multi-camera set-up and this motivates further work in that direction. Our dataset will motivate and facilitate such research. While performance of monocular pedestrian detectors saturates on common benchmarks, our densely crowded realistic set-up, which results in complex dynamics and constant occlusions among persons, and the high resolution, will prove useful for further improving monocular detection, as well as other problem of inference such as tracking, or crowd analysis at large.

Finally, in addition to the large number of individual detections, the WILDTRACK dataset also has a large fraction of unlabelled frames. This will be precious for unsupervised methods, with the possibility to be benchmarked on the annotated portion.

Acknowledgment

This work was supported by the Swiss National Science Foundation, under grant CRSII2-147693 "WILDTRACK" and by the Hasler Foundation through the "MEMUDE" project. We gratefully acknowledge NVIDIA's support through their academic GPU grant program. We also thank Salim Kayal and Florent Monay for their help and insightful discussions regarding the calibration of the cameras.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 5

[2] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41(1-2):39–58, 2011. 2, 3

[3] X. Alameda-Pineda, J. Staiano, R. Subramanian, L. Batrinca, E. Ricci, B. Lepri, O. Lanz, and N. Sebe. Salsa: A novel dataset for multimodal group behavior analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(8):1707–1720, 2016. 1, 2, 3

[4] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 3

[5] P. Baque, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-People tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 271–279, 2017. 3

[6] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Proceedings of the 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS)*, pages 1–8, 2009. 2

[7] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011. 1, 4, 7

[8] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246309, 2008. 6

[9] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[10] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011. PMID: 26162106. 5

[11] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, Dec 2008. 3

[12] T. Chavdarova and F. Fleuret. Deep multi-camera people detection. In *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 848–853, 2017. 2, 3, 7

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893 vol. 1, 2005. 3

[14] C. De Vleeschouwer, F. Chen, D. Delannay, C. Parisot, C. Chaudy, E. Martrou, A. Cavallaro, et al. Distributed video acquisition and annotation for sport-event summarization. In *NEM summit 2008:: Towards Future Media Internet*, 2008. 3

[15] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–311, 2009. 2, 3, 6

[16] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(4):743–761, 2012. 2

[17] X. Du, M. El-Khamy, J. Lee, and L. S. Davis. Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 953–961, 2017. 1

[18] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2179–2195, 2009. 3

[19] A. Ess, B. Leibe, and L. V. Gool. Depth and appearance for mobile scene analysis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. 3

[20] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 3

[21] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *Proceedings of the 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS)*, pages 1–6, 2009. 1, 2, 3

[22] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282, 2008. 1, 2, 3, 7

[23] W. Ge and R. T. Collins. Crowd detection with a multiview sampler. 2010. 2, 3

[24] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[25] M. Golbabaee, A. Alahi, and P. Vandergheynst. Scoop: A real-time sparsity driven people localization algorithm. *Journal of Mathematical Imaging and Vision*, 48(1):160–175, 2014. 3

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 7, 8

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[28] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang. Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. 31(2):319–336, 2009. 6

[29] C. Keller, D. Llorca, and D. Gavrila. Dense stereo-based roi generation for pedestrian detection. In *Pattern Recognition*, volume 5748 of *Lecture Notes in Computer Science*, pages 81–90. Springer Berlin Heidelberg, 2009. 3

[30] T. Klinger, F. Rottensteiner, and C. Heipke. Probabilistic multi-person localisation and tracking in image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:73–88, 2017. 4

[31] 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. arXiv: 1504.01942. 3, 6, 8

[32] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 120–127. IEEE, 2011. 3

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 7

[34] A. Maksai, X. Wang, F. Fleuret, and P. Fua. Non-markovian globally consistent multi-object tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2544–2554, 2017. 4, 7

[35] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, pages 4225–4232, 2017. 4

[36] N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. 22(8):831–843, 2000. 7

[37] A. Paszke, S. Gross, S. Chintala, and G. Chanan. PyTorch. `https://github.com/pytorch/pytorch`, 2017. 7

[38] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang. Robust multiple cameras pedestrian detection with multi-view bayesian network. *Pattern Recognition*, 48(5):1760–1772, May 2015. 2, 3

[39] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015. 3, 7

[40] 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. 1, 2, 3, 8

[41] 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*, pages 17–35. Springer, 2016. 6

[42] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 4

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[44] Wikipedia. Bundle adjustment — wikipedia, the free encyclopedia, 2017. [Online; accessed 13-July-2017]. 4

[45] Wikipedia. Pinhole camera model — wikipedia, the free encyclopedia, 2017. [Online; accessed 12-July-2017]. 4

[46] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 794–801, 2009. 3

[47] Y. Xu, X. Liu, Y. Liu, and S. Zhu. Multi-View People Tracking via Hierarchical Trajectory Composition. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4256–4265, 2016. 7

[48] Y. Xu, X. Liu, Y. Liu, and S. C. Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4256–4265, 2016. 1, 2, 3

[49] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. How far are we from solving pedestrian detection? In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[50] F. Ziliani and A. Cavallaro. Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection. 1999. 7