

# Illumination Aware MCMC Particle Filter for Long-Term Outdoor Multi-Object Simultaneous Tracking and Classification

François Bardet, Thierry Chateau, Datta Ramadasan  
LASMEA, Université Blaise Pascal  
24 avenue des Landais, F-63177 Aubière cedex, FRANCE  
{bardet, chateau, ramadasan}@lasmea.univ-bpclermont.fr

## Abstract

*This paper addresses real-time automatic visual tracking, labeling and classification of a variable number of objects such as pedestrians or/and vehicles, under time-varying illumination conditions. The illumination and multi-object configuration are jointly tracked through a Markov Chain Monte-Carlo Particle Filter (MCMC PF). The measurement is provided by a static camera, associated to a basic foreground / background segmentation. As a first contribution, we propose in this paper to jointly track the light source within the Particle Filter, considering it as an additional object. Illumination-dependant shadows cast by objects are modeled and treated as foreground, thus avoiding the difficult task of shadow segmentation. As a second contribution, we estimate object category as a random variable also tracked within the Particle Filter, thus unifying object tracking and classification into a single process. Real time tracking results are shown and discussed on sequences involving various categories of users such as pedestrians, cars, light trucks and heavy trucks.*

## 1. Introduction

Real-time visual tracking of a variable number of objects is of high interest for various applications. In the recent years, several works have addressed multiple pedestrian and vehicle tracking [14]. In all these applications, real time may be needed either because an immediate information is required, or because recording images is not allowed, or because the amount of data is simply too huge to be recorded and processed later. Vision has been chosen as it offers a large measuring range, required by several surveillance applications: about 200 meters for traffic surveillance. Unfortunately, this benefit also causes deep object appearance scale changes. In addition, in traffic surveillance, target objects belong to various classes, such as pedestrians, cycles, motorcycles, light vehicles, light trucks, or heavy trucks.

The tracker is thus required to deal with various target 3D sizes, and with various target projection 2D sizes, due to heavy perspective effect.

In outdoor environment, shadows cast by opaque objects interfere with object segmentation and description. This decreases tracking accuracy as object estimate may be shifted towards its shadow. Moreover, it yields tracking failures as the tracker may instantiate a *ghost* candidate object upon a cast shadow. Both failures have been described in the literature, [10, 11] among others. However, shadows cast by objects also feature relevant information about object itself, offering the opportunity to increase its observability. For these two reasons, cast shadows have to be taken into account to improve visual tracking performance [11]. A survey and benchmark of moving shadow detection algorithms has been published in [10]. Nevertheless, segmenting the image into three classes (background, objects, shadows cast by objects) is a very challenging step, yielding authors to incorporate spatial and temporal reasoning into their segmentation methods.

Reversible Jump Markov Chain Monte-Carlo Particle Filter (*RJ MCMC PF*) has become a popular algorithm for real-time tracking of a varying number of interacting objects, as it allows to smartly manage object interactions as well as object enter and leave. The benefit of *MCMC PF* is that the required number of particles is a linear function of the number of tracked objects, when they do not interact. More computation is only required in case of object interaction (*i.e.* occlusion). This technique has been proposed and successfully used for jointly tracking up to 20 ants from a top view [5], or for tracking several pedestrians in a multi-camera subway surveillance setting [13].

In this paper, we address a mono-vision infrastructure-located real-time multi-object joint tracker and classifier. The core of the tracker is based on a *RJ MCMC PF* algorithm inspired of [5, 13] and extended to jointly track and classify objects and light source. Moreover, considering the difficulty to compute a reliable low level shadow segmentation, we choose to use a basic *background / foreground*

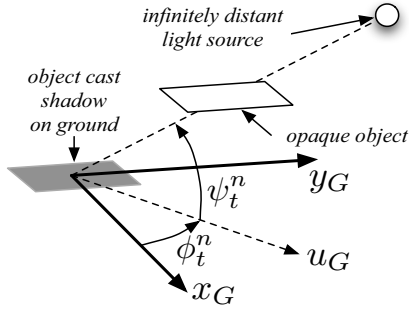


Figure 1. Infinitely distant light source and object cast shadow over the ground, assumed to be horizontal and defined by  $x_G$  and  $y_G$ . Light source position angles (azimut  $\phi_t^n$ , and elevation  $\psi_t^n$ ) are relative to the local ground reference.

segmented image as an observation. In [9], cast shadow is modeled using a 3-D object model, with a hand-defined sun position. We extend this approach to allow the *RJ MCMC PF* to automatically and continuously track sunlight estimate, allowing long-term outdoor tracking. The light source is modeled and updated over time within the particle filter, in order to manage slow but strong illumination changes caused by clouds and sun position dynamics. In section 2, we introduce joint light source and multi-object tracking. The observation likelihood is described in section 3, focusing on cast shadow representation. Object interaction weight is described in section 4. Finally, tracking results are reported and discussed in section 5.

## 2. Multi-Object MCMC PF

### 2.1. State Space

In an illumination aware visual object tracking, the system state encodes the configuration of the perceptible objects as well as illumination data:  $\mathbf{X}_t^n = \{\mathbf{I}_t^n, J_t^n, \mathbf{x}_t^{j,n}\}$ ,  $j \in \{1, \dots, J_t^n\}$ , where  $\mathbf{I}_t^n = \{\xi_t^n, \phi_t^n, \psi_t^n\}$  defines the illumination hypothesized by particle  $n$  at time  $t$ ,  $n \in \{1, \dots, N\}$ , where  $N$  is the number of particles. More precisely,  $\xi_t^n$  is a binary random variable hypothesizing sunlight to be broken by a cloud or not, while  $\phi_t^n$  and  $\psi_t^n$  are continuous random variables respectively standing for sun azimuth and elevation angles, as illustrated on figure 1. When sunlight is bright (unbroken), object shadows are assumed to be cast onto the ground or other objects.  $J_t^n$  is the number of visible objects for hypothesis  $n$  at time  $t$ , and each object  $j$  is defined by:  $\mathbf{x}_t^{j,n} = \{c_t^{j,n}, \mathbf{p}_t^{j,n}, \mathbf{v}_t^{j,n}, \mathbf{a}_t^{j,n}, \mathbf{s}_t^{j,n}\}$ . Object  $j$  category at iteration  $n$  is given by  $c_t^{j,n}$ , a discrete random variable belonging to object category set,  $C = \{\text{pedestrian, motorcycle, light vehicle, light truck, heavy truck}\}$  for instance.

Objects are assumed to move on a planar ground. Absolute position of candidate object  $j$  in particle  $n$  at time step  $t$  is defined by  $\mathbf{p}_t^{j,n} = (x_t^{j,n}, y_t^{j,n}, \rho_t^{j,n})$ , with object center of

gravity position  $x_t^{j,n}$  and  $y_t^{j,n}$ , and yaw angle  $\rho_t^{j,n}$ . Object  $j$  velocity and acceleration are described by  $\mathbf{v}_t^{j,n}$  and  $\mathbf{a}_t^{j,n}$ , with magnitude and orientation. Object shape is modeled by a cuboid with dimension vector  $\mathbf{s}_t^{j,n}$ . Considering the sun to be a unique infinitely distant light source allows to very simply cast hypothesis object shadows over the ground. Nevertheless, the method can be extended to one or more finitely distant light sources.

### 2.2. MCMC PF for Multi-Object Tracking

Let  $\mathbf{Z}_{1:t}$  the past observation sequence. Particle Filters approximate the posterior  $p(\mathbf{X}_t | \mathbf{Z}_{1:t})$  with  $N$  samples  $\mathbf{X}_t^n$ ,  $n \in \{1, \dots, N\}$  at each time step. As the posterior is dynamic, samples have to be moved at each time step. Isard *et al.* [6] proposed a sampling strategy known as *SIR PF* (Sequential Importance Resampling Particle Filter), and a monocular Multi-Object Tracker (*MOT*) based on it [3], where the posterior is resampled at each time step by an importance sampler. This method draws new samples by jointly moving along all the state space dimensions. The required number of samples and the computation load thus grow as an exponential of the space dimension, as focused in [12]. As a result, it cannot track more than 3 persons. To overcome this limitation, it is necessary to draw samples by only moving within a subspace of the state space. Khan *et al.* proposed the *MCMC PF* [4], replacing the importance sampler with a *MCMC* sampler, according to Metropolis-Hastings algorithm [7]. The chain is built by markovian transitions from particle  $\mathbf{X}_t^{n-1}$  to particle  $\mathbf{X}_t^n$  via a unique new proposal  $\mathbf{X}^*$ , which may be accepted with probability  $\alpha$  defined in 1. If refused then  $\mathbf{X}_t^n$  is a duplicate of  $\mathbf{X}_t^{n-1}$ .

$$\alpha = \min \left( 1, \frac{\pi^* P(\mathbf{X}^* | \mathbf{Z}_{1:t-1}) Q(\mathbf{X}_t^{n-1})}{\pi_t^{n-1} P(\mathbf{X}_t^{n-1} | \mathbf{Z}_{1:t-1}) Q(\mathbf{X}^*)} \right) \quad (1)$$

In eq. 1,  $\pi^* = P(\mathbf{Z}_t | \mathbf{X}^*)$  and  $\pi_t^{n-1} = P(\mathbf{Z}_t | \mathbf{X}_t^{n-1})$  are likelihoods for observation  $\mathbf{Z}_t$  under states  $\mathbf{X}^*$  and  $\mathbf{X}_t^{n-1}$ , as detailed in section 3,  $q(\mathbf{X})$  is the proposal law for a joint configuration  $\mathbf{X}$ ,  $w^* = w(\mathbf{X}^*)$  and  $w_t^{n-1} = w(\mathbf{X}_t^{n-1})$  are interaction weights detailed in section 4. As real objects do not behave independantly from each other, Khan *et al.* proposed to include it within the dynamics model, and showed that it can be moved out of the prior mixture:  $p(\mathbf{X} | \mathbf{Z}_{1:t-1}) \approx w(\mathbf{X}) \sum_n \prod_j p(\mathbf{x}_t^j | \mathbf{x}_{t-1}^{j,n})$ , where  $p(\mathbf{x}_t^j | \mathbf{x}_{t-1}^{j,n})$  is object  $j$  dynamics model. As *MCMC* sampler is an iterative strategy, Khan *et al.* proposed to draw new samples by only moving one object  $\mathbf{x}^j$  at a time, according to 2. This is the keypoint of the method: at each iteration, it lets the filter operate within object  $j$  subspace.

$$Q(\mathbf{X}^* | \mathbf{X}_t^{n-1}) \propto \begin{cases} Q(\mathbf{x}_t^{j*}) & \text{if } \mathbf{X}^{\setminus j*} = \mathbf{X}_t^{\setminus j, n-1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $\mathbf{X}^{\setminus j}$  is joint configuration  $\mathbf{X}$  without object  $j$ , and  $q(\mathbf{x}_t^{j*})$  is object  $j$  proposal law, whose approximation is:

$$q(\mathbf{x}_t^j) \approx \frac{1}{N} \sum_{n=1}^N p(\mathbf{x}_t^j | \mathbf{x}_{t-1}^{j,n}), \forall j \in \{1, \dots, J_t^{n-1}\} \quad (3)$$

The required number of particles thus is only a linear function of the number of tracked objects, when they do not interact. We adopt all the previous features.

### 2.3. Variable Number of Objects

To allow objects to enter or leave the scene, Khan *et al.* extended their *MCMC PF* to track a variable number of objects. For that purpose, the sampling step is operated by a *RJ MCMC* sampler (Reversible Jump Markov Chain Monte Carlo) [2], which can sample over a variable dimension state space, as the number of visible objects may change. This sampler involves the pair of discrete reversible moves  $\{\text{enter}, \text{leave}\}$  in order to extend the proposal law  $q(\mathbf{X})$ , thus allowing the state to jump towards a higher or lower dimension subspace [5, 12]. This sampler can approximate  $p(\mathbf{X}^* | \mathbf{Z}_{1:t})$  if the acceptance ratio  $\alpha$  is computed according to 1, involving evaluations of the proposal law  $q(\mathbf{X})$  for  $\mathbf{X}^*$  and  $\mathbf{X}_t^{n-1}$ . This leads to move-specific acceptance ratio computations, as shown in [5], and we use the same computations. In order to get time consistency, they also propose the pair of discrete reversible moves  $\{\text{stay}, \text{quit}\}$ . *Stay* allows to recover an object  $j$  which was present in the time  $t-1$  particle set, and no more is in the current particle at time  $t$ . *Quit* proposes an object  $j$  which was not present in the time  $t-1$  particle set, and is in the current particle at time  $t$ , to quit the scene. Though this pair of moves is devoted to object presence time consistency, it cannot cope with long duration occlusions or poor observation. For that reason, we do not use it and introduce object *vitality*, a continuous variable collecting the past likelihoods of each object, along iterations and time steps. It is integrated over all iterations of each time step, as detailed in appendix, and is used to drive object *leave* moves detailed in section in section 2.4. We extend the approach to reversible sun parameters and object category updates, yielding the following move set  $\mathcal{M} = \{\text{object enter}, \text{object leave}, \text{object update}, \text{sun enter}, \text{sun update}\}$  denoted  $\{e, l, u, se, su\}$  (*sun leaves* are treated with *object leaves*). Object category is tracked by proposing it to changes among set  $\mathcal{C} = \{\text{pedestrian}, \text{motorcycle}, \text{light vehicle}, \text{light truck}, \text{heavy truck}\}$  according to a transition matrix. This move extends the *MCMC PF* framework to object classification functionality. In addition to processing a geometry-based classification within the *RJ MCMC PF*, it is of high interest when object classes have obviously different dynamics such as a trailer versus a light vehicle on a windy road or a pedestrian versus a vehicle. In other words integrating object class as a random variable

within the *RJ MCMC PF* allows object time dynamics to contribute to object classification as well as object shapes.

### 2.4. Data-Driven Proposal Moves

In order to improve filter efficiency, object enter quota  $\rho_e$  is driven by observation  $\mathbf{Z}$  and particle  $\mathbf{X}_t^{n-1}$  at each iteration, according to eq. 12. Each object  $j$  leave quota  $\rho_l(j)$  depends on its vitality, according to eq. 25. Object  $j$  update, sun update, and sun enter quota are set to constant values:  $\rho_u(j) = 1$ ,  $\rho_{su} = 0.1$ , and  $\rho_{se} = 0.02$ . Move  $m$  probabilities  $P_m$  are computed from these quota, according to eq. 4, where  $J$  is the number of objects in particle  $\mathbf{X}_t^{n-1}$ .

$$P_m = \frac{\rho_m}{\rho_e + J\rho_u + \sum_{j \in \{1, \dots, J, s\}} \rho_l(j) + \rho_{se} + \rho_{su}}, \forall m \in \mathcal{M} \quad (4)$$

**Object Enter:** proposes a new object to enter with probability  $P_e$ , yielding joint configuration  $\mathbf{X}^* = \{\mathbf{X}_t^{n-1}, \mathbf{x}^{j*}\}$ . It is given a unique index  $j$ , initial dimensions, and initial vitality  $\Lambda_t^j = \Lambda_0$ . Acceptance rate is:

$$\alpha_e = \min \left( 1, \frac{\pi^* w^* P(\mathbf{X}^* | \mathbf{Z}_{1:t-1}) P_l(j)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1} | \mathbf{Z}_{1:t-1}) P_e Q(\mathbf{x}^{j*})} \right) \quad (5)$$

where object  $\mathbf{x}^{j*}$  is drawn from the *false background* distribution  $\mathbf{I}_{fb}$  (eq. 11), such that its projection fits  $\mathbf{I}_{fb}$  blob.

**Object  $j$  Leave:** proposes to withdraw object  $j$  from  $\mathbf{X}_t^{n-1}$  with probability  $P_l(j)$ , yielding the new joint configuration  $\mathbf{X}^* = \{\mathbf{X}_t^{n-1} \setminus \mathbf{x}_t^{j,n-1}\}$ . Acceptance rate is:

$$\alpha_l = \min \left( 1, \frac{\pi^* w^* P(\mathbf{X}^* | \mathbf{Z}_{1:t-1}) P_e Q(\mathbf{x}_t^{j,n-1})}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1} | \mathbf{Z}_{1:t-1}) P_l(j)} \right) \quad (6)$$

**Object  $j$  Update** with probability  $P_u$ . Proposes to change  $\mathbf{x}_t^{j,n-1}$  class according to a transition probability matrix. Randomly choose  $\mathbf{x}_{t-1}^{j,r}$ , an instance of object  $j$  from time  $t-1$  chain. Draw  $\mathbf{x}_t^{j*}$  from dynamics model  $p(\mathbf{x}_t^j | \mathbf{x}_{t-1}^{j,r})$  and build  $\mathbf{X}^* = \{\mathbf{X}_t^{\setminus j,n-1}, \mathbf{x}_t^{j*}\}$ . Object dynamics model is relative to object category (see section 5 for examples).

$$\alpha_u = \min \left( 1, \frac{\pi^* w^*}{\pi_t^{n-1} w_t^{n-1}} \right) \quad (7)$$

**Sun Enter:** proposes sunlight to become bright with probability  $P_{se}$ . Acceptance rate is:

$$\alpha_{se} = \min \left( 1, \frac{\pi^* w^* P(\mathbf{X}^* | \mathbf{Z}_{1:t-1}) P_l(s)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1} | \mathbf{Z}_{1:t-1}) P_{se}} \right) \quad (8)$$

**Sun Leave:** proposes sunlight to become cloudy with probability  $P_l(s)$ . Acceptance rate is:

$$\alpha_{sl} = \min \left( 1, \frac{\pi^* w^* P(\mathbf{X}^* | \mathbf{Z}_{1:t-1}) P_{se}}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1} | \mathbf{Z}_{1:t-1}) P_l(s)} \right) \quad (9)$$

**Sun Update** with probability  $P_{su}$ . Randomly chooses a sun position instance  $\mathbf{I}_{t-1}^r$ . Draw  $\mathbf{I}^*$  from sun dynamics laws (18) and (19). Acceptance rate is given by 7.

### 3. Observation Likelihood Function

In this section, we compute  $P(\mathbf{Z}|\mathbf{X})$ , the likelihood for observation  $\mathbf{Z}$ , given the joint multi-object configuration  $\mathbf{X}$ . Though we commonly use a multi-camera setting, a monovision setting will be considered in this section, for sake of simplicity. From the current image (Fig.2-a), and a background model (Fig.2-b), a foreground binary image  $\mathbf{I}_F(g)$  such as in Fig.2-d is computed, where  $g$  denotes a pixel location. We use  $\Sigma - \Delta$  algorithm [8], which efficiently computes an on-line adaptive approximation of background image temporal median and covariance, thus coping with outdoor illumination changes and noises for a low computational cost. On the other hand, each object hypothesized by particle  $\mathbf{X}$  is modeled as a cuboid with shape defined in section 2.1. The convex hull of its vertice projections is computed. If sunlight is unbroken, its cast shadow vertices are computed, and the corresponding convex hull also is computed. A binary mask image  $\mathbf{I}_M(g, \mathbf{X})$  is computed, with pixel  $g$  set to 1 if it is inside at least one of the convex hulls, else to 0, as drawn in Fig.2-c. Similarity image  $\mathbf{I}_S(g, \mathbf{X})$  is then computed (10), as well as *false background* image (11) used to drive object enter proposals through (12), where  $S_o$  is object projection prior area.

$$\mathbf{I}_S(g, \mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{I}_F(g) = \mathbf{I}_M(g, \mathbf{X}), \quad \forall g \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$\mathbf{I}_{fb}(g, \mathbf{X}) = \mathbf{I}_F(g) \& \overline{\mathbf{I}_M(g, \mathbf{X})}, \quad \forall g \quad (11)$$

$$\rho_e = \frac{1}{S_o} \sum_g \mathbf{I}_{fb}(g, \mathbf{X}) \quad (12)$$

The observation likelihood  $P(\mathbf{Z}|\mathbf{X})$  is computed as (13):

$$p(\mathbf{Z}|\mathbf{X}) = \left( \frac{1}{S} \sum_g \mathbf{I}_S(g, \mathbf{X}) \right)^{\beta_j}, \quad (13)$$

where  $\beta_j$  is computed according to object  $j$  projection area, according to the method detailed in [1]. This method is of high interest as it produces an observation likelihood that fairly tracks objects whatever their distance, and that fairly compares occluded and unoccluded objects. Both are highly demanded by video surveillance applications, such as highway or subway surveillance, where cameras cannot be located on a very elevated point, yielding deep occlusions and scale changes due to projection. Moreover, this method allows *MCMC PF* to operate with acceptance rate  $\alpha$  to be tuned to a target value, thus improving its efficiency.

### 4. Multi-Object Interaction Weight

As the foreground likelihood function allows fully occluded objects to survive, we must prevent them from getting stuck behind another object. For pedestrian tracking,

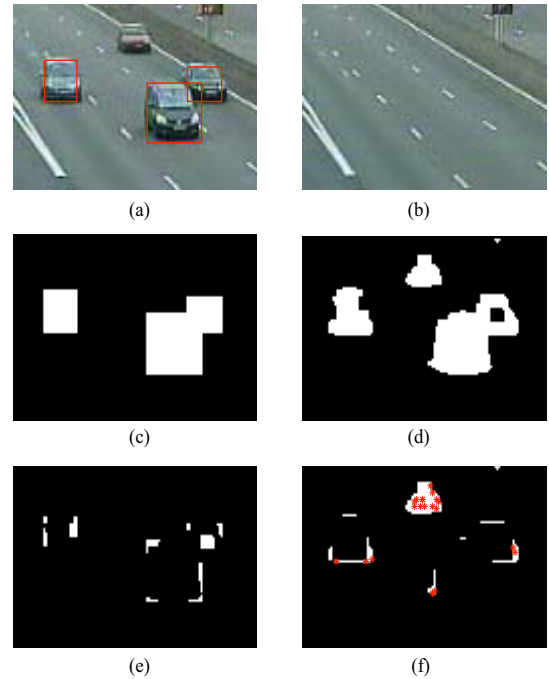


Figure 2. Background subtraction and residual images, with projected candidate objects. For readability, their projective polygons are approximated as rectangles. (a): raw color image with bounding object rectangles. (b): background model. (c): binary hypothesis image  $\mathbf{I}_M(g, \mathbf{X})$ . (d): binary foreground image  $\mathbf{I}_F(g)$ . (e): binary *false foreground* image, *i.e.* pixels covered by the projection of at least one object, but classified as background. (f): binary *false background* image  $\mathbf{I}_{fb}(g, \mathbf{X})$ , *i.e.* pixels not covered by any candidate object, but classified as foreground. Few points randomly sampled (red stars), to drive new object *enter* proposals.

[13] proposes to use a Mahalanobis distance rather than an Euclidean distance to model distances between pedestrians. We also compute an inter-object anisotropic weight based on Mahalanobis distance. This is mostly required in the case of vehicle tracking, because their lengths are much larger than their widths, and their interactions also are highly anisotropic, as 2 nearby vehicles are more likely to ride on 2 adjacent lanes rather than on the same lane. Moreover, the interaction between two vehicles depends on their dimensions. This is modeled by computing object interaction weight  $w$  as a function of an anisotropic distance between every pair of hypothesized vehicles. Both conditions are met approximating each object as a bivariate gaussian mass distribution, with covariance matrix featuring second order mass moments. Inter-vehicle distance then is:  $d_{ij} = (\Delta_{ij}^T \cdot (C_i \cdot C_j)^{-1} \cdot \Delta_{ij})^{1/2}$ , where  $\Delta_{ij}$  is the 2D position difference vector between vehicles  $i$  and  $j$ ,  $C_i$  and  $C_j$  their respective covariance matrices. Object pair interaction

weight then is computed according to equation (14):

$$w_{ij} = \left(1 + e^{-k_s \cdot (d_{ij} - d_s)}\right)^{-1}, \quad (14)$$

yielding a weight near 1 for far objects, and near 0 for materially impossibly close objects.  $d_s$  is the inter-vehicle distance corresponding to the sigmoid inflection parameter, and  $k_s$  is adjusted to tune curve slope around  $d_s$ . Interaction weight for particle  $\mathbf{X}$  involving  $J_t^n$  objects then is:

$$w(\mathbf{X}) = \prod_{i=1}^{J_t^n - 1} \prod_{j=i+1}^{J_t^n} w_{ij}. \quad (15)$$

## 5. EXPERIMENTS AND RESULTS

### 5.1. Datasets and Methodology

Tracker performance is assessed over both synthetic and real sequences. Datasets have been sampled from two different fields of applications: pedestrian tracking and highway vehicle tracking. Pedestrian tracking experiments are devoted to assessing the tracker ability to track more than 10 objects while coping with variable sunlight conditions. Highway vehicle tracking experiments are devoted to assessing the tracker ability to simultaneously track and classify vehicles such as cars, light trucks and trailer trucks, while also complying with time-evolving sunlight. As we want our tracker to comply with poor acquisition data, real sequences are provided by low-quality non-calibrated webcams with a  $320 \times 240$  pixel resolution and a high compression rate. Moreover, projection matrices have been approximated by hand. Target objects located within a selected tracking area (defined in the 3-d world and overplotted with green lines on figures 3, 4 and 5) are to be tracked and classified. We propose to assess the proposed method performance over four criteria:

- Tracking rate  $\theta_T = \frac{1}{J_t} \sum_{t,j} \delta_T(t,j)$  with  $\delta_T(t,j) = 1$  if target  $j$  is tracked at time  $t$ , else 0.  $J_t = \sum_{t,j} j_t$ , with  $j_t$  the number of objects in the tracking area.
- Classification rate  $\theta_C = \frac{1}{J_t} \sum_{t,j} \delta_C(t,j)$  where  $\delta_C(t,j) = 1$  if target  $j$  class is correct at time  $t$ , else 0.
- Ghost rate  $\theta_G = \frac{1}{J_t} \sum_{t,j} \delta_G(t,j)$  where  $\delta_G(t,j)$  is the number of *ghosts* *i.e.* candidate objects over no target.
- Position average error  $\varepsilon_T = \frac{1}{J_t} \sum_{t,j} (\delta_p^T \cdot \delta_p)^{-1}$ , with  $\delta_p = \mathbf{p}_t^{j,e} - \mathbf{p}_t^{j,gt}$ , where  $\mathbf{p}_t^{j,e}$  is object  $j$  estimated position at time  $t$ ,  $\mathbf{p}_t^{j,gt}$  is object  $j$  position ground truth.

Four methods are assessed according to  $\theta_T, \theta_C, \theta_G, \varepsilon_T$ :

- MOT - Multi Object Tracker: an implementation of *RJMCMC* algorithm with one category (object size noise has been increased in order to match different size objects) and with no light estimation.

- MOTS - Multi Object Tracker and Sun: an implementation of *RJMCMC* algorithm with one category (object size noise has been increased in order to match different size objects) and with light estimation.
- MOTC<sup>n</sup> - Multi Object Tracker and Classifier: an implementation of *RJMCMC* algorithm with  $n$  categories and with no light estimation.
- MOTC<sup>n</sup>S - Multi Object Tracker and Classifier with Sun: an implementation of *RJMCMC* algorithm with  $n$  categories and with light estimation.

### 5.2. Implementation

Two configuration proposals and their likelihoods are computed in parallel on each processing core, through threads supplied by the Boost C++ Libraries<sup>1</sup>. Code is written using *NT<sup>2</sup>* C++ Library<sup>2</sup>. We use a 3GHz Intel E6850 Core 2 Duo processor PC, with 4Go RAM, running Linux. All experiments presented below have been done at video real time (*i.e.* 25 fps), over mono-vision  $320 \times 240$  frames. The filter number of particles is set to  $N = 200$ .

### 5.3. Pedestrian tracking under variable sunlight

Datasets are sampled from pedestrian tracking sequences. Candidate pedestrians are controlled in velocity:

$$p(\mathbf{v}_t | \mathbf{v}_{t-1}^r) = \mathcal{N}(\mathbf{v}_{t-1}^r, \text{diag}([\sigma_m^2, \sigma_a^2])), \quad (16)$$

where  $\sigma_m$  and  $\sigma_a$  are the respective velocity magnitude and orientation standard deviations. Acceleration is not used. Dynamics laws then yield position  $\mathbf{x}_t^*$ . Shape is updated according to equation (17), where  $\sigma_s$  is object shape standard deviation, and  $I_3$  the 3-dimension identity matrix. Sun dynamics is defined by (18) and (19), where  $\sigma_\phi$  and  $\sigma_\psi$  respectively are sun azimuth and elevation standard deviations.

$$p(\mathbf{s}_t | \mathbf{s}_{t-1}^r) = \mathcal{N}(\mathbf{s}_{t-1}^r, \sigma_s^2 I_3) \quad (17)$$

$$p(\phi_t | \phi_{t-1}^r) = \mathcal{N}(\phi_{t-1}^r, \sigma_\phi^2), \forall r \in \{1, \dots, N\} \quad (18)$$

$$p(\psi_t | \psi_{t-1}^r) = \mathcal{N}(\psi_{t-1}^r, \sigma_\psi^2), \forall r \in \{1, \dots, N\} \quad (19)$$

**Synthetic Sequences:** Cuboid approximated pedestrians randomly move on a 12x15 meter wide tracking area, under a simulated time-evolving bright sunlight with elevation  $\psi = 0.8 \text{ rad}$  and azimuth increasing from  $\phi = 0$  to  $\phi = \pi \text{ rad}$  in 1000 frames. This is much faster than real world sun moves. Figure 3 illustrates tracking operation, showing the benefit of shadow modeling. Table 1 reports results for MOT and MOTS, and shows that modeling cast shadows decreases ghost rate and improves tracking accuracy.

<sup>1</sup><http://www.boost.org>

<sup>2</sup>Numerical Template Toolbox. <http://nt2.sourceforge.net>

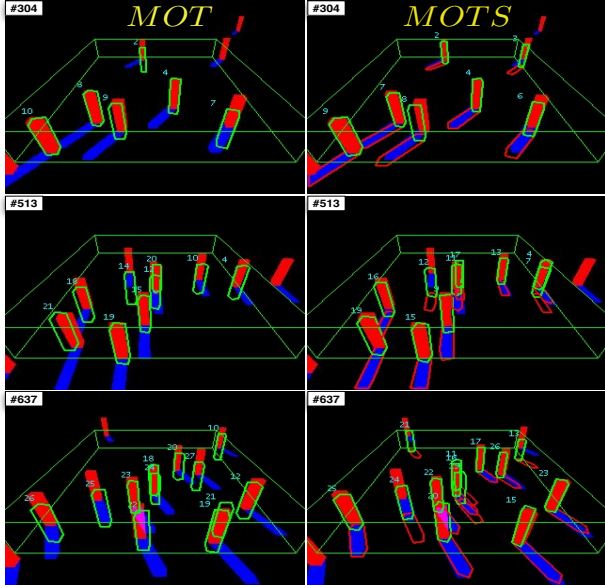


Figure 3. Excerpts from synthetic pedestrian tracking under time-evolving sunlight azimuth. Estimated object cuboids overlotted in green lines. Left column: no shadow model. Right column: estimated cast shadow overlotted in red.

Table 1. Tracking cuboid approximated pedestrians on synthetic scenes, under time-evolving sunlight, and under an alternation of bright and cloudy sunlight: sun state changes every 200 frames.

	bright sun		sun & clouds	
	MOT	MOTS	MOT	MOTS
$\theta_T$ (%)	84.7	89.7	84.1	87.0
$\theta_G$ (%)	5.7	4.9	5.1	4.3
<b>error (m)</b>	0.91	0.63	0.82	0.70

**Real Sequence:** A short sequence with clouds and sun yielding fast illumination changes. Figure 4 frame #786 illustrates three pedestrians being tracked while sunlight is estimated to be cloudy (no estimated cast shadow). Few frames later, as sunlight becomes brighter the tracker estimates it to appear at frame #823 and to remain bright until the end. The tracker fails at estimating the two targets walking side by side and occluding each other over the whole sequence: it tracks both people as a unique pedestrian (#14), due to lack of observability.

#### 5.4. Vehicle tracking and classification

These experiments aim at assessing the tracker ability to simultaneously track and classify vehicles such as cars, light trucks and trailer trucks. Vehicles are controlled through driver command proposals drawn from (20):

$$p(\mathbf{a}_t | \mathbf{a}_{t-1}^r) = \mathcal{N}(0, \text{diag}([\sigma_l^2, \sigma_t^2])), \quad (20)$$

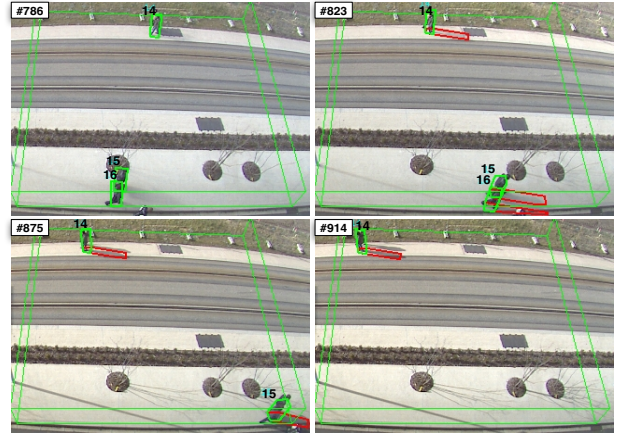


Figure 4. Excerpts from pedestrian tracking under time-evolving sunlight conditions. Estimated cuboids overlotted in green lines with estimated cast shadow in red when sunlight is bright.

Table 2. Two-class synthetic highway vehicle tracking and classification. Tracking rate  $\theta_T$  (%) / Classification rate  $\theta_C$  (%) / Ghost rate  $\theta_G$  (%). Average position error per vehicle in meters.

	MOT	MOTS	MOTC <sup>2</sup>	MOTC <sup>2</sup> S
light vehicles	.	.	59/54/0	<b>90/89/11</b>
trailer trucks	.	.	86/86/0	<b>90/89/0</b>
<b>total</b>	52/22/17	51/25/16	58/53/0	<b>90/89/11</b>
<b>error (m)</b>	6.17	5.80	2.76	<b>2.00</b>

where  $\sigma_l$  is driver longitudinal acceleration standard deviation,  $\sigma_t$  is driver steer angle standard deviation, conditioning transversal acceleration. Bicycle model equations as defined in [1] then are applied to object  $j$ . Dynamics laws then yield velocity  $\mathbf{v}_t^*$  and position  $\mathbf{x}_t^*$ .

**Synthetic Sequences:** They involve car and truck cuboid approximates on a three-lane highway, under bright sunlight. Table 2 reports results, showing that both classification and shadow modeling independently improve tracking. Best results are reached when both are activated.

**Real Sequences:** Real traffic sequences involving light vehicles, light trucks, and trailer trucks on a four-lane highway, including a highway entry lane, under variable sunlight. For real traffic tracking, a 3-class classification is necessary to take into account the three major classes of vehicles. Due to tracked object size wide range, methods without classification (MOT and MOTS) require vehicle shape dynamics  $\sigma_s$  to be dramatically increased, to let objects fit targets. Such a strategy cannot operate in presence of deep occlusions. To let them serve as reference anyway, as well as to let hand-made ground truth be affordable,

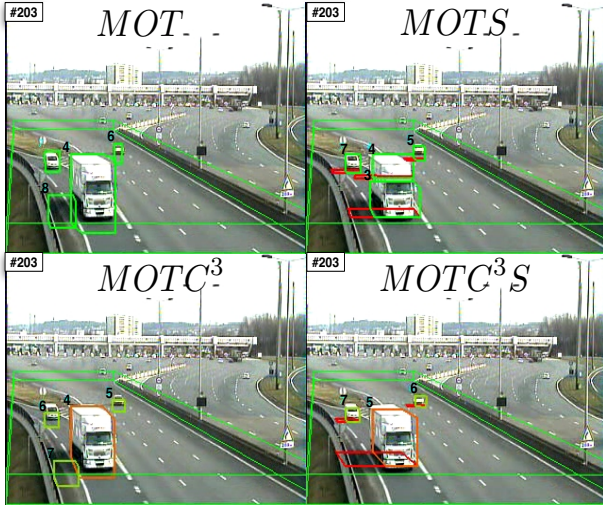


Figure 5. Frame #203 from real traffic tracking. Top row: no classification, vehicle estimated cuboids overplotted in green lines. Bottom row: classification in 3 categories with light vehicle (resp. light trucks and trailer trucks) estimated cuboids overplotted in green lines (resp. magenta and orange). Left: without cast shadow model. Right: with cast shadow modeling, overplotted in red lines.

Table 3. Three-class real highway vehicle tracking and classification. Tracking rate  $\theta_T$  (%) / Classification rate  $\theta_C$  (%) / Ghost rate  $\theta_G$  (%). Average position error per vehicle in meters.

	MOT	MOTS	MOTC <sup>3</sup>	MOTC <sup>3</sup> S
light vehicles	.	.	67/64/2.6	<b>67/67/0.05</b>
light trucks	.	.	83/36/1.0	<b>92/86/3.7</b>
trailer trucks	.	.	93/83/0	<b>100/100/2</b>
<b>total</b>	51/45/0	60/51/0	72/62/2.5	<b>70/70/3.1</b>
<b>error (m)</b>	6.80	6.22	6.13	<b>5.40</b>

we choose a sequence with light traffic, but involving all categories of vehicles. Figure 5 illustrates that both multi-category classification and cast shadow modeling improve tracking. MOT and MOTS typical failures are: two objects tracking a unique target (MOTS) or poor tracking accuracy (MOT). Without cast shadow modeling, the system fails at tracking very differently sized objects: it explains truck cast shadow pixels classified as foreground with a *ghost* car (#7 on MOTC<sup>3</sup> and #8 on MOTC<sup>3</sup>S). Modeling cast shadow explains these foreground pixels (MOTC<sup>3</sup>S). Moreover, further cars are more accurately located when shadow is modeled (MOTS and MOTC<sup>3</sup>S), as their shadows provide clues concerning their longitudinal position. Table 3 reports results and confirms section 5.4 analysis: both classification and shadow modeling improve tracking, with best results when both are activated.

## 6. Conclusion and Future Works

We have proposed a generic illumination-aware framework to simultaneously track and classify multiple objects into various classes in real time. The system can be operated in monovision or with a multi-camera setting. It is wholly integrated within a RJ MCMC Particle Filter framework. To make this possible, illumination is integrated into the global configuration state-space, and tracked as well as objects. Experiments show that joint object and sunlight estimation improves tracking, both decreasing false positives and object position error. We also have proposed to include object category as a discrete random variable to be estimated by the filter, extending RJ MCMC PF framework to object classification functionality. Experiments show that simultaneously tracking and classifying improves tracking as it proposes multiple geometric models, thus allowing better model fitting. This unified approach also is of high interest as it allows tracking and classification to cooperate through object class specific dynamics. This functionality might be used to improve tracking and classification of objects with similar geometric models, but with different dynamics models, such as cyclists and pedestrians for instance. As this tracker is designed to be generic, it is based on low level information (simple background segmentation), and complies with low-quality acquisition data. There is undoubtedly room for improvement, adding object ad-hoc features in the likelihood computation. The work presented in this paper deals with a unique illumination source, well suited to model sun illumination. It can easily be extended to multiple illumination sources, and to ground reflection modeling suitable for indoor lighting or outdoor wet conditions.

## Appendix: Vitality-Driven Leave Moves

Object and sun vitalities, ranging from 0 to 1, are updated by the same process. At iteration  $n$  of time  $t$ , we compute object  $j$  false foreground ratio  $f_t^{j,n}$ :

$$f_t^{j,n} = \frac{1}{|\mathcal{R}_t^{j,n}|} \sum_{g \in \mathcal{R}_t^{j,n}} \overline{\mathbf{I}_F(g)}, \forall j \in \{1, \dots, J_t^n, s\} \quad (21)$$

where  $s$  denotes sun as an object.  $\mathcal{R}_t^{j,n}$  denotes image region covered by the projection convex hull of each object but the sun ( $\forall j \in \{1, \dots, J_t^n\}$ ). For the sun ( $j = s$ ),  $\mathcal{R}_t^{j,n}$  is the region covered by the union of all object cast shadow projections. Object  $j$  vitality increment  $\lambda_t^j$  is computed (22) over the whole particle set, as a sum of sigmoids of  $f_t^{j,n}$ :

$$\lambda_t^j = k_d \sum_{n=1}^N \frac{e^{-k_r \cdot (f_t^{j,n} - r_f)} - 1}{e^{-k_r \cdot (f_t^{j,n} - r_f)} + 1}, \forall j \in \{1, \dots, J_t^n, s\} \quad (22)$$

where  $r_f$  is the inflection parameter of false foreground rate curve (*i.e.* the value of  $f_t^{j,n}$  yielding an increment equal to

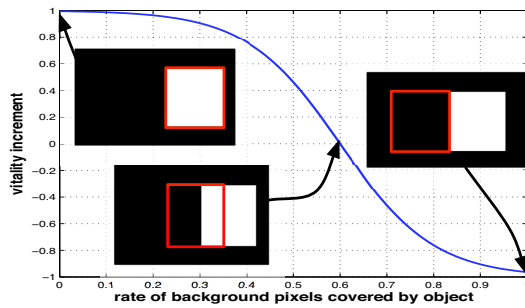


Figure 6. Vitality increment  $\lambda_t^j$  versus object  $j$  false foreground rate  $f_t^{j,n}$ , in monovision, with  $k_d = 1$ ,  $r_f = 0.6$  and  $k_r = 10$ .

0), and  $k_r$  is the curve steepness parameter. Equation (22) produces a positive increment if  $f_t^{j,n} < r_f$ , else negative, allowing object vitality to compile the history of object  $j$  likelihoods along past iterations and time steps. Vitality dynamics coefficient  $k_d$  is computed in equation (23):

$$k_d = (1 - \Lambda_0)(n_s \cdot C \cdot N)^{-1}, \quad (23)$$

where  $\Lambda_0$  denotes object initial vitality,  $n_s$  denotes the number of frames an object with maximal vitality can survive total invisibility (generally due to total occlusion by the background). This parameter allows the user to adjust vitality dynamics, depending on the duration of possible occlusions. Each object vitality is finally updated for time  $t + 1$ :

$$\Lambda_{t+1}^j = \begin{cases} \min(\Lambda_t^j + \lambda_t^j, 1) & \text{if } (j = s \text{ or } z_j) \\ \max(\Lambda_t^j + \lambda_{out}, 0) & \text{otherwise} \end{cases}, \quad (24)$$

where  $z_j$  is a binary variable set to 1 if object  $j$  is in the tracking area, else 0. In the latter case, its vitality is updated by  $\lambda_{out}$ . The values chosen for experiments, reported in table 4, yield the vitality increment illustrated on Fig. 6. At each time step  $t$ , object  $j$  leave proposal rate  $\rho_l(j)$  is driven by its own vitality, according to equation (25):

$$\rho_l(j) = \left(1 + e^{k_v \cdot (\Lambda_t^j - \Lambda_0)}\right)^{-1}, \forall j \in \{1, \dots, J_t^n, s\}. \quad (25)$$

Sigmoid inflection parameter is chosen equal to  $\Lambda_0$ , yielding object enter and leave reversibility. Less leave proposals appear as object  $j$  vitality grows higher than  $\Lambda_0$ , preventing it from leaving the scene at once when poorly segmented from background or deeply occluded. In this case, vitality allows it to survive several images.  $k_v$  is sigmoid steepness parameter. The same mechanism stands for sun, with slower dynamics driven by a higher  $n_s$  (see table 4).

## References

[1] F. Bardet and T. Chateau. MCMC particle filter for real-time visual tracking of vehicles. In *International IEEE Confer-*

Table 4. Object vitality computation parameters.  $n_s = 25$  frames means 1 second long total occlusion survival at 25 fps.

	$\Lambda_0$	$\lambda_{out}$	$r_f$	$n_s$	$k_r$	$k_v$
<b>object</b>	0.2	-0.1	0.6	10	10	10
<b>sun</b>	0.2	.	0.4	50	10	10

*ence on Intelligent Transportation Systems*, pages 539 – 544, 2008. 4, 6

[2] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 4(82):711–732, 1995. 3

[3] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision*, vol. 2 34–41, 2001. 2

[4] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. *ECCV*, 3024:279–290, 2004. 2

[5] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1805 – 1918, 2005. 1, 3

[6] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV : International Journal of Computer Vision*, 29(1):5–28, 1998. 2

[7] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 2

[8] A. Manzanera and J. Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'04)*, pages 46–51, 2004. 4

[9] J. L. M. Matthew J. Leotta. Learning background and shadow appearance with 3-d vehicle models. In *British Machine Vision Conference (BMVC)*, volume 2, pages 649–658, september 2006. 2

[10] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:918–923, 2003. 1

[11] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, 95(2):238 – 259, August 2004. 1

[12] K. Smith. *Bayesian Methods for Visual Multi-Object Tracking with Applications to Human Activity Recognition*. PhD thesis, EPFL, Lausanne, Suisse, 2007. 2, 3

[13] J. Yao and J.-M. Odobez. Multi-camera multi-person 3D space tracking with mcmc in surveillance scenarios. In *European Conference on Computer Visionworkshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*, 2008. 1, 4

[14] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008. 1