Digital Signal Processing 23 (2013) 364–374

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Radar-based road-traffic monitoring in urban environments

J. Sánchez-Oro, D. Fernández-López, R. Cabido, A.S. Montemayor, J.J. Pantrigo *

Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, c/ Tulipán s/n, 28933 Móstoles-Madrid, Spain

ARTICLE INFO

Article history:
Available online 27 September 2012

Keywords:
Radar processing
Real time performance
Visual tracking
Particle filters

ABSTRACT

This work presents a novel approach to object detection and tracking in urban environments using images obtained from a radar network, deployed in an urban environment. The proposed system detects, tracks and computes the speed of vehicles and generates alerts when vehicles exceed the predefined road speed limit. The available radar model is a low-cost device oriented to marine environments rather than terrestrial applications. For this reason, we emphasize in the development of a realistic, robust, efficient and effective algorithm which deals with the hardware limitations to provide a suitable overall performance. To reach this objective, we propose dual background subtraction model to detect objects and a tracking method based on the particle filter algorithm. Furthermore, to ensure real time restriction even in HD imagery, our method takes advantage in a natural way of multicore systems and exploits advanced SIMD capabilities available in last multicore processors families. Experimental results demonstrate that the proposed system is able to detect and track multiple objects and to provide speeding alarms when needed. It is also capable to handle target occlusions and disappearances derived from the radar limitations and the noisy urban environment.

1. Introduction

Visual target detection and tracking are technically challenging tasks devoted to the estimation of the position of interesting objects moving in a scene. Due to the important applications of the visual detection and tracking, several approaches have been proposed to tackle the problem using different algorithmic techniques [23,25,13]. Some of the most popular approaches are based on algorithms derived from the particle filter framework [20,17]. Other proposed methods are based on the use of these strategies such as Kalman filters [19], combinations of probabilistic and evolutionary strategies [22,21,15], etc. Modeling the features of the objects being tracked (i.e. color, texture, shape, etc.) is another key aspect to be considered in the context of the problem. These features can be directly obtained in the spatial domain [12,25], or indirectly from some domain transformation of the object features, such as wavelet transform [4,11], among others.

Tracking is usually performed in the context of higher-level applications like video surveillance [27,30,15], gesture and human action recognition [28,29,14], augmented reality [31,1], medicine [16], etc. These potential applications require robust and efficient tracking algorithms. On the one hand, it is neeeded that an increase in the number of targets does not drastically affect the performance of the tracking algorithm. On the other hand, the tracking algorithm needs to work with the inherent difficulties of the feature extraction from the images (i.e. noise, occlusions, etc.).

Object detection and tracking are two areas of interest in the context of radar-assisted remote surveillance, having applications in the automatic control of aerial and marine traffic. There are several papers in the literature which focus on the problem of radar tracking. An approach for radar tracking and imaging is presented in [9] based on the target recognition via Radar Cross Section (RCS) profiles. It creates a database to compare the new RCS captured in order to identify the target. It also suggests an optimization of the traditional inverse synthetic aperture approach to form images with passive radar data results. In [7], a recursive Bayesian solution to the problem of joint tracking and classification for ground-based air surveillance is presented. The proposed system is able to handle multiple targets, false alarms and missed detections. It uses the RCS to create a measurement vector for object classification. Lee [10] tries to find a matrix representation of the relations between the measures and the targets in a multi-object radar system. The problem is formulated as an energy minimization problem. A cognitive radar for tracking applications is presented in [32]. The radar utilizes the previous measurements and actions to optimize the transmitted waveform. Simulation results, based on the tracking of an object falling in space, show a superior performance over a traditional active radar with fixed waveform. There is also a survey of radar-based tracking systems in vehicles [5]. In this review, Kalman Filter and Particle Filter are the most cited techniques. According to this review, the latter obtains the best results.

* Corresponding author.
E-mail addresses: jesus.sanchezoro@urjc.es (J. Sánchez-Oro), david.fernandez@urjc.es (D. Fernández-López), raul.cabido@urjc.es (R. Cabido), antonio.sanz@urjc.es (A.S. Montemayor), juanjose.pantrigo@urjc.es (J.J. Pantrigo).

1051-2004/$ – see front matter
http://dx.doi.org/10.1016/j.dsp.2012.09.012

The detection and tracking tasks in radar images have difficulties derived from the probabilistic nature of the measurement process. For example, the object cannot be characterized by its visual features as the radar does not work in the visible spectrum range. In contrast, radar signal consists of reflections of the electromagnetic waves sent by the antenna. For that reason, there is not a direct relationship between the blob size in the image and the real target size. Another artifact, called jitter, causes that a static object appears as a moving one. Jittering is a consequence of the temporal variability in the measure. These artifacts increase the difficulty of the detection, tracking and association problems, and make unaffordable an approach based on traditional techniques of background subtraction used in the context of computer vision.

It is important to remark that most of the existing radar-based detection and tracking methods in the literature work in aerial or marine environments, which are mostly free of clutter. A priori, urban environments are more complex environments, as they contain obstacles which can produce interferences and noise in the image. Buildings, traffic signals, streetlights, handrails are structural elements typically found in an urban scenario which interfere severely with the radar measurement. The objective of this work is to present a new radar-based target detection and tracking system to control the speed of vehicles in urban areas. The main highlights of this work are summarized as follows:

(1) A model to characterize relevant targets (vehicles) in radar images is proposed.
(2) A detection system based on
   - the definition of entering and exiting areas in the image, taking advantage of the expert knowledge,
   - the use of a background model which handles a dual background. Due to the advantages to handle jitter, dual background model is also used in the measurement model of the tracking algorithm and
   - the development of an adaptive sliding window algorithm which detects new objects saving CPU time.
(3) A specific and improved radar tracking algorithm based on a particle filter scheme, which incorporates
   - expert knowledge for the diffusion stage and
   - a moving average model oriented to be robust to occlusions produced by the radar measurement process.
(4) A significant overall acceleration by adapting the most demanding tracking and detection stages to execute in multicore platforms using some performance optimization approaches. The speedup obtained (up to 58×) allows us to tackle complex tracking scenarios where multiple targets are involved.

As a result, we obtain a successful performance of a marine radar in an urban environment to vehicle detection and tracking. This is a very competitive solution due to its wide spatial range and low cost compared to most terrain radar systems. In addition, radar signal is not affected by light conditions and works in an angular range up to 360 degrees. The reminder of the paper is devoted to detail and demonstrate these highlights and it is organized as follows: Section 2 shows the system overview, from a hardware and a software perspective; Sections 3, 4 and 5 respectively present our approach for scenario modeling, target detection and target tracking; Section 6 is devoted to illustrate the system performance; finally, Section 7 summarizes the conclusions of this work.

2. System overview

This section presents an overview of the system from both a hardware and a software perspective. The hardware view is depicted in Fig. 1 and consists of two main subsystems: the radar and camera subsystems. The former is the subject of this paper and it is highlighted in the aforementioned figure. The radar subsystem consists of 14 radars placed in different relevant locations (determined by the local police department) along the city and 2 servers in the control center. Each radar device works in the 9 GHz band, with a range less than 1 km due to a short pulse transmission feature, and discretizing the analogical signal using 3 bits (8 intensity levels). As shown in Fig. 1, each radar sends images through a WiMAX dedicated network to the server every 1.25 seconds in polar coordinates. When the velocity of the detected vehicle exceeds the speed limit of a road, the server sends an alert to the "Camera network server" via the WiMAX network. The alert file contains, among other data, the geographical location of the incumbent radar, needed to associate the camera network to the location of the offending vehicle. Additionally, servers A and B are not only devoted to analyze the radar images but they also provide the interface for configuring each radar parameters. Configuration settings are sent in an appropriate configuration file to each radar.

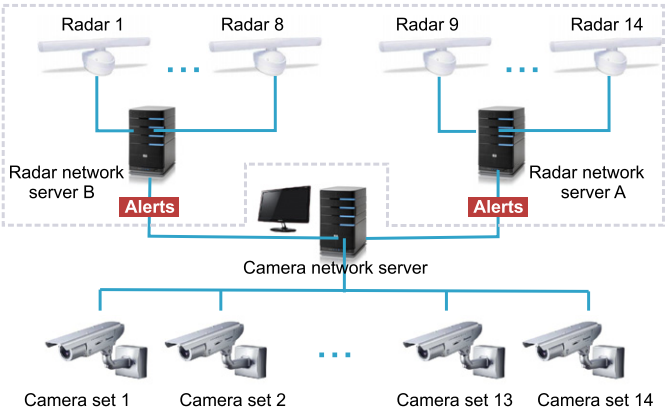

Fig. 1. System overview.

Fig. 2 presents the software view of the system. It describes the relationships among the four main modules: scenario modeling, target detection, target tracking and background subtraction modules. Scenario modeling module provides tools to describe the geometry of the environment, allowing us to categorize the different urban areas. Target detection module is devoted to search new vehicles entering the scene. Target tracking module tracks each vehicle detected in the previous stage and computes its speed. Additionally, a background subtraction module provides the measurement image used for both the detection and the tracking modules.

3. Scenario modeling

Cities are spatially organized in a structured way. As a consequence, there are regions exclusively dedicated to roads, motorways, car parks, etc. The structure of urban environments can be considered as permanent, since it is not usually changing over long periods of time. In our application, we can suppose that radars are placed in a fixed geographical location. Given these two facts as hypothesis, the algorithms for vehicle detection and tracking can take advantage of this knowledge to improve robustness and efficiency. As the radar will be used to detect vehicles, it is useful to distinguish the areas in which vehicles are moving (generally roads) from the areas where it is not possible to find any car (corresponding to buildings, pedestrian precinct, etc.). We explicitly model two different interesting areas:

- **In–out areas** represent zones where new vehicles are expected to appear or existing vehicles disappear.
- **Transit areas** define zones in which vehicles can travel, generally roads.

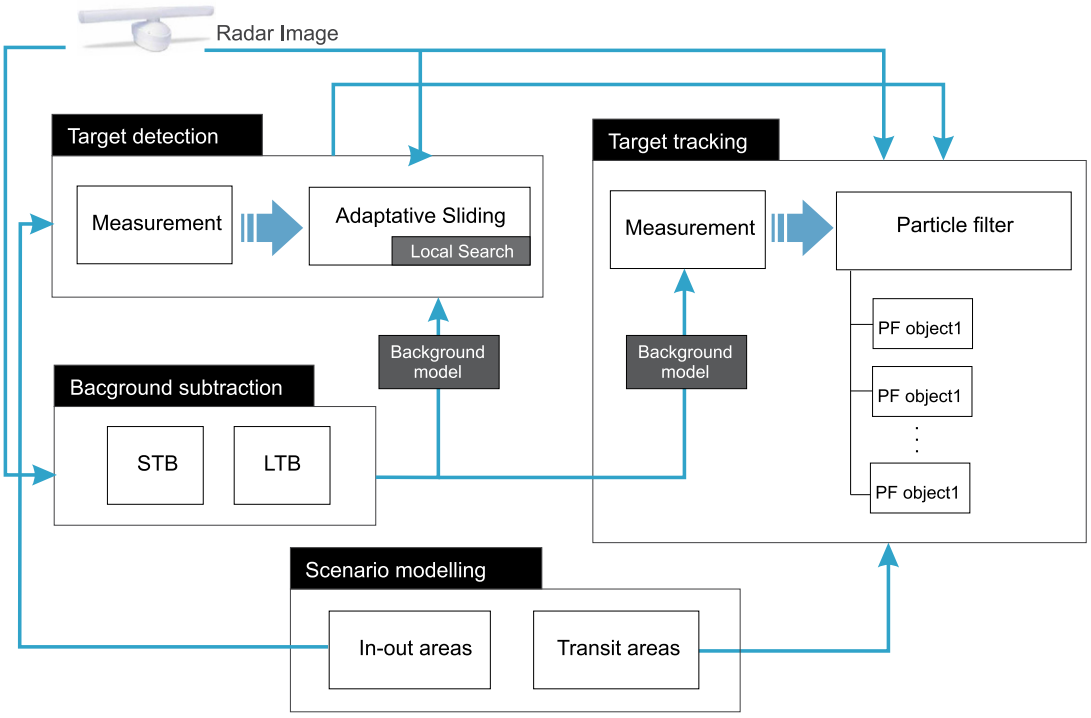

Fig. 2. Software architecture overview.

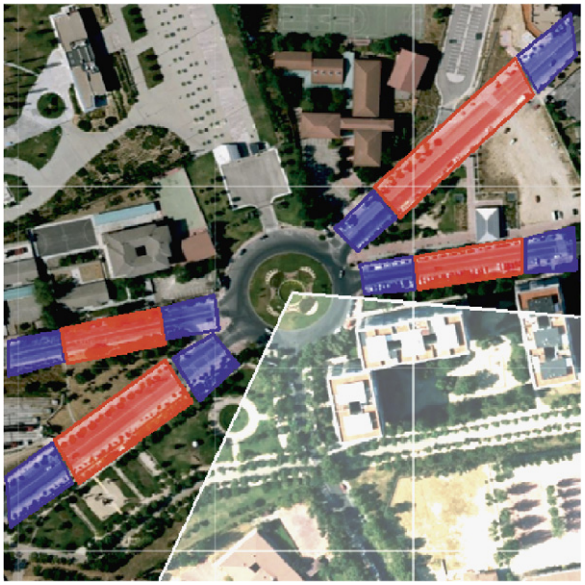

Fig. 3. Expert knowledge modeling of a roundabout. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

As a result, a third area is implicitly defined which corresponds to non-interesting zones. The system also provides the option to blank these regions. In particular, we use the in–out areas to initialize and finalize our tracking algorithm. In those areas we expect to detect new entering or exiting vehicles. The transit areas are important in order to restrict the motion of the targets along them as we will describe later.

Fig. 3 depicts an expert knowledge scenario model of a roundabout in which blue regions correspond to in–out areas, red regions represent transit areas and, finally, white regions are depicted as blanked areas.

4. Background subtraction and target detection

Target detection module searches new vehicles entering the scene within the in–out areas. It consists of two main subsystems: (i) a background subtraction method adapted to the probabilistic nature of the radar signal and (ii) a detection method which introduces the idea of adaptive sliding window to save CPU time.

Dealing with measurement noise is the main difficulty for target detection in radar images. It is desirable to filter the signal to preserve the real targets, removing the unsuitable noise. Unfortunately, the considered radar only offers 3 bits of signal (8 intensity levels), which makes difficult to design a suitable filter. For this reason, we use a background subtraction model which takes into account the vehicle kinematics for target segmentation. Traditional background subtraction methods are only based on a single background image. These models are not suitable for radar images, because they contain artifacts (such as high levels of noise, clutter and jitter) which can be included in the foreground image, resulting in a wrong target segmentation. To avoid this problem, we consider a dual background subtraction [18] which is supported by two different background images, called as Long Term Background (B_L) and Short Term Background (B_S). B_S and B_L differ in their updating frequency, and the update of both backgrounds is carried out accumulating the current frame in the corresponding background. Specifically,

$$B_X(x,y,t) = \begin{cases} B_X(x,y,t-\delta t(B_X)) + 1 \\ \quad \text{if } I(x,y,t) > B_X(x,y,t-\delta t(B_X)) \\ B_X(x,y,t-\delta t(B_X)) - 1 \\ \quad \text{if } I(x,y,t) < B_X(x,y,t-\delta t(B_X)) \end{cases} \tag{1}$$

where I_t is the radar image at time t and B_X represents both B_S and B_L since they are identically updated, except for the updating period $\delta t(B_X)$.

Then, we obtain two foreground images for each radar image, computed as the difference between each background and the cur-

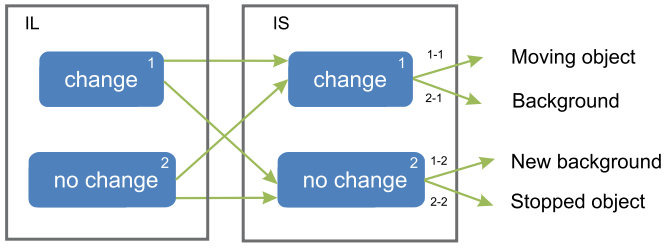

Fig. 4. Event detection using dual background subtraction.

rent frame. They are stored as Long Term Foreground (I_L) and Short Term Foreground (I_S), respectively. In mathematical terms,

$$I_L = |I - B_L| \tag{2}$$

$$I_S = |I - B_S| \tag{3}$$

At this point, we check the active regions of the foregrounds in order to detect different events in the images. Fig. 4 shows all possible events that can occur using the dual background subtraction. A blob in I_L is interpreted as an object which has not been there for a long time. This object can be a moving or static one. I_S helps us to distinguish between them. If an object detected in I_L also appears in I_S, then the object has recently moved. Otherwise, the object was moving before but it stopped recently, so it does not appear in I_S. On the other hand, it is assumed that background objects do not appear in I_L. Using I_S we can distinguish between objects recently incorporated to the background (if they appear in I_S) and objects belonging to the background for a longer time (if they do not appear in I_S). This scheme can also be used to model traffic jams or vehicle crashes.

The foreground image (I_F) is generated comparing I_L and I_S. The considered interesting pixels in the image are those that change in I_L and I_S. In that way, the foreground image will be a binary image, calculated as follows:

$$I_F = \begin{cases} 1 & \text{if } (I_L > th) \text{ and } (I_S > th) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where th is an experimentally set threshold which is tuned by an expert according to the scenario conditions. As presented in Section 2 we have only 8 levels of intensity available, so it is easy to set this threshold, as one level up or down will degrade (by excess or deficiency) the thresholding too much. Whole background subtraction process is depicted in Fig. 5.

Binary blobs in the image are detected using a sliding window algorithm in I_F. The size ($w \times h$) of such a window is empirically pre-established according to the expected dimensions of the targets taking into account the preselected range of the radar. Then, the search explores the in–out areas by sliding the window. When the density of pixels in the region enclosed by the window exceeds a predefined threshold, the region is considered as an object, and it is added to a list of candidate targets. Defining (x_p, y_p) as the center of the window, the weight (π) of a region is calculated as the zero-order moment (area) in that binary window:

$$\pi = \sum_{x=x_p-\frac{1}{2}w}^{x_p+\frac{1}{2}w} \sum_{y=y_p-\frac{1}{2}h}^{y_p+\frac{1}{2}h} I_F(x, y) \tag{5}$$

4.1. Adaptive Sliding Window (ASW)

In this work, we explore the performance of two different strategies to slide the window into the in–out areas: (i) the standard method and (ii) a novel Adaptive Sliding Window (ASW) method, respectively. The former computes the weight of all the possible windows belonging to the in–out areas in a sequential way, as an exhaustive search. This method is very time consuming and we needed to improve its execution. The ASW method takes advantage of exploring a binary image. If the weight of a window in a given location is very low, then it is possible to discard the exploration of all the consecutive windows which overlap with the former window (see left side of Fig. 6 and note the horizontal shift of the window). As we will demonstrate, this strategy results in a considerable reduction of the computing time. In addition, when a weight of a sliding window in a given location exceeds a threshold, then we perform a local search to fit the sliding window to the mode of the distribution of the nearest blob, obtaining its position (see right side of Fig. 6). After this local search, the sliding process restarts at the position of the next non-overlapping window.

5. Target tracking

We propose a tracking method based on the Sampling Importance Resampling (SIR) Particle Filter (PF) framework [6] to track each target detected in the previous stage. However, the filter adapts the size of the estimated region of interest using a local search procedure. More on this local search particle filter variant (LSPF) can be found in Cabido et al. [3]. It has been adapted to be suitable for tracking in radar images. This section is devoted to describe the tracking algorithm in detail.

5.1. Solution structure

Solutions store the set of required variables which describe the system state and its weight. The solution structure proposed for a tracked object is a state vector $s_i^t = [(x_i^t, y_i^t, vx_i^t, vy_i^t)]$, where (x_i^t, y_i^t) represents the position of the object i in the image at time t, and (vx_i^t, vy_i^t) corresponds to its velocity vector. The state s_i^t has an associated weight π^t, which is related to its likelihood (a more detailed exposition can be found in Gordon et al. [6] and Zotkin et al. [24]). Besides, many other time-dependent object features (such as their orientation or size) could be added to the solution structure.

5.2. Measurement model and PF evaluation

We consider the binary foreground image I_F obtained during the target detection stage to evaluate the particle weights. Specifically, in our proposed Sampling Importance Resampling particle filter the weight π_i^t at time t assigned to a state s_i^t is computed using the measurement image I_F and computing the number of active pixels belonging to the bounding box centered in (x_i^t, y_i^t) as expressed in Eq. (5). The higher the number of active pixels contained in the bounding box associated with the state, the higher the particle weight is.

5.3. PF selection stage

The selection stage is intended to improve the quality of the set, letting the best particles survive and replacing the worst ones by better estimators. We have implemented the SIR PF selection simulating a roulette wheel selection procedure. Eventually, this procedure will replace most low-quality particles as, in average, their weight values would not be very high. However, there should be some probability for low-quality particles to survive because they can provide diversity to the set and scape from local minima.

5.4. PF diffusion

The PF diffusion is used to keep the needed diversity in the particle set. Usually, it consists of a random perturbation of the spatial

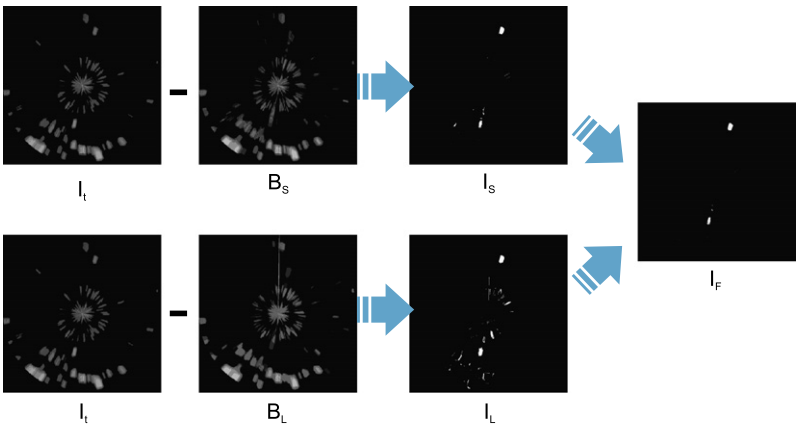

Fig. 5. Dual background subtraction.

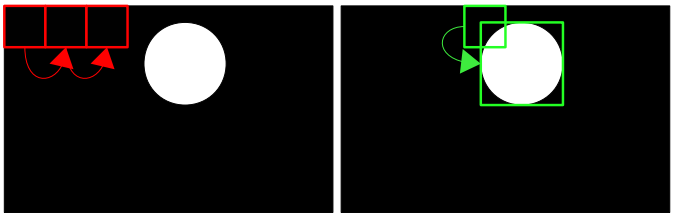

Fig. 6. Scheme of the Adaptive Sliding Window computation approach.

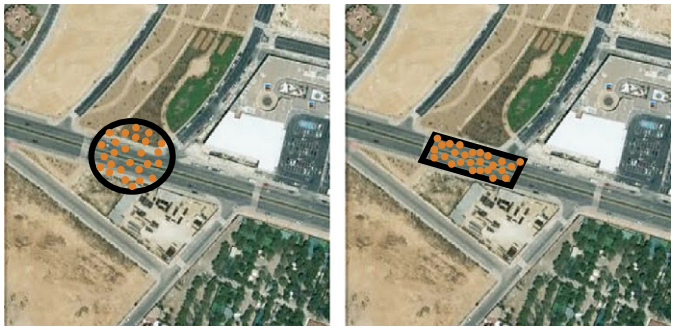

Fig. 7. PF diffusion. (a) Same relevance for longitudinal and traversal movement. (b) Particle diffusion using the scenario knowledge.

coordinates of a given particle. In this problem, we can take advantage of the scenario knowledge. Using the information of where the roads are placed, we can determine in which road is each object, and make the diffusion to fit into the road where the object is moving. In this way, the longitudinal movement along the road is more important than the transversal one (see Fig. 7).

Moreover, we also propose the use of heuristic maps in order to correct possible misaligned maps as well as to ensure a diffusion (and a later system model application) along the real road. Maps misalignments may introduce critical errors in the scenario modeling described in Section 3. Fig. 8 shows the maps of a predefined location where a radar device is placed, the heat map of experimentally detected traffic and the overlapping of both maps. The diffusion stage of the PF is constrained to place new coordinates in areas of existing roads.

5.5. PF system model

The system model describes the temporal update rule for the system state [24]. The tracked object state consists of a given number of spatial coordinates and their corresponding velocities, deriving in a first-order system model.

$$\begin{cases} x^{t+1} = v_x^t \Delta t + x^t \\ y^{t+1} = v_y^t \Delta t + y^t \end{cases} \tag{6}$$

This stage has been also adapted to take advantage of the scenario knowledge in an equivalent way as in the PF diffusion stage so, given the heuristic map of a radar device, we restrict the coordinates of next time steps to fall on existing roads, discarding new states that do not satisfy this assumption and replacing them to maintain the population.

5.6. Specific PF improvements for radar images

Visual tracking using radar images has to deal with noise and artifacts derived from the probabilistic nature of the measurement process. Those artifacts are summarized below:

- **Jitter:** We call jitter to an unexpected change on the measure received from the radar at a given time t with respect to the previous one $t-1$. As a consequence, objects in the image appear slightly displaced in each frame, although the object has not really changed its position. Therefore, it can produce errors in the position and velocity estimation.
- **Occlusions and disappearances:** As in a computer vision problem a big object placed between the antenna and the tracked target can produce occlusions. Furthermore, another common artifact consists of the disappearance of a detected object in some frames. Obviously, this effect introduces difficulties in the tracking process.
- **Correspondence problem:** Two objects moving at the same speed in the same area appear as only one object in the radar image. The system should be prepared to distinguish between them when they split.
- **Intensity difference:** Each object produces a different signal intensity based on its shape, materials and distance to the radar antenna. If a moving object is close to another reflecting object, its intensity will also change, and its detection can be more difficult.

The system should deal with the aforementioned artifacts to become a realistic approach for vehicle tracking. The jitter problem is tackled using the dual background subtraction, since that strategy allows us to check if an object is moving or its apparent movement is caused by the jitter phenomena. However, in order to face problems in which the signal fluctuates or even disappears (occlusions), we redefine the computation of the particle weight as an Exponentially Weighted Moving Average (EWMA):

$$\pi_i^t = \alpha M_{00} + (1-\alpha)\pi_i^{t-1} \tag{7}$$

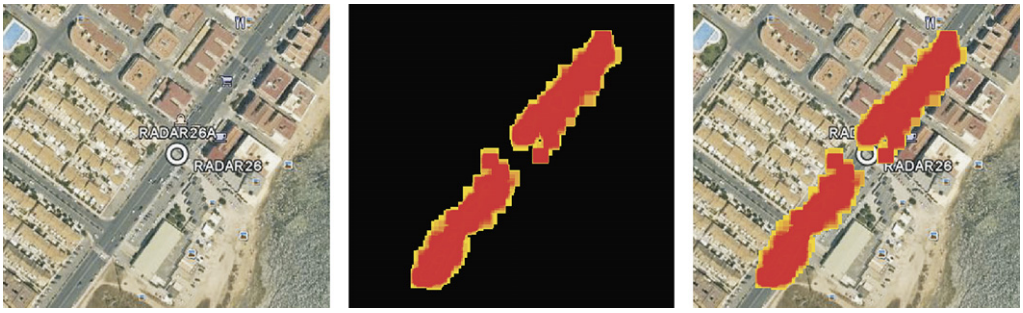

Fig. 8. Heuristic maps construction. (a) Map of a predefined location for a radar device. (b) Detected traffic on location (a) over a period of usage. (c) Heuristic map resulting from overlapping both maps.

where

$$M_{00} = \sum_{x=x_i^t-\frac{1}{2}w}^{x_i^t+\frac{1}{2}w} \sum_{y=y_t^i-\frac{1}{2}h}^{y_t^i+\frac{1}{2}h} I_F(x, y) \tag{8}$$

is, as in the detection evaluation stage, the zero-order moment or area in a binary region given by the position x_i^t and y_i^t of the bounding box associated to the particle state coordinates (x_i^t, y_i^t).

Therefore, the weight of each particle (π_i^t) is computed considering two terms. The first term computes the particle weight based on the measure at time t. The second term takes into account the influence of the previous weight (π_i^{t-1}). The experimental parameter α is the learning factor and allows us to tune the influence of both terms in the calculation of the new weight. The assumption made here is that an object which was in the frame at time $t-1$ cannot be completely disappeared at time t. This strategy allows us to maintain the state of hidden vehicles for some frames.

We face the correspondence problem by the detection system. When the area of a previously detected object in the image splits in two different targets, the previous object was actually a pair of objects moving together. In this case, the previous objects will be deleted from the list and two new objects will be added.

The intensity difference problem is approached by giving a range of intensity levels to each target in which it can move during the frames. In that way, if a target slightly changes its intensity, the new value will be in the range defined for that target. With that solution, the target will maintain its identity in spite of the change of the intensity.

5.7. Alert generation

The final aim of this work is to report alerts of anomalous situations on the area in which the radar is working. Because of that it is really important to reduce the number of false positives given by the tracking system. In order to reduce false alerts, we have set a condition that helps the system to elucidate if an alert has been produced either by a real object or by non-filtered noise.

The condition needed to generate an alert is the continuity of the vehicle in the image with a feasible trajectory, according to the roads located in the area. In that way, an alert is generated only if an object maintains its trajectory for a predefined number of frames, F.

The way to check if a vehicle is following a feasible trajectory is based on its position and speed. On one hand, we use its position to check if the vehicle is actually moving along a road in the image. On the other hand we use its velocity vector to check that the speed in consecutive frames has not changed in an abrupt way.

With those measures it is possible to filter the false positives and avoid in a great manner false alerts.

Table 1
Parameter settings for the proposed system.

Symbol	Description	Value
$\delta t(B_S)$	Short term background update period	2 frames
$\delta t(B_L)$	Long term background update period	5 frames
w	Bounding box width	100 pix.
h	Bounding box height	100 pix.
α	Learning factor	0.75
F	#frames/alarm	3 frames
N	#particles/PF	100

Table 2
Experimentation for short $(\delta t(B_S))$ and long $(\delta t(B_L))$ term background update period selection. Sequence 1 presents 37 real targets while Sequence 2 presents 19.

Update period (frames)		Sequence 1		Sequence 2	
$\delta t(B_S)$	$\delta t(B_L)$	TP	FP	TP	FP
2	3	30	12	18	76
2	4	31	23	18	88
2	**5**	**26**	**8**	**18**	**82**
2	6	28	23	17	81
2	7	31	12	17	81
2	8	31	16	17	87
3	4	30	21	16	104
3	5	30	17	17	106
3	6	34	24	16	119
3	7	33	22	17	115
3	8	34	25	16	115
4	5	30	26	19	119
4	6	34	32	18	122
4	7	34	34	18	117
4	8	36	72	19	130

6. Experimental results

This section is devoted to describe the results obtained by the detection and tracking system. The experiments were performed on an Intel Core 2 Duo E8400 3 GHz with 3 GB of RAM and Windows 7 32 bits OS, and radar images were supplied by a low-cost radar device MDC-2000 oriented to marine applications offering 8 discrete levels of intensity. Several video sequences and results of our proposal are available at: http://www.gavab.etsii.urjc.es/capo/dsp_radarTracking.

The proposed system was developed in C, using the OpenCV 2.1 (Open Source Computer Vision) library. System parameters have been experimentally set and selected values are summarized in Table 1. In particular, the number of video frames for the short/long term background update period selection are shown in Table 2. We choose the combination that performs best, giving the less number of false positives, that is, 2 and 5 video frames for $\delta t(B_S)$ and $\delta t(B_L)$, respectively.

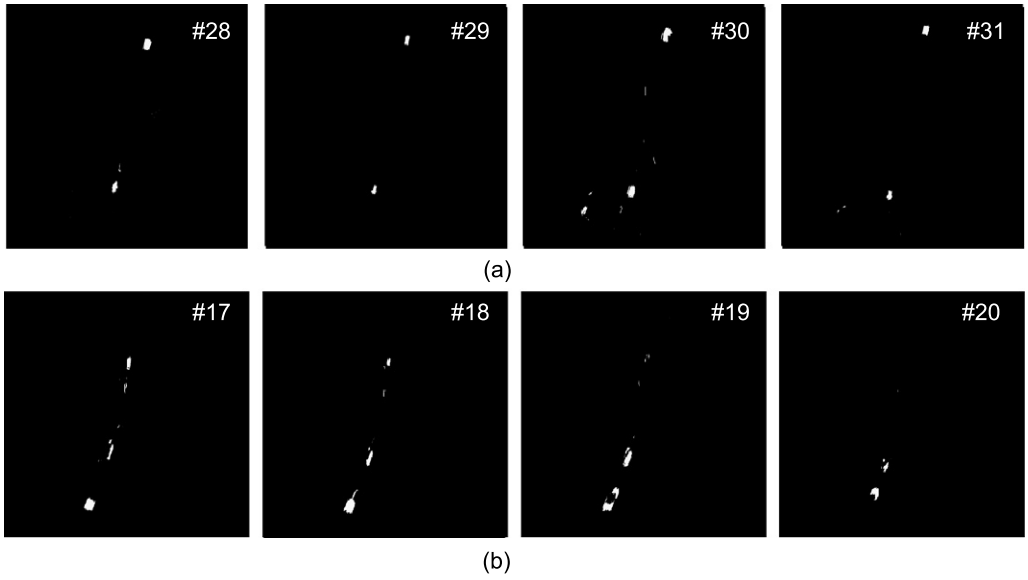

Fig. 9. Background subtraction stage. Active pixels belong to the foreground. (a) Foreground of consecutive video frames belonging to Sequence 1. (b) Foreground of consecutive video frames belonging to Sequence 2.

The bounding box size (w and h parameters in Table 1) is originally taken 100 pixels by 100 pixels. However, a local search is performed once the region of interesting pixels is located. Then, the size of the window is refined to the one that contains only foreground pixels.

The EWMA learning factor (α in Table 1) weights the region of interest against its history for the computation of the particle weight. In our experiments a learning factor of 0.75 performed slightly better than 0.25 and 0.5 when handling occlusions, so we chose it. The number of frames (F in Table 1) taken in order to send an alert is selected depending on the maximum speed of the road. Taking into account that the frame rate of the radar is 1 frame every 1.25 seconds (0.8 fps), waiting 3 frames is enough for a vehicle to travel about 50–100 meters. We can get a close idea of the kind of alert (if any) in such a long distance. The number of the considered particles for the PF (N in Table 1) is set to 100, as we got reasonable quality and performance results. They will be described in Sections 6.3 and 6.4. Note that each new target detection in an in–out area creates a new particle filter in the system.

Once the parameters are selected, the proper experimental results for the detection and tracking tasks are divided in three different sections. In the first place, the target detection method is tested. We use different radar sequences and compare the obtained results with the ground truth given by a human expert. Secondly, performance results are showed. The adaptive sliding window proposal lets us accelerate the detection stage, showing significant speedups. Moreover, multiple particle evaluation performed in the tracking stage can exploit multicore platforms facilities. Performance results show how the proposed method can benefit from using parallel computation. Finally, we measure the tracking method accuracy in the last section and present the visual results obtained in different scenarios.

6.1. Background subtraction and object detection

Fig. 9 shows the results obtained by the dual background subtraction for two different radar sequences. Sequence 1 represents an ideal scenario, where there are not bridges or buildings which can cause high levels of noise in the images. Thus, most foreground pixels correspond to the two targets labeled by an expert (see Fig. 9(a)). However, some noise remains active after the background subtraction. These noisy pixels will be discarded in the tracking stage using a density thresholding.

Table 3
Analysis of the objects detected and tracked in the two sequences. Columns describe the number of frames per sequence, real objects in the scene (RO), true detections (TD), and false detections (FD).

	Length	RO	TD	FD
Sequence 1	37	2	2	0
Sequence 2	24	5	5	0
Sequence 3	43	5	5	2
Sequence 4	57	2	2	1
Sequence 5	33	9	7	1
Sequence 6	72	5	5	1

Dual background subtraction was tested in more complex scenarios. Fig. 9(b) shows the foreground extracted for the Sequence 2. This sequence contains different urban structures which introduce higher levels of noise in the resulting image. Noise areas, as the one which appears at the bottom of the image, could not be discarded in the tracking stage, resulting in a false positive detection. As we discuss in Section 5.7, false positives are filtered by assuming a continuity premise.

The object detection stage was tested using different real sequences obtained from the radar network. In order to test the stages, a ground truth measurement was created by an expert. For each scene, the number of objects detected and their positions are stored. Then, expert information is used to analyze the performance of the automatic detection stage.

Table 3 shows the performance of the detection stage. Specifically, the system detects (true detections, TD) the 92.9% of all objects labeled by an expert as real objects (RO). On the other hand, urban structures introduce noise in the resulting image. This situation produces false detections (FD). The system detects one false object per sequence on average. Please note that we denote here as “object” to one that appears as a blob (or is eventually occluded) along the entire sequence maintaining the correspondence, not the number of apparitions frame by frame.

6.2. Adaptive sliding window performance

In this section, we measure the performance benefit obtained by using the Adaptive Sliding Window (ASW) approach described

Table 4
Adaptive sliding window performance results (time and speedup) for different configurations.

Resolution	Window size	Method		Speedup	
		Std (ms)	ASW (ms)	Average	Maximum
640 × 640	10 × 10	3.75	1.08	3.60×	5.32×
	20 × 20	10.04	0.98	10.22×	14.70×
	40 × 40	15.82	0.85	17.88×	24.72×
800 × 800	10 × 10	6.09	1.64	3.77×	4.89×
	20 × 20	17.04	1.45	11.63×	15.55×
	40 × 40	31.76	1.31	23.83×	31.15×
1024 × 1024	10 × 10	9.97	2.46	4.20×	5.95×
	20 × 20	30.20	2.12	14.18×	18.36×
	40 × 40	63.91	1.97	32.05×	43.36×

in Section 4.1. Experiments were developed taking into account different radar image resolutions and window sizes. Performance results are computed by executing the detection stage for each sequence (Sequence 1–6). Table 4 shows the average times per frame obtained with (ASW) and without (Standard) the sliding window approach. In most demanding configurations (1024 × 1024 radar images) we get upper bound speedups between 5.9×–43.3× (represented by the maximum speedup column in the table) while in a real execution we get an average of about 70%–74% of the upper bound. Note that for increasing window sizes, the standard method consumes more time in their evaluation (in the same way as a convolution with larger kernels) while the ASW method can perform in less time due to faster runs for large windows (especially in absence of noise) for computing the entire frame.

6.3. High performance PF evaluation

In visual tracking methods like particle filters, particles weight computation stage is one of the most computationally demanding tasks [26]. However, modern architectures provide multicore processors and vector registers that can be exploited in time consuming applications, typically including image and signal processing ones. In this section, we propose a shared-memory parallel approach for multiple particle evaluation.

OpenMP is a shared-memory parallel API and its directives can be used to simplify thread creation and management while parallelizing at core level. Additionally, SIMD instructions (Single Instruction Multiple Data) supported by consumer architectures can be included in order to exploit fine grain data parallelism. Intel x86 SIMD registers are available since the introduction of the 64-bit XMM registers and their MultiMedia eXtensions (MMX) in 1995 for integer data types. After that, the Streaming SIMD Extensions (SSE) family for 128-bit floating point data was included in 1999 to enhance the SIMD capabilities for 3D content. The width of the registers remained constant until the recent Intel Sandy Bridge architecture was released in 2011 with the Advanced Vector eXtensions (AVX). The AVX registers are 256-bit wide for floating point arithmetic (supporting integer data types in the recent 22nm Intel Ivy Bridge family) capable of processing up to 8 32-bit floating point values simultaneously or any other 256-bit combination.

Our experimental platform for this performance evaluation is based on a Sandy Bridge Intel Core i7-2600 at 3.4 GHz with 2 GB RAM. Parallel particle evaluation was developed using OpenMP API and different sets of SIMD instructions. Fig. 10 shows the performance results obtained when different number of targets are tracked (1–32). Note that the number of needed particles increases with the number of targets. In particular, for this experimentation our system evaluates 128 particles per object instead of 100 in order to better align the memory and fill the registers size. Trying to reduce the computational load of the evaluation stage, up to four parallel approaches have been developed and compared against an

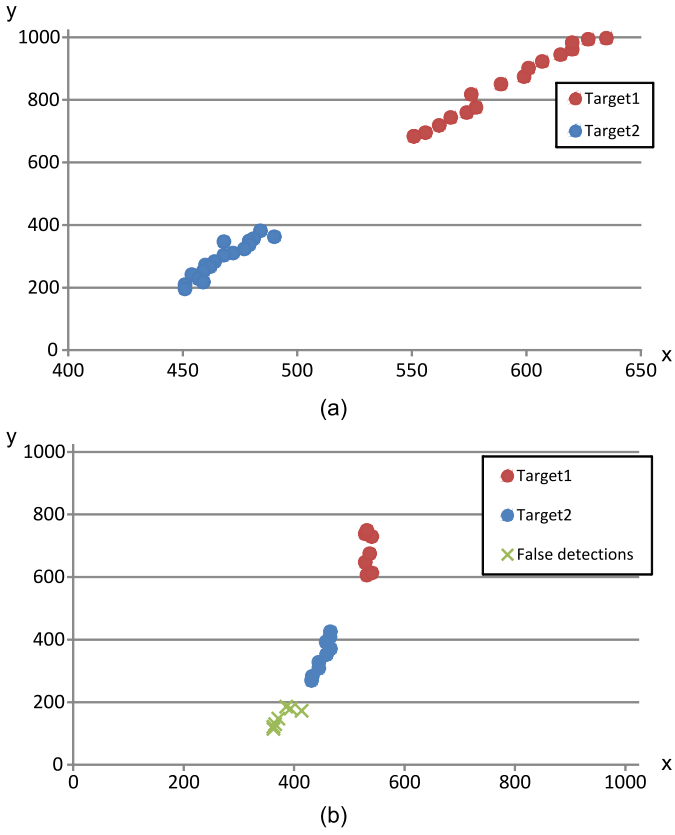

Fig. 11. Tracking results. (a) Trajectories followed by two vehicles in Sequence 1. (b) Trajectories and false positives detected in Sequence 2.

equivalent single-threaded version (called as *Serial*). *SSE* and *AVX* approaches exploit 128-bit and 256-bit wide registers respectively, while *OMP+SSE* and *OMP+AVX* versions can also take advantage from multicore computation to evaluate multiple ROIs in parallel.

Fig. 10 also depicts the best speedup obtained for each object configuration. We can see how most demanding configurations (16–32 targets) show important speedups between 28–50× (see Fig. 10).

6.4. Object tracking

Fig. 11 shows tracking results obtained for Sequence 1 and Sequence 2. The former constitutes an ideal scenario where radar images present low levels of interferences. Thus, using the binary foreground image described in Section 5.2, the particle filter algorithm is able to track the targets. Fig. 11(a) shows trajectories followed by the two targets. Identities are preserved during the tracking.

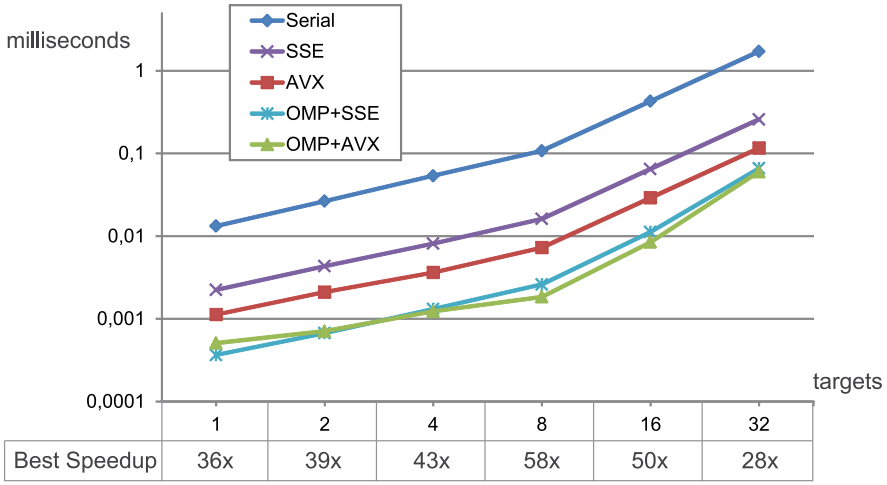

Fig. 10. Performance comparison between serial and parallel particle evaluation. Times were computed for different number of targets.

Table 5
Comparison of the tracking quality of the Particle Filter (PF) with 100 particles and the Kalman Filter (KF) for the six video sequences. Columns represent ground truth (g), miss-detections (m), false positives (fp), identity switches (mme) and the MOTA score (higher is better).

	Method	*g*	*m*	*fp*	*mme*	*MOTA*
Sequence 1	PF	37	2	0	0	0.946
	KF	37	4	0	0	0.892
Sequence 2	PF	19	2	7	0	0.526
	KF	19	2	8	0	0.474
Sequence 3	PF	25	0	15	0	0.400
	KF	25	1	13	0	0.440
Sequence 4	PF	5	0	1	0	0.800
	KF	5	0	1	0	0.800
Sequence 5	PF	20	0	9	0	0.550
	KF	20	2	12	0	0.300
Sequence 6	PF	11	0	0	0	1.000
	KF	11	0	4	0	0.636

On the other hand, Sequence 2 shows a challenging scenario in Fig. 11(b), where the radar signal presents higher levels of noise. In the tracking stage, noisy measures are translated into false positives, which are mostly filtered using a continuity criterion.

To evaluate the tracker, the Multiple Object Tracking Accuracy (MOTA) measure is used [2] as:

$$MOTA = 1 - \frac{\sum_t (c_m(m_t) + c_f(fp_t) + c_s(mme_t))}{\sum_t g_t} \tag{9}$$

where g_t is the number of ground truth detections, m_t the number of miss-detections, fp_t the false positive count, mme_t the number of instantaneous identity switches and t corresponds to each time step. This MOTA score takes into account the quality performance of the tracker in terms of the involved parameters, so the best MOTA value would give a value of 1 (no miss-detections, false positives or identity switches). According to [8], the weighting functions are set to $cm = cf = 1$, and $cs = \log_{10}$.

To better understand the performance of the different algorithms, we then compute the three individual components of MOTA: m, fp, mme. The results of the particle filter (PF) tracker using 100 particles are compared to a more traditional Kalman Filter (KF) tracker and are shown in Table 5. The PF system obtains a MOTA score of 0.7 on average while the KF is 0.59 on average (higher is better). For the proposed system this is a very acceptable result, as it is a very important issue to avoid false positives.

Finally, visual tracking results for both, KF and PF, trackers obtained for Sequence 1 and Sequence 2 are presented in Fig. 12. Tracking results are also acceptable, taking into account the high level of noise present in the images. A target represented as a cross symbol ("×") corresponds to a false positive. In Sequence 1 (Fig. 12(a)) the Kalman filter (top row) temporally looses the first target (it is temporally associated to an occluded state) as the estimated velocity at initialization is slightly lower than the real one. Later, in this case it corrects its estimated position (and then becomes tracked) but in others it looses targets and cannot recover them. In Sequence 2 (Fig. 12(b)) the Kalman filter (top row) looses the first target (and does not recover it), and also shows some inaccuracies in the estimation of target 2 (frames #18 and #19). Overall, the KF shows more false positives (see Table 5), although in this segment of the sequence it presents less false positives than the particle filter tracker.

7. Conclusions

In this paper we have presented a radar-based visual tracking system oriented to road-traffic monitoring. The use of radar technology makes the system be robust against lighting or weather conditions. The system allows to model urban areas and detect, track and compute the speed of vehicles in urban environments. The detection module consists of a dual background subtraction model for the segmentation of vehicles and an adaptive sliding window procedure to determine the initial position of entering vehicles. On one hand, experimental results demonstrate that the dual background subtraction model is suitable for noisy radar images, thus making the detection and tracking affordable. On the other hand, the sliding adaptive window procedure has proven to be substantially more efficient than the standard method, obtaining equivalent results in terms of quality. The proposed tracking module is based on the particle filter algorithm. This module, in conjunction with the detection module enables to handle entering and exiting vehicles from the scene. Experimental results revealed that the tracking precision is high enough to compute vehicle speeds and detect potential speeding offenses. Our performance optimization approaches allow us to obtain significant speedups over the non-optimized method. These speedups enable the system to execute the application for seven radars in only one computer, so we could distribute fourteen radars using two servers. Nowadays, the proposed system is being used in a real urban environment, showing a suitable performance to end-user expert opinion.

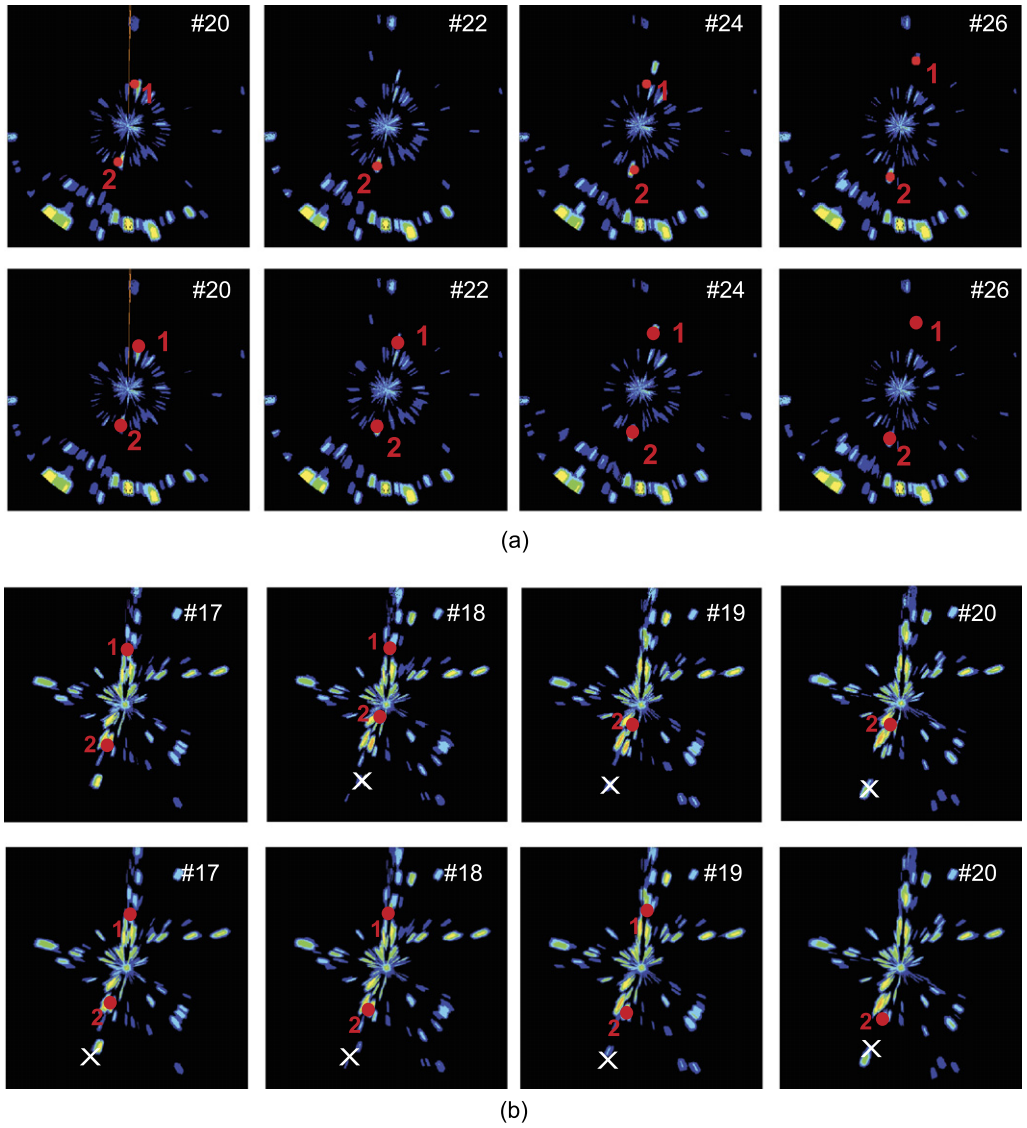

Fig. 12. Visual tracking results for Kalman Filter (KF) and Particle Filter (PF). (a) Sequence 1: KF tracking is shown in the first row, PF in the second row. (b) Sequence 2: KF tracking is shown in the first row, PF in the second row.

Acknowledgments

This research has been partially supported by the Spanish companies Inteo Media Mobile SLNE and Vigidar SL and the Spanish Government research funding ref. TIN 2011-28151.

Also we would like to acknowledge the reviewers for their constructive comments. They significantly helped us to improve the quality of the paper.

References

[1] I. Barandiarán, C. Paloc, M. Graña, Real-time optical markerless tracking for augmented reality, Journal of Real-Time Image Processing 5 (2) (2010) 129–138.

[2] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics, EURASIP Journal on Image and Video Processing (January 2008), Article ID 246309, 10 pages, http://dx.doi.org/10.1155/2008/246309.

[3] R. Cabido, A.S. Montemayor, J.J. Pantrigo, B.R. Payne, Multiscale and local search methods for real time region tracking with particle filters: local search driven by adaptive scale estimation on GPUs, Machine Vision & Applications 21 (1) (2009) 43–58.

[4] F.H. Cheng, Y.L. Chen, Real time multiple object tracking and identification using discrete wavelet transform, Pattern Recognition 39 (2006) 1126–1139.

[5] N. Floudas, A. Polychronopoulos, A. Amditis, A survey of filtering techniques for vehicle tracking by radar equipped automotive platforms, in: International Conference on Information Fusion, 2005.

[6] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE Proceedings F. Radar and Signal Processing 140 (2) (1993) 107–113.

[7] S.M. Herman, A particle filtering approach to joint passive radar tracking and target classification, Ph.D. thesis, University of Illinois, 2002.

[8] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, J. Zhang, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 319–336.

[9] A.D. Lanterman, Passive radar imaging and target recognition using illuminators of opportunity, Paper presented at the RTO SET Symposium on "Target Identification and Recognition Using RF Systems", held in Oslo, Norway, 11–13 October 2004, and published in RTO-MP-SET-080.

[10] Y.-W. Lee, A study on the identification and tracking of moving vehicles in the dense environment, in: International Conference on Industrial Technology, 2002.

[11] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, H. Wang, Automatic player detection, labeling and tracking in broadcast soccer video, Pattern Recognition Letters 30 (2) (January 2009) 103–113.

[12] R. Muñoz-Salinas, E. Aguirre, M. García-Silvestre, A. González, A multiple object tracking approach that combines colour and depth information using a confidence measure, Pattern Recognition Letters 29 (2008) 1504–1514.

[13] W. Ng, J. Li, S. Godsill, J. Vermaak, Tracking variable number of targets using sequential, in: Monte Carlo Methods, IEEE/SP 13th Workshop on Statistical Signal Processing, 2005, pp. 1286–1291.
[14] J.J. Pantrigo, A. Sánchez, J. Mira, Representation spaces in a visual based human action recognition system, Neurocomputing 72 (4–6) (2009) 901–915.
[15] J.J. Pantrigo, J. Hernández, A. Sánchez, Multiple and variable target visual tracking for video-surveillance applications, Pattern Recognition Letters 31 (2010) 1577–1590.
[16] X. Liu, Z. Lin, S.T. Acton, A grid-based Bayesian approach to robust visual tracking, Digital Signal Processing 22 (1) (2011) 54–65.
[17] L. Jing, P. Vadakkepat, Interacting MCMC particle filter for tracking maneuvering target, Digital Signal Processing 20 (2) (2010) 561–574.
[18] F. Porikli, Y. Ivanov, T. Haga, Robust abandoned object detection using dual foregrounds, EURASIP Journal on Advances in Signal Processing (2008), Article ID 197875, 11 pages, http://dx.doi.org/10.1155/2008/197875.
[19] C. Rossi, M. Abderrahim, J.C. Díaz, Tracking moving optima using Kalman-based predictions, Evolutionary Computation 16 (1) (2008) 1–30.
[20] S. Särkkä, A. Vehtari, J. Lampinen, Rao-Blackwellized particle filter for multiple target tracking, Information Fusion 8 (2007) 2–15.
[21] S. Shen, M. Tong, H. Deng, Y. Liu, X. Wu, K. Wakawbayashi, H. Koike, Model based human motion tracking using probability evolutionary algorithm, Pattern Recognition Letters 28 (2008) 1877–1886.
[22] I.K. Yalćin, M. Gökmen, Integrating differential evolution and condensation algorithms for license plate tracking, in: Proc. of the 18th International Conference on Pattern Recognition, ICPR'06, 2006, pp. 658–661.
[23] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (4) (2006).
[24] D.N. Zotkin, R. Duraiswami, L.S. Davis, Joint audio-visual tracking using particle filters, EURASIP Journal on Applied Signal Processing 2002 (1) (2002) 1154–1164.
[25] L. Zhu, J. Zhou, J. Song, Tracking multiple objects through occlusion with online sampling and position estimation, Pattern Recognition 41 (2008) 2447–2460.
[26] D.A. Gomez, P. Horain, M.K. Rajagopal, S.K. Karri, Real-time particle filtering with heuristics for 3D motion capture by monocular vision, in: Proc. of the IEEE International Workshop on Multimedia Signal Processing, MMSP 2010, pp. 139–144.
[27] C. Stauffer, W. Eric, L. Grimson, Learning patterns of activity using real-time tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 747–757.
[28] H. Uemura, S. Ishikawa, K. Mikolajczyk, Feature tracking and motion compensation for action recognition, in: Proc. of BMVA British Machine Vision Conference, BMVC'08, 2008.
[29] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: Proc. of 2009 IEEE 12th International Conference on Computer Vision, ICCV'09, 2009, pp. 104–111.
[30] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: Proc. of 2011 IEEE 13th International Conference on Computer Vision, ICCV'11, 2011, pp. 3457–3464.
[31] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, Real-time detection and tracking for augmented reality on mobile phones, IEEE Transactions on Visualization and Computer Graphics 16 (3) (2010) 355–368.
[32] S. Haykin, A. Zia, Y. Xue, I. Arasaratnam, Control theoretic approach to tracking radar: First step towards cognition, Digital Signal Processing 21 (5) (2011) 576–585.

J. Sánchez-Oro was born in 1987, in Madrid (Spain). He is MSc in Computer Science, MSc in Computer Vision at Universidad Rey Juan Carlos and Ph.D. candidate advised by Dr. Duarte and Dr. Martí. He is currently working at the CAPO research line (Computación de Altas Prestaciones y Optimización) at the GAVAB group of Universidad Rey Juan Carlos. His research interests include computer vision, image and video processing and meta-heuristic optimization.

D. Fernández-López was born in 1985, in Madrid (Spain). He is MSc in Computer Science, MSc in Computer Vision at Universidad Rey Juan Carlos and Ph.D. candidate advised by Dr. S. Montemayor and Dr. Pantrigo. He is currently working at the CAPO research line (Computación de Altas Prestaciones y Optimización) at the GAVAB group of Universidad Rey Juan Carlos. His research interests include computer vision, image and video processing, intelligent transport systems and parallel programming.

Raúl Cabido was born in Madrid (Spain) on July 11th, 1980. He received the BSc Degree in Computer Science from Universidad Rey Juan Carlos (URJC) in 2004, and the Ph.D. in Computer Science in 2010 from URJC. Nowadays, he is an Assistant Professor at URJC and the member of the GAVAB and CAPO research groups in the Department of Computer Science (DCC). His research interests focus on Computer Vision and GPU Computing. In particular, he uses GPUs to accelerate methods related to 3D/2D motion tracking, medical image reconstruction, image registration, face recognition, high-definition depth maps computation, image segmentation, super-resolution, acceleration of heuristics and metaheuristics procedures, etc.

A.S. Montemayor was born in 1975, in Madrid, Spain. He received his MS degree in Applied Physics at Universidad Autónoma de Madrid in 1999 and Ph.D. degree at Universidad Rey Juan Carlos in 2006. He is currently an Associate Professor at Universidad Rey Juan Carlos and the main leader of the CAPO research line at the GAVAB group. His research interests include soft-computing, computer vision, image and video processing and real-time implementations.

Juan J. Pantrigo was born in 1975, in Cáceres, Spain. He is currently an Associate Professor at Universidad Rey Juan Carlos. He received his MS degree in Fundamental Physics at Universidad de Extremadura in 1998 and Ph.D. at Universidad Rey Juan Carlos in 2005. His research interests include heuristic and exact optimization, computer vision, metaheuristics and hybrid approaches.