A Particle-Filtering Approach for Vehicular Tracking Adaptive to Occlusions

Jacob Scharcanski, *Senior Member, IEEE*, Alessandro Bof de Oliveira, *Member, IEEE*, Pablo G. Cavalcanti, and Yessenia Yari

***Abstract*—In this paper, we propose a new particle-filtering approach for handling partial and total occlusions in vehicular tracking situations. Our proposed method, which is named adaptive particle filter (APF), uses two different operation modes. When the tracked vehicle is not occluded, the APF uses a normal probability density function (pdf) to generate the new set of particles. Otherwise, when the tracked vehicle is under occlusion, the APF generates the new set of particles using a Normal-Rayleigh pdf. Our approach was designed to detect when a total occlusion starts and ends and to resume vehicle tracking after disocclusions. We have tested our APF approach in a number of traffic surveillance video sequences with encouraging results. Our proposed approach tends to be more accurate than comparable methods in the literature, and at the same time, it tends to be more robust to target occlusions.**

***Index Terms*—Particle filter (PF), total occlusion, traffic surveillance, vehicular tracking.**

I. Introduction

IN RECENT YEARS, image-based traffic surveillance has become a very important research area. The utilization of traffic surveillance includes, but it is not restricted to, detection and avoidance of vehicle accidents, detection of vehicles with unusual behavior, and gathering of statistical information about roadway traffic, among several other applications [1], [2]. Vehicle tracking has an important role in image-based traffic surveillance since it provides information about the position of a tracked vehicle in time. Therefore, the tracking process should not be interrupted; otherwise, the vehicle dynamics information may be lost or become incorrect, and consequently, handling occlusions is an important challenge in this area.

There are different approaches proposed for vehicle tracking in the literature. Stefano and Viarani proposed a block-matching approach for tracking a moving vehicle [3]. The vehicle is detected where there are neighboring blocks with similar motion patterns, and the vehicle tracking is performed by matching regions with similar size and motion patterns in adjacent frames. Unfortunately, this method tends to be affected by the background clutter and can present erroneous results when vehicles are spatially close to each other or have similar velocities and/or overlap in the video sequence. An alternative approach proposed by Rhee *et al.* [4] identifies shapes resembling vehicles among the segmented regions and tracks these shapes along adjacent frames (instead of individual blocks). This method is limited in terms of handling partial occlusions and may confuse targets when the vehicles have similar shapes.

Bayesian stochastic filters have been used with success in generic tracking applications [5]–[13] and in vehicular tracking applications [14]–[18]. In this case, stochastic models are used to represent the target position uncertainty and noise. Koller *et al.* [14] proposed to use Kalman-filtering concepts in vehicle tracking. First, the regions corresponding to moving vehicles are segmented along the video sequence, and then, the contours of the moving regions are detected using snakes. After that, the Kalman filter is used to estimate the corresponding vehicle contour in the subsequent frames of the video sequence. The vehicle position is defined as the center of the snake control points. This method assumes a linear (i.e., planar and translational) vehicle motion pattern and assumes that vehicle contours are convex. The Kalman filter in this method assumes linear dynamics for the tracked vehicle, and the uncertainty is modeled by Gaussians [18], [19]. These constraints can limit the method applicability, especially if the vehicle velocity changes often.

The particle-filtering approach is more flexible and robust than Kalman filtering since it does not assume *a priori* any type of dynamic behavior, or uncertainty model, for the tracked vehicle and can succeed, even in partial occlusions and severe background clutter conditions [6]–[8], [19]. Meier and Ade [15] proposed a vehicle-contour-tracking method based on the well-known particle filter (PF) condensation algorithm [20], which is able to handle the visual clutter and partial occlusions. Another particle-filtering approach for tracking vehicles was proposed by Bouttefroy *et al.* [16]. In their approach, the tracking feature is the color histogram of the tracked vehicle, instead of its contour. The color histogram is invariant to vehicle rotations and translations and is also robust to occlusions and noise [5]. Some particle-filtering approaches have been proposed to improve on these tracking methods. Deutcher *et al.* [10] proposed a simulated annealing PF (SAPF), and Maggio and Cavallaro [17] developed a hybrid mean-shift (MS) and particle-filtering approach. Their goal was to deal better with partial occlusions and the background clutter. Xiong and Debrunner [18] proposed a PF with two tracking features: 1) color histogram

Manuscript received January 29, 2010; revised October 14, 2010; accepted November 14, 2010. Date of publication December 17, 2010; date of current version February 18, 2011. The review of this paper was coordinated by Dr. M. S. Ahmed.

The authors are with the Institute of Informatics, Federal University of Rio Grande do, 91501-970 Porto Alegre-RS, Brazil (e-mail: jacobs@inf.ufrgs.br; aboliveira@inf.ufrgs.br; pgcavalcanti@inf.ufrgs.br; ydyari@inf.ufrgs.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2099676

and 2) edge-based shape detection. Their approach offers good tracking results, even with significant color changes, poor lighting, and/or background clutter edges. Kanhere and Birchfield [20] proposed a method based on the Lucas–Kanade tracker algorithm, using 3-D scene information to handle perspective and partial occlusion problems in vehicular tracking. Lou *et al.* [21] also proposed a 3-D approach for tracking vehicles, where vehicles are described by 3-D models and tracked using an extended Kalman-filtering method. This approach relies on the 3-D vehicle model to compensate for possible losses of vehicle information in partial occlusions. Zhang *et al.* [22] proposed a distinct scheme to detect and handle total occlusions in vehicular tracking. Their method assumes that partially occluded or overlapping vehicles have convex shapes and different velocities. In their approach, the motion and shape convexity of the vehicle are analyzed to detect and handle partial occlusions. Total occlusions are analyzed with a bidirectional occlusion reasoning method, assuming that vehicles under total occlusion remain with their velocities constant and assuming that the vehicles detected after an occlusion are those that became originally occluded. However, these assumptions may not be valid in several vehicle-tracking situations found in practice. Handling total occlusions is challenging for most vehicle-tracking approaches based on particle filtering. The occurrence of a vehicle total occlusion interrupts the tracking process since information about the tracked vehicle is momentarily lost in the video sequence. To avoid this process disruption, the tracking method should detect when the vehicle becomes occluded and when the vehicle becomes disoccluded. After becoming disoccluded, the vehicle-tracking process should resume. Typical examples of total occlusions are vehicles occluded by bridges, or large traffic signs, which are common in traffic surveillance. In this paper, we propose a new particle-filtering approach called adaptive particle filtering, which tracks vehicles in video sequences with partial and/or total occlusions. Our proposed PF tends to be robust to occlusions and is able to continuously track moving vehicles (i.e., without being disrupted by occlusions), as indicated by our tests with outdoor video sequences of complex scenes [23] containing total and partial occlusions. The obtained experimental results are promising. This paper is organized as follows: Section II shows a brief revision of target tracking with particle filters. Section III describes the proposed adaptive particle-filtering method, and the experimental results are discussed in Section IV. The conclusions are in Section V, as well as some ideas for future work.

II. Review of Target Tracking Using Particle Filtering

Considering the vehicle tracking context, let us define $\vec{S}_t = (x, y)$ as the position of the tracked vehicle in image coordinates (x, y) at time t and Ξ_t as an observation variable at time t as well.[1] Due to the measurement uncertainty, the position of the tracked vehicle is modeled by the *a posteriori* probability density function (pdf) $p(\vec{S}_t|\Xi_t)$, and the vehicle position can be estimated by calculating the expected value

$$\hat{\vec{S}}_t = E\left[p(\vec{S}_t|\Xi_t)\right]. \tag{1}$$

Usually, we do not know in advance the pdf $p(\vec{S}_t|\Xi_t)$; however, we can adaptively approximate this pdf while tracking the vehicle using the particle-filtering method, as discussed next. The *a posteriori probability* is approximated using a set of N local samples (i.e., particles) $\vec{s}_t^{(i)} = (x, y)$, where $i = 1, 2, 3, \ldots, N$. The observation variable Ξ_t (see footnote 1) is now estimated based on a set of local observations $\{\xi_t^{(i)}\}$, and each local observation is associated with a particle position $\vec{s}_t^{(i)}$. The particle position $\vec{s}_t^{(i)}$ is updated in time using a stochastic differential equation [6] that describes the particle dynamics

$$\vec{s}_t^{(i)} = \vec{s}_{t-1}^{(i)} + \Gamma_t \tag{2}$$

where Γ_t is a random variable generated by a sampling pdf. Equation (2) can be written in terms of probabilities as (3), shown below [19]. Therefore, a set of particles is drawn[2] from a given probability distribution [see (3)], which will later provide a discrete sampling approximation of the unknown *a posteriori*. It is assigned a weight, or importance $\pi_t^{(i)}$, to each particle position $\vec{s}_t^{(i)}$ according to its likelihood of representing a known measurable target property [see (4), shown below]. Using the set of particles and their associated weights $\{\vec{s}_t^{(i)}, \pi_t^{(i)}\}$, we can estimate the position of the tracked vehicle with a Monte Carlo Integration method [19], [24] [see (5), shown below]. This method uses a set of sampling particles to approximate the probability of occurrence of the target measurable features in the particle sampled spatial locations (represented for the particle weight) and then estimate the target location by an integration method

$$\vec{s}_t^{(i)} \sim p\left(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}\right) \tag{3}$$

$$\vec{\pi}_t^{(i)} = p\left(\xi_t^{(i)}|\vec{s}_t^{(i)}\right) \tag{4}$$

$$\hat{\vec{S}}_t = E\left[p\left(\vec{s}_t^{(i)}|\xi_t\right)^{(i)}\right] = \sum_{i=1}^{N} \tilde{\pi}_t^{(i)} \vec{s}_t^{(i)} \tag{5}$$

where

$$\tilde{\pi}_t^{(i)} = \frac{\pi_t^{(i)}}{\sum_{j=1}^{N} \pi_t^{(j)}}. \tag{6}$$

III. Our Proposed Adaptive Particle Filter

The proposed adaptive PF (APF) was designed to be robust to partial and/or total occlusions during vehicle tracking. This PF has two tracking modes (i.e., operation modes), as will be described. When the tracked vehicle is not occluded, the APF uses

[1] The observation variable Ξ_t represents a measurable property (e.g., color or texture) of the tracked vehicle located in the position $\vec{S}_t$ at time t.

[2] Drawing a new particle $\vec{s}_t^{(i)}$ from the distribution $p(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$, given the existing particle $\vec{s}_{t-1}^{(i)}$, is here denoted by $\vec{s}_t^{(i)} \sim p(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$; see (3).

the regular tracking mode (without occlusions). In this operation mode, the new particles $\vec{s}_t^{(i)}$ are generated using a Bivariate Normal pdf (i.e., $p(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}) = p_N(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$; see details in Section III-A). However, when a total occlusion is detected, the APF goes into the tracking under occlusion mode (i.e., $p(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}) = p_{NR}(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$; see details in Section III-B). In this last case, the new particles are generated by a 2-D pdf, i.e., a joint Normal-Rayleigh bivariate pdf. The Rayleigh function is oriented along the direction of greater motion activity of the tracked vehicle, and the Normal function is oriented along the orthogonal direction with respect to the Rayleigh. The asymmetry of the joint Normal-Rayleigh pdf generates a "drift" effect, i.e., moving the particles away from the location where the occlusion started.

Algorithm 1: APF algorithm pseudocode

1: $\{\vec{s}_0^{(i)}\} \sim N(\vec{S}_0, \Sigma)$, where $i = 1, \ldots, N$;
2: $\Pi_0 = 1$; and $\Pi_0^{up} = 1$;
3: **for** $t = 1$ to T **do**
4: **if** Π_{t-1} and $\Pi_{t-1}^{up} \geq \tau$ **then**
5: $(\vec{s}_{t-1}^{(i)*}, \tilde{\pi}_{t-1}^{(i)*}) = \text{resampling}(\vec{s}_{t-1}^{(i)}, \tilde{\pi}_{t-1}^{(i)})$ (see Algorithm 2);
6: $\vec{s}_{t-1}^{(i)} \leftarrow \vec{s}_{t-1}^{(i)*}$, where $i = 1, \ldots, N$.
7: $\vec{s}_t^{(i)} \sim p_N(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$, where $i = 1, \ldots, N$;
8: **else**
9: $\vec{s}_t^{(i)} \sim p_{NR}(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$, where $i = 1, \ldots, N$;
10: **end if**
11: $\vec{\pi}_t^{(i)} = p(\xi_t^{(i)}|\vec{s}_t^{(i)})$ [see (25)], where $i = 1, \ldots, N$;
12: $\tilde{\pi}_t^{(i)} = (\pi_t^{(i)} / \sum_{j=1}^{N} \pi_t^{(j)})$;
13: $\hat{\vec{S}}_t = \sum_{i=1}^{N} \tilde{\pi}_t^{(i)} \vec{s}_t^{(i)}$;
14: $\vec{\Phi}_t^* = hist(\hat{\vec{S}}_t)$;
15: $\vec{\Phi}_t = \alpha\vec{\Phi}_t^* + (1-\alpha)\vec{\Phi}_{t-1}$;
16: $\Pi_t = 1 - d_B(\vec{\Phi}_t, \vec{\Phi}_0)$;
$\Pi_t^{up} = 1 - d_B(\vec{\Phi}_t, \vec{\Phi}_{t-1})$.
17: **end for**

The APF algorithm is shown in Algorithm 1, and its steps are explained next. In the initialization of Algorithm 1, the N particles are drawn from a Bivariate Normal pdf centered at the initial position of the tracked vehicle $\vec{S}_0$ (see step 1). This initial vehicle position at time $t = 0$ can be manually defined by the user or provided by a vehicle detection method. The function $hist(.)$ returns the local color histogram in RGB color space and is requantized from 8 to 3 bits per channel, extracted from a window W of $(2w_x + 1) \times (2w_{xy} + 1)$ pixels, which is centered at the position defined by the function argument (the position $\vec{S}_0$ in the initialization and, afterward, $\vec{S}_t$). The variables Π_t and Π_t^{up} are initially set to "1" at time $t = 0$ (step 2). They represent the similarity of the local color histograms of the tracked vehicle at time t ($\vec{\Phi}_t$) with the local color histograms at time $t = 0$ ($\vec{\Phi}_0$) and $t - 1$ ($\vec{\Phi}_{t-1}$), respectively (see Algorithm 1, step 16). This color histogram similarity is measured using the Hellinger distance $d_B(.)$, which is written in terms of the Bhattachyyara coefficient [25]. In step 4 of Algorithm 1, the occurrence of a total occlusion is detected by comparing the values of Π_t and Π_t^{up} with a threshold value τ. It shall be observed that the local color histogram similarity decreases in total occlusions since the vehicle color information at time t, i.e., $\vec{\Phi}_t$, is lost. If no occlusion is detected, the regular tracking mode follows steps 5–7 of Algorithm 1. In step 5 of Algorithm 1, the resampling algorithm is called (see Algorithm 2). The resampling algorithm is used to eliminate particles with small weights, which are unlikely to represent target (internal) positions, since the local color histograms $\vec{\phi}_t^{(i)}$ at these locations are not similar to the target color histogram $\vec{\Phi}_t$. The resampling algorithm is detailed in Section III-D. Step 7 of Algorithm 1 mentions the pdf used for propagating the particles $\vec{s}_t^{(i)}$ in regular tracking mode (see details in Section III-A). In this case, the discrete *a posteriori probability* function $p(\vec{S}_t|\Xi_t)$ is approximated using a set of particles propagated with a Bivariate Normal probability function $\{\vec{s}_t^{(i)}\} \sim p_N(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$. If an occlusion is detected, the APF enters in occlusion tracking mode. Step 9 of Algorithm 1 mentions the discrete pdf used to propagate particles in occlusion tracking mode. This pdf $p_{NR}(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$ is presented in detail in Section III-B. The weights $\pi_t^{(i)}$ are calculated using the likelihood probability function mentioned in step 11 of Algorithm 1. This likelihood function and the observed variable $\xi_t^{(i)}$ are described in Section III-C. In step 12 of Algorithm 1, the particle weights $\pi_t^{(i)}$ are normalized because they now compose the discrete probability distribution, which is used to approximate the unknown *a posteriori* $p(\vec{S}_t|\Xi_t)$. In step 13 of Algorithm 1, the estimated position of the tracked vehicle is calculated using the Monte Carlo integration method [19], [24]. Finally, in step 15 of Algorithm 1, the local color histogram of the tracked vehicle is updated by interpolation of the estimated target color histogram $\vec{\Phi}_t^*$ and its previous color histogram $\vec{\Phi}_{t-1}$ at time $t - 1$, considering a learning rate parameter α [11]. The complexity of our APF proposed method is similar to the PF [5]. The PF uses a single pdf to propagate the particles, but our approach uses two pdfs, i.e., one for the regular tracking mode and another for the occlusion tracking mode.

A. Sampling Function Without Occlusions (Regular Tracking Mode)

Vehicles that are not occluded are tracked in regular tracking mode. The pdf used for propagating the particles $\vec{s}_t^{(i)}$ is the Bivariate Normal distribution function [19], with correlation $\rho = 0$, and the pdf in step 7 of Algorithm 1 is written as

$$p_N\left(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}\right) = \frac{1}{2\pi\sigma_w^2} exp\left(-\frac{1}{2}\left[\frac{x'^2 + y'^2}{\sigma_w^2}\right]\right) \quad (7)$$

where

$$x' = x_t^{(i)} - x_{t-1}^{(i)} \quad \text{and} \quad y' = y_t^{(i)} - y_{t-1}^{(i)} \quad (8)$$

and σ_w is the standard deviation of the sampling function. (σ_w was experimentally set.) We use a symmetrical function because we do not know *a priori* the dynamic behavior of the tracked vehicle; therefore, we assume no preferential

orientation for the sampling function. In (8), x' and y' are stochastically generated using the pdf in (7), and given x', y', $x_{t-1}^{(i)}$, and $y_{t-1}^{(i)}$, we obtain $x_t^{(i)}$ and $y_t^{(i)}$.

B. Sampling Function Under Occlusions (Occlusion Tracking Mode)

As mentioned before, the tracking color information is unavailable during vehicle occlusions. Therefore, if we use a symmetrical sampling function to generate the new particles $\vec{s}_t^{(i)}$, the occlusion location is resampled, and the estimation of particle weights $\pi_t^{(i)}$ will tend to return small weight values (see Algorithm 1, step 11).[3] The resemblance between $\vec{\Phi}_{t-1}$ and the original reference $\vec{\Phi}_0$ will tend to decrease during occlusions (see Algorithm 1, step 15). In addition, the estimation of the tracked vehicle position $\hat{\vec{S}}_t$ will tend to return inaccurate position values [see Algorithm 1, step 13, and (25)]. To avoid getting stuck in this situation for long, we use an asymmetrical sampling function, i.e., a joint Normal-Rayleigh pdf, which propagates new particles along the directions k_1 and k_2 [see (11)]. We define its Rayleigh and Normal components, respectively, as

$$R(k_1|\sigma_R) = \frac{k_1}{\sigma_R^2} exp\left(-\frac{k_1^2}{2\sigma_R^2}\right) \tag{9}$$

where σ_R is the parameter of the Rayleigh distribution, and

$$N(k_2|\mu_{k_2}, \sigma_N) = \frac{k_2}{\sqrt{2\pi\sigma_N^2}} exp\left(-\frac{(k_2-\mu_{k_2})^2}{2\sigma_N^2}\right) \tag{10}$$

where σ_N is the Normal standard deviation, and μ_{k_2} is the mean of the Normal distribution along the k_2 direction. (We assume $\mu_{k_2} \equiv 0$.) Thus, we define the joint Normal-Rayleigh (see Algorithm 1, step 9) as

$$p_{NR}\left(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}\right) = R(k_1|\sigma_R)N\left(k_2|\mu_{k_2}, \sigma_N\right) \tag{11}$$

or yet

$$p_{NR}\left(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)}\right) = \frac{k_1}{\sqrt{2\pi\sigma_N^2}\sigma_R} exp\left(-\frac{k_1^2}{2\sigma_N^2} - \frac{k_2^2}{2\sigma_R^2}\right) \tag{12}$$

where k_1 is defined as the principal vehicle motion direction, k_2 is the orthogonal direction, and σ_N and σ_R are experimentally adjusted. The Rayleigh pdf spreads along the principal vehicle motion direction (k_1), and the Normal spreads along the orthogonal direction k_2. The principal motion direction is estimated by initially calculating the mean velocity of the tracked vehicle $\vec{V}$ and its components $[V_x, V_y]$, where $\vec{V} = \hat{\vec{S}}_t - \hat{\vec{S}}_{t-1}$, and $\hat{\vec{S}}_0 = (0,0)$. Then, the principal motion direction k_1 is chosen as the direction of the mean velocity vector $\vec{V}$. (The direction orthogonal to $\vec{V}$ is k_2.) As mentioned before, the Normal pdf spreads along k_2, which enlarges the particle sampling area and helps in decreasing random noise (caused by measurement uncertainty) in the estimates of the mean vehicle velocity $\vec{V}$. Using this asymmetrical sampling function $p_{NR}(\vec{s}_t^{(i)}|\vec{s}_{t-1}^{(i)})$, we introduce a "*drift*" effect that will create more particles $\vec{s}_t^{(i)}$ along the direction in which the vehicle is moving, which can be particularly useful in occlusion situations. Next, we confirm this "*drift*" effect by deriving the particle propagation equations along the directions k_1 and k_2. Let us write the particle propagation equation in (2) as stochastic differential equations along k_1 and k_2

$$k_{1,t}^{(i)} = k_{1,t-1}^{(i)} + \Gamma_{R,t} \tag{13}$$

$$k_{2,t}^{(i)} = k_{2,t-1}^{(i)} + \Gamma_{N,t} \tag{14}$$

where $\Gamma_{R,t}$ and $\Gamma_{N,t}$ are Rayleigh and Normal random variables, respectively (i.e.,. $\Gamma_{R,t} = k_{1,t}^{(i)} - k_{1,t-1}^{(i)}$ and $\Gamma_{N,t} = k_{2,t}^{(i)} - k_{2,t-1}^{(i)}$). Now, let us observe the evolution of (13) at discrete times $t = 1, 2, \ldots, T$ [producing (15) and (16), shown below]. Substituting the value of $k_{1,1}^{(i)}$ in (16) by its value in (15), we can calculate $k_{1,2}$ as in (17), shown below

$$k_{1,1}^{(i)} = k_{1,0}^{(i)} + \Gamma_{R,1} \tag{15}$$

$$k_{1,2}^{(i)} = k_{1,1}^{(i)} + \Gamma_{R,2} \tag{16}$$

$$k_{1,2}^{(i)} = k_{1,0}^{(i)} + \Gamma_{R,1} + \Gamma_{R,2} \tag{17}$$

and for an arbitrary time T $(T \geq 1)$, we can infer by recursively substituting the value of $k_{1,t-1}^{(i)}$ in the subsequent equation for $k_{1,t}^{(i)}$ that $k_{1,T}^{(i)}$ is given by

$$k_{1,T}^{(i)} = k_{1,0}^{(i)} + \Gamma_{R,1} + \Gamma_{R,2} + \cdots + \Gamma_{R,T}. \tag{18}$$

Recall that, actually, particles stochastically propagate, but we can calculate the expected location of a particle $E[k_{1,T}]$ at time T[4] as

$$E\left[k_{1,T}^{(i)}\right] = E\left[k_{1,0}^{(i)}\right] + E[\Gamma_{R,1}] + E[\Gamma_{R,2}] + \cdots + E[\Gamma_{R,T}] \tag{19}$$

which can be simply written as

$$E\left[k_{1,T}^{(i)}\right] = k_{1,0}^{(i)} + (T-1)\sqrt{\pi/2}\sigma_R \tag{20}$$

where σ_R is the parameter of Rayleigh distribution. Equation (20) shows that the expectation $E[k_{1,T}^{(i)}]$ is given by the particle position at time $t = 0$ (e.g., the initial particle position $k_{1,0}^{(i)}$ when entering in occlusion mode) plus a "*drift*" term $(T-1)\sqrt{\pi/2}\sigma_R$. In other words, the particle is propagated along the k_1 direction proportionally to a term that depends on Rayleigh parameter σ_R and time T. Similarly, the expected particle

[3]The similarity between the target color histogram $\vec{\Phi}_{t-1}$ and each one of the local color histograms $\vec{\phi}_t^{(i)}$, calculated based on the vicinity of each particle $\vec{s}_t^{(i)}$, tends to be small in occlusions [see (25)].

[4]$E[k_{1,T}]$ is the expectation given all possible locations to where a particle could have propagated and then be found at time T, considering only the k_1 direction.

location $E[k_{2,T}^{(i)}]$ along the k_2 direction can be calculated at time T by

$$E\left[k_{2,T}^{(i)}\right] = E\left[k_{2,0}^{(i)}\right] + E[\Gamma_{N,1}] + E[\Gamma_{N,2}] + \cdots + E[\Gamma_{N,T}] \tag{21}$$

which results in

$$E[k_{2,T}] = k_{2,0}^{(i)} + 0. \tag{22}$$

Since the expected value of a Normal pdf with zero mean is zero, the expected value $E[k_{2,T}]$ is the same as the initial position $k_{2,0}$.

C. Using Color as an Observable Feature

In this paper, the local color distributions are used as local observable features to calculate the weight $\pi_t^{(i)}$ associated with each particle [5], [12]. The local color distribution has interesting properties, such as translational and rotational invariance, aside from being a robust feature for nonrigid objects, even in partial occlusion situations [12], [13]. The local color distribution $\vec{\phi}_t^{(i)}$ is represented by a local color histogram in RGB color space, requantized from 8 to 3 bits per channel (to reduce the number of color histogram bins from 2^{24} to 512), extracted from a window W of $(2w_x + 1 \times (2w_y + 1)$ pixels, and centered at the particle $\vec{s}_t^{(i)}$, i.e., $\vec{\phi}_t^{(i)} = hist(\vec{s}_t^{(i)})$. The window size W is defined by the user to contain the tracked vehicle. In our experiments with APF, the window size is a constant for all tracking sequences. The color histogram of the tracked vehicle $\vec{\Phi}_{t-1}$ is extracted at the estimated vehicle position at $t-1$ (i.e., $\vec{\Phi}_{t-1} = hist(\hat{\vec{S}}_{t-1})$), representing the target color distribution to be matched at time t. The similarity $\xi_t^{(i)}$ between the color histogram $\vec{\phi}_t^{(i)}$ and the reference tracked vehicle color histogram $\vec{\Phi}_{t-1}$ is calculated using the Hellinger distance [see (23), shown below], which is written in terms of the Bhattachyyara coefficient [(24)] [25]

$$\xi_t^{(i)} \equiv d_B\left(\vec{\phi}_t^{(i)}, \vec{\Phi}_{t-1}\right) = \sqrt{1 - \rho\left[\vec{\phi}_t^{(i)}, \vec{\Phi}_{t-1}\right]} \tag{23}$$

$$\rho\left[\vec{\phi}_t^{(i)}, \vec{\Phi}_{t-1}\right] = \sum_{u=1}^{m} \sqrt{\vec{\phi}_{t,u}^{(i)} \vec{\Phi}_{t-1,u}} \tag{24}$$

where u represents the uth bin of the m-bins color histogram.

The weight $\pi_t^{(i)}$ is the similarity between histograms (see Algorithm 1, step 11), and it is defined as

$$\pi_t^{(i)} = 1\sqrt{2\pi\sigma_v} exp\left(-\frac{d_B\left(\vec{\phi}_t^{(i)}, \vec{\Phi}_{t-1}\right)}{2\sigma_v^2}\right) \tag{25}$$

where σ_v is experimentally set. The value of σ_v is selected to make sure that particles lying on the tracked vehicle carry more weight (or importance), and their weights rapidly decrease as the distance to the vehicle centroid increases. In our experiments, we used $\sigma_v = 1.5$. When both color distributions $\vec{\phi}_t^{(i)}$ and $\vec{\Phi}_{t-1}$ are similar, the distance $d_B(\vec{\phi}_t^{(i)}, \vec{\Phi}_{t-1}) \cong 0$, $\pi_t^{(i)}$ assumes larger values, and the particles at those locations are weighted more. These particle locations with larger weights are more likely to correspond to spatial locations on the tracked vehicle.

D. Resampling Function

The resampling function is used in regular mode (without occlusion) to eliminate particles at locations where the local color histogram is not similar to the target color histogram $\vec{\Phi}_t$. Thus, the resampling function improves the estimate of the discrete *a posteriori probability* distribution $\{\tilde{\pi}_t^{(i)}\}$. The resampling function eliminates particles with small weights and draws new particles at time t based on the particles that had larger weights at time $t-1$ while keeping constant the number of particles at N [19].

Algorithm 2 Resampling function: $(\vec{s}_t^{(i)*}, \tilde{\pi}_t^{(i)*}) =$ resampling$(\vec{s}_t^{(i)}, \tilde{\pi}_t^{(i)})$;

1: $\Omega^{(j)} = \sum_{k=1}^{j} \tilde{\pi}_t^{(k)}$, where $j = 1, \ldots, N$;
2: **for** $i = 1$ to N **do**
3: $r = rand(0, 1)$;
4: $J = argmin_j(\Omega^{(j)} | r \leq \Omega^{(j)})$;
5: **if** $J \neq null$ **then**
6: $\vec{s}_t^{(i)*} \leftarrow \vec{s}_t^{(J)}$, and $\pi_t^{(i)*} \leftarrow \tilde{\pi}_t^{(J)}$;
7: **end if**
8: **end for**
9: $\tilde{\pi}_t^{(h)*} = (\pi_t^{(h)*} / \sum_{l=1}^{N} \pi_t^{(l)*})$;
10: return to Algorithm 1.

The particle resampling algorithm is described in Algorithm 2, which uses the importance sampling method [19] to resample the set of particles $\{\vec{s}_t^{(i)}\}$. In this process, particles with smaller weights $(\tilde{\pi}_t^{(i)})$ have a higher probability of being eliminated [because particles with smaller weights are not sufficiently representative to be used in the discrete approximation of the *a posteriori* $p(\vec{S}_t | \Xi_t)$]. In step 1 of Algorithm 2, the cumulative weights $\Omega^{(j)}$ of the particles $j = 1, \ldots, N$ are calculated. In step 3 of Algorithm 2, r is an auxiliary variable that is set using function $rand(0, 1)$, which is an uniform random number generator (i.e., $rand(0, 1) \subset [0, 1]$). When we calculate the particle cumulative probability (weight) $\Omega^{(j)}$, we nonuniformly distribute the particles $\vec{s}_t^{(i)}$ in the interval [0, 1] (i.e., $\tilde{\pi}_t^{(j)} \subset [0, 1]$). If a particle has a larger weight $\tilde{\pi}_t^{(j)}$, it will occupy a large portion of the interval [0, 1], and it is more likely to be chosen. In step 4, the function $J = argmin_j(\Omega^{(j)} | r \leq \Omega^{(j)})$ is used to select with higher probability the particles with larger weights (i.e., particles occupying larger portions of [0, 1] and for which $r \leq \Omega^{(j)}$). This process is iterated N times, where N is the number of particles in the set $\{\vec{s}_t^{(j)}\}$. The particles are chosen with reposition, i.e., the same particle can be chosen more than once since it will actually be used only as a *seed* for the propagation of new particles. In step 6 of Algorithm 2, the particles are resampled, generating a new set of particles $\{\vec{s}_t^{(i)*}, i = 1, \ldots, N\}$ and weights $\{\pi_t^{(i)*}, i = 1, \ldots, N\}$. In

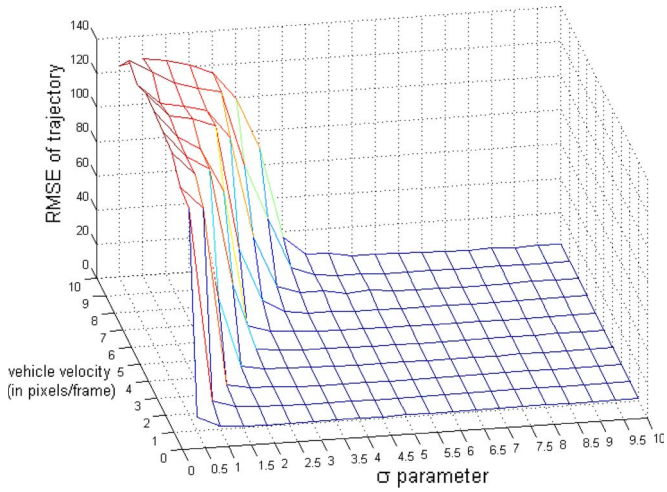


Fig. 1. Plot of the RMSE between the real trajectory of the vehicle and the tracking results for different combinations of σ and vehicle velocity values.

step 9 of Algorithm 2, the new weights $\{\pi_t^{(h)*}, h = 1, \ldots, N\}$ are renormalized, because the new set of particle weights, which is generated by the resampling algorithm, must add to 1 since this set forms a discrete probability distribution. We use this probabilistic resampling algorithm, instead of a deterministic resampling algorithm (which would choose only the particles with larger weights), because the *a posteriori probability* distribution usually is multimodal. In this case, a deterministic resampling approach could provide an unsatisfactory estimate of the discrete *a posteriori probability* distribution, because the particles could tend to concentrate in the larger mode of the multimodal distribution. This resampling algorithm is a probabilistic algorithm to choose a new set of particles from an original set, where a particle in the original set can be selected more than once.

IV. Experimental Results

In this section, we present and discuss the analysis of our APF algorithm and its sensitivity to detect occlusions, as well as our experimental results. The parameters σ_w, σ_R, and σ_N are used to control the pdf used to generate new particles. These parameters are related to the vehicle speed, and in our experiments, we used $\sigma \equiv \sigma_w = \sigma_R = \sigma_N$, and $\sigma \in [5, 10]$. We created artificial videos, where we control the velocity of the vehicle in pixel per frame. The APF was used to track the vehicle in different velocities, using different σ values. Fig. 1 shows the root mean square errors (RMSEs) for our tracking results. We can observe in Fig. 1 that there is a relation between the velocity of the vehicle and the value of σ, such that the tracking RMSE is minimized. This velocity-σ relation is approximately $\sigma = velocity/2$. We also observed for σ values above $\sigma = velocity/2$ that the RMSE also provides low values. This effect is caused by the resampling algorithm, which eliminates particles that are more distant from the tracked vehicle since these particles tend to have smaller weights and smaller contributions to estimate the *a posteriori probability* of the vehicle location. (Particle scattering increases with σ.) The sensitivity of the proposed APF method to overcome total occlusions was analyzed using artificial videos, where we can vary the width of the total occlusion. In this experiment, the vehicle velocity is kept constant. The occlusion width is varied from 0 (without occlusion) to 120 pixels. Fig. 2 shows a frame of the artificial video with a total occlusion (i.e., dark vertical bar). In this example, the width of the total occlusion is 30 pixels. Our proposed APF method was able to track the vehicle in all the analyzed cases (with the occlusion width varying from 0 to 120 pixels). Fig. 3 shows the Π_t values calculated for each frame of the artificial videos and for different occlusion widths. The Π_t values are in [0, 1] and are associated to gray levels according to the scale bar [see Fig. 3 (right)]. We can also observe that, as the occlusion width increases (in pixels), the number of frames processed in occlusion mode also increases (and vehicle tracking is not interrupted). During occlusions, when the occlusion widths increase, the similarity values Π_t tend to decrease. After the vehicle becomes disoccluded, the regular tracking mode resumes.

Fig. 2. Example of artificial video used to analyze the sensibility of the APF method to handle total occlusions. In this example, the total occlusion has a width of 30 pixels.

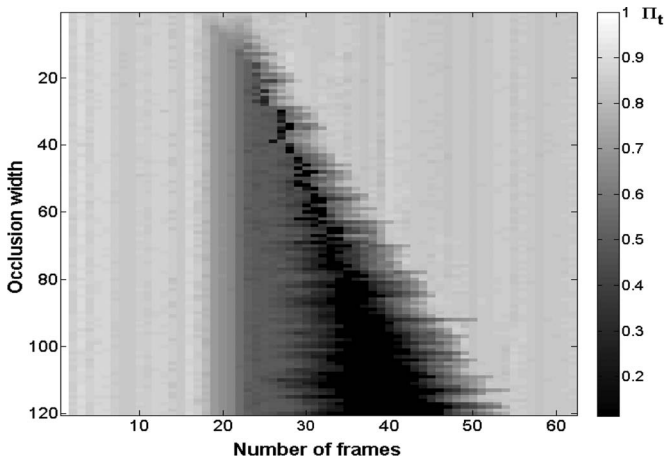


Fig. 3. Plot showing as gray levels the Π_t values for each frame of different artificial videos (x-axis) as a function of the occlusion width (y-axis). The Π_t values are presented as gray levels between zero and one [0, 1] indicated in the gray scale bar at the right of the plot.

In Fig. 4, we illustrate the detection of total occlusions by thresholding Π_t and Π_t^{up} using an artificial video example. The threshold used in the experiments with artificial videos was $\tau = 0.8$ (dashed line). The dotted and continuous lines indicate the

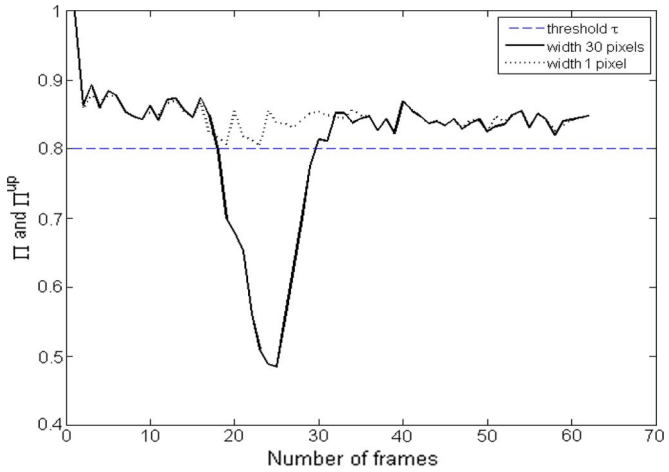


Fig. 4. Example of Π and Π^{up} values and the threshold value used in our experiments. In this example, we use a threshold $\tau = 0.8$ (dashed line) to detect total occlusions. The dotted line (artificial video with occlusion width of 1 pixel) shows that the threshold was not reached by the APF, and it did not enter in the occlusion mode. The continuous line shows the tracking results for the artificial video with an occlusion width of 30 pixels.

similarity between the local color histogram of tracked vehicle at time $t = 0$ and t (i.e., Π_t) and t and $t-1$ (i.e., Π_t^{up}) in the case of total occlusions of widths 1 and 30 pixels, respectively. When the occlusion width is 1 pixel, the similarity measures (Π_t and Π_t^{up}) are not small enough to reach the threshold and activate the APF occlusion mode. Occlusions are more reliably detected when the occlusion width increases.

Twenty different real video sequences, which were obtained with traffic surveillance cameras [23] and [26], were used in our tests. Here, we illustrate our results for a typical video (i.e., Video3) to show our APF tracking results in total occlusion situations. Table I shows the type of occlusions present in each video sequence, as well as the type of tracked vehicle (i.e., car, sport utility vehicle, or truck). The videos frame rate is 15 frame/s, and the color system is the RGB with 8 bits per channel for all videos. Fig. 5 shows the tracking results for Video3, where a vehicle is successfully tracked with the proposed APF, despite the presence of partial and total occlusions. In Fig. 5(a), the vehicle is tracked without occlusions, and therefore, the APF uses the Normal pdf to generate the new particles. In Fig. 5(b), the vehicle has gone under total occlusions (due to traffic signs). When total occlusions are detected, the APF automatically enters occlusion tracking mode. The vehicle tracking resumes after the disocclusions, and the APF goes back to the regular tracking mode. The vehicle is tracked in occlusion tracking mode after being totally occluded by a bus in Fig. 5(c), and the regular tracking mode resumes afterward.

To validate our adaptive particle-filtering approach, we compare our results with the results of four other particle filters, i.e., the PF method proposed by Perez *et al.* [5], the SAPF method proposed by Deutscher *et al.* [10], the hybrid PF MS (HPFMS) method proposed by Maggio and Cavallaro [17], and the PF color-and-line-based features (PFCL) method proposed by Xiong and Debrunner [18]. We also compare our results with an MS tracker method proposed by Comaniciu *et al.* [27]. To illustrate our experimental results, the tracking errors for the Video3 sequence are shown in Fig. 6. The tracking errors

TABLE I
TRACKED VEHICLE AND TYPE OF OCCLUSION PRESENT IN THE VIDEO SEQUENCES USED IN THE TRACKING EXPERIMENTS

Video	Tracked Vehicle	Occlusion	Resolution[1]	Reference
Video1	SUV*	Partial and Total	640 × 480	[23]
Video2	Car	Partial and Total	640 × 480	[23]
Video3	SUV*	Partial and Total	640 × 480	[23]
Video4	Car	Total	640 × 480	[23]
Video5	Truck	Total	640 × 480	[23]
Video6	Truck	Total	640 × 480	[23]
Video7	Truck	Total	640 × 480	[23]
Video8	Car	Total	640 × 480	[23]
Video9	Car	Total	640 × 480	[23]
Video10	Car	Total	640 × 480	[23]
Video11	Truck	Total	640 × 480	[23]
Video12	Car	Partial	768 × 576	[26]
Video13	Car	Partial	768 × 576	[26]
Video14	Car	Partial	768 × 576	[26]
Video15	Car	Partial	768 × 576	[26]
Video16	Car	Partial	768 × 576	[26]
Video17	Car	Partial	768 × 576	[26]
Video18	Car	Partial	768 × 576	[26]
Video19	Car	No Occlusion	640 × 480	[23]
Video20	Car	No Occlusion	640 × 480	[23]

* Sport Utility Vehicle.
[1] Resolution in pixels.

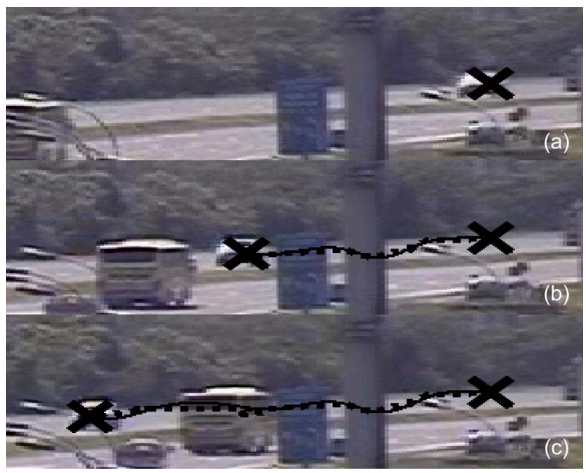


Fig. 5. Example of a tracked vehicle using the proposed adaptive particle-filtering approach. (Top to bottom) (a) Frame 03, (b) frame 35, and (c) frame 55 of the Video3 tracking sequence. The crosses indicate the initial and final tracked vehicle positions, and the dotted line indicates the tracked vehicle path. The solid line shows the vehicle path interpolated to reduce the occlusion edge effect.

are calculated as follows: First, a ground truth is defined by marking the centroid position of the vehicles at each frame of the video sequences. Next, the tracking errors are calculated for each frame using the Euclidean distance between the ground truth position and the tracking results obtained with the different tracking methods. The shaded areas in Fig. 6 indicate frames with vehicles totally occluded. We can observe that, compared with the other tested methods, our APF presents smaller errors before and after the total occlusions. These results show that our approach is able to overcome total occlusions and to resume vehicle regular tracking after the disocclusions. Fig. 7 shows the detected linear motion along the x-axis in the Video3 sequence using different tracking methods. The shaded areas in Fig. 7

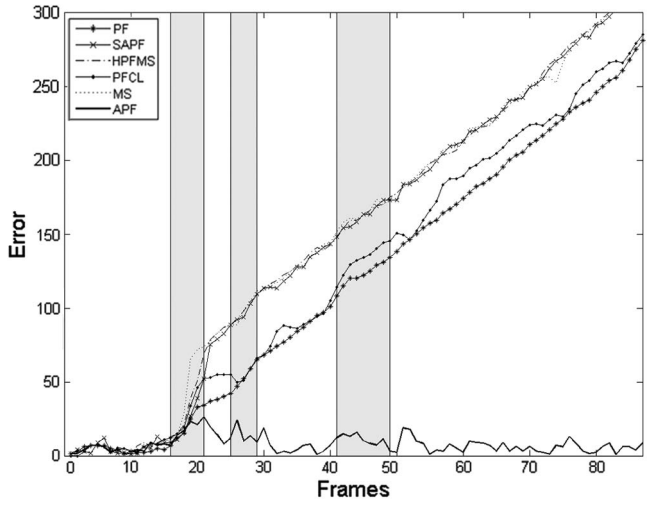


Fig. 6. Comparative results of the tracking errors for different tracking methods for each frame of the Video3 sequence. The error is the Euclidean distance between the ground truth of the tracked vehicle position and the obtained tracking results. The solid line indicates the results of the proposed APF tracking method, and the other types of line indicate the results of the PF, SAPF, HPFMS, PFCL, and MS methods (see Section IV). The shaded areas indicate the frames under total occlusion.

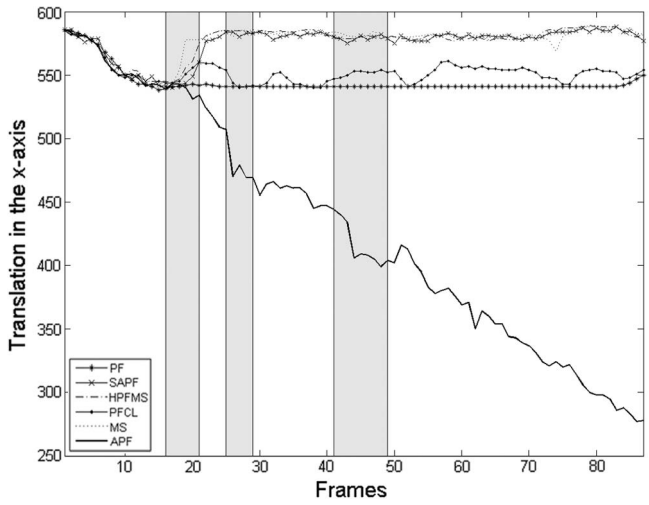


Fig. 7. Translation in the x-axis of the tracked vehicle in the Video3 sequence. The solid line indicates the results of the proposed APF tracking method, and the other types of line indicate the results of the PF, SAPF, HPFMS, PFCL, and MS methods (see Section IV). The shaded areas indicate the frames under total occlusion.

represent frames under total occlusion. These results show that the other methods used for comparison (i.e., PF [5], SAPF [10], HPFMS [17], PFCL [18], and the MS tracker [27]) failed to overcome the total occlusion since they missed the tracked vehicle at the beginning of the total occlusion. However, our APF method was able to overcome the total occlusion and to resume tracking the vehicle after its disocclusion. For the sake of comparison, using a 2.4-MHz quad-core Pentium IBM-PC with 4-GB RAM, the processing times for the tracking Video3 sequence are given as follows (in minutes): our APF method (14.5), PF (14.7), SAPF (14.8), HPFMS (15.2), PFCL (14.8), and MS (12.7).

The tracking results for all video sequences analyzed are in Table II. These results show the RMSE generated by the tracking processes using the different particle filter approaches. It is possible to observe that our approach presents the smallest RMSE values in average, because in our method, the tracking process resumes after the vehicle disocclusions, and the tracking process is not indefinitely interrupted.

TABLE II
RMSE OF THE TRACKING RESULTS FOR DIFFERENT PARTICLE FILTER METHODS

Video	PF[1]	SAPF[2]	HPFMS[3]	PFCL[4]	MS[5]	APF[6]
Video1	85.83	100.19	87.34	93.45	86.83	25.65
Video2	64.41	66.92	63.76	65.30	62.42	9.92
Video3	147.80	180.41	148.43	177.35	144.32	9.74
Video4	133.47	145.70	135.67	146.12	130.51	56.79
Video5	132.50	135.61	133.15	138.48	131.14	64.47
Video6	132.53	134.19	134.47	137.39	130.24	41.59
Video7	134.73	133.37	136.23	139.10	132.26	42.45
Video8	73.47	75.12	72.52	78.43	72.26	12.55
Video9	74.18	78.94	73.37	77.12	73.46	14.21
Video10	74.56	27.41	75.33	78.22	71.64	13.37
Video11	57.11	54.39	53.34	56.25	58.24	6.42
Video12	55.36	55.72	55.73	58.90	58.83	7.38
Video13	54.78	53.42	52.98	55.12	57.64	6.94
Video14	56.12	54.77	53.60	56.47	56.34	6.53
Video15	50.73	52.12	51.44	52.45	50.23	5.32
Video16	52.21	49.67	48.82	52.18	49.77	5.78
Video17	49.90	48.92	49.19	53.23	48.45	4.22
Video18	51.23	51.04	52.30	50.98	51.02	5.19
Video19	5.75	4.32	3.67	4.59	4.34	4.12
Video20	4.82	3.76	2.72	5.12	3.82	3.78
Average	74.57	75.30	74.20	78.81	73.69	17.32

[1] Particle Filter Method [5].
[2] Simulated Annealing Particle Filter Method [10].
[3] Hybrid Particle Filter-Mean Shift Method [17].
[4] Particle filter Color-Line [18].
[5] Mean Shift Method [27].
[6] Adaptive Particle Filter Method (our proposed approach).

V. CONCLUSION

Robust tracking methods are very important in traffic surveillance systems. This paper has presented a new particle-filtering approach that offers good results for tracking vehicles in the presence of partial and total occlusions. Most vehicle tracking methods available in the literature are disrupted by occlusions, but the proposed APF is able to overcome total occlusions and to resume the vehicle tracking process after disocclusions. Our experimental results indicate that our approach is able to track vehicle even in complex situations, such as under total occlusions caused by bridges or other vehicles.

As future work, we intend to improve our APF method to work with intense traffic situations, where the roadway can be crowded and the vehicles can be very close to each other. In addition, we intend to work on the automatic classification of vehicles to qualify and quantify road traffic.

ACKNOWLEDGMENT

The authors would like to thank CONCEPA [23] for making available the video sequences used in this paper as well as CNPq (National Council for Scientific and Technological Development), Brazil.

REFERENCES

[1] V. Kastrinaki, M. Z. ans, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image Vis. Comput.*, vol. 21, no. 4, pp. 359–381, Apr. 2003.

[2] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[3] L. D. Stefano and E. Viarani, "Vehicle detection and tracking using the block matching algorithm," in *Proc. IEEE 3rd Int. Multiconf. Circuits, Syst., Commun. Comput.*, Athens, Greece, Jul. 1999, pp. 4491–4496.

[4] S. Rhee, S. Han, P. Kim, M. Ahmad, and J. Park, "Vehicle tracking using image processing techniques," in *Rough Sets and Current Trends in Computing*. Berlin, Germany: Springer-Verlag, 2004, pp. 671–678.

[5] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. ECCV*, Copenhagen, Denmark, 2002, pp. 661–675.

[6] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[7] T. Schon, "On computational methods for nonlinear estimation," Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden, 2003.

[8] R. Karlsson, "Simulation based methods for target tracking," Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden, 2002.

[9] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *Proc. IEEE Conf. CVPR*, 2005, vol. 2, pp. 1051–1058.

[10] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. IEEE Conf. CVPR*, 2000, pp. 126–133.

[11] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool, "Object tracking with a an adaptive color-based particle filter," in *Proc. DAGM Symp.*, 2002, pp. 353–360.

[12] O. R. A. Jacquot and P. Sturm, "Adaptive tracking of non-rigid objects based on color histograms and automatic parameter selection," in *Proc. IEEE Workshop Motion Video Comput.*, 2005, vol. 2, pp. 103–109.

[13] O. Lanz, "Approximate Bayesian multibody tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1436–1449, Sep. 2006.

[14] D. Koller, J. Weber, and J. Malik, "Towards realtime visual based tracking in cluttered traffic scenes," in *Proc. IEEE Intell. Vehicles Symp.*, 1994, pp. 201–206.

[15] E. B. Meier and F. Ade, "Tracking cars in range images using the condensation algorithm," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 1999, pp. 129–134.

[16] P. L. M. Bouttefroy, A. Bouzerdoum, S. Phung, and A. Beghdadi, "Vehicle tracking using projective particle filter," in *Proc. IEEE Int. Conf. AVSS*, 2009, pp. 7–12.

[17] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. IEEE ICASSP*, 2005, pp. 221–224.

[18] T. Xiong and C. Debrunner, "Stochastic car tracking with line- and color-based features," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 324–328, Dec. 2004.

[19] C. Bishop, *Pattern Recognition and Machine Learning*. Singapore: Springer-Verlag, 2006.

[20] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, Mar. 2008.

[21] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1561–1569, Oct. 2005.

[22] W. Zhang, Q. M. J. Wu, X. Yang, and X. Fang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 161–174, Mar. 2008.

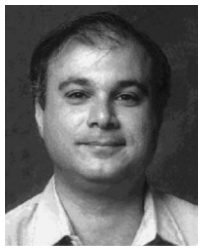
[23] CONCEPA. (Jan. 2010), Empresa concessionaria da rodovia porto alegre - osorio (in Portuguese). [Online]. Available: http://www.concepa.com.br

[24] S. Weinzierl, "Introduction to Monte Carlo methods," NIKHEF Theory Group, Amsterdam, The Netherlands, High Energy Physics - Phenomenology (hep-ph) NIKHEF-00-012, Jun. 2000.

[25] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.

[26] IAKS. (Jan. 2010), Institute fur algorithmen und kognitive systeme (universitat karlsruhe). [Online]. Available: http://i21www.ira.uka.de/image_sequences/

[27] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.



Jacob Scharcanski (SM'03) received the B.Eng. degree in electrical engineering and the M.Sc. degree in computer science from the Federal University of Rio Grande do Sul, Porto Alegre-RS, Brazil, in 1981 and 1984, respectively, and the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1993.

He is currently an Associate Professor with the Institute of Informatics, Federal University of Rio Grande do Sul, and an Adjunct Associate Professor with the Department of Systems Design Engineering, University of Waterloo. He also held research and development positions with Brazilian and North American Industry. He has authored and coauthored more than 100 refereed papers in journals and conference proceedings. His research interests are image processing, pattern recognition, and computer vision.

Alessandro Bof de Oliveira (M'09) received the B.Sc. degree in physics and the M.Sc. degree in computer science in 2004 and 2008, respectively, from the Federal University of Rio Grande do Sul, Porto Alegre-RS, Brazil, where he is currently working toward the Ph.D. degree in computer science.

He is currently a Lecturer with the Federal University of Pampa, Alegrete-RS, Brazil. His research interests are image processing, object tracking in video sequences, and higher education.

Pablo G. Cavalcanti received the B.Eng. degree in computer engineering from the Federal University of Rio Grande, Rio Grande-RS, Brazil, in 2008. He currently is working toward the Ph.D. degree in computer science with the Federal University of Rio Grande do Sul, Porto Alegre-RS, Brazil.

His research interests are image processing and computer vision.

Yessenia Yari received the B.Eng. degree in systems engineering from San Agustin University of Arequipa, Arequipa, Peru, in 2008. She is currently working toward the M.Sc. degree in computer science with the Federal University of Rio Grande do Sul, Porto Alegre-RS, Brazil.

Her research interests are image processing and computer vision.