Adaptive Interacting Multiple Models applied on pedestrian tracking in car parks

Julien Burlet, Olivier Aycard[1], Anne Spalanzani[2] and Christian Laugier
Inria Rhône-Alpes & Gravir-Imag Lab.,
Grenoble, France
Email: {firstname.name}@inrialpes.fr

***Abstract*— To address perception problems we must be able to track dynamics targets of the environment. An important issue of tracking is filtering problem in which estimates of the target's state are computed while observations are progressively received.**

This paper presents an adaptive Interacting Multiple Models (IMM) based filtering method. Interacting Multiple Models have been successfully applied to many applications as they allow, using several filters in parallel, to deal with the uncertainty on motion model, a critical component of filtering. Indeed targets can rapidly change their motion over a lapse of time. This is the case of pedestrians for which it is difficult to define an unique motion model which matches all their possible displacements.

Nevertheless, the Transition Probability Matrix (TPM) which models the interaction between different filters in an IMM is in currently defined *a priori* or needs an important amount of tuning to be used efficiently.

In this paper, we put forward a method which automatically adapts online the TPM. The TPM adaptation using on-line data significantly improves the effectiveness of IMM filtering and so better target estimates are obtained. To validate our work we applied our method to pedestrian tracking in car parks on a real platform.

I. Introduction

To address perception problems we must be able to track dynamics targets of the environment. An important issue of tracking is filtering problem in which estimates of the target's state are computed while observations are progressively received.

In this paper, we address the filtering problem of a highly maneuvering target (i.e a target which could have different motions in a short lapse of time). Classically, filtering methods aim to compute estimations of the target's state from measurement data.

The most general algorithm for calculating such estimations is the *Bayesian filter* algorithm [1]. This recursive algorithm updates the state estimation through time by processing two essential steps. In the first step (the prediction step), the current state estimation is updated according to a specific target's motion model and the previous computed estimate. This step confers a prediction of the target's state at the current time. In the second step (measurement update step), the prediction and the observation are used to compute the current estimation of the target's state. This recursive algorithm relies on Markov's assumption, computing at one given time a state estimation using the estimation computed at the previous time.

Nevertheless, straight implementation of Bayesian filter can not be applied to realistic problems. Indeed, the prediction step requires an integration over state space which is intractable in practice without restricting ourselves to finite state spaces or without using assumptions. So *Bayes filters* are implemented in several different ways. There exists a variety of techniques and algorithms, all derived from the Bayes filter, relying on different assumptions.

The most classical implementation is the well known *Kalman filter* [2] in which estimations and observations are supposed Gaussian. This assumption allows to define such a filter as a set of linear equations. Furthermore, nonparametric methods such as particle filters [3] have become as popular because of their computational effectiveness and as they do not assume a functional form of the estimate.

Moreover, the motion model is the main part of the prediction step of all kinds of filters. However, in the presence of uncertainties on target motion, defining a suitable motion model is a real difficulty. Indeed, under real world conditions, the target can have very different displacement modes and it is therefore quite impossible to define an unique motion model which can match all different motions a highly maneuverable target could execute. Thus it is necessary to cope with motion uncertainties in such a case.

To deal with these motion uncertainties, Interacting Multiple Models (IMM) [4] [5] have been successfully applied in several applications [6] [7] [8]. The IMM approach overcomes the difficulty due to motion uncertainty by using more than one motion model. The principle is to assume a set of models as possible candidates of the true displacement mode of the target at one time. To do so, a bank of elemental filters is ran at each time, each corresponding to a specific motion model, and the final state estimation is obtained by merging the results of all elemental filters. Also, the probability the target changes of displacement mode is encoded in a transition probability matrix(TPM), *i.e* the transition between modes which is assumed Markovian.

Nevertheless, to apply IMM on a real application a number of critical parameters have to be defined for instance the set of motion models and the transition probability matrix(TPM). In practice, the TPM is often assumed known and is chosen *a priori*. Even if the design of TPM for different applications

[1]Associate Professor at Joseph Fourier University, Grenoble(FR).
[2]Associate Professor at Pierre-Mendès France University, Grenoble(FR).

have been studied [9] [10], its definition and construction do not rely on the real on-line data and so such TPMs can not be adapted to the real application. therefore it is an important issue to automatically adapt the TPM to fit the application of the IMM algorithms.

Few publications address this specific problem and in most of them, simplest problems are considered (binary system case) [11] or the TPM is assumed to belong to a set of finite candidates TPMs [12]. Also, in [13] modes transition chain is formalized as a bayesian network and a maximum likelihood estimator of the TPM is proposed. However, papers of V. P. Jilkov [14] [15] gives algorithms to adapt on-line the TPM under the assumption that the unknown TPM is random but time-invariant. In all this works, assumption on estimated TPM is relatively strong or algorithm complexity is too hight to address real time applications.

In this paper, we are interested in this TPM adaptation problem. In particular we focus on defining an adaptive IMM filter suitable to the problem of tracking pedestrians in a car park environment. We define an on-line method to automatically adapt the TPM of an IMM according to the real observation data. The use of IMM filters is directly induced by the uncertainty on the pedestrian behavior. Indeed, a pedestrian could have various motions and could suddenly change their current motion. To add to this, a car park's configuration is not static and pedestrian trajectories vary according to this configuration. Thus, the TPM has to be continuously updated to allow an effective filtering by the IMM.

We have developed a fast method which adapts on-line the TPM according to pedestrian trajectories and so we obtain a suitable and robust filtering. Moreover, the effectiveness of our method has been validated on a real car park environment.

This paper is organized as follows. In the next section, we present Interacting Multiple Models filters. The third section is dedicated to the presentation of the method defined to automatically adapt the transition probability matrix. Application of our method on a real platform and results are the topics of the fourth section. And finally conclusion and perspectives are given in the fifth section.

II. Interacting Multiple Models (IMM)

The basic idea of IMM is to simultaneously use several filters and mix their outputs to obtain a better estimation. This method allows to cope with the uncertainty on the target motion by running a set of possible displacement modes at the same time. Even if the target is supposed to possibly be in each displacement mode, the probability that it is in each of them is considered and updated during execution of the IMM.

The TPM, as it models the transition between modes, plays a major role in the update of the modes probabilities. Let M the number of modes, we note μ_t the variable over modes at time t, so $\mu \in [1..M]$. Thus, $P([\mu_t = i])$ gives the probability the target is in mode i at time t. For convenient, we note μ_t^i this probability. Using this notation the TPM gives the probability $P(\ \mu_t|\ \mu_{t-1})$ of transition from modes μ_{t-1} at time $t-1$ to modes μ_t at time t.

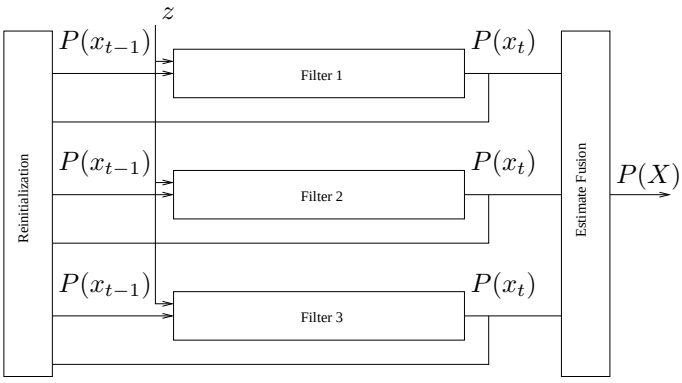

Fig. 1. Principle of IMM

One cycle of an IMM is composed of tree steps (Fig. 1): A step in which filter execution is done and μ_t is updated, a fusion step allowing to compute estimate fusion and a reinitialization step.

Filters execution and update of μ_t: In a first step, a new observation z comes as input and all filters are ran independently to obtain estimations $P(x_t)$. Also variable μ_t is updated according to the likelihood of observation with filter internal prediction.

The decomposition of the joint distribution used to model a filter is :

$$P(x_{t-1}\ x_t\ z_t\ a_{t-1}) = P(x_{t-1})P(z_t|\ x_t)P(x_t|\ x_{t-1}\ a_{t-1}) \tag{1}$$

with x_{t-1} the state variable at the input of the filter, x_t the state variable at the output of the filter, z_t the current observation received and a_{t-1} the action the target is supposed to execute.

The first distribution $P(x_{t-1})$ is the state repartition computed at the last time or defined *a priori* at the first run. The second distribution $P(z_t|x_t)$ is called the sensor model and gives the probability of having the observation z_t knowing the current state x_t. The last one is the prediction model $P(x_t|x_{t-1}\ a_{t-1})$ with gives the predicted current state knowing the previous state and the action done. This last distribution defines a specific motion model associated to the filter.

Using a filter we want to estimate the current state of the target knowing z_t and the action done. So the classical inference is on $P(x_t|z_t)$. Using marginalization and Bayes rules we obtain [1] :

$$\begin{aligned} P(x_t|z_t\ a_{t-1}) &= \frac{1}{\alpha} \sum_{x_{t-1}} P(x_t\ x_{t-1}\ z_t\ a_{t-1}) \\ &= \frac{1}{\alpha}\ P(z_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}\ a_{t-1})P(x_{t-1}) \end{aligned} \tag{2}$$

[1]Note that we obtain the same classical equations of the Bayesian filter as it is defined in [1] for instance

Compute estimate fusion $P(X)$: In a second step, a final estimate is obtained by mixing all filter outputs according to computed μ_t. Supposing we have M different modes, the decomposition of the joint distribution used is:

$$P(x_{t-1}^{1:M}\ X_t\ z_t\ \mu_{t-1}\ \mu_t) = P(\mu_{t-1})\ P(x_{t-1}^{1:M})\ P(\ \mu_t|\ \mu_{t-1})\ P(z_t|\ \mu_t) P(X_t|x_{t-1}^{1:M}\ z_t\ \mu_t) \quad (3)$$

Where x_{t-1}^i is the state variable at the input of the mode i, X_t is the state variable at the output of the IMM, z_t is the current observation, μ_{t-1} and μ_t are the mode variables respectively at time $t-1$ and t.

The first distributions $P(\mu_{t-1})$ and $P(x_{t-1}^{1:M})$ are first given *a priori* and are then computed during the process. The distribution $P(\mu_t|\mu_{t-1})$ corresponds to the TPM, it gives the transition probability between modes and so is defined as a matrix. The next distribution $P(z_t|\mu_t)$ gives the likelihood of the observation according to the filter prediction. More precisely, for a given value of $\mu_t = i$ we obtain $P(z_t|[\mu_t = i])$ using a function (defined *a priori*) computing the likelihood between observation and prediction. The last one $P(X_t|x_{t-1}^{1:M} z_t \mu_t)$ is obtained by the same way through filter programs : for a given value of μ_t we have $P(X_t|x_{t-1}^{1:M} z_t[\mu_t = i]) = P(x_t|x_{t-1}^i z_t)$. The semantic of this last distribution can be illustrated by the following case : if we know with certainty that the target is in a given mode j at time t, that is we have $P([\mu_t = j]) = 1$ and $P([\mu_t = i]) = 0\ \forall i \neq j$, thus the estimate fusion is only given by the j^{th} filter.

We want to obtain the estimate fusion according to the filter inputs, the observation and the mode probabilities, so the infered distribution is :

$$P(X_t|\ z_t\ \mu_{t-1}) = \frac{1}{\alpha} \sum_{\mu_t, \mu t-1, x_{t-1}^{1:M}} [\ P(\mu_{t-1})\ P(\ \mu_t|\ \mu_{t-1})\ P(z_t|\ \mu_t) P(X_t|x_{t-1}^{1:M}\ z_t\ \mu_t)\ ] \quad (4)$$

Reinitialization of filters: In a last step, each filter is reinitialized[2] according to all previous estimates, previous μ_{t-1} and the TPM. In particular, the TPM is used to obtain the new μ_t. Updating μ_t allows to obtain the weight of all previously computed estimates for a given filter.

The decomposition of the joint distribution we use for this program is the following:

$$P(x\ \mu_{t-1}\ \mu_t) = P(\mu_{t-1})\ P(\mu_t|\ \mu_{t-1})\ P(x|\ \mu_{t-1}) \quad (5)$$

Where x is the state variable of the target, μ_t is the mode probability at time t and μ_{t-1} is the mode probability at the previous time $t-1$. The first distribution $P(\mu_{t-1})$ is a computed distribution obtained in the estimation program when inference 4 is computed. The distribution $P(\mu_t|\mu_{t-1})$ corresponds to the TPM. And the last distribution $P(x|\mu_{t-1})$ is obtained as follows : for a given value of μ_{t-1} we have $P(x|[\mu_{t-1} = i]) = P(x_{t-1}^i)$ the previous outputs of the filter i (*i.e* $P(x_t^i)$ computed at the previous step).

Using this we compute the reinitialization of each filter, the inferred distribution is :

$$P(x|\mu_t) = \frac{1}{\alpha} \sum_{\mu_{t-1}} P(\mu_t|\ \mu_{t-1})\ P(x|\ \mu_{t-1}) \quad (6)$$

This inference allows to compute the input of all filters : a given filter i having its initialization $P(x_{t-1}^i)$ set to the value $P(x|[\mu_t = i])$.

III. Adaptive IMM

In this section, we present the method used to automatically adapt the TPM. In a first part, we explain how we define the on-line adaption of the TPM. In a second part, we detail how the re-estimation of the TPM is done and the algorithm we have developed.

A. Principle

Figure 2 illustrates the principle of our method. In this figure programs are in dark color while data is in light.

To adapt the TPM in our specific situation *i.e* tracking pedestrians in a car park, trajectories of pedestrians are considered. Indeed, observations are taken into account by set, each set corresponding to a specific pedestrian's trajectory. The first observation of a trajectory is the first time the pedestrian is observed in the environment and the last one is the last observation before the pedestrain leaves the environment. For instance in figure 2, $\{z_0 z_1 ... z_k\}$ is the first set of received observations corresponding to the first trajectory and $\{z_{k+1} ... z_{k'}\}$ is the second one corresponding to the second trajectory.

While pedestrians are tracked by the IMM receiving each observation z_t, each mode probability μ_t is computed and stored by trajectory. Thus for the i^{th} trajectory, we obtain S_i the corresponding sequence of mode probability. For instance, for the first trajectory, we store the sequence S_1 composed of $\{\ \mu_0 \mu_1 ... \mu_k\}$ computed by IMM using observations $\{z_0 z_1 ... z_k\}$. When data is collected for a given number of trajectories (n in the figure 2), the TPM is adapted using mode probabilities and is reused in the IMM for the next estimations. In this way an on-line adaptation of the TPM is obtained.

B. Re-estimation of the TPM

Algorithm 1, given in pseudo-code, is the algorithm defined to compute one adaptation of the TPM. In our adaptive method, we aim to adapt the TPM using pedestrians' trajectories. More precisely, an adaptation of the TPM is done after a given number N of trajectories, to update TPM using a window on trajectories (*cf.* loop line 3-19 of algorithm 1). Moreover trajectories are processed one by one in three steps:

1) Mode probabilities are collected *via* the execution of the IMM
2) Most probable modes' sequence is computed
3) Most probable mode transitions are quantified

[2]For the first cycle, all filters are initialized with arbitrary values

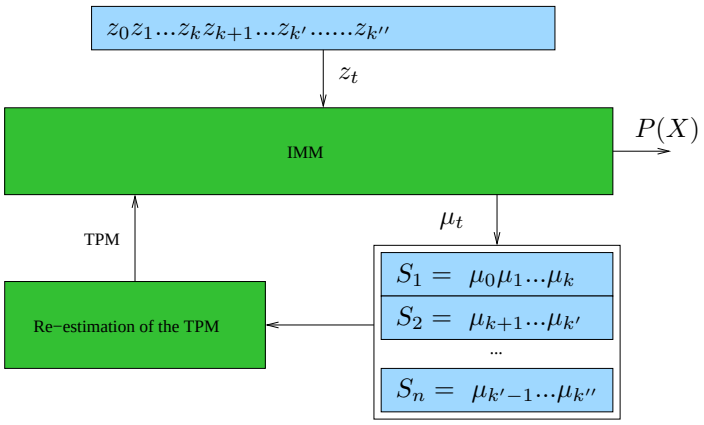

Fig. 2. Principle of our adaptive program

Algorithm 1 Adaptive IMM Algorithm

1: **Adaptation_of_TPM(**$z_0, ..., z_{k''}$**)**
2: $n \leftarrow 0$
3: **repeat**
4: $S_n \leftarrow [\,]$
5: /* Obtain μ_k,...$\mu_{k'}$ from IMM excution */
6: **for all** $Observations\ z_k\ in\ T_n$ **do**
7: $\{\mu_k,\ P(X)\} \leftarrow IMM(z_k)$
8: $S_n \leftarrow S_n \cup [\mu_k]$
9: **end for**
10: /* Compute the most probable mode sequence MPS */
11: $MPS \leftarrow Viterby(S_n)$
12: /* Quantification of mode transitions */
13: **for all** $Couple\ (\ MPS_k,\ MPS_{k+1})\ in\ MPS$ **do**
14: $i \leftarrow MPS_k$
15: $j \leftarrow MPS_{k+1}$
16: $F_{ij} = F_{ij} + 1$
17: **end for**
18: $n \leftarrow n + 1$
19: **until** $n = N$
20: /* Update of TPM in IMM */
21: $TPM \leftarrow Normalization(F)$
22: Return *TPM* in IMM

Collection of mode probabilities: For each observation of a given trajectory, IMM is ran and estimates and mode probabilities are computed (lines 7). Mode probabilities' sequence S_n obtained in such a way is stored to be processed (line 8).

Computation of the most probable mode sequence: In a next step, the most probable modes' sequence of S_n is computed (line 11). More precisely, considering the actual TPM and a set $S_n = \mu_0 ... \mu_K$ of mode probabilities through time 0 to K, we aim to obtain the most probable modes' sequence knowing the estimates computed by the IMM:

$$Max\ P(\mu_0\ \mu_1 ... \mu_k \mid x_0\ x_1 ...\ x_K) \tag{7}$$

So by extension of the decomposition (5) taking into account temporal dimension, we obtain a new decomposition:

$$P(x\ \mu_0\ \mu_1 ...\ \mu_K) = P(\mu_0) \prod_{k=1}^{K} P(\mu_k |\ \mu_{k-1})\ P(x |\ \mu_{k-1}) \tag{8}$$

Where x is the state variable of the target and μ_k is the mode probability at time k. All distributions remain the same as defined in decomposition (5).

Using this decomposition, the inferred distribution is distribution (7). Also, as we just need to obtain the maximum of the distribution $P(\mu_1\ \mu_2 ... \mu_K \mid x_0\ x_1 ...\ x_K)$, the inference is made using the Viterbi Data Algorithm [16]. As complexity of this algorithm is in $O(KM^2)$, we efficiently obtain the most probable modes' sequence.

Quantification of most probable mode transitions: Using this most probable modes' sequence, the number of transitions from one mode to an other is quantified (lines 13 to 17). To do so a frequencies matrix is considered. This matrix models the number of transitions which have occurred from one mode to an other. We note F this matrix and so F_{ij} gives the number of transitions which has occurred from mode i to j. Using the most probable modes' sequence corresponding to a specific trajectory and computed by the Viterby algorithm, the update of F is directly obtained by counting transitions in this sequence. Furthermore, F is kept in memory to be used in next adaptation and before the first update all its elements are set to 1.

Finally, when N trajectories have been treated, the new TPM is obtained by normalization of the frequencies matrix F. Thus the TPM is re-estimated using all mode sequences $S_1 ... S_N$ and is reused in the IMM for next executions (lines 21 and 22). In practice, before the first run, the TPM is chosen uniform (according to F initialization) as we do not want to introduce *a priori* data.

In the next section we will see how this algorithm is applied to a real application.

IV. Application and Results

In this section, we first present the experimental platform used to validate our work on the pedestrian tracking problem. In a second part, the IMM defined for our application is presented. In a last part, experimental results are given and commented.

A. Experimental platform

The experimental setup used to evaluate our method is an evolution of the ParkView platform [17], initially developped for a French national project designed for the Interpretation of Complex Dynamic Scenes and Reactive Motion Planning.

The ParkView platform is composed of a set of six off-board analog cameras, installed in a car-park setup such that their field-of-views partially overlap , and three Linux(tm) workstations in charge of data processing, connected by a standard Local Area Network.

The workstations run a specifically developped client-server software composed of three main parts, called the *map server*, the *map clients* and the *connectors* (figure 3).

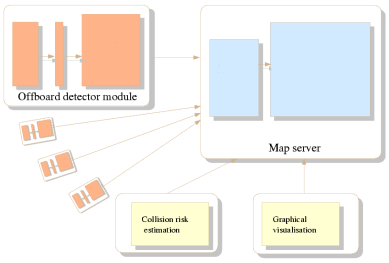

Fig. 3. The ParkView platform software organization

The map server: processes all incoming observations, provided by the different clients, in order to maintain a global high-level representation of the environment; this is where data fusion occurs. A single instance of the server is running.

The connectors: receive the raw sensor-data, perform the pre-processing, and send the resulting *observations* to the map server. Each computer connected with one or several sensors run such *connectors*. For the application described here, all data preprocessing basically consists in pedestrians detection. Therefore, the video stream of each camera is processed independantly by a dedicated detector. The role of these detectors is to convert each incoming video frame to a set of bounding rectangles, one for each target detected in the image plane. The set of rectangles detected at a given time constitutes the detector observation, and is sent to the map server.

The map clients: connect to the server and provides users with a graphical representation of the environment; they can also process this data further and perform application-dependant tasks. For instance, in a driving assistance application, the on-board vehicle computer runs such a specialized client to estimate the collision risk.

B. IMM definition

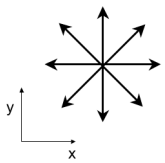

Fig. 4. The eight chosen motion models in the car park's frame

The first step to apply our method to our platform is to define an appropriated IMM and, in particular, modes which compose it.

In this specific application, pedestrians can move in any directions and can often change their motion. Thus in our aim we choose various IMM's modes to model the set of possible directions. As each mode corresponds to a specific motion model, we have to define each motion model. Assuming speed is relatively constant and fixing eight directions in the set of possible directions a pedestrian can follow, we obtain eight motion models (fig. 4). Each of them models a motion in a specific direction with a fixed velocity of one meter per second.

Hence, according to the definition of these eight motion models, our IMM is composed of eight modes. Kalman filters are chosen to implement modes as they allow fast computation.

We must usually also define the TPM. As we develop a method which computes the TPM online, we do not need specific informations concerning the TPM and no modeling are needed. So the TPM is initially chosen to be uniform. As eight modes are defined, the TPM is an uniform square 8×8 matrix.

C. Experimental Results

To validate our method, experiments on the ParkView platform have been carried out. In these experiments, a pedestrian moving in the car park is tracked. This pedestrian enters and exits the car park several times describing several trajectories. The pedestrian describes in this way approximatively hundred trajectories to meet the needs of our experiment.

Pedestrian positions are computed using a detector (*cf* section IV-A) and are used as observations by our program. Using these observations, our adaptive method is used to compute estimates and the TPM of the IMM is updated for each ten trajectories. Estimates are displayed *via* a map client.

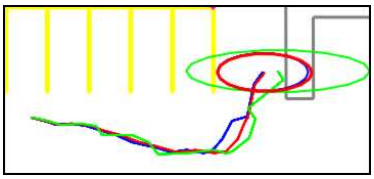

Fig. 5. Tracking result after 20 trajectories (2 online re-adaptation of the TPM)

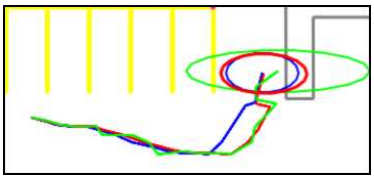

Fig. 6. Tracking result after 50 trajectories (5 online re-adaptation of the TPM)

To illustrate the effectiveness of our method, traces of tracking with and without adaptation of the TPM are showed in figures 5 and 6. In these figures, the green (lightest) line corresponds to the trajectory composed by observations, the blue(darkest) line is the trajectory described by estimates computed without adaptation of the TPM and red line corresponds to the trajectory obtained with estimates computed using our method. The ellipses at the end of the trajectories give indications on the size of uncertainty on the final position and thus the estimates' shape.

So these figures illustrate two tracking of the pedestrian at different time. In both cases the pedestrian preforms relatively

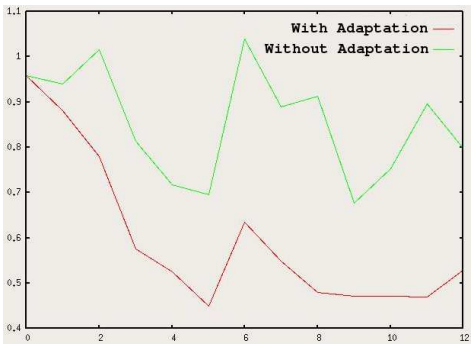

Fig. 7. Evolution of the distance between observations and estimates according to the number of trajectories performed

the same trajectories: after a straight motion he suddenly changes his direction along the North. In figure 5, he has achieved twenty random trajectories (thus our system has re-estimate the TPM twice). In both cases with or without TPM adaption, the pedestrian is tracked. However, without adaptation (blue trajectory), estimates are far from observations due to prediction's errors whereas with the TPM adaptation, estimates are closer to observations. Thus, these results show that our method allows a better tracking of pedestrian positions.

In figure 6, the pedestrian has achieved fifty random trajectories. Here, the tracking performed by our method (red trajectory) is significantly improved after five re-estimations while without adaptation, computed estimates are far from observations during pedestrian's motion changes.

Figure 7 depicts the evolution of the average distance (in meters) between observations and estimates according to the number of trajectories performed. The average distance is computed every ten trajectories, therefore after each TPM re-adaptation by our method. this graph shows that without adaptation (light green top curve) the distance stays roughly constant whereas with adaptation (dark red bottom curve), predictions are improved, the average distance decrease drastically after three or four adaptations and remains one and a half meters closer than without adaptation.

Also, as adaptation is continuous using on-line data, even if pedestrian trajectories vary because of changes in car park configuration, for instance if cars exit the car park, the TPM is automatically readapted to fit this variation. Thus the computed estimations are always better than using an *a priori* TPM, or a learned TPM with a finite set of trajectories since our method is robust to pedestrian behavior changes.

V. Conclusion and Perspectives

This paper presents an adaptive IMM suitable to address pedestrian tracking in car parks. After the IMM has been redefined in a specific formalism, we have presented a method allowing an on-line adaptation of a critical parameter of IMM: the TPM. Indeed, by considering pedestrian trajectories, most probable transitions between IMM modes are quantified and used to update the TPM. In this way, the TPM is more adapted to the pedestrian trajectories and hence IMM computed estimations are better. Application on a real platform and results obtained show the effectiveness of our method.

The next step of this work is to extend our method to track both pedestrians and cars. Also, the introduction of Variable Structure Multiple Models (VSMM) [5] which allow the use of a varying number of modes could improve the adaptability of our method. In particular, if different types of targets are tracked, a different set of modes can be use in relation with the targets' types.

Acknowledgment

This work was partially supported by Provusion project, CNRS (Centre National de la Recherche Scientifique) and DGA (Délégation Générale pour l'Armement).

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotic*. MIT press, 2005.

[2] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 35, Mars 1960.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filter for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[4] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 34, no. 1, pp. 103–123, 1998. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=640267

[5] X. Rong Li and V. P.Jilkov, "A survey of maneuvering target tracking-part v: Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, 2003.

[6] S. Blackman, *Multiple Target Tracking with Radar Applications*. Artech House: Dedham, 1986.

[7] M. Busch and S. Blackman, "Evaluation of imm filtering for an air defence system application," in *Proceedings SPIE Signal Data Process. Small targets*, vol. SPIE 2561, 1995, pp. 435–447. [Online]. Available: http://www.fusion2004.foi.se/papers/IF04-0130.pdf

[8] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 2000.

[9] W. D. Blair and G. A. Watson, "IMM algorithm and aperiodic data," in *Proc. SPIE Vol. 1697, p. 83-91, Acquisition, Tracking, and Pointing VI, Michael K. Masten; Larry A. Stockum; Eds.*, Nov. 1992, pp. 83–91.

[10] O. M.Miller and A. Perrella, "Multiple-Model Filters for Boost-to-Coast Transition of Theater Ballistic Missiles," in *Proceedings SPIE Signal Data Process. Small targets*, vol. SPIE 2561, 1998, pp. 355–376.

[11] Y. Sawaragi, T. Katayama, and S. Fujishige, "Sequential state estimation with interrupted observation," *IEEE Trans. Automat. Contr.*, vol. 21, no. 1, pp. 56–71, aug 1972.

[12] J. K. Tugnait82, "Adaptive estimation and identification for discrete systems with markov jump parameters," *IEEE Trans. Automat. Contr.*, vol. 27, no. 1, pp. 1054–1065, oct 1982.

[13] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Computation*, vol. 12, no. 4, pp. 831–864, 2000.

[14] V. P. Jilkov, X. R. Li, and D. S. Angelova, "Estimation of markovian jump systems with unknown transition probabilities through bayesian sampling," in *NMA '02: Revised Papers from the 5th International Conference on Numerical Methods and Applications*. London, UK: Springer-Verlag, 2003, pp. 307–315.

[15] V. P. Jilkov and X. R. Li, "Online bayesian estimation of transiton probabilities for markovian jump systems," in *IEEE transaction on signal processing*, vol. vol 52, No 6, 2004, pp. 1620–1630.

[16] G. D. Forney, "The viterbi algorithm," *Proceedings of The IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[17] O. Aycard, A. Spalanzani, J. Burlet, T. Fraichard, C. Laugier, D. Raulo, , and M. Yguel, "Puvame - new french approach for vulnerable road users safety," in *Procs. IEEE Intelligent Vehicles Symposium 2006*, Tokyo, Japan, June 2006.