Machine Vision and Applications (2013) 24:1751–1765
DOI 10.1007/s00138-013-0534-9

ORIGINAL PAPER

Tracking-by-detection of multiple persons by a resample-move particle filter

Iker Zuriarrain · Alhayat Ali Mekonnen · Frédéric Lerasle · Nestor Arana

Received: 30 July 2012 / Revised: 23 May 2013 / Accepted: 8 July 2013 / Published online: 3 August 2013

Abstract Camera networks make an important component of modern complex perceptual systems with widespread applications spanning surveillance, human/machine interaction and healthcare. Smart cameras that can perform part of the perceptual data processing improve scalability in both processing power and network resources. Based on these insights, this paper presents a particle filter for multiple person tracking designed for an FPGA-based smart camera. We propose a new joint Markov Chain Monte Carlo-based particle filter (MCMC-PF) with short Markov chains, devoted to each individual particle, in order to sample the particle swarm in relevant regions of the high dimensional state-space with increased particle diversity. Finding an efficient sampling method has become another challenge when designing particle filters, especially for those devoted to more than two or three targets. A proposal distribution, combining diffusion dynamics, learned HOG + SVM person detections, and adaptive background mixture models, limits here the well-known burst in terms of particles and MCMC iterations. This informed proposal based on saliency maps has only been marginally used in the literature in a joint state space PF framework. The presented qualitative and quantitative results—for proprietary and public video datasets—clearly show that our tracker outperforms the well-known MCMC-PF in terms of (1) tracking performances, i.e. robustness and precision, and (2) parallelization capabilities as the MCMC-PF processes the particles sequentially.

Keywords Video surveillance · Monocular color vision · Particle filtering · MCMC · Data fusion · Multi person tracking

I. Zuriarrain (✉) · N. Arana
Goi Eskola Politeknikoa, Mondragon Goi Unibertsitatea, Mondragon, Spain
e-mail: izuriarrain@gmail.com

N. Arana
e-mail: narana@mondragon.edu

A. A. Mekonnen
LAAS-CNRS, 7 Avenue Colonel Roche, 31077 Toulouse, France
e-mail: aamekonn@laas.fr

F. Lerasle
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, 31077 Toulouse, France
e-mail: lerasle@laas.fr

1 Introduction

Visual multiple object tracking (MOT) has received tremendous attention in the vision community due to its numerous applications such as video surveillance in public or private human-centered environments (see a survey in [18]). Deploying a network of ceiling-mounted cameras is challenging as it should be easy to install by a non-expert user while the cameras should include on-board CPU resources to exchange high level data such as positions and characteristics of the target persons over the network. Since no intelligent camera dedicated to human tracking, contrary to human detection, is currently available off-the-shelf, our tracker is devoted to an intelligent camera with an FPGA board to execute parts of the algorithm in parallel (Fig. 1).

Besides this broader technological aim, the traditional challenge with MOT is to simultaneously track persons that can a priori enter, exit, pass close to one another or merge in the scene. The objectives are twofold: (1) to correctly detect entering, leaving, and temporarily occluded targets, that is, characterize the targets' status and, (2) to obtain a record of trajectories corresponding to the observed targets, i.e. the

Fig. 1 Close-up of the wireless camera with an FPGA board from Delta Technologies [1]

targets' positions over time—maintaining a correct and unique identification for each target throughout.

To cope with these difficulties, approaches based on tracking-by-detection and sequential Monte Carlo have become increasingly popular as they improve robustness in MOT by coupling target detection and tracking in the well-known particle filtering framework. They are well suited for parallelization [12,22] as the standard PF weighs particles independently based on a likelihood function and then propagates these weighted particles according to an importance stage with an underlying proposal distribution based on a classical motion model. Various researchers have attempted to extend to MOT using decentralized approaches (namely one particle filter per target) [9,10,35] or centralized approaches [11,26]; the latter deals more appropriately with the problem of joint data association between multiple targets. Recently, the introduction of Markov Chain Monte Carlo (MCMC) in the importance sampling stage has proved: (1) to deal more efficiently with high- and trans-dimensional state spaces and, (2) to require far fewer samples to adequately track the joint target state devoted to MOT [6,27,37]. The resulting MCMC-PF centralized approach propagates a set of unweighted particles (over time) which are drawn iteratively through a first-order Markov process. The limitations therefore are (1) its non parallelizability on clusters as the framework remains sequential like pure MCMC strategies and, (2) the high number of required burn-in iterations, especially for handling the continuous parameters, i.e. the targets' positions.

Our filtering approach combines the strengths of the two afore-mentioned algorithms—centralized PF and MCMC sampling steps combined with detection routines. We design a new filtering strategy where an MCMC sampling step using a resample-move strategy handles the discrete variables of the system, namely the targets' status, leading to a more relevant sample cloud diversity. This breaks the time consuming burn-in iteration phase into multiple short Markov chains which are easily parallelizable. The continuous parameters can be handled with the traditional particle weighting stage.

Regarding the person detection routines, we propose an efficient proposal distribution based on saliency maps that combine several information sources like diffusion dynamics, learned HOG+SVM person detections, and adaptive background mixture models. Combining detector responses within saliency maps is also considered in [4,42], but it is fed into single or multiple object meanshift-based tracker in which the prediction is solely based on dynamics. Choudhury et al. [14] also use probability maps produced by a face detection routine in the PF weighting stage while the sampling stage is based on a zero-order dynamic model. Such probability maps, as far as we are aware, have only been marginally used in the importance sampling stage. This choice of proposal distribution is crucial in PF as it dictates ways to draw/predict the particles in the relevant state space areas.

Finally, we propose experiments that demonstrate how the proposed approach outperforms the conventional and most closely related MCMC-PF strategy and how it is more suited for parallelization.

The rest of the paper is organized as follows: Sect. 2 presents a brief survey of the literature in the area of multiple human tracking putting our work in perspective. Section 4 recalls some basic concepts about PF and MCMC-based methods; it then depicts our hybrid MCMC-based particle filtering framework. Sections 5 and 6, respectively, detail the tracker implementation and a discussion on the relative performance of our approach and prior filtering strategies in the context of monocular color vision-based video-surveillance. Finally, a brief summary and future works are discussed in Sect. 7.

2 Related work

Literature review on visual tracking is beyond the scope of this paper. As the proposed approach mixes the ideas of MCMC-PF and detection driven multiple person tracking, we mainly discuss these contexts in visual tracking.

Particle filtering [3] offers a framework for representing the tracking uncertainty in a Markovian manner by only considering information from the current and the previous frame. The strength of this stochastic formulation and its numerous variants (CONDENSATION[1] [24], Mixed-state CONDENSATION [25], Auxiliary [32], etc.) lies in their simplicity, flexibility, and systematic treatment of non-linearity and non-Gaussianness while being more suitable for time-critical, online applications. In the PF framework, the classical MOT literature proposes decentralized or centralized solutions which are based on independent filters (one per target) or a joint state-space state representation respectively.

As pointed out in [41], the classical *decentralized PF-based approach* based on multiple independent PF performs poorly when the posterior is multimodal as the result of

[1] for "Conditional Density Propagation".

multiple targets. To circumvent this problem, several extensions have been proposed. Some researchers [34,41] introduce a mixture PF, where each component/mode is modeled with an individual PF that forms part of the mixture. This mixture strategy is shown to require fewer samples but necessitates re-clustering of particles at each time step.

Another decentralized solution is to propose interactively distributed filters, i.e. one filter per target, which is computationally inexpensive [31,35,45]. Wu et al. [45] use multiple collaborative trackers for MOT modeled by a Markov random network but they do not deal with the false labeling problem. The decentralized approach is carried further in [35] which proposed an interactively distributed MOT (IDMOT) framework using a magnetic-inertia potential model. Schemes like [43] are shown to improve the efficiency but also face limitations.

However, the decentralized strategy suffers from the "data association error" whenever targets pass close to one another [5]. Consequently, the target with the best likelihood score typically "hijacks" the filters of nearby targets. To overcome this problem, recent approaches [9,44] combine tracking with detection in decentralized PFs to re-initialize in case of target loss.

In contrast, *centralized PF-based approaches* estimate a joint state which concatenates all of the targets' states and so estimates both discrete (number of targets) and continuous variables (targets' positions) [11,19,23,26]. By characterizing all possible associations between targets and observations, this formulation deals more appropriately with the joint data association problem. Some variants like kernel PF [11] improve efficiency, but also face the intrinsic limitations of centralized methods. Indeed, centralized PF suffers from exponential complexity in the number of targets due to the inefficiency of importance sampling which classically draws the particles from the system dynamics, i.e. "blindly" w.r.t the measurements. A remedy would be to steer sampling towards the high likelihood state space regions by incorporating both the dynamics and the measurements in the proposal distribution.

Going one step further, an alternative addressed in [6,13,27,37] is to replace the traditional importance-based sampling by a *Markov Chain Monte Carlo* (MCMC) sampling step within the joint PF. An unweighted sample swarm is obtained by storing the samples after the initial burn-in iterations in the Markov chain. Yet, this filtering strategy is shown to outperform their pure PF or MCMC counterparts. The required iteration number is yet worsened by the MCMC sampling which usually draws the particles solely according to the dynamics and so is possibly subject to a very high rejection rate. A major issue is then to draw the particles in the relevant areas of the high dimensional state-space while limiting the notorious burst in terms of particles and MCMC iterations.

3 Our approach

A first step to reduce the MCMC complexity is to design a novel MCMC-based PF where the MCMC sampling step is devoted only to the targets' status with respect to people leaving/entering the scene. This sampling step through a small number of Markov chain moves is applied to each individual particle. These moves increase particle diversity with respect to the number of targets to track. This filtering strategy, which remains well suited for parallelization (except for the short Markov chain), is inspired by the *resample-move filtering* formalized by Berzuini et al. [8]. But, we leverage this strategy to address the visual tracking of a variable number of interacting targets, while Berzuini's work centers on single track synthesis for trajectory calculation and uses MCMC to switch between different dynamic models.

A second step to reduce the MCMC complexity is to construct an efficient data driven proposal distribution; it is widely accepted that proposal distributions that incorporate the recent observation outperform naive transition proposals considerably [33,34,36]. We propose a novel *saliency map-based proposal* to overcome the tricky/computational problem of combining multiple detector outputs in the tracking loop. There are a few attempts to combine the strengths of detection and tracking recently, but we can notice that almost all tracking-by-detection approaches consider a single detector. This is a major limitation of tracking-by-detection approaches especially those based on a single motion detector which often integrate static objects into their dynamically updated background models. To detect and track both moving and static persons, the expression of our proposal is given by a mixture of saliency maps corresponding to persons' appearances and motion, but it can easily be extended to a wider set of detectors.

A last consideration about tracking-by-detection paradigm concerns detectors trained either online or off-line. Many recent approaches [2,9,20] consider, both in the detection and tracking routines, the same observation model which is adapted on-the-fly. It is widely accepted that such unsupervised adaptation is prone to jitter and model errors may accumulate gradually [40,44]. Off-line trained detectors are ideal means to provide such supervision as it relates to absolute information. In the vein of [10,34], our approach merges two distinct data sources from preceding frames in traditional recursive fashion[2] with that provided by off-line detectors devoted to the proposal. In essence, we combine the strengths of methods that rely on absolute information with those based on chained transformations. The former do not drift but cannot provide enough precision for every frame and so result in jitter. The latter do not jitter but tend to drift or even lose track altogether. Another fundamental

[2] i.e. a color distribution in the particle weighting stage.

difference from our detection routine is that classical tracking-by-detection approaches [9,14,20,29,30] assume detectors with high recall rates provide sufficient detection results. In contrast, we assume detectors with high recall merely output reliable detection results occasionally. Hence, contrary to [9, 14], the detectors are not used in the particle weighting stage but in the importance sampling stage which also involves the dynamics.

The above review highlights the contributions of our centralized PF-based approach. The work most closely related to ours is the well-known MCMC-PF proposed by Khan et al. [27] but two major extensions are exhibited here. First, we use saliency maps (produced by several off-line trained detection routines) in the construction of the proposal. Second, we propose a new resample-move particle filter based on a short Markov chain to increase the parallelization capabilities compared to MCMC-PF. The result of interleaving saliency maps with our resample-move filter is a powerful and fully automatic MOT which outperforms the traditional MCMC-PF [27] (annotated MCMCPFv1 subsequently) and its upgraded version with saliency maps (annotated MCMCPFv2).

4 Monte carlo-based tracking strategies

Recall that we aim to fit a template relative to each target all along the video stream through the estimation of its status $r \in \{New, Tracked, Lost, Dead\}$, its image coordinates (u, v) and its scale factor s. These parameters are accounted for in the state vector $\mathbf{x}_k$ related to the kth frame. With regard to the system dynamics, the unpredictable motion of humans leads to define the state vector $\mathbf{x}_k = (r_k, u_k, v_k, s_k)^{'}$ and to assume that its entries evolve according to a mutually independent random walk model, *viz.* $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k-1}, \Sigma)$ where $\mathcal{N}(.; \mu, \Sigma)$ is a Gaussian distribution with mean μ and covariance $\Sigma = \text{diag}(\sigma_u^2, \sigma_v^2, \sigma_s^2)$. Finally, the global state vector is defined by $\mathbf{X}_k = (\mathbf{x}_k^1, \ldots, \mathbf{x}_k^{N_t})^{'}$ where N_t is the number of targets.

Each target might be in one of the four states: $Dead$, in which case the target is outside the camera FOV; New, in which it is a new target that appeared in the current image; $Tracked$, in which the target has been tracked for at least one previous image, and therefore, is already known; and $Lost$, in which the target has disappeared in the current frame. Logically, New targets turn into $Tracked$ targets after processing the current image, and $Lost$ targets likewise turn into $Dead$ targets and as such, are removed from the state vector.

In this section, we will take the above as a base and examine which different tracking strategies might be used to reliably track people in the scene.

4.1 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) methods have been used in a number of MOT systems, e.g. in [38,46]. This class of algorithms, of which the most well-known is the Metropolis–Hastings algorithm, attempts to approximate an unknown distribution. This is achieved by creating a Markov chain of samples that successively approach the target distribution; in tracking systems, this corresponds to the actual scene configuration.

As mentioned, the renowned MCMC algorithm is the Metropolis-Hastings (M-H) algorithm, which is shown in Table 1, applied to MOT. This algorithm starts with a starting configuration $\mathbf{X}$, from which a new state $\mathbf{X}'$ moving the state subspace $\mathbf{x}'$ of a randomly drawn target j (step #5) according to a transition model (the so-called proposal density) $q(\mathbf{x}'_k|\mathbf{x}_k, z_k)$, with $z_{1:k} = z_1, \ldots, z_k$ the measurements set, is proposed. The acceptance ratio $\mathscr{R}_a$ of this $\mathbf{X}'$ state is then calculated (step #7) where the $\Psi(.)$ are the pairwise interaction potentials between targets. This ratio is then compared to a uniformly drawn threshold β, which determines if the new state is accepted as state $\mathbf{X}$ for the next iteration, or rejected. This process continues for a high enough number of iterations that the algorithm can be said to have converged to a point in which drawing any further states is approximately equivalent to drawing from the distribution the Markov chain is simulating.

While an MCMC with an adequate state transition model and acceptance ratio calculation will eventually converge, the number of iterations necessary to do so heavily depends on the starting position $\mathbf{X}_0$ and the discriminative capabilities of the measures used in the calculation of $\mathscr{R}_a$. As a result, it becomes necessary to run the Markov chain for a number of burn-in iterations before any accurate sampling can be attempted.

An MCMC algorithm can adequately handle variable dimension state spaces, which is one of the challenges of MOT, as shown by the Reversible Jump algorithm [21]. The criteria for dimensionality changes are that if the drawing of a new state $\mathbf{X}'$ includes a probability for a dimensionality change, there must exist a corresponding probability for the opposite operation. This ensures that the search for the convergence can properly navigate the state-space.

A common technique that simplifies both the drawing of $\mathbf{x}'^m$ and the evaluation of $\mathscr{R}_a$ is for the state transition model to consider only changes to a randomly chosen subset of the state (in the case of MOT, this translates into changing a single target per iteration). As a result, the parts of the state vector that remain constant can be simplified from the $\mathscr{R}_a$ calculation. While this does increase the number of necessary iterations, the complexity of the system is greatly reduced as a consequence.

Table 1 MCMC tracking strategy with Metropolis–Hastings algorithm

$[\{\mathbf{X}_k^i\}]_{i=1}^N = \text{MCMC}(\mathbf{X}_{k-1}, z_k)$

IF $k=0$, **THEN** Set status $r_{k-1}^t = Dead$ for each target t **END IF**
IF $k \geq 1$ **THEN** {}
  FOR $m = 1,\ldots,N_{mcmc}$, **DO**
    Randomly pick a target j and propose an associated event/jump τ (events have all the same probability of happening)
    Apply τ to $\mathbf{X}_k$ to generate $\mathbf{X'}_k$ with update of the j-th target's status.
    Draw threshold β according to a uniform distribution over $(0,1]$
    Calculate the acceptance ratio $\mathscr{R}_a = \frac{p_1(z_k|\mathbf{x'}_k).q(\mathbf{x}_k|\mathbf{x'}_k, z_k).\Psi(\mathbf{x}_k^j, \mathbf{x}_k^m)}{p_1(z_k|\mathbf{x}_k).q(\mathbf{x'}_k|\mathbf{x}_k, z_k).\Psi(\mathbf{x'}_k^j, \mathbf{x'}_k^m)}$
    IF $\mathscr{R}_a > \beta$ **THEN**
      $\mathbf{x}_k = \mathbf{x'}_k$
    END IF
  END FOR
END IF

Table 2 MCMCPFv1 tracking strategy [27]

$[\{\mathbf{X}_k^i\}]^N = \text{MCMCPFv1}(\mathbf{X}_{k-1}, z_k)$

IF $k=0$, **THEN** Set status $r_{k-1}^t = Dead$ for each target t **END IF**
IF $k \geq 1$ **THEN** {}
  $x_{burnin} = \text{MCMC}(\mathbf{X}_0, z_k)$ $(seeTable\ 1)$ $with\ N_{mcmc} = N_{burnin}$.
  FOR $i = 1,\ldots,N$, **DO**
    $x_k^i = \text{MCMC}(\mathbf{X}_{burnin}, z_k)$ $with\ N_{mcmc} = N_{interval}$.
  END FOR
  Compute the MAP estimator $E_{p(\mathbf{X}_k|z_{1:k})}[\mathbf{X}_k] = argmax_{\mathbf{X}_k^i}[count(x_k^i)]$ to approximate the posterior $p(\mathbf{x}_k|z_{1:k})$
END IF

4.2 MCMC-based particle filtering (MCMC-PF)

Khan et al. [27] proposed a mixed Markov Chain and Particle Filtering (MCMC-PF) algorithm, which we will henceforth call MCMCPFv1.

MCMCPFv1 commences by running a number of burn-in MCMC iterations, so that the Markov chain converges before it is used for particle sampling (Step #3, Table 2). The initial state x_k^m is chosen randomly from the set of particles of instant $k-1$. Once the requisite number of iterations (N_{burnin}) has passed, a particle is chosen every N_{interval} particles, until the desired number of particles ($N_{\text{particles}}$) has been reached. Recall that these intervals are necessary because the MCMC process is modifying only a single target at a time, as mentioned in Sect. 4.1. Once all the particles have been drawn, the MAP estimate corresponds to choosing the particle with the highest repeat count.

The addition of the Particle Filtering step to the MCMC core renders MCMCPFv1 less vulnerable to "unlucky" draws, in which an extremely low threshold is drawn in the last few iterations that allow a low-probability state to be accepted without having enough iterations left to recover.

It should be noted that, in its original implementation, MCMCPFv1 does not incorporate any detector data, instead relying on knowledge of the entry point of any new targets and having a very discriminant appearance model for targets. As MCMCPFv1 was intended to track ants on a white surface where there is a single entry point with closed borders, such simplification is possible. On a human tracking situation where the borders are open, this becomes more difficult,

and for this evaluation it has been implemented as a uniform entry field along the left, right and lower borders of the image (since the upper border is closed, and there are no off-border entry points). Therefore, we consider a variation annotated MCMCPFv2, where the drawing for new targets incorporates detector information, using the detectors outlined in Sect. 5. As in the case of MCMC, MCMC-PF is an inherently sequential process, which makes it poorly suited to optimization via parallelization.

4.3 Our filtering strategy (HYBRID)

Our filtering scheme, called HYBRID from here on and detailed in Table 3, combines Markov chain iterations and principles of the CONDENSATION algorithm [24].

Multiple saliency maps $\{S_k^d\}_{d=1,\ldots,N_\mathrm{d}}$ are processed from visual detector outputs $\mathrm{Det}^d(z_k)$ and from the targets' dynamics $p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j)$. These maps, independent and so computed in parallel on multi-core processing units, are then merged in a unified saliency map (step #5) as illustrated in Fig. 3, which highlights "relevant" areas of the state space. The underlying data driven proposal densities and particle sampling mechanisms will (re)-concentrate the particles on the right regions of interest. The calculation of the saliency maps is independent of both the number of targets and number of particles, depending only on the detector's runtime and the number of detectors.

In step #7, the continuous parameters $(\mathbf{x}_k^{i,1}, \ldots, \mathbf{x}_k^{i,N_\mathrm{t}})$ of the ith particle are randomly drawn from a uniform distribution, and the values for these positions in the saliency map lead to a likelihood $\sum_{j=1}^{N_\mathrm{t}} S_k(u_k^j,\ v_k^j)$ which allows to accept/reject the sample based on rejection sampling mechanism. This involves simply looking up the value in the saliency map S_k, and permits: (1) the reduction of the computation cost, (2) an efficient sampling in the high likelihood areas of the continuous state-subspace and so a drastic limitation of the particle burst, (3) an easier implementation of the proposal density where this is difficult to model analytically[3].

This first sampling ignores possible target jumps, i.e. we assume the tracked targets remain the same from one frame to the next. In step #12, a second sampling step through MCMC moves leads each individual particle from the current targets' configuration to another one (based on likely jumps) which is more representative of the current image contents. Drawing new samples by moving jointly all the dimensions suffers from exponential complexity with the state-space dimensionality [37]. The popular resolution is to propose target-wise marginal moves, i.e. only moving one target at each iteration. The transition from state hypothesis $\mathbf{X}_k$ to the proposed next $\mathbf{X'}_k$ is conditioned by a proposal density $q(\mathbf{x'}_k|\mathbf{x}_k, z_k)$ for each jump in the targets' status.

Traditionally, an MCMC process requires a high number of burn-in iterations as the Markov chain must usually, given an initial state, move between trans-dimensional state-spaces. The iteration number N_mcmc, whose role is to increase diversity in the particle swarm, is here reduced drastically as: (1) we only have to deal with targets' configuration as the continuous component subspace has been already sampled, (2) the associated target number does not change significantly from one frame to the next, and (3) we consider Markov chain moves for each individual particle and the resulting particle swarm induces diversity itself. A last observation concerns the low computation cost of each iteration since it changes the computation of the acceptance ratio $\mathscr{R}_a$ into a lookup into the saliency map S_k, instead of a more expensive operation, e.g. the comparison of color distributions.

Step #21 corresponds to the particle weighting update. To this end, we assign each particle $\mathbf{X}_k^i$ a weight w_k^i involving its likelihood $p_2(z_k|\mathbf{X}_k^i)$. This likelihood involves three different calculations, depending on the target status. If a target is $Dead$ or $Lost$, it is calculated as a similarity measure to the background (i.e., likelihood the target *is not* present); if the target is New, it is a dissimilarity measure to the background (i.e., likelihood the target *is* present); finally, if the target is $Tracked$, it is calculated as both dissimilarity to the background, and similarity to an adaptive per-target appearance model (i.e., likelihood the target *is* present *and* is similar in appearance to what we have seen in earlier frames).

Finally, the particle with the highest weight w_k^i is chosen as the most probable configuration of the system, and is used as input to the dynamic model for the next frame.

5 Implementation issues

Recall that we aim to combine several (complementary) detector outputs and associated saliency maps denoted $\{S_k^d(\mathbf{x}|z_k)\}_{d=1,\ldots,N_\mathrm{d}}$ in Table 3. We focus hereafter on two state-of the-art detectors based, respectively, on motion and human appearance. Such detectors provide a more robust coverage of the targets: the motion detector reliably detects moving targets, while the person detector is able to find people that can be either mobile or static in the video stream. Our primary motivation is to illustrate the data fusion capabilities and impact of the saliency map in the tracker; the principle can be straightforwardly extended to a larger bank of detectors.

The implemented motion detector is based on the classical background subtraction where the background is represented by an adaptive Mixture of Gaussians models (MGM) [39] where each pixel is modelled by multiple Gaussians. This well-known MGM detector is not detailed here for space reasons; it has just been extended to handle colour images,

[3] Even if only Gaussian mixtures are considered in this work.

Table 3 Our filtering strategy(HYBRID)

$[\{\mathbf{X}_k^i, w_k^i\}]_{i=1}^N = \text{HYBRID}(\mathbf{X}_{k-1}, z_k)$

IF $k = 0$, **THEN** Set status $r_{k-1}^t = Dead$ for each target t **END IF**

IF $k \geq 1$ **THEN** {}

  Generate saliency maps $\{S_k^d(\mathbf{x}|z_k)\}_{d=1,\ldots,N_d}$ for each visual detector $Det^d(z_k)$

  Generate a saliency map from the targets' dynamics $S_k^{dyn}(\mathbf{x}) = \sum_{j=1}^{N_t} p(\mathbf{x}^j|\mathbf{x}_{k-1}^j)$

  Generate the unified saliency map $S_k(\mathbf{x}|z_k) = S_k^{dyn}(\mathbf{x}) + \sum_{d=1}^{D} S_k^d(\mathbf{x}|z_k)$

  FOR $i = 1, \ldots, N_p$, **DO**

    FOR $j = 1, \ldots, N_t$, **DO**

      IF $r_k^j == Tracked$ **THEN**

        Sample $\mathbf{x}_k^{i,j}$ from S_k *via* rejection sampling

      END IF

    END FOR

    FOR $m = 0, \ldots, N_{mcmc}$, **DO**

      Randomly pick a target j and propose an associated event/jump τ (events have all the same probability of happening)

      Apply τ to $\mathbf{X}_k^i$ to generate $\mathbf{X'}_k^i$ with update of the j-th target's status.

      Draw threshold β according to a uniform distribution over $(0, 1]$

      Calculate the acceptance ratio $\mathscr{R}_a = \frac{p_1(z_k|\mathbf{x'}_k^i).q(\mathbf{x}_k^j|\mathbf{x'}_k^i, z_k).\Psi(\mathbf{x}_k^j, \mathbf{x}_k^m)}{p_1(z_k|\mathbf{x}_k^i).q(\mathbf{x'}_k^i|\mathbf{x}_k^i, z_k).\Psi(\mathbf{x'}_k^j, \mathbf{x'}_k^m)}$

      IF $\mathscr{R}_a > \beta$ **THEN**

        $\mathbf{x}_k^i = \mathbf{x'}_k{}^i$

      END IF

    END FOR

    Update the weight w_k^i associated to $\mathbf{X}_k^i$ according to $w_k^i \propto p_2(z_k|\mathbf{X}_k^i)$, prior to a normalization step so that $\sum_i w_k^i = 1$

  END FOR

  Compute the MAP estimator $E_{p(\mathbf{X}_k|z_{1:k})}[\mathbf{X}_k] = argmax_{\mathbf{X}_k^i}[w_k^i]$ to approximate the posterior $p(\mathbf{x}_k|z_{1:k})$

  Remove all targets j where $\{r_k^j\}_{j=1,\ldots,N_t} == Lost$, and set $\{r_k^j\} = Tracked$ for all targets where $r_t^j == New$ in $E_{p(\mathbf{X}_k|z_{1:k})}$

END IF

Table 4 Comparative studies of visual person detectors

Visual person detection results			
Detector	Recall (%)	Precision (%)	Average time per frame (s)
HOGs+latent SVM [17]	64.0	95.7	1.4
HOGs+Adaboost [28]	49.8	89.2	2.5
HOGs+linear SVM [15]	70.1	99.5	0.1

The experiments are carried out on an Intel(R) Core(TM)2 Duo CPU T8100 @ 2.1 GhZ with 3 GB of RAM

which is trivial. Figure 4c shows an example of foreground segmentation by this detector.

For person detection based on appearance, three visual person detectors, namely that of Felzenszwalb et al. [17], Dalal et al. [15], and Laptev [28], have been evaluated on our proprietary dataset introduced in Sect. 6. All the three evaluated detectors consider person detection as part of a general object detection problem. To highlight the similarities and differences of the three detectors, it is best to consider their candidate generation, feature set, and utilized classifier separately. All of them use an exhaustive sliding window approach with fixed aspect ratio in image scale-space for candidate generation. Dalal et al. [15] use Histogram of Oriented Gradients (HOGs) as feature sets and a linear SVM as a classifier to detect full human bodies. Similarly, Felzenszwalb et al. [17] use HOGs with analytically reduced dimension as features and a latent-SVM as a classifier with a parts-based approach. Laptev [28] uses weighted local HOGs in

Fig. 2 Sample detection outputs from Felzenszwalb et al. [17] (*top*), Dalal et al. [15] (*middle*), Laptev [28] (*bottom*)

all rectangular sub-windows of the target as features and the AdaBoost framework to select prominent features for full human body detection.

Source codes provided by the authors, using their default parameter settings and trained classifiers, with the exception of [17] for which our complete C implementation of the original Matlab version, are used for evaluation on our proprietary dataset composed of 958 image frames. To quantify the person detection performance, true positives (TP), false positives (FP), and false negatives (FN) are counted in each frame. Based on the counts, Recall (true positive rate) and Precision are computed as shown in Eq. 1. Table 4 summarizes the obtained results; sample images corresponding to each detector output are shown in Fig. 2. The latter shows some examples of successful detections and false alarms where local image patches look like humans. All the detectors run with high precision rates and acceptable rates without any specific training for this corpus. Going a step further, Dalal et al.'s visual person detector outperforms the other two with a higher detection rate as well as precision. Computationally, Dalal et al.'s detector is faster with a 14-fold improvement compared to [17]. Therefore, given our setting, we have retained Dalal et al.'s person detector as a detector due to its superior true detection rate, precision and computation time.

$$\text{Recall (in \%)} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \ \text{Precision (in \%)} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

The generation of the saliency map involves both the detector output for all detectors and the dynamic motion model for the targets. The only assumption made as to the data input by each part is that it is in the form of a probability distribution throughout the image, i.e. each point has a value in the [0, 1] range that indicates the probability assigned by the detector or the dynamic motion model that the target is present. These probability distributions are then combined through a weighted sum into a complete saliency map for the image, as seen in Fig. 3.

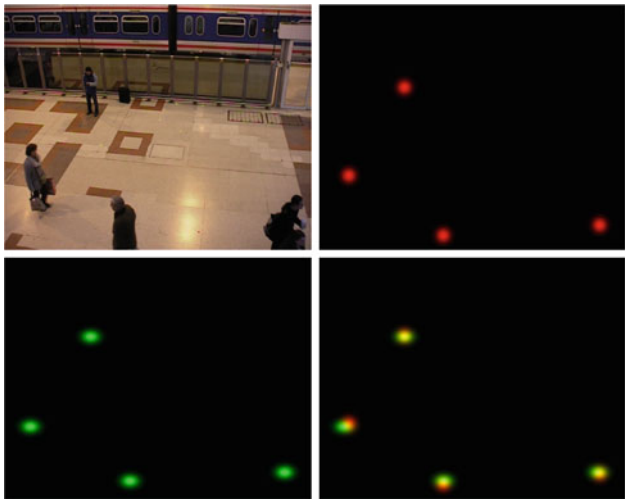

Fig. 3 Saliency map generation: original image on the *upper left*; saliency generated by a detector on the *upper right*; saliency generated by the dynamic model on the *lower left*; and combined saliency map on the *lower right*

Given the previous notations, the joint state vector to estimate in the tracker is composed of a variable number of N_t targets and so follows:

$$\mathbf{X}_k = \left\{ (r_k^1, u_k^1, v_k^1, s_k^1), \ldots, (r_k^{N_\text{t}}, u_k^{N_\text{t}}, v_k^{N_\text{t}}, s_k^{N_\text{t}}) \right\} \tag{2}$$

To compare our data driven proposal-based filter against the state of the art, we also implemented the standard MCMC-PF pioneered by Khan et al. [27] and extended the approach to take into account visual measurements, namely two conventional visual detector outputs. The first uses background subtraction via a Multiple Gaussian Mixture model in the vein of [39] but extended to color images; the second is a standard person detector based on HOG and SVM classification [15]. Let B be the number of detected regions for a given detector $\text{Det}^d(z_k)$ and $\mathbf{p}_i = (u_i, v_i), i = 1, \ldots, B$, the centroid coordinate of each such region. The associated saliency map $S_k^d(\mathbf{x}_k|z_k)$ (step #3) follows the Gaussian mixture proposal (although, as remarked earlier, it can accommodate more complex distributions, even if they are not easy to describe analytically)

$$S_k^d(\mathbf{x}|z_k) = \sum_{i=1}^{B} \mathcal{N}(\mathbf{x}; \mathbf{p}_i, \text{diag}(\sigma_{u_i}^2, \sigma_{v_i}^2)) \tag{3}$$

An example of a particle swarm drawn from these saliency maps is shown in Fig. 4. As can be seen, the particles naturally concentrate in areas where the saliency map shows a high probability of target presence given the outputs of the above detectors.

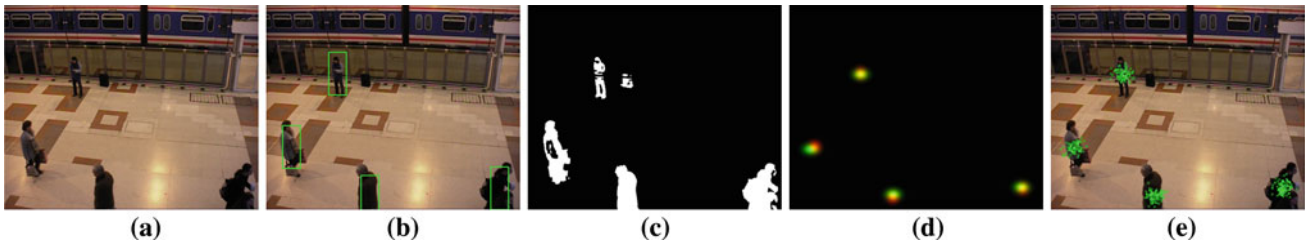

Fig. 4 Detection outputs: Original image (**a**), HOG+SVM (**b**) and MGM (**c**), associated unified saliency map (**d**) and particle (*green dots*) sampling (**e**)

In the vein of [46], the likelihood function $p_2(z_k|\mathbf{X}_k)$ involved in step #21 favors both the difference to the background and the similarity with its correspondence in the previous frames. Given a state $\mathbf{X}_k$, we partition the image into different regions corresponding to the targets and the background. We denote R_k^j as the region (an elliptic mask) of a target j defined by $\mathbf{x}_k^j$. This function is evaluated within the entire target region $R_k = \bigcup_{j=1}^{N_t} R_k^j$. Let h_{back}^j be the color histogram of the background image within the target mask j, $h_{\mathbf{x}}$ and h_{ref}^j be the color histograms corresponding to a region R_x parameterized by the state $\mathbf{X}_k$ and to the current appearance model of the jth target[4]. These multiple color distributions provide some level of person discrimination from the clothes appearance and so limit drastically the targets' label switching during the tracking process. To overcome the appearance changes of the region R_k^j in the video stream, the associated target reference model is updated at time k from the computed estimates through a first-order filtering process i.e.

$$h_{\text{ref},k}^j = (1-\kappa).\, h_{\text{ref},k-1}^j + \kappa \cdot h_{E[\mathbf{x}_k]}^j, \tag{4}$$

where κ weights the contribution of the mean state histogram $h_{E[\mathbf{x}_k]}^j$ to the target model $h_{\text{ref},k-1}^j$ of the target j. To reflect the similarity of two N_{bi}-normalized histograms h_1 and h_2, we use the Bhattacharyya distance:

$$B(h_1, h_2) = \sum_{l=1}^{N_{\text{bi}}} \sqrt{h_{1,l} \cdot h_{2,l}} \tag{5}$$

The likelihood function in step #21 of the HYBRID strategy follows $p_2(z_k|\mathbf{X}_k) = \Pi_{j=1}^{N_t} w_k^{i,j}$ with

$$w_k^{i,j} = \begin{cases} 1 - \exp\{\lambda_1 B(h_{\text{back}}, h_{\mathbf{x}_k^j})\} & \text{if } r_j == New \\ \exp\{\lambda_2 B(h_{\text{ref}}, h_{\mathbf{x}_k^j})\}/(1 - \exp\{\lambda_1 B(h_{\text{back}}, h_{\mathbf{x}_k^j})\}) & \text{if } r_j == Tracked \\ (1 - \exp\{\lambda_2 B(h_{\text{ref}}, h_{\mathbf{x}_k^j})\}) * \exp\{\lambda_1 B(h_{\text{back}}, h_{\mathbf{x}_k^j})\} & \text{if } r_j == Lost \end{cases} \tag{6}$$

while in MCMCPFv1 and MCMCPFv2 they are used to calculate the term $p_1(z_k|\mathbf{X}_k)$ in the MCMC acceptance ratio, namely $p_1(z_k|\mathbf{X}_k) = p_2(z_k|\mathbf{X}_k)$ for a fair comparison with our algorithm HYBRID. Note that only the target moves in the current MCMC iteration is used in this computation, as opposed to HYBRID, which uses all the targets.

As for the proposal density $q(\mathbf{x'}_k|\mathbf{x}_k, z_k)$, there are two cases to take into account: in MCMCPFv1, this density is conceptually equal to sampling from a saliency map that is composed solely of the dynamic model probabilities, while in MCMCPFv2 and HYBRID, it is equivalent to sampling from a mixed saliency map created by adding the detector data to the dynamic model (in the case of HYBRID, this is exactly how it is implemented). As a result, in MCMCPFv1 this density is built such that $q(\mathbf{x'}_k|\mathbf{x}_k) == q(\mathbf{x}_k|\mathbf{x'}_k)$, i.e., the reversal for any given status change is equiprobable w.r.t. that status change. In MCMCPFv2 and HYBRID, the addition of detector data makes that equality impossible, and so $q(\mathbf{x'}_k|\mathbf{x}_k, z_k)$ turns into a sampling of the probability values at that point (which, in HYBRID, means a lookup into the saliency map).

An interaction function $\Psi(\mathbf{x}_k^j, \mathbf{x}_k^m)$ also intervenes in the calculation of acceptance function $\mathscr{R}_a$. This interaction function models the interaction between targets that are spatially close, in order to avoid having multiple targets tracking the same object. Similar to Khan et. al. [27], a Markov Random Field (MRF) is adopted to address this. A pairwise MRF where the cliques are restricted to the pairs of nodes (targets define the nodes of the graph) that are directly connected to the graph is implemented as part of our tracker.

Regarding the likelihood function in the MCMC acceptance ratio, our algorithm HYBRID uses the already computed saliency map which results in

$$p_1(z_k|\mathbf{X}_k^i) = \prod_{j=1}^{N_t} p_1(z_k|\mathbf{x}_k^{i,j}) \tag{7}$$

with

[4] In fact two histograms to represent the appearance of the upper and lower human body.

$$p_1(z_k|\mathbf{x}_k^{i,j}) = \begin{cases} S_k(\mathbf{x}_k^{i,j}|z_k) & \text{if } r_k^j == New \text{ or } r_k^j == Tracked \\ 1 - S_k(\mathbf{x}_k^{i,j}|z_k) & \text{if } r_k^j = Lost \end{cases} \quad (8)$$

giving us a much faster computation of the acceptance ratio. This allows HYBRID to use MCMC to evolve each particle separately.

The comparative studies are performed for the empirically designed filtering strategies parameter values listed in Table 5. They are tuned empirically on a small subset of the video sequence. The program is written in C++ using OpenCV functions.

Finally, some consideration is given to processing time: in order to compare both algorithms fairly, the comparison is set up in a way that both algorithms consume similar computational resources. All strategies are normalized with respect to the number of likelihood evaluations, which is the most time consuming part. MCMCPFv1 and MCMCPFv2 perform exactly one likelihood evaluation per MCMC iteration, while HYBRID performs one such computation per particle and target present in the current image. Therefore, MCMCPFv1 and MCMCPFv2 will use $\bar{N}_t$ times more iterations than HYBRID does particles, where $\bar{N}_t$ is the average number of targets per image in the test sequence.

6 Comparative studies and discussion

6.1 Datasets and methodology

We have tested our three strategies on a public dataset, namely the sequence taken by camera 3 in the S7 dataset of *PETS* 2006[5]. We have also evaluated the HYBRID algorithm on our own dataset, composed of images acquired in a robotic lab hall. In either case, the sequences are captured with a stationary camera, mounted a few meters above the ground oriented towards the floor.

[5] See the URL www.cvg.cs.rdg.ac.uk/slides/pets.html.

The sequence from the PETS dataset is composed of 3,400 images (size 720×576 pixels), which consist of one or more people (up to 6), generally with similar clothing. The targets are seen walking alone or together across a train station, passing each other, and meeting at the center of the scene. This makes correct tracking very challenging.

The proprietary dataset, which features an in-lab hall, is composed of 1,800 images at a resolution of 640×480, in which up to five targets move in an area with a heavily cluttered background. The clothing of the people involved is more distinctive than in the PETS sequence, which, along with the small number of targets to track and somewhat more simplistic interactions, is easier to perform accurate tracking.

For a quantitative assessment, we annotated every frame of the two sequences manually. We marked all image location with 2D bounding box in which a person is visible. We then derived similar bounding boxes from the tracker and compared them to the annotations. Following recent tracking evaluations, we consider a box as correct if it overlaps with the ground-truth annotation by more than 50 % using the intersection-over-union criterion like in [16].

To compare the algorithms, we derive three metrics in the vein of the CLEAR MOT metrics [7]:

- *False positive rate per image (FPR)* average false positives per image, i.e. $\frac{1}{K}\sum_{k,j}\delta_{k,j}$ where $\delta_{k,j} = 1$ if a target j is tracked in the k th frame where there is none, and 0 otherwise.
- *Tracking success rate (TSR)* ratio between correctly tracked targets per frame and the actual amount of targets per frame, i.e. $\frac{1}{J}\sum_{k,j}\delta_{k,j}$ where $\delta_{k,j} = 1$ if target j is correctly tracked in frame k, and 0 otherwise.
- *Precision error (PE)* measures how precisely the targets are tracked, as the sum of the squared error from the position given by the tracker to the one in the ground truth i.e. $\sqrt{\frac{1}{J}\sum_{k,j}(\delta'_{k,j}.\delta_{k,j})}$, with $\delta_{k,j} = \mathbf{x}_k^j - \mathbf{x}_{k,\text{truth}}^j$.

The afore-mentioned metrics are computed by averaging over a number of runs, for each sequence, to account for their stochastic nature. Consequently, we have also included

Table 5 Parameter values used in the filtering strategies HYBRID, MCMCPFv1, and MCMCPFv2

Symbol	Meaning	Value (HYBRID)	Value (MCMCPFv1, MCMCPFv2)
N	Number of particles	150	10
$(\sigma_u, \sigma_v, \sigma_s)$	Standard deviation in random walk models	(64, 48, 1)	(64, 48, 1)
N_{bi}	Number of color bins in the Bhattacharyya distance	512	512
N_{d}	Number of involved visual detectors	2 i.e. [15,39]	2 i.e. [15,39]
N_{mcmc}	Number of MCMC iterations	3 per particle	150
N_{burnin}	Number of burn-in iterations	0	225
κ	Color learning factor	0.15	0.15

the standard deviation σ for each metric to give an idea of repeatability.

As previously mentioned, in order to put all algorithms in a level playing field w.r.t. processing resources, the HYBRID algorithm has been scaled down due to the fact that processing a HYBRID particle is more processor-intensive than processing a single MCMC iteration in MCMCPFv1 or MCMCPFv2. The scale-down factor corresponds to the average number of targets per frame $\bar{N}_t$, as mentioned, which in our case is 2.3 (there are 7,800 targets to be processed in 3,400 images, roughly). Rounding up to $\bar{N}_t = 2.5$, we have settled on $N = 150$ particles in HYBRID to $N_{mcmc} + N_{burnin} = 375$ iterations in MCMCPFv1 and MCMCPFv2.

6.2 Results and comparative study

Table 6 shows the results of all three strategies in the *PETS* sequence mentioned earlier. Some snapshots of this sequence are also shown in Fig. 5. This figure shows snapshots of the video where people are coming in and out. The tracker quickly detects a new person in the scene, then immediately assigns particles to this target and starts tracking it. The entire video for the HYBRID strategy has been submitted as supplementary material but can also be found at the URL http://homepages.laas.fr/aamekonn/videos.htm.

It can be seen that all methods work reasonably well on the video corpus. More specifically, MCMCPFv1 has a slight edge over both MCMCPFv2 and HYBRID when it comes to false positives. However, both MCMCPFv2 and HYBRID show better results in target tracking and precision, as expected. Clearly, the addition of more precise information in the form of detector data aids successful tracking.

The performance measures confirm that HYBRID outperforms both MCMCPFv1 and MCMCPFv2 when it comes to correctly tracking targets. This can be explained by (1) the greater variety of possible hypotheses to choose from because of the evolution of the whole particle cloud, as opposed to a single particle, and (2) the more precise placing of the said hypothesis by taking advantage of the saliency maps. It also contributes to its being slightly ahead of MCMCPFv2 in terms of consistency of its results, as the lower σ values attest. Consequently, HYBRID is also more precise than both MCMCPFv1 and MCMCPFv2.

The main factor contributing to MCMCPFv1's less accurate tracking performance is that a lost target can only be re-initialized near the entry point, and will thus remain lost until the target moves close to said entry point (in this case, the image border). This can be seen in the first samples of Fig. 6, where two targets have just crossed each other. MCMCPFv2 performs better and does not suffer from this problem, so it recovers just a frame later, but HYBRID still recovers faster. This is likely because of the higher variety between the particles, which allows HYBRID to explore the solution space faster.

As a complementary test, we have run HYBRID on another sequence, taken in an open space of our laboratory, in which the color characteristics of the people in the scene are more easily differentiable. The results are shown in Table 6,

Table 6 MOT results on datasets *PETS* 2006 and in-lab sequences

Dataset	FPR (σ)(%)	TSR (σ)(%)	PE (σ) pixels
PETS (MCMCPFv1) [27]	2.6 ($8.3.10^{-4}$)	68.9 (1.80)	10.17 (0.965)
PETS (MCMCPFv2)	3.0 ($1.2.10^{-3}$)	72.2 (0.52)	8.65 (1.217)
PETS (HYBRID)	2.3 ($2.2.10^{-3}$)	77.9 (0.35)	7.62 (1.136)
In-lab (HYBRID)	0.1 ($3.53.10^{-5}$)	85.27 ($2.73.10^{-3}$)	8.15 (1.119)

Fig. 5 From *top-left* to *bottom-right* sample results from PETS dataset for the HYBRID algorithm

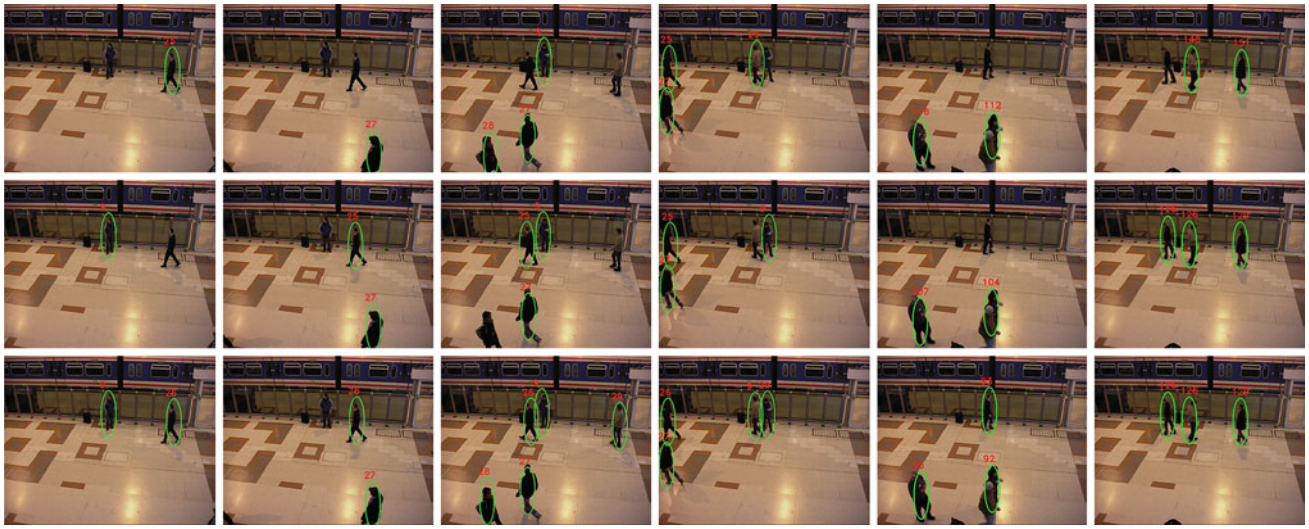

Fig. 6 Sample results from PETS dataset: the three lines correspond to frames 1,424, 1,450, 1,485, 1,554, 2,369, and 2,527 as a result of MCMCPFv1, MCMCPFv2 and HYBRID from *top to bottom*. In frame #2369, the target on the *upper side* is not too dissimilar to the background, and so MCMCPFv1 and MCMCPFv2 have not been able to follow it correctly. In frame #2527, the targets have just crossed, and neither MCMCPFv1 nor MCMCPFv2 has recovered the lost target yet. MCMCPFv2 will recover a frame later, MCMCPFv1 will not recover at all

Fig. 7 Sample images from a run of HYBRID on our own dataset, showing particle locations and results. The differences in the shirt colors (*red*, *black* and *grey*) aid in the tracking, although occlusion remains a problem

and some images from this sequence can be seen in Fig. 7. The images show how the particles are sampled and the results it leads to. It is worth noting that the great majority of particles are tightly clustered around the location of the target, as a result of the saliency map and rejection sampling. As can be seen, the results are markedly better, particularly on the tracking success rate. In all likelihood, this is due to the better discriminating power of the clothes' colors, as well as the more simplistic interactions between targets. Similarly, the slight increase in position error may be attributed to a difference in perspective allowing slightly higher vertical movement without being penalized by the appearance model.

6.3 Discussion about the computational cost

The tests in the previous sections have been performed on a PC with a 2 GHz processor and 4 GB RAM, where it performs at a frequency of 1.5 fps without any optimization. SystemC simulations of a straight-forward integration of the algorithm in a Spartan 3 FPGA result in a frequency of 18 fps, once again without optimization. These results are presumed accurate in a tentative manner by implementing part of the algorithm deemed to be complex (hence, representative of the whole flow), and yet simple enough that the implementation effort is not prohibitive [47]. However, by taking advantage of the parallel nature of the algorithm itself, the execution time can be substantially improved.

To begin with, the execution time per frame can be defined as in Eq. (9), where T_{frame} is the time spent per frame, which is obtained via the time spent in the different detectors ($T_{\text{detectors}}$), the time spent generating the saliency maps (T_{saliency}), the time to process all the particles ($T_{\text{particles}}$, which in the unoptimized case is $N \times T_{\text{particles}}$), and finally the time required to select the most probable particle (T_{MAP}). These times correspond, respectively, to steps #3, #4 and #5, #6 to #22, and #23 in algorithm 3.

$$T_{\text{frame}} = T_{\text{detectors}} + T_{\text{saliency}} + T_{\text{particles}} + T_{\text{MAP}}. \tag{9}$$

Of these terms, T_{saliency} depends only on the size of the image and the number of detectors involved, while T_{MAP} depends on the number of particles the system uses for tracking. Furthermore, the time spent on these two steps can be considered negligible compared to the other two. Therefore,

the two best candidates for optimization are $T_{\text{detectors}}$ and $T_{\text{particles}}$, which take up around 95 % of the processing time, distributed roughly evenly. $T_{\text{detectors}}$ depends on the implementation of the chosen detectors and is independent of the tracking algorithm itself, so we will set it aside and examine $T_{\text{particles}}$ more closely.

As mentioned before, in an implementation that takes no advantage of the parallel nature of the algorithm, $T_{\text{particles}}$ equals the time spent processing a single particle times the number of particles (N). However, as the processing (sampling and evaluation) of each particle is independent of all other particles, it is possible to use multiple particle processing units to drastically reduce the time necessary to process all particles.

In practical terms, the architecture necessary for such implementation in an FPGA is shown in Fig. 8. This architecture takes advantage of the built-in dual-port block memory (BRAM) present in many modern FPGAs to replicate the necessary data to avoid collisions in the memory access, which proved to be the bottleneck in the SystemC simulations. As can be seen in the figure, each pair of particle processors needs 3 BRAM blocks: 2 of these are read-only, for the saliency map and the appearance reference data for the targets; the remaining block, which the processors write their output to, is shared between the two processors, with each processor using a separate area of that memory.

With this architecture, $T_{\text{particles}}$ can be substantially reduced, as far as the resources available in the FPGA allow, up to the limiting case of $T_{\text{particles}}$ (in the case where there are as many particle processors as there are particles). In reality, this improvement is subject to the law of diminishing returns, with just eight particle processors already reducing $T_{\text{particles}}$ by 87%, which reduces the total processing time by 43 %.

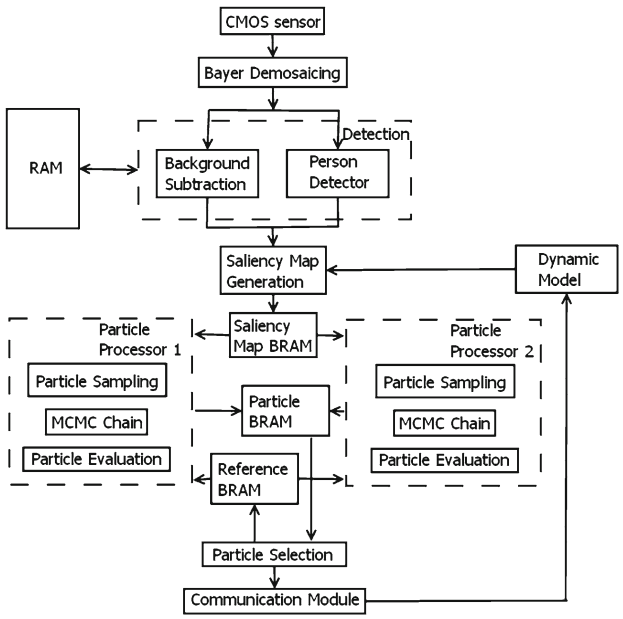

Fig. 8 Hardware architecture with parallel particle processing. The number of particle processing units can be increased as long as FPGA resources allow

7 Conclusion and future works

In this paper, we have proposed a novel MCMC-PF algorithm for multiple person tracking intended for implementation in a FPGA-based smart camera. The principal distinction of our approach from standard MCMC-PF is twofold. First, we sample the particles in high probability areas of the high dimensional state-space thanks to short Markov chains devoted to each individual particle. Second, this sampling step is driven by saliency maps from multiple person detector outputs and a rejection sampling algorithm to limit the well-known burst in terms of particles and MCMC iterations.

We have compared this algorithm with another similar algorithm, showing that it outperforms both the original form and a variant with added detectors in terms of (1) tracking performances, and (2) parallelization capabilities as the standard MCMC-PF processes particles sequentially.

Current investigations concern extending the algorithm using knowledge of the camera perspective model and the assumption that motion is on a known plane; this allows us to make inferences in 3D and account for changes in image due to perspective effects. Our mid-term research goal deals with the effective algorithm implementation in a manner suitable for an FPGA-based intelligent camera, which will greatly improve performance by taking advantage of the parallelizable nature of the algorithm. Our long-term research concerns the use of such multiple communicating smart camera nodes to reliably track people across large scale human-centered environments. The major challenge is to avoid sending many full-resolution, real-time images to the video processor of the networked PCs, by offloading the processing power into the cameras themselves.

References

1. Delta Technologies Society (DTSO), Toulouse. http://www.delta-technologies.fr
2. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: International Conference on Computer Vision and Pattern Recognition (CVPR'08). Anchorage, Alaska (2008)
3. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. Trans. Signal Process. **2**(50), 174–188 (2002)
4. Avidan, S.: Ensemble tracking. Trans. Pattern Anal. Mach. Intell. (PAMI'07) **29**(2), 261–271 (2007)
5. Bar-Shalom, Y., Jaffer, A.: Tracking and Data Association. Academic Press, San Diego (1998)
6. Bardet, F., Chateau, T., Ramasadan, D.: Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous

tracking and classification. In: International Conference on Computer Vision (ICCV'09), Kyoto, Japan (2009)
7. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. J. Image Video Process. **2008**, 1 (2008)
8. Berzuini, C., Gilks, W.: RESAMPLE-MOVE filtering with cross-model jumps. Series Statistics For Engineering and Information Science. Springer, New York (2000)
9. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: International Conference on Computer Vision (ICCV'09), Kyoto, Japan, pp. 1515–1522 (2009)
10. Cai, Y., De Freitas, N., Little, J.: Robust visual tracking for multiple targets. In: European Conference on Computer Vision (ECCV'06), Graz, Austria (2006)
11. Chang, C., Ansari, R., Khokhar, A.: Multiple object tracking with kernel particle filter. In: International Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, USA (2005)
12. Cho, J., Hun Jin, S., Dai Pham, X., Jeon, J.: Multiple object tracking circuit using particle filters with multiple features. In: International Conference on Robotics and Automation (ICRA'07), , Roma, Italy, pp. 4639–4644 (2007)
13. Choi, W., Savarese, S.: Multiple target tracking in world coordinate with single, minimally calibrated camera. In: Computer Vision—ECCV 2010, pp. 553–567 (2010)
14. Choudhury Verma, R., Schmid, C., Mikolajczyk, K.: Face detection and tracking in a video by propagating detection probabilities. Trans. Pattern Anal. Mach. Intell. (PAMI'03) **25**(10), (2003)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: International Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, USA (2005)
16. Everingham, M., others (34 authors): The 2005 PASCAL visual object classification challenge. In: PASCAL Challenges Workshop, Graz, Austria (2006)
17. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. Trans. Pattern Anal. Mach. Intell. (PAMI'09) **99**(1) (2009)
18. Gabriel, P., Verly, J., Piater, J., Genon, A.: The state of the art in multiple object tracking under occlusion in video sequences. In: International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS'03), Ghent, Belgium (2003)
19. Gatica-Perez, D., Odobez, J., Ba, S., Smith, K., Lathoud, G.: Tracking people in meetings with particles. In: Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'05), Montreux, Switzerland (2005)
20. Grabner, H., Bischof, H.: On-line boosting and vision. In: International Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, USA (2006)
21. Green, P.: Reversible jump Markov Chain Monte Carlo computation and bayesian model determination. Biometrika **82**(4), 711–732 (1995)
22. Gumpp, T., Azad, P., Welke, K., Oztop, E., Dillman, R., Cheng, G.: Unconstrained real-time markerless hand tracking for humanoid interaction. In: International Conference on Humanoid Robots (HUMANOID'06), Genoa, Italy (2006)
23. Hue, C., Le Cadre, J., Pérez, P.: Sequential Monte Carlo methods for multiple target tracking and data fusion. Trans. Signal Process. **50**(2), 309–325 (2002)
24. Isard, M., Blake, A.: CONDENSATION—conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**(1), 5–28 (1998)
25. Isard, M., Blake, A.: ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In: European Conf. on Computer Vision (ECCV'98), London, UK, pp. 893–908 (1998)
26. Isard, M., MacCormick, J.: BraMBLe: a bayesian multiple blob tracker. In: International Conference on Computer Vision (ICCV'01), vol. 1, pp. 34–41. Vancouver, Canada (2001)
27. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. Trans. Pattern Anal. Mach. Intell. (PAMI'05) **27**(11), 1805–1818 (2005)
28. Laptev, I.: Improvements of object detection using boosted histograms. In: British Machine Vision Conference (BMVC'06). Edinburgh, UK, pp. 949–958 (2006)
29. Leibe, B., Schindler, K., Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil (2007)
30. Li, Y., Huang, C., Nevatia, R.: Learning to associate: hybridboosted multi-target tracker for crowded scene. In: International Conference on Computer Vision and Pattern Recognition (CVPR'09), Miami, USA (2009)
31. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. Int. J. Comput. Vis. (IJCV'00) **39**(1), 57–71 (2000)
32. Mc Kenna, S., Nait-Charif, H.: Tracking human motion using auxiliary particle filters and iterated likelihood weighting. Image Vis. Comput. (IVC'07) **25**(6), 852–862 (2007)
33. Van der Merwe, R., Doucet, A., De Freitas, J., Wan, E.: The unscented particle filter. Adv. Neural Inf. Process. Syst. **8**, 351–357 (2000)
34. Okuma, K., Taleghani, A., De Freitas, N.: A boosted particle filter: multitarget detection and tracking. In: European Conference on Computer Vision (ECCV'04), Prague, Czech Republic (2004)
35. Qu, W., Schonfeld, D., Mohamed, M.: Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. EURASIP J. Adv. Signal Process. (2007)
36. Rui, Y., Chen, Y.: Better proposal distributions: object tracking using unscented particle filter. In: Int. Conference on Computer Vision and Pattern Recognition (CVPR'01), Kauai, USA, pp. 786–793 (2001)
37. Smith, K., Gatica-Perez, D., Odobez, J.: Using particles to track varying numbers of interacting people. In: International Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, USA, pp. 962–969 (2005)
38. Songhwai, O., Russell, S., Sastry, S.: Markov chain monte carlo data association for general multiple-target. In: International Conference on Decision and Control (CDC'04), Atlantis, Bahamas, pp. 735–742 (2004)
39. Stauffer, C., Grimson, W.: Adaptative background mixture models for real-time tracking. In: International Conference on Computer Vision and Pattern Recognition (CVPR'99), Fort Collins, USA, vol. 2, pp. 22–46 (1999)
40. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. Trans. Pattern Anal. Mach. Intell. (PAMI'04) **26**(10), 1385–1391 (2004)
41. Vermaak, J., Doucet, A., Pérez, P.: Maintining multi modality through mixture tracking. In: International Conference on Computer Vision (ICCV'03), Nice, France (2003)
42. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially, occluded humans by Bayesian combination of edgelet based part detectors. Trans. Pattern Anal. Mach. Intell. (PAMI'07) **75**(2), 247–266 (2007)
43. Yang, C., Duraiswami, R., Davis, L.: Fast multiple object tracking via a hierarchical particle filter. In: International Conference on Computer Vision (ICCV'05), Beijing, China, pp. 212–219 (2005)
44. Yang, M., Lv, F., Xu, W., Gong, Y.: Detection driven adaptive multi-cue integration for multiple human tracking. In: International Conference on Computer Vision (ICCV'09), Kyoto, Japan, pp. 1554–1561 (2009)

45. Yu, T., Wu, Y.: Collaborative tracking of multiple targets. In: International Conference on Computer Vision and Pattern Recognition (CVPR'04), Washington, USA, pp. 834–841 (2004)
46. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. In: International Conference on Computer Vision and Pattern Recognition (CVPR'04), Washington, USA (2004)
47. Zuriarrain, I., Arana, N., Lerasle, F.: Implementation analysis for a hybrid particle filter on a FPGA based smart camera. In: International Conference on Computer Vision Theory and Applications (VISAPP'10), Angers, France (2010)

Author Biographies

Iker Zuriarrain received his M.Sc. degree in Computer Engineering at the University of Mondragon in 2006, and his Ph.D. in 2012, with a dissertation titled "Methods for the Implementation of Computer Vision Algorithms in FPGA-based Smart Cameras". He is currently working on safety-related software for railway applications.

Alhayat Ali Mekonnen received the B.Sc. degree in Electrical Engineering from Bahir Dar University in 2007, and an Erasmus Mundus M.Sc. degree in Computer Vision and Robotics from the VIBOT consortium (www.vibot.org), in 2010. He is currently a Ph.D. student in the Robotic, Action, and Perception (RAP) research group at LAAS-CNRS. His research focuses on detection and tracking of humans from mobile robots and surveillance cameras.

Frédéric Lerasle is an assistant professor at Paul Sabatier University since September 1997, and researcher at LAAS-CNRS in vision for robotics in Toulouse. His Ph.D. thesis was on human motion capture by multi-ocular vision at the LASMEA, graduating from Blaise Pascal University of Clermont-Ferrand in 1997. His current research at LAAS-CNRS concerns vision for robotics, more particularly: (1) detection, recognition, tracking of people, as well as interpretation of their gestures and activities for human-robot interaction, (2) landmark detection/recognition for metrical or topological navigation of mobile robots in indoor environments.

Nestor Arana received his "Ingeniería Superior en Sistemas Automáticos" (M.Sc.) degree from the University of Mondragon in 1997. In 2002 he received his Ph.D. degree from INSAT-Toulouse with a dissertation titled "Contributions à une méthodologie d'analyse pour l'implantation d'algorithmes sur FPGA, et au développement d'un multi-capteur 3D". Currently, Nestor is a lecturer and researcher at the Signal Theory and Communications Group of the University of Mondragon, focusing on image processing and the design of fast processing architectures based on co-design, high-level synthesis and reconfigurable hardware technologies.