Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker

Laura Leal-Taixé, Gerard Pons-Moll and Bodo Rosenhahn
Institute for Information Processing (TNT)
Leibniz University Hannover, Germany
leal@tnt.uni-hannover.de

Abstract

Multiple people tracking consists in detecting the subjects at each frame and matching these detections to obtain full trajectories. In semi-crowded environments, pedestrians often occlude each other, making tracking a challenging task. Most tracking methods make the assumption that each pedestrian's motion is independent, thereby ignoring the complex and important interaction between subjects.

In this paper, we present an approach which includes the interaction between pedestrians in two ways: first, considering social and grouping behavior, and second, using a global optimization scheme to solve the data association problem. Results on three challenging publicly available datasets show our method outperforms state-of-the-art tracking systems.

1. Introduction

Multiple people tracking is a key problem for many computer vision tasks, such as surveillance, animation or activity recognition. In crowded environments occlusions and false detections are common, and although there have been substantial advances in the last years, tracking is still a challenging task. Tracking is often divided in two steps: detection and data association. Researchers have presented improvements on the object detector [5, 26] as well as on the optimization techniques [14, 16] and even specific algorithms have been developed for tracking in crowded scenes [1]. Though each object can be tracked separately, recent works have proven that tracking objects jointly and taking into consideration their interaction can give much better results in complex scenes. Current research is mainly focused on two aspects to exploit the interaction between pedestrians: the use of a global optimization strategy [4, 28] and a social motion model [22, 27]. The focus of this paper is to marry the concepts of global optimization and social and grouping behavior to obtain a robust tracker able to work in crowded scenarios.

This work was partially funded by the German Research Foundation, DFG projects RO 2497/7-1 and RO 2524/2-1.

(a) (b) (c)

Figure 1: Including social and grouping behavior to the network flow graph. (a) Constant velocity assumption. (b) Avoidance forces. (c) Group attraction forces.

1.1. Related work

The optimization strategy deals with the data association problem, which is usually solved on a frame-by-frame basis or one track at a time. Several methods can be used such as Markov Chain Monte Carlo (MCMC) [15] or inference in Bayesian networks [20]. In [3] an efficient approximative Dynamic Programming (DP) scheme is presented, in which trajectories are estimated one after the other, which obviously does not guarantee a global optimum for all trajectories. Recent works show that global optimization can be more reliable in crowded scenes as it solves the matching problem jointly for all tracks. The multiple object tracking problem is defined as a linear constrained optimization flow problem and Linear Programming (LP) is commonly used to find the global optimum. The idea was first used for people tracking in [12], although this method needs to know a priori the number of targets to track, which limits its application in real tracking situations. Other works formulate the tracking problem as a maximum flow [4] or a minimum cost problem [24, 28], both efficiently solved using LP and with a far superior performance when compared with DP [3].

Most tracking systems work with the assumption that the motion model for each target is independent. This simplify-

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00 ©2011 IEEE

ing assumption is especially problematic in crowded scenes. In order to avoid collisions and reach the chosen destination at the same time, a pedestrian follows a series of social rules or social forces. These have been defined in what is called the Social Force Model (SFM) [11], which has been used for abnormal crowd behavior detection [19], crowd simulation [21] and has only recently been applied to multiple people tracking: in [25], an energy minimization approach is used to predict the future position of each pedestrian considering all the terms of the social force model. In [22] and [17], the social forces are included in the motion model of the Kalman or Extended Kalman filter. In [10] a method is presented to detect small groups of people in a crowd, but it is only recently that grouping behavior has been included in a tracking framework [7, 23, 27]. In [23] groups are included in a graphical model which contains cycles and, therefore, Dual-Decomposition is needed to find the solution, which obviously is computationally much more expensive than using Linear Programming. Moreover, the results presented in [23] are only for short time windows. On the other hand, the formulations of [7, 27] are predictive by nature and therefore too local and unable to deal with trajectory changes (e.g. when people meet and stop to talk).

Social behavior models have only been introduced within a predictive framework, which are suboptimal due to the recursive nature of filtering. Therefore, in contrast to previous works, we propose to include social and grouping models into a global optimization framework which allows us to better estimate the true maximum a-posteriori probability of the trajectories.

1.2. Contributions

We present a novel approach for multiple people tracking which takes into account the interaction between pedestrians in two ways: first, using global optimization for data association and second, including social as well as grouping behavior. The key insight is that people plan their trajectories in advance in order to avoid collisions, therefore, a graph model which takes into account future and past frames is the perfect framework to include social and grouping behavior. We formulate multiple object tracking as a minimum-cost network flow problem, and present a new graph model which yields to better results than existing global optimization approaches. The social force model (SFM) and grouping behavior (GR) are included in an efficient way without altering the linearity of the problem. Results on several challenging public datasets show the improvement of the tracking results in crowded environments.

2. Multiple people tracking

Tracking is commonly divided in two steps: object detection and data association. First, the objects are detected in each frame of the sequence and second, the detections are matched to form complete trajectories. In this section we define the data association problem and describe how to convert it to a minimum-cost network flow problem, which can be efficiently solved using Linear Programming.

The idea is to build a graph in which the nodes represent the pedestrian detections. These nodes are fully connected to past and future observations by edges, which determine the relation between two observations with a cost. Thereby, the matching problem is equivalent to a minimum-cost network flow problem: finding the optimal set of trajectories is equivalent to sending flow through the graph so as to minimize the cost.

2.1. Problem statement

Let $\mathcal{O} = \{\mathbf{o}_i\}$ be a set of object detections with $\mathbf{o}_i = (\mathbf{p}_i, t_i)$, where $\mathbf{p}_i = (x, y, z)$ is the 3D position and t_i is the time stamp. A trajectory is defined as a list of ordered object detections $T_k = \{\mathbf{o}_{k_1}, \mathbf{o}_{k_2}, \cdots, \mathbf{o}_{k_N}\}$, and the goal of multiple object tracking is to find the set of trajectories $\mathcal{T}* = \{T_k\}$ that best explains the detections. This is equivalent to maximizing the a-posteriori probability of $\mathcal{T}$ given the set of detections $\mathcal{O}$. Assuming detections are conditionally independent, the objective function is expressed as:

$$\mathcal{T}* = \underset{\mathcal{T}}{\mathbf{argmax}} \prod_i P(\mathbf{o}_i|\mathcal{T})P(\mathcal{T}) \tag{1}$$

$P(\mathbf{o}_i|\mathcal{T})$ is the likelihood of the detection. In order to reduce the space of $\mathcal{T}$, we make the assumption that the trajectories cannot overlap (i.e., a detection cannot belong to two trajectories), but unlike [28], we do not define the motion of each subject to be independent, therefore, we deal with a much larger search space. We extend this space by including the following dependencies for each trajectory T_k:

- Constant velocity assumption: the observation $\mathbf{o}_i \in T_k$ depends on past observations $[\mathbf{o}_{i-1}, \mathbf{o}_{i-2}]$
- Grouping behavior: If T_k belongs to a group, the set of members of the group $\mathcal{T}_{k,\text{GR}}$ has an influence on T_k
- Avoidance term: T_k is affected by the set of trajectories $\mathcal{T}_{k,\text{SFM}}$ which are close to T_k at some point in time and do not belong to the same group as T_k

The first and third dependencies are grouped into the SFM term. The sets $\mathcal{T}_{k,\text{SFM}}$ and $\mathcal{T}_{k,\text{GR}}$ are disjoint, i.e., a pedestrian can have an attractive effect or a repulsive effect on another pedestrian, but not both. Therefore, we decompose $P(\mathcal{T})$ as:

$$\begin{aligned} P(\mathcal{T}) &= \prod_{T_k \in \mathcal{T}} P(T_k \cap \mathcal{T}_{k,\text{SFM}} \cap \mathcal{T}_{k,\text{GR}}) \\ &= \prod_{T_k \in \mathcal{T}} P(\mathcal{T}_{k,\text{SFM}}|T_k)P(\mathcal{T}_{k,\text{GR}}|T_k)P(T_k) \end{aligned} \tag{2}$$

where the trajectories are represented by a Markov chain:

$$P(\mathcal{T}) = \prod_{T_k \in \mathcal{T}} P_{in}(\mathbf{o}_{k_1}) \dots P(\mathbf{o}_{k_i}|\mathbf{o}_{k_{i-1}}) \quad (3)$$
$$P_{k,\text{SFM}}(\mathbf{o}_{k,\text{SFM}}|\mathbf{o}_{k_i}, \mathbf{o}_{k_{i-1}}) P_{k,\text{GR}}(\mathbf{o}_{k,\text{GR}}|\mathbf{o}_{k_i}, \mathbf{o}_{k_{i-1}})$$
$$\dots P_{out}(\mathbf{o}_{k_N})$$

where $P_{k,\text{SFM}}$ evaluates how well the social rules are kept if $\mathbf{o}_{k_i}$ is matched to $\mathbf{o}_{k_{i-1}}$, and $P_{k,\text{GR}}$ describes how well the structure of the group is kept.

2.2. Tracking with Linear Programming

We linearize the objective function by defining a set of flow flags $f_{i,j} = \{0, 1\}$ which indicate if an edge (i, j) is in the path of a trajectory or not. In a minimum cost network flow problem, the objective is to find the values of the variables that minimize the total cost of the flows over the network. Defining the costs as negative log-likelihoods, and combining Equations (1), (2) and (3), the following objective function is obtained:

$$\begin{aligned}\mathcal{T}* &= \underset{\mathcal{T}}{\mathbf{argmin}} \sum_{T_k \in \mathcal{T}} -\log P(T_k) - \log P(\mathcal{T}_{\text{SFM}}|T_k) \\ &\quad - \log P(\mathcal{T}_{\text{GR}}|T_k) + \sum_i -\log P(\mathbf{o}_i|\mathcal{T}) \\ &= \underset{\mathcal{T}}{\mathbf{argmin}} \sum_i C_{in,i} f_{in,i} + \sum_i C_{i,out} f_{i,out} \\ &\quad + \sum_{i,j} (C_{i,j} + C_{\text{SFM},i,j} + C_{\text{GR},i,j}) f_{i,j} + \sum_i C_i f_i\end{aligned}$$

subject to the following constraints:

- Edge capacities: we assume that each detection can only correspond to one trajectory, therefore, the edge capacities have an upper bound of $u_{ij} \leq 1$ and:

$$f_{in,i} + f_i \leq 1 \qquad f_{i,out} + f_i \leq 1 \quad (4)$$

- Flow conservation at the nodes:

$$f_{in,i} + f_i = \textstyle\sum_j f_{i,j} \qquad \textstyle\sum_j f_{j,i} = f_{i,out} + f_i \quad (5)$$

To map this formulation into a cost-flow network, we define $G = (N, E)$ to be a directed network with a cost $C_{i,j}$ and a capacity u_{ij} associated with every edge $(i, j) \in E$. An example of such a network is shown in Figure 2; it contains two special nodes, the source s and the sink t; all flow that goes through the graph starts at the s node and ends at the t node. Thereby, each flow represents a trajectory T_k and the path that each flow follows indicates which observations belong to each of the trajectories. Each observation $\mathbf{o}_i$ is represented with two nodes, the beginning node $b_i \in N$ and the end node $e_i \in N$ (see Figure 2). A detection edge connects b_i and e_i.

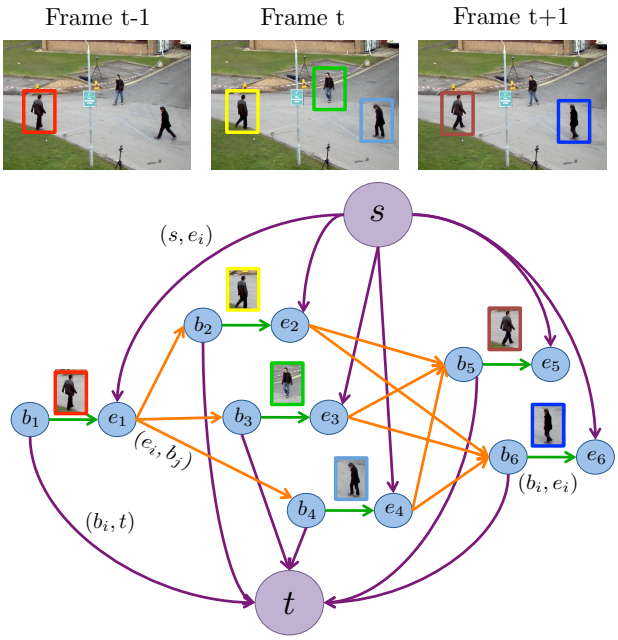


Figure 2: Example of a graph with the special source s and sink t nodes, 6 detections which are represented by two nodes each: the beginning b_i and the end e_i.

Below we detail the three types of edges present in the graphical model and the cost for each type:

Link edges. The edges (e_i, b_j) connect the end nodes e_i with the beginning nodes b_j in following frames, with cost $C_{i,j}$ and flow $f_{i,j}$, which is 1 if $\mathbf{o}_i$ and $\mathbf{o}_j$ belong to T_k and $\Delta f \leq F_{\text{max}}$, and 0 otherwise. Δf is the frame number difference between nodes j and i and F_{max} is the maximum allowed frame gap.

The costs of the link edges represent the spatial relation between different subjects. Assuming that a subject cannot move a lot from one frame to the next, we define the costs to be a decreasing function of the distance between detections is successive frames. The time gap between observations is also taken into account in order to be able to work at any frame rate, therefore velocity measures are used instead of distances. The velocities are mapped to probabilities with a Gauss error function as shown in Equation (6), assuming the pedestrians cannot exceed a maximum velocity V_{max}. The choice of parameter V_{max} is detailed in Section 4.

$$E(V_t, V_{\text{max}}) = \tfrac{1}{2} + \tfrac{1}{2}\text{erf}\left(\frac{-V_t + \frac{V_{\text{max}}}{2}}{\frac{V_{\text{max}}}{4}}\right) \quad (6)$$

The advantage of using Equation (6) over a linear function is that the probability of lower velocities decreases more slowly, while the probability for higher velocities decreases more rapidly. This is consistent with the probability distribution of speed learned from training data.

Therefore, the cost of a link edge is defined as:

$$\begin{aligned} C_{i,j} &= -\log\left(P(\mathbf{o}_j|\mathbf{o}_i)\right) + C(\Delta f) \\ &= -\log E\left(\frac{\|\mathbf{p}_j - \mathbf{p}_i)\|}{\Delta t}, V_{\max}\right) + C(\Delta f) \end{aligned}$$

where $C(\Delta f) = -\log\left(B_j^{\Delta f - 1}\right)$ is the cost depending on the frame difference between detections.

Detection edges. The edges (b_i, e_i) connect the beginning node b_i and end node e_i, with cost C_i and flow f_i, which is 1 if $\mathbf{o}_i$ belongs to T_k, and 0 otherwise.

$$C_i = \log\left(1 - P_{det}(\mathbf{o}_i)\right) + \log\left(\frac{\text{BB}_{\min}}{\|p_{\text{BB}} - \mathbf{p}_i)\|}\right)$$

If all the costs of the edges are positive, the solution to the minimum-cost problem is the trivial null flow. Consequently, we represent each observation with two nodes and a detection edge with negative cost. The higher the likelihood of a detection $P_{det}(\mathbf{o}_i)$ the more negative the cost of the detection edge, hence, confident detections are likely to be in the path of the flow in order to minimize the total cost. If a map of the scene is available, we can also include this information in the detection cost. If a detection is far away from a possible entry/exit point, we add an extra negative cost to the detection edge, in order to favor that observation to be matched. The added cost depends on the distance to the closest entry/exit point p_{BB}, and is only computed for distances higher than $\text{BB}_{\min} = 1.5m$.

Entrance and exit edges. The edges (s, e_i) connect the source s with all the end nodes e_i, with cost $C_{in,i}$ and flow $f_{in,i}$, which is 1 if T_k starts at $\mathbf{o}_i$ and 0 otherwise. Similarly, (b_i, t) connects the end node b_i with sink t, with cost $C_{i,out}$ and flow $f_{i,out}$, which is 1 if T_k ends at $\mathbf{o}_i$, and 0 otherwise.

By connecting the s node with the end nodes (or t to begin nodes), we make sure that when a track starts (or ends) it does not benefit from the negative cost of the detection edge. Therefore, we define $C_{in} = C_{out} = 0$ and the flow constraints of Eqs. (4) and (5). In [28], the authors propose to create the opposite edges (s, b_i) and (e_i, t). The advantage of our formulation is that it does not depend on P_{in} and P_{out}, which are data dependent terms that need to be calculated during optimization.

3. Modeling social behavior

If a pedestrian does not encounter any obstacles, the natural path to follow is a straight line. But what happens when the space gets more and more crowded and the pedestrian can no longer follow the straight path? Social interaction between pedestrians is especially important when the environment is crowded. In this section we consider how to include the social behavior, which we divide into the Social Force Model (SFM) and the Group behavior (GR), into our minimum-cost network flow problem.

3.1. Social Force Model

The social force model states that the motion of a pedestrian can be described as if they were subject to "social forces". There are three main terms that need to be considered: the desire of a pedestrian to maintain a certain speed, the desire to keep a comfortable distance from other pedestrians and the desire to reach a destination. Since we cannot know a priori the destination of the pedestrian in a real tracking system, we focus on the first two terms.

Constant velocity assumption. The pedestrian tries to keep a certain speed and direction, therefore we assume that in $t + \Delta t$ we have the same speed as in t and predict the pedestrian's position in $t + \Delta t$ accordingly.

Avoidance term. The pedestrian also tries to avoid collisions and keep a comfortable distance from other pedestrians. We model this term as a repulsion field with an exponential distance-decay function with value α learned from training data.

$$\mathbf{a}_i^{t+\Delta t} = \sum_{g_m \neq g_i} \exp\left(-\frac{\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \tilde{\mathbf{p}}_m^{t+\Delta t}\|}{\alpha \Delta t}\right)$$

The only pedestrians that have this repulsion effect on subject i are the ones which do not belong to the same group as i and $\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \tilde{\mathbf{p}}_m^{t+\Delta t}\| \leq 1m$. The different avoidance terms are combined linearly.

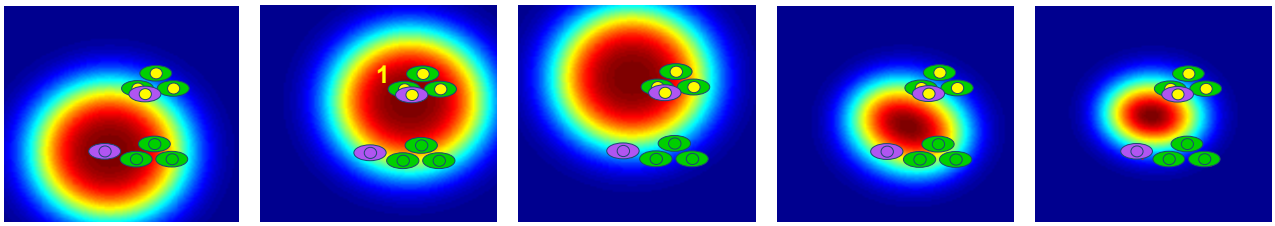
Now the prediction of the pedestrian's next position is also influenced by the avoidance term from all pedestrians:

$$\tilde{\mathbf{p}}_i^{t+\Delta t} = \mathbf{p}_i^t + (\mathbf{v}_i^t + \mathbf{a}_i^{t+\Delta t} \Delta t)\Delta t \tag{7}$$

The distance between prediction and real measurements is used to compute the cost:

$$C_{\text{SFM},i,j} = -\log E\left(\frac{\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \mathbf{p}_j^{t+\Delta t}\|}{\Delta t}, V_{\max}\right)$$

In Figure 3 we plot the probability distribution computed using different terms. Note, this is just for visualization purposes, since we do not compute the probability for each point on the scene, but only for the positions where the detector has fired. There are 4 pedestrians in the scene, the purple one and 3 green ones walking in a group. As shown in 3(b), if we only use the predicted positions (yellow heads) given the previous speeds, there is a collision between the purple pedestrian and the green marked with a 1 collide. The avoidance term shifts the probability mode to a more plausible position.



(a) (b) (c) (d) (e)

Figure 3: Three green pedestrians walk in a group, the predicted positions in the next frame are marked by yellow heads. The purple pedestrian's linearly predicted position (yellow head) clearly interferes with the trajectory of the group. Representation of the probability (blue is 0 red is 1) distribution for the purple's next position using: 3(a) only distances, 3(b) only SFM (constant velocity assumption and avoidance term), 3(c) only GR (considering the purple pedestrian belongs to the group), 3(d) distances+SFM and 3(e) distances+SFM+GR.

3.2. Group Model

The social behavior [11] also includes an attraction force which occurs when a pedestrian is attracted to a friend, shop, etc. We model the attraction between members of a group. Before modeling group behavior we determine which tracks form each group and at which frame the group begins and ends (to deal with splitting and formation of groups). The idea is that if two pedestrians are close to each other over a reasonable period of time, they are likely to belong to the same group. From the training sequence in [22], we learn the distance and speed probability distributions of the members of a group P_g vs. individual pedestrians P_i. If m and n are two trajectories which appear on the scene at $t = [0, N]$, we compute the flag $G_{m,n}$ that indicates if m and n belong to the same group.

$$G_{m,n} = \begin{cases} 1, & \sum_{t=0}^{N} P_g(m,n) > \sum_{t=0}^{N} P_i(m,n) \\ 0, & \text{otherwise} \end{cases}$$

For every observation $\mathbf{o}_i$, we define a group label g_i which indicates to which group the observation belongs to, if any. If several pedestrians form a group, they tend to keep a similar speed, therefore, if i belongs to a group, we can use the mean speed of all the other members of the group to predict the next position for i using Equation (7). The distance between this predicted position and the real measurements is used in (6) to obtain the probability for the grouping term.

An example is shown in Figure 3(c), where we can see that the maximum probability provided by the group term keeps the group configuration. In Figure 3(d) we show the combined probability of the distance and SFM information, which narrows the space of probable positions. Finally, 3(e) represents the combined probability of DIST, SFM and GR.

4. Implementation details

To compute the SFM and grouping costs, we need to have information about the velocities of the pedestrians, which can only be obtained if we already have the trajectories. We solve this chicken-and-egg problem iteratively; on the first iteration, the trajectories are estimated only with the information defined in Section 2.2. The minimum cost solution is found using the Simplex algorithm [8], with the implementation given in [18]. To reduce the computational cost, we prune the graph using the physical constraints represented by the edge costs. If any of the costs C_{ij}, $C_{\text{SFM},i,j}$ or $C_{\text{GR},i,j}$ is infinite the edge (i, j) is erased from the graphical model. For long sequences, we divide the video into several batches and optimize for each batch. For temporal consistency, the batches have an overlap of $F_{\text{max}} = 10$ frames. With our non-optimized code, the runtime for a sequence of 800 frames (114 seconds), 4837 detections, batches of 100 frames and 6 iterations is 30 seconds on a 3GHz machine. All parameters defined in the previous sections are learned from training data; in our case we use one sequence of the publicly available dataset [22]. The parameter to penalize for the frame difference is $B_j = 0.3$, the avoidance term $\alpha = 0.5$. Our approach works well for a wide range of V_{max} and F_{max}. Values between 5 and 25 were tested for both parameters, and the difference between worst and best tracking accuracy obtained was 1%. For all experiments shown in the following sections, we use $V_{\text{max}} = 7$ and $F_{\text{max}} = 10$.

5. Experimental results

In this section we show the tracking results of our method on three publicly available datasets and compare with existing state-of-the-art tracking approaches using the CLEAR metrics [13], DA (detection accuracy), TA (tracking accuracy), DP (detection precision) and TP (tracking precision).

5.1. Evaluation with missing data, noise and outliers

We evaluate the impact of every component of the proposed approach with one of the sequences of the dataset [22], which contains images from a crowded public place,

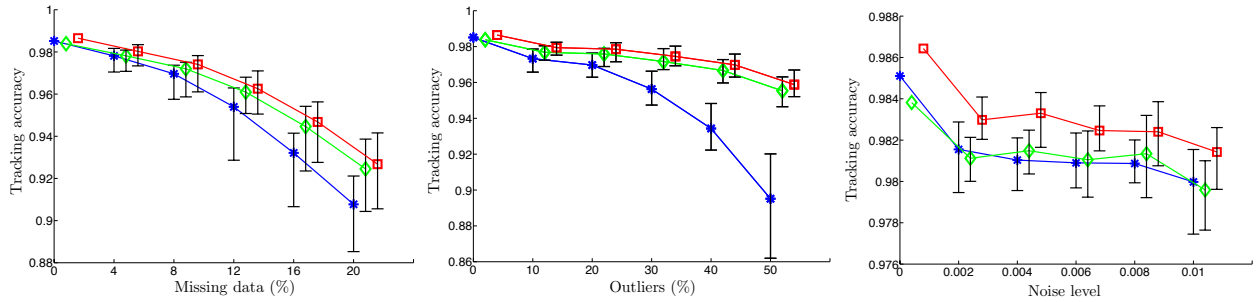
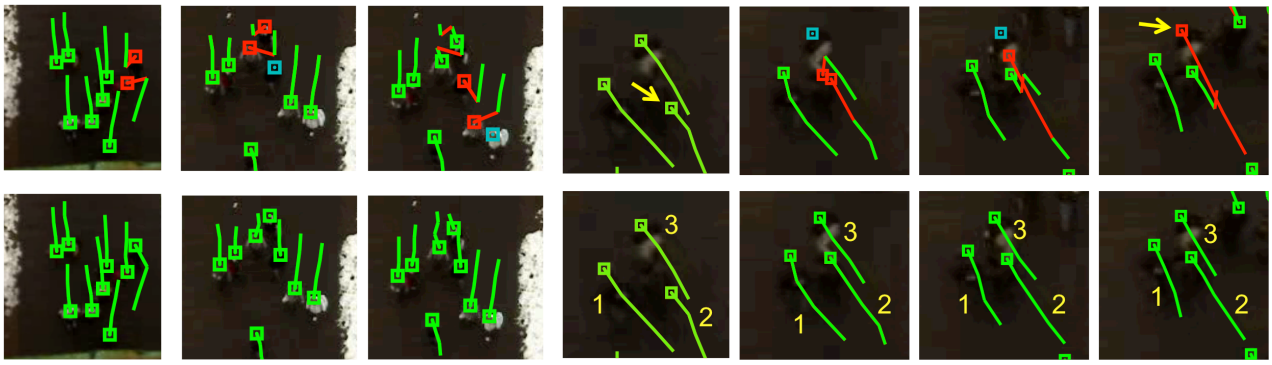


Figure 4: Experiments are repeated 50 times and average result, maximum and minimum are plotted. *Blue star* = results with DIST, *Green diamond* = results with SFM, *Red square* = results with SFM+GR. *From left to right*: Experiment with simulated missing data, with outliers, and with random noise.



(a) (b) (c)

Figure 5: *Top row*: Tracking results with only DIST. *Bottom row*: Tracking results with SFM+GR. *Green* = correct trajectories, *Blue* = observation missing from the set, *Red* = wrong match. 5(a) Wrong match with DIST, corrected with SFM. 5(b) Missing detections cause the matches to shift due the global optimization; correct result with SFM. 5(c) Missed detection for subject 3 on two consecutive frames. With SFM, subject 2 in the first frame (yellow arrow) is matched to subject 3 in the last frame (yellow arrow), creating an identity switch; correct result with grouping information.

with several groups as well as walking and standing pedestrians.

Using the ground truth (GT) pedestrian positions as the baseline for our experiments, we perform three types of tests, missing data, outliers and noise, and compare the results obtained with:

- DIST: proposed network model with velocities
- SFM: adding the Social Force Model (Section 3.1)
- SFM+GR: adding SFM and grouping behavior (Section 3.2)

Missing data. This experiment shows the robustness of our approach given missed detections. This is evaluated by randomly erasing a certain percentage of detections from the GT set. The percentages evaluated are $[0, 4, 8, 12, 16, 20]$ from the total number of detections over the whole sequence. As we can see in Figure 4, both SFM and SFM+GR increase the tracking accuracy when compared to DIST.

Outliers. With an initial set of detections of GT with 2% missing data, tests are performed with $[0, 10, 20, 30, 40, 50]$ percentage of outliers added in random positions over the ground plane. In Figure 4, the results show that the SFM is especially important when the tracker is dealing with outliers. With 50% of outliers, the identity switches with SFM+GR are reduced 70% w.r.t the DIST results.

Noise. This test is used to determine the performance of our approach given noisy detections, which are very common mainly due to small errors in the 2D-3D mapping. From the GT set with 2% missing data, random noise is added to every detection. The variances of the noise tested are $[0, 0.002, 0.004, 0.006, 0.008, 0.01]$ of the size of the scene observed. As expected, group information is the most robust to noise; if the position of pedestrian A is not correctly estimated, other pedestrians in the group will contribute to the estimation of the true trajectory of A.

These results corroborate that having good behavioral

models becomes more important as the observations deteriorate. In Figure 5 we plot the tracking results of a sequence with 12% simulated missing data. Only using distance information can see identity switches as shown in Figure 5(a). In Figure 5(b) we can see how missing data affects the matching results. The matches are shifted, this chain reaction is due to the global optimization. In both cases, the use of SFM allows the tracker to interpolate the necessary detections and find the correct trajectories. Finally, in Figure 5(c) we plot the wrong result which occurs because track 3 has two consecutive missing detections. Even with SFM, track 2 is switched for 3, since the switch does not create extreme changes in velocity. In this case, the grouping information is key to obtaining good tracking results. More results are shown in Figure 7, first row.

5.2. Tracking results

We evaluate the proposed algorithm on two publicly available datasets: a crowded town center [2] and the well-known PETS2009 dataset [9]. We compare results with (1) [2] using the results provided by the authors; (2) [28], a tracking algorithm based on network flows, for which we use our own implementation of the algorithm; (3) [22], which includes social behavior, using the code provided by the authors; (4) [27], which includes social and grouping behavior, using our own implementation. For a fair comparison, we do not use appearance information for any method.

5.2.1 Town Center dataset

We perform tracking experiments on a video of a crowded town center [2]. To show the importance of social behavior and the robustness of our algorithm at low frame rates, we track at 2.5fps (taking one every tenth frame).

	DA	TA	DP	TP	IDsw
HOG Detections	63.1	–	71.9	–	–
Benfold et al. [2]	64.9	64.8	80.5	80.4	259
Zhang et al. [28]	66.1	65.7	71.5	71.5	114
Pellegrini et al. [22]	64.1	63.4	70.8	70.7	183
Yamaguchi et al. [27]	64.0	63.3	71.1	70.9	196
Proposed	**67.6**	**67.3**	**71.6**	**71.5**	**86**

Table 1: Town Center sequence.

Note, the precision reported in [2] is about 9% higher than the input detections precision; this is because the authors use the motion estimation obtained with a KLT feature tracker to improve the exact position of the detections, while we use the raw detections. Still, our algorithm reports 64% less ID switches. As shown in Table 1, our algorithm outperforms [22], which includes social behavior, and [27], which includes also grouping information, by almost 4% in accuracy and with 50% less ID switches. In Figure 6 we can see an example where [22, 27] fail. The errors are created in the greedy phase of predictive approaches, where people fight for detections. The red false detection in the first frame takes the detection in the second frame that should belong to the green trajectory (which ends in the first frame). In the third frame, the red trajectory overtakes the yellow trajectory and a new blue trajectory starts where the green should have been. None of the resulting trajectories violate the SFM and GR conditions. On the other hand, our global optimization framework takes full advantage of the SFM and GR information and correctly recovers all the trajectories. More results of the proposed algorithm can be seen in Figure 7, last row.

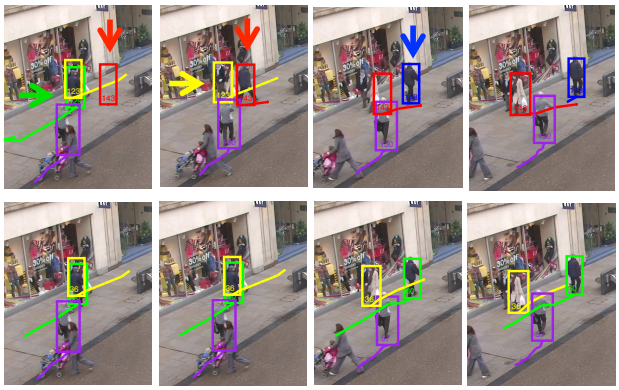


Figure 6: Predictive approaches [22, 27] (first row) vs. Proposed method (second row)

5.2.2 Results on the PETS2009 dataset

In addition, we perform monocular tracking on the PETS2009 sequence L1, View 1 and obtain the detections using the Mixture of Gaussians (MOG) background subtraction method. We obtain a tracking accuracy of 67% compared to 64.5% for Pellegrini et al. [22].

This dataset is very challenging from a social behavior point of view, because the subjects often change direction and groups form and split frequently. Since our approach is based on a probabilistic framework, it is better suited for unexpected behavior changes (like destination changes), where other predictive approaches fail [22, 27].

6. Conclusions

In this paper we argued for integrating pedestrian behavioral models in a linear programming framework. Our algorithm finds the MAP estimate of the trajectories total posterior including social and grouping models using a minimum-cost network flow with an improved novel graph structure that outperforms existing approaches. People interaction is persistent rather than transient, hence the proposed probabilistic formulation fully exploits the power of

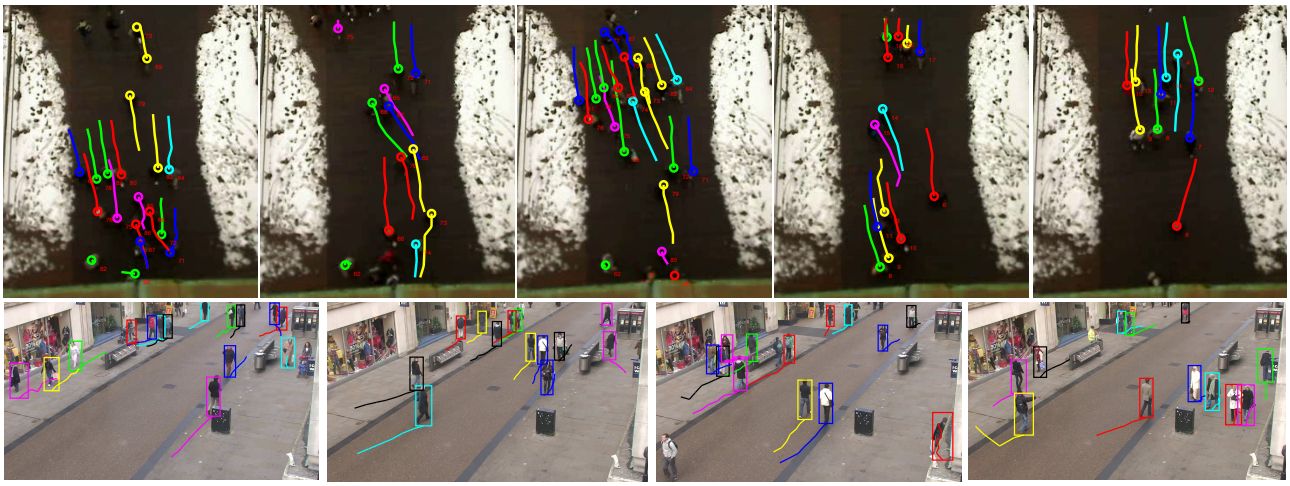


Figure 7: *First row*: Results on the BIWI dataset (Section 5.1). The scene is heavily crowded, social and grouping behavior are key to obtaining good tracking results. *Last row*: Results on the Town Center dataset (Section 5.2.1).

behavioral models as opposed to standard predictive and recursive approaches such as Kalman filtering.

Experiments on three public datasets reveal the importance of using social interaction models for tracking in difficult conditions such as in crowded scenes with the presence of missed detections, false alarms and noise. Results show that our approach is superior to state-of-the-art multiple people trackers. As future work, we plan on extending our approach to even more crowded scenarios where individuals cannot be detected and therefore features might be used as in [6].

References

[1] S. Ali and M. Shah. Floor fields for tracking in high density crowded scenes. *ECCV*, 2008. 1

[2] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. *CVPR*, 2011. 7

[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. *CVPR*, 2006. 1

[4] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 2011. 1

[5] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. van Gool. Robust tracking-by-detection using a detector confidence particle filter. *ICCV*, 2009. 1

[6] G. Brostow and R. Cipolla. Unsupervised detection of independent motion in crowds. *CVPR*, 2006. 8

[7] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. *ECCV*, 2010. 2

[8] G. Dantzig. *Linear programming and extensions*. Princeton University Press, Princenton, NJ, 1963. 5

[9] J. Ferryman. Pets 2009 dataset: Performance and evaluation of tracking and surveillance. 2009. 7

[10] W. GE, R. Collins, and B. Ruback. Automatically detecting the small group structure of a crowd. *WACV*, 2009. 2

[11] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282, 1995. 2, 5

[12] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. *CVPR*, 2007. 1

[13] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation for face, text and vehicle detection and tracking in video: data, metrics, and protocol. *TPAMI*, 31(2), 2009. 5

[14] R. Kaucic, A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. *CVPR*, 2005. 1

[15] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *TPAMI*, 2005. 1

[16] B. Leibe, K. Schindler, N. Cornelis, and L. van Gool. Coupled detection and tracking from static cameras and moving vehicles. *TPAMI*, 30(10), 2008. 1

[17] M. Luber, J. Stork, G. Tipaldi, and K. Arras. People tracking with human motion predictions from social forces. *ICRA*, 2010. 2

[18] A. Makhorin. Gnu linear programming kit (glpk). *http://www.gnu.org/software/glpk/*, 2010. 5

[19] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. *CVPR*, 2009. 2

[20] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking - linking identities using bayesian network inference. *CVPR*, 2006. 1

[21] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2007. 2

[22] 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, 5, 7

[23] S. Pellegrini, A. Ess, and L. van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. *ECCV*, 2010. 2

[24] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. *CVPR*, 2011. 1

[25] P. Scovanner and M. Tappen. Learning pedestrian dynamics from the real world. *ICCV*, 2009. 2

[26] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. *IJCV*, 75(2), 2007. 1

[27] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where are you going? *CVPR*, 2011. 1, 2, 7

[28] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. *CVPR*, 2008. 1, 2, 4, 7