A Comparison of Multi Hypothesis Kalman Filter and Particle Filter for Multi-target Tracking

Loris Bazzani, Domenico Bloisi, Vittorio Murino

Department of Computer Science University of Verona, Verona, Italy 37134
{loris.bazzani,domenico.bloisi,vittorio.murino}@univr.it

Abstract

Visual tracking of multiple targets is a key step in surveillance scenarios, far from being solved due to its intrinsic ill-posed nature. In this paper, a comparison of Multi-Hypothesis Kalman Filter and Particle Filter-based tracking is presented. Both methods receive input from a novel online background subtraction algorithm. The aim of this work is to highlight advantages and disadvantages of such tracking techniques. Results are performed using public challenging data set (PETS 2009), in order to evaluate the approaches on significant benchmark data.

1. Introduction

In recent years, the interest of many researcher has been captured by the deployment of automated systems for video surveillance. Automatic surveillance task can be broken down into a series of subproblems [11]: 1) *object detection and categorization* which detects and classifies the *interesting* objects in the field of view of the camera(s); 2) *Multi-Target Tracking* (MTT) where the objective is to estimate the trajectories of targets, keeping the identification of each target; 3) *MTT across cameras* tracks the targets while observing them through multiple overlapping or non-overlapping cameras.

State-of-the-art systems are not able to deal completely with all the dynamics of the complex scenarios (*e.g.*, occlusions, illumination changes, shadows, and tracking failures in crowded environments). In particular, MTT is a challenging task when partial and complete occlusions occur among the targets. In order to solve such problems, integrating different tracking algorithms can reduce erroneous track association, improving system performance, and producing better results.

In this work, we focus our attention on MTT step, presenting a comparison of two well-known techniques: Multi-Hypothesis Kalman Filter (MHKF) and Particle Filter (PF) based tracking. Moreover, we propose an online background subtraction method dealing with the image clutter that causes failures in MTT. Indeed, accurate moving target detection allows to achieve more reliable tracking results.

A Kalman Filter (KF) is an optimal recursive data processing algorithm [16]. While KF is used to track a single target, a multi-target tracking method can be built adding data association and track management steps. MHKF allows for solving the problem of assignment ambiguity: when the correct association is not known (*e.g.*, due to a crowded situation), the same observation can be associated to multiple existing tracks at the same time.

A PF based approach [9] is a three step online procedure for MTT. In sampling step, several events (particles) which describe the state of the system, *i.e.*, the displacement of the targets by sampling a candidate probability distribution (*pdf*) over a state space, are hypothesized. In dynamical step, dynamic is applied to the particles, and, in observational step, each hypothesis is evaluated given the observations of the system and the best fitting ones are selected, thus avoiding brute-force search in the prohibitively large state space of the possible events. In this way, the candidate *pdf* is refined for the next filtering step.

The paper is organized as follows. After discussing related work in Section 2, the architecture of the system is described in Section 3. In Section 4 we present background modeling process, while in Sections 5 and 6 we detail MHKF and PF tracking algorithms respectively. Section 7 shows results obtained by our comparison, while Section 8 provides the conclusions.

2. Related Work

The emphasis in the MTT is given by two general problems: 1) *filtering* and 2) *data association*.

The objective of *filtering* is to estimate the state of the system given all the measurements up to the current time. A number of filtering techniques can be found in literature: KF [12] represents the optimal solution with the assumptions of linear equations and Gaussian noise. Relaxing linearity assumption, a sub-optimal solution is the Extended Kalman Filter [3]. A general solution (*i.e.*, without assumptions) is given by PF that deals with the non-linearity and non-Gaussianity of the system [8].

Particle filtering was originally designed for single-target tracking with CONDENSATION [9], and later extended in a MTT scenario with BraMBLe [10]. MTT with PFs follows different strategies to achieve strong tracking performances avoiding huge computational burdens, exponential in the number of objects to track [10]. In particular, two kind of techniques are discussed in literature: 1) independent PF for each target with Sequential Importance Sampling (SIS) allows to sample in independent state spaces; 2) a Markov Chain Monte Carlo (MCMC) visits efficiently the joint state space [8].

The main drawback of the independent PFs approach is the difficulty to build a reliable interaction model, able to face occlusions among targets. A probabilistic exclusion principle based on an active contours framework is proposed in [15]. MCMC is more efficient than SIS [8] and it enables to build an interaction model in the joint state space, such as MCMC PF and Reversible-Jump MCMC PF [13]. A limitation of those filters is that the interaction model acts only on state space and not on the information given by observations. Hybrid Joint-Separable (HJS) filter [14] is a general solution based on PF, that maintains a linear relationship between number of objects and particles, and it builds a interation model on the observation space.

The general MTT problem concerns with multiple targets and multiple measurements, therefore each target needs to be validated and associated to a single measurement in a *data association* process [3]. Nearest Neighbor (NN) and Probabilistic Data Association (PDA) methods deal with "single targets-multiple measurements" data association problem, while Joint Probabilistic Data Association (JPDA) and Multiple Hypothesis (MH) methods deal with "multiple targets-multiple measurements" data association problem [3]. Those methods are usually combined with KF, *e.g.*, NN filter [20], MH tracker [5], and JPDA filter [4]. The drawbacks of those strategies are twofold. First, occlusions cannot be handled because of the lack of an interaction model. Second, the assumptions of linearity and Gaussianity of the KF cannot manage complex scenarios.

Techniques combining data association methods with particle filtering to accommodate general non-linear and non-Gaussian models are Monte Carlo JPDA filter, Independent Partition PF [23], and Joint Likelihood filter [19].

3. System Overview

The general architecture of the system is shown in Fig. 1. Data coming from PETS 2009 database are processed by three modules: segmentation module, PF tracking module and MHKF tracking module.

Segmentation module creates background model and extracts moving objects, *i.e.*, the foreground image.

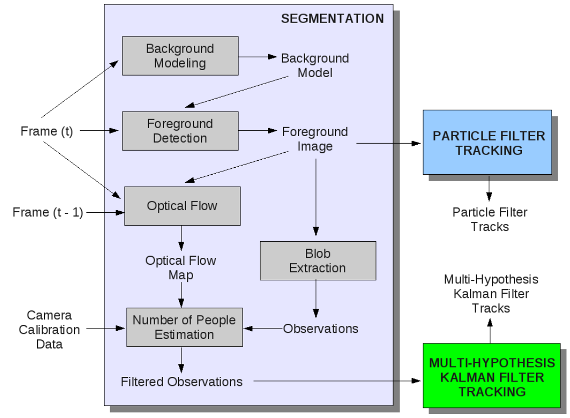

Figure 1: General architecture of the system.

PF tracking module receives foreground image as input in addition to the current frame of the sequence, while MHKF module receives a more refined input, *i.e.*, a filtered set of observations. Such set is the result of a Number of People Estimation (NPE) method (see Section 4) combining inputs from a blob extraction algorithm[1] and from an optical flow computation module [6], in order to overcome the *under-segmentation* problem (*i.e.*, multiple objects that are very close in the scene could be detected as a single blob).

The output of the two tracking approaches is a set of tuple for each time t:

$$\mathcal{X}_t = (\mathrm{ID}, X, Y, Z, x_u, y_l, h, w)$$

where ID is the target identifier, X, Y, Z are the coordinates of the 3D point, x_u, y_l, h, w define the bounding box on the image plane that contains the target. In particular, x_u, y_l is the top-left corner of the bounding box and h, w are the height and the weight, respectively. A list of $\mathcal{X}_t$ with the same ID define a target track.

4. Background Modeling

An overview and comparison of background techniques can be found in [18]. The background can be modeled in a recursive way as: a Gaussian distribution [24], under the condition of static background, and a Gaussian mixture [22, 25], when the background has a multi-modal distribution. Other methods like median filtering [7] and Eigenbackground [17] are based on a non-recursive modeling to compute the foreground.

[1] We use cvBlobsLib library : `http://opencv.willowgarage.com/wiki/cvBlobsLib`

Our approach to background modeling is a non-recursive technique based on a statistical analysis of a buffer L of N_S frames. According to a sampling period P, current frame is added to L becoming a background sample S_i, $1 \leq i \leq N_S$.

An online process clusters sample color values into an RGB histogram, creating a background model M. M is a multi-dimensional image, *i.e.*, every background model pixel is represented by a set of clusters. The complete algorithm for a single color channel is provided in Alg. 1.

Algorithm 1: Background Modeling

Let M be the background model, S_i the ith background sample ($1 \leq i \leq N_S$), D the minimal cluster dimension, and A the association threshold.

foreach *pixel* $\in S_i$ *with color value* V **do**
- **if** $i = 1$ **then**
  - Create new cluster C with centroid $c := V$;
- **else if** $i = N_S$ **then**
  - **foreach** *cluster* C **do**
    - **if** $dim(C) \geq D$ **then**
      - Add centroid $c \in C$ to M;
- **else**
  - **foreach** *cluster* C **do**
    - **if** $|c - V| \leq A$ **then**
      - $c := \frac{V + c \cdot dim(C)}{dim(C)+1}$;
      - Add the pixel V to cluster C;
      - **break**;
    - **else**
      - Create new cluster C with centroid $c := V$;

Each pixel color value for every S_i is associated to a cluster according to a threshold A. After analyzing the last sample, if a cluster has a dimension greater than a value D, then its centroid becomes a background value of M. Up to $\lfloor L/D \rfloor$ background values for each pixel of M are considered at the same time. Such a solution allows for managing noise in sensor data, changes in illumination conditions and movement of small background elements. Furthermore, computational load can be distributed over time P because online clustering computation does not need to wait until L is full.

A new background model is computed with a period equal to $N_S P$, without memory of previous models. N_S and P can vary in time, in order to adapt to both gradual and sudden illumination changes in the scene.

4.1. Background Subtraction

Background subtraction is performed according to a threshold T and a search window $W \times W$. A pixel p belongs to foreground image if at least one of its RGB color values differs from one of the corresponding background model values in the search window. From the resulting foreground image, a set O of observations (namely, blobs) is extracted.

As stated in Section 3, in order to refine O, optical flow and calibration data are exploited (see Fig. 1). A Number of People Estimation (NPE) module calculates the expected number of people E inside an observation blob b. We model a person as a cylinder whose axis is vertical in world coordinate system, extracting E as the volume of b divided by the volume of cylinder model. The filtered set of observations is made of E new observations calculated as a k-means clustering ($k = E$) of optical flow points belonging to b. NPE computation examples are given in Fig. 2.

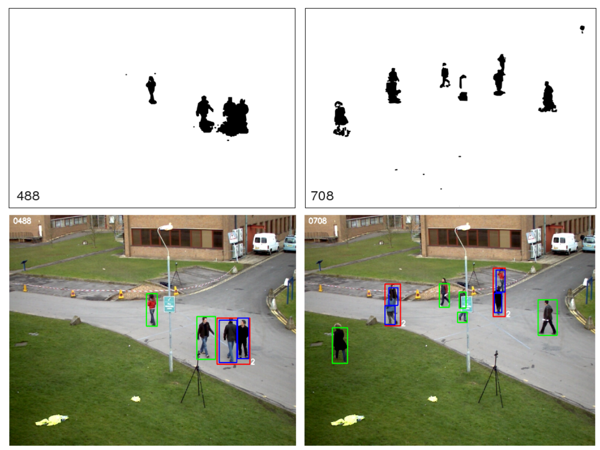

Figure 2: NPE execution examples. Green bounding box represents observations with expected number of people $E = 1$, red bounding box observations with $E = 2$. Blue bounding box represents a filtered observations after NPE is applied.

5. Kalman Filter Tracking

A Kalman Filter (KF) is an optimal recursive data processing algorithm [16], representing an efficient solution to the problem of estimating the state of a discrete-time controlled process, under the hypothesis of linearity and Gaussianity. While KF is used to track a single target, a multi-target tracking method is built adding data association and track management steps.

A set of KFs is used to keep track of a variable and unknown number of moving targets. Each time a new observation is received, it is associated to the correct track among the set of the existing tracks, or, if it represents a new target, a new track has to be created. This is a single hypothesis approach which means that every time an observation is associated to only one of the existing tracks. If a wrong association occurs (*i.e.*, an observation is associated to a wrong track) the system cannot recover from this error. When dealing with crowded scenes, it is not straightforward to assign an observation to a certain track, for this reason we use a MHKF tracking system.

The technique used for the data association is the Nearest Neighbor rule [3]. When a new observation is received, all existing tracks are projected forward to the time of the new measurement (predict step of the filter) and the observation is assigned to the nearest of such predicted state.

The distance between observations and predicted filter states is computed considering the relative uncertainties (covariances) associated with them. The most widely used measure of the correlation between two mean and covariance pair (x_1, Σ_1) and (x_2, Σ_2), which are assumed to be Gaussian-distributed random variables, is:

$$D(x_1, x_2) = \frac{\exp\left(-\frac{1}{2}(x_1 - x_2)(\Sigma_1 + \Sigma_2)^{-1}(x_1 - x_2)^T\right)}{\sqrt{2\pi \mid (\Sigma_1 + \Sigma_2) \mid}} \tag{1}$$

If this quantity is above a given threshold, the two estimates are considered to be feasibly correlated. An observation is assigned to the track with which it has the highest association ranking. In this way, a multiple-target problem can be decomposed into a set of single-target problems. In addition, when an observation is "close enough" to more than one track, multiple hypotheses are generated.

In a multi-hypothesis MTT, tracks are managed by allowing *track split* and *track merge* as well as *track initialization*, *track update*, *track deletion* concerning the single-hypothesis methods. These phases will be described in the following.

Track initialization. When a new observation is obtained, if it is not highly correlated with any existing track, a new track is created and a new KF is initialized with the position (x, y) observed, and a null value is given to all the non observed components (*e.g.*, velocity) with a relatively high covariance. If the subsequent observations confirm the track existence, the filter converges to the actual state.

Track update. Once observations are associated to tracks, standard update of the KFs are performed and the filters normally evolve.

Track split. When an observation is highly correlated with more than one track, new association hypotheses are created. The new observation received is used to update all the tracks with which it has a probability association that exceeds a threshold value. A copy of each not updated track is also kept. Subsequent observations can be used to determine which assignment is correct.

Track merge. This step aims at detecting redundant tracks, *i.e.*, tracks that lock onto the same object (typically after being split). At each step, the correlation with all the other tracks is calculated for each track using equation (1). If the probability association between two tracks exceeds a threshold, one of the two tracks is deleted, keeping only the most significant hypothesis.

Track deletion. Finally, when a track is not supported by observations, the uncertainty in the state estimate increases and when this is over a threshold, we can delete the track from the system. We have considered, as a measure of the uncertainty in the state estimate of each target, the KF gain relative to the track.

6. Particle Filter Tracking

PF offers a probabilistic framework for recursive dynamic state estimation [2], that fits with MTT. The goal is to determine the posterior distribution $p(x_t|z_{1:t})$, where x_t is the current state, z_t is the current measurement, and $x_{1:t}$ and $z_{1:t}$ are the states and the measurements up to time t, respectively. We refer as x_t the state of a single object, and $\mathbf{x}_t = \{x_t^1, x_t^2, \ldots, x_t^K\}$ the joint state (for all objects). The Bayesian formulation of $p(x_t|z_{1:t})$ and the Chapman-Kolmogorov equation enable us to find a sequential formulation of the problem:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|z_{1:t-1}) dx_{t-1} \tag{2}$$

PF is fully specified by an initial distribution $p(x_0)$, the dynamical model $p(x_t|x_{t-1})$, and the observation model $p(z_t|x_t)$. The posterior distribution at previous time $p(x_{t-1}|z_{1:t-1})$ is approximated by a set of N weighted particles, *i.e.* $\{(x_{t-1}^{(n)}, w_{t-1}^{(n)})\}_{n=1}^{N}$, because the integral in Eq. (2) is often analytically intractable. Equation (2) can be rewritten using the *Monte Carlo* approximation:

$$p(x_t|z_{1:t}) \approx \sum_{n=1}^{N} w_{t-1}^{(n)} \, \delta(x_t - x_t^{(n)}). \tag{3}$$

The update of the weights is computed according the follows relation (detailed in [2]):

$$w_t^{(n)} \propto w_{t-1}^{(n)} \, \frac{p(z_t|x_t^{(n)}) \, p(x_t^{(n)}|x_{t-1}^{(n)})}{q(x_t^{(n)}|x_{t-1}^{(n)}, z_t)} \tag{4}$$

where q is called *proposal distribution*. The design of an optimal proposal distribution is a critical task. A common choice is $q(x_t^{(n)}|x_{t-1}^{(n)}, z_t) = p(x_t^{(n)}|x_{t-1}^{(n)})$ because simplifies equation (4) in $w_t^{(n)} \propto w_{t-1}^{(n)} \; p(z_t|x_t^{(n)})$. Thus, the weight at the current time is updated using the weight at the previous time and evaluating the likelihood of the observation with respect to the hypothesis $x_t^{(n)}$.

When only few particles have considerable weights, tracking degenerates to this few particles for estimating the posterior $p(x_t|z_{1:t})$. This issue is called degeneracy problem and can be solved introducing a resampling step [2]. When the estimate of effective sample size N_{eff} =

$\frac{1}{\sum_n (w_t^{(n)})^2}$ is under a threshold N_T, the method resamples the particles generating a new particle set with uniform weights.

6.1 HJS Filter

The HJS approach [14] represents a theoretical grounded compromise between dealing with a strict joint process [10] and instantiating a single independent tracking filter for each distinct object. Roughly speaking, HJS alternates a separate modeling during the sampling step with a joint formulation using a hybrid particle set in the dynamical and observational steps.

The rule that permits the crossing over joint-separable treatments is based on the following approximation (see [14] for details):

$$p(\mathbf{x}_t|z_{1:\tau}) \approx \prod_k p(x_t^k|z_{1:\tau}) \tag{5}$$

that is, the joint posterior could be approximated via product of its marginal components (k indexes the objects). This assumption enables us to sample the particles in a single state space (thus requiring a linear proportionality between the number of objects and the number of samples), and to update the weights in the joint state space. The updating exploits a joint dynamical model which builds the distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ (explaining how the system does evolve) and a joint observational model that provides estimates for the distribution $p(z_t|\mathbf{x}_t)$ (explaining how the observations are related to the state of the system).

Both models take into account the interactions among objects; in particular the joint dynamical model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ accounts for physical interactions between the targets, thus avoiding track coalescence of spatially close targets. The joint observational model $p(z_t|\mathbf{x}_t)$ quantifies the likelihood of the single measure z_t given the state $\mathbf{x}_t$, considering inter-objects occlusions.

The joint dynamical model is approximated in the following way:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{x}_t) \prod_{k=1}^{K} q(x_t^k|x_{t-1}^k) \tag{6}$$

where $q(x_t^k|x_{t-1}^k)$ is the single target dynamical model, that spread independently the particles of each target, and $p(\mathbf{x}_t)$ is a joint factor that models the interaction among the targets.

The PF dynamic process is split into two step: 1) applying the dynamic of the single target hypotheses, and 2) evaluating jointly the interactions among the hypotheses of all the targets. In particular, we model $q(x_t^k|x_{t-1}^k)$ as a first order autoregressive model adding a Gaussian zero-mean noise. The joint factor $p(\mathbf{x}_t)$ can be viewed as an exclusion principle: two or more targets cannot occupy the same volume at the same time. In [14], $p(\mathbf{x}_t)$ is modeled with a pairwise Markov Random Field (MRF). Inferring on a MRF with Belief Propagation, the hypotheses that do not agree with the exclusion principle have a low probability to exist.

The joint observational model relies on the representation of the targets, that here are constrained to be human beings. Person representation assumes the human body in three parts: head, torso, and legs as in [10]. For the sake of clarity, we assume the body as a whole volumetric entity, described by its position in the 3D plane, with a given volume and appearance captured by HSV intensity values. The joint observational model works by evaluating a separate appearance score for each object, encoded by a distance between the histograms of the model and the hypothesis (a sample), involving also a joint reasoning captured by an *occlusion map*.

The occlusion map is a 2D projection of the 3D scene which focuses on the particular object under analysis, giving insight on what are the expected visible portions of that object. This is obtained by exploiting the hybrid particles set $\{x_p\}_{p=1}^{NK}$ in an incremental visit procedure on the image plane: the hypothesis nearest to the camera is evaluated first, its presence determines an occluding cone in the scene, where the confidence of the occlusion depends on the observational likelihood achieved. Particles farther in the scene which fall in the cone of occlusion of other particles are less considered in their observational likelihood computation. The process of map building is iterated as far as the farthest particle in the scene and the observation model is defined as:

$$p(z_t|x_p) \propto \exp\left(-\frac{\mathrm{fc}_p + \mathrm{bc}_p}{2\,\sigma^2}\right) \tag{7}$$

where fc_p is the foreground term, *i.e.*, the likelihood that an object matches the model considering the unoccluded parts, and bc_p, the background term, accounts for the occluded parts of an object. These terms are computed accounting for the occlusion map, that deals with the partial occlusions among different persons on the image plane. Moreover, in order to improve the robustness and the reliability of the method, we integrate the occlusion map and the foreground image, derived from the background/foreground segmentation process discussed in Section 4.

7. Results

The testing session has been focused on the PETS dataset [1], in order to compare and to certify the performances of both approaches with public challenging data. In particular, we choose S2 dataset, using the first camera view ("View_001") of the first sequence.

The objective is to track all of the persons (targets) in the sequence with both methods in a monocular setup. In order to give an evaluation and validation of the methods, the sequence has been manually labeled generating the ground truth[2] of the targets. The ground truth consists of: the identifier (ID) of each target, its 3D position on the ground plane, and its 2D bounding box which is associated to a specific view.

We prefer not to use background model provided by PETS training dataset in order to simulate a real computation of the system. Our background modeling method initialization step needs 220 frames to create the first background model, thus people tracking starts from the frame number 220 to the end of the sequence (795).

7.1 Segmentation Results

For the experiments, we set background modeling parameters as follows: number of samples N_S = 15, sampling period P = 1.8 seconds, minimal cluster dimension D = 2 and association threshold A = 4. Background subtraction parameters are T = 10 and W = 5.

The segmentation error e_t for each frame t is computed as

$$e_t = \frac{|\hat{n} - n|}{n} \tag{8}$$

where $\hat{n}$ is the number of detected people and n is the real number of people in the scene. The average accuracy A for a set of N_f frames is computed as

$$A = \frac{1}{N_f} \sum_{t=1}^{N_f} (1 - e_t) \tag{9}$$

BS method	No. of frames	Accuracy
without NPE	100	0.841
with NPE	100	0.949

Table 1: Segmentation accuracy.

The results are showed in Table 1, where two different types of background subtraction (BS) are considered: with and without NPE module. NPE allows to correct errors when people are very close each other and when occlusions occur. Qualitative results are shown in Fig. 3.

The algorithm performance in terms of frame per seconds (fps) are shown in Table 2, for evaluating the real-time performances of the background subtraction algorithm. The first row shows the test on PETS 2009 dataset (recorded at 7 fps). Our system emulates PETS video acquisition setting loading a new frame every $1/7$ seconds, thus achieving PETS provided fps. Live acquisitions has been carried out in order to evaluate real performances of the proposed method (last rows in Table 2). The tests are sorted from lower to higher resolution images obtaining an evaluation that permits to discover the ideal resolution, for real-time computation.

Video acquisition	Frame Dim.	FPS
PETS 2009	768×576	7
Live 1	320×240	25
Live 2	640×480	14
Live 3	768×576	10

Table 2: Segmentation algorithm speed.

7.2 Tracking Results

An evaluation has been carried out on the image plane in terms of: False Positives (FP), Multiple Objects (MO), False Negatives (FN), Multiple Trackers (MT), and Tracking Success Rate (TSR), see [21] for further details. The results, averaged over all the frames of the sequence and for all the moving targets, are summarized in Table 3.

	FP	MO	FN	MT	TSR
MHKF	0.279	0.009	**0.203**	0.212	0.624
HJS	**0.086**	**0.007**	0.279	**0.042**	**0.712**

Table 3: Tracking results comparison on PETS 2009 dataset: task S2, video L1, view 1.

HJS filter performs better than MHKF considering FP, MO, MT, and TSR, because MHKF generates more than one tracks for a single target. In this case, it is possible that a target is tracked from more identifiers and, on the other hand, it is possible that a target is not tracked. However, the FN ratio is higher for HJS, motivated from the fact that sometimes the tracking of an target is lost and it converges toward another one or the clutter. The TSR gives us a general value of the tracking reliability and it summarizes the performances. It is clear from the results that HJS is more reliable than MHKF.

In Fig. 4, we report some tracking results (the complete sequence can be found at `http://profs.sci.univr.it/~bazzani/PETS2009`), in order to evaluate qualitatively the methods discussed in this paper. In particular, we show three frames of the task S2, video L1, view 1 from PETS 2009 dataset. First row shows the tracking results regarding MHKF, whereas second row shows HJS results.

From the experiments we find that drawbacks of MHKF are: 1) targets are tracked with multiple tracks, leading to a proliferation in the number of tracks; 2) after an occlusion

[2] Publicly available at `http://profs.sci.univr.it/~bazzani/PETS2009`

the target ID changes, *i.e.*, a new track instance is created; 3) MHKF tracking fails when people motion is non-linear. Vice versa, HJS overcomes the problems of MHKF: 1) only one track is kept for each target, because the data association is inherent in the particle filtering formulation; 2) after an occlusion the target ID is kept, thanks to the integration of the occlusion map into the observation model; 3) HJS can deal with non-linearity characterizing people motion.

However, HJS has some problems with complete occlusion. In this case, the tracker cannot infer the position of the target because of the lack of foreground observations for the occluded target. Thus, tracking of a target fails in case of long-term occlusions.

Integrating MHKF and HJS can lead to reduce erroneous track associations. HJS can help MHKF in merging redundant tracks belonging to the same target, reducing the problem of track proliferation. MHKF can help HJS in reinitialization after a track loss and in deleting track not corresponding to foreground observations.

8. Conclusions

In this work, two kinds of multi-target tracking approaches have been presented: Multiple-Hypothesis Kalman filter and Particle filter based (HJS). A comparison of these methods has been carried out analyzing a recent challenging dataset (PETS 2009), and the results show the robustness of both approaches. In particular, HJS performs better than MHKF when occlusions occur keeping the identity of the target after occlusion, while MHKF tends to generate a new target ID.

In addition, a novel online background subtraction algorithm has been proposed and investigated. Testing session shows the real-time performances and the accuracy of the algorithm in PETS datasets and live video acquisitions.

As future works, we intend to integrate both the tracking methods in a single framework, in order to exploit advantages minimizing drawbacks.

Acknowledgments

This research was partially funded by the European Project FP7 SAMURAI, Suspicious and Abnormal Monitoring Using a netwoRk of cAmeras & sensors for sItuation awareness enhancement, Grant Agreement No. 217899. The authors thank Dr. Marco Cristani for his helpful work and his insightful comments.

References

[1] Pets 2009: Performance evaluation of tracking systems 2009 dataset, `http://www.pets2009.net`.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.

[3] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[4] Y. Bar-Shalom, T. Formann, and M. Scheffe. Joint probability data association for multiple targets in clutter. *Proc. Conf. Information Science and Systems*, 1980.

[5] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, 2004.

[6] J. Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: description of the algorithm, 2002.

[7] S. Calderara, R. Melli, A. Prati, and R. Cucchiara. Reliable background suppression for complex scenes. In *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 211–214, New York, NY, USA, 2006. ACM.

[8] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Birkhauser, 2001.

[9] M. Isard and A. Blake. Condensation: Conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29:5–28, 1998.

[10] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *IEEE Int. Conf. on Computer Vision*, 2001.

[11] O. Javed and M. Shah. *Automated Multi-Camera Surveillance: Algorithms and Practice*. Springer Publishing Company, Incorporated, 2008.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Tran. of the ASME Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[13] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

[14] O. Lanz. Approximate bayesian multibody tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, 2006.

[15] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *Int. J. Comput. Vision*, 39(1):57–71, 2000.

[16] P. S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[17] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, Aug 2000.

[18] D. H. Parks and S. S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199, 2008.

[19] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking multiple and compound visual objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:560–576, 2000.

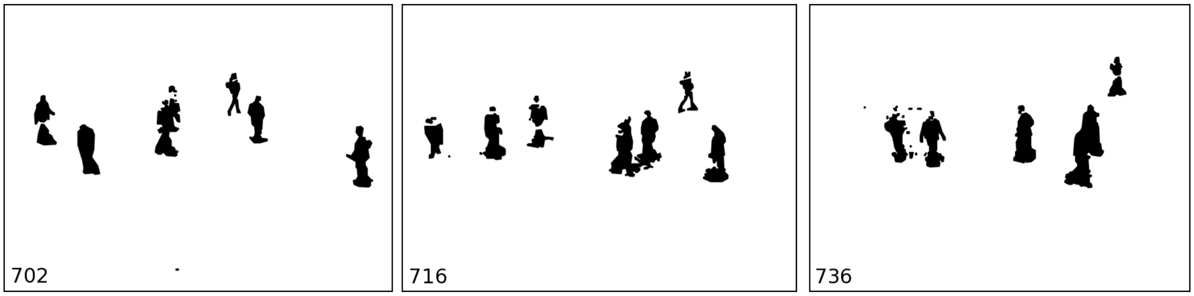

Figure 3: Foreground images (moving objects in black) for frames 702, 716, and 736 of task S2, video L1, view 1

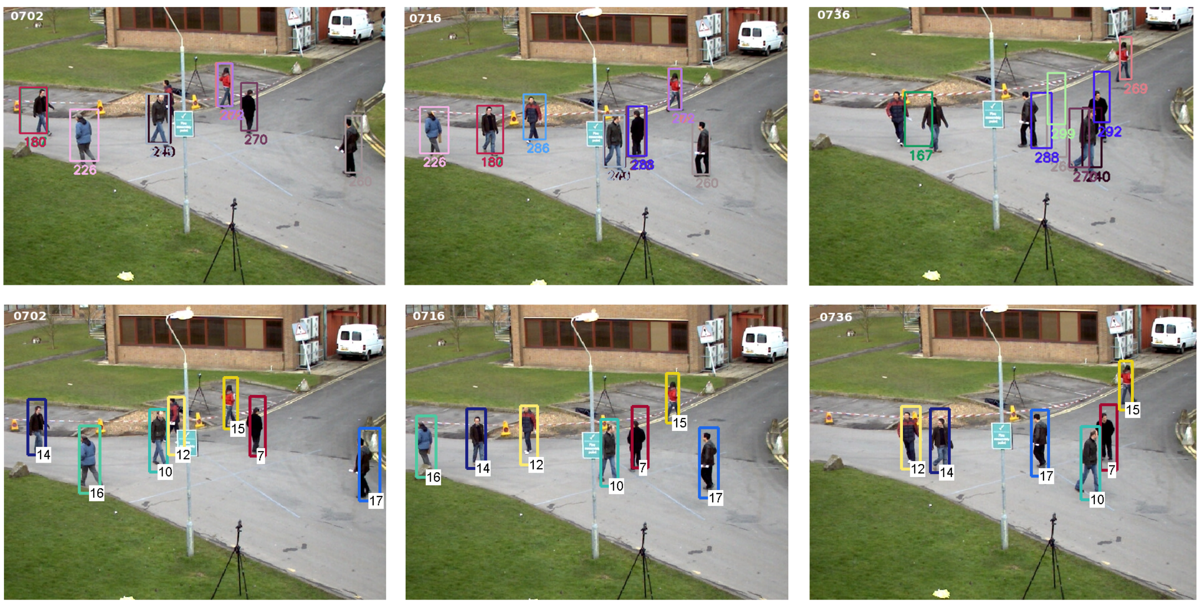

Figure 4: Tracking results for frames 702, 716, and 736 of task S2, video L1, view 1: first row shows MHKF and second row show HJS filter.

[20] X. Rong Li and Y. Bar-Shalom. Tracking in clutter with nearest neighbor filters: analysis and performance. *IEEE Trans. on Aerospace and Electronic Systems*, 32(3):995–1010, 1996.

[21] K. Smith, D. Gatica-Perez, J. Odobez, and S. Ba. Evaluating multi-object tracking. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 36–43, 2005.

[22] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 252–259, 1999.

[23] J. Vermaak, S. Godsill, and P. Perez. Monte carlo filtering for multi target tracking and data association. *IEEE Trans. on Aerospace and Electronic Systems*, 41(1):309–332, 2005.

[24] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[25] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, 2006.