

# Triple-feature-based Particle Filter Algorithm Used in Vehicle Tracking Applications

Abdalgail Alsagair ABDULLA, Stevica GRAOVAC, Veljko PAPIC, Branko KOVACEVIC  
School of Electrical Engineering, University of Belgrade, 11000, Belgrade, Serbia  
aa155048p@student.etf.bg.ac.rs

**Abstract**—This work is oriented toward video tracking of vehicles in a typical traffic environment, based on particle filters. The proposed tracking algorithm is based on simultaneous usage of three different image features – color, edge orientation, and texture. All three features are related to the contents of a rectangular window that includes both the vehicle that is tracked and local background and they are represented in the form of appropriate histograms. Based on individual estimates produced by every single feature, the resultant estimate is made by their weighted averaged. Weighting factors are adaptively changing depending on the quality of a particular feature, estimated by calculations of average similarities between the reference window and the set of windows on particles' positions. The tracking accuracies of single-feature and three-features-based filters have been verified using the set of traffic sequences illustrating the presence of typical disturbances (shadows, partial and full occlusions, maneuvering etc.).

**Index Terms**—image color analysis, image edge detection, image sequence analysis, image texture analysis, particle filters.

## I. INTRODUCTION

Tracking of moving objects is the subject of many applications such as surveillance systems, traffic monitoring, security, etc. For most of these tasks, automatic visual tracking based on the processing of video sequences is used [1]. We have focused our attention on the tracking of vehicles on the road/street environment. A vehicle tracking in this context is the basis for higher-level tasks such as traffic control, event detection, extraction of information from the traffic flow, etc. [2-4]. Typical problems for reliable vehicle tracking in these circumstances are variable background and shadows, variable existence of other close objects (moving or stationary), partial or even full occlusions by elements of the scene, variable size of an object during the sequence, etc. [5-7]. Moreover, visual tracking is usually affected by the problems caused by camera vibration due to the wind, large lighting transitions, etc.

From a mathematical point of view, most of the encountered problems can be addressed to the modeling of non-linear, non-Gaussian, multi-modal systems, or some combination of these. Regarding the choice of tracking algorithm concept, we have been motivated by the fact that a *Particle Filter* (PF) is very robust for non-linear and non-Gaussian dynamic state estimation and works well where the clutter and occlusions are present. The work presented in [8] is usually referenced as an origin in this particular area.

This work was supported in part by the Public Enterprise "Roads of Serbia", through the project: Improvement of RWIS (Road Weather Information System).

Particle Filters are sequential Monte Carlo methods that are dependent on point mass representations of probability densities that are implemented to any state model [7]. In the case of tracking a moving target, the state model includes the object location (coordinates in the image) and the rate of their change (velocity), while the size of the object also could be added as an additional part of the model. In the context of the processing of a series of images, probability density functions consist of histograms representing the distributions of particular image features.

Tracking methods are usually categorized as generative and discriminative ones [9]. Generative tracking methods typically are used to search for the image region with minimal reconstruction error, while discriminative tracking is based on a separation of the object from the background. We have adopted here one generative method approach, assuming that the window (sub-image) of interest consists of a tracked vehicle and surrounding local background. Features characterizing this window make the mathematical representation of it.

The very first image feature used for these purposes was the histogram of colors (in RGB or HSV space). While color-based trackers are robust to object deformations, the practice has shown that PF based on this feature only, was not effective in many cases where the content of the tracked window was the subject of rapid changes (partial occlusion, strong shadows, etc.). There were several approaches based on other features: gradients and texture. All of these features have shown their own sensitivities and none could be suggested as the universal candidate. Gradient-based cues are sensitive to the existence of other "non-target" objects in the window, while their calculations might require a huge computational effort. The texture-based cues have relatively low sensitivity to the illumination variations, while they are highly sensitive to cluttered backgrounds [10].

The reasonable next step was the fusion of multiple cues in order to compensate for their individual sensitivities. This type of approach is increasingly present in many of applications derived from "Fuzzy systems and theory", evolving toward artificial intelligence applications, especially in the context of data fusion [11-12]. This was followed by some further research and developments, basically via specification of concept of application of signatures to "Expert system modeling" [13-14] as well as by specification of multi-relational classifiers based on correlation analysis, [15]. Moreover, this fundamental multi-model approach is evolved even toward civil engineering applications, [16].

There are many approaches of this type in the context of image processing, where the most frequent case consists in

combining the color feature with some representation of the object's edge [5], [17-18] or, color and texture [1], [19]. Since the PF has generally been a time-consuming algorithm, one should make the proper parameterization of the integrated algorithm, finding the compromise between the number of calculations (increasing by the adoption of new features), and the achieved tracking accuracy. In the algorithm shown here the fusion is made by weighted averaging of estimates based on the usage of three features (color, texture, and edge descriptors). Using a relatively small number of particles, the results obtained by this "Multi-Feature Particle Filter" (MFPF) have shown advantages relative to all separate "Single-Feature Particle Filter" (SFPF) algorithms and were still acceptable from the overall computing time aspect of view.

The very basic novel idea standing behind the approach suggested in this paper is based on the following assumptions:

1. The proper fusion of results obtained via usage of more than one feature will be more robust in comparison to the case of usage on one feature only, because of their different sensitivities to typical disturbances;
2. In comparison to the existing approaches based on combination of two image features, we extended the number of used features to three;
3. In order to save computational time, we assume relatively small number of particles to be used;
4. In comparison to more complex approaches based on application of Artificial Intelligence, our tracking algorithm has no need to "learn anything in advance". Its success in object's tracking and adaptability in this process, are based on the "on-line" measurements and the variable way of their combining.

A short overview of related work in this area is given in Section 2. Basic principles of object tracking based on PF as well as the choice of particular image features, their formalization, and individual efficiency are exposed in Section 3. In Section 4, the new algorithm that is consisting of a fusion of three individual SFPF is explained. Typical traffic situations are used as the verification cases in Section 5, where the set of obtained results, followed by appropriate discussion is given. Concluding remarks are given in Section 6. A list of the references is given in the last part of the paper.

## II. RELATED WORK

There are a huge number of references in the literature related to the moving object tracking in video sequences. Our research work was focused onto traffic applications and the matters of primary interest have been regarding to the general tracking strategy, appearance model of the tracked object, its motion model, overcoming the problems induced by typical disturbances as image clutter, shadows, and occlusions, as well as onto the ideas of fusion of information obtained by different ways.

### A. Tracking Strategies

According to the existing computer vision literature, there are four main approaches used for object video tracking [6]: 1) Model-based, which relies on the appearance and shape of tracked objects [20]. This strategy is basically threatened

by a lack of detailed geometric model information; 2) Region-based [21], where a sub-region of the image is created as the bounded box around the initially known position of a moving object. This area as the object of tracking is encompassing both road vehicle and a surrounded portion of a background, and generally provides higher robustness; 3) Contour-based strategy, which is relying on the extraction of an object's contour [6], [22]. It is belonging to a discriminative method because, as a prerequisite, some kind of image segmentation is used for distinguishing the object as a part of an image. When the bounded border of the object is specified, its representation becomes the information of interest in consecutive frames. While the computational complexity here could be acceptable, partial occlusion highly deteriorates contour representation and this is considered as the main problem; and 4) Feature-based approach consists in calculation of particular image features characterizing the object of interest and their following in consecutive frames [6], [23]. The simplest example is that the region of interest is represented as the set of pixels of different colors/light intensities, and that the same pixel pattern is searched in consecutive frames. This leads to time-consuming correlation-based methods. The region of interest is more frequently characterized by derived features like color distribution, orientations of edge-points, corners' locations, or texture. We have found that the region-feature-based strategy is the most appropriate for our purposes.

### B. Appearance Model

The appearance model might be related to the way how the tracked object itself is represented as well as to the way which features have been extracted as the description of a region of interest, and the way how they are formally represented. In characterizing the region, historically the very first approach used was the one based on color distribution [10]. These trackers are very robust relative to object deformations, while they are highly sensitive to the variations in illumination that could be compensated by some kind of model adaptation [24]. Several papers in this area are related to some representation of object/region based on the image contrast, leading toward the extraction of contour points/corners. A fundamental step in this characterization is consisting of the calculation of image gradients. Many approaches based on Harris Corner Detector – HCD [25], Scale Invariant Feature Transform – SIFT [26], Speeded-Up Robust Features – SURF [27] are used for detection and tracking purposes. Histograms of oriented gradients (HOG) have been used even for human detection [28], while this approach is also well applied to region contents description. An additional image feature applicable for region contents description is a texture (Local Binary Pattern – LBP, Grey Level Co-occurrence Matrix – GLCM, Harralick texture features, etc.). This feature has not been used too frequently in tracking applications, but still there are some examples for it [29].

### C. Motion Model

The vehicular motion along the road is assumed as the planar and translational one. The additional assumption that the magnitude and direction of a velocity vector characterizing the motion are constant is valid under the

assumption that the interval between measurements is short. If these intervals are prolonged due to the occlusion of an object, the prediction capabilities of the particular filter which is used are becoming important. Traditionally, Kalman filtering concepts in vehicle tracking was used as the first approach [30]. The need for explicit measurements of model states (ex. position of object's centroid, etc.) as well as the assumption about the Gaussian nature of measurement errors and state uncertainty, made this approach to be less attractive, in spite of the fact that this is still actual [31]. The Particle filtering (PF) does not assume precisely known dynamic behavior or type of uncertainty of a motion model. That is the reason why PF prevails in actual tracking applications [32-34]. This is an inherent advantage, allowing the PF to have good performances in the cases of typical image disturbances.

#### D. Disturbances

The typical disturbances in video tracking of road vehicles are related to the existence of variable shadows and reflections, as well as the occurrence of occasional partial or even complete occlusion of the tracked object. There are a few fundamental steps in these cases: 1) the existence of these irregular working conditions should be recognized, based on the model of shadow [35], or some indications that are characterizing the occlusion; 2) the action appropriate for this working regime should be made, as it is made via modification of the distribution of motion parameters [33], by analyzing the probability of object existence in a particular image region [36], by trying to find sub-regions that are not occluded [23], etc.; 3) tracking algorithm should be able to recognize when the irregular situation exists no more, usually based on some hypotheses about the object's behavior during the occlusion [37].

#### E. Fusion of Multiple Cues

An effective way to overcome the particular disadvantages of using some of the techniques mentioned before consists of a fusion of multiple cues. The relevant information may be produced by different sensors combining color and infrared [36], as well as by different color channels [9], or by combining color and segmentation cues [39]. The texture of the region could be combined with depth cues [40]. The adaptation abilities which are especially important in the presence of disturbances are mostly based on the multi-feature approach [18], [41] and it is adopted here as well.

As a kind of conclusion of this section where we have given the analysis of the literature in the field of "Visual Tracking", we can summarize our basic decisions as follows:

1. From the "Strategic" point of view we have adopted one mixture of strategies – our approach is based on the extraction of features, which are describing not the tracked object, but the rectangular sub-region around it, while one of the adopted features is implicitly oriented to the object's contour;
2. From the "Appearance model" point of view, our choice was to hold on usage of color, gradients, and texture, in parallel, while in all three cases, their interpretation should be equivalent – all of them are treated in uniform way – via particular histograms;

3. "Motion model" is adopted according to our preference to use Particle filter as the most appropriate approach in the presence of typical disturbances affecting the moving road vehicles;
4. The effects of "Disturbances" are not identified via attempts to recognize some particular type, but by indirect detection via decreasing of the overall similarity measures;
5. Triple-feature-based "Fusion" of results is dynamically based on variable relative importance of each of them.

### III. PARTICLE FILTER APPLIED IN VIDEO TRACKING

The problem of tracking of moving objects is present in many tasks related to military applications (surveillance, sighting, homing toward targets, etc.), security applications (registering of presence of moving persons and/or vehicles in protected area), traffic control (observation of traffic situation – counting of the moving vehicles, detection of their presence in some restricted zones, estimating their velocities, etc.) and in many applications in the fields of robotics and autonomous vehicle guidance (detection and tracking of stationary and/or moving obstacles).

Any task of this type starts by detection of an object which should be tracked. This phase can be automated or made by human operator. From this point on, the process should be completely automatic. Based on the assumed model of motion (2D – 3D, continuous – with stops, maneuverable – non-maneuverable, etc.), tracking algorithm predicts the next position of object. After the new set of measurements is made, the estimated position of object is corrected and the procedure is continued. The way how the set of previous measurements and estimates are used in the next prediction, varies between different filters/estimators (data block processing, iterative ones – based on balancing between state and measurement uncertainties, based on maximal likelihood, etc.)

Our choice from tracking algorithm point of view is given in this section.

#### A. Fundamentals of Particle Filtering

In general, PF recursively builds the posterior probability density function (PDF) of the state space using the Monte Carlo method [1], [22], [42-43]. The main concept of PF is to represent the posterior PDF across a group of randomly chosen weighted samples known as particles. Each particle expresses a hypothesis of the state, and it is randomly scattered from the preceding density and diffused based on the transition model. Each spread particle is weighted using the likelihood model – the weight describes the quality of a particle [5]. The posterior probability density function is approximated recursively by a group of weighted particles sampled randomly. The state of the tracked object at time instant,  $t$  is denoted by  $x_t$  and is represented through a set of samples in the motion space, while the observation of the system is denoted by  $z_t$ . At each time instant, the PF makes two repeated steps: prediction and update [5], [44]:

- Prediction: For each frame  $k$ , each particle is distributed independently by propagating the samples according to the system model. The step gives an approximation of the prior PDF:

$$p(x_t) \approx \frac{1}{N} \sum_{i=1}^N \delta(x_t - x_t^i) \quad (1)$$

where  $N$  is the total number of particles,  $i=1, 2, \dots, N$  is the particle index, and  $\delta(x_t - x_t^i)$  is the Dirac delta function.

• Update: The updating step is achieved by calculating the weight of each particle based on the observation model (likelihood model). A discrete approximation of the posterior PDF at time instant  $t$  can be written as follows:

$$p(x_t / z_{t:1}) \approx \frac{1}{N} \sum_{i=1}^N w_t^i \delta(x_t - x_t^i) \quad (2)$$

with:

$$w_t^i \approx w_{t-1}^i \frac{p(z_t / x_t^i) \cdot p(x_t^i / x_{t-1}^i)}{q(x_t^i / x_{0:t-1}^i, z_{1:t})} \quad (3)$$

where:  $p(z_t / x_t^i)$  is the likelihood function,  $p(x_t^i / x_{t-1}^i)$  is the state transition density distribution,  $q(x_t^i / x_{0:t-1}^i, z_{1:t})$  is a proposed distribution (importance distribution/ function) in which the particles are samples and  $w_t^i$  are normalized weights, so that  $\sum_{i=1}^N w_t^i = 1$ .

PF tracking algorithm in video tracking context consists of the following steps:

- State transition – prediction, using the adopted motion model;
- A computing of selected image features;
- Measurements – computation of particle weights (likelihood estimation), based on selected image feature(s);
- Estimation of state, based on maximal likelihood, and
- Resampling – a selection of the new set of particles for the next iteration [5].

### B. State Transition

Due to an assumption that the vehicular movement along the road is planar and translational one and that the magnitude and direction of a velocity vector are constant in the short interval between measurements, the general state model defined as follows:

$$\bar{X}_k = \begin{bmatrix} x \\ y \\ V_x \\ V_y \end{bmatrix}_k = F \begin{bmatrix} x \\ y \\ V_x \\ V_y \end{bmatrix}_{k-1} + \bar{v}_k \quad (4)$$

This model reduces to:

$$\bar{X}_k = \begin{bmatrix} x \\ y \end{bmatrix}_k = \begin{bmatrix} x \\ y \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}_{k-1} + \bar{v}_k \quad (5)$$

where the state vector consists of a position and velocity components in Image Coordinate Frame (ICF), while  $\bar{v}_k$  is a white Gaussian process noise,  $\Delta x$  and  $\Delta y$  are the position increments determined in the previous step.

The validity of the previously stated assumptions is compromised if the sampling intervals are relatively long, simultaneously followed by the fact that the vehicle is accelerating/braking, or makes the angular maneuvers (especially in the intervals when it is occluded).

### C. Selection of Image Features

#### 1) Color feature

A variety of color spaces can be used in tracking applications: RGB (red, green, blue), HSV (hue, saturation, value), and YCbCr (luminance and chrominance). The RGB color space is most frequently used. The color representation can be made via local-global color histograms, partitioning of color spaces and use of a clustering algorithm, etc. [45]. The color histogram is most frequently used for this purpose due to its invariance to rotation variations and acceptable robustness to scaling variations [45-46]. In turn, it has poor performance when the illumination changes and when the background has a similar color as the target object [47]. The color histogram used in our approach consists of three consecutive histograms (R, G, B), with 21 bins each, producing the compound histogram of 63 bins. Typical color histograms for candidate windows in few different frames from a real sequence are shown in the second column of Fig.1.

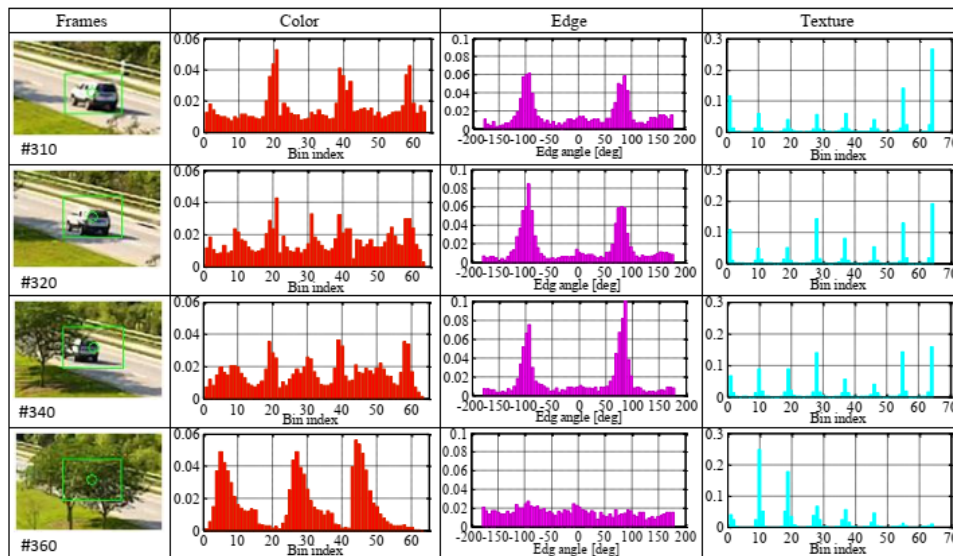


Figure 1. Histograms of Color, Edge, and Texture features for the candidate windows in real video sequence (frames: #310 - initial, #320 – with shadow, #340 – partial occlusion, and #360 – full occlusion)

## 2) Edge (border) feature

The object's edge is the set of pixels with high intensities of a gradient. The edge feature gives shape information which has proven to be relatively effective in visual contour tracking. The sensitivity of the edge feature to illumination changes is less in comparison to color features [48]. On the other hand, this representation of the object's contour is usually sensitive to the existence of some other pixels in the image that don't belong to the object but are satisfying the high gradient intensity criterion. In our work, the edge feature is specified as a histogram representing a distribution of gradient directions for the edge points.

Many gradient-based operators are used for distinguishing the edges of the object (Sobel, Canny, Roberts, and Prewitt). The edges are extracted in our approach by estimating Sobel operators  $K_x$  and  $K_y$ , applied on the grey-scale image. The horizontal gradient  $G_x$  and the vertical gradient  $G_y$  have magnitudes given by:

$$\begin{aligned} G_x(x, y) &= K_x \times I(x, y) \\ G_y(x, y) &= K_y \times I(x, y) \end{aligned} \quad (6)$$

where,  $K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ ,  $K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$  and  $I(x, y)$  is the candidate window.

The magnitude ( $G$ ) and phase ( $\theta$ ) of the gradients along edges are determined as follows:

$$\left. \begin{aligned} G(x, y) &= \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \\ \theta(x, y) &= \tan^{-1} \left( \frac{G_y(x, y)}{G_x(x, y)} \right) \end{aligned} \right\} \quad (7)$$

Histogram of gradient phase angles is the distribution characterizing the object's contour. The overall span of phase angles is from  $-180^\circ$  to  $+180^\circ$  for the unsigned gradient, and it is divided into 64 equal bins. The steps for finding the histogram of contour points' phases of gradients are:

- Find the pixels (positions) where the magnitude ( $G$ ) is determined to be above some specified threshold (e.g., 80% of maximal magnitude in this window);
- Obtain the phases ( $\theta$ ) of the gradients corresponding to the extracted pixels;
- Calculate the histogram of these phase angles;

The histograms of gradient phase angles for selected windows are shown in the third column of Fig.1.

## D. Texture feature

The texture is a scale of the intensity change of a surface that defines properties like regularity, linearity, uniformity, density, directionality, smoothness, and coarseness [49]. Texture demonstrates the spatial arrangements of intensity values or color in an image, which may be periodic, stochastic, or both [17]. There are different approaches used for describing the texture by structural, statistical or spectral approaches [50]. We have chosen the statistical approach, while we tested three of them:

- Local Binary Patterns (LBP) - a grey-level scale used to describe texture characteristics of the surface, computed by comparing eight neighbors for each pixel 'p' in the image. The neighbor pixel becomes 1 if its value is greater

or equal to the central pixel 'p' value; otherwise, its value becomes 0. Neighbors are labeled using binary code {0, 1}. A histogram is built to represent the frequency of occurrences of particular binary patterns over the image [44].

- Grey Level Co-occurrence Matrix (GLCM) – illustrates the number of occurrences of different combinations of gray levels for every pixel image and its specified neighbor. Several statistic parameters can be extracted from GLCM, like maximum probability, contrast, homogeneity, uniformity (also called Energy), and entropy. These statistic parameters (Harralick descriptors) expressed as a vector provides information about the texture of an image.

- GLCM elements converted to a vector (histogram) which has appeared to be the most successful in our case and adopted here. The selected size of the GLCM matrix was 8x8 (eight classes of grey level). We used a 1-pixel offset, and the orientation was towards the east (from left to right). Elements of GLCM (number of occurrences) become probabilities after normalization. The probabilities of occurrence being the elements of GLCM are converted to a vector of size 64, forming the histogram type descriptor with 64 bins. The fourth column in Fig.1 shows the GLCM based histograms.

The illustration is done for the set of tracking windows in a selected sequence of frames, typical for traffic scenarios. It is obvious that three features are not equally sensitive to the disturbances acting during this sequence. Edge feature histograms are pretty stable in the cases of existing shadow and partial occlusion (appropriate histograms are of recognizable bipolar nature). Color histograms are more sensitive to these influences, while histograms generated by GLCM are somewhere in between. In the case of full occlusion (last row), none of histograms is informative at all.

## E. State estimation

Measurements made in each new frame consist of the calculation of the chosen image feature for each particle. Whichever single feature is used, the appropriate histogram is representing the contents of the considered particle (window). The weight of the particle is proportional to the similarity measure. In our approach, we have adopted Bhattacharyya distance ( $BD$ ) between the actual particle (window) histogram and reference one, as the information about local similarity at a particular position. Bhattacharyya distance is calculated as follows:

$$BD^j(k) = \sqrt{1 - \sum_{u=1}^m \sqrt{hist_u^i(k) \cdot hist_u^{ref}(k-1)}} \quad (8)$$

where:  $u$  – bin index,  $i$  – particle index,  $k$  – frame index,  $m$  – number of bins,  $hist_u^i$  and  $hist_u^{ref}$  stand for the histograms of a considered particle and the reference one.

Reference histogram is obtained using the initial one  $hist(0)$ , which is characterizing the initial window around the object and the previous one  $hist(k-1)$ , which is characterizing the window around the last estimated position of the object.

$$hist^{ref}(k) = \alpha \cdot hist(0) + (1 - \alpha) \cdot hist(k-1) \quad (9)$$

$\alpha \in [0,1]$  - weighting coefficient.

The weight of the particle is calculated using an exponential function as given in equation (10), where the deviation was selected as  $\sigma = 1.5$ .

$$w^j(k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(BD^j)^2}{2\sigma^2}} \quad (10)$$

The weights are then normalized:

$$\bar{w}^j(k) = \frac{w^j(k)}{\sum_{j=1}^N w^j(k)} \quad (11)$$

A new estimation of state (position of the tracked object) can be adopted as the position of the particle which is the most similar to the reference template (with the highest value of  $w^j(k)$ ),

$$(x_k, y_k) = (x_{\max}^j, y_{\max}^j), \quad (12)$$

$$w_{\max}^j(k) = \max(w^j(k)) \quad \text{for } i = 1, 2, \dots, N$$

or as a weighted sum:

$$(x_k, y_k) = \sum_{i=1}^N \bar{w}^i(k) \cdot (x^i, y^i) \quad (13)$$

#### F. Resampling

When a few particles, due to their high similarity measures, become dominant, an effective number of particles are highly reduced. This is known in PF literature as a “particle degeneracy problem” and it is affecting the tracker’s ability to respond properly to possible stochastic changes in the object’s motion. It is usual to check whether the effective number of particles ( $N_{\text{eff}}$ ) which is determined using equation (14), becomes less than some threshold value. If that’s the case, a resampling process is started.

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w^i)^2} \quad (14)$$

Among the number of existing resampling procedures [51], our choice was the algorithm that follows:

(i) Generate a standard uniform random number,  $u_j \sim U[0,1]$

(ii) Calculate the *Cumulative Density Function* (CDF) of normalized particles’ weights:

$$wc^i = \sum_{s=1}^i \bar{w}^s \quad (15)$$

(iii) *FOR*  $j=1: N$ , choose the particle  $x^j$  for replication from the original set of particles, based on the condition:

If  $u_j \in [wc^{j-1}, wc^j]$ , then  $x^j = x^i$

*END FOR*  $j$

Fig.2 shows the generation of events ( $I_j$ ), using CDF graphically.

As a result of resampling, some particles with lower weights will be omitted, and some higher weight particles will be multiplied.

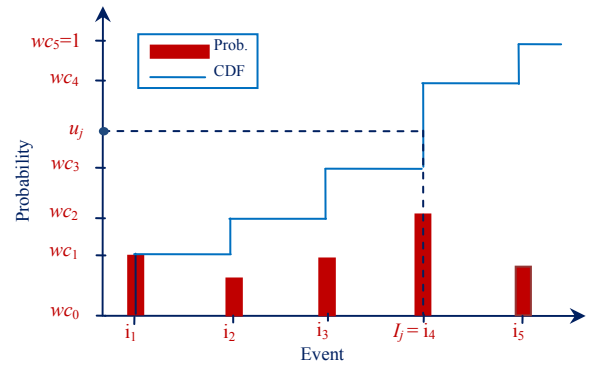


Figure 2. Probabilities of the events (brown), the CDF (solid) and a transformation from standard uniform number to the event (dashed) [51]

#### IV. MULTI-FEATURE-BASED TRACKING ALGORITHM

*Single-Feature-based Particle Filter* (SFPF) used for vehicle video tracking suffers from different individual disadvantages depending on actual visibility conditions and existence of typical disturbances. All image features mentioned in Section 3 are legitimate candidates to be applied for SFPF. The main idea behind this *Multi-Feature-based Particle Filter* (MFPPF) was that, by fusion of the particular SFPF results, one is able to overcome particular drawbacks of each SFPF. Based on the estimated quality of all SFPF outputs, the relative importance of these outputs will be adaptively changed in order to give preference to image features that are less affected by the existing disturbance condition.

In comparison to the basic SFPF algorithm described in Section 3, there are some additional algorithm steps and parameters that should be specified in the building of MFPPF:

- Initial distribution of particles – made as the same one used for all three single features (*SF*). Initial dispersion radius,  $DR$ , used in the first step has got a doubled value of radius  $DR_0$ , used for subsequent steps, due to the high initial uncertainty of the object’s motion.
- Average similarity,  $AS_n$  for any  $SF_n$ , is calculated based on an average value of Bhattacharyya distances,  $BD^i$ , for all  $N$  particles. This is a basic parameter specifying the relative importance of a particular feature.
- Particular similarity measure is defined as:

$$\text{sim}^i(k) = 1 - BD^i(k) \quad (16)$$

- Level of confidence,  $LC_n$  for any  $SF_n$ , is specified as the ratio of actual  $AS_n$  and initial  $AS_n(1)$  existing in the first frame after initialization. Generally, this level of confidence could be greater than 1 (if some consecutive average similarity  $AS_n$  is greater than  $AS_n(1)$ ).
- Relative importance (weighting factor,  $WF_n$ ) for each  $SF_n$  is obtained after the normalization of all three values of  $LC_n$ .

$$WF_n(k) = \frac{LC_n(k)}{\sum_{j=1}^3 LC_j(k)} \quad (17)$$

Weighting factors are used in final state estimation:

$$(x_{obj}, y_{obj}) = \sum_{n=1}^3 WF_n \cdot (x_{\max}^n, y_{\max}^n) \quad (18)$$



- If the particular  $AS_n$  is less than the specified threshold value,  $AS_n^{thr}$  its  $LC_n$  goes to zero and particular  $SF_n$  is not used in final state estimation.
- If all  $SF_n$  are less than their threshold values (which is typical for full occlusion), PF works in “prediction mode”, based on previous predictions, without valid new measurements.
- Radius of particle dispersion in this mode is gradually increased during the occlusion phase, while it is doubled in comparison to the nominal value  $DR$ , used in “estimation mode”.
- Reference model (reference histograms) during “prediction mode” stay equal as in case of the window content on the last estimated position in “estimation mode”.
- Resampling in MFPF algorithm environment has a slightly different meaning than in the case of SFPF. After resampling procedure is done for all three features, final step is the extraction of particles that belong to all three sets of resampled particles simultaneously.

The overall MFPF flow chart algorithm is shown in Fig.3, where:

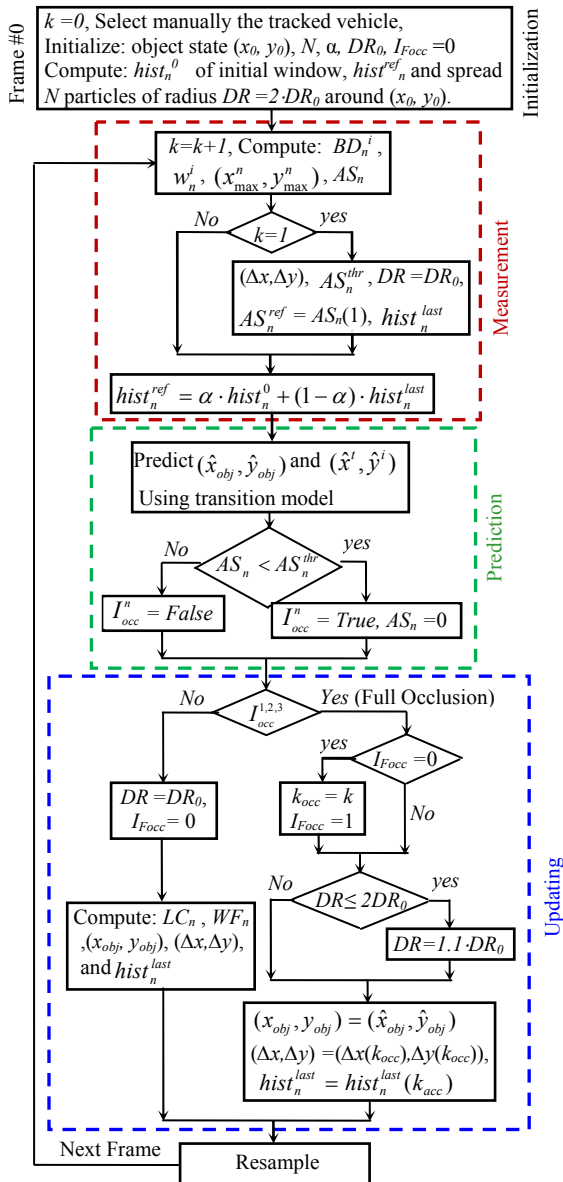


Figure 3. MFPF flow chart algorithm

- $I_{occ}^n$  are the flags for appearance of occlusion in  $SFPF^n$ ;
- $k_{occ}$  is the starting time of the occlusion;
- $I_{Focc}$  is a flag for full occlusion;
- $(x_{max}^n, y_{max}^n)$  positions with largest weights for  $SFPF^n$ ;
- $hist_n^0$  are initial reference histograms of  $SFPF^n$ ;
- $hist_n^{ref}$  are updated reference histograms of  $SFPF^n$ ;
- $hist_n^{last}$  are the last updated histograms;
- $(x_{obj}, y_{obj})$  resultant estimate of object's position;
- $(\hat{x}_{obj}, \hat{y}_{obj})$  predicted object's position;
- $(\hat{x}^i, \hat{y}^i)$  predicted particle's positions.

As it is shown in the flowchart on Fig. 3, the modeling steps are:

1. **Initialization:** Select manually the tracked object bounded by a rectangular window at its position  $(x_0, y_0)$ . Compute the initial histograms for the selected window which are set as reference ones. Generate the set of  $N$  particles with radius  $DR_0$  around  $(x_0, y_0)$ ;
2. **Measurements:** Starting from the first frame, calculate Bhattacharyya distances (equation 8). Calculate the similarity measures for all  $N$  particles (equation 16). Calculate the particle's weight (equation 10). Normalize the weight (equation 11), and modify reference histograms (equation 9);
3. **Prediction:** Predict the moving vehicle's state  $(\hat{x}_{obj}, \hat{y}_{obj})$  and positions of particles based on state transition model (equation 5);
4. **Updating:** If there is no full occlusion, calculate the level of confidence parameter  $LC_n$ , and weight factors  $WF_n$  (equation 17). Calculate updated increments and estimated vehicle's position (equation 18). Calculate histograms on updated position. In the case of full occlusion at  $k^{th}$  frame, set the last valid histogram (one at the moment before occlusion) and use the last position increment for the whole period of occlusion;
5. **Resampling:** Resample the particles in all three features by omitting and replacing all particles that meet the elimination criterion, according to section E

## V. EXPERIMENTAL RESULTS

Different sequences of images illustrating realistic traffic scenarios are analyzed here. In all these cases the overall scene is shown as a first, followed by the set of frames during the tracking process. Average similarities for three image features are shown graphically with specified threshold levels, which determine when a particular feature is becoming invalid and is excluded from the estimation process. The other type of graphical illustration is used to show how the tracking error is changing (distance between estimated vehicle position and ground through data, in pixels). These data have been used to calculate the *Root Mean Square* (RMS) tracking error specified in equation (19).

$$RMS = \sqrt{\frac{1}{i_{max}} \sum_{i=1}^{i_{max}} [(x_{es} - x_{tr})^2 + (y_{es} - y_{tr})^2]} \quad (19)$$

where:

- $i_{max}$  is a number of validation points;

- $x_{es}, y_{es}$  – are the estimated object coordinates in pixels, and
- $x_{tr}, y_{tr}$  – are the true object coordinates in pixels.

RMS errors are shown finally in Table I, and it is used as the condensed way of comparison of tracking accuracies of individual three SFPFs and proposed MFPP.

#### A. Case-1: Influence of Small Partial Occlusion

This simple example represents the tracking of a car, which is from time to time partially occluded by surrounding trees as it is shown on Fig. 4.



Figure 4. Illustration of Case 1

Average similarities for “Edge” and “Texture” features are generally above their threshold values all the time, as it is shown in Fig. 5 (a), while the “RGB” feature is maximally affected and in the final phase of sequence, it is even excluded from the estimation process. From Fig. 5 (a) one can conclude that the “Integrated” filter is not superior in any isolated particular frame in comparison to particular SFPFs. However, one can see in the first column of Table I that the RMS tracking error on the whole interval is the least in the case of MFPP.

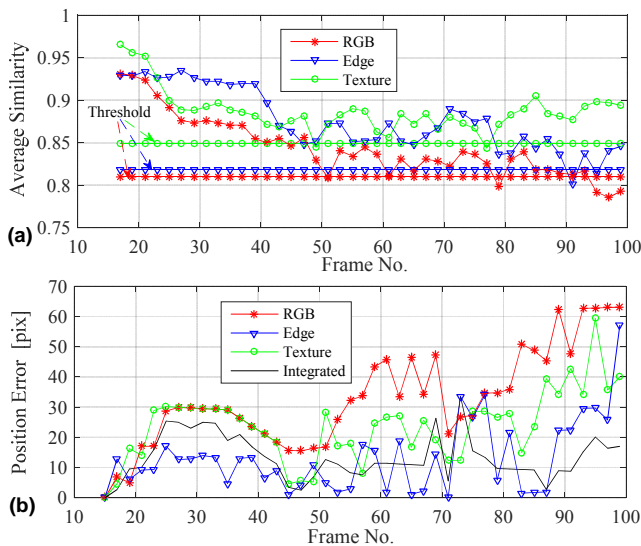


Figure 5. Case-1: (a) Individual average similarities, (b) Individual SFPF errors Vs. MFPP error

#### B. Case-2: Influence of Longer partial Occlusion

In this example the incoming car is partially occluded during the set of frames while it is overtaking another vehicle as it is shown in Fig. 6.

Due to the fact that initial average similarities of all three features was not high, the appropriate threshold values are relatively low, practically always keeping all of SFPFs as components in the estimation process (see Fig. 7). Edge

feature is the most sensitive having in minds that the contour orientations of vehicle which is tracked are also characterizing the vehicle which is being overtaken. Second column in Table I shows that all three SFPFs are producing pretty similar accuracies, while MFPP is slightly better of them.



Figure 6. Illustration of Case 2

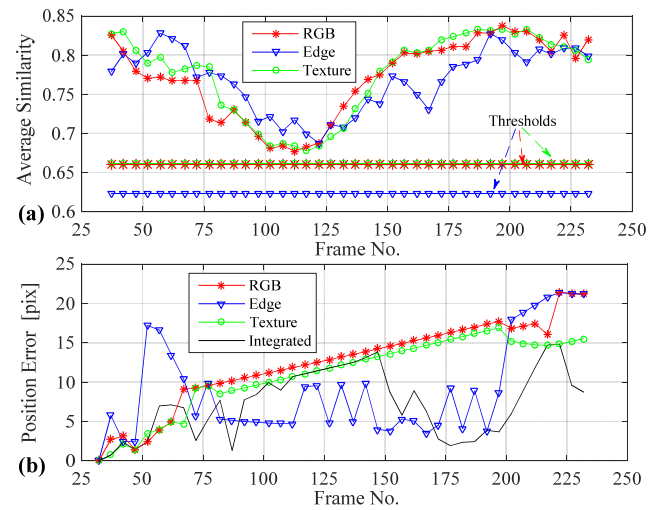


Figure 7. Case-2: (a) Individual average similarities, (b) Individual SFPF errors Vs. MFPP error

#### C. Case-3: Influence of Shadows

This is the case where the background is variable due to the existence of shadows produced by trees as it is shown in Fig. 8.



Figure 8. Illustration of Case 3

The fact that some amount of shadowed road is always present, makes that all three average similarities on Fig. 9 (a) are always above their threshold values. Fig. 9 (b) shows that “Edge” feature-based SFPF is the most sensitive one because of dependence on the actual amount of shadows, the tracking error oscillates between minimal and maximal one for the whole sequence. MFPP tracking error behaves as the most stable. The third column of Table I shows that RMS tracking error is superior in the case of MFPP.



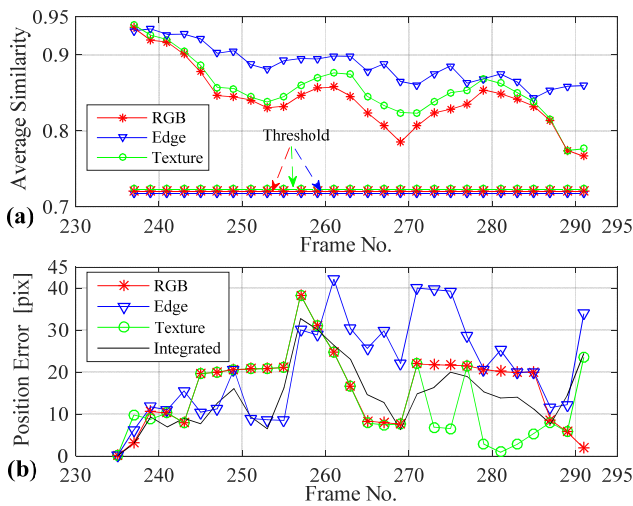


Figure 9. Case-3: (a) Individual average similarities, (b) Individual SFPF errors Vs. MFPP error

#### D. Case-4: Influence of Full Occlusion

This is an example where the tracked vehicle is occluded by the other vehicle moving along the neighboring road. In one relatively long interval of time (about 70 frames), the tracked vehicle is not visible at all (Fig. 10).

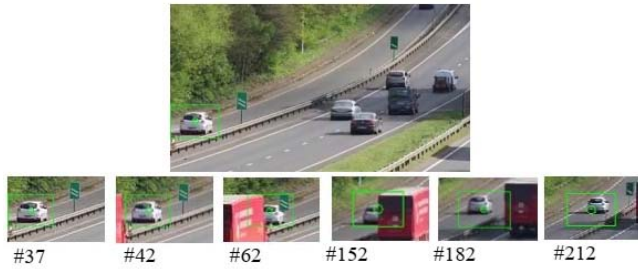


Figure 10. Illustration of Case 4

Fig.11(a) shows that in the interval of full occlusion, all three average similarities are below threshold values, and MFPP works in “Prediction” mode. Starting from frame #150, the “Edge” feature produces meaningful results, and MFPP practically works as SFPF based on the “Edge” feature. The fourth column in Table I is evidencing that MFPP guaranties the best tracking accuracy in the RMS sense on the whole interval.

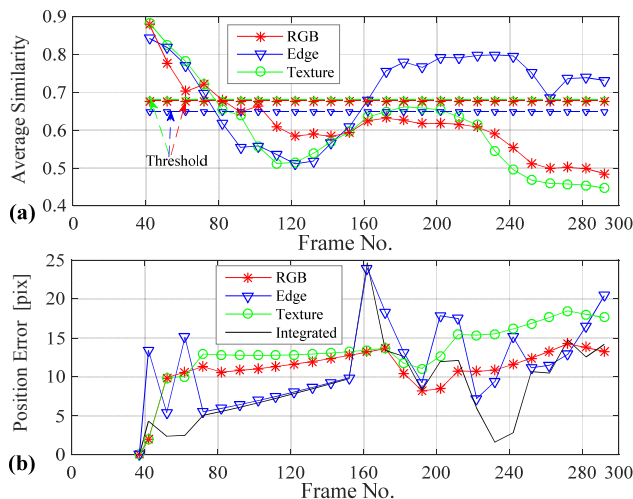


Figure 11. Case-4: (a) Individual average similarities, (b) Individual SFPF errors Vs. MFPP error

#### E. Case-5: Combined Effect of Shadows and Full Occlusion

The following example combines the effects of variable shadows and partial occlusions, including some short intervals of full occlusion (Fig. 12). According to Fig. 13 (a), the “Edge” feature is the least sensitive to these disturbances. However, it is evident that in interval #350 - #380, MFPP works in “Prediction mode” due to full occlusion of the tracked vehicle. Immediately before and after this interval, MFPP was guided by “Edge” feature SFPF only. Outside this, there is an almost equal influence of the other two features.



Figure 12. Illustration of Case 5

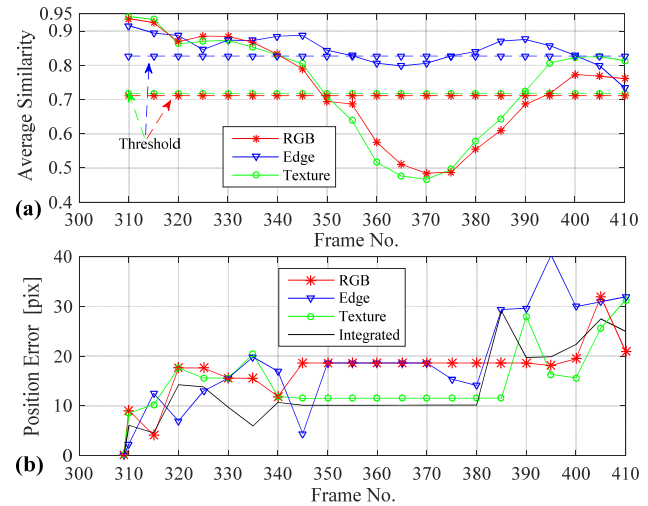


Figure 13. Case-5: (a) Individual average similarities, (b) Individual SFPF errors Vs. MFPP error

#### F. Case-6: Influence of full occlusion accompanied by maneuvering

This is the case where the tracked vehicle is subject to partial and full occlusion, while it makes a maneuvering motion (turning to the left) during the interval of full occlusion. In the interval of full occlusion (Fig. 14), there is a very short interval when all three features are producing invalid estimates (due to the fact that average similarities are less than appropriate thresholds).

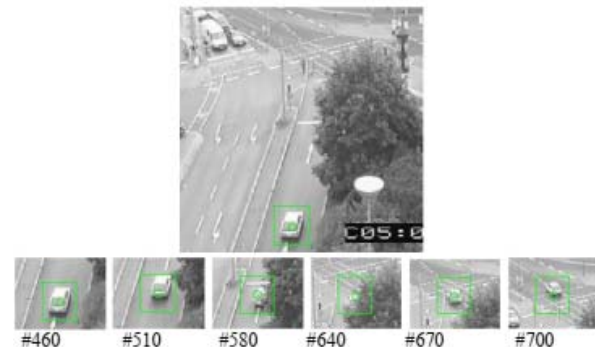


Figure 14. Illustration of Case 6

When the occlusion has vanished, both “RGB” and “Texture” features are becoming acceptable sources of information (Fig. 15). Results in the sixth column of Table I clearly show the superiority of the MFPP approach.

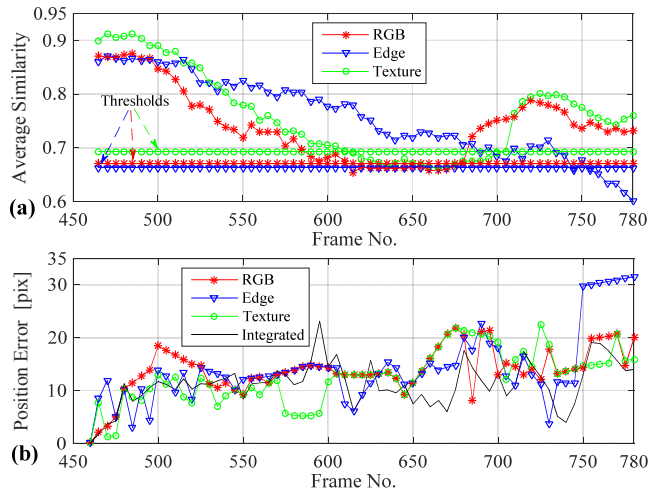


Figure 15. Case-6: (a) Individual average similarities, (b) Individual SFPP errors Vs. MFPP error

The RMS errors illustrating SFPP and MFPP accuracy performances, obtained for six scenarios, are finally presented in Table I. It is obvious that the MFPP has produced the least RMS tracking error in all six cases. Comparing the results obtained for individual SFPP, one can conclude that there is no particular SFPP that could be declared as the best in all cases. There are the cases where each one of them might be the best candidate, and at the same time, this SFPP could be the worst in some other scenarios. Three-feature-based filter that uses all three image features produces RMS tracking errors that are always for 10% – 25% less than the least one among SFPP RMS tracking errors.

TABLE I. RMS TRACKING ERRORS (PIXELS)

		Case1	Case2	Case3	Case4	Case5	Case6
SFPP	RGB	36.40	13.50	18.51	11.23	17.81	14.52
	Edge	17.61	10.89	24.01	12.54	21.02	15.77
	Texture	26.35	11.95	16.69	13.66	16.04	13.50
MFPP		15.21	8.38	16.16	9.97	15.11	11.93
MFPP Relative to the best SFPP [%]		86.37	76.95	96.82	88.78	94.20	88.37

There are two additional statistical measures of quality added in order to point out comparative advantages of MFPP algorithm. Table II illustrates probabilities that any of particular SFPP have had the average similarities above its particular threshold that was set as 80% of initial one. In the case of MFPP, final row in Table II illustrates the probabilities that MFPP is not in prediction mode. It is obvious that while particular SFPP in some cases loose the tracked object, MFPP has to go in prediction mode only when all SFPP are unacceptable and it is less frequently in comparison to any SFPP.

Finally, Table III illustrates the probabilities that the tracking error is less than threshold value of 15 pixels. According to the set of cases analyzed before, average distances from the tracked objects, and their sizes, as well as

the image resolution, we have adopted this threshold value as relevant one. One can conclude that MFPP is superior from this aspect also.

TABLE II. PROBABILITY THAT THE AVERAGE SIMILARITY IS ABOVE ITS THRESHOLD

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
SFPP	RGB	88.0 %	100 %	100 %	19.2 %	57.1 %	79.7 %
	Edge	97.6 %	100 %	100 %	69.2 %	76.2 %	89.1 %
	Texture	95.2 %	100 %	100 %	15.4 %	61.9 %	71.9 %
MFPP		100 %	100 %	100 %	73.1 %	85.7 %	100 %

TABLE III. PROBABILITY THAT THE TRACKING ERROR IS LESS THAN 15 PIXELS

		Case1	Case2	Case3	Case4	Case5	Case6
SFPP	RGB	6.9 %	63.4 %	58.6 %	96.3 %	18.1 %	70.7 %
	Edge	67.4 %	78.0 %	34.4 %	70.3 %	31.8 %	76.9 %
	Texture	23.2 %	78.0 %	62.0 %	66.6 %	59.0 %	72.3 %
MFPP		62.7 %	100 %	62.0 %	96.3 %	72.7 %	84.6 %

## VI. CONCLUSION

This paper introduces the algorithm of video tracking of vehicles based on a particle filter that uses a fusion of three individual filters based on three single image features. All three features (the color, the orientation of gradients, and gray-level-co-occurrence-matrix) are represented in the standard form of histograms. Bhattacharyya distance measure is used to specify the particle position of maximal similarity of the appropriate histogram with the reference one, in order to estimate three candidates for the estimated position of tracked object. The final estimate is obtained via weighted averaging of three candidates using adaptive weighting factors based on average similarities on the set of particles. Individual particle filters based on single feature have different sensitivities relative to typical disturbances affecting the tracking process like variable shadows and reflections, partial and full occlusions, variable background, etc.

A number of typical traffic scenarios were analyzed in order to verify the performance of the proposed multi-feature-particle-filter. While in some sub-intervals of tracking sequence this MFPP was inferior relative to the best SFPP, it has appeared as superior in the sense of RMS tracking error through the whole sequence. RMS errors on the whole interval have been from 10 % to 25 % less than the least RMS errors of any SFPP. As it was shown in Tables II and III, MFPP algorithm is also superior relative to any of SFPP because of less frequent need to go to prediction mode, as well as because of higher probability that the tracking error is inside the specifies tolerance zone.

The computational burden due to the tripling of particle filter calculation times was reduced by adopting of a relatively low number of particles (200). By appropriate specification of minimal acceptable average similarities, the algorithm excludes some of SFPP results if their average similarity regarding a particular feature is below the threshold. If all three features produce unacceptable results (full occlusion) the algorithm continues the work in prediction mode, reinitializing the distribution of particles

and gradually increasing the area where they are distributed. Weighting factors that characterize individual influences of image features are adapted automatically, according to their average similarities in the set of particles. Parameters of MFPF algorithm have been constant here for all illustrative cases. Wherever exists some higher level of a prior knowledge about traffic scenario on the particular part of road, there is a possibility for re-parameterization of MFPF (thresholds, window size, number of particles, forgetting factor, etc.) in order to adjust to this situation, and to improve the tracking accuracy.

## REFERENCES

- [1] L. Mihaylova, P. Bransett, N. Canagarajah, and D. Bull, "Object tracking by particle filtering techniques in video sequences," In E. Lefebvre, & N. S. Series (Ed.), *Advances and Challenges in Multisensor Data and Information Processing*, 2007, vol. 8, pp. 260-268, ISBN print 978-1-58603-727-7
- [2] S. A. Meshram, and A. V. Malviya, "Vehicular traffic surveillance for real time using multiple methodologies," *International Journal of Scientific & Engineering Research*, 2013, vol. 4, no. 5, pp. 1988-1992
- [3] R. Kachacha, and J. M. Cañas, "Hybrid three-dimensional and support vector machine approach for automatic vehicle tracking and classification using a single camera," *Journal of Electronic Imaging*, 2016, vol. 25, no. 3, pp. 033021. doi:10.1117/1.JEI.25.3.033021
- [4] S. Srivastava, and E. J. Delp, "Video-based real-time surveillance of vehicle," *Journal of Electronic Imaging*, 2013, vol. 22, no. 4, pp. 041103. doi:10.1117/1.JEI.22.4.041103
- [5] N. K. Ki, and E. J. Delp, "New models for real-time tracking using particle filter," *Visual Communications and Image Processing*, 2009, vol. 7257, pp. 72570B. doi:10.1117/12.807311
- [6] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, 1998, vol. 6, no. 4, pp. 271-288. doi:10.1016/S0968-090X(98)00019-9
- [7] M. Z. Islam, C.-M. Oh, and C. W. Lee, "Video based moving object tracking by particle filter," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2009, vol. 2, no. 1, pp. 119-132
- [8] M. Isard, and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, 1998, vol. 29, no. 1, pp. 5-28. doi:10.1023/a:1008078328650
- [9] G. Han, S. Dong, N. Sun, J. Liu, K. Du, and X. Li, "Robust local L2,1 tracker via red-green-blue color channel fusion," *Journal of Electronic Imaging*, 2016, vol. 25, no. 5, pp. 053002. doi:10.1117/1.JEI.25.5.053002
- [10] W. Dong, F. Chang, and Z. Zhao, "Visual tracking with multifeature joint sparse representation," *Journal of Electronic Imaging*, 2015, vol. 24, no. 1, pp. 013006. doi:10.1117/1.JEI.24.1.013006
- [11] P. Angelov, "Fuzzily connected multi-model systems evolving autonomously from data streams," *IEEE Transactions on Systems, Man, and Cybernetics - Part B, Cybernetics*, Vol. 41, No. 4, 2011, pp. 898-910. doi:10.1109/TSMCB.2010.2098866
- [12] P. Angelov, R. Yager, "Density-based averaging - A new operator for data fusion," *Information Sciences*, Vol. 222, 2013, pp. 163-174. doi:10.1016/j.ins.2012.08.006
- [13] C. Pozna, N. Minulete, R.-E. Precup, L. T. Kóczy, Á. Ballagi, "Signatures: Definitions, operators and applications to fuzzy modeling," *Fuzzy Sets and Systems*, Vol. 201, 2012, pp. 86-104. doi:10.1016/j.fss.2011.12.016
- [14] C. Pozna, R.-E. Precup, "Applications of signatures to expert systems modelling," *Acta Polytechnica Hungarica*, Vol. 11, No. 2, 2014, pp. 21 - 39
- [15] R. Zall, M. R. Kangavari, "On the construction of multi-relational classifier based on canonical correlation analysis," *International Journal of Artificial Intelligence*, 17 (2), 2019, pp. 23-43
- [16] E. L. Hedrea, R.-E. Precup, R. C. Roman, E. M. Petriu, "Tensor product-based model transformation approach to tower crane systems modeling," *Asian Journal of Control, Special Issue*, March 2021. doi:10.1002/asjc.2494
- [17] P. Bransett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequence," *Image and Vision Computing*, 2007, vol. 25, no. 8, pp. 1217-1227. doi:10.1016/j.imavis.2006.07.017
- [18] P. G. Bhat, B. N. Subudhi, T. Veerakumar, V. Laxmi, and M. S. Gaur, "Multi-feature fusion in particle filter framework for visual tracking," *IEEE Sensors Journal*, 2020, vol. 20, no. 5, pp. 2405-2415. doi:10.1109/jsen.2019.2954331
- [19] S. M. Muddamsetty, D. Sidibé, A. Trémeau, and F. Mériaudeau, "Salient objects detection in dynamic scenes using color and texture features," *Multimedia Tools and Applications* 2017, pp. 1-14. doi:10.1007/s11042-017-4462-y
- [20] A. D. Worrall, R. F. Marslin, G. D. Sullivan, and K. D. Baker, "Model-based tracking," Springer-Verlag, London, 1991, pp. 310-318. doi:10.1007/978-1-4471-1921-0\_39
- [21] P. Paizakis, "Region-based object detection and tracking," Doctoral dissertation, Technical University of Crete, June 2009
- [22] A. Almeida, J. Almeida, and R. Araujo, "Real-time tracking of moving objects using particle filter," *IEEE International Symposium on Industrial Electronics*, 2005, pp. 1327-1332. doi:10.1109/ISIE.2005.1529124
- [23] A. Nakhmani, and A. Tannenbaum, "Particle filtering with region-based matching for tracking of partially occluded and scaled target," *SIAM Journal on Imaging Sciences*, 2011, vol. 4 no. 1, pp. 220-242. doi:10.1137/090779280
- [24] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "Object tracking with an adaptive color-based particle filter," *Lecture Notes in Computer Science, Pattern Recognition*, 2002, vol. 2449, pp. 353-360. doi:10.1007/3-540-45783-6\_43
- [25] G. Chenguang, L. Xianglong, Z. Linfeng, and L. Xiang, "A fast and accurate corner detector based on harris algorithm," *IEEE 3rd International Symposium on Intelligent Information Technology Application*, 2009, pp. 49-52. doi:10.1109/IITA.2009.311
- [26] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding*, 2009, vol. 113 no. 3, pp. 345-352. doi:10.1016/j.cviu.2008.08.006
- [27] S. Hu, N. Wu, and H. Song, "Object tracking method based on SURF," *AASRI Conference on Modeling, Identification, and Control*, 2012, vol. 3, pp. 351-356. doi:10.1016/j.aasri.2012.11.055
- [28] C. Q. Lai, and S. S. Teoh, "An efficient method of HOG feature extraction using selective histogram Bin and PCA feature reduction," *Advances in Electrical and Computer Engineering (AECE)*, 2016, vol. 13, pp. 101-108. doi:10.4316/AECE.2016.04016
- [29] H. T. Nguyen, and A. W. M. Smeulders, "Robust tracking using foreground-background texture discriminati," *International Journal of Computer Vision*, 2006, vol. 69, no. 3, pp. 277-293. doi:10.1007/s11263-006-7067-x
- [30] D. Koller, J. Weber, and J. Malik, "Towards realtime visual based tracking in cluttered traffic scenes," *IEEE Intelligent Vehicles Symposium*, 1994, pp. 201-206. doi:10.1109/ivs.1994.639503
- [31] S. H. Lee, "Real-time camera tracking using a particle filter combined with unscented Kalman filters," *Journal of Electronic Imaging*, 2014, vol. 23, no. 1, pp. 013029. doi:10.1117/1.JEI.23.1.013029
- [32] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process*, 2002, vol. 50, no. 2, pp. 174-188. doi:10.1109/78.978374
- [33] J. Scharcanski, A. Oliveira, P. G. Cavalcanti, and Y. Yari, "A particle filtering approach for vehicular tracking adaptive to occlusions," *IEEE Transactions on Vehicular Technology*, 2001, vol. 60, no. 2, pp. 381-389. doi:10.1109/TVT.2010.2099676
- [34] B. Sharma, and N. Kaur, "Various methods for object tracking-A review," *Journal of Computer Engineering (IOSR-JCE)*, 2016, vol. 18, no. 3, pp. 103-105
- [35] P. Barcellos, V. Gomes, and J. Scharcanski, "Shadow detection in camera-based vehicle detection: survey and analysis," *Journal of Electronic Imaging*, 2016, vol. 25, no. 5, pp. 051205, doi:10.1117/1.JEI.25.5.051205
- [36] R. Canals, A. Ganoun, and R. Leconge, "Occlusion-handling for improved particle filtering-based tracking," *17th European Signal Processing Conference (EUSIPCO)*, 2009, pp. 1107-1111
- [37] C. Reta, L. Altamirano, J. A. Gonzalez, and R. Medina-Carnicer, "Three hypothesis algorithm with occlusion reasoning for multiple people tracking," *Journal of Electronic Imaging*, 2015, vol. 24, no. 1, pp. 013015. doi:10.1117/1.JEI.24.1.013015
- [38] H. P. Liu, and F. C. Sun, "Fusion tracking in color and infrared images using joint sparse representation," *Science China Information Sciences*, 2012, vol. 55, no. 3, pp. 590-599. doi:10.1007/s11432-011-4536-9
- [39] P. Kumar, M. J. Brooks, and A. Dick, "Adaptive multiple object tracking using colour and segmentation cues," *Asian Conference on Computer Vision (ACCV)*, 2007, pp. 853-863. doi:10.1007/978-3-540-76386-4\_81
- [40] J. C. Chen, and Y. H. Lin, "Accurate object tracking system by integrating texture and depth cues," *Journal of Electronic Imaging*, 2016, vol. 25, no. 2, pp. 023003, doi:10.1117/1.JEI.25.2.023003

- [41] W. L. Khong, W. Y. Kow, Y. K. Chin, I. Saad, and K. T. K. Teo, "Overlapping vehicle tracking via adaptive particle filter with multiple cues," IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2011, pp. 25-27. doi:10.1109/iccsce.2011.6190570
- [42] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "Object tracking based on color information employing particle filter algorithm," In D. H. Goszczynska, Object Tracking, 2011, pp. 69-88. doi:10.5772/15277
- [43] M. Jaward, L. Mihaylova, N. Canagarajah, and D. Bull, "Multiple object tracking using particle filters," IEEE Aerospace Conference, 2006, pp.1-8. doi:10.1109/AERO.2006.1655926
- [44] X. Lu, T. Izumi, L. Teng, and L. Wang, "Particle filter vehicle tracking based on SURF feature matching," IEEE Journal of Industry Application, 2014, vol. 3 no. 2, pp. 182-191. doi:10.1541/ieejia.3.182
- [45] H. B. Kekre, and K. Sonawane, "Performance evaluation of bins approach in YCbCr color space with and without scaling," International Journal of Soft Computing and Engineering (IJSCE), 2013, vol. 3 no. 3, pp. 203-210, ISSN: 2231-2307
- [46] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "A moving object tracking based on color information employing a particle filter algorithm," The 14th International Symposium on Artificial Life and Robotics, 2009, pp. 39-42. doi:10.1007/s10015-009-0718-6
- [47] Y. Dai, and B. Liu, "Robust video object tracking using particle filter with likelihood based feature fusion and adaptive template updating," In Computer Science - Computer Vision and Pattern Recognition (cs.CV), 2015. doi:arxiv.org/abs/1509.08182
- [48] B. Deori, and D. M. Thounaojam, "A survey on moving object tracking in video," International Journal on Information Theory (IJIT), 2014, vol. 3 no. 3, pp. 31-46. doi:10.5121/ijit.2014.3304.
- [49] R. C. Gonzalez, and R. E. Wood, Digital image processing, (3rd ed.). NJ., USA: Tom Robbins, 2008, pp. 665-669
- [50] F. Bashar, A. Khan, F. Ahmed, and H. Kabir, "Face recognition using similarity pattern of image directional edge response," Advances in Electrical and Computer Engineering (AECE), 2014, vol. 14, pp. 69-76. doi:10.4316/AECE.2014.01011
- [51] J. D. Hol, "Resampling in particle filters," Internal report, Institutionen för systemteknik, Linköping University, Sweden, 2004. <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-2366>