

Research Article

Vehicle Trajectory Estimation Using Spatio-Temporal MCMC

Yann Goyat,¹ Thierry Chateau,² and Francois Bardet²

¹LCPC, Route de Bouaye, 44341 Bouguenais, France

²LASMEA, Université Blaise Pascal, 24 Avenue des landais, 63177 Aubière, France

Correspondence should be addressed to Yann Goyat, yann.goyat@lcpc.fr

Received 19 October 2009; Accepted 23 March 2010

Academic Editor: Robert W. Ives

Copyright © 2010 Yann Goyat et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an algorithm for modeling and tracking vehicles in video sequences within one integrated framework. Most of the solutions are based on sequential methods that make inference according to current information. In contrast, we propose a deferred logical inference method that makes a decision according to a sequence of observations, thus processing a spatio-temporal search on the whole trajectory. One of the drawbacks of deferred logical inference methods is that the solution space of hypotheses grows exponentially related to the depth of observation. Our approach takes into account both the kinematic model of the vehicle and a driver behavior model in order to reduce the space of the solutions. The resulting proposed state model explains the trajectory with only 11 parameters. The solution space is then sampled with a Markov Chain Monte Carlo (MCMC) that uses a model-driven proposal distribution in order to control random walk behavior. We demonstrate our method on real video sequences from which we have ground truth provided by a RTK GPS (Real-Time Kinematic GPS). Experimental results show that the proposed algorithm outperforms a sequential inference solution (particle filter).

1. Introduction

Efficient target tracking is a critical component in many computer vision applications such as visual surveillance or robotics. The object-tracking procedure is intended to estimate the state (position, velocity, ...) of an object at each time given an observation sequence.

Tracking methods can be divided into two major categories: The first category relates to sequential inference tracking (also called online or causal tracking), for which the state of the object at a given time step has been estimated as a function of the record of past and current observations and the record of past states. The second concerns deferred logical inference (also called offline or noncausal tracking), for which the state estimation at a given point in time uses the entire observation sequence.

Sequential tracking is needed when the tracker's output controls real-time processes, which cannot be delayed (such as robotic applications). Sequential tracking is also needed when it is not possible to record the observation data, due to its size, or due to regulations. Much work has been done on sequential visual tracking (model-based approaches [1–3] or learning-based approaches [4–6]). Therefore, most of

the applications use sequential methods even though it is not necessary.

For other situations, deferred tracking is much more appealing, as it is not causal. This allows the optimisation process to operate over a larger data set (the whole observation sequence), thus allowing to hope for better results. Deferred visual multiobject tracking have already been successfully experienced on pedestrian tracking in [7] and with a MCMC search in [8].

The solution presented in this paper is a spatio-temporal deferred logical inference approach. One of the main challenges of such methods is that the solution space of hypotheses grows exponentially related to the (duration) depth of observation. In the specific case of vehicle tracking, priors on both driver behavior and the road geometry can be used. Moreover, the trajectory of the object to be tracked is driven by a kinematic model. Therefore, we propose an 11-dimensional reduced state vector of the vehicle trajectory.

Since we use a probabilistic framework, the tracking problem can be seen as the estimation of the distribution of the state vector posterior distribution, given a video sequence. We propose a Markov Chain Monte Carlo (MCMC) method to sample the posterior distribution.

MCMC have been already used in visual tracking. In [9, 10], a MCMC based particle filter is presented for multiobject tracking and an extension is proposed to handle a varying number of objects (Reversible Jump Markov Chain Monte Carlo, RJMCMC). In [8], the RJMCMC algorithm is used in a deferred logical inference framework to track several vehicles offline from a video sequence.

In MCMC methods, the random walk behavior is driven by proposal distributions. We use priors on driver behavior and road geometry to define efficiently the proposal. Exploration is achieved with the Metropolis-Hasting rule according to a global likelihood function.

We use a likelihood function based on a background subtraction algorithm. A discrete set of positions of the vehicle into the video sequence is generated from the trajectory state. A generic 3D model of a vehicle is then projected into each image and then compared to a background/foreground map of the video sequence. We propose an efficient implementation of the likelihood function using a line integral image to decrease computation time.

Experiments have been done to compare, on real video sequences, the deferred logical inference approach with a classic sequential particle filter.

The remaining of the paper is organized as follows. Section 2 presents the probabilistic framework proposed to solve the tracking problem. Section 3 provides a detailed description of the vision likelihood function. A set of experimental results along with both qualitative and quantitative analysis is presented in Section 4, before we conclude in Section 5.

2. Proposed Method

This section describes the core of the method, based on a probabilistic framework. Figure 1 represents an illustration of the algorithm. Given a video sequence and an initial state, the method samples the posterior distribution of the trajectory using a random walk method (MCMC). In the following, we begin by presenting the state vector associated to the trajectory model. Then, we give an overview of the Monte Carlo Markov Chain algorithm used to sample the posterior distribution. Finally, we show how to generate new proposals by sampling from an object-specific proposal distribution.

2.1. State Vector Reduction. In a spatio-temporal deferred logical inference approach, the solution space of hypotheses grows exponentially related to the number of observation frames. Estimating a single vehicle planar trajectory along a 100 frame sequence, involves a state space of dimension 300 (planar position and orientation are estimated for each frame). Conducting a Monte-Carlo search in such a space is computationally intractable! To avoid this problem, we do not consider the vehicle position sequence as the state vector, but we implement a trajectory generator, lying on driver behavior priors, road geometry priors, and vehicle kinematic priors. The vehicle trajectory generator detailed in the following, generates trajectory samples, defined in

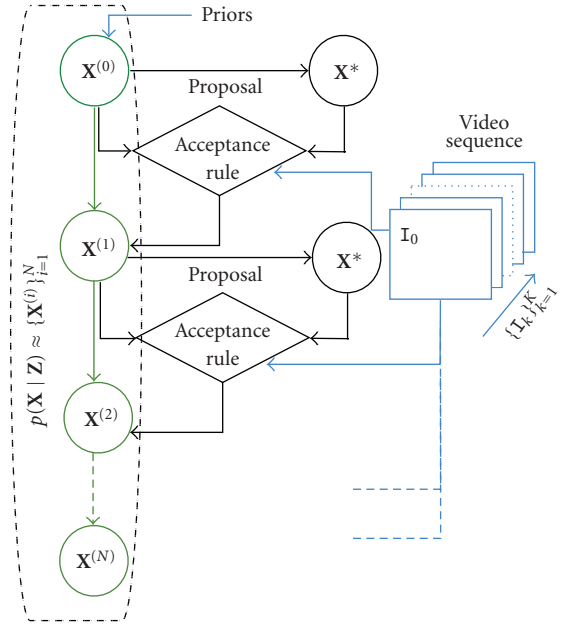


FIGURE 1: Given a video sequence and an initial state, the method samples the posterior distribution of the trajectory using a random step method (MCMC).

a 11-dimensional state space by driver temporal command parameters. This method drastically reduces the dimension of the state space, thus improving computational efficiency.

2.2. Driver Command and Vehicle Priors. The driver commands are the steering wheel angle, and the vehicle longitudinal acceleration, from which we deduce the vehicle speed through integration. The experiments presented below have been conducted on a mid-velocity curve. While traveling such a curve, a light vehicle driver's command law is commonly modelled by a trapezoid, with steering wheel angle velocities lying between 1.5 and 4 degrees per second, and with absolute longitudinal accelerations lying between $1 \text{ m}\cdot\text{s}^{-2}$ and $3 \text{ m}\cdot\text{s}^{-2}$. In order to take into account a wider range of driver commands, and the steering system nonlinearities due to frictions and mechanical compliances, we use a more compliant model: we model the steering command with a double sigmoid (one for entering the curve, and one for releasing from it).

As the experiments presented below have been conducted on the second half of the curve, a single sigmoid is used to define the steering angle generator, from parameters defined in Section 2.2 (cf. Figure 2)

$$f_{\delta}(\theta_{\delta}, k) \doteq \frac{\theta_{\delta,2}}{1 + \exp[\theta_{\delta,3}(\theta_{\delta,4} - k)/|\theta_{\delta,2}|]} + \theta_{\delta,1}, \quad (1)$$

$k = \text{time.}$

The same reasoning applies to the vehicle speed generator $f_v(\theta_v, k)$, calculated as $f_{\delta}(\theta_{\delta}, k)$, swapping index δ into index v .

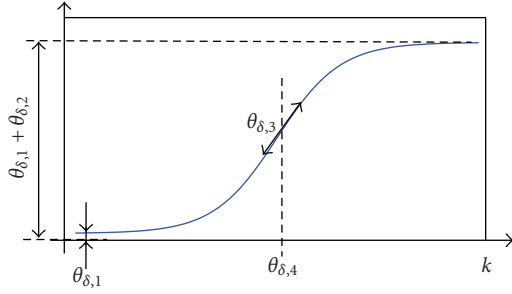


FIGURE 2: Graphic representation of the sigmoid parameters.

To model the vehicle, we use a plain kinematic model, as described in Section 3.1. This model allows us to iteratively generate x_k , y_k , and α_k , for every time step k .

The vehicle trajectory generator is represented by a random state vector $\mathbf{X} \doteq (\mathbf{1}, \boldsymbol{\theta}_\delta, \boldsymbol{\theta}_v)^T$ with

- (i) $\mathbf{1} \doteq (x_0, y_0, \alpha_0)$ represents the initial position and orientation of the vehicle (into a world reference frame),
- (ii) $\boldsymbol{\theta}_\delta \doteq (\theta_{\delta,1}, \dots, \theta_{\delta,4})$ are the parameters of a sigmoid function $\delta_k = f_\delta(\boldsymbol{\theta}_\delta, k)$ representing the discrete temporal evolution of the steering angle,
- (iii) $\boldsymbol{\theta}_v \doteq (\theta_{v,1}, \dots, \theta_{v,4})$ are the parameters of a sigmoid function $v_k = f_v(\boldsymbol{\theta}_v, k)$ representing the discrete temporal evolution of the vehicle velocity.

2.3. MCMC. We want to estimate $p(\mathbf{X} | \mathbf{Z})$, the posterior probability density for a model's parameters \mathbf{X} , given some observed data \mathbf{Z} . Monte-Carlo methods assume that the posterior distribution can be approximated by a set of N samples:

$$p(\mathbf{X} | \mathbf{Z}) \approx \left\{ \mathbf{X}^{(n)} \right\}_{n=1}^N. \quad (2)$$

Sampling from $p(\mathbf{X} | \mathbf{Z})$ is a hard problem and many methods have been proposed. Metropolis-Hasting is a random walk algorithm designed to approximate a stationary distribution. At each step, a state \mathbf{X}^* is proposed according to a proposal density $q(\mathbf{X}^* | \mathbf{X})$. The proposal state is then accepted or rejected according to an acceptance ratio defined by the Metropolis-Hasting rule

$$\alpha = \min \left(1, \frac{p(\mathbf{Z} | \mathbf{X}^*)}{p(\mathbf{Z} | \mathbf{X})} \times \frac{q(\mathbf{X} | \mathbf{X}^*)}{q(\mathbf{X}^* | \mathbf{X})} \right). \quad (3)$$

Metropolis-Hasting rule can be used to build a Markov Chain which approximates the posterior distribution $p(\mathbf{X} | \mathbf{Z})$. The resulting method is called Markov Chain Monte Carlo. Moreover, the N_b first elements of the chain are removed (burn-in) in the final sampling set. An estimate of the state is given by a maximum likelihood rule applied to the particle set

$$\hat{\mathbf{X}} \doteq \arg \max_{\mathbf{X}} \sum_{n=1}^N \delta(\mathbf{X} - \mathbf{X}^{(n)}), \quad (4)$$

where δ is the Dirac function.

Input: The first element of the chain \mathbf{X}^0 and its weight proportional to its likelihood: $\pi(\mathbf{X}^0) \propto P(\mathbf{Z} | \mathbf{X}^0)$
for $n = 1$ to $N + N_B$ **do**
 - Choose a move $m \in \{1, \dots, M\}$ among all the parameter of the state \mathbf{X} according to prior $q'(m)$.
 - Draw a proposal \mathbf{X}^* from the distribution $q(\mathbf{X}^* | \mathbf{X})$ with $\mathbf{X} = \mathbf{X}^{n-1}$
 - Evaluate its joint likelihood: $p(\mathbf{Z} | \mathbf{X}^*)$
 - Compute the acceptance ratio using Metropolis-Hasting rule:

$$\alpha = \min \left(1, \frac{p(\mathbf{Z} | \mathbf{X}^*)}{p(\mathbf{Z} | \mathbf{X})} \times \frac{q(\mathbf{X} | \mathbf{X}^*)}{q(\mathbf{X}^* | \mathbf{X})} \right)$$

- Add a n th element to the chain $\mathbf{X}^n = \mathbf{X}^*$ with probability α , (otherwise $\mathbf{X}^n = \mathbf{X}^{n-1}$).

end for

Burn-in: delete the N_B first elements of the chain.

Output: N -element Markov Chain of state hypothesis:

$\{\mathbf{X}^n\}_{n=N_B+1, \dots, N_B+N}$

ALGORITHM 1: MCMC algorithm.

2.4. Proposals. At iteration n , the MCMC generates a new proposal by sampling from a proposal distribution $q(\mathbf{X}^* | \mathbf{X}^{(n-1)})$ defined by

$$q(\mathbf{X}^* | \mathbf{X}^{(n-1)}) = \sum_{m \in \{1, \dots, M\}} q'(m) q(\mathbf{X}^* | \mathbf{X}^{(n-1)}, m), \quad (5)$$

where $q'(m)$ is a prior distribution used to select the parameter index of \mathbf{X} to be modified (M denotes the size of \mathbf{X}). A parameter-specific proposal distribution is then defined by

$$q(\mathbf{X}^* | \mathbf{X}, m) \doteq p(X_m^* | X_m^{(n-1)}) \prod_{j \neq m} \delta(X_j^* - X_j^{(n-1)}). \quad (6)$$

Here, only the m th component (m is selecting with the prior distribution $q'(m)$) of the state vector is moved at iteration n ; the other parameters remain unchanged. The MCMC is summarized in Algorithm 1.

3. Observation

This section presents the observation function defined to compute the likelihood $p(\mathbf{Z} | \mathbf{X} = \mathbf{X}^{(n)})$ probability to observe the video sequence, given a sample $\mathbf{X}^{(n)}$. Figure 3 illustrates the observation process. A discrete set of positions of the vehicle into the video sequence is generated from the trajectory sample \mathbf{X} . A generic 3D model of a vehicle is then projected into each image and compared to a background/foreground map of the video sequence.

We propose an efficient implementation of the likelihood function using a line integral image to decrease computation time.

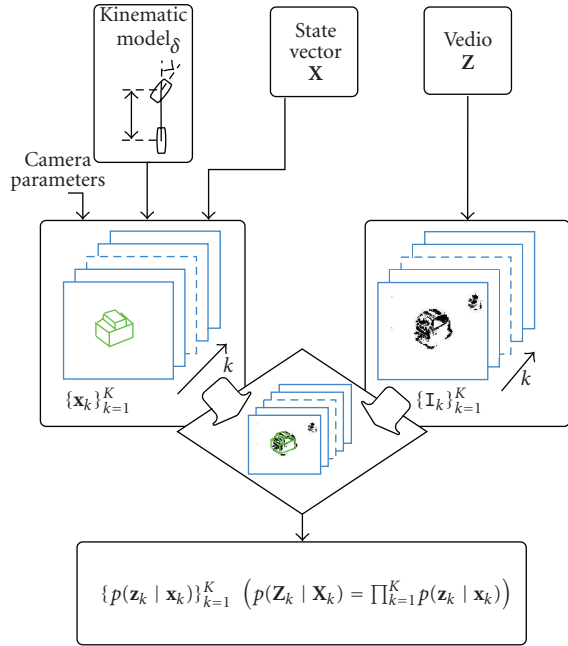


FIGURE 3: Illustration of the likelihood function. A discrete set of positions of the vehicle into the video sequence is generated from the trajectory sample. A generic 3D model of a vehicle is then projected into each image and compared to a background/foreground map of the video sequence.

3.1. *Building a Discrete Set of Vehicle Positions.* Let \mathcal{X} define a discrete set of temporal positions and orientations of the vehicle, associated to a sample \mathbf{X} of the posterior distribution

$$\mathcal{X} \doteq \{\mathbf{x}_k\}_{k=1}^K, \quad (7)$$

with $\mathbf{x}_k \doteq (x_k, y_k, \alpha_k)^T$ is a vector which gives the position and orientation of the vehicle at time k into a world reference frame \mathcal{R}_w associated to a planar ground. \mathbf{x}_k can be computed in a recursive way using a simple kinematic model of the vehicle. Here, we used a bicycle model (cf. Figure 4)

$$\begin{aligned} x_k &= x_{k-1} + T \cdot v_{k-1} \cdot \cos(\alpha_{k-1}), \\ y_k &= y_{k-1} + T \cdot v_{k-1} \cdot \sin(\alpha_{k-1}), \\ \alpha_k &= \alpha_{k-1} + T \cdot \frac{v_{k-1}}{L} \cdot \tan(\delta_{k-1}), \end{aligned} \quad (8)$$

where T is the sample time used for the video acquisition and L denotes the wheelbase (distance between front and rear wheels). δ_k and v_k are given by the steering angle and velocity parametric functions presented into Section 2.2.

The likelihood function $p(\mathbf{Z} | \mathbf{X})$ can be written by

$$p(\mathbf{Z} | \mathbf{X}) = p(\mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_K | \mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_K) \quad (9)$$

and assuming independence of random variables

$$p(\mathbf{Z} | \mathbf{X}) = \prod_{k=1}^K p(\mathbf{z}_k | \mathbf{x}_k). \quad (10)$$

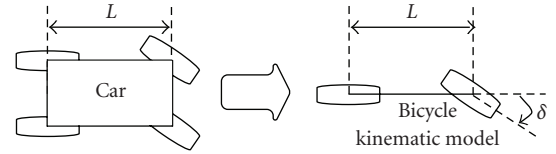


FIGURE 4: The bicycle model synthesizes the displacement of a four-wheel vehicle, through the displacement of two wheels whose centers are connected by a rigid axis of length L . Ackerman's theory serves to estimate the steering angle of the front axis of a vehicle traveling at low speed.

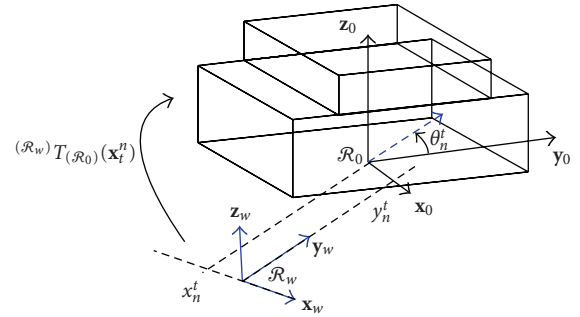


FIGURE 5: Example of a simple three-dimensional geometric model used for a vehicle. It is composed of two cubes. The coordinate system associated with the cube and the other system associated with the scene are related according to pure translation. The plane (Oxy) of the world coordinate system and component axes are merged with the GPS coordinate system.

3.2. *Computing $p(\mathbf{z}_k | \mathbf{x}_k)$.* Since the video sequence comes from a static camera, vehicle extraction is achieved using a background/foreground extraction approach. We use a nonparametric method [11], based on discrete modelization of the background probability density of the pixel color (RGB). The algorithm provides a set of binary images $\mathcal{I} \doteq \{\mathbf{I}_k\}_{k=1}^K$, where $\mathbf{I}_k(\mathbf{u}) = 1$ if the pixel $\mathbf{u} \doteq (u_x, u_y)^T$ is associated to foreground and $\mathbf{I}_k(\mathbf{u}) = -1$ if the pixel is associated to background.

A simplified three-dimensional geometric model of the vehicle is used, as depicted in Figure 5. This model is composed of two nested parallelepipeds. In a general case, the model may be more complex and contain P_M parallelepipeds. Let $\mathcal{M}^{(\mathcal{R}_0)} = \{M_i^{(\mathcal{R}_0)}\}_{i=1, \dots, N_M}$ represent the model's set of cube vertices ($N_M = 8 \times P_M$), expressed within a coordinate system associated with model \mathcal{R}_0 . This coordinate system is selected such that the 3 axes all lie in the same direction as that of the world coordinate system \mathcal{R}_w .

Each point of the vehicle model is projected onto the image via the following equation

$$\tilde{\mathbf{m}}_i \propto \mathbf{C}_c \cdot {}^{(\mathcal{R}_w)}\mathbf{T}_{(\mathcal{R}_0)}(\mathbf{x}_k) \cdot \tilde{\mathbf{M}}_i^{(\mathcal{R}_0)}, \quad (11)$$

with $\tilde{\mathbf{M}}$ homogeneous coordinates associated with point M ; \mathbf{C}_c is the camera projection matrix, and ${}^{(\mathcal{R}_w)}\mathbf{T}_{(\mathcal{R}_0)}(\mathbf{x}_k)$ the homogeneous transformation matrix between the world coordinate system and the system associated with the 3D model (cf. Figure 5).

The set $\mathcal{M}^{(R_i)} = \{\mathbf{m}_i\}_{i=1,\dots,N_M}$ is thus built based on the projection of 3D model points within the image.

For a given position \mathbf{x}_k , the likelihood is linked to the difference between the number of foreground and background pixels inside the vehicle model projection in the image. This computation performed for each particle spends SIGNIFICANT processing time, and we are proposing herein a fast likelihood calculation method based on an approximation of the 3D model projection in the image through its convex hull.

Let $\mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{x}_k) \doteq \{\mathbf{e}_i\}_{i=1,\dots,N_e}$ ($\mathbf{e}_i = (x_i^e, y_i^e)$ as coordinates of \mathbf{e}_i in the image plane) be the list of convex hull points. (Calculation of the convex hull is not developed in this article; the calculation procedure is conducted using a classical algorithm with a complexity expressed in $O(N \cdot \log N)$.) We will now define $\mathcal{E}_k \doteq \mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{x}_k)$ in order to streamline notations. The likelihood calculation may be performed efficiently by use of a line-by-line integral image defined by

$$\mathbb{I}_k^\Sigma((x, y)^T) = \sum_{i=1}^x \mathbb{I}_k((i, y)^T). \quad (12)$$

Points \mathbf{e}_i are categorized by pairs featuring the same y -coordinate values, such that

$$\mathcal{E}_k = \left\{ (x_1^e, y^e), (x_2^e, y^e), (x_3^e, y^e + 1), (x_4^e, y^e + 1), \dots \right. \\ \left. (x_{N-1}^e, y^e + \frac{N_e}{2}), (x_N^e, y^e + \frac{N_e}{2}) \right\}. \quad (13)$$

Convex hull coding within the set \mathcal{E}_k necessitates a shape discretization along the image lines. Moreover, special attention needs to be paid to coding the upper and lower extremities. On the other hand, it is not at all necessary to sort points positioned on the same line. A compliance measurement relative to a convex hull is computing from the integral image by application of the following relation:

$$a(\mathcal{E}_k) = \sum_{j=1}^{N_e/2} \left[2 \cdot (\mathbb{I}_k^\Sigma(\mathbf{e}_{2j}) - \mathbb{I}_k^\Sigma(\mathbf{e}_{2j-1})) - (x_{2j}^e - x_{2j-1}^e) \right]. \quad (14)$$

Figure 6 describes the principle behind the likelihood calculation method using the integral image. A line-by-line scanning is performed as part of this method.

Finally, the likelihood expression is written by

$$p(\mathbf{z}_k | \mathbf{x}_k) \propto C_{\mathcal{E}_k}^{-1} \max(0, a(\mathcal{E}_k)) \quad (15)$$

with the normalization constant $C_{\mathcal{E}_k} \doteq \sum_{j=1}^{N_e/2} |x_{2j}^e - x_{2j-1}^e|$ defining the surface of the convex hull.

4. Experimental Validation

In this section, experimental results are presented to highlight the relevance of our tracker. We compare the Offline proposed approach to a sequential stochastic filter (particle filter).

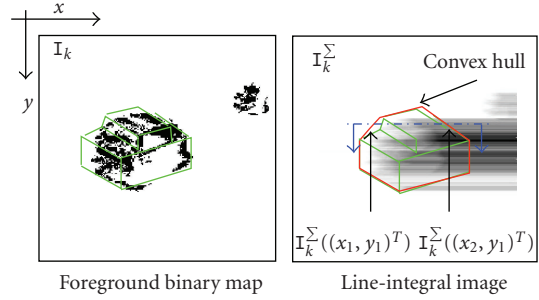


FIGURE 6: Illustration of the vision likelihood computation. The 3D model of the vehicle (shown in green/clear) is reprojected onto the image generated from the background-shape extraction. This projection is approximated by its convex hull (shown in red/dark on the right image). The likelihood calculation proceeds in a line-by-line integral image of the log-likelihood ratio.

4.1. Experimental Details

4.1.1. Initialisation. The first sample of the MCMC must be initialized using priors. We use a data driven method to compute the initial position of the vehicle on the road (x_0, y_0) . A nonparametric blob detector [11] is applied to the background/foreground image I_0 . The initial velocity is provided by a specific sensor. Other parameters are initialized using priors given by vehicle or driver behavior. The dimensions of the geometric model are defined for each vehicle with a stochastic process on width and length parameters and using the likelihood computation (cf. Section 3.2). If dimensions seem to be incoherent, a standard vehicle is chosen.

4.1.2. Proposals. A key point of the method concerns the control of the random walk behavior using proposal distributions. Parameter-specific proposals are defined. Since both lower and upper bounds can be defined for all parameters, we choose proposals according to a Beta distribution

$$P(X_m^* | X_m^{(n)}) \propto (X_m^*)^{\xi_1 - 1} \cdot (1 - X_m^*)^{\xi_2 - 1} \quad (16)$$

parameters ξ_1 and ξ_2 are computed such as the maximum of the distribution is obtained for $X_m^* = X_m^{(n)}$.

4.1.3. Details about the Sequential Method. Behavior of the proposed method is compared with a sequential particle filter. The state vector is defined as $\mathbf{X}_k \doteq (x_y, y_x, \alpha_x, v_x, \delta_x)^T$. Dynamics are controlled by the kinematic bicycle model with a zero centered normal law applied to both the steering angle and velocity variation. Moreover, the likelihood function is slightly modified by removing the normalizing constant. The particle set is resampled at each iteration using a SIR algorithm.

4.2. Results. In order to compare the two methods, a vehicle, equipped with a RTK GPS accurate to within one centimeter, has been used. (A calibration between the GPS reference frame and the camera has been achieved but details are

TABLE 1: Position error (the true position is given by a RTK GPS) for the proposed deferred logical inference method and a sequential particle filter.

Method	Position error (m)	Position std. (m)	Orientation error (degrees)	Orientation std. (degrees)
Sequential filter	0.27	0.26	3.67	3.36
Deferred logical inference	0.20	0.22	1.12	0.97

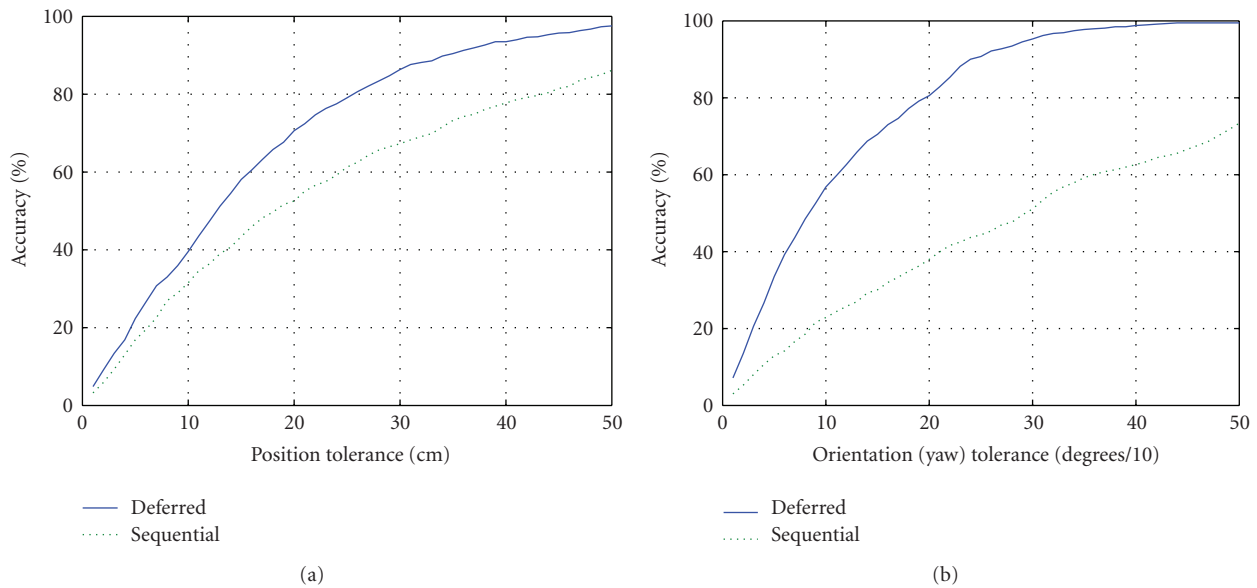


FIGURE 7: Percentage of correct position/orientation related to the tolerance. (a) position absolute tolerance. (b) orientation (yaw angle) absolute tolerance.

not presented in this paper.) This vehicle traveled through the test section 20 times along various trajectories at speeds ranging from 40 to 80 km/hr. The error was quantified as the average distance between each estimated vehicle position and the straight line passing through the two closest GPS points. For each test, at least five vehicle runs were carried out, which enabled deriving a very rough statistic on the recorded measurements. For the tests actually conducted, the vehicle has been tracked in a curve over a distance of approximately 100 m (minimum radius = 130 m).

All the experiments presented here have been done using 200 particles for the two methods.

Table 1 presents the average error and related standard deviations for the two tested methods. The proposed deferred logical inference provides a lower global error than the sequential particle filter.

Figure 7 plots the estimation accuracy as a percentage of correct positions (vertical axis) versus an error tolerance (horizontal axis) for both methods. On the left graph the error tolerance is the position absolute error, ranging from 0 to 50 cm, while on the right graph the error tolerance is the vehicle orientation (yaw angle) absolute error, ranging from 0 to 5 degrees. The curve associated to the proposed method outperforms the sequential particle filter both for the position and the orientation estimation. Moreover, the right graph emphasizes the benefit of the deferred method, which integrates the vehicle and driver priors in every generated

trajectory, thus bringing more time consistency than the sequential method.

Figure 8 illustrates the two methods on a real sequence. Curves on the right column show zooms on local trajectories. The middle column illustrates the image projection of the vehicle position for the sequential method. The right column illustrates the image projection of the vehicle position for the deferred method. It is of high interest to notice the noisy estimation provided by the sequential method, where the estimated trajectory does not seem to match the vehicle kinematic model. The reason for this weak consistency is that the maximum a posteriori estimate may be found on different particles at every time step. In contrast, the spatio-temporal deferred approach ensures faithfulness to the model, thus explaining the observed improvement.

5. Conclusion

We have presented a solution for estimating vehicle trajectories using a single static color camera. A spatio-temporal deferred logical inference solution which takes into account both vehicle kinematics and driver behavior has been proposed, using a stochastic approach to estimate the posterior distribution of the trajectory. By choosing a MCMC, the random walk evolution is controlled by injecting priors on both driver and vehicle behavior and on geometric knowledge about the road. Moreover, a global likelihood

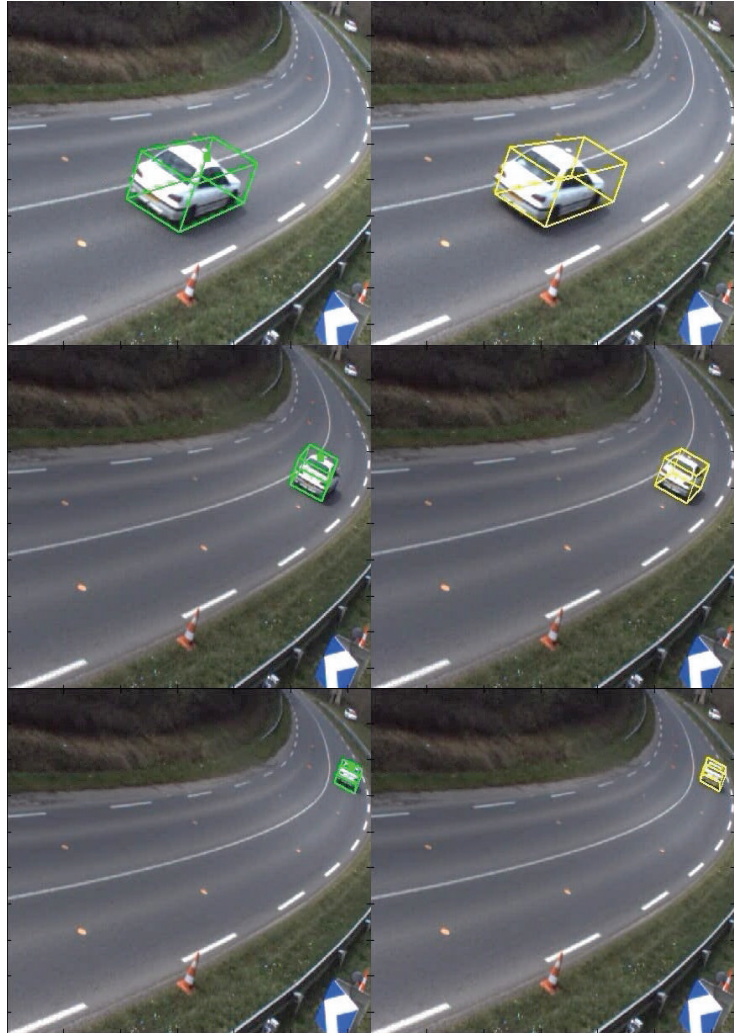
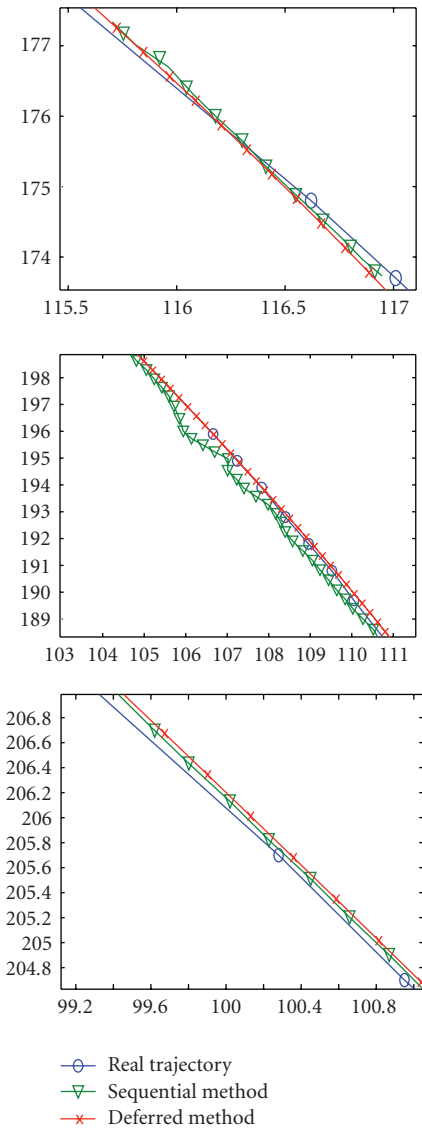


FIGURE 8: Snapshots illustrating the two methods. Left column: zoom on local trajectories. Middle column: the bounding box illustrates the position of the vehicle estimated with the sequential method. Right column: the bounding box illustrates the position of the vehicle estimated with the sequential method.

function using background/foreground binary extraction has been proposed, with an efficient implementation.

Experiments have been achieved to demonstrate that the proposed method outperforms a classic sequential particle filter solution using statistics performed on real video sequences. Two points explain this improvement. First, the spatio-temporal deferred approach processes over the whole data set, thus ensuring time consistency. Second, the spatio-temporal deferred approach, unlike the sequential approach, ensures total faithfulness to the model at any time step, because the maximum a posteriori estimate may be found on different particles at every time step.

The method discussed in this paper is currently operating 24 hours a day with various weather conditions to provide statistics on curve trajectories. For the purpose of covering

the entire curve, the system is composed of three color cameras with very little overlap. The system has successfully analyzed observations recorded under actual traffic conditions over several-day periods.

References

- [1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, 2000.
- [2] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

- [3] M. Isard and J. MacCormick, "BraMBLe: a Bayesian multiple-blob tracker," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 34–41, 2001.
- [4] O. Williams, A. Blake, and R. Cipolla, "A sparse probabilistic learning algorithm for real-time tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, pp. 353–360, Nice, France, 2003.
- [5] S. Avidan, "Support vector tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 184–191, Kauai, Hawaii, USA, 2001.
- [6] S. Avidan, "Ensemble tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, pp. 494–501, IEEE Computer Society, Washington, DC, USA, 2005.
- [7] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.
- [8] Q. Yu, G. Medioni, and I. Cohen, "Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, 2007.
- [9] 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*, vol. 27, no. 11, pp. 1805–1819, 2005.
- [10] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using particles to track varying numbers of interacting people," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 962–969, 2005.
- [11] Y. Goyat, T. Chateau, L. Malaterre, and L. Trassoudaine, "Vehicle trajectories evaluation by static video sensors," in *Proceedings of the 9th IEEE International Conference on Intelligent Transportation Systems (ITSC '06)*, pp. 864–869, 2006.



Preliminary call for papers

The 2011 European Signal Processing Conference (EUSIPCO-2011) is the nineteenth in a series of conferences promoted by the European Association for Signal Processing (EURASIP, www.urasip.org). This year edition will take place in Barcelona, capital city of Catalonia (Spain), and will be jointly organized by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and the Universitat Politècnica de Catalunya (UPC).

EUSIPCO-2011 will focus on key aspects of signal processing theory and applications as listed below. Acceptance of submissions will be based on quality, relevance and originality. Accepted papers will be published in the EUSIPCO proceedings and presented during the conference. Paper submissions, proposals for tutorials and proposals for special sessions are invited in, but not limited to, the following areas of interest.

Areas of Interest

- Audio and electro-acoustics.
- Design, implementation, and applications of signal processing systems.
- Multimedia signal processing and coding.
- Image and multidimensional signal processing.
- Signal detection and estimation.
- Sensor array and multi-channel signal processing.
- Sensor fusion in networked systems.
- Signal processing for communications.
- Medical imaging and image analysis.
- Non-stationary, non-linear and non-Gaussian signal processing.

Submissions

Procedures to submit a paper and proposals for special sessions and tutorials will be detailed at www.eusipco2011.org. Submitted papers must be camera-ready, no more than 5 pages long, and conforming to the standard specified on the EUSIPCO 2011 web site. First authors who are registered students can participate in the best student paper competition.

Important Deadlines:



Proposals for special sessions	15 Dec 2010
Proposals for tutorials	18 Feb 2011
Electronic submission of full papers	21 Feb 2011
Notification of acceptance	23 May 2011
Submission of camera-ready papers	6 Jun 2011

Webpage: www.eusipco2011.org

Organizing Committee

Honorary Chair

Miguel A. Lagunas (CTTC)

General Chair

Ana I. Pérez-Neira (UPC)

General Vice-Chair

Carles Antón-Haro (CTTC)

Technical Program Chair

Xavier Mestre (CTTC)

Technical Program Co-Chairs

Javier Hernando (UPC)

Montserrat Pardàs (UPC)

Plenary Talks

Ferran Marqués (UPC)

Yonina Eldar (Technion)

Special Sessions

Ignacio Santamaría (Universidad de Cantabria)

Mats Bengtsson (KTH)

Finances

Montserrat Najar (UPC)

Tutorials

Daniel P. Palomar

(Hong Kong UST)

Beatrice Pesquet-Popescu (ENST)

Publicity

Stephan Pfletschinger (CTTC)

Mònica Navarro (CTTC)

Publications

Antonio Pascual (UPC)

Carles Fernández (CTTC)

Industrial Liaison & Exhibits

Angeliki Alexiou

(University of Piraeus)

Albert Sitjà (CTTC)

International Liaison

Ju Liu (Shandong University-China)

Jinhong Yuan (UNSW-Australia)

Tamas Sziranyi (SZTAKI -Hungary)

Rich Stern (CMU-USA)

Ricardo L. de Queiroz (UNB-Brazil)

