Multi-target tracking with context from interaction feature strings

Laura Leal-Taixé[1,2], Michele Fenzi[2], Alina Kuznetsova[2], Bodo Rosenhahn[2], and Silvio Savarese[3]

[1]Photogrammetry and Remote Sensing, ETH Zürich
[2]Institute for Information Processing, Leibniz University Hannover
[3]Computational Vision and Geometry Lab, Stanford University

1. Introduction

Many computer vision tasks are related to the problem of understanding the semantic content of a scene from a video sequence. Humans are often the center of attention of a scene, therefore, the ability to detect and track multiple people from a video has emerged as one of the top tasks to address in our field. A common approach to multiple people tracking follows the idea of estimating the hypotheses of the locations of people using a detector for each frame. Those hypotheses are then associated in time, so as to form consistent tracks for each individual. Recent works show that it is more reliable to jointly solve the data association problem for all tracks and all frames, either in discrete space using Linear Programming (LP) [9, 4] or in continuous space [6]. Most tracking systems work with the assumption that the motion model for each target is independent, but in reality, a pedestrian follows a series of social rules, *i.e.* is subject to social forces according to other moving targets around him/her. These have been defined in what is called the social force model (SFM), which has been recently used for multiple people tracking [4, 7]. One of the problems in tracking-by-detection methods is that they are highly dependent on detection results. Methods that use a physical model to estimate pedestrians' motion [4, 7] are completely unaware of the effect of undetected pedestrians, which reduces its effectiveness in semi-crowded environments, where it is very common to observe occlusions and it is very hard to estimate a pedestrian's trajectory.

We propose to construct a model that estimates how a pedestrian moves according to the motion and appearance features around him/her. We introduce the *interaction feature strings* which are used in a Random Forest (RF) framework [1] trained to estimate the velocity of a pedestrian at a certain frame. A clear advantage of our method is that it relaxes the dependency on detections, since the effect of a partially occluded (and potentially not detected) pedestrian can still be encoded in the interaction feature string, and not ignored as in common tracking-by-detection methods.

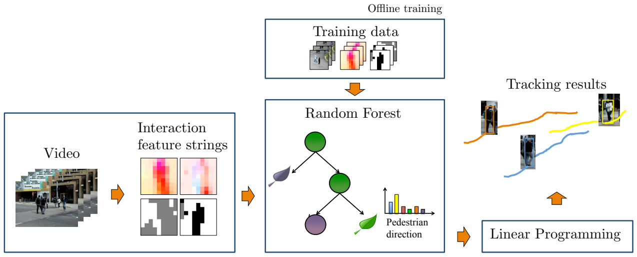
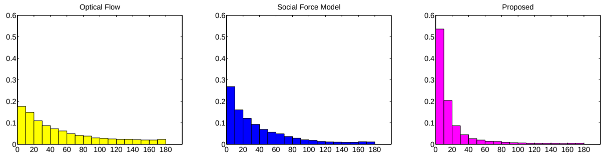


Figure 1: Diagram of the proposed approach.

2. Learning a motion model from interaction feature strings

Our method is based on what we call *interaction feature strings*, which encode image features that represent a particular scene configuration. A scene $\mathcal{I}(\mathbf{p}_i^t)$ is defined as a patch centered around a detected pedestrian i at time t and position $(x, y) \in \mathbb{R}^2$ in pixel coordinates. It has a size of $[h_i s_h, h_i s_w]$, where h_i represents the pedestrian's height, s_h and s_w are scaling factors. The patches are scaled according to the pedestrian's height to obtain a scale-invariant representation that allows us to deal with scenes both closer and further away from the camera.

We divide the scene in N_B blocks and compute a set of features per block. For different types of interactions between pedestrians, different blocks will contribute to describing the scene. For example, in a scene where a pedestrian walks alone, the central blocks will contain most of the relevant information. If there are more pedestrians involved in a scene, the outer blocks will become more and more relevant. For each of the blocks, we compute several descriptive features $\mathbf{F}_b(\mathbf{p}_i^t) = \left(F_b^1(\mathbf{p}_i^t), F_b^2(\mathbf{p}_i^t) \dots F_b^{N_f}(\mathbf{p}_i^t)\right) \in \mathbb{R}^{N_f}$, where N_f is the total number of feature channels per block and b is the block index. We concatenate the block features into one interaction feature string $\mathcal{F}(\mathbf{p}_i^t) = \{\mathbf{F}_b(\mathbf{p}_i^t)\}_{b=1}^{N_B}$. The features we use include: (i) Mean Optical Flow (MOF) of N_{FR} frames, (ii) Difference of Op-



(a) (b) (c)

Figure 2: Histogram of velocity estimation errors given by: (a) Optical Flow, (b) Social Force Model, (c) Proposed approach. Mean results on the 4 datasets.

tical Flows (DOF) between each pair of frames from t to $t + N_{FR}$, (iii) Histogram of Optical Flows (HOF), (iv) Ternary Optical Flow (TOF) which compares the norm of the optical flow of frame t and frame $t + N_{FR}$ and encodes it with values $\{0, 1, -1\}$ and (v) Ternary Angular Optical Flow (TOAF) which does the same comparison for the angle of optical flows. Note that we do not compensate for camera motion, and therefore it is included in our feature strings and will also be taken into account in the pedestrian velocity estimation. This is an important property of our feature strings, since we are tracking in image coordinates. Once we have a descriptive set of features for a scene, the goal is to train a Hough Random Forest [1] to be able to estimate the velocity of a pedestrian. This information is then used in a probabilistic tracking framework, such as Linear Programming [4, 9], to obtain the final set of trajectories. We refer the reader to [3] for more details.

3. Experimental results

In order to evaluate the multiple people tracking performance of the proposed algorithm, we use four publicly available datasets: BAHNHOF, SUNNYDAY, JELMOLI, and LINTHESCHER from [2]. The datasets are taken from a mobile camera moving around in crowded scenarios.

This first experiment aims at showing how well our approach estimates the velocity of a pedestrian. The setup of this experiment is the following: we use one sequence for testing and the other three for training. We compare the performance of our method with two baselines: (i) the Social Force Model (SFM) [4, 7] and (ii) the Optical Flow (OF). In Figure 2 we plot the quantized relative frequency of the velocity estimation errors in degrees. As we can see, our method makes more than 50% of the estimations with less than 10 degrees of error, compared to 20% of OF and 30% of SFM.

Method	Rec.	Prec.	MT	PT	ML	Frg	Ids
Zhang *et al.* [9]	74.6	77.8	55.6	38.1	6.2	178	138
Leal-Taixé *et al.* [4]	74.1	75.3	55.1	36.9	7.9	184	131
Pellegrini *et al.* [7]	72.3	84.1	51.6	42.7	5.6	206	77
Milan *et al.* [6]	77.3	87.2	66.4	25.4	8.2	69	57
Yang & Nevatia [8]	79.0	**90.4**	68.0	24.8	7.2	**19**	**11**
LP + 2D	80.7	83.6	64.1	29.6	6.2	91	70
LP + OF	76.1	80.2	55.9	33.5	10.5	104	75
Proposed	**83.8**	79.7	**72.0**	**23.3**	**4.7**	85	71

Table 1: Results on BAHNHOF and SUNNYDAY datasets. MT = mostly tracked. PT = partially tracked. ML = mostly lost. Frg = fragmented tracks. Ids = identity switches.

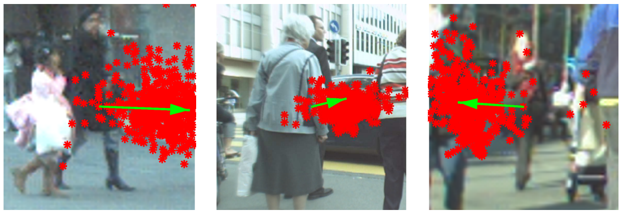


Figure 3: The green arrow indicates the ground truth velocity of the pedestrian, and in red we plot the votes made by the corresponding leafs of the learned RF.

In the second experiment, we use the learned model for multiple people tracking. We compare the results with 5 state-of-the-art tracking algorithms and two baselines using the same LP formulation as presented in this paper: (i) LP + 2D: only uses pixel distance to solve the data association problem, (ii) LP + OF: uses pedestrian velocity estimation coming only from Optical Flow. We use the same detections and the metrics described in [5]. We show the comparative results averaged for both datasets in Table 1. As we can see, our method obtains the highest recall rate, outperforming state-of-the-art by almost 5%. Even if Optical Flow features contain a lot of information on the pedestrian velocity, their naive use leads to a poor performance as shown by the results obtained with LP+OF. This shows that the proposed method is able to take the most out of a feature channel (OF) that on its own is not able to provide good velocity estimations.

References

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 1, 2

[2] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. *CVPR*, pages 1–8, 2008. 2

[3] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. *CVPR*, 2014. 2

[4] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. *ICCVW. 1st Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, 2011. 1, 2

[5] Y. Li, C. Huang, and R. Nevatia. Learning to associate: hybrid boosted multi-target tracker for crowded scene. *CVPR*, 2009. 2

[6] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. *CVPR*, 2013. 1, 2

[7] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: modeling social behavior for multi-target tracking. *ICCV*, 2009. 1, 2

[8] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. *CVPR*, 2012. 2

[9] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. *CVPR*, 2008. 1, 2