



HAL
open science

Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder

Yann Goyat, Thierry Chateau, Laurent Trassoudaine

► **To cite this version:**

Yann Goyat, Thierry Chateau, Laurent Trassoudaine. Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder. *Machine Vision and Applications*, 2008, 21 (3), pp.275-286. 10.1007/s00138-008-0160-0 . hal-00596536

HAL Id: hal-00596536

<https://hal.science/hal-00596536>

Submitted on 5 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Y. Goyat · T. Chateau · L. Trassoudaine

Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder

Received: date / Accepted: date

Abstract This article presents a probabilistic method for vehicle tracking using a sensor composed of both a camera and a laser rangefinder. Two main contributions will be set forth in this paper. The first involves the definition of an original likelihood function based on the projection of simplified 3D vehicle models. We will also propose an efficient approach to computing this function using a line-based integral image. The second contribution focuses on a sampling algorithm designed to handle several sources. The resulting modified particle filter is capable of naturally merging several observations functions in a straightforward manner. Many trajectories of a vehicle equipped with a kinematic GPS¹ have been measured on actual field sites, with a video system specially developed for the project. This field input has made it

possible to experimentally validate the result obtained from the algorithm. The ultimate goal of this research is to derive a better understanding of driver behavior in order to assist road managers in their effort to ensure network safety

Keywords Visual tracking · particle filter · sensor fusion

1 Introduction

We will be presenting a vehicle trajectory tracking method that relies upon a sensor composed of a color camera and a one-dimensional scanning laser rangefinder. The objective of this system is to accurately estimate the trajectory of a vehicle traveling through a curve. The research reported herein lies within the scope of an French ANR-PREDIT project ²

The sensor, installed in a curve, is composed of three cameras placed on a tower approximately 5 m high to cover the beginning, middle and end of the curve, in addition to a scanning laser rangefinder laid out parallel to the ground. Since the cameras offer only limited coverage, their observations do not overlap and we will be

Y. Goyat
LCPC
Route de Bouaye
44341 Bouguenais
FRANCE
Tel.: +33-240845852
Fax: +33-240845992
E-mail: yann.goyat@lcpc.fr

T. Chateau, L. Trassoudaine
LASMEA
24, av. des Landais
63177 Aubire Cedex
FRANCE
Tel.: +33-473407660
E-mail: thierry.chateau@lasmea.univ-bpclermont.fr

¹ This device is an absolute localization sensor with a level of accuracy on the order of one centimeter

² See acknowledgments section

considering in the following discussion that the system can be divided into three subsystems, each composed of a camera-rangefinder pair, with the recalibration between each pair performed by means of rigid transformations, which will not be addressed in the present article. The object-tracking procedure is intended to estimate the state of an object at each moment within a given scene, based on a scene observation sequence. Tracking methods can be broken down into two major categories: the first concerns off-line or non-causal tracking, for which the state estimation at a given point in time uses the entire observation sequence [5]. The second category relates to online or causal tracking, for which the state of the object at a given point in time has been estimated as a function of the record of past and current observations and the record of past states [9]. This second category may also encompass the notion of realtime when the period necessary to estimate a state is shorter than the sensor acquisition frequency. The method described herein would be classified as a causal method.

In this article, we are proposing a solution based on a probabilistic formalization of the tracking problem with a stochastic framework (particle filter). The literature, which contains many references in the areas of vision and data merging, proceeds with the recursive time estimation of a state through application of Monte-Carlo methods [1].

The rest of the article has been divided into four parts. The first will discuss the tracking principle, on the basis of a probabilistic model. The next part will focus on the likelihood functions proposed for estimating the weights associated with the particle set. The third part will provide a detailed description of our proposed sampling method (called multi-source sampling). The last

section will then present the measurement campaign conducted for the purpose of quantifying method accuracy and robustness. A large number of trajectories could be estimated and a kinematic GPS was used to determine the actual field values associated with each trajectory.

2 Related Work

Recently, traffic video surveillance has become an important topic in the Intelligent Transport Systems (ITS), so vehicle detection, description, and/or recognition have been an active research field. Most of the solutions assume that the camera is at a high angle. Tracking vehicles from a static camera is challenging for several reasons:

- Outdoor vision tracking solutions must handle with variation of illumination and with shadows.
- Generic model based approaches are complex solutions due to the huge appearance variability of the vehicle class.
- Tracking several vehicles simultaneously implies to be able to handle with mutual occlusions.

Tracking objects from a static camera often uses a background/foreground subtraction [17]. In outdoor environment such method must handle with variations of illumination conditions and shadow. In [23], Stauffer and al. propose a parametric model with a temporal update. The resulting solution deals efficiently with small illumination variations. Detecting shadow can be done by color based analysis [10] and injecting temporal information into the models [18].

Many object descriptions can be used in order to obtain a model of a vehicle. Some approaches are based on a generic model, uses to detect, then track the vehicle, from a luminance [2] or Haar Wavelets based model

[3,21]. 3D wireframes models have been also used but they require to define many models associated to different types of vehicles [26,14,15,24,8]. In [19] a non-rigid 3D model is used into a EM based contour tracking. Recently, Kanhere and al. [12] have proposed an interest point based method to segment and track vehicles.

The method presented here uses a 3D simplified model of a vehicle, projected into the image, and then compared to the background/foreground subtraction map to provide an efficient observation function.

In [11], Kamijo and al. propose a probabilistic model to track multiple vehicles with spatio-temporal Markov random fields. Recently, stochastic methods [13,22] have been presented to handle with realtime multi object tracking. In [25] Yu and al. propose a Monte Carlo Markov Chain method to estimate, from a video, global trajectories of the vehicles.

In this paper, we propose a probabilistic framework to solve the problem of online vehicle trajectory estimation. The trajectory is modeled by a random state vector and the distribution associated to this random vector is approximated in a sequential scheme with a particle filter. Since the trajectory is highly driven by the kinematic model of the vehicle, we propose to inject this model within the dynamics associated to the filter. Moreover, an original data fusion sampling algorithm is proposed to handle with several observation functions.

3 Principle of the Method

This section will set forth the principle behind the trajectory tracking method.

The core of this proposed method (see Fig. 1) consists of a recursive filter that has been formalized stochastically (using a particle filter). The vehicle state is repre-

sented by its corresponding bicycle model. The prediction function adopts the hypothesis of a constant driving angle acceleration and speed. A 3D geometric model of the vehicle, projected onto the image, is then compared with image data described by a probabilistic shape assimilation map. The observation function also comprises a likelihood function that reflects the consistency of telemetric data with respect to the hypothesis. The filter is able to produce, at each iteration, an estimation of the vehicle state (position, heading direction, speed, steering angle).

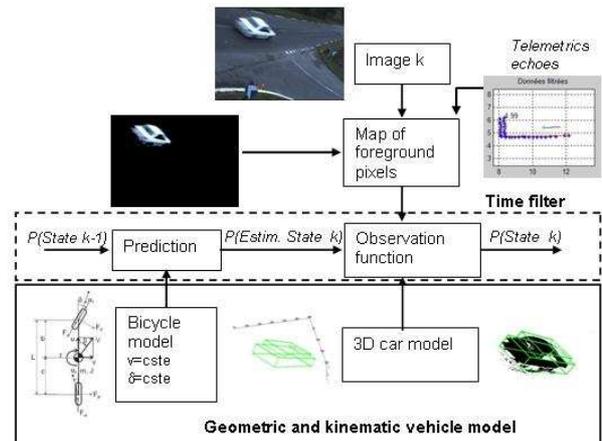


Fig. 1 Block diagram of the proposed method

3.1 Particle Filter

Vehicle trajectory is estimated recursively using a nonlinear filter, whose resolution entails a widespread stochastic method for vision applications: the particle filter. This choice is dictated by the nonlinear nature of the system. Particle filtering [1,16] is based on an estimation of the a posteriori probability density $p(\mathbf{X}_t|\mathbf{Z}_{0:t})$ of state \mathbf{X}_t conditioned by the historical record of measurements $\mathbf{Z}_{0:t}$, at time t , by a set of N weighted particles $\{(\mathbf{X}_t^n, \pi_t^n)\}_{n=1}^N$ with their associated weights. In the case of an observa-

tion stemming from several sources, we are proposing to merge observations intrinsically during the re-sampling phase. The algorithm derived (see Algorithm 1) is a variant of the CONDENSATION algorithm [16]. A weight vector (composed of weights generated from observations of each source) is to be associated with each particle. This multi-source re-sampling method (called MSS) will be further developed in Section 5, page 10.

Algorithm 1 CONDENSATION in the multi-source case

Init : particles $\{(\mathbf{X}_0^n, \mathbf{1}/N)\}_{n=1}^N$ according to the initial distribution \mathbf{X}_0

for $t = 1, \dots, T_{end}$ **do**

Prediction : generation of $\{(\mathbf{X}_t^n, \mathbf{1}/N)\}_{n=1}^N$ from $p(\mathbf{X}_t | \mathbf{X}_{t-1} = \mathbf{X}_{t-1}^n)$

Observation : estimation of the weight vector according to the various sources $\{(\mathbf{X}_t^n, \boldsymbol{\pi}_t^n)\}_{n=1}^N$ with $\boldsymbol{\pi}_t^n \propto \mathbf{p}(\mathbf{Z}_t | \mathbf{X}_t = \mathbf{X}_t^n)$

Sampling : build $\{(\mathbf{X}_{t-1}^n, \mathbf{1}/N)\}_{n=1}^N$ from $\{(\mathbf{X}_0^n, \boldsymbol{\pi}_0^n)\}_{n=1}^N$ using Multi Source Sampling (MSS)

Estimation : $\hat{\mathbf{X}}_t \doteq \frac{1}{N} \sum_{n=1}^N \mathbf{X}_t^n$

end for

Output : The set of estimated states during the video sequence $\{\hat{\mathbf{X}}_t\}_{t=1, \dots, T_{end}}$

Implementing a particle filter necessitates defining three models: 1) a state model that serves to define the kinematic characteristics of the object to be tracked; 2) an evolution model that defines the state of an object at a given point in time depending on the state at the previous point in time; and 3) an observation model that defines a measurement between a state hypothesis and the observations. The state and evolution model are presented in the following sections and the observation model is detailed in section 4.

3.2 Background Extraction

The majority of methods for tracking objects within a static scene introduce a background extraction step, which

consists in a binary case of ascribing each image pixel a "Background/Foreground" class. Most of these assignments are statistical, which forwards the hypothesis that each pixel may be modeled by a random variable capable of assuming either the "Background" or "Shape" state. Stauffer and Grimson [23] proposed employing a parametric Gaussian Mixture Model (GMM) in order to depict the probability density associated with each pixel. It is also possible to use a nonparametric probability density model, such as the one in [9]. This will be the approach adopted herein.

Let \mathbf{I}_t be an image acquired at time t , and $\mathbf{y}_t(\mathbf{u}) \doteq \{y_t(i, \mathbf{u})\}_{i=1,2,3}$ the function that provide information on pixel \mathbf{u} of the CCD sensor according to the three colorimetric components: Red, Green and Blue. Each pixel constitutes a discrete random variable capable of assuming either one of the two following states: 1) "Background" (ω_1), and 2) "Foreground" (ω_2).

3.2.1 Background Model

We are proposing to discretely model the likelihood $p(\mathbf{y}_t(\mathbf{u}) | \omega_1)$ by marginalizing the three color planes, given that: 1) parametric methods in most cases use a Gaussian modeling approach; and 2) the space discretization of parameters is very costly in terms of computation time should the parameters be considered as dependent upon one another. A discretization of $p(\mathbf{y}_t(\mathbf{u}) | \omega_1)$ actually corresponds to N^3 elements if each parameter of $\mathbf{y}_t \in \mathbb{R}^3$ has been sampled using N points. If parameters were independent, $p(\mathbf{y}_t(\mathbf{u}) | \omega_1)$ could be discretized with just $3N$ elements. The hypothesis of independent variables in this instance is not treated rigorously since the three color components are correlated among each other. Nonetheless, this hypothesis often gets adopted in order to reduce

computation time and memory space requirements. Now, let's express $\mathbf{b}_t(\mathbf{u}) \doteq \{b_t(i; \mathbf{u})\}_{i=1, \dots, 3}$ as the histogram associated with color vector $\mathbf{y}_t(\mathbf{u})$ of position \mathbf{u} in the image at time t . It then becomes possible to approximate the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$ by:

$$p(\mathbf{y}_t(\mathbf{u})|\omega_1) = \prod_{i=1}^3 p(y_t(i; \mathbf{u})|\omega_1) \quad (1)$$

with :

$$p(y_t(i; \mathbf{u})|\omega_1) \approx K \sum_{j=1}^N q_t^b(i, j; \mathbf{u}) d(b_t(i; \mathbf{u}) - j), \quad (2)$$

where d represents the Kronecker function and K a constant assigned to standardize the term (dependent on i and \mathbf{u}). $\sum_{j=1}^N q_t^b(i, j; \mathbf{u}) = 1$. $q_t^b(i, j; \mathbf{u})$ is a weight function associated with the discrete model of the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$. The model evolves in sync with this weight function. $q_{t+1}^b(i, j; \mathbf{u})$ is updated with each image by application of the following AR equation:

$$q_{t+1}^b(i, j; \mathbf{u}) = \frac{1}{1 + \alpha} \cdot [q_t^b(i, j; \mathbf{u}) + \alpha \cdot d(b_{t+1}(i; \mathbf{u}) - j)] \quad (3)$$

3.2.2 Shape Model

The shape model is defined by the likelihood of assimilation in the shape category $p(\mathbf{y}_t(\mathbf{u})|\omega_2)$. Even though statistically speaking, some pixels (e.g. those positioned above the horizon line) display a smaller probability of belonging to this shape, we are considering herein that the likelihood of assimilation to the shape category does not depend on the position of the observed pixel. It thus becomes possible to approximate the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_2)$ by :

$$p(y_t(i; \mathbf{u})|\omega_2) \approx K \sum_{j=1}^N q_t^f(i, j) d(b_t(i; \mathbf{u}) - j), \quad (4)$$

The weights $q_t^f(i, j)$ represent a distribution discretization, which has been marginalized *a priori* from the color of the shape object being sought. When no information is available on the model of objects present in the foreground, an equal probability hypothesis is to be considered. In this particular case, the terms $q_t^f(i, j)$ stem from a color histogram of the tracked object, as extracted during an initialization phase.

3.2.3 Probabilistic Shape Assimilation Map

A probabilistic shape assimilation map is generated from the likelihood ratios $p(y_t(i; \mathbf{u})|\omega_2)/p(y_t(i; \mathbf{u})|\omega_1)$. By expressing this ratio in log-likelihood form, it becomes possible to build a pseudo-image of the log-likelihood ratio in which the value associated with the pixel located at coordinates \mathbf{u} is calculated using the following $l_{m,t}(\mathbf{u})$ function :

$$l_{m,t}(\mathbf{u}) \doteq \log(p(y_t(i; \mathbf{u})|\omega_2)) - \log(p(y_t(i; \mathbf{u})|\omega_1)) \quad (5)$$

3.3 State Model

The state model, which identifies the trajectory characteristics to be tracked, must integrate the constraints related to vehicle kinematics. We are proposing to use a bicycle type of model and then recognize and focus on many behavioral and stability properties. The hypotheses inherent in this model are as follows:

- no transfer of lateral load; the vehicle is thus compressed onto a single path,

- no longitudinal transfer,
- no roll or pitch motion,
- tires in a linear configuration,
- constant forward speed V ,
- no aerodynamic effects,
- position control, and
- no effect from suspension and chassis flexibility.

These hypotheses all imply that: vehicle acceleration at all times remains below 0.4 g (i.e. linear operating regime for tires); the steering angle, drift angle, etc. are small; and the ground surface is smooth (no suspension displacement). As an initial approach, it has been considered that vehicles are negotiating the turn at low speed; centrifugal forces are thus negligible and the tires must not develop any lateral forces, i.e.:

- rolling without sliding or drift,
- in order for tires not to slide laterally, the instantaneous center of rotation (ICR) of each tire must lie in the middle of the curve.

Let (x, y) be the selected coordinate system; the kinematic relations governing this model can then be written as follows:

$$\begin{aligned} \dot{x} &= v \cdot \cos \beta \\ \dot{y} &= v \cdot \sin \beta \\ \dot{\beta} &= \frac{v}{L} \cdot \tan \delta \end{aligned} \quad (6)$$

with β representing the vehicle direction and δ the steering angle of the front wheel within the world coordinate system (the extended Lambert II system³) along the vehicle axis.

According to Ackerman's theory at low-speed behavior, the ICR lies at the intersection of the extension to

³ The absolute reference frame used here is the NTF geodesic system in extended Lambert II projection.

the rear wheel axis and the perpendicular to the front wheel plane [6]. The ideal front wheel deviation can be deduced from the construct illustrated in Figure 2 and its angle then written: $\tan \delta = \frac{L}{R}$, with L denoting the vehicle wheelbase and R the curve radius of the road. By adopting the small angle hypothesis $\delta = \frac{L}{R}$, the system state vector is expressed as follows:

$$X_t \doteq (\mathbf{P}_t, \beta_t, \delta_t, v_t)^t \quad (7)$$

with $\mathbf{P}_t \doteq (x_t, y_t)$ representing the vehicle position and v_t the vehicle speed within the world coordinate system.

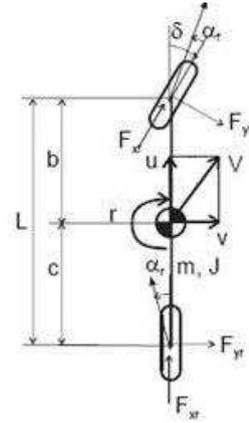


Fig. 2 The bicycle model synthesizes the displacement of a four-wheel vehicle, through the displacement of two wheels whose centers are connected to a rigid axis of length L . Ackerman's theory serves to estimate the steering angle of the front axis of a vehicle traveling at low speed.

3.4 Prediction Model

The bicycle-type kinematic model applied to each particle evolves according to the following nonlinear form:

$$\begin{aligned} x_{t+1} &= x_t + T.v_t \cdot \cos(\beta_t) \\ y_{t+1} &= y_t + T.v_t \cdot \sin(\beta_t) \\ \beta_{t+1} &= \beta_t + T \cdot \frac{v}{L} \cdot \tan \delta_t \\ \delta_{t+1} &= \delta_t + T.b_{\delta} \\ v_{t+1} &= v_t + T.b_a \end{aligned} \quad (8)$$

with $b_{\delta} \sim \mathcal{N}(0, \sigma_{\delta})$ and $b_a \sim \mathcal{N}(0, \sigma_a)$. The two terms δ and a are randomly distributed (according to a Gaussian), while the terms σ_{δ} and σ_a represent respectively the speed deviation range in the front wheel steering angle and the acceleration range, as performed by a standard vehicle during sampling period T .

3.5 Initialization

The filter initialization process consists of ascribing a hypothesis to each filter particle, such that the entire set of particles offers a stochastic representation of the density associated with the particular state, at the initial time. The vehicle position on the road pavement is initialized based on observations generated from the first image. The steering angle is initialized from an *a priori* distribution calculated as a function of characteristics describing the target curve. The heading direction is initialized using an *a priori* distribution calculated with respect to curve position and characteristics. The speed is calculated by the rangefinder at initial position of the vision process, the two sensors being synchronized temporally.

Determining the *a priori* distribution on the vehicle position relies upon a method for detecting cluster cen-

ters, which associates an assimilation probability at the cluster center with each pixel labeled "Shape".

To proceed with this method we take the hypothesis $\alpha \ll 1$ and a pixel becomes "Background" should its value remains stable for an image number $k \gg \alpha$. Then, equation 2 can be simplified and the set of shape pixels \mathcal{F}_t are built by means of thresholding the likelihood function associated with the background assimilation :

$$\mathcal{F}_t \doteq \bigcup_{\mathbf{u} \in \mathbf{I}_t} \{\mathbf{u} | p(\mathbf{y}_t(\mathbf{u}) | \omega_1) < k \cdot \alpha\} \quad (9)$$

Typically $k = 25$ and $\alpha = 0.01$). This probability is obtained from a non-parametric model based on Parzen estimators [4]:

$$\pi_t^n \propto p(\mathbf{Z}_t | \mathbf{X}_t = \mathbf{X}_t^n) \approx \frac{1}{|\mathcal{F}_t|} \sum_{\mathbf{u} \in \mathcal{F}_t} \varphi(\mathbf{p}_t^n, \mathbf{u}) \quad (10)$$

where $\varphi(\mathbf{p}_t^n, \mathbf{u})$ is a Gaussian kernel defined by:

$$\varphi(\mathbf{p}_t^n, \mathbf{u}) = \frac{1}{(2\pi) \cdot |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{p}_t^n - \mathbf{u})^t \Sigma^{-1} (\mathbf{p}_t^n - \mathbf{u}) \right] \quad (11)$$

The covariance matrix Σ of the kernel depends on the estimated vehicle projection size in the image plane. This dimension, in pixels, is a function of the vehicle position in the image; as the vehicle approaches the camera, its size grows in the image. Figure 3 shows the assimilation probability distribution at the cluster center for the background/shape extraction example.

The complexity of this method lies in $O(n^2)$. In [7], we have proposed a stochastic approach in order to reduce the cost of running this algorithm.

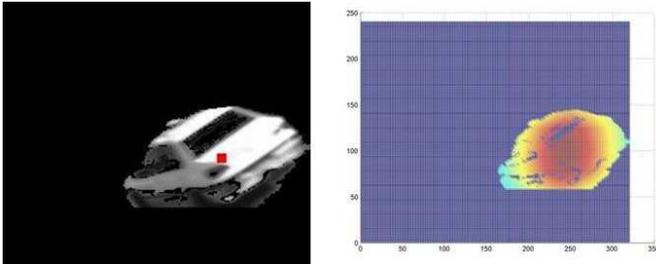


Fig. 3 Illustration of the method for searching the "Shape" point cluster center: Each pixel of the set of "Shape" points displays an assimilation probability at the cluster center. The right-hand side figure is an image of the probability of belonging to the cluster center; as the pixel color becomes redder, the associated weight rises.

The position part (\mathbf{P}_t^n) of the filter is initialized as a function of Equation (10). Figure 4 displays an example of an initialized particle set. To improve this initialization, the filter is iterated on the first image, by noising just the position and heading as well as by applying the observation model described in the next section.

The shape model (marginalized color histogram \mathbf{q}_t^f) can then be built from the pixels belonging to the estimated vehicle.

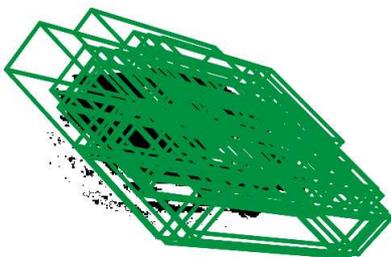


Fig. 4 Example of particle distribution during the initialization phase (shown in green).

4 Observation Function

The proposed observation function utilizes the measurements provided by the two available sensors (i.e. color camera and 1D scanning laser rangefinder). A likelihood function must be defined for each sensor.

4.1 Vision Likelihood Function

The likelihood function proposed herein relies upon a simplified three-dimensional geometric model of the vehicle, 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 $\mathcal{M}^{(R_0)} = \{M_i^{(R_0)}\}_{i=1,\dots,N_M}$ represent the model's set of cube vertices ($N_M = 8 \times P_M$), expressed within a coordinate system associated with model R_0 . This coordinate system is selected such that the 3 axes all lie in the same direction as that of the coordinate system associated with scene R_w . For a given particle \mathbf{X}_t^n , the likelihood (weighting) calculation is determined as the product of the shape/background likelihood ratios located inside the vehicle model projection in the image.

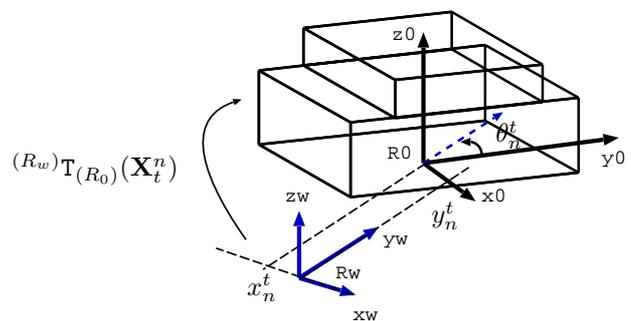


Fig. 5 Example of the simple three-dimensional geometric model used for a vehicle: it is composed of two cubes. The coordinate system associated with the cube and the other coordinate system associated with the scene are related according to pure translation. The plane (Oxy) of the world coordinate system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

This computation performed for each particle consumes valuable 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. Each model point is projected onto the image via the following equation:

$$\tilde{\mathbf{m}}_i \propto \mathbf{C}_c \cdot {}^{(R_w)}\mathbf{T}_{(R_0)}(\mathbf{X}_t^n) \cdot \tilde{\mathbf{M}}_i^{(R_0)} \quad (12)$$

with $\tilde{\mathbf{M}}$ homogeneous coordinates associated with point M ; \mathbf{C}_c is the camera projection matrix, and ${}^{(R_w)}\mathbf{T}_{(R_0)}(\mathbf{X}_t^n)$ the homogeneous transformation matrix between the world coordinate system and the system associated with the 3D model (cf. figure 5). This matrix, which depends on \mathbf{X}_t^n , may be simply written as:

$${}^{(R_w)}\mathbf{T}_{(R_0)}(\mathbf{X}_t^n) = \begin{pmatrix} \cos(\theta_t^n) & -\sin(\theta_t^n) & 0 & x_t^n \\ \sin(\theta_t^n) & \cos(\theta_t^n) & 0 & y_t^n \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (13)$$

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

Let $\mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{X}_t^n) = \{\mathbf{e}_i\}_{i=1, \dots, N}$ ($\mathbf{e}_i = (x_i^e, y_i^e)$ as coordinates of \mathbf{e}_i in the image plane) be the list of convex hull points⁴. We will now define $\mathcal{E}_t^n \doteq \mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{X}_t^n)$ in order to streamline notations. The likelihood calculation may be performed efficiently by use of a line-by-line integral image derived from the log-likelihood ratios calculated in Equation (5) page 5. Since the concept of integral image is often used in computer vision to increase performances, it can not be used directly here because

the shapes are not combination of rectangles. So, we propose to extend the concept of integral image to line-by-line integral image. The resulting image is build using integration along the current raw of the image (each raw is independent) and the likelihood calculation is:

$$l_{I,t}((x, y)^T) = \sum_{i=1}^x l_{m,t}((i, y)^T) \quad (14)$$

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

$$\begin{aligned} \mathcal{E}_t^n = \{ & (x_1^e, y^e), (x_2^e, y^e), \\ & (x_3^e, y^e + 1), (x_4^e, y^e + 1), \dots \\ & (x_{N-1}^e, y^e + N/2), (x_N^e, y^e + N/2)\} \end{aligned} \quad (15)$$

Convex hull coding within the set \mathcal{E}_t^n 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 envelope is calculated in the integral image by application of the following relation:

$$\begin{aligned} a(\mathcal{E}_t^n) = \sum_{j=1}^{N/2} [& 2 \cdot (l_{i,t}(\mathbf{e}_{2j}) - l_{i,t}(\mathbf{e}_{2j-1})) \\ & - (x_{2j}^e - x_{2j-1}^e)] \end{aligned} \quad (16)$$

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.

The vision weight associated with each particle is directly correlated with $a(\mathcal{E}_t^n, \mathbf{I}_I)$ by means of the following expression:

⁴ Calculation of the convex hull will not be developed in this article; the calculation procedure is conducted using a classical algorithm with a complexity expressed in $O(N \cdot \log N)$.

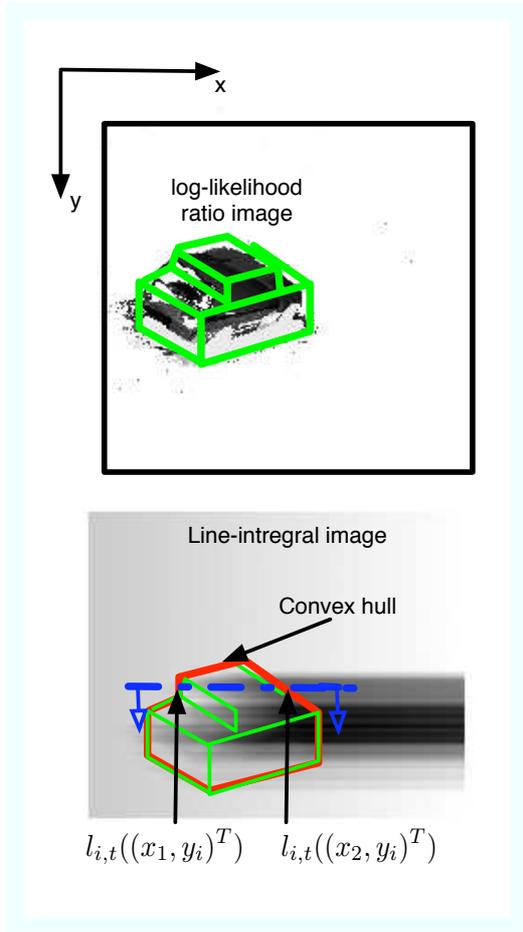


Fig. 6 Illustration of the vision likelihood computation. The 3D model of the vehicle (shown in green) is re-projected onto the image generated from the background-shape extraction. This projection is approximated by its convex hull (shown in red on the lower image). The likelihood calculation proceeds in a line-by-line integral of the log-likelihood ratio.

$$\pi_{v,t}^n \propto p(\mathbf{Z}_t | \mathbf{X}_t = \mathbf{X}_t^n) \doteq \max(0, a(\mathcal{E}_t^n)) \quad (17)$$

4.2 Telemetric Likelihood Function

The telemetric sensor provides, on a horizontal plane, the distance from the first obstacle with a resolution of 1 degree. The intersection of the laser beam with a vehicle yields a set of points; for each particle, a model is now available for calculating the laser echoes generated from the intersection of the beam with the simplified 3D model

of the projected vehicle in the world coordinate system. The likelihood associated with the telemetric observation can then be calculated from a modified Hausdorff distance (denoted d_h) between the actual and simulated echoes (cf figure 7) using the following expression:

$$\pi_{l,t}^n \propto p(\mathbf{Z}_t | \mathbf{X}_t = \mathbf{X}_t^n) \doteq \exp(-\lambda_t d_h) \quad (18)$$

where λ_t is an adjustment parameter (typical value is 20).

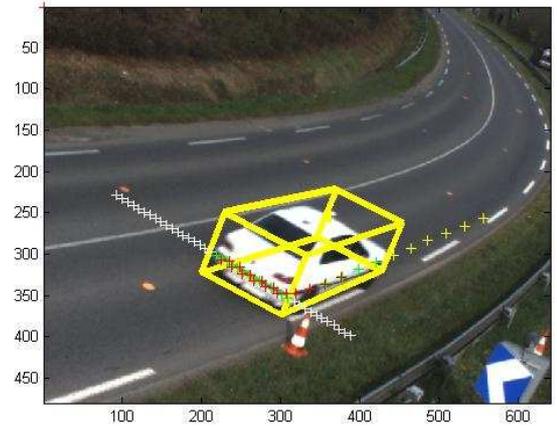


Fig. 7 Illustration of the distance computation between simulated echoes on the 3D model of the cube (shown in green) and the actual echoes (red).

5 Multi-Source Sampling

When the observation function is composed of terms stemming from multiple sources, it becomes necessary to lay out a strategy for combining these terms. A classical approach consists of assembling a weighting function composed of a combination of observations generated from each source. The choice of merge operator often takes place empirically. We propose an alternative to

this approach, which calls for performing the combination step as part of the particle filter re-sampling step.

Upon observation, the filter may be represented by a set of N particles with an associated weight vector: $\{\mathbf{X}_t^{(i)}, \boldsymbol{\pi}_t^i\}_{i=1, \dots, N}$. The weight vector $\boldsymbol{\pi}_t^i$, of size M in correspondence with the number of sources, is composed of the individual particle weight estimated by each source. To facilitate comprehension, the notations contained in the remainder of this section will omit the time index t .

5.1 Principle

Multi-source sampling entails generating a new particle set, by implementing a three-step approach:

1. M particles are sorted (one for each source) according to an importance sampling strategy associated with each source (*importance sampling*). The output of this step consists of a set of M candidate particles along with their respective weight vector $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^{(i)}\}_{i=1, \dots, M}$:
2. A confidence vector, of dimension M , is built using likelihood ratios estimated for each candidate particle (this calculation will be detailed in the following discussion).
3. The chosen particle is derived from a selection performed among candidate particles by applying an importance sampling strategy on the confidence vector.

These three steps are then repeated a total of N times in order to obtain the complete set.

5.2 Confidence Vector

In the following section, we will describe in detail the second step of this multi-source sampling procedure, which

is aimed at building a confidence vector associated with the set of candidate particles. The underlying principle consists of calculating the product of likelihood ratios between weights of the same source, for each pair of candidate particles.

As an example, in the three-sensor case, three candidate particles are drawn; for each particle, a likelihood ratio product is calculated, which for the first candidate particle yields:

$$r_1 \doteq \frac{\pi_1^1}{\pi_1^2} \cdot \frac{\pi_1^1}{\pi_1^3} \quad (19)$$

where π_j^i represents the j th component of vector $\boldsymbol{\pi}^i$. In the case of a blind source (i.e. the values returned by the observation function associated with this source are constant), those likelihood ratios in which the source is present equal one and do not exert any influence on the calculation of terms r_i . In a general case, it is preferable to introduce log ratios, which makes it possible to compute a vector \mathbf{l}_r , i.e. the log of \mathbf{r} , featuring coefficients r_i , as indicated in the following expression:

$$\mathbf{l}_r \doteq M \begin{pmatrix} \mathbf{1}_{(1 \times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_1} - \frac{1}{M} \sum_{i=1}^M \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \mathbf{1}_{(1 \times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_2} - \frac{1}{M} \sum_{i=1}^M \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \dots \\ \mathbf{1}_{(1 \times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_M} - \frac{1}{M} \sum_{i=1}^M \mathbf{l}_{\boldsymbol{\pi}_i} \right) \end{pmatrix} \quad (20)$$

where $\mathbf{l}_{\boldsymbol{\pi}_i}$ is the log of vector $\boldsymbol{\pi}_i$ and $\mathbf{1}_{(1 \times M)}$ is a matrix composed of a single line and M columns all containing one.

By setting $\mathbf{C}_\pi \doteq \frac{1}{M} \sum_{i=1}^M \mathbf{1}_{\pi_i}$, \mathbf{l}_r , the following may be written:

$$\mathbf{l}_r = M \begin{pmatrix} \mathbf{1}_{(1 \times M)} (\mathbf{1}_{\pi_1} - \mathbf{C}_\pi) \\ \mathbf{1}_{(1 \times M)} (\mathbf{1}_{\pi_2} - \mathbf{C}_\pi) \\ \dots \\ \mathbf{1}_{(1 \times M)} (\mathbf{1}_{\pi_M} - \mathbf{C}_\pi) \end{pmatrix} \quad (21)$$

Confidence vector \mathbf{c} is obtained by normalizing \mathbf{r} through a coefficient C_c that results in the sum of its elements equaling one.

$$\mathbf{c} \doteq C_c \cdot \exp(\mathbf{l}_r) \quad (22)$$

Algorithm 2 Multi-source sampling

Input : Particle set and associated weight vector $\{\mathbf{X}^{(i)}, \pi^i\}_{i=1, \dots, N}$, M sources

for $n = 1$ to N **do**

- Choose M candidate particles on the basis of $\{\mathbf{X}^{(i)}, \pi^i\}_{i=1, \dots, N}$ and build $\{\mathbf{X}^{*(j)}, \pi^{*(j)}\}_{j=1, \dots, M}$ where $\mathbf{X}^{*(j)}$ is derived from an *importance sampling* drawn on source j weights;

- Calculate vector \mathbf{l}_r based on Equation 21, and then calculate confidence vector $\mathbf{c} \doteq C_c \cdot \exp(\mathbf{l}_r)$

- Select the designated particle $\mathbf{X}^{e(n)}$ from among the candidate particles by proceeding with an *importance sampling* drawing.

end for

Output : Particle set $\{\mathbf{X}^{e(i)}\}_{i=1, \dots, N}$ composed of the selected particles.

Figure 8 demonstrates how the multi-source sampling method works for two distinct scenarios, by comparing method behavior with that displayed when sampling from both a weight function composed of the weight product from each source and another weight function composed of the sum of weights from each source. In

the first scenario (left-hand column), source 1 is a blind source (i.e. returns a constant measurement) and source 2 is unimodal. In this case, the blind source must not intervene to disturb the sampling, and the new set must fit the source 2 set. It should be noted that in the case of sampling using the sum of weights from two sources, the blind source actually deteriorates the resultant set. On the other hand, the other two types of sampling behave quite well. The second scenario illustrates the situation of two dissonant sources (unimodal, yet centered on a different point). The set of particles generated from these two sources must serve to create two separate modes around the two dissonant hypotheses. It can be stated that in the case of product-type sampling, neither of the modes actually gets conserved. In contrast, the other two types of sampling yield results consistent with the desired distribution.

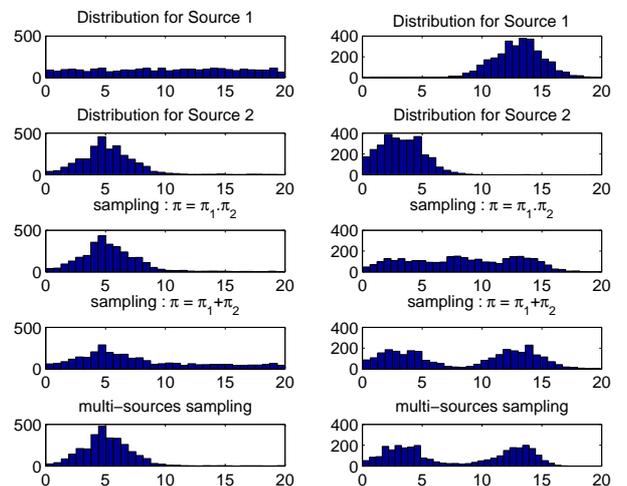


Fig. 8 Illustration of multi-source sampling method operations for 2 different scenarios (one in each column). In the left column, from top to bottom: the first two curves represent the source response (observation). The third shows the result of a weight-based importance sampling, calculated as the product of weights from the two sources. In the fourth curve, this weight-based importance sampling result is generated by summing weights from the two sources. The last curve then displays the result of our proposed multi-source sampling method.

6 Experimental Validation

This section will present the experimental campaigns carried out in order to validate the trajectory tracking method.

The video system developed as part of the SARI project has made it possible to continuously record three color 640 x 480 video streams (cf. Fig. 9), at a frequency of 30 fps, over a several-day period along with data recorded by a scanning telemetry sensor. The calibration between sensors enables expressing all measurements within a common absolute coordinate system tied to the GPS sensor coordinate system, which provides the actual situation in the field.

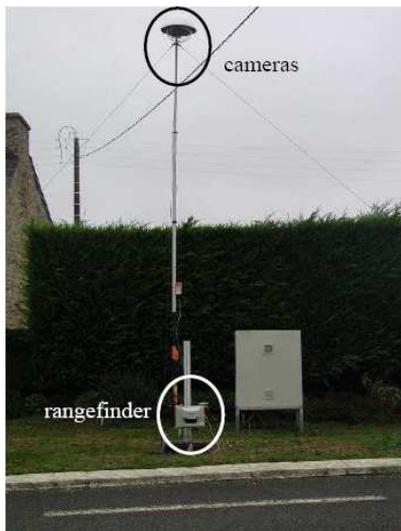


Fig. 9 The video system is composed of three color cameras, capable of being easily deployed onsite.

In order to assess the level of measurement accuracy, a Peugeot 406 type vehicle, equipped with a kinematic GPS accurate to within a centimeter, has been used.

This vehicle traveled through the test section 20 times along various trajectories at speeds ranging from 40 to 80 km/hr. The results listed here are aimed at examining system accuracy as a function of travel speed. Com-

Speed <i>km/hr</i>	Vision ave/std	Rangefinder ave/std	Sensor merge ave/std
40	0.25/0.18	0.65/0.54	0.17/0.10
60	0.19/0.16	0.72/0.67	0.09/0.06
80	0.18/0.15	0.33/0.22	0.14/0.10

Table 1 Precision for the right camera (error and standard deviation of error between the estimation and actual field value output by a kinematic GPS) of the proposed method, for several travel speeds. It may be remarked that merging the two sensors serves to considerably improve estimation precision. Actual field values have been furnished by a kinematic GPS type of sensor.

puted trajectories were then compared with trajectories estimated by a vision-only approach, a rangefinder-only approach and an approach merging the two sources. 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 passes 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. The vehicle model used for these purposes comprises a single cube. Finally, according to the empirical results of the Table 2, for every tracking, we decide to use 150 particles.

particules number	50	100	150	200	300
average	0.61	0.22	0.13	0.14	0.14
standard deviation	0.61	0.24	0.14	0.14	0.14

Table 2 Precision for the right camera (error and standard deviation of error between the sensor merge estimation and GPS) of the proposed method, for several particles number.

Figure 10 shows a tracking example from the test campaign carried out as part of this research. The model corresponding to vehicle localization has been re-projected onto the current image. For the left-hand column, the method makes exclusive use of data stemming from the vision sensor. The middle column reflects results based solely on telemetric data. The green crosses correspond

with the simulated laser firings re-projected onto the image, while red crosses indicate the actual re-projected laser data. Lastly, for the right-hand column, the method employs both sensors. It is observed that at the beginning of the curve, the vision estimation tends to be rather poor; this is due to the presence of another vehicle crossing the tracked vehicle, thus producing noise in the background/shape extraction. As the vehicle moves away from the sensor, the telemetric approach reveals its flaws, caused by a decrease in the number of laser echoes returned by the vehicle. The approach merging these two sensors allows taking advantage of the accurate information provided by the laser sensor upon entering the curve and then that provided by the vision sensor when exiting the curve.

Table 1 presents the average error and related standard deviation vs. vehicle speed. It can be noticed that the level of accuracy varies only slightly with respect to the tested speed range. Moreover, the two-sensor merge serves to significantly improve this accuracy, which lies on the order of 15 cm with a standard deviation of approximately 10 cm.

Note that these results come from the right camera, but are extensive to the others cameras. The calibration phase is same for all the system and the reference frame is absolute. Figure 11 shows the three sensors merging with the ground truth.

7 Conclusion - Prospects

This article has set forth a method for estimating vehicle trajectories by use of a sensor composed of a color camera and a 1D scanning laser rangefinder. Based on a time monitoring approach formalized by a particle filter, the algorithm has output, at every point in time, an estima-

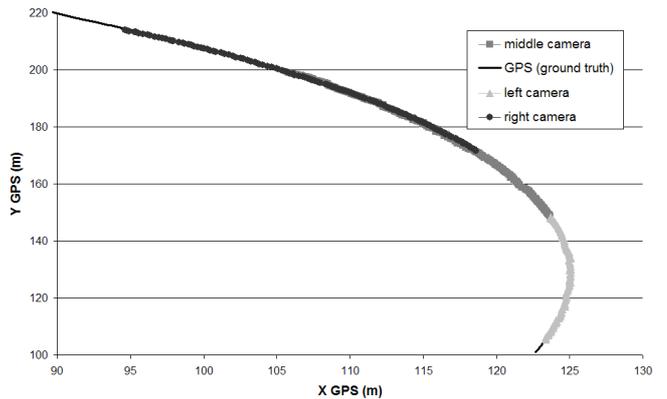


Fig. 11 representation of the three trajectories calculated by each camera and the ground truth associated.

tion of vehicle state (position, direction, speed, driving angle). We have proposed an original likelihood function that can be evaluated efficiently through use of a line-by-line integral image. We also introduced an alternative to importance sampling, as classically employed in particle filters. This method has made it possible to implicitly merge observations stemming from different sources. The resulting behavior proves to be better than the classical techniques that rely upon combining weights using algebraic operators.

The experiments conducted have enabled quantifying the precision associated with the proposed approach, thanks to generating actual field values (using a kinematic GPS accurate to within 1 cm). The accuracy obtained lies on the order of 20 cm. These tests have also served to demonstrate the contribution offered by merging vision with telemetry.

Upcoming research will primarily be focused on improving the background subtraction method, which has the potential to introduce bias due to the presence of a vehicle shadow projected onto the pavement [20].

The method discussed in this article is currently operating within the scope of an ANR-PREDIT project. For the purpose of covering the entire curve, the system is

composed of three color cameras with very little overlap plus a scanning laser rangefinder, which has successfully analyzed the observations recorded under actual traffic conditions over several-day periods.

Acknowledgements This research work was conducted as part of the PREDIT project entitled "SARI" (French acronym for Automated Road Monitoring to Inform drivers and facility managers). This project is intended to help lead to significant reductions in the number of accidents due to loss of contact with the roadway or loss of vehicle control, by means of better informing drivers of the upcoming difficulties facing them on the road (<http://www.sari.prd.fr>).

References

1. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* **50**(2), 174–188 (2002). URL citeseer.nj.nec.com/arulampalam02tutorial.html
2. Avidan, S.: Support vector tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*. Hawaii (2001)
3. Chateau, T., Gay-Belille, V., Chausse, F., Lapresté, J.: Real-time tracking with classifiers. In: *WDV 2006 - WDV Workshop on Dynamical Vision at ECCV2006, Grazz* (2006)
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification, Second Edition*. Wiley-Interscience (2001)
5. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2007)
6. Gillespie, T.: *Fundamentals of vehicle dynamics*. Society of Automotive Engineers (SAE) (1992)
7. Goyat, Y., Chateau, T., Malaterre, L., Trassoudaine, L.: Vehicle trajectories evaluation by static video sensors. In: *ITSC06 2006 - 9th International IEEE Conference on Intelligent Transportation Systems* (2006)
8. Haag, M., Nagel, H.: Combination of edge element and optical flow estimate for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision* **35**(9), 295–319 (1999)
9. Isard, M., MacCormick, J.: Bramble: A bayesian multiple-blob tracker. In: *Int. Conf. Computer Vision*, vol. 2, pp. 34–41 (2001)
10. Ivanov, Y., Bobick, A., Liu, J.: Fast Lighting Independent Background Subtraction. MIT Media Laboratory (1997)
11. Kamijo, S., Ikeuchi, K., Sakauchi, M.: Vehicle tracking in low-angle and front view images based on spatio-temporal markov random fields. In: *8th World Congress on Intelligent Transportation Systems (ITS)* (2001)
12. Kanhere, N.K., Pundlik, S.J., Birchfield, S.T.: Vehicle segmentation and tracking from a low-angle off-axis camera. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. San Diego, USA (2005)
13. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (**in press**) (2005)
14. Kim, Z.W., Malik, J.: Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In: *International Conference on Computer Vision (ICCV)*, pp. 521–528 (2003)
15. Koller, D., Dandilis, K., Nagel, H.H.: Model based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision* **10**(3), 257–281 (1993)
16. M. Isard, A. Blake: Condensation – conditional density propagation for visual tracking. *IJCV : International Journal of Computer Vision* **29**(1), 5–28 (1998)
17. Magee, D.R.: Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing* **22**(2), 143–155 (2004)
18. Mikik, I., Cosman, P., Kogut, G., Trivedi, M.: Moving shadow and object detection in traffic scenes. In: *International Conference on Pattern Recognition*. Barcelona, Spain (2000)
19. Pece, A., Worrall, A.: Tracking with the em contour algorithm. In: *ECCV European Conference on Computer Vision*, vol. 1, pp. 3–17. Copenhagen (2002)
20. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. *Comput.*

-
- Vis. Image Underst. **95**(2), 238–259 (2004). DOI <http://dx.doi.org/10.1016/j.cviu.2004.03.008>
21. Schneiderman, H., Kanade, T.: A statistical model for 3d object detection applied to faces and cars. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2000)
 22. Smith, K., Gatica-Perez, ., , Odobez, J.: Using particles to track varying numbers of interacting people. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
 23. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8) (2000)
 24. Tan, T.N., Baker, K.D.: Efficient image gradient based vehicle localization. *IEEE Transactions on Image Processing* **9**(8), 1343–1356 (2000)
 25. Yu, Q., Medioni, G., Cohen, I.: Multiple target tracking using spatio-temporal markov chain monte carlo data association. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
 26. Zhao, T., Nevatia, R.: Car detection in low resolution aerial image. In: ICCV, pp. 710–717 (2001)

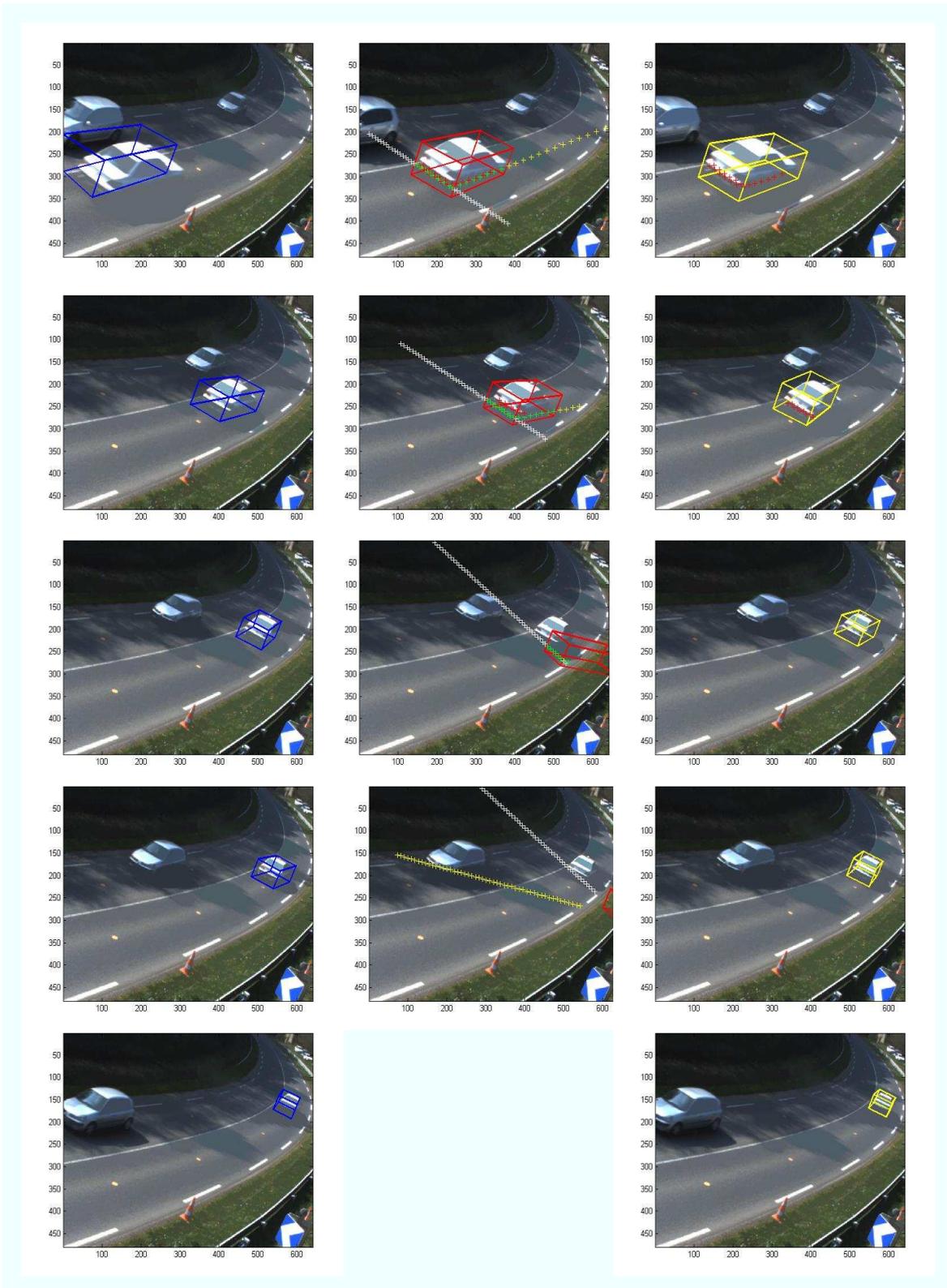


Fig. 10 Tracking example drawn from the testing campaign performed as part of this research (for the right camera). The model corresponding to vehicle localization is re-projected onto the current image. For the left-hand column, the method is confined solely to data stemming from the vision sensor. For the middle column, the method uses only the telemetric data, with the green crosses corresponding to simulated laser firings re-projected onto the image and red crosses indicating actual re-projected laser data. The right-hand column depicts method application by combining the two sensors.