Machine Vision and Applications (2017) 28:341–359
DOI 10.1007/s00138-017-0832-8

ORIGINAL PAPER

Motion priors based on goals hierarchies in pedestrian tracking applications

Francisco Madrigal[1] · **Jean-Bernard Hayet**[1]

Received: 13 April 2016 / Revised: 15 November 2016 / Accepted: 2 March 2017 / Published online: 15 March 2017

Abstract In this paper, the problem of automated scene understanding by tracking and predicting paths for multiple humans is tackled, with a new methodology using data from a single, fixed camera monitoring the environment. Our main idea is to build goal-oriented prior motion models that could drive both the tracking and path prediction algorithms, based on a coarse-to-fine modeling of the target goal. To implement this idea, we use a dataset of training video sequences with associated ground-truth trajectories and from which we extract hierarchically a set of key locations. These key locations may correspond to exit/entrance zones in the observed scene, or to crossroads where trajectories have often abrupt changes of direction. A simple heuristic allows us to make piecewise associations of the ground-truth trajectories to the key locations, and we use these data to learn one statistical motion model per key location, based on the variations of the trajectories in the training data and on a regularizing prior over the models spatial variations. We illustrate how to use these motion priors within an interacting multiple model scheme for target tracking and path prediction, and we finally evaluate this methodology with experiments on common datasets for tracking algorithms comparison.

Keywords Multi-pedestrian tracking · Interacting multiple model methods · Scene understanding · Particle filter

Electronic supplementary material The online version of this article (doi:10.1007/s00138-017-0832-8) contains supplementary material, which is available to authorized users.

✉ Francisco Madrigal
pacomd@cimat.mx

Jean-Bernard Hayet
jbhayet@cimat.mx

[1] Centro de Investigación en Matemáticas (CIMAT A.C.), Jalisco S/N, Col. Valenciana, 36240 Guanajuato, Gto, Mexico

1 Introduction

Automated scene understanding is a key element in making possible that the thousands of video surveillance systems in operation all around the world could be used to their full potential, in particular to trigger early alarms, without the need for a constant supervision of human operators. In this work, we propose a methodology for modeling, tracking, and predicting human motions within crowds. Our motion models are fed by data received from a video surveillance camera and are goal oriented. The main idea is to build goal-oriented statistical motion models that could be used as priors for both tracking and path prediction, based on a coarse-to-fine modeling of the target goal. This way, we infer the goal of each pedestrian present in the visible part of the environment, and we can monitor closely targets that will likely reach one specific zone of interest.

Our main underlying assumption is that pedestrians do, most of the time, follow a path with a precise goal well in mind. Also, given a specific goal, we assume that the typical trajectories of pedestrians in a scene are more likely to be modeled by simple statistical models. However, under the observation of a limited field of view surveillance camera, the notion of *goal* is not always easy to define. In this work, goals will be either areas where targets leave or enter the field of view of the camera, or crossroads within the targets paths, i.e., areas where paths have often abrupt changes of direction. Moreover, as the notion of goal may be relatively coarse when one is located far from it, and as it could be refined when getting closer to it, our idea is to extract a hierarchical set of representative goals, with "coarse" goals (root) and refined ones (sub) and learn a statistical model that represents the typical dynamics of those pedestrians *sharing the same goal*. Then, in a second time, the learned models are used as statistical priors in visual tracking so as to cope better with the traditional problems of visual tracking, such as occlusions. We also use these priors to perform prediction on whether one target will reach one specific goal, so as to launch preventive alarms. The main elements of our system are illustrated in Fig. 1.

This paper extends a work previously published in [28]. In particular, in this version, we extend the concepts presented in the aforementioned work in two directions: (1) we propose a new scheme for handling possible goals for pedestrians moving in a scene, with a hierarchical structure that allows to refine the predictive models when getting closer to them, and (2) we propose and evaluate a framework for target path prediction that integrates the uncertainties on the prediction horizon.

This paper has the following structure: Related work is presented in Sect. 2; then, an overview of our methodology is given in Sect. 3; in Sect. 4, we describe the goal-wise statistical models for our motion priors; in Sect. 5, we give an insight about how visual tracking uses these motion priors; Sect. 6 describes a proposal that use our goal-wise motion prior models to predict the pedestrian path and, last, Sect. 7 presents evaluations of our methodology, both on their tracking and prediction performance.

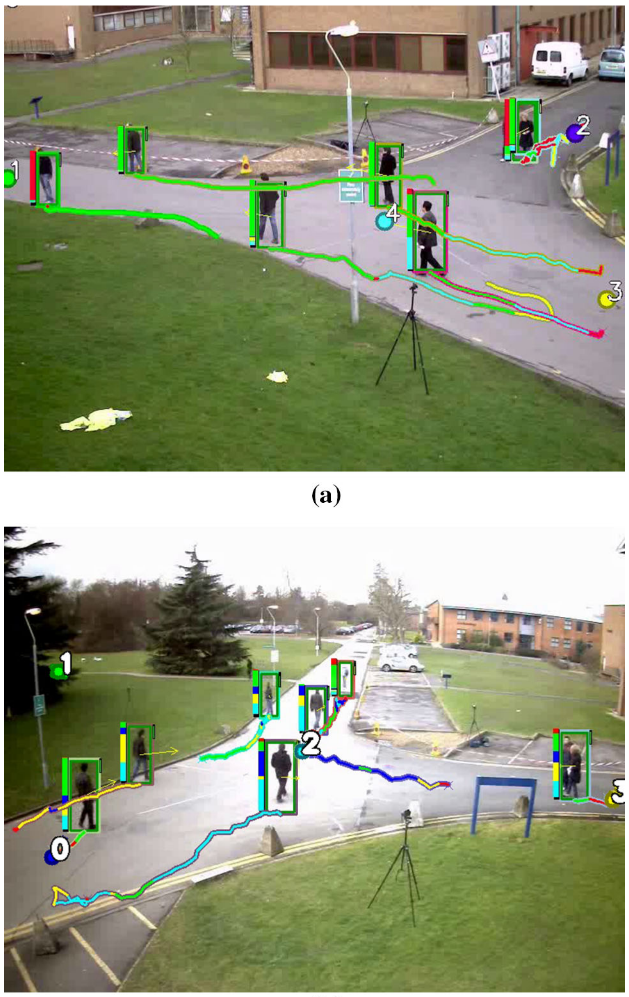


Fig. 1 Our system performs simultaneous tracking and goal estimation through model selection, with an interacting multiple model scheme integrated within a Bayesian tracking framework. The multiple models in competition use goal-wise motion priors learned from training data. In that figure, we display target visual tracks at some point of a test sequence, and the *color* indicates the most probable goal at each instant. **a** View 1, **b** View 2

2 Related work

In the fields of mobile robotics and computer vision, there have been many works focused on extracting statistical models for the motion of pedestrians and using them as motion priors in tracking applications. Here we mention a few of them.

In [7], a framework to perform the tracking of people from a mobile robot is defined around the idea of *motion patterns*. The training data are acquired with a laser scanner as fixed-length trajectories of individuals wandering in the environment. The trajectories are clustered with a variant of expectation–maximization (EM) into representative trajectories (coined as motion patterns). A hidden Markov model is then learned to give a prediction of the people position. Here, we do not focus on statistical models at the level of trajectories but instead at the level of locations.

In [25], the context is very similar to [7], as the application is the tracking of people in indoor spaces with laser scanners, and an approach based on Kalman filters is proposed. It includes social forces as described in the simulation community, with driving forces derived from the intended velocity and direction, reactive forces derived from other pedestrians and obstacles, and contact forces derived from the environment. The driving force is defined based on the notion of a virtual goal, namely a short-term goal to reach. This is an important difference with our work, since we reason on long-term goals instead. Also, the reactive and contact forces depend on the knowledge of a map of the environment, which we do not use here.

In [26], in the same context, the authors propose a place-dependent so-called *spatial affordance map*, that stores a grid of Poisson models for the appearance/disappearance rate of pedestrians at a specific place. The parameters in each cell are built with data acquired during a learning phase. Moreover, they include a place-dependent predictive model to be used during occlusion events, based on a prior on the walkable area built with the same track confirmation events used in the training phase. In our work, we also rely on the place-dependent nature of the prior motion models to be used, and we use a similar strategy for the case of visual tracking. However, we do not use an explicit probabilistic model for the map, nor rely on large databases of samples to build the models and we introduce the concept of hierarchical goals to produce several layers of prior maps that are used in an interacting multiple model.

In [5], in the context of providing algorithms for a safe cooperation between walking humans and industrial robots within an assembly line, the authors use the aforementioned work [7] to classify trajectories of humans walking within the working area. Then, with the help of a hidden Markov model, the learned patterns are used to predict which one of the preselected regions of the environment will be attained by the walking human, and at which time. This work shares some elements with ours, but lacks of generality, as, in particular, the interest zones are entirely predefined.

In [10], the idea of conditioning the motion models to the goal of pedestrians, that lies at the core of our approach, is developed; the set of goals is learned from a database of trajectories, and a motion planner generates trajectories to the goal that give the main trend of the motion direction in the tracking phase; however, for the planning purposes, it needs a full map of the environment, which we do not assume to have in our case. Instead, we rely on statistical models built with the training data to learn the most probable trajectories directions, without the need of a motion planner.

In [1], based on the analysis of very large train stations datasets, the authors use the linear programming methodology of [23] to perform detection-based tracking among crowds. A social affinity model learned from the collected data enforces social behaviors, which are incorporated as transition costs between tracklets. Also, priors between typical entrance/exit pairs are used. In [22], a similar tracking technique is integrated through random forests with interaction models learned automatically from low-level features.

The work in [8] also shares a number of characteristics with ours. The targets motion is modeled through so-called *behavioral maps*, learned from video data. What is different from our work is that the statistical model built this way contains only position-related information and also that the distributions learned at each point are conditioned to behavioral maps and positions. In our case (see Sect. 4), the conditioning is relative to the targets position and velocity. Hence, when “complex” behaviors are represented with different map transitions in [8], they can be modeled within a single goal-specific map in our model. This allows us for example to simply encode a piece of information like: “at that particular place, people often make u-turns”. Another difference is that, instead of using interpolation of map transitions and local motions, based on their evaluations at a limited number of control points, we use more training data than only tracklets and rely on regularization to cope with missing information.

In [18], the authors predict the long-term behavior of pedestrians in a public space, where these pedestrians may be the object of interactions with service robots. By analyzing hours of video footages from the corresponding scene, the authors first determine goals in the scene, which are points at which trajectories cross relatively often. As in our work, a statistical model for typical motion directions is learned on a cell basis with an n-gram representation of the motion direction, which allows to determine the location of the goals with an EM-like scheme. Then, the prediction of the next goal to reach is done within a Bayesian framework. In comparison, our statistical model is also aimed at predicting the goal of a moving target, but it includes additional information about velocity magnitudes, conditionally to the motion direction, which may give better estimates, e.g., of time-to-goals. Moreover, we use a regularization framework that avoids the use of huge amounts of training data that are needed in their approach.

In [13], the authors develop the notion of semantic regions, which are clusters of frequent trajectories among sets of trajectories sharing the same entrance/exit. These semantic regions are then used for path prediction. For this application to crowd analysis, they do not rely on pedestrian detection or tracking, but on a grid-like decomposition of the scene, similar to ours, and histogram-based motion models defined over the grid, and built with optical flow data. Entrances, exits, and semantic regions are discovered automatically by simulating particles over the learned distributions.

In the work of [33], a hidden Markov model models the probabilistic structure of possible states (positions) in relation to the goals pursued by the moving targets. Goals are determined (in a similar fashion than ours) by clustering termination points of trajectories. The growing neural gas algorithm is used to update this model structure and its parameters incrementally, making this approach a learn-and-predict framework, instead of a learn-then-predict one. The adaptive nature of this approach is a feature we want to integrate in the long run on our own system; however, in this paper, we have focused mainly on (1) the probabilistic model by itself, (2) the hierarchical structure of goals that allows us to control the computational times in tracking and prediction, and (3) the use of techniques (regularization) that allows not to depend on too many training data.

In [2], tracking on very dense crowds is handled by simulating collections of interacting particles, driven by collective behavioral patterns encoded by potential functions called floor fields. These fields capture both “normal” crowds behaviors, local behaviors of the crowd around the tracked target, and the repulsion induced by the boundaries around the walking areas. In our approach, first, we keep a *probabilistic* representation for the prior, conditioned to positions and velocities, which allows to model different local motion tendencies; also, we do not handle very dense crowds. The approach shares features with ours, as for example, the grid-based representation of the motion priors.

In an another related field, at the intersection of mobile robotics and study on human locomotion, researchers have shown that, for a pair of configurations (position and orientation) in the plane and without obstacle, pedestrians tend to choose a common walking pattern among an infinite number of solutions [4,17]. In particular, the aforementioned works confirmed the intuition that human walking can be reasonably approximated by a differential system with the classical non-holonomic constraint.

3 Approach overview

Here follows a general overview of our three-step methodology. The basic input we need is a set of training video sequences from the observed scene, with ground-truth (GT) extracted trajectories. Then:

– We process the training video sequences to produce a hierarchy of goals with the GT pedestrians trajectories. This is done automatically (see Sect. 4.1). A typical output of this step appears in Fig. 5.
– We learn one motion statistical model, coined as "motion prior field" (MPF), per goal in the learned hierarchy. Each MPF is learned by collecting motion information from GT data (i.e., labeled trajectories) and optical flow data and by regularizing the obtained model (see Sect. 4.5). Examples of learned models are given in Fig. 9a, b.
– We run our tracking system while taking advantage of this learned multiple-goal motion prior field (MG-MPF) (see Sect. 5) to perform target tracking while simultaneously inferring the goal a target is aiming at. Note that we will illustrate the goal inference with a visual tracking system, but any form of tracking (i.e., from a mobile robot) could be used.

4 Multiple-goal motion prior field

In this proposal, the inputs are a set of ground-truth target trajectories taken from the same camera with which the automatic visual tracking will be done. These GT trajectories are automatically segmented in terms of goals and sub-goals (Sects. 4.1, 4.2, and 4.4). Then, they are used to learn the set of goal-based motion prior fields (MPF, Sect. 4.5). Figure 2 depicts an example of manually labeled GT trajectories, taken from the PETS2009 database, that we will describe in Sect. 7. The only requirement is that as many individual trajectories as possible should be available for the learning stage. However, we use much less than the hours of data required in many approaches mentioned in Sect. 2, such as [18]. As explained below, we first automatically extract a hierarchical set of goals within the scene and we generate

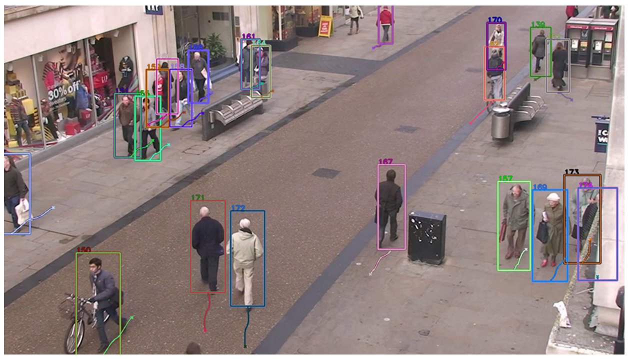


Fig. 2 Ground-truth (GT) example: GT trajectories for all visible pedestrians in a frame of the Town Centre video sequence. Those trajectories are used in the learning step and are extracted from [30]

from each GT trajectory a set of sub-trajectories which are associated to one of the goals. Figure 5 illustrates such a set of sub-sequences, where the color indicates the goal id (one color per goal).

4.1 Determination of goals

The goals are generated automatically by clustering a set of key points extracted from the GT trajectories. Let us state that, given a GT trajectory, we first select as key points the entrance and exit points in the field of view of the camera, as several other approaches do. In addition, we select any point where there exists a strong change of orientation. These points may represent sudden goal changes, or points where physical constraints impose strong curvatures to the trajectories. This could be for example because of the curvature of a road. To detect these changes of orientation, we detect local maxima of orientation variations along the ground-truth paths. A set of key points extracted in this manner is depicted as the set of crosses in Fig. 4.

The next step is the automatic determination of clusters among these key points. A critical element—and a quite classical problem— is to select the number of clusters k to search for in the clustering method. This number is estimated through the "jump method" [32]. Given an upper bound n to the number of clusters, it makes use of a standard clustering algorithm such as K-means [19] for $k = \{1 \dots n\}$ to compute the distortion of the resulting clusters. It approximates the features distribution through a mixture of k Gaussians of fixed variance. The distortion is evaluated as the minimal average Mahalanobis distance among clusters:

$$d_k = \frac{1}{p} \min_{\mu_1 \dots \mu_k} E_{\mathbf{r}} \left[(\mathbf{r} - \mu_{\mathbf{r}})^T \Gamma^{-1} (\mathbf{r} - \mu_{\mathbf{r}}) \right],$$

where $\mathbf{r} = (x, y)$ are the GT key points and $p = 2$ the dimension of $\mathbf{r}$. Γ is a fixed covariance matrix, $\mu_1 \dots \mu_K$

are the centroids computed by K-means and $\mu_{\mathbf{r}}$ is the closest centroid to the key point $\mathbf{r}$.

With successive computed values of d_k for all k between 1 and n, this algorithm generates a distortion curve which tends to zero when k becomes large. As in [32], we select k^* (the optimal k) as the position of the highest decrease along the d_k, taken at some power Y. The pseudo-code for the jump method is as follows:

Algorithm 1 Jump method

```
Let $Y = (p/2)$ .
Set an array $A$ of size $n + 1$ with $A[0] = 0$
**for** $k = 1 \ldots n$
    – Apply k-means.
    – Set $d_k$ as the distortion of the clustering (see above)
    – $A[k] = d_k^{-Y}$
Compute the variations along the distortion curve.
**for** $k = 1 \ldots n$
    – $J(k) = A[k] - A[k-1]$.
**Return** $k^*$ maximizing $J$.
```

We use all the key points in this process. However, some points can be spurious or noisy, causing the jump method to group them into small clusters, i.e., 5 points or less. In that case, we assign them to another cluster by finding the closest centroid to each point. We define a new cluster from a key point label if the minimal distance to existing centroids is higher than a given threshold. Finally, any cluster with too few points is discarded and so is the corresponding sub-trajectory.

Then, K-means is applied again over all the key points, with $k = k^*$ and starting with the centroids found in the previous step. Finally, the goals are defined as the centroids of the key points clusters. They appear as circles in Fig. 3 and their covariance is depicted with the ellipses. In the offscreen region, at the bottom of this image, we can observe some clusters/trajectories. This is because the ground-truth trajectories (depicted as the detected bounding box feet points) are inferred from the detected head of pedestrian (the pedestrian height being the average height of an adult). Thus, a trajectory starts when a pedestrian head is detected and it ends when the head leaves the field of view.

4.2 Hierarchies of goals

After computing the clusters, our proposal puts in competition all the potential goals through goal-dependent motion models (see Sect. 4.5). However, we observe that some of the obtained centroids are spatially close enough to correspond to the same real entrance/exit, see for example Fig. 3. Hence, for targets far enough from those centroids,

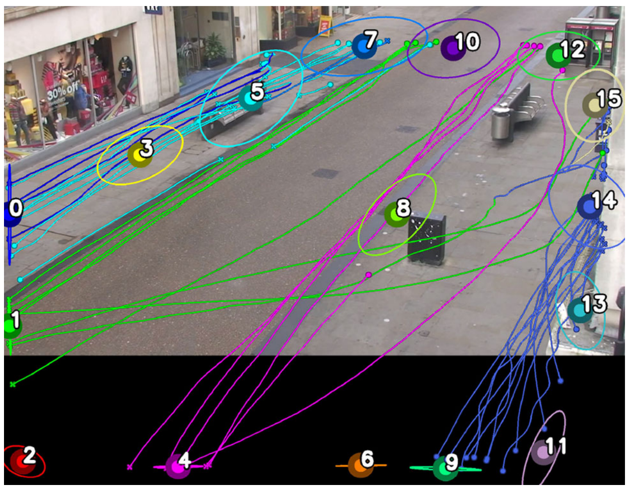


Fig. 3 Clusters obtained with training trajectories from the Town Centre video sequence [6]. The *circles* and the *crosses* depict the starting and ending points of each sub-trajectory, respectively. The *colors* are related to each of the six clusters, and the sub-trajectories take the colors corresponding to their associated goals

with one of it being the real goal, there is computational waste at estimating motion distributions that differ significantly only when the targets become close to their goals. Moreover, as will be described in the next sections, the complexity of our approach is linear in the number of goals in competition. To make it scalable, we need to control the number of models simultaneously active in the tracking module. That is why we propose a hierarchical goal representation that both reduces the computational cost of our overall approach and maintains the precision of the motion prediction. We build it in a bottom-up fashion as explained hereafter.

This process consists in applying again the steps of Sec. 4.1 using the cluster centroids instead of the key points. First, we compute the optimal number of clusters through the jump method, with a range of search between 1 and the current number of clusters. The results of this second clustering gives us upper-level goals. A key point is finally associated to one or more clusters at different levels of clustering, forming a tree of goals. The process is repeated while necessary, i.e., until the new optimal number of clusters does not evolve anymore. In all our examples, only two levels are used. Hence, in the remaining, the n^g top-level centroids are called root-goals and bottom-level centroids are called sub-goals. There are $n^s(g)$ such goals for a given root-goal g. Figure 4 shows the two-level clustering resulting from this proposal. The lines at ground-level represent target trajectories. The crosses depict the key points, while circles are the root-goals, squares are the sub-goals, and ellipses are their covariance. We can observe that some goals have two sub-goals (cluster 2, in red), while others do not have any (cluster 3 in yellow) and some have 3 (bottom-right cluster

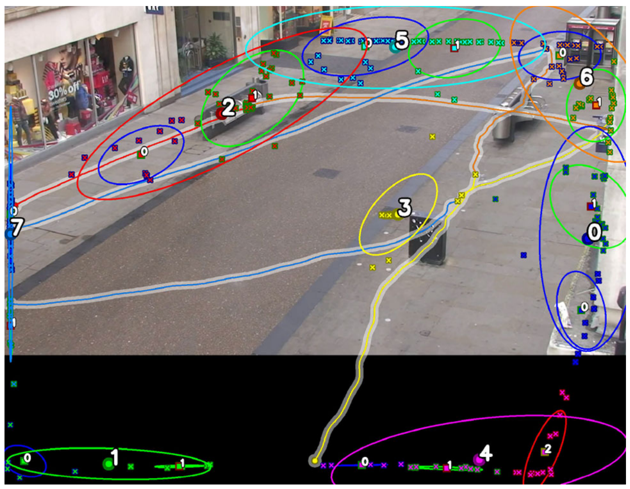


Fig. 4 Ground-truth (GT) trajectories key points and their corresponding clusters in the Town Centre video sequence. The *ellipses* with numbers depict the goal id, each with a *different color*. Each key point (cross) is associated to one goal and is drawn with the same color as the goal target. The ground-level lines depict GT trajectory segmented according to the associated goal

4, in magenta). In the following, we will mainly refer to the sub-goals indices with the letter s, and to the root-goals with the letter g. We will also refer to the root-goal associated to s as $g(s)$.

4.3 Goals zones of influence

As mentioned above, sub-goals s are taken into account only when a target is close to a goal zone of influence. These zones are computed through the covariance matrix of the key points associated to the root-goal $g(s)$, assuming that they follow a Gaussian distribution. This zone, referred to as $R_{g(s)}$, is defined as a 2σ ellipse, given the covariance matrix eigenvalues. An example is given in Fig. 6. As some zones could be too narrow, like ellipse 7 (left side) of Fig. 4, we project the corresponding ellipses to the world plane, with the camera calibration, and set any of the axis to a minimum size of 2 m if they have a length inferior to that value. Then, we re-project those rectified ellipses back to image plane.

4.4 Segmentation of GT trajectories

Finally, the GT trajectories are segmented in function of the determined key points and labeled with the id of their associated goals. In other words, a trajectory is segmented from its beginning until the first key point found or between two key points. Then, each fragment is labeled according to the labels of the final key point, which consists of the branch computed by the hierarchical technique of Sect. 4.2. This

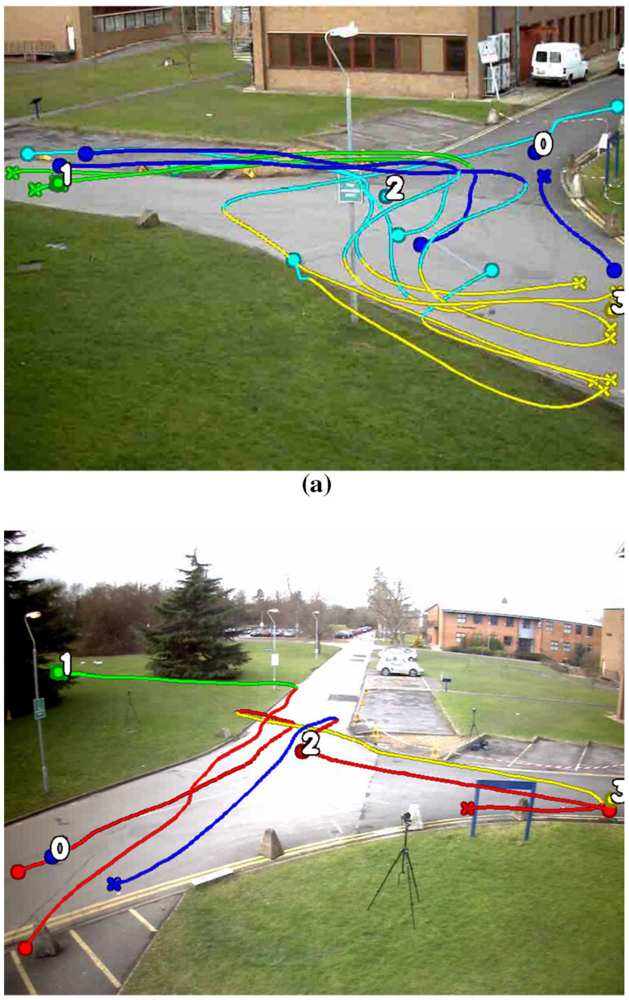
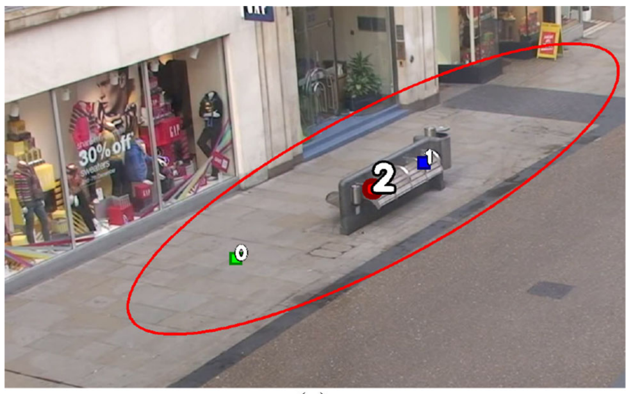


Fig. 5 Clustering of the training trajectories with samples from the PETS2009 S2L2 video sequence [30]. The *circles* and crosses depict the starting and ending points of each sub-trajectory, respectively. The colors are related to each of the four clusters and the sub-trajectories take the colors corresponding to their associated goals. **a** View 1, **b** View 2

way, all segments have their leaf-root branch from the goal hierarchy.

Thus, they can be used for the learning of goal-dependent motion priors. Typically, this process may generate goals in the middle of the scene where the pedestrians are walking to, or goals at the exit zones. Figs. 3, 4 and 5 show the trajectory segmentation resulting from one- and two-level goal hierarchies within the TownCentre dataset and one-level goal hierarchy within the PETS2009 dataset, respectively. The lines at ground-level represent target trajectories already segmented, and their color is the same as the top-level goals. We can observe goal switches as they change direction, i.e., the yellow line at the bottom of Fig. 4 continues its upward path until it reaches goal 3 (yellow) and then switches to goal 6 (orange) to finally attain goal 7 (light blue).


(a)

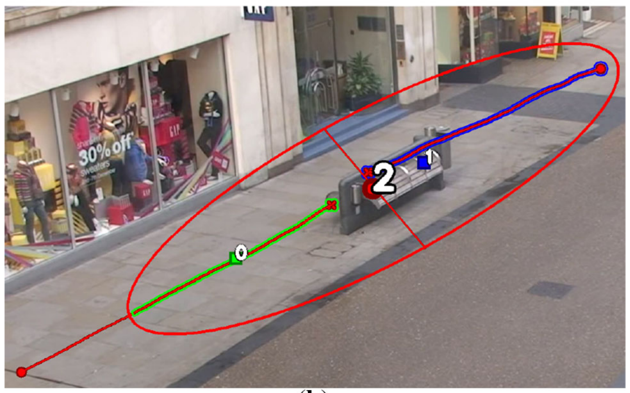

(b)

Fig. 6 Illustration for the learning of sub-goal motion prior field. **a** The 2σ ellipse associated to root-goal 2 is computed from the covariance matrix learned from the key point clustering process. **b** The MPFs for sub-goals 0 and 1 are only learned within the ellipse R_2 (which is $R_0 = R_1 = R_2$). **a** Root-goal 2 and its sub-goals 0 and 1, **b** MPF learning from GT trajectories

4.5 Learning goal-specific motion prior fields

The learning step is adapted from a framework we have developed in a previous work [29], and that we briefly review here. Observe that our new clustering approach described before gives us a hierarchy of goals; this allows to learn goal-driven motion priors and to take advantage of the hierarchical structure of goals. For each *root-goal* (i.e., a goal which is not a sub-goal of another), we compute a motion prior model as follows.

4.5.1 Motion prior fields for root-goals

First, we detect visual features along the video sequences and track them as long as possible with a sparse optical flow algorithm, i.e., Kanade-Lucas-Tomasi tracker (KLT) [9]. This gives us a set of *tracklets* (short, pointwise trajectories). Each tracklet is associated to its closest GT sub-trajectory, so that we can label the tracklet with a root-goal id (the one associated to the closest GT sub-trajectory) and use it as a data to build the motion prior field (MPF) of the respective root-goal.

The MPF is a statistical model associated to each root-goal. We denote the state for a single target as a vector including the target position, orientation and its velocity (in polar coordinates) $\mathbf{X}_t = [x_t, y_t, \theta_t, v^l_t, v^\theta_t]^T$. To define the MPF for each goal, we first divide the full image into a number of spatial cells and divide each 2D cell again into D possible discretized entry velocity orientations v^θ_q. This way, for a goal g, we obtain a 3D grid of C cells $\left\{\mathbf{c}_q = (\mathbf{r}_q, v^\theta_q)\right\}_{1\le q\le C}$. Then, in each cell, based on the goal-specific data from the KLT tracklets, and on the collected ground-truth trajectories, a state transition distribution conditioned to the root-goal g and the current cell is built as [29]:

$$p\left(\mathbf{X}_t|\mathbf{X}_{t-1}, g\right) \approx p\left(v^l_t|\mathbf{c}_q\left(\mathbf{X}_{t-1}\right), g\right) \times p\left(v^\theta_t, \mathbf{r}_t|\mathbf{c}_q\left(\mathbf{X}_{t-1}\right), g\right),$$

where $\mathbf{r} = (x, y)$ is the position. Implicitly here, the pedestrians motions are non-holonomic [4] ($v^\theta_t \approx \theta_t$). The model includes, for each goal g and each cell $\mathbf{c}_q$:

– A discrete distribution for the orientation transitions, $p(v^\theta_t, \mathbf{r}_t|\mathbf{c}_q(\mathbf{X}_{t-1}), g)$, where $\mathbf{c}_q(\mathbf{X}_{t-1})$ is the cell associated to state $\mathbf{X}_{t-1}$. For the D possible directions, a target may follow from the current cell, we keep the corresponding frequency among the training data.
– The first two moments of $p(v^l_t|\mathbf{c}_q(\mathbf{X}_{t-1}), g)$ (as a continuous, Gaussian distribution), also computed from the training data.

Experimentally, we obtained acceptable results starting from $D = 8$ directions. We present a few raw results in Figs. 8a, b, where we have also superimposed some trajectories associated to the corresponding goal.

4.5.2 Motion priors fields for sub-goals

Once motion prior fields have been computed for each root-goal, we refine these MPFs for the sub-goals s with the following two guidelines: (1) far from any sub-goal, the motion prior will be given by the MPFs associated to root-goals g; only within the goal zone of influence $R_{g(s)}$ will the MPF specific to sub-goal s be estimated, with specific GT data labeled with its id; (2) the raw MPF for the root-goal is estimated at the same time as the raw MPF fields for its sub-goals. This is because the corresponding GT trajectories are already labeled for both the root- and sub-goals. Figure 6 depicts these ideas in the sub-goal MPF learning. The sub-goal zone of influence R_g is computed as described in Sect. 4.3. Therefore, sub-goal MPFs are learned

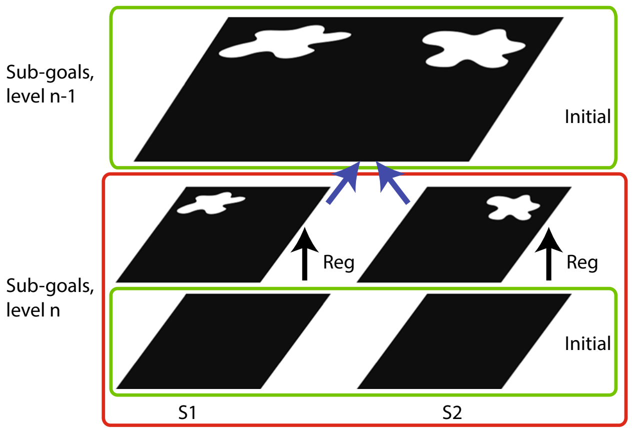


Fig. 7 Regularization scheme. First, we regularize the sub-goals models (*red square*). The optimization process is initialized with an empty histogram (*bottom green square*). Then, the regularized sub-goals MPFs are used as inputs to regularize the root-goals MPFs

only with labeled trajectories inside R_g. Figure 6b shows how the learning phase uses two trajectories, that both contribute to build the statistical model associated to root-goal 2 (inner red line). However, the (outer line) blue sub-trajectory also contributes to the model of sub-goal 1 and the (outer line) green sub-trajectory contributes to the model of sub-goal 0.

4.5.3 Regularization

It is clear that these raw local distributions within the MPFs are not smooth, mainly because of the scarceness of training examples gathered from the ground-truth dataset. As explained above, we do not expect here to be able to collect hours and hours of training data, as in [18]. That is why, as described in our previous work [29], we add a regularization phase (optimization over a Markov random field with smoothness terms) of the obtained data so as to complete the model when data are missing, by using a strong smoothness prior. The variables to optimize in this optimization scheme are the histogram entries. This process is done recursively by taking advantage of the hierarchical structure of our set of goals and sub-goals. We depict it in Fig. 7. We regularize the sub-goals priors first (within their corresponding zones of influence), i.e. the raw data illustrated in Fig. 8, and then, we proceed to build the regularized model for the common root-goal shared by these sub-goals as follows:

1. In all intersections of the supports of sub-goals sharing the same root-goal, merge (sum and normalize) the corresponding histograms;
2. Use this merged representation as an initial input for root-goal regularization.

Figs. 9a, b show some results after regularization. This is a key element in our approach, since it allows not to rely on too many data for this learning stage.

5 Target and goal tracking

Taking advantage of the motion priors built according to the previous Section, we propose an algorithm capable of track-

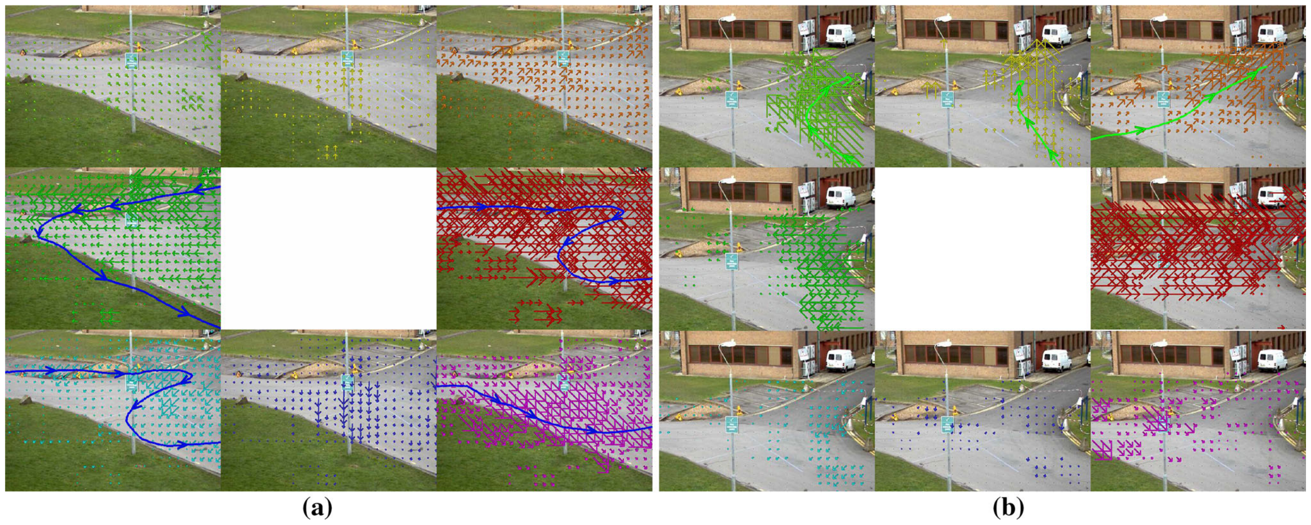


Fig. 8 A few raw, un-regularized results for the statistical models associated to goals 1 and 2. A subset of all trajectories are drawn to explain the learning behavior in some particular regions. Each *arrow* depicts the value of the marginal over input orientations v_i^θ at one fixed direction v_j^θ (the position of the image in the frame giving which direction) with the *arrow* magnitude representing the probability associated to this direction. The images have been cropped for better readability. **a** Goal 1, **b** Goal 2

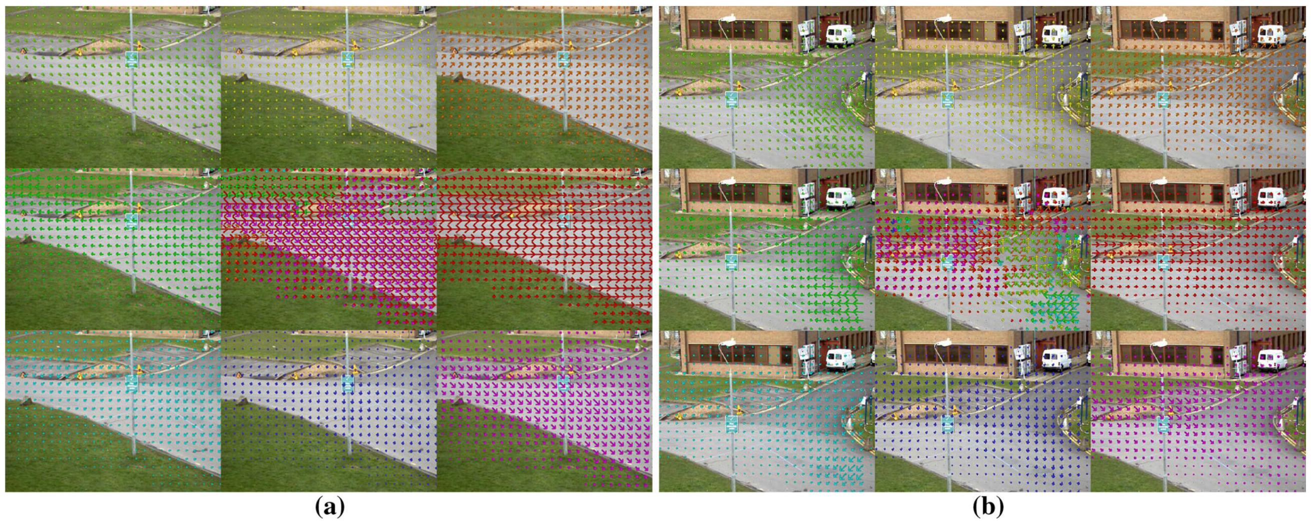


Fig. 9 Regularized results for goals 1 and 2. Each *arrow* depicts the value of the marginal over input orientations θ_i at one fixed direction θ_j (the position of the image in the frame giving which direction) with the arrow magnitude representing the probability associated to this direction. The central image gives the two main modes (as vectors) per cell. The images have been cropped for better readability. **a** Goal 1, **b** Goal 2

ing the pedestrians motion and also of estimating each target goal. For this purpose, we estimate the target state by following a Bayesian scheme, implemented through *particle filtering* jointly with an interactive multiple model (IMM) algorithm which allows to incorporate our aforementioned multiple-goal motion prior fields. The idea is that each model corresponds to one of the goals (root-goals or sub-goals) and that those compete with each other through the IMM. Thus, the tracker should explore at the same time all the possible goals a target could take. With this in mind, each IMM motion model uses one goal-specific MPF. At the end, the filter selects the model (hence, the goal) that fits better to the target trajectory and appearance.

5.1 Motion model: predicting targets next moves

In the particle filter, the posterior $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, which we cannot evaluate directly, is represented by a set of N weighted random samples generated from a known proposal distribution. Those weighted samples are called particles. The multi-modality is implemented by assigning one (out of G) goal-oriented motion prior field (MPF) to each particle, indicated by a label $g \in \{1 \dots G\}$. Thereby, a particle n with weight ω at time t is represented by $(\mathbf{X}_t^{(n)}, \omega_t^{(n)}, g^{(n)})$. The sampling step (i.e., the handling of the probabilistic motion model within the particle filter framework to set a new particle state) is realized per MPF by a transition model $tr_g(\cdot)$. The sampling of a new particle based on the previous state is given by

$$\mathbf{X}_t^{(n)} = tr_{g^{(n)}}(\mathbf{X}_{t-1}^{(n)}) + \nu, \qquad (1)$$

where ν is an additive Gaussian noise with fixed variance. The index $g^{(n)}$ indicates which goal model the particle n is associated to. For example, $g^{(n)} = g$ means that the particle n is associated to the goal g.

This probabilistic transition model $tr_g(\mathbf{X}_{t-1})$ includes the information of the MPF associated to goal g and is used as a sampling distribution as follows:

1. Determine the cell $\mathbf{c}_q(\mathbf{X}_{t-1}^{(n)})$ the current particle $\mathbf{X}_{t-1}^{(n)}$ corresponds to;
2. From the learned distribution of $g^{(n)}$, sample a direction v_*^θ that the target is likely to take;
3. Sample a velocity magnitude v_*^l in the same way;

Hence, the particle state $\mathbf{X}_t^{(n)}$ is updated as a linear combination of this sample and the current state:

$$tr_g(\mathbf{X}_{t-1}^{(n)}) = \begin{bmatrix} x_{t-1}^{(n)} + v_{t-1}^l * \cos(\theta_{t-1}^{(n)}) \\ y_{t-1}^{(n)} + v_{t-1}^l * \sin(\theta_{t-1}^{(n)}) \\ \theta_{t-1}^{(n)} + v_{t-1}^{\theta(n)} \\ (1-\alpha)v_{t-1}^{l(n)} \\ (1-\alpha)v_{t-1}^{\theta(n)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \alpha * v_*^l \\ \alpha * v_*^\theta \end{bmatrix},$$

where $\alpha = 0.5$ is a fixed value which controls the contribution of the current and sampled velocity.

Since all the described models apply on individual targets, when performing multiple-target tracking, we also include social forces terms, in that case, repulsion forces among close targets to avoid trajectory id switches [27].

Last, we have verified that the model presented up to now is not very useful when the targets are *static*. In that case, the only hint about the current goal pursued by the pedestrian is his orientation, observed through the image. As explained below (5.5), we use an observation model that allows to estimate the orientation [11], but that alone, is quite unreliable. Hence, we add another "goal" model (a random walk with zero velocity), that we refer to as *Constant Position model*, corresponding to static targets "without current goal" for handling this case.

5.2 Goal estimation

In the IMMPF methodology, the goal model $g \in \{1 \dots G\}$ contributes to the posterior estimation according to its importance, which is defined by its weight π_t^g. Each model g has N_g particles associated to it, with a total of $N = \sum_{g=1}^{G} N_g$ particles. The posterior is represented by considering both particle weights ($\omega_t^{(n)}$) and models weights (π_t^g):

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) = \sum_{g=1}^{G} \pi_t^g \sum_{n\in\psi_g} \omega_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(\mathbf{x}_t),$$

$$\text{s.t.} \sum_{g=1}^{G} \pi_t^g = 1 \text{ and } \sum_{n\in\psi_g} \omega_t^{(n)} = 1, \qquad (2)$$

where $\psi_g \overset{\text{def}}{=} \{n \in \{1 \dots N\} : g^{(n)} = g\}$ represents the indexes of the particles that belong to model g and π_t^g is the posterior probability associated to goal g given the set of observations $\mathbf{Z}_{1:t}$, i.e., $p(g|\mathbf{Z}_{1:t})$, which can be updated as

$$\pi_t^g = p(g|\mathbf{Z}_{1:t}) = \frac{\pi_t^g - 1\tilde{\omega}_t^g}{\sum_{g=1}^{G} \pi_{t-1}^g \tilde{\omega}_t^g},$$

where $\tilde{\omega}_t^g$ are the unnormalized weights of the particles, the terms π_t^g weigh each model contribution in the mixture, and $\tilde{\omega}_t^g = \sum_{j\in\psi_g} \tilde{\omega}_t^{(j)}$ sum up the unnormalized weights of all the particles associated to goal model g (i.e., a set ψ_g of particles).

Thus, the most probable goal of the target is the model g with the highest value π_t^g.

5.3 Resampling

We implement the resampling process as in [27]. First, we sample over all particles, following a common cumulative distribution function built with the weights of particles and goal models. Thus, the best particles from the best goal models are sampled more often, leaving more particles with goal models fitting better the observed target motion.

We also re-sample on a per model basis. Each model keeps at least a percentage of the whole set of particles to preserve diversity. Note that this resampling process manages the model transition implicitly, so no prior transition information is required (such as in [18]).

The resampling applied over all particles is done every 4 frames and the one per model every 5 frames.

5.4 Goal activation

When a target is far away from any goal, particles are used only with the n^g root-goal models to estimate their new state as described in Sect. 5.2. It is only when a target is inside a goal zone of influence that a particle switches to one sub-goal models s associated to a root-goal $g(s)$. This happens when the target tracker state (the mean of all particles) gets close to $g(s)$. In that case, any root-goal particle selects randomly a sub-goal s in such a way that each sub-goal associated to the root-goal $g(s)$ has finally the same number of particles. As, at that moment, there is no indication on which sub-goal will be the most probable, the selection is done uniformly and particles associated to a given root-goal g (among n^g) are distributed uniformly among its $n^s(g)$ sub-goals. Then, these particles are updated according to their sub-goal motion prior field instead of the root-goal one. Thereby, the number of goal models G in competition is: $G = n^g$ when the target is far from root-goals; and $G = n^g + n^s(g^*) - 1$ when it is inside region R_{g^*} (see Sect. 4.3), which includes all the $n^s(g^*)$ sub-goals associated to $g^*$and the n^g root-goals, except g^*. The resampling step is performed as in Sect. 5.3 and considers a minimum number of particles for each sub-goal too.

5.5 Observation model

We have focused our proposal in the learning of a motion model with multiple goals, more than in the definition of new observation models. Hence, we have implemented our tracking algorithm likelihood models as a combination of known models from [31] and [11]. The model proposed in [31] relies on HSV-space color histograms and motion histograms. From the work of [11], we include a likelihood term that helps us in discriminating target orientations with respect to the sensor. The body pose angle, which is discretized into eight directions, is evaluated with a set of multiple-level histogram of oriented gradients features (HOG) extracted from the image inside each target candidate position. They are decomposed into a sparse, positive linear combination of training samples, and the orientation likelihoods are determined based on the resulting weights. The resulting likelihoods are then classically used to update the particle weights.

6 Path prediction

Path prediction is performed similarly as the sampling step of Sect. 5.1. The predicted path is created by drawing samples ρ at each step from the proposal distribution, within a window of time $\tau \in [t \dots t+T]$. These samples are drawn conditionally to the model g. Starting from the target state at t:

$$\rho_t = \bar{\mathbf{x}}_{i,t} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_{i,t}^{(n)}.$$

The configurations $\rho_{\tau+1}$ are sampled as follows:

1. Let $\mathbf{c}_q = (v_q^\theta, \mathbf{r}_q)$ (velocity direction+position) the cell the sample ρ_τ corresponds to;
2. Sample a destination cell $\mathbf{c}_{q'}$ from a mixture of the learned motion prior $p(v_{q'}^\theta, \mathbf{r}_{q'}|v_q^\theta, \mathbf{r}_q)$ and a von Mises distribution (with mean θ_τ and variance 0.5).
3. Sample a velocity magnitude $v_{i,\tau+1}^{m,(n)}$ from $p(v_{i,\tau+1}^m, |\mathbf{v}_{i,\tau}^{(n)}, \mathbf{r}_{i,\tau}^{(n)})$;
4. Given cells $\mathbf{c}_q$ and $\mathbf{c}_{q'}$, build a cubic curve joining the center cells and tangent to the initial and final orientations (Hermite interpolation);
5. Apply this path to particle ρ_τ and sample a point on it around the position reached given $v_{i,\tau+1}^{m,(n)}$ (lateral noise is also added at this point).

In other words, from the current target state, we find the cell c_q where the target is (according to its position). Then, we sample a cell from $p(v_{q'}^\theta, \mathbf{r}_{q'}|v_q^\theta, \mathbf{r}_q)$. The cell location $\mathbf{r}_{q'}$ defines a pointing direction. We sample another pointing direction from a von Mises distribution centered around the current target orientation. The former extends the path from the learned MPF while the latter gives preference to path extensions according to the present target orientation. Finally, we select the cell $\mathbf{c}_{q'}$ from the mean of both samples. Also, we sample velocity magnitude from the learned distribution. Next, we perform a Hermite interpolation between both cell positions so that we can sample a new position that the target reaches given the sampled velocity magnitude.

Figure 10 shows the output of this process.

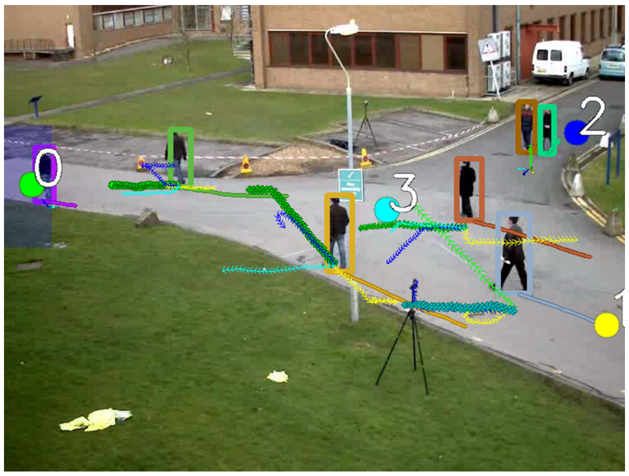


Fig. 10 Output of our path prediction system. Ground-level arrows depict the possible future positions of each target in some time horizon. The set of *arrows* depict the predicted path for a given goal and share with it the *same color*. The *arrow* thickness is associated to the model weights, such that the *thickest arrows* depict the most probable path

7 Results

We have tested the described proposal on two benchmarks of common use in video surveillance, known as PETS2009 [15] and Town Center [6]. The PETS09 dataset consists of a set of 8 outdoor camera video sequences. We evaluate the performance of our proposal using views 1 and 2. We train our proposal with all sequences of view 1 of this dataset but the test sequence. We do the same with view 2. We chose the sequence S2-L1:12-34 to test our learned multiple-goal motion model in pedestrian tracking. Although it is not the most difficult sequence in terms of crowdedness, it presents challenging situations for pedestrian tracking, such as occlusions by clutter, and is likely to generate identity switches. It includes 21 pedestrians, and some of them wear similar clothing or change their direction at some point. We have manually generated a ground-truth (GT) dataset, for each pedestrian in the scene over all frames of views 1 and 2 of the PETS2009 S2-L1 scenario. Both videos are processed independently.

The second dataset consists in a single high definition video, with 1920×1080 resolution and a frame rate of 25 fps, of a Town Centre street with a medium-dense crowd, with always around of sixteen pedestrians visible (see Fig. 11). The latter makes this sequence challenging due to the multiple occlusions that occur. Also, it contains a lot of pedestrians with different whereabouts, so that it is a convenient scenario for our hierarchical goal approach. It is provided with a hand-labeled ground-truth (GT) data for the first 7500 frames. In this GT, bounding boxes are determined based on the head regions, using the camera calibration and the average size of an adult. Hence, the GT trajectory at the ground-level could start outside of the image. We use the first 1000 frames for testing and the rest for training our goal-oriented motion field.

As mentioned above, our tracker is based on the particle filter which includes an additive Gaussian random noise when sampling new states $\mathbf{X}_t^{(n)}$ (Eq. 1). Thus, every execution of the tracking algorithm gives unique results. Hence, the results presented here are the median of 30 executions

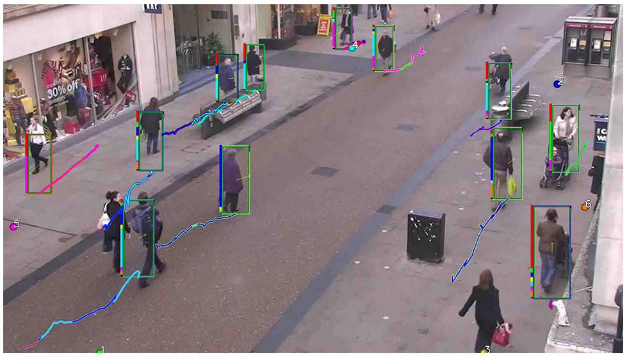


Fig. 11 Tracking on the TownCentre dataset. The ground-level lines depict the target trajectory, where the *color* indicates the most probable goal at each instant

of the same experiment. In all the cases, the variance of the results is smaller than 0.0001.

7.1 Indicators for tracking performance

We evaluate our tracking algorithm with six standard tracking evaluation metrics developed under the US Government Video Analysis and Content Extraction (VACE) program and the CLassification of Events, Activities, and Relationships (CLEAR) consortium [21]:

1. Detection-based metrics:
   Sequence Frame Detection Accuracy (SFDA) is a frame-level measure evaluating the spatial alignment of system output and ground-truth objects. It finds the best match between targets and GT detections at frame-level and averages their overlapping over all the sequence. Small values reflect missed detections and false positives.
   Multiple Object Detection Precision (MODP) measures the target detection rate and precision, through the average overlap per trajectory.
   Multiple Object Detection Accuracy (MODA) evaluates the ratio of missed detections and false positive.
2. Tracking-based metrics:
   Average Tracking Accuracy (ATA) estimates the spatiotemporal overlapping between the output and ground-truth trajectories, finds their best association through the Hungarian algorithm and evaluates their average overlapping. It penalizes short trajectories and indicates how well a tracking system works as what is desired is to keep the identity of the target as long as possible, ideally one tracker per target. It includes the number of detected targets, missed detections and false positives.
   Multiple Object Tracking Precision (MOTP) measures the tracks spatiotemporal precision, through the average overlap per frame.
   Multiple Object Tracking Accuracy (MOTA) is similar to MODA but it takes into account the identity switching of the trackers.

Those six indicators give a tracking quality score between 1 (best) and 0 (worst) and allow to compare proposals. Note that in the literature, there are several metrics to measure the performance of a multi-target tracking system. We chose these indicators since each one is focused to measure the performance of a specific characteristic. For example, ATA favors to have only one trajectory than multiple tracklets, even though the latter covers better the real trajectory. Meanwhile, MOTA considers gives more importance to the detection quality. For more details and the mathematical formulation, please see [16] or [21].

A few more details about our implementation: new trackers are created from the blobs provided by a foreground detector algorithm. Trackers are deleted when the overall likelihood stays under a threshold for a given number of frames. We set the number of particles per goal model at 100, so that $N = G \times 100$.

7.2 Automatic goal detection

Figures 3 and 5 depict the GT trajectories used in the goal learning process for both PETS2009 and TownCentre datasets. The number of goals is automatically computed using the jump method described above, and as illustrated in Fig. 12. We note that some trajectories may have different goals along time, i.e., trajectories switching to goal 7 in Fig. 3. In Fig. 13, we see the distortion curve obtained using points in the image plane as the data points. The optimal value of k is 4 for PETS2009, view 1; it is 8, with multiple sub-goals (1–3), for Town Centre. For PETS09, view 2, we obtained similar results as in view 1. We have set the search range between 1 and 20 as we know the number of goals should be within this range. We have tested the clustering algorithm using key points in both the world and the image plane. In the former case, key points are projected from the image to the world plane by using an approximate homography. The results are very similar both qualitatively (see Fig. 12) and quantitatively (i.e., small variance). For purposes of clarity, we present tracking results using the clustering step in the image plane.

7.3 Evaluation of tracking performance

Figures 1 and 11 show qualitative results on the PETS2009 and TownCentre datasets (respectively) where the bounding boxes depict the tracking results. The left rectangles at each bounding box represent the contribution weight of each model (the $p(g|\mathbf{Z}_{1:t})$ described above), i.e., the dominant color indicates the model that fits best to the dynamic of the target. In those images, we can observe the switches between

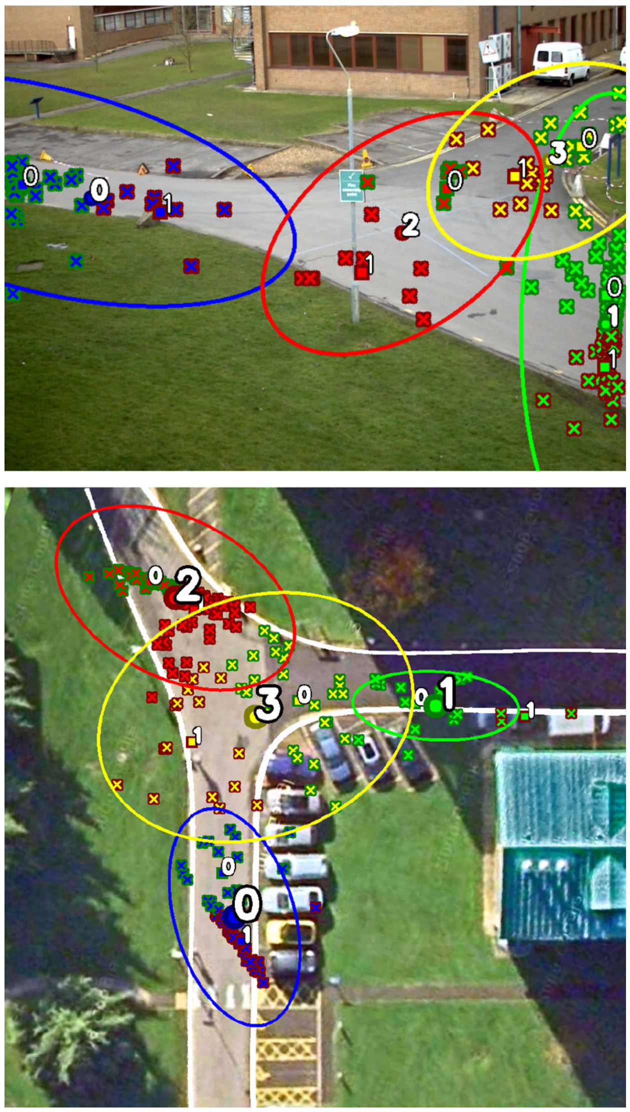


Fig. 12 Goal clustering. *Top*: Results in image plane. *Bottom*: Clustering in world plane. The *circles* depict the centroids computed by the clustering algorithm, the crosses are the key points and the ellipses show the covariance of each cluster. The *color* of each element is the one of the closest centroid

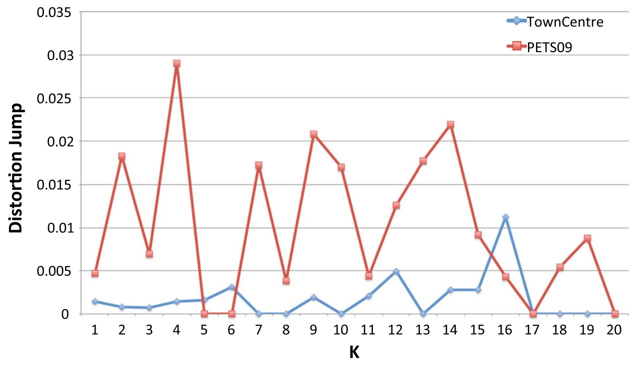


Fig. 13 Distortion curve computed by the jump algorithm using PETS2009 and TownCentre datasets. We use the key points in the image plane

goal models. When the target remains at the same position, the dominant color in the left rectangle is red which means that the *Constant Position* (CP) model is the one who contributes most to the state estimation. When the target moves, the dominant color changes to the model associated to the goal that fits best to the target motion. An example of this is the leftmost person in Fig. 1 who stops before reaching the green goal. However, this goal switching may be incorrect sometimes, since when the target is moving slowly, the tracker belief may be balanced between the CP model and a proper goal model. We can observe this trend with the couple at the upper right of Fig. 1.

We present quantitative results over both datasets with the metrics mentioned above. We have tested three models: a classic constant velocity model (CV), our goal-oriented motion prior proposal alone (IMMPF GMP) and our proposal including the social forces (IMMPF GMP SF). The rest of the implementation (observation model, initialization, termination, etcetera) remains in the same. The goal-specific models and the repulsion forces give an overall improvement in the global performance. Table 1 presents the results for the PETS09, view 2 sequence. This sequence is tested with two goal-oriented based motion prior maps, one with four goals (without sub-goals) and one with eight root-goals, each one with one or two sub-goals for a total of 15 goals. Figure 14 compares our performance (last two diagrams) against other approaches which results were extracted from [16] using view 1 of PETS09. In Fig. 15, we show the results using the TownCentre dataset. One can observe in particular that our approach ATA stands out. This means that our proposal, with the SFs, can track the same target longer than other techniques that fail in preserving the identity of targets with similar appearance. More globally, when considering all the scores, we obtain a good balance between indicators, compared to other approaches.

7.4 Evaluation of prediction performance

We measure the accuracy of our path predictor (see Sect. 6) by considering the missed detections and false positives in the following way: First, we define an area in which we intent to predict whether the target reaches this zone in a given interval of time, i.e., the blue squared zone at the left side of Fig. 10. Then, we check the ground-truth (GT) trajectories and the predicted paths over this area to count the number of missed detections (md_s) and false positives (fp_s) within the window of time $f \in \{5, 10, 15, 20, 25\}$. From this information, we can compute the normalized Multiple Object Detection Accuracy metric (MODA) as:

Table 1 Quantitative results of our proposals with PETS09 dataset view 2

Auth	SFDA	ATA	MODP	MOTP	MODA	MOTA
CV	0.47	0.39	0.79	0.79	0.41	0.40
IMMPF GMP View 2 4G	0.68	0.57	0.91	0.72	0.52	0.50
IMMPF GMP View 2 4G SF	0.72	0.59	0.91	0.75	0.63	0.62
IMMPF GMP View 2 8G	0.67	0.58	0.91	0.66	0.48	0.46
IMMPF GMP View 2 8G SF	0.74	0.64	0.91	0.71	0.65	0.64

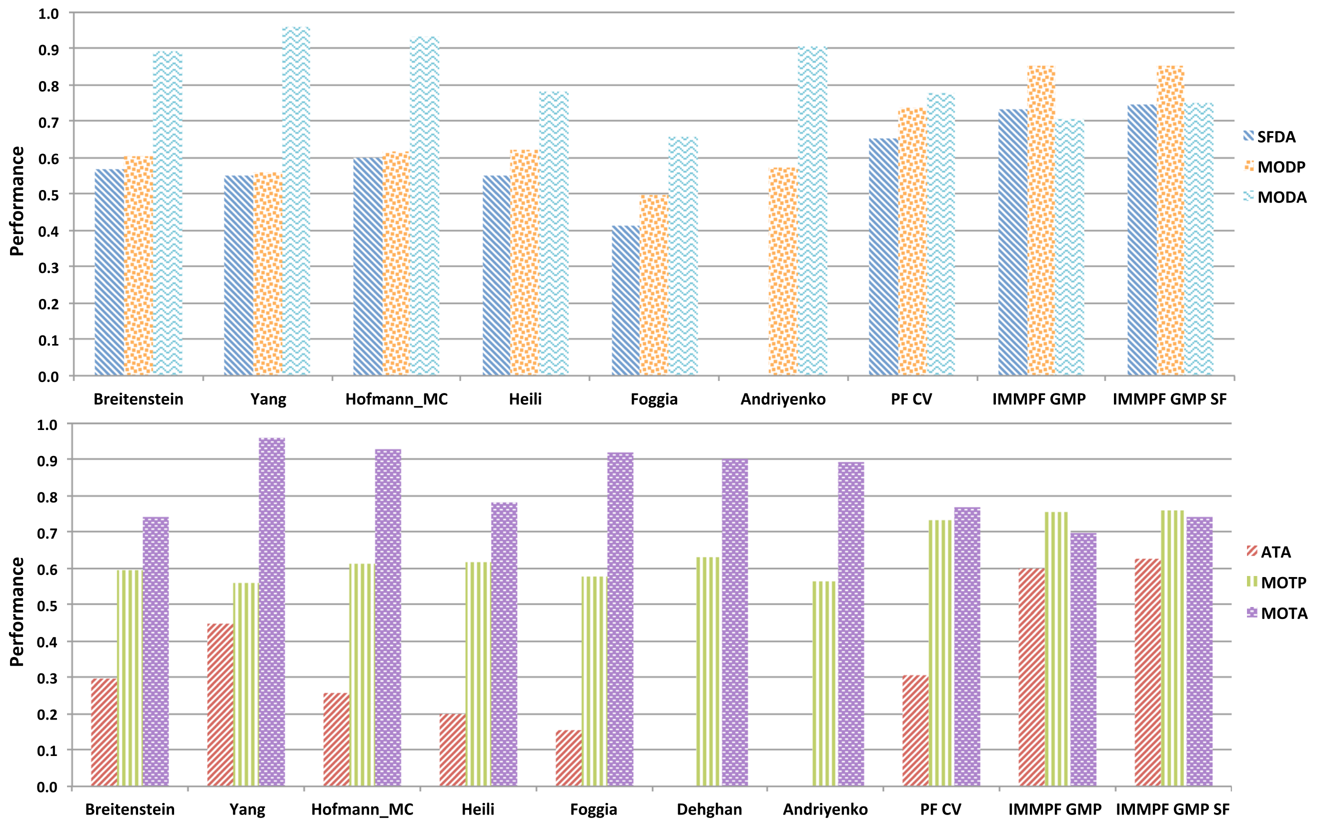
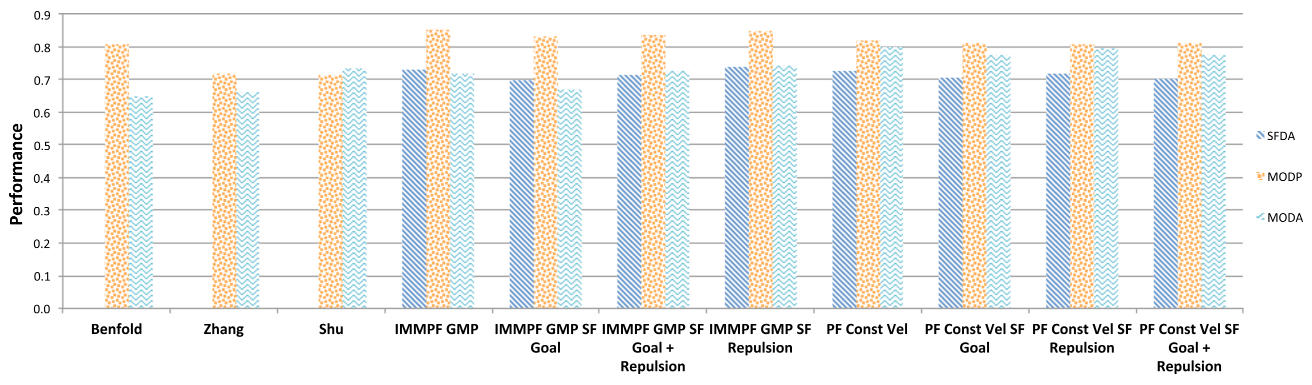


Fig. 14 Tracking performance evaluation in the PETS video sequence: On the *upper part*, we depict detection and precision-related indicators; on the *lower part*, continuity and absence of id switches indicators. The different groups of columns (indicators) represent state-of-the-art algorithms (*left*), a particle filter with a simple linear motion (3rd from *right*) and our proposal (without/with repulsion forces). The results of Breitenstein, Yang, Hofmann, Heili and Foggia were extracted from [16], Dehgan from [14] and Andriyenko from [3]

$$\mathrm{MODA}(f) = 1 - \frac{\sum_{f=1}^{N_{\text{frames}}} md_s + fp_s}{\sum_{f=1}^{N_{\text{frames}}} N_{\text{GT}}^{(f)}}$$

where $N_{\text{GT}}^{(f)}$ is the number of GT trajectories in the region of interest at frame step f and N_{frames} is the total number of frames in the sequence. We use the Hungarian algorithm to make matches between the GT and our proposal output at a given time t. The associated GT trajectory is then compared against our predicted path through the MODA metric. The results are depicted in Table 2. In Fig. 16 we plot the MODA evaluation with the PETS2009 dataset. The red and green lines are the MODA metric resulting from our MPF predicting path. They differ in the use of social forces (SF) included in the tracking system and not in the prediction itself. We compare them with a third path predictor (PP) which consists in a simple constant velocity model. We apply our goal-oriented tracking framework with this predictor in such a way that the results labeled CV PP and MPF PP use the same tracking system but with a different path predictor. We can observe that the results with the motion prior field predictor are very similar regardless the use of the social force model, which moves the target state positions a little bit. Instead, if we compare them with the CV path predictor, we observe that the latter has lower values of MODA. This means that its prediction is less accurate than the first two. In all the cases, without surprise, the predictors accuracy decreases with longer prediction times.



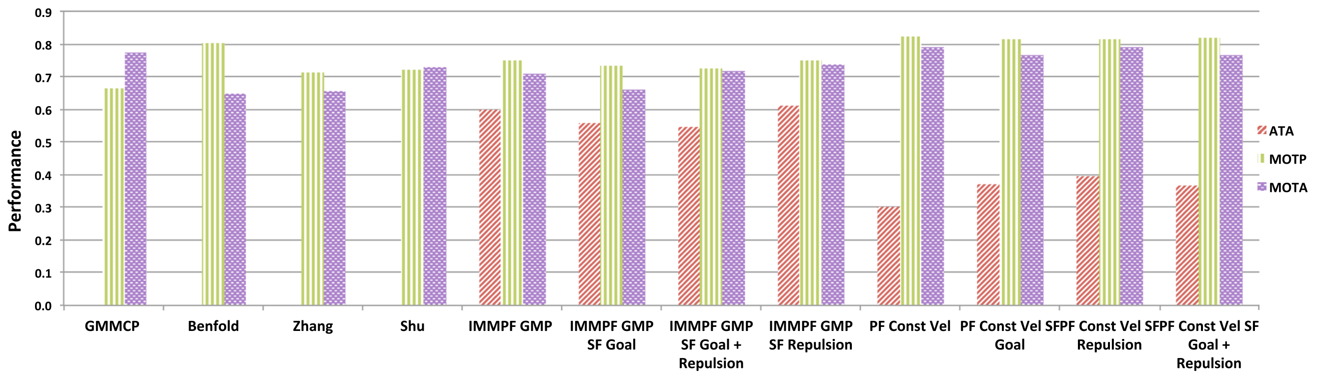


Fig. 15 Tracking performance evaluation in Town Centre video sequence: The different groups of columns (indicators) represent a particle filter with a simple linear motion (*left*) and our proposal considering single-level (GMP) and two-level (HGMP) of hierarchy (without/with social forces). The results of GMMCP were extracted from [12] and Benfold, Zhang and Shu are taken from [24]

Table 2 Path prediction evaluation

Frames	5	10	15	20	25
Model					
CV PP	0.59	0.46	0.34	0.19	0.07
MPF PP	0.63	0.51	0.39	0.28	0.19
MPFSF PP	0.61	0.49	0.38	0.26	0.20

MODA metric evaluated in an area of the PETS2009 sequence, the *blue area* in Fig. 10, left side. GT trajectories are compared against predicted paths for 5 time window horizons ($t = \{5, 10, 15, 20, 25\}$)

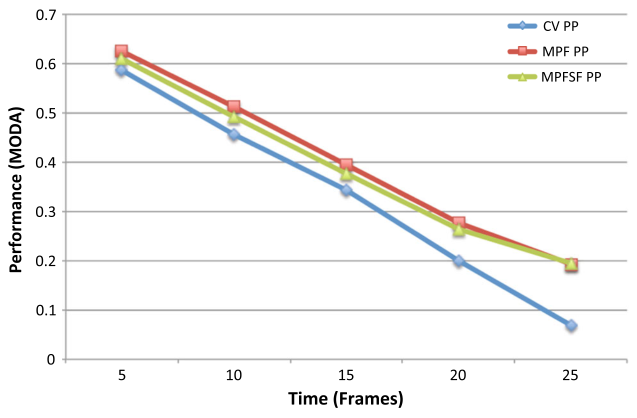


Fig. 16 Path prediction evaluation using the normalized MODA metric on the PETS2009 dataset. We compare GT trajectories against predicted paths in the *blue square* at the *left side* of Fig. 10. We consider 5 time window horizons ($t = \{5, 10, 15, 20, 25\}$)

In another experiment, we have calculated the MODA metric defined above for all predicted paths over the entire sequence. At each evaluation time, the Hungarian algorithm generates the GT/estimated target matches. We consider the distance between the GT and predicted trajectories and we mark it as a missed detection/false positive if the distance is higher than a given threshold (i.e., 20 pixels in image plane). Also, we compute the mean squared errors (MSE) between the GT and prediction and calculate the average over all trajectories and frames. Tables 3 and 4 depict the normalized MODA and the average MSE, respectively, for both PETS2009 and TownCentre datasets. We show in Fig. 17 the results of PETS2009. We observe how, in the course of time, the CV predictor decreases faster compared with the MPF predictors, as seen in Fig. 18.

7.5 Computational times analysis

Finally, we calculate the time that our method takes under the PETS09 and TownCentre (TC) datasets. This evaluation is

Table 3 Path prediction evaluation using MODA metric

Dataset	Model					
	Frames	5	10	15	20	25
PETS2009 view 1	CV PP	0.47	0.32	0.21	0.15	0.11
	MPF PP	0.49	0.38	0.30	0.25	0.21
	MPFSF PP	0.48	0.39	0.31	0.26	0.22
PETS2009 view 2	CV PP	0.47	0.37	0.28	0.22	0.17
	MPF PP 4 Goals	0.45	0.37	0.31	0.26	0.23
	MPF PP 8 Goals	0.48	0.42	0.36	0.31	0.27
	MPFSF PP 4 Goals	0.45	0.38	0.32	0.27	0.24
	MPFSF PP 8 Goals	0.52	0.46	0.39	0.34	0.30
TownCentre	CV PP	0.17	0.12	0.09	0.07	0.05
	MPF PP	0.21	0.20	0.18	0.15	0.13
	MPFSF PP	0.23	0.22	0.19	0.17	0.14

GT trajectories are compared against predicted paths for 5 time window horizons ($t = \{5, 10, 15, 20, 25\}$)

Table 4 Path prediction evaluation: Average of the mean squared error between GT trajectories and predicted paths for all the tracked objects

Dataset	Mod.					
	t	5	10	15	20	25
PETS2009 view 1	CV	18.27	21.16	25.48	30.27	35.17
	MPF	16.52	18.70	21.83	25.22	26.69
	MPF SF	16.88	18.95	21.96	25.30	26.65
PETS2009 view 2	CV PP	15.44	18.53	22.14	26.10	30.31
	MPF PP 4 Goals	14.51	17.16	20.21	23.29	26.69
	MPF PP 8 Goals	13.69	15.34	17.44	19.74	22.04
	MPFSF PP 4 Goals	13.58	16.22	19.16	22.15	25.21
	MPFSF PP 8 Goals	12.10	13.71	15.87	18.17	20.49
TownCentre	CV PP	51.44	52.47	56.68	61.74	67.41
	MPF PP	32.36	32.32	32.75	33.66	34.91
	MPFSF PP	33.03	33.03	33.53	34.34	35.55

We evaluate those for 5 time window horizons

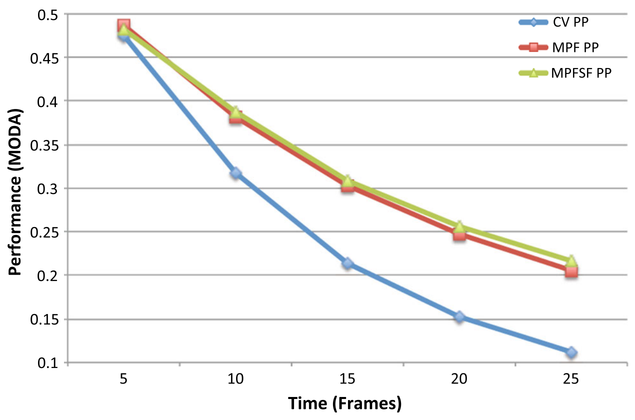


Fig. 17 Path prediction evaluation using normalized MODA metric on the PETS2009 dataset. We compare GT trajectories against predicted paths for 5 time window horizons ($t = \{5, 10, 15, 20, 25\}$)

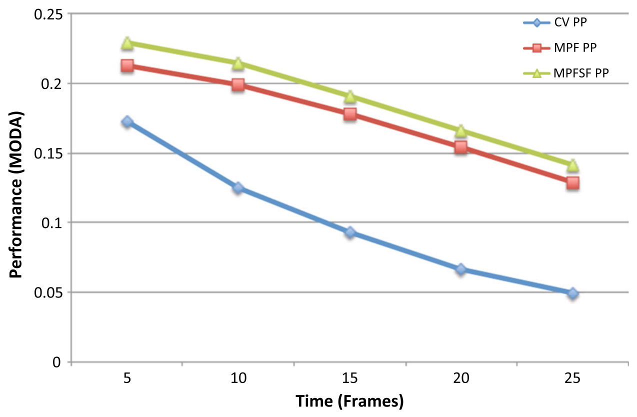


Fig. 18 Path prediction evaluation using normalized MODA metric on the TownCentre dataset. We compare the GT trajectories against predicted paths for 5 time window horizons ($t = \{5, 10, 15, 20, 25\}$)

Table 5 Learning (per frame) and regularization steps performance time comparison.

Time *Model*	Learning	Regularization
PETS09v1	0.08s	0.55s
PETS09v2 1L 4G	0.08s	0.55s
PETS09v2 2L 8G 15S	0.08s	0.55s
TC 1L 16G	0.44s	4.3s
TC 2L 8G 15S	0.45s	4.02s

We calculate the average time which our framework takes in learning and regularization steps. Results are the median of 30 executions

Table 6 Performance time comparison.

Time *Model*	Per frame	Per tracker
PETS2009v1	0.46s	0.08s
PETS2009v1 SF	0.52s	0.09s
PETS2009v2 4G	0.40s	0.07s
PETS2009v2 8G	0.66s	0.10s
PETS2009v2 4G SF	0.45s	0.09s
PETS2009v2 8G SF	0.75s	0.15s
TC 1L 16G	4.79s	0.29s
TC 2L 8G 15S	2.78s	0.18s
TC 1L 16G SF	5.38s	0.34s
TC 2L 8G 15S SF	3.13s	0.22s

We calculate the average time that our tracking system takes to process each frame and each tracker. Results are the median of 30 executions

done with a C++ implementation using a single core of a 2.2 GHz Inter Core i7 processor. Table 5 shows the required time in the learning and regularization steps (Sect. 4.5). We show the obtained time for the PETS09, view 1 dataset (referred to as PETS09v1) and TownCentre dataset in a non-hierarchical way, with 1 level and 16 root-goals (referred to as TC 1L 16G); we also show results on PETS09, view 2 for a hierarchical version with 2 Levels and 15 sub-goals spread in 8 root-goals (referred to as PETS09v2 2L 8G 15S). We can observe how the processing times for Town Centre under the hierarchical version is smaller than the non-hierarchical version, even if the latter must regularize 23 MPFs (see Fig. 4). This is thanks to our hierarchical regularization scheme. Table 6 presents the performance time of the tracking process (Sect. 5). We see how the models that take social forces into consideration are the ones that consume more time. However, they are more accurate. Moreover, by comparing the Town Centre results, we see that hierarchical versions (TC 2L 8G 15S) are computed faster than regular versions (TC 2L 16G). This is because the latter have to evaluate more models at once.

7.6 Discussion on the approach performance

The previous results show that our approach can improve the tracking performance under challenging scenarios. It is interesting to see that our method shows a significant improvement in the ATA score. This is related to the fact that our per goal motion prior field fits better to the complex dynamic of the pedestrians. However, this is not the case in terms of MOTA and MODA. This behavior is because the other methods compared to us use a different (and probably better) pedestrian detector, which has an influence on the number of false positives and missed detections. In our case, we use a simple blob detector which uses the output of a foreground/background (FG/BG) technique to find region with motion. Depending on the quality of the FG/BG algorithm, it may fail in the presence of illumination changes, noise or targets with a similar color as the background. Meanwhile, other proposals use feature-based detectors (i.e., Histogram of oriented gradient) which have higher rate of detection.

Our approach is suitable for structured scenarios where the exit points of the pedestrians, in the field of view of the camera, are well defined, in an environment with bounded areas of motion. This method could be used under unstructured scenes as long as shared common goals exist.

Our proposal is based on the fact that pedestrians follow a path with a goal in mind. This also applies for pedestrians in group (i.e., families, friends, couples, among others), because their individual goals are the same as others in the group. Even when a group member does not know its goal, i.e., following a leader, it will act as if it know. This is due to each pedestrian is influenced by the others in the group, making the individual goal be the same as the goal of the group.

With the proposed model, we can capture the trending of long-term goals and dynamic motions of the pedestrians. However, it supposes that the trajectories do not exhibit frequent abrupt changes, which is the case in the datasets we used. More generally, it is adapted to mid-dense public places scenarios, such as shopping malls. In very highly dynamic scenarios with abrupt changes in trajectories, we could use more goals/sub-goals, but it would nevertheless lead to large variations between trajectories sharing the same goal.

Another limitation of the method is that it requires training data from ground-truth trajectories. This limits the practicality of the proposal, but the aquisition of a few ground-truth trajectories could be done either manually in a calibration phase or with a standard tracker with very confidence level parameters, so that no necessarily all the pedestrians could be tracked, but the tracked ones would be very safe trajectories.

Also, our system is built for static cameras only. If the camera moves, the prior model will not match with the current image. However, if we know the calibration parameters and the motion of the camera (e.g., with a PTZ camera), we could adjust the goal-oriented model automatically with a corresponding homography, but this is out of the scope of this paper.

7.7 Overview of critical parameters

As commented in Sect. 4.1, we add as key points any position where a trajectory has a strong change of orientation, superior to 45 degrees. By changing this threshold, more or less key points may appear, resulting in either the detection of many irrelevant or spurious goals or the loss of a real goal.

After the clustering process, we remove the clusters with a number of keypoints lower than 10% of the average number of keypoints per cluster. Thus, we do not estimate motion prior fields for these clusters.

Hierarchies of goals reduce the computational cost by fusing multiple goals when the target is far enough from these goals. The choice of the number of levels depends on the number of goals. One level is fine for few widely separated goals, i.e., for PETS2009 dataset. More levels are needed when the scene complexity and the number of goals increases, i.e., TownCentre dataset.

The MPF regularization fills incomplete information and smooths transitions between cells. It is controlled by two variables $\lambda_1 = 0.25$ and $\lambda_2 = 0.25$. If we increase λ_1, data are shared among neighbor cells and missing information due to lack of trajectories may be inferred. By increasing λ_2 the cell histogram is smoothed.

During tracking, resampling consists in two independent operations: (a) resampling over all particles, every four frames; and (b) resampling on a per model basis, every five frames. The former guides the particles to high values of the posterior. The sampling frequency depends of how fast the particles degenerate. We use a fixed value for practicality. However, there are several criteria in the literature that can be used. The latter controls the model transition. In general, the more complex the targets motion will be (i.e., pedestrian in a mall), the higher the resampling frequency should be.

The proposed framework does not require an exhaustive tuning of the parameters. We have determined most of them based on experimenting individually the different sub-modules of the system. However, the use of inadequate parameters could decrease the system performance, i.e., the reduction in number of key points could results in the losing relevant sub-goals, or could waste computational time, i.e., applying more resampling steps than needed, or produce undesired outputs, i.e., an over-regularized prior map.

8 Conclusions

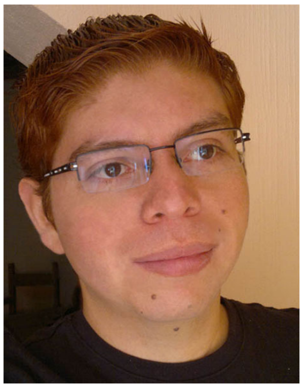
We have presented an IMM-based methodology to perform both target tracking and goal estimation, based on a competition among goal-related motion models, and with an implementation with particle filters. We have shown on two standard datasets that the tracking can be improved on several indicators, and moreover that one can estimate online the intentionality of the pedestrian (i.e., his current goal). As ongoing and future works, we plan to develop motion strategies for mobile agents based on the output from this visual tracking system; also, we would like to adapt our framework to be able to learn continuously the motion priors and the structure of goals. Another objective is to integrate our framework within tracking-by-detection methods, in particular online versions of these methods, such as [20].

Acknowledgements The authors would like to acknowledge Intel and Conacyt for partially funding this research.

References

1. Alahi, A., Ramanathan, V., Fei-Fei, L.: Socially-aware large-scale crowd forecasting. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
2. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: Proceedings of the European Conference on Computer Vision, pp. 1–14 (2008)
3. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1926–1933 (2012)
4. Arechavaleta, G., Laumond, J.P., Hicheur, H., Berthoz, A.: An optimality principle governing human walking trajectories. IEEE Trans. Robot. **24**, 5–14 (2008)
5. Bascetta, L., Ferretti, G., Rocco, P., Ardo, H., Bruyninckx, H., Demeester, E., Di Lello, E.: Towards safe human–robot interaction in robotic cells: an approach based on visual tracking and intention estimation. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 2971–2978 (2011)
6. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3457–3464 (2011)
7. Bennewitz, M., Burgard, W., Cielniak, G., Thrun, S.: Learning motion patterns of people for compliant robot motion. Int. J. Robot. Res. **24**, 31–48 (2005)
8. Berclaz, J., Fleuret, F., Fua, P.: Multi-camera tracking and atypical motion detection with behavioral maps. In: Proceedings of the European Conference on Computer Vision, pp. 112–125 (2008)
9. Bouguet, J.: Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. In: Technical Report Intel Corporation Microprocessor Research Labs (1999)
10. Bruce, A., Gordon, G.: Better motion prediction for people-tracking. In: Proceedings of the International Conference on Robotics and Automation (2004)
11. Chen, C., Heili, A., Odobez, J.: Combined estimation of location and body pose in surveillance video. In: Proceedings of the IEEE Interenational Conference on Advanced Video and Signal-Based Surveillance, pp. 5–10 (2011)
12. Dehghan, A., Assari, S.M.: Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In: Proceedings of the IEEE Interenational Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4091–4099 (2015)

13. Dehghan, A., Kalayeh, M.M.: Understanding crowd collectivity: a meta-tracking approach. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1–9 (2015)
14. Dehghan, A., Tian, Y., Torr, P.H.S., Shah, M.: Target identity-aware network flow for online multiple target tracking. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1146–1154 (2015)
15. Ellis, A., Ferryman, J.: PETS2010 and PETS2009 evaluation of results using individual ground truth single views. In: Proceedings of the IEEE Intereational Conference on Advanced Video and Signal-Based Surveillance, pp. 135–142 (2010)
16. Ferryman, J., Ellis, A.L.: Performance evaluation of crowd image analysis using the PETS2009 dataset. Pattern Recognit. Lett. **44**, 3–15 (2014)
17. Hicheur, H., Pham, Q.C., Arechavaleta, G., Laumond, J.P., Berthoz, A.: The formation of trajectories during goal-oriented locomotion in humans. Eur. J. Neurosci. **26**, 2376–2390 (2007)
18. Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., Kanda, T.: Modeling and prediction of pedestrian behavior based on the sub-goal concept. In: Proceedings of the International Conference on Robotics: Science and Systems (2012)
19. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Inc, Upper Saddle River (1988)
20. Jiang, H., Wang, J., Gong, Y., Rong, N., Chai, Z., Zheng, N.: Online multi-target tracking with unified handling of complex scenarios. IEEE Trans. Image Process. **24**(11), 3464–3477 (2015)
21. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 319–336 (2009)
22. Leal-Taixe, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
23. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops (2011)
24. Lee, D., Hwang, I., Oh, S.: Optimus:online persistent tracking and identification of many users for smart spaces. Mach. Vis. Appl. **25**(4), 901–917 (2014)
25. Luber, M., Stork, J.A., Tipaldi, G.D., Arras, K.O.: People tracking with human motion predictions from social forces. In: Proceedings of the International Conference on Robotics and Automation, pp. 464–469 (2010)
26. Luber, M., Tipaldi, G.D., Arras, K.O.: Place-dependent people tracking. Int. J. Robot. Res. **30**(3), 0278364910393538 (2011)
27. Madrigal, F., Hayet, J.B.: Evaluation of multiple motion models for multiple pedestrian visual tracking. In: Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (2013)
28. Madrigal, F., Hayet, J.B.: Goal-oriented visual tracking of pedestrians with motion priors in semi-crowded scenes. In: Proceedings of the IEEE International Conference on Robotics and Automation (2015)
29. Madrigal, F., Rivera, M., Hayet, J.: Learning and regularizing motion models for enhancing particle filter-based target tracking. In: Proceedings of Pacific-Rim Symposium Image and Video Technology, vol. 2, pp. 287–298 (2011)
30. Milan, A., Schindler, K., Roth, S.: Challenges of ground truth evaluation of multi-target tracking. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
31. Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proc. IEEE **92**(3), 495–513 (2004)
32. Sugar, C.A., James, G.M.: Finding the number of clusters in a dataset: an information theoretic approach. J. Am. Stat. Assoc. **98**(463), 750–763 (2003)
33. Vasquez, D., Fraichard, T., Aycard, O., Laugier, C.: Intentional motion on-line learning and prediction. Mach. Vis. Appl. **19**(5–6), 411–425 (2008)



Francisco Madrigal received the B.E. degree from the Zamora Institute of Technology, Mexico, in 2008 and got his M. Sc. degree in 2010 and his D.Sc. degree in 2015, both in Computer Science from the Center for Research in Mathematics (CIMAT), Guanajuato, Mexico, under the supervision of Prof. Jean-Bernard Hayet. His research is centered on the mathematical modeling of the pedestrian motion to improve the performance of a visual multi-target tracking system in surveillance video sequences. From 2016 to 2018, he does a postdoctoral stay in visual head-pose detection at the Laboratoire d'Analyse et d'Architecture des Systèmes, France. His interests include computer vision, probabilistic robotics, image processing and machine learning.

Jean-Bernard Hayet graduated from Ecole Nationale Superieure de Techniques Avancees in 1999. He got his master degree from University Paris VI and his Ph.D. degree from University of Toulouse in 2003. His doctoral dissertation tackled the problem of landmark-based navigation with computer vision in indoor environments and was supervised by Dr. M. Devy and Dr. F. Lerasle. From 2003 to 2006, he did a postdoctoral stay in visual tracking at University of Liege with Pr. Justus Piater. Since 2007, he is researcher at the Centro de Investigacion en Matematicas, in Mexico, and is focusing his research on probabilistic methods in visual tracking and in handling vision constraints in motion planning and locomotion problems.