

# Optimized UAV object tracking framework based on Integrated Particle filter with ego-motion transformation matrix

Wesam Askar<sup>1,\*</sup>, Osama Elmowafy<sup>2</sup>, Aliaa Youssif<sup>3</sup>, and Gamal Elnashar<sup>4</sup>

<sup>1</sup>Department of Electrical Engineering, Military Technical College, Egypt

<sup>2</sup>Computer Engineering Department, New Cairo Academy, Egypt

<sup>3</sup>Computer Engineering Department, Helwan University, Egypt

<sup>4</sup>Electrical Department, Military Technical College, Egypt

**Abstract.** Vision based object tracking problem still a hot and important area of research specially when the tracking algorithms are performed by the aircraft unmanned vehicle (UAV). Tracking with the UAV requires special considerations due to the flight maneuvers, environmental conditions and aircraft moving camera. The ego motion calculations can compensate the effect of the moving background resulted from the moving camera. In this paper an optimized object tracking framework is introduced to tackle this problem based on particle filter. It integrates the calculated ego motion transformation matrix with the dynamic model of the particle filter during the prediction stage. Then apply the correction stage on the particle filter observation model which based on two kinds of features includes Haar-like Rectangles and edge orientation histogram (EOH) features. The Gentle AdaBoost classifier is used to select the most informative features as a preliminary step. The experimental results achieved more than 94.6% rate of successful tracking during different scenarios of the VIVID database in real time tracking speed.

## 1 Introduction

Tracking objects in Unmanned Aerial Vehicle (UAV) camera image had been an important area of research many years ago. Many applications still require improvements in the performance of the video trackers due to many challenges related to this topic. The fast-abrupt motion, low resolution, noisy imagery, cluttered background, low contrast, and small target size are common issues that should be taken into considerations [1, 2]. There are also many factors govern the selection of the suitable object tracker such as the environment nature, object scale and types of the object motion [11, 12].

Thanks to a “bird’s eye view” delivered by the UAV camera, the UAV images is a vital choice for many critical application including traffic planning and surveillance, road conditions and emergency response, search and rescue operations, counter-terrorism and fighting against illegal immigration missions. The aerial images provide detailed information about the interesting target/targets that can be considered as the first and important step in the implementation of these application systems [2, 3, 5, 10].

The detection and tracking algorithms can be divided into single object or multi-object tracking [1]. Single target detection and tracking is very important in many critical applications. For instance, in military

applications, the process of alignment of the enemy target in the center of the camera field of view (FOV) is necessary for shooting purpose. However, the difficulty of the tracking will increase in case of tracking using UAV camera images. This is due to the moving of this platform which produce an ego motion effect between the moving object and the background. Also, the changing of the altitude of the UAV results a change in object scale which adds extra challenges on the tracking algorithms.

In most cases the tracking systems occurs in highly environments disparity. Not all the features components extracted by certain feature extraction method may be considered as an optimum feature to contribute in an object representation model. For this reason, the proposed system performs a preliminary classification step to determine the best feature components using boosting algorithm [11].

The particle filter is a Bayesian-based framework that gives an optimal solution for tracking problem involving a recursive prediction and correction step [1, 6]. The prediction step computes the prior probability density function (PDF) of the current state based on the system dynamics of the target and the PDF of the previous state. The correction step gives the posterior distribution of the object current state through the likelihood of the measurement (in the new frame) and the Bayes' rule. The particle filter uses a set of weighted samples (particles)

\* Corresponding author: [wesamaaa@gmail.com](mailto:wesamaaa@gmail.com)

to represent the target object PDF. Each particle represents a potential state for the object candidate state. In each time step, the particles will be updated based on the dynamics of the target object, and per their measurements, they will be re-weighted. a higher particle weight represents a higher probability state of the object. The next section will survey some of related work to the proposed system which is discussed in section 3. In section 4 the experimental results and discussion will be clarified and section 5 will contain the conclusion.

## 2 Related works

M. Josh et.al implemented an algorithm for victim detection and tracking in search and rescue operations using Unmanned Ground Vehicle (UGV) [3]. They aimed to reduce the operator effort by developing a semi-autonomous system that can follow the victim. This system consisted of two main stages includes Ego motion compensation stage and particles filter and clustering stage. The ego motion compensation algorithm tracks good selected features from frame to frame then constructs compensation matrix and compensated image. They improve the system performance by adapting the pyramidal level per velocity feedback.

In 2015 M. Abdelwahab et. al proposed a real-time technique for detecting, tracking and counting vehicles in simultaneously manner for airborne and stationary camera video [4]. They used Kanade–Lucas–Tomasi (KLT) Feature tracker to detect good features to track in the image frame. The non-stationary background points were removed by measuring the changes in the histogram of the pixels around each feature point with time to obtain the foreground features (FGF). Then they clustered and grouped them into separate trackable vehicles per the movement angles and displacement magnitudes. Their algorithm achieved real time performance for tracking vehicles in airborne videos without any prior knowledge for their locations and independent on their number.

Cao et.al use the KLT features and Random sample consensus (RANSAC) method to separate background features from moving objects and estimate the ego motion of the moving camera fixed in UAV [6]. They incorporated ego motion transformation matrix with the system model of the particle filter prediction step. The HSV color histogram and Hu moments were weighted and combined for computing the similarity measure and used them in the observation model of the particle filter correction step. The performance of their algorithm achieved tracking rate of 95%. However, the average tracking speed of the proposed method is 13.1 frames per second which is may not satisfy some application requirements.

Saif et.al presented and updated framework to handle six Uncertainty Constraint Factors (UCF) issues and challenges for moving object detection and tracking problems from UAV aerial images [2]. Theas six UCFs including the illumination change, environment clutter,

object type, Camera motion, moving object direction and motion complexity. They also dealt with the feature extraction problem as a separate unsolved issue because of the increasing of computation time related to the selections of large feature vector that suitable for optimum detection performance. They proposed a general framework for object detection problem from UAV aerial images. They employed a combination of frame difference and segmentation techniques for motion vector estimation and blob detection respectively. After that they suggested to use clustering to give physical meaning of overall detection. Finally, they recommended a proper classification step to distinguish between different types of objects that may include humans, vehicles, etc.

Moti et.al proposed a tracking method based on arbitration between Optical Flow (OF) and Kalman Filter (KF) techniques that can predict a target position in an efficient manner even it turns suddenly during its motion [7]. Their attention had been drawn to different situations where either the OF worked better or the KF did. They measured the distances to the nearest obstacle using the laser and used infrared camera images to detect the target object. Then fused these two types of data with the arbitrate OFKF filter for real-time tracking of a man in an indoor lab environment. By the same way, Shantaiya et. al proposed a simultaneously multiple objects tracking algorithm using Kalman Filter and improved Optical Flow [8]. They achieved better tracking accuracy using this improvement and combination relative to using each other separately, but the computation time still not suitable to track objects in UAV tracking requirements frame works.

## 3 Proposed Framework

In this paper the tracking framework can be divided into four stages includes pre-tracking, prediction, correction and post-tracking stages as illustrated in Fig. 1. The following sub-sections will give detailed descriptions for these stages.

### 3.1 Pre-Tracking Stage

The pre-tracking stage consists of four steps aimed to select the best features to represent the target from the first frame in the search area as illustrated in the following.

#### 3.1.1 Target detection

The pre-tracking stage is executed to extract the best features indices vector before start tracking process. Many tracking systems proposed algorithms to detect the target object from image sequence using manual or automatic manners. The proposed framework starts with manual selection for the target object by the user as the region inside a rectangular region from the whale image frame. This region of interest (ROI) is considered as a target object template that will be exposed to the feature

extraction step as illustrated in the following sub-sections.

where  $T$  is the number of all candidate features components.

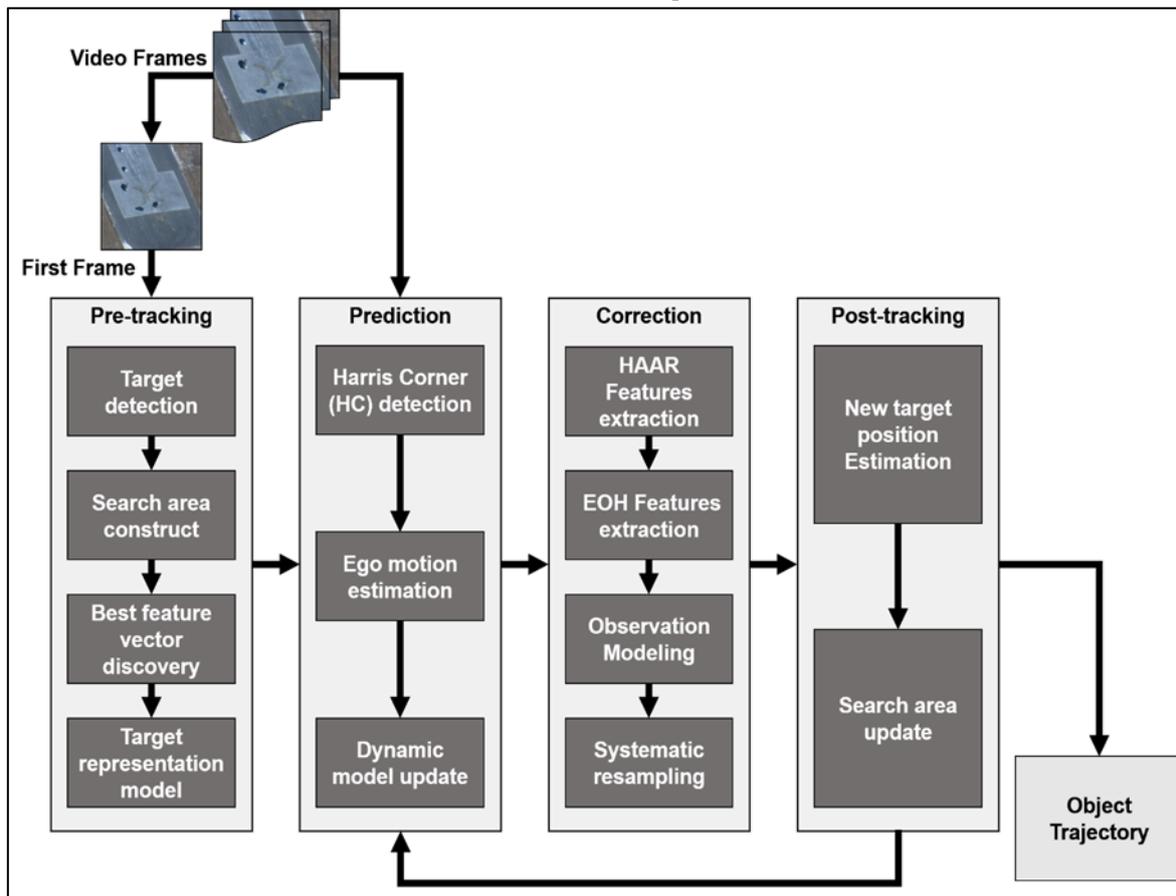


Fig. 1. The schematic diagram of the proposed framework.

### 3.1.2 Search area construction

The proposed framework starts with the construction of padding image  $IP$  before constructing the search area due to the possibilities of existing the target object position near the image borders. This may restrict formation of the search area to have the same bounding box during tracking process for all frame sequence. By finding the width  $w$  and height  $h$  of the ROI, the padding value  $P$  can be determined as  $n$  multiples of  $w$  or  $h$  (which one is bigger). Then search area can be constructed based on the ROI bounding box and  $P$ .

### 3.1.3 Best feature vector discovery

In this step the proposed system applies the Gentle AdaBoost classifier on a pre-constructed training dataset based on the search area foregrounds and backgrounds. By knowing target ROI, the  $f$  foregrounds can be extracted by different random views or positions around the ROI center and the  $b$  backgrounds can be extracted from different location the rest of the search area. All foregrounds and backgrounds have the same size of ROI  $w$  and  $h$ . The features extraction step based on Haar rectangles (HR) and edge orientation histogram (EOH) is applied on all foregrounds and backgrounds to extract all candidate features. These candidates are concatenated together to form a meas input vector of size  $(f + b) \times T$ ,

The Gentle AdaBoost classifier trains  $T$  weak classifiers (the same number of the candidate features components) and combines them into a linear fashion. Each classifier  $h_i(x)$  is a simple threshold function trained on one feature only. They are weak in the sense that they can only decide whether a value is below or above a given threshold. Although one-week classifier is not accurately to describe a whole dataset alone, a combination of them would lead to a strong classifier. A simple weak classifier example used in this paper is as the following:

$$h_i(x) = \begin{cases} 1 & \text{if } f_i(x) < \theta_i \\ -1 & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, T \quad (1)$$

Where  $f_i$  is the selected feature and  $\theta_i$  is the learned threshold. The output of the classifier ( $F_{idx}$ ) is in the form of vector with size  $T_{opt} \times 1$ , where  $T_{opt}$  is a pre-selected number of the strongest week classifiers (i.e. candidate features components).

### 3.1.4 Target representation model

As illustrated above, the target is represented by a rectangle region that contains  $w \times h$  pixels which is difficult to deal with them directly. So, a feature extraction process should be applied and specifically on the features with the index numbers inside  $F_{idx}$  vector. There are many types of features can describe the target

object. Due to the real-time requirements of the UAV tracking system, in this paper the Haar-like rectangles (*HR*) and the edge orientation histogram (EOH) are used because of their simplicity and fast computation cost [13, 14].

### 3.1.4.1 Haar-like Rectangles (*HR*) features

The HR features can describe the object with its color and much spatial information. They can be defined as a filter that computes the gray level difference between two defined white and black areas as the following:

$$Haar(x, y, w, h, type, C) = E_w(ROI) - E_k(ROI) \quad (2)$$

Where  $x, y$  are the coordinates of the top-left corner for any candidate ROI and  $w$  and  $h$  are its width and height respectively, the *type* is one of the Haar-like rectangles patterns listed in Fig. 2, the channel of the color space is represented in term  $C$  and  $E_w(ROI)$  and  $E_k(ROI)$  are white and black parts of *ROI*. The pool of the features consists of various *HR* per patterns, sizes and color channels. Thanks to the integral image (*II*) technique, the *HR* features can be extracted in very lower executional time. Using *II*, the summed values of a certain *ROI* can be efficiently computed by only four *II* accesses.

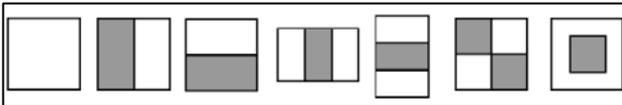


Fig. 2. different Haar-like rectangles patterns

### 3.1.4.2 Edge Orientation Histogram (EOH)

The EOH features are invariant with color and local spatial information. So, they are good for tracking targets in low light where the colors can be hard to distinguish [14]. To compute EOH first the search area is converted to grayscale image and is convoluted with the horizontal and vertical Sobel kernels to detect the edges on it. After that, the magnitude and direction are calculated for all pixels. Only pixels that have the magnitude greater than a pre-defined threshold ( $T$ ) are considered and the rest considered as noise. Then all edge directions fall between  $0$  to  $2\pi$  are quantized into ( $M$ ) number of bins and calculate. Using the integral image technique of the  $M^{\text{th}}$  bin matrix, the system can compute all possible components of EOH features as the sum of the magnitude for each bin relative to ROI.

## 3.2 Prediction stage

The prediction stage is the first step in the Particle filter algorithm that concerns for estimating the new position of each particle using a state transition model (dynamic model). In many cases the first order auto-regressive model is used and achieves good results in tracking systems [6, 7, 8]. However, in case of tracking objects using moving platform (UAV), this model should be optimized to compensate the ego motion resulted by the

moving UAV camera. The proposed system uses the optical flow technique and k-means clustering method for calculating the ego motion transformation matrix to increase the tracking rate. Then the transformation matrix is combined with the dynamic model to produces the proposed state transition model for the particle filter technique as discussed later.

### 3.2.1 Harris Corner (HC) detection

To calculate the ego motion, the system starts with dividing the current frame into sub-frame regions (blocks) then detecting the Harris Corner points (HC) [15] on them. This HC points may relate to fixed objects (background) or moving objects (foregrounds). However, we first assume that the most of them are related to the background. The optical flow method is used to estimate the new position of HC points in the next frame. Fig. 3(a, b) shows HC points and their correspondences in all frame blocks.

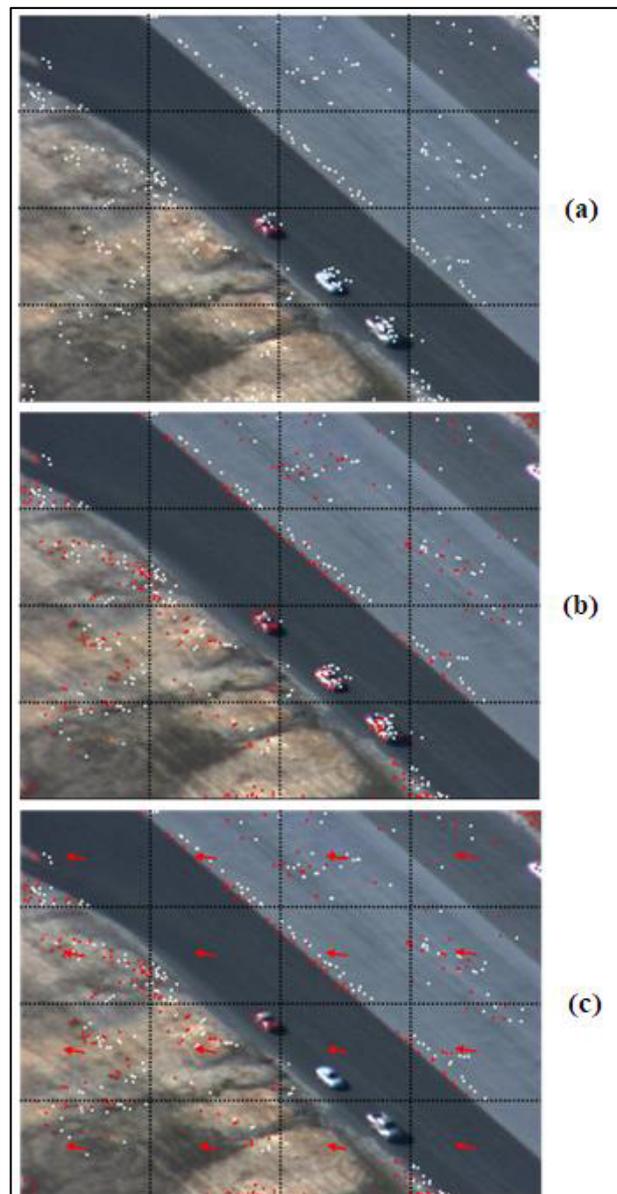


Fig. 3. Harries Corner (HC) detection and estimation.

### 3.2.2 Ego Motion Estimation

After having the new HC points and their correspondences, the Euclidian distance (ED) and angles ( $\theta$ ) are calculated for each HC pairs ( $(x_i, y_i)$  and  $(y'_i, y'_i)$ ) per equation 3, 4. The proposed system applies the k-means method on all HC points belong to each block. It aims to cluster the HC data ( $\theta$  and ED) into two groups representing the foreground (moving objects) and backgrounds. The group contains of the most HC points is considered as a background and the foreground points are removed as shown in Fig. 3(c).

$$ED = \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2} \quad (3)$$

$$\theta = \frac{y_i - y'_i}{x_i - x'_i} \quad (4)$$

The Maximum Likelihood Estimation Sample Consensus (MLESAC) method is applied on the background HC pairs to estimate the affine geometric transform for each block. The MLESAC is a generalization of RANSAC estimator with a maximization of the likelihood rather than just the number of inliers [16]. Due to the possibilities of image rotation, skewing, and warping, not all blocks have the identical affine transformation. So the median value of all affine transformations is computed and is considered as the ego motion transformation (E). The transformation E is an 3x3 affine transformation matrix with the following form:

$$E = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where a, b, c, d, e and f are the matrix components which determine the values of translation, scale, sheer and rotation transformation types.

### 3.2.3 Dynamic model update

As mentioned above the calculate ego motion transformation matrix is cooperated with auto-regression model to formulate the state transition model for the (N) particles. Each particle can represent a candidate object state (X) as the following:

$$\mathbf{X} = (x, y, 1, \dot{x}, \dot{y}, w, h)^T \quad (6)$$

Where x and y are the left-top coordinate of the ROI rectangle and w and h are its width and height respectively. The velocity information is represented by  $\dot{x}$  and  $\dot{y}$  components. The constant number 1 inserted for matrix operations convenient. The proposed first-order auto-regression model can be expressed by the following equations:

$$x_t = x_{t-1} + \dot{x}_{t-1} + Gx_{t-1}(0, \sigma_x) \quad (7)$$

$$y_t = y_{t-1} + \dot{y}_{t-1} + Gy_{t-1}(0, \sigma_y) \quad (8)$$

$$\dot{x}_t = \dot{x}_{t-1} + G\dot{x}_{t-1}(0, \sigma_{\dot{x}}) \quad (9)$$

$$\dot{y}_t = \dot{y}_{t-1} + G\dot{y}_{t-1}(0, \sigma_{\dot{y}}) \quad (10)$$

$$w_t = w_{t-1} + Gw_{t-1}(0, \sigma_w) \quad (11)$$

$$h_t = h_{t-1} + Gh_{t-1}(0, \sigma_h) \quad (12)$$

Where  $G_*(*)$  is zero mean white Gaussian noise components for several unidentified dynamic factors that may happened during tracking (sudden random motion, slight acceleration, ...).  $\sigma_*$  is the variance of  $G_*(*)$ . After calculating the ego motion transformation matrix  $E_{3 \times 3}$ , the system takes only the first two rows in E and updates only the two position components in the state  $\mathbf{X}$  (i.e.  $x_t$  and  $y_t$ ) as the following equation:

$$\begin{bmatrix} xe_t \\ ye_t \end{bmatrix}_{2 \times N} = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}_{2 \times 3} \times \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix}_{3 \times N} \quad (13)$$

Where  $xe_t$  and  $ye_t$  are the two updated position components of the target object at current frame t.

The equations (11, 12) are updated by inserting the effect of scale change represented by a and d components in the ego motion transformation matrix E as the following:

$$we_t = w_t * a \quad (14)$$

$$he_t = h_t * d \quad (15)$$

Empirically, the nominal values of a and d components were between (0.97 – 1.3) which denotes to decreasing or increasing the object scale.

### 3.3 Correction Stage

The main objective of the correction stage is to calculate the weights corresponding to the new samples in each time step. It consists of three steps including the first two HR and EOH features extraction steps as discussed before. In the observation modeling step, the system assigns weights to all particles according to a similarity score between the target object and the candidate particle ROI.

Based on the new weights of the particles (samples), the systematic resembling step is applied to reject samples which have very low weight and to concentrate on particles with large weights. In the beginning the system assigns equal weights to the particles. Then, by calculating the observation likelihoods in the next frames, each particle represents a good candidate to the object will be assigned to a high weight.

### 3.4 post-tracking

In this stage the proposed system can predict the new state of the target by taking the median of all candidate states represented by the particles. Because of the probability that the particles may have status positions relatively far from the object center, the median filter is used. It achieves good results better than average filter due to the minimal influence to the noise. Based on the estimated new position of the target object, the new

search area is determined again to use in object tracking in the next frame.

#### 4 Experimental results

To evaluate the proposed object tracking framework in different scenarios, the VIVID database [17] is used with small modification to simulate more difficult environments. This modification performed by adding some drifting noise in each new frame. This drift can be considered as UAV vibrations or camera shacking. The VIVID includes civilian and military vehicles moving on a runway with different kinds of motions (straight and loop) and changing speed. Also, the scale of them are changing in the image due to the change of UAV altitude. In some situations, the tracked vehicle looks very similar to another when it passes.

The algorithms implemented using MATLAB and all the experiments were executed on a 2.6 GHz computer with 4 GB RAM. The performance results achieved 94.6% successful tracking rate with calculation speed more than 48 frames per second (fps) for 640×480 pixels images.

At the first frame the system selects randomly 20 different sub-images around the object as foregrounds and 200 sub-images from the background. Then a pool of 106 extracted features (90 HR and 26 EOH) are created to be the inputs for the AdaBoost classifier for selecting the best  $F$  features that represent the object. In calculating the ego motion the system divides the image frame into 4×4 sub-image and choose only the strongest 20  $HC$  in each. Finally, the particle filter set at 500 particles to represent the candidate object locations.

To evaluate the system the recall and precision are calculated in same manner as in [6]. Fig. 4 and Fig. 5 show the average accuracy and the speed of the system using different values of  $F$ .

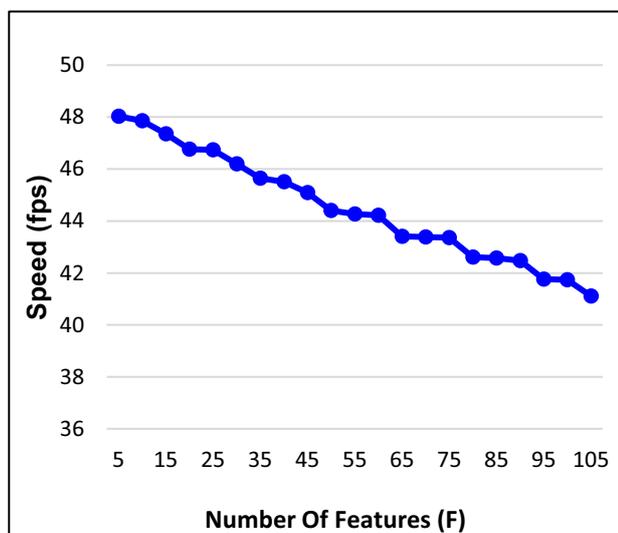


Fig. 4. Accuracy of the proposed system for different values of number of features.

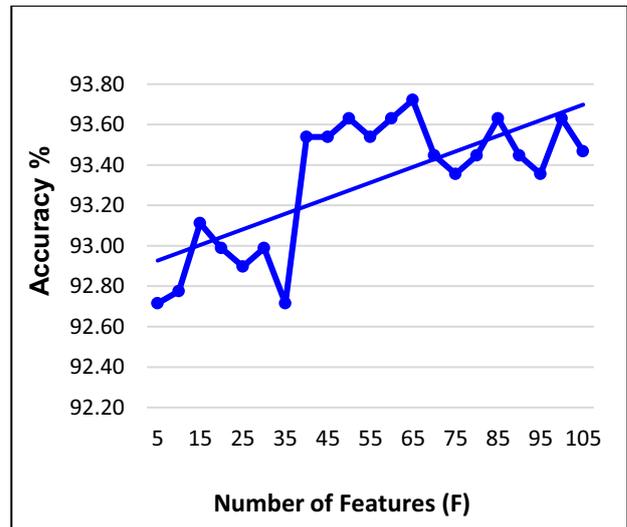


Fig. 5. Speed of the proposed system for different values of number of features.

As mentioned before the ego motion incorporation with the dynamic model improves the accuracy of the framework. Fig. 6. Illustrates the accuracy of the system for the first image sequence of VIVID database (egtest01) which contain vehicles that loop around on a runway, then drive straight. Starts from frame 900 vehicle being tracked speeds up and passes very similar to another vehicle. When the vehicle speeds up the particles lags the center of which means that the dynamic model does not guides these particles and predicts the new position as better as before. So, the accuracy values are decreasing (red curve). By incorporating the ego motion transformation matrix with the dynamic model, the particles directed to closer position to the vehicle center resulted in improving the accuracy value (green curve).

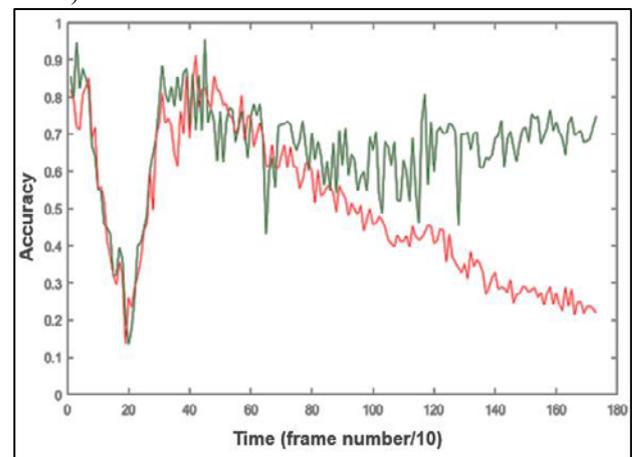
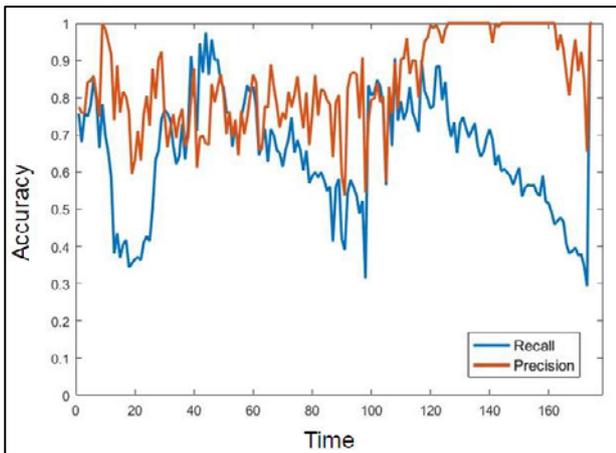


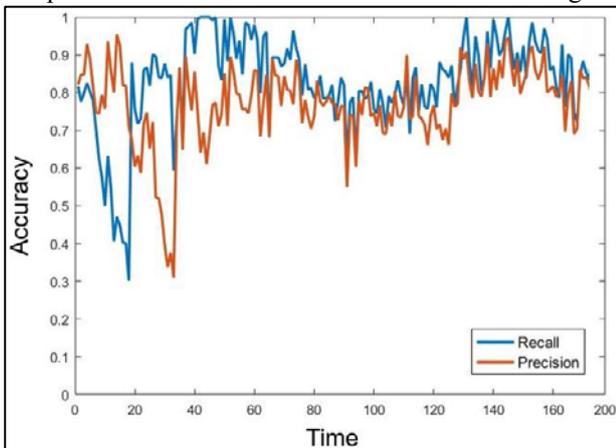
Fig. 6. Accuracy for VIVID (egtest01): (red) the accuracy without ego motion incorporation, (green) the accuracy with ego motion incorporation.

The effect of scale change appears obviously when we examine the recall and precision measures separately as shown in Fig. 7. Approximately after frame 1300 the UAV altitude was slightly decreasing results in increasing in the object scale which leads to decreasing in recall results.



**Fig. 7.** Recall and Precision Accuracies without compensating the scale change effect.

As mentioned before by updating the dynamic model again using equations (13) and (14), the width and the height of each candidate particle bounding box will get closer values as the object ground truth. Then the recall and precision accuracies will increase as shown in Fig. 8.



**Fig. 8.** Recall and Precision Accuracies with compensating the scale change effect.

Table 1 is showing a comparison between the proposed framework and the tracking method in [6] in both speed and accuracy to illustrate the improvement of the performance specially the real-time requirement with the satisfactory accuracy. The tracking speed using the proposed framework is approximately four times the speed in the other method. The accuracy didn't reduce by large value and can be consider the same in the both methods due to the probabilistic nature of the experiments.

**Table 1.** Comparison between the proposed tracking framework and the classical particle filter.

Tracking Method	Speed (fps)	Accuracy %
Classical Particle Filter tracking method	13.1	95.1
Proposed Particle Filter tracking framework	47.6	94.6

There are many factors for the improvements in the speed. The use of the integral images for calculating the HR and EOH features reduced the execution time

significantly. However, this execution time is directly proportional to the image size. In single object tracking purposes using UAVs, the object size is small. So, it is intuitively to concentrate the calculations on a search area as minimum as possible to reduce the calculations but as maximum as possible to includes the object candidate positions. In this paper, the search area was chosen first to be three times greater than the object size while the object speed per frame not exceeds this value.

## 5. Conclusion

In this paper an optimized object tracking framework is introduced to achieve the UAV vision based object tracking requirements. The tracking process starts with selecting the best features form a pool of Haar-like rectangles (HR) and edge orientation histogram features (EOH). This is done by applying AdaBoost classifier to all features that extracted from a pre-created set of foregrounds and backgrounds in the first frame. The selected features are considered as the observation model of the target object. This observation model is used for the particle filter correction stage to compare with candidate position denoted by each particle. Before the correction step the new position of the object is predicted using a modified dynamic model of the object. This modification is done by calculating the ego motion of the UAV using fixed background points and comparing between them to extract the affine transformation matrix which incorporates with the object dynamic model. By using this framework, the tracking speed and accuracy achieves better results than most of recent UAV tracking methods.

## References

- [1] A. Yilmaz, O. Javed and M. Shah, ACM Computing Surveys, "Object Tracking: A Survey", Vol. **38**, No. 4, Article 13, (2006).
- [2] A.F.M. Saifuddin Saif, A. Prabuwno, Z. R. Mahayuddin and H. Himawan, International Journal of Advancements in Computing Technology (IJACT), "A Review of Machine Vision based on Moving Objects Object Detection from UAV Aerial Images", Vol. **5**, (2013).
- [3] M. Joshi, R. Madri and M. Joshi, TENCON 2013 - 2013 IEEE Region 10 Conference (31194), "Real time motion tracking algorithm for search and rescue robots", (2013).
- [4] M. A. Abdelwahab and M. M. Abdelwahab, Multimedia (ISM), IEEE International Symposium on, "A Novel Algorithm for Vehicle Detection and Tracking in Airborne Videos", (2015).
- [5] P. Burdziakowski, M. Przyborski and J. Szulwic, 1st Int. Con. on Innovative Research and Maritime App. of Space Technology IRMAST, "A Vision-Based Unmanned Aerial Vehicle Navigation Method", (2015).

- [6] X. Cao, C. Gao, J. Lan, Y. Yuan and P. Yan, Neurocomputing, “*Ego motion guided particle filter for vehicle tracking in airborne videos*”, Vol. **124**, pp. 168–177, (2014).
- [7] Y. Motai, S. K. Jha and D. Kruse, Image Communication, “*Human tracking from a mobile agent: Optical flow and Kalman filter arbitration*”, Vol. **27**, pp. 83–95, (2012).
- [8] A. Shantaiya, K. Verma and K. Mehta, Europ. J. of Adv. in Eng. and Tech., “*Multiple Object Tracking Using Kalman Filter And Optical Flow*”, pp. 34–39, (2015).
- [9] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, ECCV 2002, LNCS 2350, “*Color-Based Probabilistic Tracking*”, pp. 661–675, (2002).
- [10] H. Yang, L. Shao, F. Zheng, L. Wangd and Z. Song, Neurocomputing, “*Recent advances and trends in visual tracking: A review*”, Vol. **74**, pp. 3823–3831, (2011).
- [11] S. Sivaraman and M. M. Trivedi, IEEE transactions on intelligent transportation systems, “*Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis*”, Vol. **14**, no. 4, (2013).
- [12] M. Coskuna, S. Ünala, 9th Int. Conf. Inter. Eng., “*Implementation of Tracking of a Moving Object Based on Camshift Approach with a UAV*”, (2015).
- [13] B. L. Sefidgari, Int. J. of Artificial Intelligence & App. (IJAIA), “*Feed-Back Method Based On Image Processing for Detecting Human Body Via Flying Robot*”, Vol. **4**, No. 6, (2013).
- [14] D. Geronimo, A. Sappa, A. Lopez and D. Ponsa, Proc. 5th Int. Conf. Comput. Vis. Syst., “*Adaptive image sampling and windows classification for on-board pedestrian detection*”, (2007).
- [15] J. Shi and C. Tomasi, IEEE Conf. on Computer Vision and Pattern Recognition, “*Good Features to Track*”, (1994).
- [16] P. H. S. Torr and A. Zisserman, Computer Vision and Image Understanding, “*MLESAc: A New Robust Estimator with Application to estimating Image Geometry*”, (2000).
- [17] R. T. Collins, X. Zhou, and S. K. Teh, IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January, 2005. “*An Open Source Tracking Testbed and Evaluation Web Site*”, available online until 14<sup>th</sup> June 2017 at: <http://vision.cse.psu.edu/data/vividEval/main.html>