Directional Particle Filter Using Online Threshold Adaptation for Vehicle Tracking

Mustafa Eren Yıldırım[1], Yücel Batu Salman[2]
[1] Department of Electrical and Electronics Engineering, Bahcesehir University
34353, Istanbul, Turkey
[e-mail: mustafaeren.yildirim@eng.bau.edu.tr]
[2] Department of Software Engineering, Bahcesehir University
34353, Istanbul, Turkey
[e-mail: batu.salman@eng.bau.edu.tr]
*Corresponding author: Mustafa Eren Yıldırım

Received March 19, 2017; revised June 21, 2017; accepted September 15, 2017; published February 28, 2018

Abstract

This paper presents an extended particle filter to increase the accuracy and decrease the computation load of vehicle tracking. Particle filter has been the subject of extensive interest in video-based tracking which is capable of solving nonlinear and non-Gaussian problems. However, there still exist problems such as preventing unnecessary particle consumption, reducing the computational burden, and increasing the accuracy. We aim to increase the accuracy without an increase in computation load. In proposed method, we calculate the direction angle of the target vehicle. The angular difference between the direction of the target vehicle and each particle of the particle filter is observed. Particles are filtered and weighted, based on their angular difference. Particles with angular difference greater than a threshold is eliminated and the remaining are stored with greater weights in order to increase their probability for state estimation. Threshold value is very critical for performance. Thus, instead of having a constant threshold value, proposed algorithm updates it online. The first advantage of our algorithm is that it prevents the system from failures caused by insufficient amount of particles. Second advantage is to reduce the risk of using unnecessary number of particles in tracking which causes computation load. Proposed algorithm is compared against camshift, direction-based particle filter and condensation algorithms. Results show that the proposed algorithm outperforms the other methods in terms of accuracy, tracking duration and particle consumption.

Keywords: Automatic thresholding, adaptive weighting, computer vision systems, direction-based particle filter, particle reduction, surveillance system, vehicle tracking, video object tracking

A preliminary version of this paper appeared in International Conference on Advances in Computing, Electronics and Communication - ACEC 2016, December 15-16, Rome, Italy. This version includes a concrete analysis and supporting implementation results on more videos and algorithms.

https://doi.org/10.3837/tiis.2018.02.010 **ISSN : 1976-7277**

1. Introduction

Intelligent transportation systems (ITS) attract interest of researchers more and more every day. Under the title of ITS, there are various types of technologies and applications; such as visual vehicle tracking [1], vehicular sensor networks [2], routing [3], vehicle-to-vehicle (V2V) communication [4], in-car entertainment [5]. Physical aspects of the medium, hardware quality, speed of the vehicle may have direct effect on these services [6, 7].

Among these technologies, tracking has become a primary research area of surveillance systems with the recent innovations in object detection techniques, high definition cameras, signal processing and networks [8]. In addition to its role in ITS, it has appeared also various applications such as human-computer interfaces [9-11]; surveillance systems [8, 12, 13], radar detection [14, 15]; behavioral analysis [16], image processing [17] and robotics [18-20].

The primary objective of tracking is to estimate the location of a target object in real world coordinates. Robustness and accuracy of vehicle detection have a great significance in tracking and higher level processing. Although several vehicle tracking algorithms have been proposed [11, 21-23], it still remains as a burdensome problem due to illumination, collision, background clutters, deformation and speed of the vehicle in the recorded video [24]. Shape, size, pose, and color restricts of the vehicle restrict the recognition and tracking performance [25].

Particle filter, also known as sequential Monte Carlo is a sampling-based object tracking method. It is an acknowledged algorithm in filtering and predicting non-linear and non-Gaussian state space models in a wide range of application areas [26-28]. It is used to generate an approximation of posterior probability density function for the state variable by deriving weighted particles with empirical distributions. Generation of new particles, computation of particle weights and resampling are essential operations of particle filter [29]. However, the main drawback of particle filter algorithm is its particle degeneracy and sample impoverishment [30]. In order to overcome these drawbacks, large number of particles is required. With no doubt, large number of particles is going to increase the computational complexity. Increased computation complexity does not affect the tracking accuracy, but it can extend the system capacity. Therefore, a tracking algorithm, which can operate rapidly and accurately with minimum computation complexity, is required.

In literature, there exist several approaches integrating and modifying particle filter algorithm for vehicle tracking. In [31], authors presented a modified particle filter, which is able to assemble to the true state of an object as close as possible by sampling the particles iteratively. Although the accuracy of tracking is increased, the computational load remained high. Adaptive region-wise linear subspace expression with an iterative particle filter was presented in [32]. The approach could not perform accurately under full occlusion. In [33], a layered particle filter architecture is introduced which is embedding continuous adaptive mean shift, along with the combination of probabilistic and deterministic approaches in order to overcome well-known obstacles in multiple tracking processes. The accuracy is reduced under occlusion, although, better posterior probability density representation was displayed. In [11], authors proposed a top-down visual attention computation model, which depends on frequency analysis with particle filter integration. They aimed to solve major tracking problems such as longtime occlusion, and abrupt motion. A multi-feature target representation with particle filter was presented which is able to increase the accuracy but could not reduce the computational burden [17]. Authors proposed a mean shift embedded particle filter algorithm in order to improve the sampling efficiency but not integrated in vehicle tracking operations [34]. Additionally, some other modified particle filters were recommended which could only affect the tracking speed, but not the accuracy [35, 36]. An improved particle filter based on firefly algorithm was established regarding with the problem of sample impoverishment and has outperformed the condensation particle filter method in robustness [37]. In addition, a color histogram based tracking algorithm was established against noise and occlusion but could not succeed due to illumination changes [38]. Particle filter combining color histogram and edge-based shape features under a sequential Monte Carlo framework could not perform accurately in difficult sequences with color changes under occlusion [39].

In [40], authors proposed a modified particle filter for airborne vehicle tracking by focusing on particle sampling and observation model. They achieved to increase the robustness but their tracker still fails in case of abrupt changes in appearance. In [41], authors used particle filter for fusing measurements from multi sensors for vehicle tracking and obtained better results. Authors in [42] introduced a modified particle filter for multi-vehicle tracking which can automatically initialise occluded vehicles. They improved the accuracy was computation load remained as a problem. In [43], authors presented a modified particle filter in which the importance density function is updated according to the residual in each stage. In [44], authors made a new approach for multiple target tracking with particle filter. They made observations on targets to make decision to conduct resampling or not. They improved the tracking accuracy but did not consider computation load.

As the initial version of our novel algorithm, we implemented an extended particle filter algorithm that weights and distributes particles depending on their angular difference to the target vehicle's motion. Feature weighting is used in different areas other than tracking also [45]. In [46], experiments proved that this method reduced particle consumption while increasing the tracking accuracy and operating robust against even in noisy and occluded mediums. Since both the tracking accuracy and amount of particles highly rely on a predefined and constant threshold, the decision of threshold gains importance. In each frame, the amount of particles required for tracking does not have to be same. For a highway surveillance camera, view might contain both straight and bendy roads. A large amount of particles is required when the vehicle is changing its direction. However, in case of a translational motion, small amount of particles is enough. In this sense, threshold must be adaptive during execution.

In this paper, we present a modified particle filter algorithm that distributes the particles according to an online automatic threshold scheme. As the first step of the proposed scheme, the direction of the vehicle is calculated. Next, we weight particles according to their angular similarities with target vehicle. The ones having the same or similar motion direction with the target increases their probability of likelihood whereas the others are eliminated. The number of particles used has become decisive in valuing the online threshold. Usage of excessive amount of particle is prevented in this model. We tested our algorithm on three videos, which contain different physical and visual aspects. According to results, proposed model outperformed camshift, direction-based particle filter and condensation algorithms in terms of tracking duration and precision.

The rest of the paper is organized as follows; section 2 explains background and core concept of particle filter, section 3 describes the direction-based particle filter. Section 4 and 5, give proposed algorithm and experimental results, respectively. At last, conclusion is given in section 6.

2. Particle Filter

2.1 Recursive Bayesian Inference

The purpose of bayesian inference is to provide a mathematical machinery that can be used to model systems, where uncertainties of the system are taken into account and decisions are made according to a rational principle [10-26]. A dynamic model represents the change in state parameters with time.With state sequence $\{x_t, t \in \mathbb{N}\}$ of a target, dynamic model is given as;

$$
\begin{aligned}
x_t &= f_t(x_{t-1}, u_t) \\
y_t &= g_t(x_t, v_t)
\end{aligned} \tag{1}
$$

where f_t and g_t are state transition and measurement model functions and y_t is the measurement model. u_t and v_t are white noises which are more usually called the process noise and measurement error. These two noise sequences consist of mutually independent random variables of previously known distributions. First expression in (1) is a Markov process and can be written as $p(x_t|x_{t-1})$. The second expression of (1), can be written as $p(y_t|x_t)$. The rule for propagation of state density over time is given in (2).

$$p(x_t|y_t) = \alpha\, p(y_t|x_t)\, p(x_t|y_{t-1}) \tag{2}$$

where α is a normalisation constant that does not depend on x_t. In (2), bayesian inference estimates the posterior density depending on prior density and measurement in that time step.

2.2 Particle Filter

At this point, particle filter offers to make this recursive process by using particles. The key idea is to represent the required posterior density function given in (2) by a set of random particles with associated weights and to compute estimates based on these particles [4-20]. Each sample $\boldsymbol{z_t^i}$ in the set $\{\boldsymbol{z_t^i}, \boldsymbol{w_t^i}; \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{N}\}$ represents a hypothetical state of the object being tracked and has a corresponding discrete sampling probability w, where $\sum_i \boldsymbol{w_t^i} = \mathbf{1}$, at time t and $N \in \mathbb{N}$ is number of particles. The evolution of the sample set is described by propagating each sample according to a system model. Next step is to weight each particle with its corresponding observation and draw a new set of N particles with probability$\boldsymbol{w_t^i} = \boldsymbol{p}(\boldsymbol{y_t}|\boldsymbol{x_t} = \boldsymbol{z_t^i})$. In the final stage, the mean state of the tracked object is estimated at each time step as in (3).

$$E[z_t] = \sum_{i=1}^{N} z_t^i\, w_t^i \tag{3}$$

Target vehicle and particles can be represented by various types of geometric models such as rectangle, circle or ellipse. In this study, ellipse is used. Elliptic dynamic state model has four dimensions as $\{x, y, H_x, H_y\}$, where x and y refer to the coordinates of the center location of the ellipse, and H_x and H_y refer to the lengths of the major and minor axes of the ellipse, respectively [46].

3. Direction-based Particle Filter

In case of occlusion of the target by another vehicle or pole, it can be difficult to obtain new observations. Therefore, in such cases, the likelihood function cannot be utilized and tracking might fail. In [46], we have presented a novel approach to vehicle tracking problem with a modified particle filter. The main idea was to use the direction of the target motion as a dynamic feature and change the particle distribution in a more efficient way for tracking. This dynamic feature supports the tracker when scalar features cannot be extracted.

At first, we calculate the direction of the target. This direction can be denoted in terms of a displacement vector, by using the center point of the target vehicle in the current frame (x_t, y_t) and center point from the previous frame(x_{t-1}, y_{t-1}), as is shown in (4).

$$\theta_{t-1} = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) = \tan^{-1}\left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}\right) \tag{4}$$

In real world conditions, vehicles do not always move linearly. A vehicle's relative direction to the camera may change when it changes its lane or goes into bendy road. Thus, we update θ periodically by using approximate median method [47], not to lose dynamic information, as follows;

$$H_{t+1} = \begin{pmatrix} H_t + 1 & if & M_t > H_t \\ H_t - 1 & if & M_t < H_t \\ H_t & if & M_t = H_t \end{pmatrix} \tag{5}$$

where H represents a parameter to be updated at time steps t and $t+1$, respectively. The new measurement at each time step is denoted by M.

In conventional particle filter, particles are evenly distributed around the target. Unlike a conventional filter, direction-based filter does not use the entire particle set. As second step, the angle between the target vehicle and each particle is calculated. The aim is to obtain the angular distance of each particle to the direction of the vehicle. Angular distance is used to weight the particles and update their distribution around the vehicle. The calculation for angular distance is performed by using the vehicle's estimated location in the previous time step and particle locations in the current time step as shown in (6).

$$\varphi_{t-1}^{i} = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) = \tan^{-1}\left(\frac{y_{t-1} - y_t^i}{x_{t-1} - x_t^i}\right), \forall\ i \leq N \tag{6}$$

After this calculation, the difference between the direction of vehicle and each value of φ_{t-1}^{i} is obtained by subtracting the results of (6) and (4).

$$\sigma_{t-1}^{i} = \left|\theta_{t-1} - \varphi_{t-1}^{i}\right|, \forall\ i \leq N \tag{7}$$

In (7), σ_{t-1}^{i} refers to the angular distance of the i^{th} particle with respect to direction angle of the vehicle. The minimum and maximum values that angular distance can take are 0° and 180°, respectively.

In next step, we filter the particles by applying a threshold control to the results of (7). Filtering process is as given in (8).

$$w_t^i = \begin{pmatrix} 0 & \text{if} \quad \sigma_{t-1}^i \geq T \\ w_t^i \times \left(1 - \left[\frac{\sigma_{t-1}^i}{T}\right]\right) & else \end{pmatrix} \tag{8}$$

In (8), T is a predefined angle threshold measured in degrees. According to (8), if the angular distance of a particle to the direction of the target is more than distance T, it will be multiplied by 0 and will not be used. Otherwise, the likelihood probability of that particle will be re-weighted. This weight is inversely proportional to the distance and calculated by (8). The above expression reveals that, as the angle difference between target vehicle and particle decreases, the weight of that particle increases. If the difference is equal to "0", then the weight will be equal to "1". After the decision on each particle is made, re-weighted values are normalized. In this way, the likelihood probabilities of those particles that are moving in the same or similar direction to the target are increased.

Fig. 1 shows the change in the particle probability distribution function for T = π/8. In conventional particle filter algorithm, probability distribution of particles is independent from angle information as shown with the black dotted line on **Fig. 1**. Red dotted line shows the probability distribution function after applying direction-based weighting to particles. Since some particles are eliminated, remained particles will share the total probability. This leads a significant increase in their probabilities. The initial probabilities in conventional algorithm were around approximately 0.004 to 0.007 for 100 particles. In direction-based filter, new probability values of the selected particles increased up to 0.045.

In **Fig. 1**, the x-axis range is [0, 180°] because the result of (7) is always non-negative. According to the algorithm, the maximum difference can be 180°, when the target and particle are moving in opposite directions. The red dotted line starts from 45°, which is double of given threshold T= π/8. The reason is that, we check the distances on both left and right sides of the motion angle.

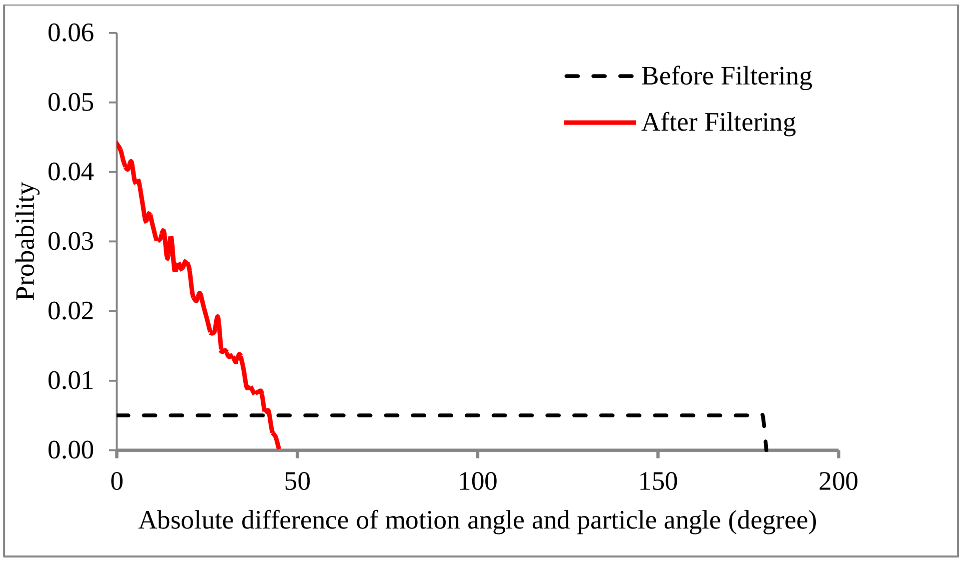


Fig. 1. Probability density function of particles before and after filtering

4. Proposed Approach

As explained in Section 3, the value of T directly affects the performance of tracking. Thus, an inaccurately assigned threshold is most likely considered as a system drawback. Several recent studies have proven that tracking accuracy of particle filter highly depends on number of particles consumed [48]. If T has a small value such as $\pi/8$, particle consumption is minimized but the accuracy is relatively lower. Therefore, it is not unexpected that a failure can occur. In contrast, when T has a large value like $\pi/2$, highest accuracy is obtained with highest computation load.

In this content, our study introduces a new approach, which reduces the tracking failures while avoiding the use of excessive number of particles. We developed a novel thresholding system that updates itself automatically during tracking. By adapting itself to the medium, tracker will be able to continue tracking and keep the particle consumption under control. In this study, the initial value of N is 100.

Table 1. Conditions of threshold adaptation

Number of Particles (N)	Threshold (T)
[25, 40] || [80, 100]	$\pi/8$
(0, 25) || [70, 80)	$\pi/2$
(40, 70)	$\pi/4$

Table 1 shows the updating conditions of the proposed algorithm. At this phase of the research, we tested the condensation algorithm on different videos with various visual and physical aspects in order to decide the particle range. In each video, condensation algorithm was tested five times for each different value of N {25, 40, 60, 80, 100}.

Test results showed that the risk of failure is relatively higher when N is less than and equal to 25. Therefore, algorithm is set to increase the T to $\pi/2$. Thus, the amount of particles left inside the threshold region is raised. In addition, tracker performs well without any risk of failure when the number of particles is between 25 and 40. Thus, T is assigned kept as $\pi/8$.

According to the given conditions, when the number of particles is excessive, T will be decreased to $\pi/8$. In this way, we prevent unnecessary computation load. Computation load was significantly lower with 70 to 80 particles, compared to the load with 80 to 100 particles, while achieving similar accuracy.

In the remaining particle range, T is assigned as $\pi/4$. **Fig. 2** shows an illustration of distribution of T values on particle spectrum.

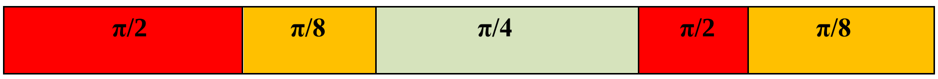


Fig. 2. Threshold assignment of particles

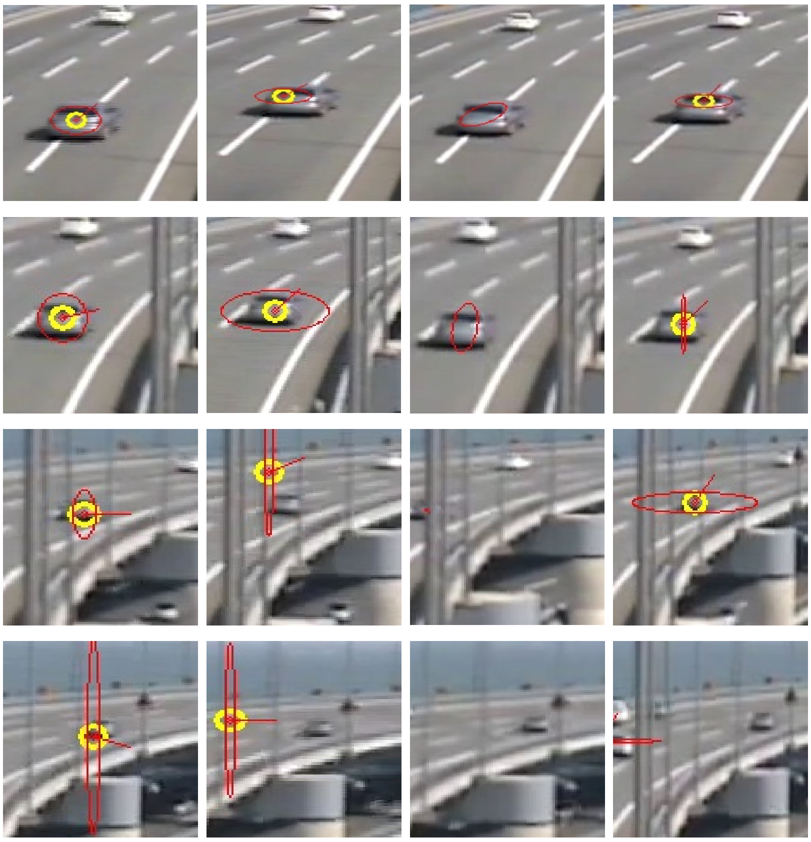
This update process is repeated periodically in every three frames. Updating frequency is a heuristically obtained value. Updating it very frequently causes computation load. Whereas, rare update will cause information loss and decrease the effect of the algorithm on tracking.

5. Experimental Results

5.1 Qualitative Results

Experiments were conducted on three different videos, which were recorded by standard quality cameras. For benchmarking study, proposed algorithm was compared with condensation, direction-based particle filter and camshift algorithms. First video is taken from a camera located on a sideway pole of a bridge in Busan, Rep. of Korea. It differs from the others in terms of the stability of the camera. The camera vibrates at certain times of the video. The other two videos are from a public dataset [49]. In each video, horizontal and vertical distances between camera and road are different. All videos contain view angle difference, illumination and scale variations, occlusions, motion blur, in-plane rotation and background clutter.

In our experiments, color histograms for measuring the similarity are calculated in hue saturation value (HSV) space. In order to make the proposed method less sensitive to illumination conditions, we used a HSV color space of 8 × 8 × 4 bins (where value (V) is represented by four bins). This section contains captured images showing four trackers for the test videos. All the experiments were conducted in an Intel (R)-Core (TM) i5-4570 CPU at 3.20 GHz and 8 GB of RAM computer.



(a) (b) (c) (d)

Fig. 3. Tracking captures of (a) proposed particle filter, (b) condensation particle filter, (c) camshift and (d) direction-based particle filter with $T = \pi/8$ in Video_1

Fig. 3 shows the image stills from the first video. The quality of the video is poor. In addition to the listed visual difficulties, another handicap in this video is that the color of the target vehicle is very similar with the road. As expected, our algorithm keeps tracking for the longest duration. Direction-based tracker with T=π/8 outperforms condensation and camshift algorithms. However, when the target starts to change its direction and is occluded by poles, tracker requires more particles for tracking. At this point, direction-based tracker fails due to insufficient number of particles but the proposed algorithm increases the T value and removes this failure risk. In **Fig. 3**, **Fig. 4** and **Fig. 5**, frames in the first row show the first location of tracking and the frames in the bottom row show the end of tracking.

(a) (b) (c) (d)

Fig. 4. Tracking captures of (a) proposed particle filter, (b) condensation particle filter, (c) camshift and (d) direction-based particle filter with T= π/8 in Video_2

Fig. 4 shows the captured frames of the second video [49]. Vertical distance between the road and camera is very short. In this video, target is blue car and it is not occluded but its size and rotation changes by time. Proposed algorithm continues tracking until the car looks like a small ball where as the other trackers fail at some points.

Fig. 5 presents captured frames of the third video. Unlike the first and second videos, in this video vertical distance between the road and camera is very high.

There is a strong correlation between the results of all videos. In all cases, proposed algorithm gives the highest performance in terms of tracking duration and it is followed by direction-based tracker. Direction-based weighting of particles results in a higher tracking accuracy and duration compared to two other trackers. However, in cases of not having enough number of particles for tracking, it may lose the target. As we can see in above examples, the proposed algorithm eliminates this problem.

(a) (b) (c) (d)

Fig. 5. Tracking captures of (a) proposed particle filter, (b) condensation particle filter, (c) camshift and (d) direction-based particle filter with $T = \pi/8$ in Video_3

5.2 Quantitative Results

This section presents the benchmarking analysis of quantitative results. Comparisons were conducted in terms of tracking precision and particle consumption. For comparison of particle consumption, camshift is not included since it does not use particles. In case of tracking precision, Euclidean distance between center coordinates of the estimated and ground truth locations gives the center location error.

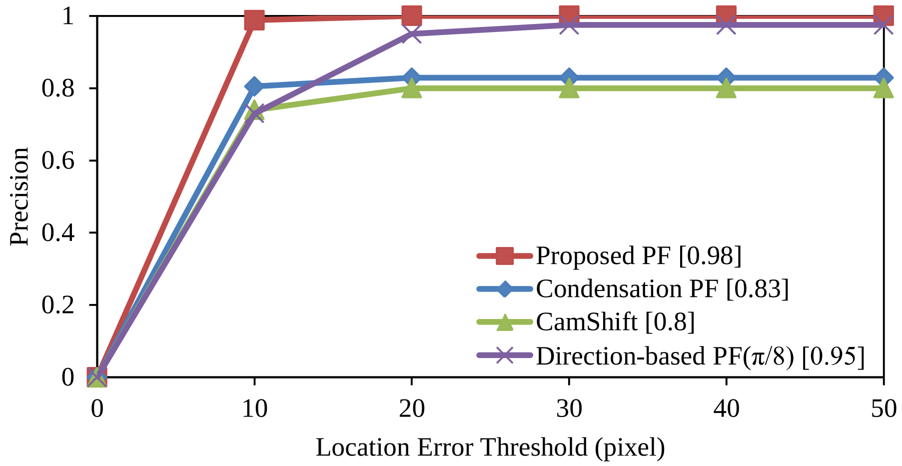


Fig. 6. Precision plot for video_1

A drawback of this measure occurs when a tracker loses the target and moves to a random location on the frame. This will cause an abrupt increase in error. Thus, this will not reflect the real performance of the tracker [50]. That is why, we use precision plot, which shows the percentage of frames in which the center location error is less than a threshold. The values near each tracker on the plots are their precision scores. A precision score of a tracker presents its score at a threshold of 20 pixels [50].

Fig. 6 shows the precision plot of first video with one-pass evaluation (OPE). It is used for precision [51]. It is a common way to evaluate the performance of a tracker throughout a sequence by initializing the target in the first frame it appears.

In addition to highest performance in tracking duration, the proposed tracker also outperforms the compared algorithms in terms of accuracy. For the proposed tracker, Euclidean distance between the centers is less than 20 pixels for 98% of all frames, whereas it is 83%, 80% and 95% for condensation, camshift, and direction-based particle filter respectively.

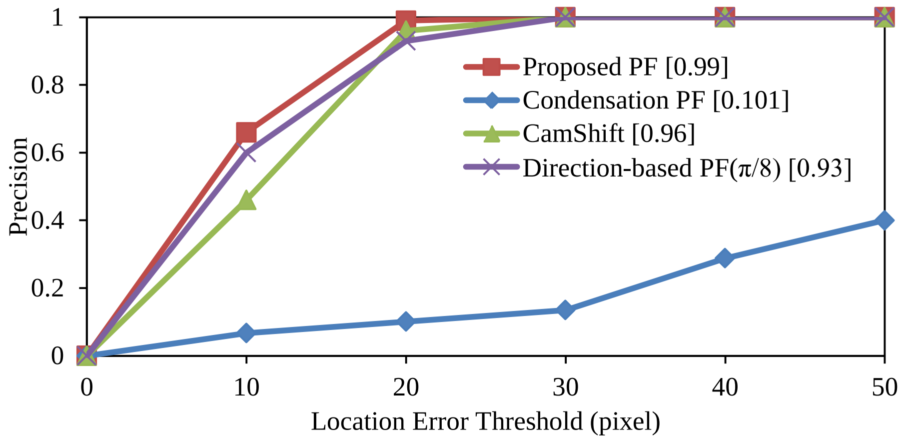


Fig. 7. Precision plot for video_2

Fig. 7 and **Fig. 8** show the precision plots for the second and third videos respectively. Proposed tracker has the highest precision score for all videos. As shown in both figures, proposed tracker is the more robust compared to others. In **Fig. 7**, condensation particle filter has the lowest accuracy compared to other trackers.

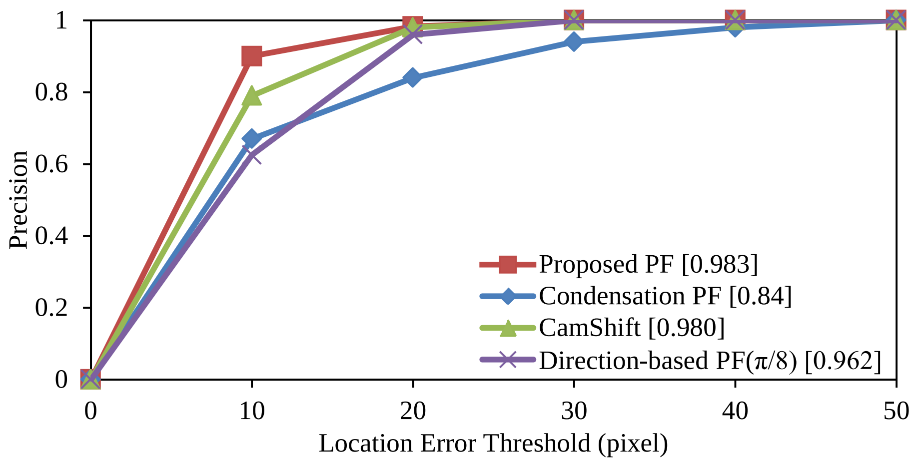


Fig. 8. Precision plot for video_3

Center location errors of all algorithms in all videos are shown in **Table 2**. The values are calculated for the time duration that the target exists on the scene. In all videos, the proposed tracker continues tracking, whereas other trackers already fail. Although the other trackers show sufficient performance for a duration, their failure results in a sudden increase in average center location.

Table 2. Average for Center Location Error (pixel)

Video	Proposed PF	Condensation PF	Camshift	Direction-based PF
Video_1	4	82	96	36
Video_2	9	200	220	145
Video_3	5	174	153	54

Another contribution of the proposed algorithm is to decrease the particle number consumption while increasing the accuracy. **Table 3** shows the average number of particles used in proposed, condensation and direction-based particle filters. The number of particles used in proposed particle filter is roughly between 50-60% of the condensation filter. On the other hand, it uses more particles than the direction-based particle filter with threshold, $T=\pi/8$. Although, this may seem as a drawback, it actually leads the proposed tracker to decrease the failure risk.

As a result, the proposed method is feasible in mediums in which the direction of the target vehicle can be extracted. In case of two similar vehicles moving together on the frame, usage of direction prevents tracker to jump to the wrong vehicle. With a combination of proper detection algorithm, this method can be used for both online and offline tracking.

Table 3. Average Particle Number

Video	Proposed PF	Condensation PF	Direction-based PF
Video_1	63	100	29
Video_2	52	100	32
Video_3	59	100	27

6. Conclusion

An extended particle filter with adaptive angle thresholding scheme for single vehicle tracking is presented. Particles are filtered and redistributed with respect to an angle threshold. We propose a scheme which updates this angle threshold according to the number of particles that is used in the previous iteration. The system counts the number of particles in every 3 frames and updates the threshold according to a set of predefined conditions. Excessive use of particles and also risk of failure is prevented as well as an increase in accuracy is obtained. Experiments are conducted on three videos with different physical and visual conditions. Comparison results show that the proposed algorithm outperforms condensation, camshift, and direction-based particle filter in terms of tracking duration and accuracy.

However, despite the improvements made in this study, some issues still remain. One of them is the decision of updating frequency. The need of an update may vary among roads depending on their physical characteristics such as curves. Another issue is to modify this algorithm for multi-vehicle tracking. Our further studies will focus on these problems.

References

[1] M. E. Yildirim, J. K. Song, J. S. Park, B. W. Yoon, Y. S. Yu, “Robust Vehicle Tracking with multi-feature particle filter,” in *Proc. of Int. Conf. on Multimedia, Computer Graphics and Broadcasting*, pp. 191-196, December 8-10, 2011. Article (CrossRef Link)

[2] P.G.V. Naranjo, M. Shojafar, H. Mostafaei, “P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks,” *Journal of Supercomputing*, vol. 73, no. 2, pp. 733-755, February, 2017. Article (CrossRef Link)

[3] S. Shamshirband, M. Shojafar, A.R. Hosseinabadi, A. Abraham, “OVRP_ICA: An imperialist-based optimization algorithm for the open vehicle routing problem, ” in *Proc. of Int. Conf. on Hybrid Artificial Intelligence Systems*, pp. 221-233, June 21-23, 2015. Article (CrossRef Link)

[4] M. A. Qureshi, R. M. Noor, S. Shamshirband, S. Parveen, M. Shiraz, A. Gani, “A survey on obstacle modeling patterns in radio propagation models for vehicular ad hoc networks,” *Arabian Journal for Science and Engineering*, vol. 40, no. 5, pp. 1385-1407, February, 2015. Article (CrossRef Link)

[5] R. Kodama, M. Koge, S. Taguchi and H. Kajimoto, “COMS-VR: Mobile virtual reality entertainment system using electric car and head-mounted display,” in *Proc. of IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 130-133, March 18-19, 2017. Article (CrossRef Link)

[6] E. Baccarelli, P. G. Vinueza Naranjo, M. Scarpiniti, M. Shojafar, J. H. Abawajy, "Fog of Everything: Energy-efficient Networked Computing Architectures, Research Challenges, and a Case Study, " *IEEE Access* ,vol. 5, pp. 9882-9910, May, 2017. Article (CrossRef Link)

[7] M. Shojafar, N. Cordeschi, E. Baccarelli, "Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services," *IEEE Transactions on Cloud Computing*, vol. PP, no.99, pp.1-1, April, 2016. Article (CrossRef Link)

[8] Z.L. Szpak and J.R. Tapamo, "Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set," *Expert Systems with Applications*, vol. 38, no. 9, pp. 6669-6680, June, 2011. Article (CrossRef Link)

[9] Y.H. Ma, Z.H. Mao, W.Y. Jia, C.L. Li, J.W. Yang, M.G. Sun, "Magnetic hand tracking for human-computer interface," *IEEE Transactions on Magnetics*, vol. 47, no. 5, pp. 970-973, May, 2011. Article (CrossRef Link)

[10] V.A. Prisacariu and I. Reid, "3D hand tracking for human computer interaction," *Image and Vision Computing*, vol. 30, no. 3, pp. 236-250, March, 2012. Article (CrossRef Link)

[11] W. Li, P. Wang, H. Qiao, "Top-down visual attention integrated particle filter for robust object tracking," *Signal Processing- Image*, vol. 43, pp.28-41, April, 2016. Article (CrossRef Link)

[12] C. M. Huang and L.C. Fu, "Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 41, no. 1, pp. 234-247, February, 2011. Article (CrossRef Link)

[13] I. Elafi, M. Jedra, N. Zahid, "Unsupervised detection and tracking of moving objects for video surveillance applications," *Pattern Recognition Letters*, vol. 84, pp.70-77, December, 2016. Article (CrossRef Link)

[14] D. Y. Kim and H. Jeon, "Data fusion of radar and image measurements for multi-object tracking via Kalman filtering," *Information Sciences*, vol. 278, pp. 641-652, September, 2014. Article (CrossRef Link)

[15] J. Cesic, I. Markovic, I. Cvisic, I. Petrovic, "Radar and stereo vision fusion for multitarget tracking on the special Euclidean group," *Robotics and Autonomous Systems*, vol. 83, pp.338-348, September, 2016. Article (CrossRef Link)

[16] A. Xydes, M. Moline, C. G. Lowe, T. J. Farrugia, C. Clark, "Behavioral characterization and Particle Filter localization to improve temporal resolution and accuracy while tracking acoustically tagged fishes," *Ocean Engineering*, vol. 61, pp. 1-11, March, 2013. Article (CrossRef Link)

[17] J. F. Dou and J. X. Li, "Robust visual tracking based on adaptively multi-feature fusion and particle filter," *Optik*, vol. 125, no. 5, pp.1680-1686, March, 2014. Article (CrossRef Link)

[18] A. Ibarguren, I. Maurtua, M. A. Perez, B. Sierra, "Multiple target tracking based on particle filter for safety in industrial robotic cells," *Robotics and Autonomous Systems*, vol. 72, pp.105-113, October, 2015. Article (CrossRef Link)

[19] K. D. Sharma, A. Chatterjee, A. Rakshit, "A PSO – Lyapunov hybrid stable adaptive fuzzy tracking control approach for vision-based robot navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 7, pp.1908-1914, July, 2012. Article (CrossRef Link)

[20] T. Li, M. Bolic, P. M. Djuric, "Resampling methods for particle filtering: classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70-86, May, 2015. Article (CrossRef Link)

[21] N. C. Mithun, T. Howlader, S.M.M. Rahman, "Video-based tracking of vehicles using multiple time-spatial images," *Expert Systems with Applications*, vol. 62, pp.17-31, November, 2016. Article (CrossRef Link)

[22] I. Majeed and O. Arif, "Non-linear eigenspace visual object tracking," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 363-374, October, 2016. Article (CrossRef Link)

[23] N. Zhao, Y. Xia, C. Xu, X. Shi, Y. Liu, "APPOS: An adaptive partial occlusion segmentation method for multiple vehicles tracking," *Journal of Visual Communication and Image Representation*, vol. 37, pp. 25-31, May, 2016. Article (CrossRef Link)
[24] X. Fei and K.Hashimoto, "An object-tracking algorithm based on particle filtering with region-based level set method," in *Proc. of Int. Conf. on Intelligent Robots and Syst*ems, pp.2908-2913, October 18-22, 2010. Article (CrossRef Link)
[25] B. Tian, Y. Li, B. Li, D. Wen, "Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 597-606, April, 2014. Article (CrossRef Link)
[26] N. J. Gordon, D. J. Salmond, A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," *Radar and Signal Processing*, vol.140, no.2, pp.107-113, May 1993. Article (CrossRef Link)
[27] C. L. Dong and Y. N. Dong, "Survey on video based vehicle detection and tracking algorithms," Journal of Nanjing University of Posts and Telecommunications: Natural Sciences, vol. 29, no. 2, pp. 88-94, April, 2009. Article (CrossRef Link)
[28] G. S. Walia and R. Kapoor, "Human detection in video and images a-state-of-art survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol.28, no. 3, pp.1 -25, May, 2014. Article (CrossRef Link)
[29] S. J. Hong, M. Bolic, P. M. Djuric, "An efficient fixed-point implementation of residual resampling scheme for high-speed particle filters," *IEEE Signal Processing Letters*, vol. 11, pp. 482-485, June, 2004. Article (CrossRef Link)
[30] H. Zhou, Z. Deng, Y. Xia, M. Fu, "A new sampling method in particle filter based on Pearson correlation coefficient," *Neurocomputin*g, vol. 216, pp. 205-215, December, 2016. Article (CrossRef Link)
[31] Z. Fan, H. Ji, Y. Zhang, "Iterative particle filter for visual tracking," *Signal Processing- Image*, vol. 36, pp.140-153, August, 2015. Article (CrossRef Link)
[32] M. C. Ho, C. C. Chiang, Y. Y. Su, "Object tracking by exploiting adaptive region-wise linear subspace representations and adaptive templates in an iterative particle filter," *Pattern Recognition Letters*, vol. 33, pp.500-512, April 2012. Article (CrossRef Link)
[33] W. Qi, X. Zhang, L. Chao, O. Yuanxin, S. Hao, "A robust approach for multiple vehicles tracking using layered particle filter," *International Journal of Electronics and Communications*, vol. 65, pp. 609-618, July, 2011. Article (CrossRef Link)
[34] C. Shan, T. Tan, Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, pp.1958-1970, July, 2007. Article (CrossRef Link)
[35] H. Liu and F. Sun, "Efficient visual tracking using particle filter with incremental likelihood calculations," *Information Sciences*, vol.195, pp.141-153, July, 2012. Article (CrossRef Link)
[36] H. A. A. El-Halym, I. I. Mahmoud, S. E. D. Habib, "Proposed hardware architectures of particle for object tracking," *EURASIP Journal on Advances in Signal Processing*, vol.1, pp.1 -19, January, 2012. Article (CrossRef Link)
[37] M. L. Gao, L. L. Li, X. M. Sun, L. J. Yin, H. T. Li, D. S. Luo, "Firefly algorithm (FA) based particle filter method for visual tracking," *Optik*, vol. 126, pp. 1705-1711, September, 2015. Article (CrossRef Link)
[38] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564-567, May, 2003. Article (CrossRef Link)
[39] T. Xiong and C. Debrunner, "Stochastic car tracking with line and color based features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp.324-328, January, 2005. Article (CrossRef Link)
[40] I. Szottka and M. Butenuth, "Advanced Particle Filtering for Airborne Vehicle Tracking in Urban Areas, " *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 3, pp. 686-690, March, 2014. Article (CrossRef Link)

[41] R. Hostettler and P. M. Djurić, "Vehicle Tracking Based on Fusion of Magnetometer and Accelerometer Sensor Measurements With Particle Filtering," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 4917-4928, November, 2015. Article (CrossRef Link)
[42] P. Liu, W. Li, Y. Wang and H. Ni, "On-road multi-vehicle tracking algorithm based on an improved particle filter, " *IET Intelligent Transport Systems*, vol. 9, no. 4, pp. 429-441, May, 2015. Article (CrossRef Link)
[43] J. Q. Li, R. H. Zhao, J. L. Chen, C. Y. Zhao, Y. P. Zhu, "Target tracking algorithm based on adaptive strong tracking particle filter, " *IET Science, Measurement & Technology*, vol. 10, no. 7, pp. 704-710, October, 2016. Article (CrossRef Link)
[44] L. Ubeda-Medina, Á. García-Fernández, J. Grajal, "Adaptive auxiliary particle filter for track-before-detect with multiple targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol.PP, no.99, pp.1-1, April, 2017. Article (CrossRef Link)
[45] R. Tavoli, E. Kozegar, M. Shojafar, H. Soleimani and Z. Pooranian, "Weighted PCA for improving Document Image Retrieval System based on keyword spotting accuracy, " in *Proc. of 36th Int. Conf. on Telecommunications and Signal Processing*, pp. 773-777, July 2-4, 2013. Article (CrossRef Link)
[46] M. E. Yildirim, F. I. Ince, Y. B. Salman, J. K. Song, J. S. Park, B. W. Yoon, "Direction based modified particle filter for vehicle tracking," *ETRI Journal*, vol. 38, no. 2, pp. 356 – 364, April, 2016. Article (CrossRef Link)
[47] S.K. Zhou, R. Chellappa, and B. Moghaddam, "Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters," *IEEE Transactions on Image Process*ing, vol. 13, no. 11, pp. 1491–1506, November, 2004. Article (CrossRef Link)
[48] M. Bolic, S.J. Hong and P. M. Djuric, "Performance and complexity analysis of adaptive particle filtering for tracking applications, " in *Proc. of the 36th Asilomar Conf. on Signals, Systems and Computers*, pp.853-857, November 3-6, 2002. Article (CrossRef Link)
[49] N. Saunier, H. Ardo, J.-P. Jodoin, A. Laureshyn, M. Nilsson, A. Svensson, L. Fernando Miranda-Moreno, G.-A. Bilodeau, and K. Astro, "Public video data set for road transportation applications," in Transportation Research Board Annual Meeting Compendium of Papers, pp. 14-2379, November 15, 2014. Article (CrossRef Link
[50] B.Babenko, M.H. Yang, and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.33, no. 7, pp. 1619-1632, December, 2011. Article (CrossRef Link)
[51] Y. Wu, J. Lim and M. H. Yang, "Online Object Tracking: A Benchmark," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2411-2418, June 23-28, 2013. Article (CrossRef Link)

Mustafa Eren Yildirim received his BS degree in Electrical Engineering from Bahcesehir University, Istanbul, Turkey, in 2008 and his MS and PhD degrees in Electronics Engineering, from the Graduate School of Electrical and Electronics Engineering, Kyungsung University, Pusan, Rep. of Korea, in 2010 and 2014, respectively. He worked as a researcher and lecturer for Kyungsung University, until August 2015. He is currently an assistant professor in Department of Electrical and Electronics Engineering at Bahcesehir University. His research interests include image processing, computer vision, recognition and object tracking.

Yucel Batu Salman received his BS and MS degrees in Computer Engineering from Bahcesehir University, Istanbul, Turkey, in 2003 and 2005, respectively and his PhD in IT Convergence Design from Kyungsung University, Pusan, Rep. of Korea, in 2010. Since 2010, he has been with the Department of Software Engineering, Bahcesehir University, Istanbul, Turkey where he is an assistant professor. He is currently vice-director of Graduate School of Natural and Applied Sciences at Bahcesehir University. His research interests include human computer interaction, mobile programming and computer vision.