Multi-view Tracking Using Weakly Supervised Human Motion Prediction

Martin Engilberge
EPFL, Lausanne, Switzerland
martin.engilberge@epfl.ch

Weizhe Liu
Tencent XR Vision Labs
weizheliu@tencent.com

Pascal Fua
EPFL, Lausanne, Switzerland
pascal.fua@epfl.ch

Abstract

Multi-view approaches to people-tracking have the potential to better handle occlusions than single-view ones in crowded scenes. They often rely on the tracking-by-detection paradigm, which involves detecting people first and then connecting the detections. In this paper, we argue that an even more effective approach is to predict people motion over time and infer people's presence in individual frames from these. This enables to enforce consistency both over time and across views of a single temporal frame. We validate our approach on the PETS2009 and WILDTRACK datasets and demonstrate that it outperforms state-of-the-art methods.

1. Introduction

When it comes to tracking multiple people, tracking-by-detection [2] has become a standard paradigm and has proven effective for many applications such as surveillance or sports player tracking. It involves first detecting the target objects in individual frames, associating these detections into short but reliable trajectories known as tracklets, and then concatenating them into longer trajectories [40, 23, 29, 64, 31, 41, 50, 46, 30, 56, 19]. The grouping of detections into full trajectories can also be formulated as the search for multiple min-cost paths on a graph [9, 58]. More recently, tracking-by-regression [66, 61] has been advocated as a potential alternative. It readily enables tracking while being end-to-end differentiable, unlike the detection-based approaches.

However, these single-view tracking techniques can be derailed by occlusions and are bound to fragment tracks when detections are missed. Using multiple cameras is one way to address this problem, especially in locations such as sports arenas where such a setup can be installed once and for all [8, 60, 12]. This can be highly effective but can still fail when occlusions become severe. This is in part because detection algorithms typically operate on single frames and fail to exploit the fact that we have videos that exhibit time consistency. In other words, if someone is detected in one frame, chances are they should be found at a neighboring location in the next frame, as depicted by Fig. 1. Furthermore, even though people's motion and scale is consistent across views, that consistency is rarely enforced when fusing results from different views.

Figure 1: **Predicting human motion.** Our model learns to detect people by predicting human flows. It generates the probabilities that a person moves from one location to one of its eight neighbors or itself, depicted by the yellow grid in the top image. The white triangles depict detections in the ground-plane while the green ones denote the predicted location at the next time step. The bottom image corresponds to the top view re-projection of the top one, the blue arrows illustrate the motion predicted by our model. On both images the region of interest is overlaid in yellow. People outside of that region are ignored.

Project code at https://github.com/cvlab-epfl/MVFlow

In this paper, we address both these issues by training networks to detect *flows of people* across images. Our model directly leverages temporal consistency using self-supervision across video frames. Furthermore, it can fuse information from different cameras while retaining spatial consistency for human position from different viewpoints. As a result, we outperform state-of-the-art multi-view techniques [24, 25, 12] on challenging datasets.

2. Related works

Early work on tracking objects in video sequence rely on model evolution technique which focuses on tracking a single object using gating and Kalman filtering [44]. Because of their recursive nature, they are prone to errors such as drift, which are difficult to recover from. Therefore, this method is largely replaced by tracking-by-detection techniques which have proven to be effective in addressing people tracking problems. In this section we first briefly introduce previous work of tracking-by-detection and then discuss previous work in modeling human motions.

2.1. Tracking-by-Detection.

Tracking-by-detection [2] aims to track objects in video sequences by optimizing a global objective function over many frames given frame-wise object detection information. They reply on Conditional Random Fields [33, 62, 43], Belief Propagation [63, 13], Dynamic or Linear Programming [5, 51], or Network Flow Programming [1, 15]. Some of these algorithms follow the graph formulation with nodes as either all the spatial locations where an object can be present [18, 9, 8] or only those where a detector has fired [27, 55, 52, 6].

Among these graph-based approaches, the K-Shortest Paths (KSP) algorithm [9] works on the graph of all potential locations over all time instants, and finds the ground-plane trajectories that yield the overall minimum cost. This optimality is achieved at the cost of multiple strong assumption about human motion, in particular it treats all motion direction as equiprobable. Similar to the KSP algorithm, the Successive Shortest Paths (SSP) approach [48] links detections using sequential dynamic programming. [36] extends this SSP approach with bounded memory and computation which enables tracking in even longer sequences. The memory consumption is further reduced in [58] by exploiting the special structures and properties of the graphs formulated in multiple objects tracking problems. More recent work [59] proposes to learn a deep association metric on a large-scale person re-identification dataset which enables reliable people tracking in long video sequence.

Occlusion makes it extremely challenging to achieve reliable object tracking in long sequences. Some algorithms address this by leveraging multiple viewpoints, some approaches first detect people in single images before reprojecting and matching detections into a common reference frame [60, 18]. [4] propose to directly combine view aggregation and prediction with a joint CNN/CRF. More recently [25] proposed to use spatial transformer networks [26] to project feature representation in the ground plane resulting in an end-to-end trainable multi-view detection model. [53] proposed to combine multiple views using an approximation of a 3D world coordinate system by projecting features in planes at different height levels. Finally [24] proposed to use multi-view data augmentation combined with a transformer architecture to fuse ground plane features from multiple points of view and obtains state-of-the-art results for multiple object detection on the WILDTRACK dataset [12].

2.2. Modeling human motion

Modeling human motion as flow when tracking people has been a concern long before the advent of deep learning [47, 57, 11, 37, 14, 34, 20, 10, 45, 42, 3, 9]. For example, in [9], people tracking is formulated as multi-target tracking on a grid and gives rise to a linear program that can be solved efficiently using the K-Shortest Path algorithm [54]. The key to this formulation is to optimize the people flows from one grid location to another, instead of the actual number of people in each grid location. In [48], a people conservation constraint is enforced and the global solution is found by a greedy algorithm that sequentially instantiates tracks using shortest path computations on a flow network [65]. Such people conservation constraints have since been combined with additional ones to further boost performance. They include appearance constraints [7, 16, 8] to prevent identity switches, spatiotemporal constraints to force the trajectories of different objects to be disjoint [22], and higher-order constraints [11, 14]. More recent work extends this flow formulation with deep learning [38, 39] to formulate people as people flows which contributes to reliable people counting in even dense regions. However, none of these methods leverage such people flow formulation to address tracking problems with deep neural networks. These kinds of flow constraints have therefore never been used in a deep people tracking context.

3. Approach

Most recent approaches rely on the *tracking-by-detection* paradigm. In its simplest form, the detection step is disconnected from the association step. In this section, we propose a novel method to bring closer those two steps. First we introduce a detection network predicting people flow in a weakly supervised manner. Then we show how we modify existing association algorithms to leverage predicted flows to generate unambiguous tracks.

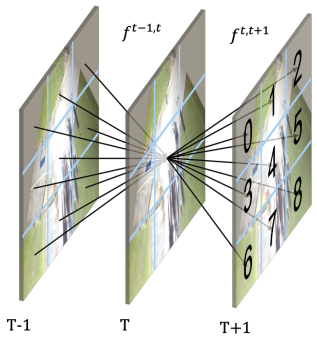

Figure 2: **Grid flow representation** For each location i we predict the probability that a person is moving from i to one of its eight neighbors or itself in the next time step. Detection probability at a given location at time t can be computed by summing the nine outgoing flows or the nine flows reaching that location from t-1.

3.1. Formalism

Let us consider a multi-view video sequence $S = \{\mathbf{I}^1, \mathbf{I}^2, ..., \mathbf{I}^{T-1}, \mathbf{I}^T\}$ consisting of T time steps. Each time step $\mathbf{I}^t = \{\mathbf{I}_1^t, \dots \mathbf{I}_V^t\}$ consists of a set of synchronized frames taken by V cameras with overlapping fields of view. For each camera the calibration $\mathbf{C}_v$ is known and contains both intrinsic and extrinsic parameters. Each frame $\mathbf{I}_v^t \in (0, 255)^{W \times H \times 3}$ is a color image with a spatial size (W,H).

To combine multiple views we choose to work in the common ground plane. For each frame we define $\mathbf{G}_v^t = P(\mathbf{I}_v^t, \mathbf{C}_v)$ as the projection of frame $\mathbf{I}_v^t$ on the ground plane using the projection function P producing $\mathbf{G}_v^t \in (0, 255)^{w \times h \times 3}$ with (w, h) the spatial size of the ground plane image.

Finally, we adopt similar grid world formalism as previous work [9]. At each time step t we discretize the physical ground plane to form a grid of $w \times h$ cells, giving us a scene representation of dimensionality $w \times h \times t$ for a full sequence.

3.2. People Flow

Given a pair of consecutive multi-view time steps we define the human flow $f^{t,t+1}$ as follows: For a given location i, the flow $f_{i,j}^{t,t+1}$ is the probability that a person in cell i at time t moves to location j at time $t+1$. Where $j \in \mathcal{N}(i)$ is a neighbor of i. Concretely, for each cell in the ground plane we represent people flow by a 9-dimensional vector of probability (one dimension per neighbors of that cell). The grid representation and the definition of neighborhood are illustrated in Fig. 2

To accurately model human motion, the flow need to respect three constraints:

First, people conservation constraints, if a person is present at time t, he should be present at time $t+1$ in the same location, or in a neighboring one. In other words, if we consider three time steps $\mathbf{I}^{t-1}$, $\mathbf{I}^t$, and $\mathbf{I}^{t+1}$ the sum of the incoming flow in cell j between time $t-1$ and t should be equal to the sum of outgoing flow between time t and $t+1$. More formally it reads:

$$\sum_{i \in \mathcal{N}(j)} f_{i,j}^{t-1,t} = x_j^t = \sum_{k \in \mathcal{N}(j)} f_{j,k}^{t,t+1}. \quad (1)$$

The sums of the flow are equal to x_j^t the probability that there is a person in j at time t.

Second, non-overlapping constraints, at any time there should be at most one person in every cell.

$$\forall k, t, \sum_{j \in \mathcal{N}(k)} f_{k,j}^{t,t+1} \leq 1. \quad (2)$$

Finally, a temporal consistency constraint, if we reversed a sequence, the flow should be the same with the flow direction being flipped.

$$f_{i,j}^{t-1,t} = f_{j,i}^{t,t-1}. \quad (3)$$

Reconstructing detection from human flow is trivial (Eq. (1)) and has a unique solution, on the other hand, generating flow from detection can have multiple solutions. Therefore we introduce Multi-View FlowNet (MVFlow), trained to generate human flow. By predicting flow instead of detection our model is able to take advantage of the asymmetric mapping between flow and detection. It learns to predict flow in a weakly supervised manner, using only detection annotation. Enforcing flow constraints in Eq. (1), Eq. (2) and Eq. (3) also serves as a regularization for the final detection. Predictions are temporally consistent and represent natural human motion.

3.3. Multi-View architecture

In this section we detail the architecture of MVFlow, our multi-view detection model.

The proposed model consists of 5 steps and takes as input a pair of multi-view frames. Each frame is processed by a ResNet. The resulting features are projected in the ground plane. Ground features from the same point of view at time t and $t+1$ are aggregated. Afterwards, the spatial aggregation module combines the features from the different points of view into human flow. Detection predictions are reconstructed from the flow for both time steps. The 5 steps are illustrated in Fig. 3. More formally, the model is defined as follows:

$$\begin{matrix}\mathbf{I}^t \\ \mathbf{I}^{t+1}\end{matrix} \overset{g_{\boldsymbol{\theta}_0}}{\longmapsto} \begin{matrix}\mathbf{F}^t \\ \mathbf{F}^{t+1}\end{matrix} \overset{\mathbf{C}}{\longmapsto} \begin{matrix}\mathbf{G}^t \\ \mathbf{G}^{t+1}\end{matrix} \overset{c_{\boldsymbol{\theta}_1}}{\longmapsto} \mathbf{G}^{t,t+1} \overset{\mathrm{s}_{\boldsymbol{\theta}_2}}{\longmapsto} f^{t,t+1} \overset{\mathrm{rec.}}{\longmapsto} \begin{matrix}\mathbf{x}^t \\ \mathbf{x}^{t+1}\end{matrix}, \quad (4)$$

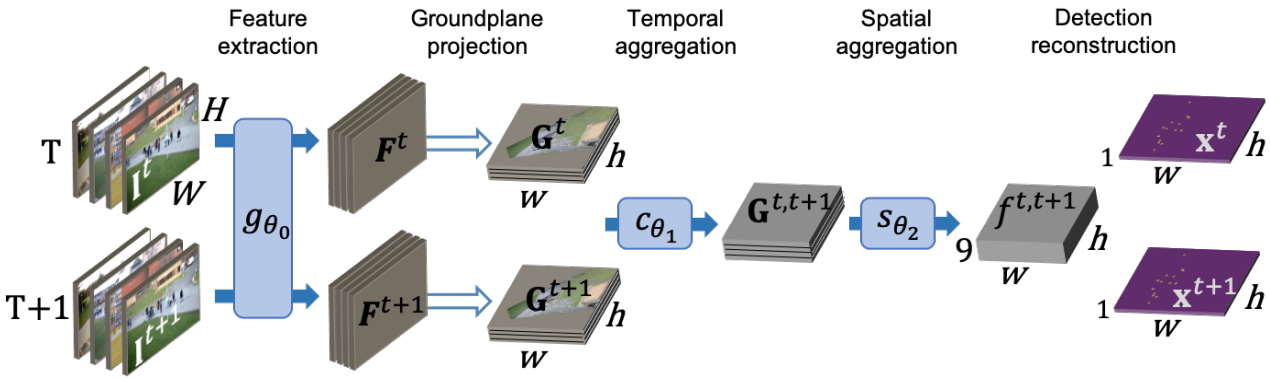

Figure 3: **Proposed Multiview Detection Architecture**. Two consecutive sets of multi-view frames are transformed into human flow $f^{t,t+1}$ by the proposed multi-view prediction model. The human flow is then used to reconstruct detection heatmaps $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$. The architecture with parameters $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is trained with ground truth detection only. Blue background boxes are trainable modules (with parameters indicated on top). Output dimensions $h = w = 128$ is used in the experiments.

Where: $g_{\boldsymbol{\theta}_0}(\mathbf{I}_v^t) \in \mathbb{R}^{W/8 \times H/8 \times D}$ is the output of the ResNet parametrized by weights $\boldsymbol{\theta}_0$. $P(\mathbf{F}_v^t, \mathbf{C}_v)$ project features onto the groundplane. Temporal aggregation is achieved with a convolution parametrized by weights $\boldsymbol{\theta}_1$ and output $c_{\boldsymbol{\theta}_1}(\mathbf{G}_v^t, \mathbf{G}_v^{t+1}) \in \mathbb{R}^{w \times h \times D'}$. The spatial aggregation layer parametrized by weights $\boldsymbol{\theta}_2$ generates the human flow $s_{\boldsymbol{\theta}_2}(\mathbf{G}^{t,t+1}) \in \mathbb{R}^{w \times h \times 9}$. The detection heatmaps $\mathbf{x} \in \mathbb{R}^{w \times h}$ are reconstructed from the flow as follows,

$$x_j^t = \sum_{k \in \mathcal{N}(j)} f_{j,k}^{t,t+1}, \quad \bar{x}_j^t = \sum_{k \in \mathcal{N}(j)} f_{k,j}^{t+1,t}. \tag{5}$$

We denote by $\bar{x}_j^t$ the reconstruction obtained by reversed flow. According to Eq. (3), for every forward flow there is an equivalent reversed flow. The inverted flow $f^{t+1,t}$ is obtained by swapping the inputs of the model.

Spatial Aggregation The use of ground plane representations greatly simplifies the spatial aggregation of multiple views. However the camera calibrations are never perfect and minor misalignments between views are common. Our aggregation mechanism is designed to be robust to such misalignment. The feature representations for all the points of view are concatenated alongside their channel dimension, then a convolutional layer with large kernel size (5×5) allows for realignment between neighboring features. An efficient spatial aggregation mechanism also needs to handle occlusion, objects hidden in some views should still be predicted from the rest of the views. To make this process easier we propose a multi-scale module: occlusions are easier to detect at a coarse level, while fine level features are needed for precise localization. The multi-scale feature works as follows: The features are pooled to 4 different sizes, and are processed by four sets of convolutions, batch normalizations and ReLUs, one for each scale. The 4 scale representations are then upscaled back to the same dimension and combined with a convolutional layer. Finally, a convolutional layer followed by a sigmoid activation function transform the aggregated feature into the human flow $f^{t-1,t}$.

3.4. Training

The goal of the training is to learn the parameters $\boldsymbol{\theta}_{0:2}$. Through a combination of loss functions, we aim at applying the constraints defined in Section 3.2. Each frame $\mathbf{I}_V^t$ is annotated with the position of every human foot. Using P we obtain the 2d coordinate of every human in the ground plane and generate binary ground truth detection maps $\mathbf{y}^t \in (0,1)^{w \times h}$. Note that the ground truth heatmaps are independent from the point of view.

Given two multiview set of frames $\mathbf{I}^t$, $\mathbf{I}^{t+1}$ and their respective ground truth detection map $\mathbf{y}^t$, $\mathbf{y}^{t+1}$ we define the loss function as follows.

$$\begin{aligned} L = L_{\text{det}}(\mathbf{x}^t, \mathbf{y}^t) + L_{\text{det}}(\mathbf{x}^{t+1}, \mathbf{y}^{t+1}) \\ + L_{\text{cycle}}(f^{t,t+1}, f^{t+1,t}), \end{aligned} \tag{6}$$

with L_{det} as the detection loss between prediction from flow and inverted flow and ground truth detection. Applied both at time t and $t+1$ together L_{det} enforces the constraints defined in Eq. (1) and Eq. (2).

$$L_{\text{det}}(\mathbf{x}^t, \mathbf{y}^t) = \left(\mathbf{x}^t - \mathbf{y}^t\right)^2 + \left(\bar{\mathbf{x}}^t - \mathbf{y}^t\right)^2. \tag{7}$$

The temporal consistency constraint defined in Eq. (3) is directly applied on the flow with the loss L_{cycle} between the

flow and its inverted counterpart.

$$L_{\text{cycle}}\left(f^{t,t+1}, f^{t+1,t}\right) = \sum_{j \in G^t} \sum_{k \in \mathcal{N}(j)} \left(f^{t,t+1}_{j,k} - f^{t+1,t}_{k,j}\right)^2, \tag{8}$$

where $j \in G$ correspond to every cell in ground plane grid.

Dealing with unbalanced flow When dealing with video sequences with large frame rates, people barely move between consecutive frames. It results in a large imbalance in terms of flow, with static flow being occasionally 20 times more common than any other flow direction. To help with the issue we introduce a motion-based reweighting of our detection loss which reads as follows:

$$L_{\text{det}}\left(\mathbf{x}^t, \mathbf{y}^t\right) = \left(\left(\mathbf{x}^t - \mathbf{y}^t\right)^2 + \left(\bar{\mathbf{x}}^t - \mathbf{y}^t\right)^2\right) \times (1 + \lambda_r |\mathbf{y}^t - \mathbf{y}^{t+1}|), \tag{9}$$

where $|y^t - y^{t+1}|$ correspond to the ground truth detection map containing only the people moving between time t and $t+1$. λ_r controls the strength of the regularization. Larger values will penalize strongly incorrect prediction of people in motion. With $\lambda_r = 0$ no regularization is applied and we get the original detection loss defined in Eq. (7).

3.5. Track reconstruction

We have introduced a detection model that is able to produce detection heatmaps as well as human flow. In this part we extend two existing association algorithm to leverage human flow while generating tracks.

In KSP[9] they reformulate the association problem as a constrained flow optimization problem. Starting from detection probability maps, they build a dense graph containing all possible location across time. Each location is connected to its neighbors in the previous and next time step. The weight on the edges is proportional to the detection probability at this location. All the locations in the first time step are connected to a source node, and all the locations in the last time step are connected to a sink node.

The optimization on this graph is maximizing the number of paths connecting the source to the target while minimizing the overall cost. Given the graph previously defined this can be done using Suurballe's algorithm [54].

One limitation of KSP, is that when building the graph it assumes an equal probability for all the different directions (neighbors). In crowded scene such an assumption can easily result in identity switches. We propose to reformulate KSP to have separate probabilities for each direction. We replace Eq. 12 from the original paper [9], in order to integrate the predicted flow from our model. The edge cost for the graph is derived from our predicted flow as follows:

$$c_{KSPFlow}\left(e^t_{i,j}\right) = -\log\left(\frac{f^{t,t+1}_{i,j}}{1 - f^{t,t+1}_{i,j}}\right). \tag{10}$$

Everything else is kept as in the original KSP paper. We will denote this new method KSPFlow for the rest of the paper.

MuSSP[58] propose to solve the association step using a min-cost flow method. As opposed to KSP, a sparse graph is built from pre-extracted detection instead of the full-probability map. It improves computation efficiency and allow to model longer range of motion. However the edge costs of the graph only take into consideration, the spatial and temporal distance between detection. We propose to update the cost formulation to use the flow predicted by our model as follows:

$$c_{muSSPFlow}\left(e^t_{i,j}\right) = -e^{-(\delta_t - 1) * \sigma_t} * e^{-d(\mathbf{x_i}, \mathbf{x_j}) * \sigma_d} * e^{-d(\mathbf{x_i} + \delta_t * f^{t,t+1}_{i,j}, \mathbf{x_j}) * \sigma_f}, \tag{11}$$

where δ_t is the temporal distance between detection i and detection j and $d(\mathbf{x_i}, \mathbf{x_j})$ is the physical distance between them. We propose to add the term $d(\mathbf{x_i} + \delta_t * f^{t,t+1}_{i,j}, \mathbf{x_j})$ which model the distance between detection j and the estimated position of detection i in the future based on the flow prediction. σ_t, σ_d and σ_f are the hyperparameters that respectively control the contribution of the temporal, spatial and motion-based distances between the detections.

4. Experiments

To validate our approach, we compare it to state-of-the-art ones on the multi-view multi-object detection and tracking using PETS2009 and WILDTRACK datasets.

4.1. Datasets, Metrics, and Training

We endeavored to use the most recent and up-to-date baseline and dataset to evaluate our multi-view approach. However, we were limited by the availability of the kind of multi-view datasets we need.

	WILDTRACK dataset		
model	MODA	Prec.	Rec.
DeepOcclusion [12]	74.1	95.0	80.0
MVDet [25]	88.2	94.7	93.6
SHOT [53]	90.2	96.1	94.0
MVDeTr [24]	91.5	**97.4**	94.0
MVFlow (Ours)	**91.9**	96.4	**95.7**

Table 1: **Multi-view multi-person detection** Detection performance of our proposed MVFlow model on the WILDTRACK dataset. MODA, precision and recall are reported.

Figure 4: **Visualization of the predicted flow, viewed best zoomed in.**. For each detected person in the image, we visualize the predicted flow. Centered around each detection we reproject a 3×3 grid corresponding to the ground plane division defined in Section 3.2. The green triangle mark the cell of the flow direction with the highest probability. Or, in other words, the predicted position for the next time step. If the prediction is incorrect, a pink dot marks the true destination. Note that ground truth flow is used for visualization purpose only and is never used during training. The two left images are different points of view from the PETS2009 dataset, the two right images are coming from the WILDTRACK dataset.

Datasets. To train our model we used two datasets of calibrated multiview video sequences. The WILDTRACK dataset [12] consists of 400 annotated frames with 7 points of view covering a crowded square. As in [12], we use the first 360 frames for training and validation and the last 40 for testing. From the PETS2009 dataset [17], we use the sequence S2L1, which consists of 795 frames from 7 points of view. It has a low density and consists of 10 people walking on a road in different directions. We use the first half of the frames for training and test on the second half. This is a departure from the standard evaluation procedure that uses S2L1 only for evaluation. We did this because our algorithm needs to be trained on the same set of views it is tested. Note that our approach is trained from scratch using only the dataset mentioned above while other works commonly rely on pretrained object detectors.

Metrics. We rely on the standard CLEAR MOT metric [28]. We report Multiple Object Detection Accuracy (MODA), Multiple Object Tracking Accuracy (MOTA), and Multiple Object Tracking Precision (MOTP). We also include identity preservation metric IDF1 [49], computed similarly to the F1 score using the Identity Precision (IDP) and Identity Recall (IDR). To compute these metrics we use the publicly available *py-motmetrics* implementation.

Baselines. For detection purposes, we compare to DeepOcclusion[12], MVDet[25], SHOT[53] and MVDeTr[24] which are four approaches that tackle multi view detection on the WILDTRACK dataset. As in these papers, the detection are evaluated directly in the groundplane.

For linking purposes, we use DeepSort [59], muSSP [58] and KSP [9] for which code is publicly available. Since we are only interested in linking pointwise detections, we ignore appearance models and bounding boxes during tracking. For computational reasons we generate the tracks in two steps, first we generate tracklets up to a length of 40 detections. Then we merge the tracklets to form the final tracks. We provide additional tracking baseline using DeepSort, muSSP and KSP on the detection results of MVDet and MVDeTr.

Model Training. Our model trained using the Adam optimizer [32] with a batch size of one and a learning rate of 0.001. The learning rate is halved after epoch 20, 40, 60, 80 and 100. Random rectangular crops are taken from the input image and resized to a fixed-size of 536×960. The ResNet 34 [21] reduces this dimensionality by a factor 8. The ground plane projection P produce the final spatial dimension of the flow and detection output of $w = 128, h = 128$.

4.2. Comparing to the State-of-the-Art

Multiview Detection. Our model builds detection heatmaps from human flow. To produce actual detections from these heatmaps, we use Non Maximum Suppression (NMS). We then select the top-200 and use K-mean clustering to filter-out spurious ones.

We report our results on the WILDTRACK dataset in Table 1. Our model consistently outperforms the baselines. In particular it improves MODA by 0.4 over MVDeTr [24], the previously best performing model that uses a complex transformer architecture and additional supervision in the image plane.

Multiview Tracking. We report our tracking results on both WILDTRACK and PETS 2009 in Table 2. We report result for 5 associations algorithm, 3 of which are baselines: DeepSort is online, while KSP and muSSP are offline and graph based. KSPFlow and muSSPFlow introduced in Section 3.5 are implemented by replacing the edge cost function of KSP and muSSP with Eq. (10) and Eq. (11).

Combining MVFlow with muSSPFlow delivers the best MOTA. Note that both KSPFlow and muSSPFlow respec-

	PETS S2L1 dataset						
model	MOTA	MOTP	IDF1	IDP	IDR	ML	MT
B&P [35]	76	-	-	-	-	-	-
POM + KSP [9]	78	-	-	-	-	-	-
HTC [60]	89	-	-	-	-	-	-
MVFlow + deepSORT (Ours)	71.8	1.05	72.0	64.3	81.7	0	8
MVFlow + KSP (Ours)	88.8	0.67	83.3	82.6	84.0	0	7
MVFlow + KSPFlow (Ours)	91.4	0.67	**88.3**	**90.7**	**85.9**	0	7
MVFlow + muSSP (Ours)	92.2	0.65	78.5	79.6	77.4	0	8
MVFlow + muSSPFlow (Ours)	**93.3**	**0.64**	84.0	85.1	82.8	0	8

	WILDTRACK dataset						
model	MOTA	MOTP	IDF1	IDP	IDR	ML	MT
DeepOcclusion+KSP [12]	69.6	-	73.2	83.8	65.0	-	-
DeepOcclusion+KSP+ptrack [12]	72.2	-	78.4	84.4	73.1	-	-
MVDet [25] + deepSORT	8.3	0.95	48.8	46.9	50.8	20	12
MVDet [25] + KSP	46.9	0.81	64.8	95.5	49.0	28	13
MVDet [25] + muSSP	80.6	0.80	79.4	79.2	79.6	4	29
MVDeTr [24] + deepSORT	24.0	0.94	56.3	56.4	56.2	20	14
MVDeTr [24] + KSP	48.5	0.64	65.8	97.0	49.8	28	13
MVDeTr [24] + muSSP	89.4	0.58	90.7	90.5	90.9	3	33
MVFlow + deepSORT (Ours)	52.8	0.89	73.3	65.2	83.7	**2**	33
MVFlow + KSP (Ours)	81.9	0.61	79.8	80.2	79.5	4	32
MVFlow + KSPFlow (Ours)	83.5	0.61	81.0	81.7	80.3	5	29
MVFlow + muSSP (Ours)	**91.3**	**0.57**	**93.5**	**92.7**	**94.2**	**2**	**38**
MVFlow + muSSPFlow (Ours)	91.2	**0.57**	93.4	92.6	**94.2**	**2**	**38**

Table 2: **Multi-view multi-person tracking** Tracking performance of our proposed MVFlow model on two datasets: PETS2009 S2L1 and WILDTRACK. For each dataset we show the result of our detection model combined with five different association algorithms. Our model MVFlow + muSSPFlow achieves state-of-the-art results both on PETS2009 and WILDTRACK. We report the CLEAR MOT metrics MOTA, MOTP, Mostly Loss (ML), Mostly Track (MT), as well as identity conservation metrics IDF1, IDP and IDR.

tively match or outperform KSP and muSSP both in terms of MOTA and IDF1 on both datasets. This gap in performance clearly confirm the benefit of predicting human motion and using it for tracking.

4.3. Ablation Study

To evaluate the contribution of the different component of the proposed approach we conduct and ablation study. We use the WILDTRACK dataset and combine the ablated model with muSSP to generate tracking results. We report the results in Table 3 and the performance numbers should be compared to the WILDTRACK numbers in the penultimate row of Table 2.

No Flow Prediction. To ignore the flow, we modify the last convolutional layer of our model such that it directly produces detection heatmaps. In other words, we remove the flow prediction and detection reconstruction steps, everything else is kept as is. As can be seen by comparing the fourth row of Table 3 to the penultimate row of Table 2, the performance decreases substantially, which confirms the importance of flow prediction. This experiment demonstrates the regularization property of our flow formulation. Predicting the flow under realistic motion constraints improve the reconstructed detection on its own, even without explicitly using the flow.

No Temporal Consistency Temporal consistency as defined in Eq. (3) enforces similarity between flow and its in-

model	MOTA	MOTP	IDF1	IDP	IDR	ML	MT	MODA
$\lambda_r = 0$	58.5	0.62	62.0	73.8	53.4	18	13	61.1
No aug.	76.8	0.65	73.0	76.0	70.2	4	29	80.6
Single scale	85.3	0.59	85.5	87.1	84.0	5	33	87.9
No flow	87.4	0.61	90.6	88.4	92.8	3	40	88.6
No L_{cycle}	87.6	0.61	86.7	85.7	87.8	5	38	89.3
$\lambda_r = 10$	87.8	0.59	86.4	86.9	85.9	5	35	89.7

Table 3: **Ablation result on multi-view multi-person tracking** Using the WILDTRACK dataset, we test how each component of our model contributes to the overall performance. We remove: the flow balancing term in the loss ($\lambda_r = 0$), the data augmentation, the multi-scale view aggregation and replace it with a single scale one. We replace the flow by directly regressing detection, we remove the temporal consistency term of our loss. All the results are computed using the muSSP association algorithm (comparable to the penultimate row of Table 2)

verted counterpart. It forces the network to model temporal information about the order of frames and to not purely rely on visual cues. As can be seen in Table 3, removing it during training reduce both MOTA and identity preservation measured by IDF1.

Flow Reweighting Across the different datasets, we observed a dominance of static flow over others. To avoid overfitting to the most common flow direction, we introduced in Eq. (9) a motion-based reweighting scheme in our loss function. We now test different values for λ_r that controls how much re-weighting there is. Our best model was obtained with $\lambda_r = 5$. In the first and last rows of Table 3, we report results with $\lambda_r = 0$—no re-reweigthing—and $\lambda_r = 10$. The performance is degraded in both cases, especially when $\lambda_r = 0$. Note that the training remains robust over a wide range of λ_r and value between one and six all gave decent results.

Single Scale Spatial Aggregation. Robust view aggregation requires the ability to handle misalignment between views. To do so we proposed a multiscale aggregation mechanism. To test its efficiency we only retain the highest resolution branch in our aggregation module. Everything else is kept the same. As shown in the third row of Table 3, this also degrades performance, thus confirming the importance multiscale aggregation. Doing the aggregation at multiple scales helps to deal with slight misalignments between views, the model learns which scale contains the most reliable information in order to obtain the optimal combination of the different views.

4.4. Limitations

Human Motion Assumption The proposed approach makes assumptions about human motion, which are stated in Section 3.2. In particular we assume that people are not sharing the same space and are moving at most one cell between two frames. Both of this assumptions are conditioned on the discretization of the space. This is not a problem in the ground plane since people cannot share the same physical space and the ground plane grid can be sized to respect the motion constraints. On the other hand, it can be problematic to respect both constraints in the image plane. Therefore our method is best suited for calibrated setup: full-camera calibration or at minimum a homography mapping the image plane to the ground plane. This is not a problem in a multi-view setting, but applying our method on a single view dataset would require an extra calibration step.

Weakly Supervised Convergence Predicting human motion while only being supervised with detection annotation is one of the merits of our approach. However it also has its downside, from Eq. (7) and Eq. (8) the flow is guaranteed to be accurate for small values of the loss L. To reach high flow accuracy the training needs to have fully converged, this can be problematic on smaller datasets. Once the training reaches this converged state, it already starts to overfit the training data. On smaller datasets we observed this trade off between flow accuracy and generalization.

5. Conclusion

In this paper, we propose a weakly supervised approach to detect people flow given only detection supervision. Our flow-based framework directly leverages temporal consistency across video frames and explicitly enforces scale and motion consistency over multiple viewpoints. Our experiments show that our model consistently outperforms state-of-the-art multi-view people tracking approaches. In the future, we plan extend our work to applications that track people in extremely crowded scene.

Acknowledgments This work was funded in part by the Swiss Innovation Agency.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

[2] M. Andriluka, S. Roth, and B. Schiele. People-Tracking-By-Detection and People-Detection-By-Tracking. In *Conference on Computer Vision and Pattern Recognition*, June 2008.

[3] A. Andriyenko and K. Schindler. Globally Optimal Multi-Target Tracking on a Hexagonal Lattice. In *European Conference on Computer Vision*, pages 466–479, September 2010.

[4] P. Baqué, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-Target Detection. In *International Conference on Computer Vision*, 2017.

[5] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[6] B. Benfold and I. Reid. Stable Multi-Target Tracking in Real-Time Surveillance Video. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[7] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking Multiple People Under Global Apperance Constraints. In *International Conference on Computer Vision*, 2011.

[8] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1614–1627, 2014.

[9] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):1806–1819, 2011.

[10] A. Butt and R. Collins. Multiple Target Tracking Using Frame Triplets. In *Asian Conference on Computer Vision*, 2012.

[11] A. Butt and R. Collins. Multi-Target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In *Conference on Computer Vision and Pattern Recognition*, pages 1846–1853, 2013.

[12] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret. The Wildtrack Multi-Camera Person Dataset. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[13] W. Choi and S. Savarese. A Unified Framework for Multi-Target Tracking and Collective Activity Recognition. In *European Conference on Computer Vision*, 2012.

[14] R.T. Collins. Multitarget Data Association with Higher-Order Motion Models. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[15] A. Dehghan, Y. Tian, P. Torr, and M. Shah. Target Identity-Aware Network Flow for Online Multiple Target Tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 1146–1154, 2015.

[16] C. Dicle, O. I. Camps, and M. Sznaier. The Way They Move: Tracking Multiple Targets with Similar Appearance. In *International Conference on Computer Vision*, 2013.

[17] J. Ferryman and A. Shahrokni. Pets2009: Dataset and Challenge. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillanc*, 2009.

[18] 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*, 30(2):267–282, February 2008.

[19] Yiyang Gan, Ruize Han, Liqiang Yin, Wei Feng, and Song Wang. Self-supervised multi-view multi-human association and tracking. In *ACM International Conference on Multimedia*, 2021.

[20] A. Gijsberts, M. Atzori, C. Castellini, H. Muller, and B. Caputo. Movement Error Rate for Evaluation of Machine Learning Methods for Semg-Based Hand Movement Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22:735–744, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[22] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang. Connected Component Model for Multi-Object Tracking. *IEEE Transactions on Image Processing*, 25(8), 2016.

[23] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[24] Y. Hou and L. Zheng. Multiview Detection with Shadow Transformer (And View-Coherent Data Augmentation). In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1673–1682, 2021.

[25] Y. Hou, L. Zheng, and S. Gould. Multiview Detection with Feature Perspective Transformation. In *European Conference on Computer Vision*, pages 1–18, 2020.

[26] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[27] H. Jiang, S. Fels, and J.J. Little. A Linear Programming Approach for Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009.

[29] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion Segmentation & Multiple Object Tracking by Correlation Co-Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[30] C. Kim, F. Li, M. Alotaibi, and J.M. Rehg. Discriminative Appearance Modeling with Multi-Track Pooling for Real-Time Multi-Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[31] C. Kim, F. Li, and J. M. Rehg. Multi-Object Tracking with Neural Gating Using Bilinear LSTM. In *European Conference on Computer Vision*, 2018.

[32] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2015.

[33] J. Lafferty, A. Mccallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning*, pages 282–289, 2001.

[34] I. Laptev, B. Caputo, C. Schuldt, and T. Lindeberg. Local Velocity-Adapted Motion Events for Spatio-Temporal Recognition. *Computer Vision and Image Understanding*, 108:207–229, 2017.

[35] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Branch-And-Price Global Optimization for Multi-View Multi-Target Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[36] P. Lenz, A. Geiger, and R. Urtasun. Followme: Efficient Online Min-Cost Flow Tracking with Bounded Memory and Computation. In *International Conference on Computer Vision*, pages 4364–4372, December 2015.

[37] J. Liu, P. Carr, R. T. Collins, and Y. Liu. Tracking Sports Players with Context-Conditioned Motion Models. In *Conference on Computer Vision and Pattern Recognition*, pages 1830–1837, 2013.

[38] W. Liu, M. Salzmann, and P. Fua. Estimating People Flows to Better Count Them in Crowded Scenes. In *European Conference on Computer Vision*, 2020.

[39] W. Liu, M. Salzmann, and P. Fua. Counting People by Estimating People Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[40] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In *International Conference on Multimedia and Expo*, 2018.

[41] A. Maksai and P. Fua. Eliminating Exposure Bias and Loss-Evaluation Mismatch in Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[42] A. Milan, S. Roth, and K. Schindler. Continuous Energy Minimization for Multitarget Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:58–72, 2014.

[43] A. Milan, K. Schindler, and S. Roth. Detection- And Trajectory-Level Exclusion in Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 3682–3689, 2013.

[44] A. Mittal, L. Zhao, and L.S. Davis. Human Body Pose Estimation Using Silhouette Shape Analysis. In *Conference on Advanced Video and Signal Based Surveillance*, page 263, 2003.

[45] F. Nater, T. Tommasi, H. Grabner, L. V. Gool, and B. Caputo. Transferring Activities: Updating Human Behavior Analysis. In *International Conference on Computer Vision*, 2011.

[46] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu. Quasi-Dense Similarity Learning for Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[47] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In *International Conference on Computer Vision*, 2009.

[48] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In *Conference on Computer Vision and Pattern Recognition*, pages 1201–1208, June 2011.

[49] 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*, 2016.

[50] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould. Probabilistic Tracklet Scoring and Inpainting for Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[51] A. V. Segal and I. Reid. Latent Data Association: Bayesian Model Selection for Multi-Target Tracking. In *International Conference on Computer Vision*, pages 2904–2911, 2013.

[52] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-Based Multiple-Person Tracking with Partial Occlusion Handling. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[53] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[54] J. W. Suurballe. Disjoint Paths in a Network. *Networks*, 4:125–145, 1974.

[55] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph Decomposition for Multi-Target Tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[56] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon. Learning to Track with Object Permanence. In *International Conference on Computer Vision*, 2021.

[57] C. Vogel, S. Roth, and K. Schindler. 3D Scene Flow Estimation with a Rigid Motion Prior. In *International Conference on Computer Vision*, 2011.

[58] C. Wang, Y. Wang, Y. Wang, C.T. Wu, and G. Yu. muSSP: Efficient Min-Cost Flow Algorithm for Multi-Object Tracking. In *Advances in Neural Information Processing Systems*, pages 423–432, 2019.

[59] N. Wojke, A. Bewley, and D. Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In *International Conference on Image Processing*, 2017.

[60] Y. Xu, X. Liu, Y. Liu, and S.C. Zhu. Multi-View People Tracking via Hierarchical Trajectory Composition. In *Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[61] Y. Xu, A. Osep, Y. Ban, R. Horaud, L. Leal-Taixe, and X. Alameda-Pineda. How to Train Your Deep Multi-Object Tracker. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[62] B. Yang and R. Nevatia. An Online Learned CRF Model for Multi-Target Tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 2034–2041, June 2012.

[63] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized Belief Propagation. In *Advances in Neural Information Processing Systems*, pages 689–695, 2000.

[64] K. Yoon, Y.-M. Song, and M. Jeon. Multiple Hypothesis Tracking Algorithm for Multi-Target Multi-Camera Tracking with Disjoint Views. *IET Image Processing*, 2018.

[65] L. Zhang, Y. Li, and R. Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *European Conference on Computer Vision*, 2020.