An Approach to Track a Moving Object using Multiple Image Features Based Particle Filter

Abdalgalil Alsagair Abdulla, Stevica Graovac, and Veljko Papić

***Abstract*— There is a continuous increase in the applications for tracking moving objects in a video sequence, this paper focuses on tracking a moving vehicle in common situations for traffic scenarios. Particle filter-based algorithms (PF) have been widely used for this purpose. The algorithm analyzed in this paper is based on the fusing of three independent image features (color contents, texture, and object contour) by averaging their weighted similarities. To describe the tracked object, histograms of the features are used, with the object being a window that includes the tracked vehicle and the surrounding local background. The "Integrated PF" algorithm has been shown to be more effective than all Single-Feature PF algorithms. The algorithm's performance and verification was evaluated using both real and synthetic video sequences for some typical statuses of traffic scenarios, such as variations in shadows, passing other vehicles, and partial or full occlusion.**

Key Terms*— Vehicular tracking, particle filter, image features, multiple features, color histogram, texture, Grey Level Co-occurrence Matrix, image gradient.

I. Introduction

TRACKING objects is a crucial and essential aspect of many applications that involve visual object tracking (VOT). Some of these applications include road traffic control, monitoring, security, autonomous robot navigation, and tracking moving objects over a sequence of images, etc... [1] [2]. Usually, the goal is to monitor and track a moving single object or multiple objects, and obtain the trajectory by analyzing the data from sensors.

The task of detecting and tracking a moving object in a video sequence is a complex one, as it is subject to numerous challenges and unexpected events. These challenges include the presence of shadows or changes in illumination, variations in the background, partial or complete occlusion of the object, sudden changes in object motion, the presence of snow or dust, and others. From the mathematical perspective, most of these problems can be characterized as non-linear, non-Gaussian, multi-modal systems, or any mixture of these [3]. In our paper, we have focused on the tracking of vehicles, and an effective vehicle tracking on the road is an important component of traffic monitoring, as it serves as the foundation for more advanced tasks like traffic control and event detection, etc. [4]. We have chosen Particle Filter (PF) as our tracking technique, due to its ability to address the issue of non-linear and non-Gaussian dynamic states estimation, particularly in scenarios where clutter and occlusions are present. In the context of the processing of a series of images, probability density functions are represented by histograms that describe the distributions of particular features. To track a vehicle and its surrounding background within a specific window, one must select the appropriate features for the window and specify their mathematical representation.

The approach we present here, the "multiple features based particle filter," involves the fusing of weighted average of estimations using three features (color, texture, and edge descriptors) to track objects in video sequences. The results obtained using this “Integrated PF” algorithm has demonstrated clear advantages compared to the "Single-Feature PF". The authors have made several changes to the approach presented in their previous works [3, 5]. They have replaced the Bhattacharya distance with the Euclidean distance for measuring similarity. Additionally, they conducted several analyses related to the choice of similarity thresholds to improve tracking quality. Moreover, the authors have analyzed the reduction of some algorithm parameters, such as the propagation radius and the number of particles, as a measure for reducing the computational burden; this was done to find a compromise between tracking quality and real-time applicability of the algorithm.

In Part 2, the fundamental notions of object tracking based on PF are presented. Part 3 discusses the selection of certain features of an image and their effectiveness. The proposed algorithm is described in Section 4. In Section 5, a selection of artificial and real traffic scenarios has been used for algorithm validation, followed by the obtained results and appropriate explanation. Section 6 contains the conclusions. In the last section a list of references is presented.

II. Principles of Object Tracking Based on Particle Filter

Object tracking is the process of estimating the kinematic parameters (position, speed, direction of motion...) characterizing the moving object [6]. The sequence of images is a source of information for traffic monitoring, video surveillance, sports reporting, etc.

Tracking Strategies

Many different methods have been suggested for object tracking in computer vision literature. These methods mostly differ from each other base on the way how they are dealing

Abdalgalil Alsagair Abdulla, Technical College of Civil Aviation and Meteorology, Tripoli, Libya (e-mail: abdalgalilsagir@yahoo.com).

Stevica Graovac, retired professor of the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail:graovac@etf.rs).

Veljko Papic, School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: papic@etf.rs)

 ISBN 978-86-7466-970-9

with the following questions: How to represent the tracked object? Which features of the image should be used? What should be the model of the shape, appearance, and motion of the object? The answers to these questions are usually based on the status and surrounding environment in which the tracking is accomplished. According to the existing computer vision literature, there are four main approaches used for object tracking. They could be classified as model-based, region-based, active contour-based, and feature-based [2].

1. *Model-based:* In the estimation of the position of an object one uses the information about the appearance and shape of objects [7]. The weakest point of this method is the dependence on detailed geometric object models. It is impractical and/or un-expectable to have detailed models for all types of road vehicles as tracked objects.

2. *Region-based*: The tracked vehicle is a sub-region "rectangular window" of the frame that includes the vehicle and the local background. The main idea here is to use more information than in the case of the object alone, providing a greater robustness.

3. *Contour based*: The contour model is based on extracting the outline of an object [2, 8]. The advantage of a border representation instead of a region representation consists in decreasing computational complexity. However, the most important disadvantage of this approach is the inability to separate the vehicles that are in a partial occlusion situation and in the case of several vehicles grouped as one object.

4. *Feature-based*: Instead of tracking the object as a whole, sub-features like color, edge, points, corners or texture may be used for the tracking purpose. The advantage of using features based tracking is that even in the existence of partial occlusion, some of the features of the object in motion stay obvious. Besides, the same algorithm can be applied for tracking in daylight or night-time at different conditions [2]. In our work, the object of tracking is represented as a rectangular region, surrounding the object and encompassing both features of the object itself as well as of the local background.

III. VISUAL TRACKING BASED ON IMAGE FEATURES

In terms of tracking strategy, we decided to use a region/feature-based approach.

Tracking based on a single feature may lead to losing the object, and might start to track a wrong one. The adequate solution to this problem is employing multiple features. Features derived from the color, object's contour, and the texture inside the window, are intensively considered here in the context of tracking.

A. Color Feature

Several color spaces, including RGB, HSV, and YCbCr, are employed in the tracking applications described in the literature. The color feature is handy due to the fact that simple to compute, and the color contents remain invariant in rotational, strong in object size change as well [9]. But in case of illumination changes or if the colors of tracked object and the background are similar, the performance is poor [10]. The most common tool for describing color characteristics is the color histogram. In this work the RGB color space has adopted. The color histograms are typically calculated using 64 bins for each of the color channels. The predicted position of the tracked window could be chosen as the position of the particle that has best histogram compared to the others. Typical RGB color histograms for candidate windows in two different frames, in cases of synthetic and real sequences, are shown in Fig. 1. and Fig. 2. respectively.

B. Edge (Border) Feature

The object's edge in an image is defined as continuous set of pixels with high gradient intensities, and gives shape data. The change of intensities inside an image is highly connected to object borders because along the object boundary the intensity instantly changes. The sensitivity of edge features to illumination changes is less in comparison to color features [11]. In this paper, the edge feature is determined as a histogram representing a distribution of gradient directions for the edge points. In our proposed approach, the edges are extracted by calculating horizontal and vertical Sobel operators, K_x and K_y on the gray-scale image. The horizontal gradient G_x and vertical gradient G_y have magnitudes given by:

$$G_x(x,y) = K_x \times I(x,y), \;\; G_y(x,y) = K_y \times I(x,y) \qquad (1)$$

$$where \quad K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

where *I(x,y)* is the candidate window.

The magnitude (G) and phase (θ) of the edges are defined as:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}, \; \theta(x,y) = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \qquad (2)$$

The distribution defining the contour of the object is represented by the gradient phase angle histogram. For the unsigned gradient, the total range of phase angles is divided into 64 equal bins, ranging from -180^{o} to +180^{o}.

A gradient phase angle histogram for the candidate's windows in two different frames for synthetic and real images are shown also in Fig. 1. and Fig. 2., respectively.

C. Texture Features

The texture demonstrates how an image's intensity values or color are distributed spatially, these spatial distributions may be periodic, stochastic, or both [12]. Texture is also a measure of the intensity variation that defines numerous characteristics of a surface including linearity, directionality, regularity, smoothness, coarseness, uniformity, and density. The texture could be described by a variety of methods [13]. The statistical approach has been our choice here due to generality and simplicity of computing. The approach used in this work to represent texture descriptor is the Elements of Grey Level Co-occurrence Matrix (GLCM). The GLCM's chosen size is 8x8 (eight classes of grey level). The probabilities of occurrence are the GLCM's elements, which are converted to a vector forming the texture descriptor's histogram of 64 bins. Steps in forming this descriptor are given [14]. Fig. 1. and Fig. 2. show histograms for RGB color, gradient phase angles along the edge and texture, for candidate's windows in two different frames for artificial and real images in video sequences respectively.

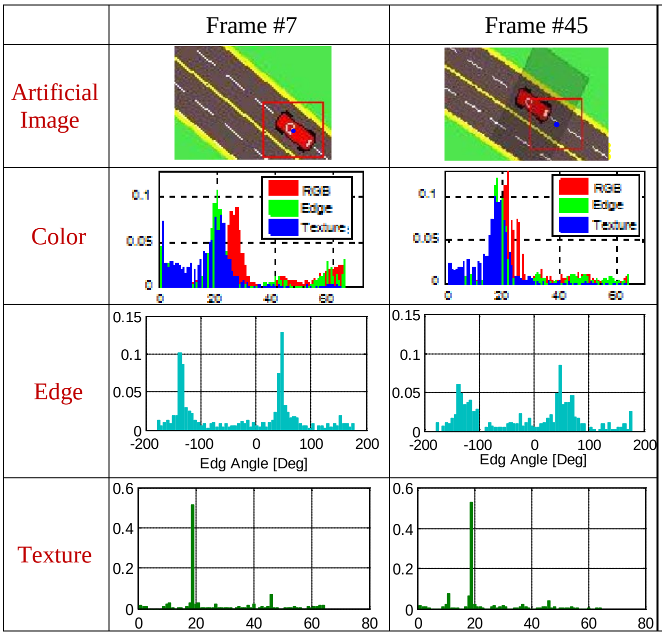


Fig. 1. Histograms of Color, Edge, and Texture features for the candidate's windows in artificial video sequence in frames 7 and 45

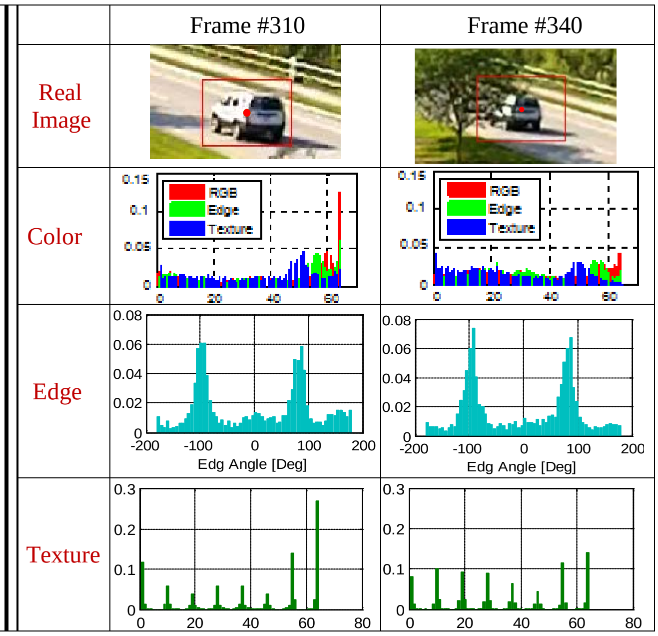


Fig. 2. Histograms of Color, Edge, and Texture features for the candidate windows in real video sequence in frames 310 and 340

The predicted position of the tracked window is chosen as the position of the particle where the matching between the reference region histogram and candidate region histogram is the best. Whenever the discrepancy between the histograms of the candidate and the reference windows was small, the likelihood of placing the object inside the candidate region is large. This discrepancy is measured by calculating the Euclidean distance (EC_{dist}) between histograms. Euclidian distance is calculated by

$$EC_{dist}^{i}(k) = \sqrt{\sum_{j=1}^{m} (H_{part}(j) - H_{ref})^2} \quad (3)$$

Where: 'i' is the particle index, 'k' is the frame number, 'j' is bin number, 'm', is the total number of bins, 'H_{ref}', is the extracted histogram of target or reference's window, and 'H_{part}', is the extracted histogram of a particular particle's window to be compared.

The observation likelihood model is used to assign a weight to a specific particle based on the similarity between the histogram of the reference window and the particle's window [2]. The similarity between the reference histogram and the particle's histogram is specified in (4):

$$SIM^{i}(\mathrm{k}) = 1 - EC_{\mathrm{dist}}^{\mathrm{i}}(\mathrm{k}) \quad (4)$$

D. Multiple Features

Using of one feature usually does not provide reliable performance of tracking when there are disturbances in the background or intervals when some typical interferences exist (the appearance of other objects in the tracking window), therefore a multiple-features based tracking can provide more information and more object's descriptions and get better robustness. The main disadvantage of a color feature is its weakness in the presence of objects or regions that have similar color features to those of the object of interest [1]. Also, edges and texture have ambiguity due to the rotation of the object. These ambiguities can be minimized by properly combining all three of these characteristics. The combination of features has some advantages provided that even if there is partial occlusion, some features of the moving object remain visible/ recognizable. Under these assumptions, the tracking algorithm is based on the simultaneous use of tracking algorithms based on three particular single features and the fusion process of these three results.

IV. INTEGRATED PARTICLE FILTER ALGORITHM

A particle filter can be described as a hypothesis tracker that can make an approximation of a posterior probability density distribution using a group of weighted random particles. The particles are weighted based on a likelihood grade and then spread according to transition motion [15]. Particle filter consists of three steps [2]:

- State transition (prediction) using the motion model: The state motion model specifies how objects move between frames. The assumption is made, that the vehicle moves along the straight line and with constant velocity in magnitude and direction, from frame to frame. The motion of the region of interest can be specified by:

$$\vec{X}_{k+1} = F\,\vec{X}_{k+1} \quad (5)$$

where $\vec{X}$ is the state vector, $F = \begin{pmatrix} 1 & \Delta k \\ 0 & 1 \end{pmatrix}$ is a transition matrix that expresses the dynamics of the state over time, and Δk is the sampling interval [12].

 ISBN 978-86-7466-970-9

- The Particle weights calculation (likelihood estimation) by monitoring the features using the likelihood model,
- Re-sampling: In this step, the new sample of particles set is chosen, and it neglects or multiplies particles based on the level of importance weights w_k^i to get N particles. The importance weight criterion is based on histogram distance, in the sense, that the low distance corresponds to the high weight and vice versa. If the similarity of any particle is less than a specified threshold, this particle is eliminated and replaced by a particle of a higher similarity which is selected randomly from the original set of particles.

Integrated PF algorithm steps

1. Initializing the target states X_0, the reference histograms for proposed features, and generate N random samples around the target point “x_0” in first frame,
2. Predict new state for each particle using a transition model,
3. Compute a histogram and a histogram distance (Euclidean distance for proposed features) for particles’ windows,
4. Weight each particle based on specified similarity measure and normalize the weights,
5. Select the location of the target as the particle that has maximal weight for each feature separately,
6. Merge adaptively the three locations based on the average similarity measures for updating the target's location,
7. Resample the particles for the next iteration,
8. Increase the time step, go to step 3.

V. OBTAINED RESULTS AND DISCUSSION

There were several scenarios for vehicle tracking tests that we have used. Here we have chosen some representative ones, divided into two groups - Synthetic video sequences and Real video sequences.

A. *Synthetic video sequences*

The assumed parameters for synthetic video sequence simulations have been:

- The time interval between frames (IBF) – 50ms.
- Tracked vehicle’s velocity – 90 km/h (25m/s, 1.25 m/IBF)
- The number of particles used in PF -120.
- The spreading radius of the particles 12 pixels.
- Tracking window size – 34 pix. × 34 pix.
- 1 meter ≈ 7 pixels.

Case-1: Tracking a vehicle moving along a straight line, without disturbance.

This case illustrates the tracking situation without any disturbances on the road, as shown in Fig. 3.

Frame #4 Frame #20 Frame #50

Fig.3 Tracked vehicle on the road.

Fig.4 shows the average similarity measures among 150 particles for each descriptor (4.a), and the difference between the estimated car positions and ground true values, using the proposed three descriptors separately (4.b).

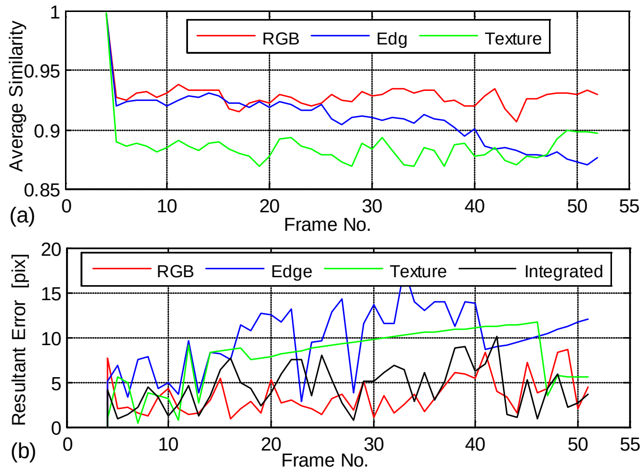


Fig.4 Case 1, a) Average similarity, b) Tracking error

From Fig.4.a one can see that the average similarity gives the preference to color feature. The RMS tracking errors for three individual descriptors are 4.00, 10.43, 8.45, while for the integrated algorithm, it was 4.99 pixels.

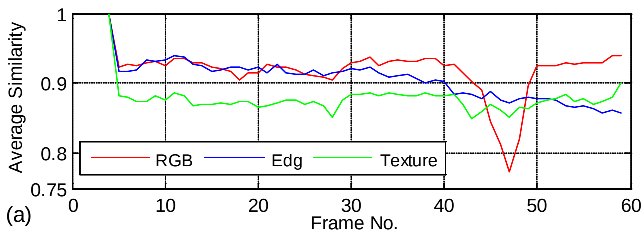
Case-2: Tracking in the presence of vehicle’s self-shadow shadow and shadow produced by road environment

The disturbances such as self-shadow, shadows produced by trees and buildings or clouds are inserted in this scenario and they appear abruptly at different intervals. Fig. 5.b illustrates the car with self-shadow (frame #15), while Fig. 5.c shows the presence of a shadow produced by nearby buildings (frame #45).

a b c

Fig.5 a) Reference frame (#4), b) Self-shadow effect, c) The effect of shadow produced by environment objects.

The obtained average similarities and deviations between the actual and estimated positions of the tracked vehicle are shown in Fig. 6. The error diagram (6.b) shows that in the presence of building shadow, the features are more affected than in the situation of self shadow. The RMS tracking errors for the proposed individual features and integrated algorithm were 11.02, 12.03, 7.40 and 10.84 pixels. The average similarity for color descriptor sharply decreased in the interval of existence of building shadow (see Fig. 6.a).



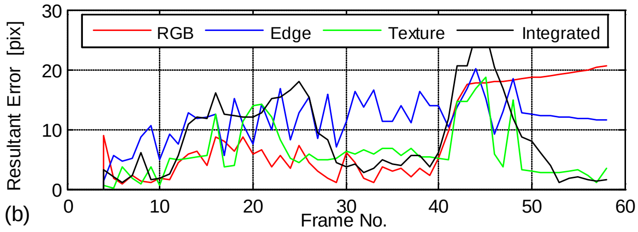


Fig.6 Case 2, a) Average similarity, b) Tracking error

B. *Real video sequences*

Some of the selected parameters for real video sequences have been changed in comparison to synthetic sequences. The spreading radius of particles was selected to be 15 pixels because the size of the vehicles in the selected real scenarios was bigger. The numbers of particles were selected to be 150. The sizes of windows for the selected four real scenario cases have been: 120×110, 96×68 and 46×50 pixels respectively

Case-3: A vehicle tracking in a real video sequence in the presence of its self-shadow and trees shadow

This case describes the tracking situation in the presence of self-shadow and shadow produced by the nearby trees. Fig 7. shows the target window for the reference frame (#235), and consecutive frames #263 and #291 with two types of shadows.

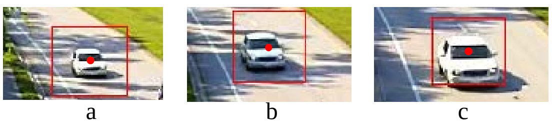


Fig 7 a) Reference frame (#235), b) car passes through a tree shadow (#263), c) car after tree shadow (#291)

The average similarities and tracking errors are shown in Fig. 8.a and Fig. 8.b.

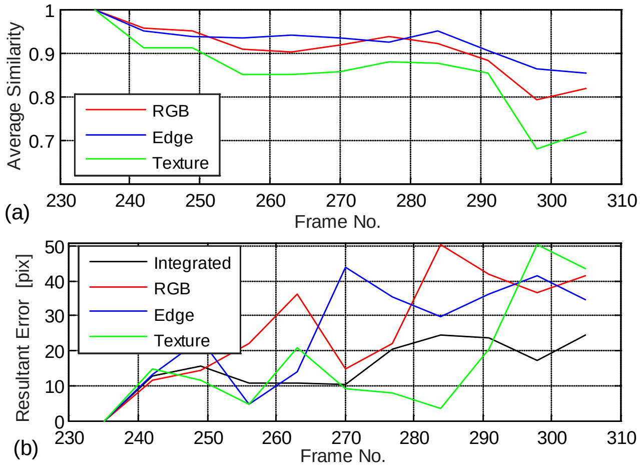


Fig.8 Case-3, a) Average Similarity, b) Tracking error

It is obvious from Fig. 8.a, that the texture queue rapidly loses importance after frame #280 due to changes in window contents (increasing of the non-shadowed area in the background). Based on the support of the other two features, Integrated PF remained with high quality due to the excluding of texture feature then. RMS tracking errors, in this case, have been 30.42, 28.97, 23.01 and 17.07 pixels, respectively.

Case-4: Vehicle exposed to partial and full occlusions

This case describes the complex situation where the vehicle is simultaneously exposed to a few typical problems in tracking (tree shadow, partial occlusion, full occlusion, variable vehicle scaling). The frames representing these situations are shown in Fig. 9.

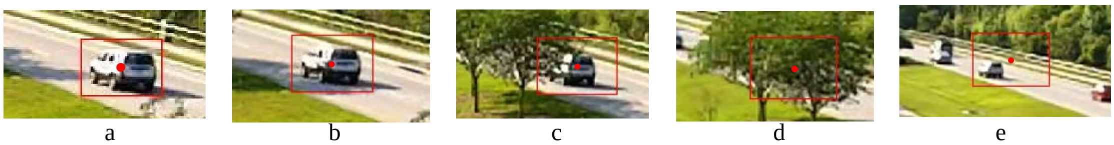


Fig. 9. a) Reference (# 310), b) Passing through a shadow (# 320), c) partial occlusion (# 340), d) full occlusion (# 360), e) reduced car scale (# 440).

Fig. 10 illustrates the average similarities and RMS errors.

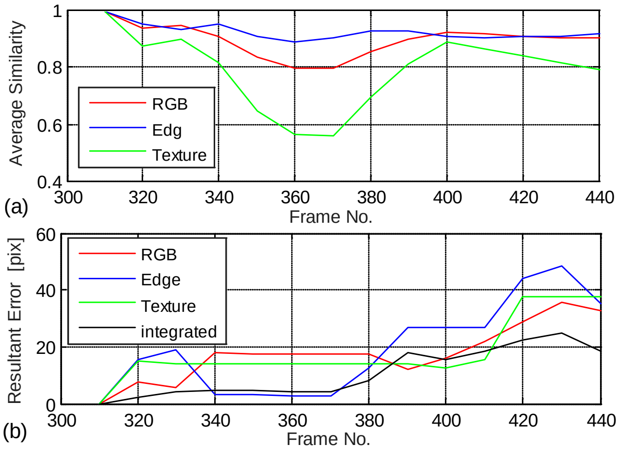


Fig 10. Case-4 a) Average similarities, b) Tracking errors

The RMS tracking errors for separate “Single feature PF” and “Integrated PF” are given in Table I. The tracking relies on prediction only at the period between the frames #350 and #370, due to occlusion, (average similarities of descriptors are less than their predefined similarity thresholds “0.90”). It is obvious from Fig. 10.a that the average similarity for all descriptors have been decreased along the interval of occlusion and increased after that.

Case-5 Vehicle exposed to partial and full occlusions

This case describes the complex tracking situation whereas the target vehicle is exposed to: occlusion, maneuver, and size changing. Fig.11. illustrates some important frames for these situations.

Fig. 11. a) Reference (# 448), b) partial occlusion (# 580), c) full occlusion (# 628), d) after occlusion (# 700)

Fig.12. shows the average similarities and tracking errors.

 ISBN 978-86-7466-970-9

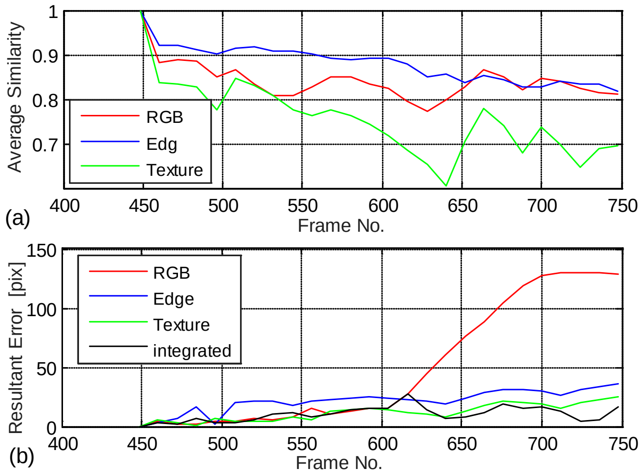


Fig. 12. Case-5, a) Average similarity, b) Tracking error

The Fig. 12.a shows that the average similarity of texture descriptor is the most sensitive in comparison to the others. The tracking errors (Fig.12.b) illustrate an increase of the error after the frame #600 due to missing during and after the occlusion period (Even a complete tracking failure for “Colour PF”). The RMS tracking errors obtained by the “Single feature PF” and “Integrated PF” are 70.64, 23.32, 13.33 and 12.11 pixels, respectively.

Table 1 illustrates the RMS errors in pixels for all seven considered cases. Analyzing the results obtained for real sequences (Cases 3 to 5), one can see the superiority of the integrated tracking algorithm.

Table-I below illustrates the RMS error in pixels for all 5 cases.

TABLE I
RMS OF TRACKING ERRORS IN PIXELS FOR ALL CASES

	Synthetic Sequences		Real Sequences		
	Case1	Case2	Case3	Case4	Case5
RGB	4,00	11,79	30,42	20,19	Lost
Edge	10,43	12,03	28,97	26,26	23,32
Texture	8,45	7,40	23,01	21,01	13,33
Integrated	4,99	10.84	17,07	13,42	12,11

VI. CONCLUSION

One approach to the Particle Filter based tracking of objects in a video sequence has been proposed and discussed in this paper. The basic assumption was that none of the typical image features could be used alone for different traffic situations and disturbances. Rather than attempting to improve any single features, we decided to fuse the three independent features (color, edge, texture) that have been expressed by particular probability distributions – histograms. A weighted averaging of independent PF estimates is made based on their average similarity measures of all particles. A higher level of adaptability has been reached by excluding any or all of the features when their average similarity measures are below specified thresholds. Verification of this approach was initially done using simple artificial image sequences and typical real scenarios. In the synthetic image sequences, the Integrated PF algorithm was not the best, but just “attracted” towards the best “Single Feature PF”, while it was superior in all realistic traffic sequences. It has especially been seen in occlusion cases, where some of the “Single Feature PF” was completely ineffective. While the presented approach has been acceptable for some simple traffic application scenarios (counting of vehicles, measurements of speed on pre-specified segments of a street/highway, etc.), our future work will be oriented on improving the algorithm regarding the effect of the vehicle size change as well as multiple-vehicle tracking.

REFERENCES

[1] L. Mihaylova2, P. Brasnett, N. Canagarajah, D. Bull, "Object Tracking by Particle Filtering Techniques in Video Sequences," in *Advances and Challenges in Multisensor Data and Information*, Amestrdam, IOS Press, Published in cooperation NATO Public Diplomacy Division, pp. 260-268, 2007.

[2] N. K. Ki and J. D. Edward,"New Models For Real-Time Tracking Using Particle Filtering," *Proc. SPIE 725,7 Visual Communications and Image Processing*, 2009.

[3] A. A. Abdulla, S. Graovac, V. Papic, and B. Kovacevic, "Triple-feature-based Particle Filter Algorithm Used in Vehicle Tracking Applications," Advances in Electrical and Computer Engineering (AECE), vol. 21, pp. 3-14, 2021.

[4] Ms. Susmita A. Meshram, Prof. A. V. Malviya, "Vehicular Traffic Surveillance for Real Time Using Multiple Methodologies," *International Journal of Scientific & Engineering Research*, vol. 4, no. 5, pp. 1988-1992, 2013.

[5] A. A. Abdulla, and S. Graovac, “Comparative analysis of the usage of different image descriptors in object’s video tracking," in *6th International Conference IcETRAN 2019*, Silver Lake, Serbia, 2019, pp. 174-179,

[6] M. Jaward, L. Mihaylova, N. Canagarajah and D. Bull, "Multiple Object Tracking Using Particle Filters," in *Aerospace Conference,IEEE*, pp. 1-8, 2006, doi: 10.1109/AERO.2006.1655926.

[7] N. Easwar, J. Shah, "Object Tracking using Particle Filter," CIS 601, https://slideplayer.com/slide/7259778/, 2003.

[8] K. Nummiaro, E. Koller-Meier and L. Van Gool, "A Color-based Particle Filte," in *First International Workshop on Generative-Model-Based Vision, in conjunction with ECCV’02, pp.53-60*, 2002.

[9] H. B. Kekre, & K. Sonawane,"Performance Evaluation of Bins Approach in YCbCr Color Space with and without Scaling," International Journal of Soft Computing and Engineering (IJSCE), vol. 3, no. 3, pp. 203-210, July, 2013

[10] Y. Dai, B. Liu, "Robust Video Object Tracking Using Particle Filter With Likelihood based feature fusion and adaptive template updating," in *eprint arXiv:1509.08182, Computer Science - Computer Vision and Pattern Recognition*, September, 2015.

[11] B. Deori and D. M. Thounaojam,"Asurvey On Moving Object Tracking In Video", International Journal on Information Theory (IJIT) , vol. 3, no. 3, pp. 31-46, 2014.

[12] P. Brasnett, L. Mihaylova, D. Bul, & N. Canagarajah, " Sequential Monte Carlo Tracking by Fusing Multiple Cues in Video Sequences," Image and Vision Computing, vol. 25, no. 8, pp.1217-1227, 2007.

[13] R. C. Gonzalez and R. E. Woods," *Digital Image Processing* (2nd ed.)", NJ., USA: Tom Robbins, 2002.

[14] X. Chen, "Automatic vehicle detection and tracking in aerial video", Retrieved from https:// dspace.lboro.ac.uk/2134/21743, 2015.

[15] Md. Z. Islam, C. Oh and C. Lee, "Video Based Moving Object Tracking by Particle Filter," *International Journal of Signal Processing, Image Processing and Pattern*, vol. Vol. 2, no. 1, pp. 119-132, 2009.

 ISBN 978-86-7466-970-9