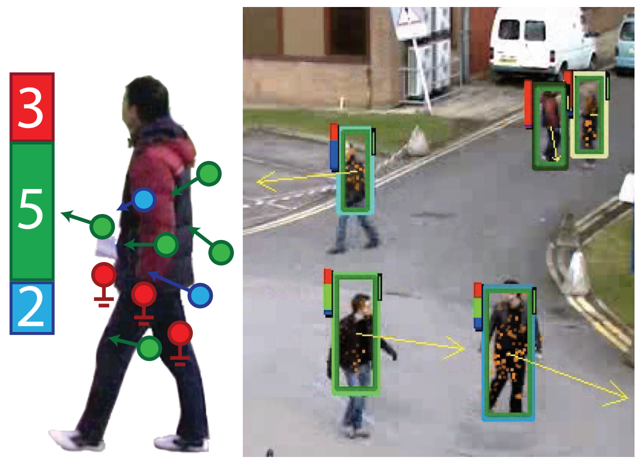
Evaluation of Multiple Motion Models for Multiple Pedestrian Visual Tracking

Francisco Madrigal and Jean-Bernard Hayet
Centro de Investigación en Matemáticas (CIMAT)
Guanajuato, GTO., México
{pacomd|jbhayet}@cimat.mx

Abstract

Multiple targets tracking is a challenging problem due to occlusions or identity switching. Although the use of prior information about the motion of the targets improves the tracking results, a single motion model may not capture the complex dynamic of the targets. This is a common situation with pedestrians, as each person moves in its own way, making tracking a difficult task. In this paper, this problem is faced by using a proposal based on the Interacting Multiple Model (IMM) and implemented in a Bayesian scheme through a particle filter. The core of this approach is to leave the filter choose the motion model that fits better the motion of the targets. The algorithm is evaluated, under several combinations of motion models, with middle-dense crowded scenes from the PETS 2009 dataset.



(a) (b)

Figure 1. Example of particle filter (PF) with interacting multiple motion model. 1(a) Target with a PF of 10 particles (circles), each one with its own model (red: constant position; green: constant velocity; and blue: constant acceleration). The arrows show the target next position. The top-left bar indicate the percent of particles of each model. 1(b) Examples of real world tracking with the proposal.

1. Introduction

In recent years, multiple object tracking (MOT) in real world scenes has become a popular topic, due to its applicability in many areas like surveillance, activity recognition, robotics, among others. Many techniques for pedestrian tracking have been proposed, most of them belonging to two categories: Detection-based methods ([1]) and Bayesian filtering methods ([13]). Regardless of the category the method belongs to, all approaches share at some level the incorporation of a prior knowledge about the targets motion.

The pedestrian motion is governed by different causes such as environment constraints or social forces. Hence, the motion of one pedestrian is somewhat unique and even the motion of a same target can change in more than one occasion. For example, in Fig. 1(b) the couple at the upper right of the image is standing in place, while other people are moving, some at constant velocity (like the couple at the image bottom) or at constant acceleration (like the pedestrian in the middle of the image). However, most of the approaches rely on one classic linear model (*i.e.* constant velocity) to describe this complex dynamic.

In this paper, the motion is modeled as a mixture of models to provide improvements in the visual pedestrian tracking performance. We use the Interacting Multiple Model (IMM) scheme under a Particle Filter (PF) methodology, which allows to integrate many different models in one single framework [2]. The original algorithm has been changed and we propose a new resampling process in the PF, allowing the filter to draw more samples for the motion model that fits better to the pedestrian motion. This modification allows the filter to choose the best model (or models). In the proposed algorithm, a single PF tracker follows one pedestrian and each particle has a motion model associated to it. This is shown in Fig. 1(a), where circles represent the particles with a different color per model. Fig. 1(b) shows the output of our algorithm, where the upper left bar of each

978-1-4799-0703-8/13/$31.00 ©2013 IEEE

tracker indicates the percentage of particles that is assigned to each model, red for constant position, green for constant velocity and blue for constant acceleration.

The rest of the paper is organized as follows: related work is discussed in section 2. The formulation of the IMM is presented in section 3 and we give the details of the particle filter implementation in section 4. The experiments and results of this algorithm are shown in section 5. Finally, conclusions are drawn in section 6.

2. Related work

Multiple motion models are an elegant way to capture the complex dynamics of targets. The IMM methodology can be used to fuse several models in one context by weighting each model and making it contribute to the final distribution [4]. In [12], a hybrid foreground subtraction and pedestrian tracking algorithm is presented that relies on the tracking results as a feedback to improve the foreground subtraction. The tracking is done by a bank of Kalman filters per target, each one with a distinct linear model. The estimation of the filters are merged with the Interacting Multiple Model technique (IMM-KF). The IMM-KF is fast and suitable for a large number of targets. In works similar to ours, this methodology is used in pedestrian tracking in [6]. They use a people detector based on histogram of oriented gradient (HOG) to initialize a IMM-KF tracker. In [5], the authors propose an algorithm that use a bank of KF's and choose the combination of those that optimize the state estimation. However, KFs are limited to linear or linearizable models and can not recover well when one filter fails, which decreases the tracking performance.

To overcome the limitations of IMM-KF, [2] proposes an Interacting Multiple Model implemented with Particle Filter (IMM-PF), for tracking maneuvering targets. They assign a fixed number of particles (1000) to each motion model. The models are weighted according to their importance in the filter. Hence, the most suitable model will contribute more than the others. Instead of the classic resampling of the PF, Boers performs a parametric representation of the distributions by mixtures of Gaussians (MoG), with a high number of terms. However, it suffers of a waste of computational resources at processing particles of models with low importance. Moreover, the MoGs imply clustering, which adds more computational complexity.

In [7], a variant of the IMM-PF is proposed, where the particle motion model is assumed to evolve over time. In this case, particles can keep the same motion or pass from *moving* to *stopped* model or conversely. A prior on the transitions between models is included by a transition matrix (TM) with fixed values, which indicates the jump probability from a model to another. At each time, a model is first sampled from the TM and the previous particle, and then states are sampled with this motion prior. In many cases, fixed transition values cannot represent well the real model changes. Moreover, sampling is done only from the motion transition prior, then there is nothing that guides the particles to high values of the posterior. These works show improvements at tracking simulated targets.

Contributions. Our approach for pedestrian tracking is based on the IMM-PF algorithm. In this paper, we present two contributions for the visual pedestrian tracking problem: first, we present an efficient IMM-PF implementation, that makes no use of transition priors between models, but uses the PF resampling step to redistribute particles among motion models, allowing better fit to multiple motion changes; second, we evaluate the performance of the IMM-PF in multiple pedestrian tracking and test it on challenging public datasets. We evaluate the performance of individual motion models, or combinations of them.

3. Interacting Multiple Motion Models

The Bayes filter, under the Markov assumption for the state, is often written in two steps, the prediction step and the correction step, that correspond to the following two equations updating the posterior distribution over the state

$$\left\{ \begin{array}{ccc} p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) & = & \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})d\mathbf{X}_{t-1}, \\ p(\mathbf{X}_t|\mathbf{Z}_{1:t}) & \propto & p(\mathbf{Z}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}). \end{array} \right. \tag{1}$$

We propose to use the Interacting Multiple Model strategy [2] and, for that, model the multi-modality of the motion prior $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ by a mixture of M distributions:

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) \quad = \sum_{m=1}^{M} \pi_t^m p^m(\mathbf{X}_t|\mathbf{X}_{t-1}). \tag{2}$$

The original Bayes filter is then modified into

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) \propto \sum_{m=1}^{M} \pi_t^m p(\mathbf{Z}_t|\mathbf{X}_t) p^m(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \tag{3}$$

with $p^m(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \quad = \quad \int p^m(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})d\mathbf{X}_{t-1}$. Within this scheme, at each step, the weights of each motion model in the mixture are updated in function of their respective likelihoods,

$$\pi_t^m = \pi_{t-1}^m \int p(\mathbf{Z}_t|\mathbf{X}_t)p^m(\mathbf{X}_t|\mathbf{Z}_{1:t-1})d\mathbf{X}_t. \tag{4}$$

4. Particle filter implementation of the IMM

The general formulation of the previous section could be implemented under different schemes, *i.e.* Kalman filters (by assuming linear models and Gaussian additive noise distributions). In our case, we have chosen an implicit representation of the posterior based on the particle filter strategy, due to the easiness of its use, its capability of managing different kinds of distributions and non-linear motion or observation models. Hereafter, we describe the implementation of particle filter for multi-pedestrian tracking.

The targets are defined with a bounding box. Hence our state $\mathbf{X}$ is:

$$\mathbf{X} \rightarrow (u, v, \dot{u}, \dot{v}, \ddot{u}, \ddot{v})^T, \tag{5}$$

where (u, v) is the position in the image plane, $(\dot{u}, \dot{v})$ is the velocity along the two directions and $(\ddot{u}, \ddot{v})$ is the acceleration in the u and v directions, respectively. The real dimensions (h, w) of the bounding box (BB) around the pedestrians are fixed to the average size of an adult (in world coordinates) and they are projected into a corresponding box in image plane by supposing that the camera projection matrix is known. Hence, the BB of each pedestrian changes with the distance of the target from the camera.

4.1. Particle representation

The PF approximate the posterior in Eq. 1 by a set of N weighted samples or particles. To implement the multimodality, each particle is assigned to a motion model, indicated by a label $l \in \{1 \dots M\}$. Thereby, a particle n at time t is represented by $(\mathbf{X}_t^{(n)}, \omega_t^{(n)}, l^{(n)})$.

In the IMM-PF methodology, the model $m = \{1 \dots M\}$ contributes to the posterior estimation according to its importance, which is defined by the weight π_t^m. Therefore, the model m has N_m particles, making a total of $N = \sum_{m=1}^M N_m$ particles. The posterior estimation can be finally represented by considering both weights of the particles ($\omega_t^{(n)}$) and the models (π_t^m):

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) = \sum_{m=1}^{M} \pi_t^m \sum_{n \in \psi_m} \omega_t^{(n)} \delta_{\mathbf{X}_t^{(n)}}(\mathbf{X}_t), \tag{6}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} \pi_t^m = 1 \quad \text{and} \quad \sum_{n \in \psi_m} \omega_t^{(n)} = 1,$$

where $\psi_m = \{n \in \{1 \dots N\} : l^{(n)} = m\}$ represents the indices of the particles that belongs to model m.

4.2. Sampling and dynamic model

Since drawing samples from $p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Z}_{1:t})$ is generally impossible, we use an importance proposal distribution $q(\cdot)$, that approximates the posterior and from which we can easily draw samples. In multi-motion model case, we have M proposals, such as:

$$\mathbf{X}_t^m \sim q^m(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Z}_{1:t}).$$

Here, we sample a new state of each particle from the motion model corresponding to its label $l^{(n)}$, $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$. This model is a Gaussian $N(\mathbf{X}_t; tr_{l^{(n)}}(\mathbf{X}_{t-1}^{(n)}), \Sigma_{l^{(n)}})$, with $tr_{l^{(n)}}(\cdot)$ the deterministic form of the motion model. Then, sampling is realized by

$$\mathbf{X}_t^{(n)} = tr_{l^{(n)}}(\mathbf{X}_{t-1}^{(n)}) + \nu_{l^{(n)}}, \tag{7}$$

where $\nu_{l^{(n)}}$ is additive noise. The index $l^{(n)} \in \{1 \dots M\}$ indicates which motion model the particle n is associated to, *i.e.* $l^{(n)} = m$ if the particle n is following the model m.

4.3. Observation model and correction step

The probabilistic observation model we implemented, $p(\mathbf{Z}_t|\mathbf{X}_t)$, is rather simple, our aim being to evaluate probabilistic *motion* models and, in particular, Interactive Multiple Models. Our approach relies on color histograms (in the HSV space) and motion histograms (absolute difference between consecutive images), as proposed in [10]. We define a reference histogram h_{ref} anytime we create a new tracker. The likelihood is evaluated through the Bhattacharya distance between h_{ref} and the current histogram $h^{(n)}$ corresponding to the state $\mathbf{X}_t^{(n)}$ of the particle n. Also, we include spatial information with the color observation by using two vertical histograms per target, one for the top part of the pedestrian and another for the bottom part. With the specification of this observation model, we update the weight of particle n within each competing motion model

$$\omega_t^{(n)} = \frac{\tilde{\omega}_t^{(n)}}{\sum_{i \in \psi_m} \tilde{\omega}_t^{(i)}}, \tag{8}$$

$$\tilde{\omega}_t^{(n)} = \frac{\omega_{t-1}^{(n)} p(\mathbf{Z}_t|\mathbf{X}_t^{(n)}) p_{l^{(n)}}(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)})}{q(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)}, \mathbf{Z}_{1:t})},$$

where $p(\mathbf{Z}_t|\mathbf{X}_t^{(n)})$ is the likelihood of observation $\mathbf{Z}_t$ evaluated at the state of particle n and $q(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)}, \mathbf{Z}_{1:t})$ is the proposal distribution. Since $q(\cdot|\mathbf{X}_{t-1}^{(n)}, \mathbf{Z}_{1:t}) = p_{l^{(n)}}(\cdot|\mathbf{X}_{t-1}^{(n)})$, we have:

$$\tilde{\omega}_t^{(n)} = \omega_{t-1}^{(n)} \cdot p(\mathbf{Z}_t|\mathbf{X}_t^{(n)}).$$

The new weight of the models is given by:

$$\pi_t^m = \frac{\pi_{t-1}^m \tilde{\omega}_t^m}{\sum_{i=1}^M \pi_{t-1}^i \tilde{\omega}_t^i}, \tag{9}$$

$$\tilde{\omega}_t^m = \sum_{j \in \psi_m} \tilde{\omega}_t^{(j)}.$$

Thus, Eqs. 8 and 9 ensure that the constraints on Eq. 6 are always satisfied.

4.4. Resampling

The resampling allows to avoid sampling impoverishment. We apply it in two ways:

Sampling over particles. The sampling is done over all particles according to its weight $\omega_t^{(n)}$ and the weight of the model π_t^m. From these weights, we build a common Cumulative Distribution Function and we draw new samples according to this distribution. Hence, the best particles of the best model are sampled more often. In this case, the number of particle changes, leaving more particles to the model that fits better to the motion of the target.

Sampling per model. The sampling is done on a per model basis, leaving the same number of particles per model. To preserve diversity, each model has always a minimum of γ particles. If the model has less particles than this threshold ($N_m < \gamma$), we draw new particles from $N(\bar{\mathbf{X}}_{t-1}, \mathbf{S}_{t-1})$, i.e., a Gaussian distribution with mean $\bar{\mathbf{X}}_{t-1}$ and covariance $\mathbf{S}_{t-1}$ of all particles of the previous distribution. To keep the number of particles N unchanged, we take less samples of the model with more particles. We have set $\gamma = 0.1 * N$.

Our approach models the transitions between models implicitly, by resampling from time to time. Hence, no prior transition information is required. The resampling over all particles is applied every $\rho = 4$ frames and over models every $\varphi = 5$ frames. These parameters allow the method to be sufficiently reactive to motion changes without increasing the computational complexity.

5. Experiments

This algorithm has been implemented in C++ with the open source library OpenCV. We have used the PETS'2009 dataset [11] to test and evaluate the performance of our tracking system using the VACE and CLEAR metrics. We test our algorithm with different motion models described hereafter: a few classic linear models, a learned motion prior model, and a combination of those. All the results presented in this section are the median over 30 experiments. The parameter values are selected manually according to the observations in the videos. Finally, we compare our results with other state of the art tracking algorithms.

5.1. Evaluated motion models

In the literature, different motion models exist that help to describe the movement of different targets, such as vehicles, pedestrians, etcetera. The most used models are simple linear models, since they fit well in most of the cases. On the other hand, models also exist that first learn statistically how the target moves in a given scene. Those models fit better the motion of the targets in complex dynamic scenarios. In this work, we tested both types of models (linear models and learned priors).

Constant position model (CP). The target remains in the same position as the time before. The velocity and acceleration are reduced to zero since the target is not moving. Thus, we define this model with $tr(.)$ as a linear function:

$$A_1 = \begin{bmatrix} I_{2\times2} & 0_{2\times4} \\ 0_{4\times2} & 0_{4\times4} \end{bmatrix}. \tag{10}$$

Constant velocity model (CV). This is the most classical model in different tracking approaches.

$$A_2 = \begin{bmatrix} I_{2\times2} & T * I_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & I_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \end{bmatrix}. \tag{11}$$

Constant acceleration model. (CA) In literature, there are many models with different assumptions on how to model the acceleration process of a target [8]. We consider the CA model as:

$$A_3 = \begin{bmatrix} I_{2\times2} & T * I_{2\times2} & \frac{T^2}{2} * I_{2\times2} \\ 0_{2\times2} & I_{2\times2} & T * I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & I_{2\times2} \end{bmatrix}, \tag{12}$$

where T is the elapsed time between estimations. In our case, it is always one since we process all frames. Then, for these three linear models ($k \in \{1, 2, 3\}$) the transition $tr(\cdot)$ is defined by:

$$tr(\mathbf{X}_{t-1}) = A_k \mathbf{X}_{t-1}. \tag{13}$$

Motion Priors (MP). Finally, we have tested our algorithm with a learning-based motion model. This kind of approach works by recollecting information on how the targets moves, in a given scene, to generate a model. In our case, we use the learned motion prior model presented in [9]. To describe it briefly, the image plane is divided into cells of 30×30 pixels and, in each cell, the probability of moving from this cell to another in its neighborhood is estimated from learning data, with an optical flow algorithm (like Lucas-Kanade). Here, for this evaluation, we have used sequences of the same view of PETS'09 datasets, except those sequences that we used to test our algorithm. The use of these priors in the tracking system is the same as in [9]: For a given particle, we determine its corresponding cell and then sample a velocity from the prior.

5.2. Performance evaluation

The PETS'2009 dataset is a challenging benchmark data, used to test and evaluate the performance of pedestrian tracking algorithms. This dataset consists of a set of 8 camera video sequences of an outdoor scene with different levels of difficulty. We use a single camera and we apply our proposal to views 1 and 2 of the S2-L1 scenario. The task in this sparse crowd scenario is to track all the individuals

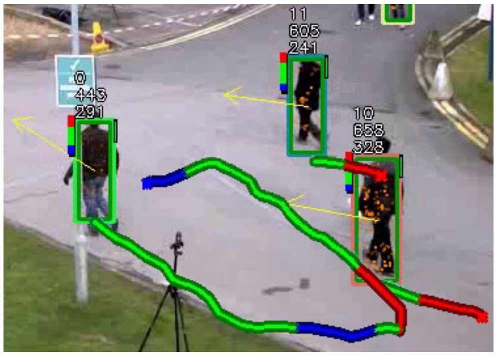


Figure 2. Evolution of the motion models importance. The line is the target trajectory. Each color depicts the model with the highest weight at that time. (red: constant position; green: constant velocity; and blue: constant acceleration).

in the sequence. We chose this sequence since it presents challenging situation for pedestrian tracking.

We have generated ground-truth data to evaluate the performance of our proposal. We have manually labelled each pedestrian in the scene over all frames to generate a ground-truth for both views described in the previous section.

We measure the performance of our system with the standard methodology presented in [3]. We use four metrics: (1) Sequence Frame Detection Accuracy (SFDA), measured at frame-level, takes into account spatial alignment of the system output and ground-truth objects, penalizing missed detections and false positive; (2) Average Tracking Accuracy (ATA) is a spatio-temporal measure that penalizes shorter or longer trajectories, missed trajectories and false positive; (3) Multiple Object Tracking Precision (MOTP) and (4) Multiple Object Detection Precision (MODP), measures the tracks spatio-temporal precision and spatial precision respectively. Those metrics represent the tracking quality with scores between 0 (worst) and 1 (perfect).

5.3. Results

The initialization of a new tracker is automatic. We find the blobs (regions of white pixels) on a binarized image (generated by background subtraction algorithm) to detect a new target. The tracker is automatically removed if the mean likelihood within the particles is lower than a threshold for some consecutive frames. In total, we utilize $N = 200$ particles, assigning at the beginning $\frac{N}{M}$ to each model. We test our algorithm with different combinations of the models presented in section 5.1. First, we depict in Fig. 2 a few examples of trajectories resulting from our framework. Each color indicates the model with highest weight at that moment of the tracking. It is remarkable how the different models adjust to the different parts of the trajectories: In the straight line parts of the trajectories, CV dominate, while at turning points, CA or CP dominate.

Method	SFDA	ATA	N-MODP	MOTP
CA	0.49	0.36	0.53	0.53
CV	0.51	0.36	0.56	0.55
MP	0.47	0.40	0.53	0.53
CV_CA	0.50	0.47	0.56	0.56
CV_CA_MP	0.48	0.49	0.56	0.55
CP_CV_CA	0.45	0.46	0.53	0.53
CP_CV_CA_MP	0.45	0.45	0.53	0.52

Table 1. Results for the S2.L1 sequence, view 1. The first three rows are the results obtained by using only one motion model. The rest are the results of our proposal with a combination of the models presented in section 5.1. Note: All results are the median value of 30 experiments.

Method	SFDA	ATA	N-MODP	MOTP
CA	0.46	0.37	0.53	0.53
CV	0.51	0.40	0.57	0.56
MP	0.48	0.37	0.47	0.47
CV_CA	0.51	0.43	0.53	0.54
CV_CA_MP	0.50	0.44	0.55	0.54
CP_CV_CA	0.50	0.48	0.54	0.56
CP_CV_CA_MP	0.49	0.46	0.51	0.54

Table 2. Results for the S2.L1 sequence, view 2. The first three rows are the results obtained by using only one motion model.

Second, we show quantitative results for the sequences S2-L1 view 1 in Tab. 1. In the first experiments, we evaluate each single model: CA, CV, MP. The best result in the ATA column is obtained using MP with 0.4. This measure indicates that this model allows to track the same target with the same tracker for a longer duration than the others.

Now, we choose a combination of models to evaluate the performance of our approach. We observe that using two simple models like constant velocity and constant acceleration (row four of Tab. 1), the ATA score is improved, which means that we increase the tracking length of the same target, without switching of identity or losing it. We can observe that with more models the quality of the tracking decreases. This is because models themselves permit to change direction more easily. So, the identity switching with targets of similar appearances occurs more often. In Tab. 2 we show the results for the sequences S2-L1, view 2 with the same experiments described earlier. The ATA evolution is similar as in view 1. The SFDA score is highly similar in all experiments, this is due to the way we create new targets. This metric could be improved by using a human detector. Note that the last two scores correspond to precision, they are not really improved by Multiple Motion Models, as they essentially depend on the quality of the observation models (see also Fig. 3 and comparisons with other approaches).

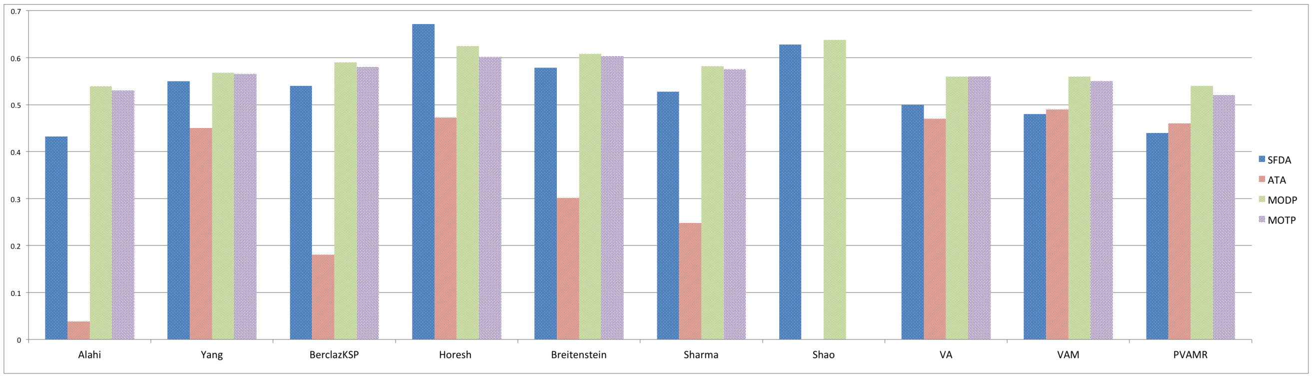


Figure 3. Performance comparison with other proposals in sequence S2-L1, view 1 of PETS09. The y-axis represents the quality of the tracking, zero meaning a bad result and one an optimal performance. We evaluate each proposal (x-axis) with the four metrics described in section 5.2. The results are the median of 30 experiments. The first four groups are multi-camera approaches, meanwhile the rest are monocular proposals. The last three groups are our results using (1) constant velocity and constant acceleration (VA), (2) VA and motion prior model and (3) all models proposed in section 5.1. References: Horesh [1], Shao [12] and the others [3].

Others proposal comparison. In Fig. 3, we compare our results with other proposals, extracted from [3] and [1]. The first four approaches are multi-camera tracking system, the rest (including ours) are monocular tracking systems. The results are for view 1 of S2-L1 PETS'09 sequence. We obtained good results with the ATA score in our monocular system compared with other multi-camera approaches.

6. Conclusion

In this paper, we have first presented a multiple pedestrian visual tracking scheme using multiple motion models within individual particle filters associated to each target. This IMM-PF allows to handle models of different nature, with efficiency improvements over other IMM-PF schemes. Second, we have evaluated several combinations of motion models, for which, with some particular combinations, our overall algorithm behaves nicely on evaluation on the PETS09 dataset, although with a rather rough observation model, and even sometimes in a comparable manner to much more sophisticated tracking algorithms.

References

[1] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2011.

[2] Y. Boers and J. N. Driessen. Interacting multiple model particle filter. In *Proc. of the IEEE Conf. on Radar Sonar and Navigation*, 2003.

[3] A. Ellis and J. Ferryman. Pets2010 and pets2009 evaluation of results using individual ground truth single views. *Proc. of the IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, pages 135 – 142, 2010.

[4] P. Foo. Combining the interacting multiple model method with particle filters for manoeuvring target tracking with a multistatic radar system. *Radar, Sonar & Navigation, IET*, 5(7):697–706, 2011.

[5] T.-J. Ho and B.-S. Chen. Novel extended Viterbi-based multiple-model algorithms for state estimation of discrete-time systems with Markov jump parameters. *IEEE Trans. on Signal Processing,*, 54(2):393–404, 2006.

[6] Z. Jiang, D. Q. Huynh, W. L. J. Moran, and S. Challa. Tracking pedestrians using smoothed colour histograms in an interacting multiple model framework. In *18th IEEE Int. Conf. on Image Processing (ICIP)*, 2011.

[7] C. Kreucher, A. Hero, and K. Keith. Multiple model particle filtering for multitarget tracking. In *Adaptive Sensor Array Processing, 2004. Proc. of Workshop on*, 2004.

[8] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. Part I. dynamic models. *IEEE Trans. on Aerospace and Electronic Systems*, 4048:212–236, 2000.

[9] F. Madrigal, M. Rivera, and J.-B. Hayet. Learning and regularizing motion models for enhancing particle filter-based target tracking. In *PSIVT (2)*, volume 7088 of *LNCS*, pages 287–298. Springer, 2011.

[10] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. of the IEEE*, 92:495–513, 2004.

[11] PETS'09. 11th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance. In *www.pets2009.net*.

[12] J. Shao, Z. Jia, Z.-p. Li, F.-Q. Liu, and J. Zhao. Spatiotemporal energy modeling for foreground segmentation in multiple object tracking. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011.

[13] W. Yi, M. Morelande, L. Kong, and J. Yang. A computationally efficient particle filter for multitarget tracking using an independence approximation. *IEEE Trans. on Signal Processing*, 61(4):843–856, 2013.