# Real-time UAV Target Tracking System Based on Optical Flow and Particle Filter Integration

WESAM ASKAR
Electrical Engineering
Military Tech. College
EGYPT
wesamaaa@gmail.com

OSAMA ELMOWAFY
Computer Engineering
New Cairo Academy
EGYPT
elmowafy@hotmail.com

ALIAA YOUSSIF
Computer Engineering
Helwan University
EGYPT
aliaay@fci.helwan.edu.eg

GAMAL ELNASHAR
Electrical Engineering
Military Tech. College
EGYPT
enjygamal@gmail.com

*Abstract:* - This paper presents a design and implementation of a real-time, vision-based target tracking system for unmanned aerial vehicle (UAV). The particle filter framework integrated with Lucas-Kanade optical flow technique to predict and correct the state of the moving target based on its dynamic and observation models. The optical flow estimates the corresponding feature points in the new image frame related to the previously detected/estimated points. The Maximum Likelihood Estimation SAmple Consensus (MLESAC) method is applied to estimate the ego-motion transformation matrix using the old and new sets of the feature points. This matrix is incorporated with the target dynamic model to give more accurate prediction results of its state. Two optimized types of features are extracted to build the target observation model. They include extended Haar-like rectangles and edge orientation histogram (EOH) features. A Gentle AdaBoost classifier is applied on these features to distinguish and choose the best predefined number of features that highly represent the target. The vectorization approach is used to reduce the calculation cost due to the matrix manipulations. The proposed tracking system is tested on different scenarios of the on-time modified VIVID database and achieved real time tracking speed with 95% successful tracking rate.

*Key-Words:* - Machine Vision, Image Analysis, Video Tracking, UAV Tracking, Lucas-Kanade Optical Flow, Bayesian Particle Filter, Ego-motion

## 1 Introduction

The Unmanned Aerial Vehicle (UAV) applications have received a great attention in recent years. The UAV camera delivers a "bird's eye view" mapped on 2D digital images. The UAV images provide massive information about the captured scene along with the static and dynamic objects inside it [1] [2].

Many systems inspire the extracted information to support numerous applications with the needed data to take important decisions. These applications include target detection, recognition and tracking systems in addition to surveillance, traffic planning, emergency response, search and rescue operations, counter-terrorism and fighting against illegal immigration missions [3] [4] [5].

Vision based target tracking systems still require improvements in their accuracy, scalability and real time performance. Also, they should take into considerations solving many problems related to environment nature and background cluttering, the size and the scale of the target, types of the target motion, noisy and low contrast imagery [6] [7].

The detection and tracking algorithms can be classified according the number of targets to be tracked into single or multi-object tracking [1] [2]. Single target tracking is very important demand for special applications such as shooting a single enemy target. It requires centering the targeted enemy on the field of view (FOV) of the UAV camera and keep locking it. One of the main difficulties related to this process is the camera platform movement with the UAV flying which produce a non-fixed background in the captured image sequence. This results in an ego-motion effect between the moving object and the background [8]. The object scale has no fixed values throughout tracking process because of the varying UAV altitude during the flight. This also another challenge that should be taken into consideration in the implementation of the tracker system [1] [2].

The particle filter is a Bayesian-based framework that gives an optimal solution for tracking problem involving a recursive prediction and correction steps [8] [5]. In prediction step target state is predicted in the new frame based on the system dynamics. The correction step updates the target current state through the likelihood of the new measurement. The particle filter uses a set of weighted particles to represent this target state. Each particle specifies a candidate potential state for the target. In each time

step, the prediction and correction steps updates the particle distribution based on t he target dynamics and new measurements. Then, they are re-weighted giving the resulted likelihood.

The target observation model is constructed based on the target extracted features. In first frame, the proposed system extracts a pool of two kinds of features includes extended Haar-like rectangles *HR* (normal and oriented $45^{\circ}$) and edge orientation histogram *EOH* features because of their simplicity and fast computation cost [9] [10]. Then Gentle AdaBoost classifier is applied on these features to select the best $(n)$ features to represent the target and carry information about it [6].

The proposed system implemented utilizes the vectorization technique in all matrix operations to reduce the calculation cost. It provides speed, efficiency and performance better than the use of for-loop manner. It reduces the memory overhead of physically expanding the data before performing the binary functions. The next section will converse a literatures survey about the related works. The proposed system is discussed in section 3 in details. Section 4 will clarify and discuss the experimental results. The conclusion about the proposed system will be in section 5.

## 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) [11]. 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 [12]. 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 [8]. 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 [3]. 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 sep arate 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 individuals, 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 [13]. 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 simultaneously multiple objects tracking algorithm using Kalman Filter and

improved Optical Flow [14]. 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

The proposed tracking system consists of four main phases as shown in Fig.1 In first phase the target is selected within search area and apply feature extraction algorithms to build the observation model. The second phase calculates the ego-motion transformation matrix to incorporate with the target optimized dynamic model and perform the prediction step. The third phase corrects the target state using the new observations. Finally, the new target position is estimated in the fourth phase. Detailed descriptions for the four phase will be discussed in the following sub-sections.

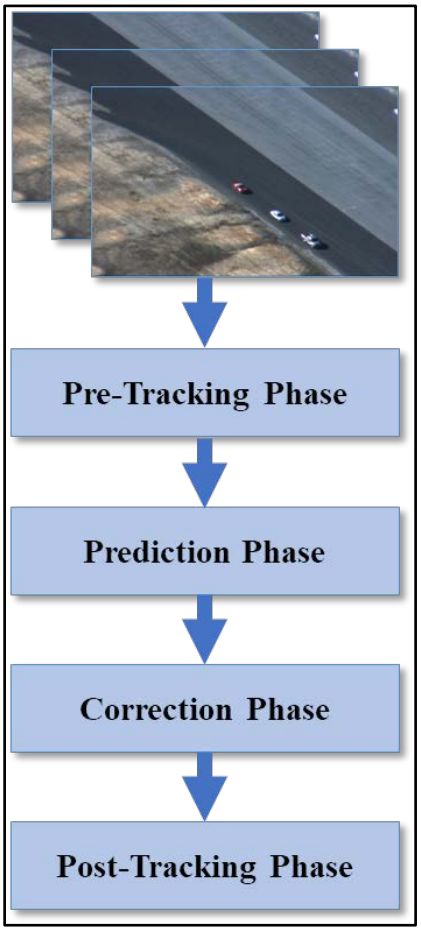


**Fig**.1 Proposed tracking system block diagram.

### 3.1 Pre-Tracking Phase

In the first frame, the target is selected manually and represented by a rectangle. Then all *HR* and *EOH* features are extracted for the selected target within a pre-defined search area. The main purpose of this step is to represent this target observations $O_T$ to compare with each candidate target in the next frames. The extracted features don't represent the target in identical degrees. So, only the best $F$ number of features are selected. The Gentle AdaBoost classifier are used to distinguish and classify between the features. The following subsections will describe the steps of pre-tracking phase in details.

#### 3.1.1 HR features extraction

The Haar-like rectangle features *HR* designate target color and its spatial information. They compute the difference between the defined white and black areas of the *HR* filter using the following equation:

$$HR(x, y, w, h, type, C) = E_w - E_k \quad (1)$$

Where $x, y$ stands for the top-left corner coordinates of the *HR* filter defined in pattern $type$. Fig.2 shows examples of normal and oriented $45^{\circ}$ *HR* patterns used in this paper. The $w$ and $h$ are the *HR* width and height respectively. The term $C$ stands for one channel of the used color space (*R*, *G*, *B*). $E_w$ and $E_k$ are the summation of pixels inside white and black parts of *HR* filter.

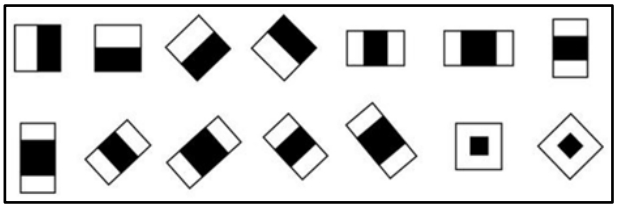

**Fig**.2 different Haar-like rectangles patterns.

The pool of the extracted features contains numerous *HR* patterns, sizes and color channels. The computational time of extracting *HR* features depends on m any aspects includes the number of pixels to be summed within each area of *HR* filter. The Integral Image is used to reduce this time because It requires only four memory access [15]. The integral image $II$ of an image $I$ for the pixel at location $(x, y)$ can calculated using equation (2):

$$II(x, y) = I(x, y) + II(x - 1, y) + II(x, y - 1) - II(x - 1, y - 1) \quad (2)$$

where $II(x, -1) = II(-1, y) = 0$. Once the integral image $II$ is calculated over image $I$, the summation $S$ of all values of the pixels within any rectangular area with upper left corner $(x_1, y_1)$ and lower right corner $(x_2, y_2)$ can be computed using equation (3) as shown in Fig.3:

$$S = D - B - C + A \quad (3)$$

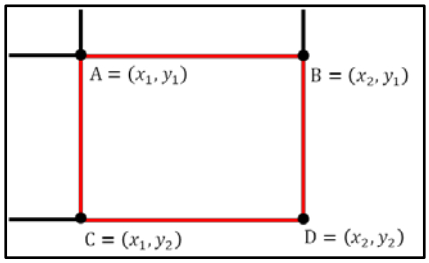


**Fig**.3 Calculating summation of pixel values within certain area of image *I*.

### 3.1.2 EOH features extraction

To increase the robustness of the observation model, the proposed system combines between *HR* and *EOH* features. The *EOH* features have discriminant power towards abrupt changes in illumination intensities. Also, the objects which have same colors as the background can be distinguished by *EOH*. So, they give good results for tracking targets in low light environment where the colors are hardly distinguished, such as the case of infra-red images [10]. The first step for *EOH* features extraction from an image is to apply vertical and horizontal Sobel filters on the corresponding grayscale image. This produces the corresponding horizontal and vertical edge maps $G_x$ and $G_y$, respectively. Then the magnitude $M$ and direction $\theta$ are computed for every pixel $(x, y)$ using the following equations:

$$M = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \quad (4)$$

$$\theta = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (5)$$

The $\theta$ greater than pre-defined threshold are considered as noise. All edge directions are quantized between 0 to 2π into *B* number of bins as shown in Fig.4.

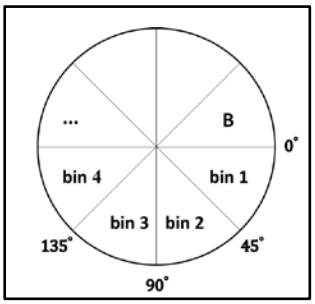


**Fig**.4 Edge quantization bines.

The direction ranges of each bin $b$ are defined by equation (6).

$$binRange_b = \left[(b-1) * \left(\frac{2\pi}{B}\right), m * \left(\frac{2\pi}{B}\right) - 1\right] \quad (6)$$

This process produces a binary image corresponding to each bin and has the same size as the image frame. Each binary image $b$ is multiplied by the magnitude $M$ and generates $B$ number of quantized magnitude images. Finally, the *EOH* features for a certain region are computed as the summation of the magnitude of all pixels within that region. Similarly, the integral image is computed in advance to reduce the summation time and increase the performance of the system.

### 3.1.3 AdaBoost features classification

The Gentle AdaBoost classifier trains $T$ week classifiers and combines them into a linear fashion, where the value of $T$ equals to number of features $F$. Each classifier $h_i(x)$ is a simple threshold function trained on numerus values of one type of feature. 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:

$$h_i(x) = \begin{cases} 1 & if\ f_i(x) < \theta_i \\ -1 & otherwise \end{cases} \quad i = 1,2,\dots T \quad (7)$$

where $f_i$ is the selected feature and $\theta_i$ is the learned threshold. Finally, The AdaBoost classifier produces the optimal $N$ features index to be used for target/candidate targets representations.

## 3.2 Prediction Phase

The first step in the particle filter framework is the prediction of the target state. It estimates the new position of each particle that represent a candidate target position using the state transition model. The first order auto-regression model is usually used to describe the transition of the target during its movement and achieves good results for tracking systems [8] [13] [14]. The proposed system modifies this model to compensate the ego-motion effect resulted by the motion of the UAV and hence its camera. First, the system uses the optical flow technique and k-means clustering method to calculate the ego-motion transformation matrix that reflects the motion of the UAV and accordingly the images background. Then it is combined with the dynamic model to produces the proposed state transition model. The following subsections explains the steps of this phase in more details.

### 3.2.1 Harries corner *HC* detection and estimation

The proposed system assumes image background to occupies the most of image area. So, it detects the

Harris corner *HC* points to represent the background in the current image frame, Fig.5(a) [16]. In the next image frame, the Lucas-Kanade optical-flow algorithm estimates the new position of *HC* points and the system creates the *HC* pair set, Fig.5(b).

Although most of *HC* and their correspondences belongs to the background, a considerable portion of them are related to the moving targets (foreground). So, the system uses the k-means algorithm to distinguish between them and then eliminates the foreground points.

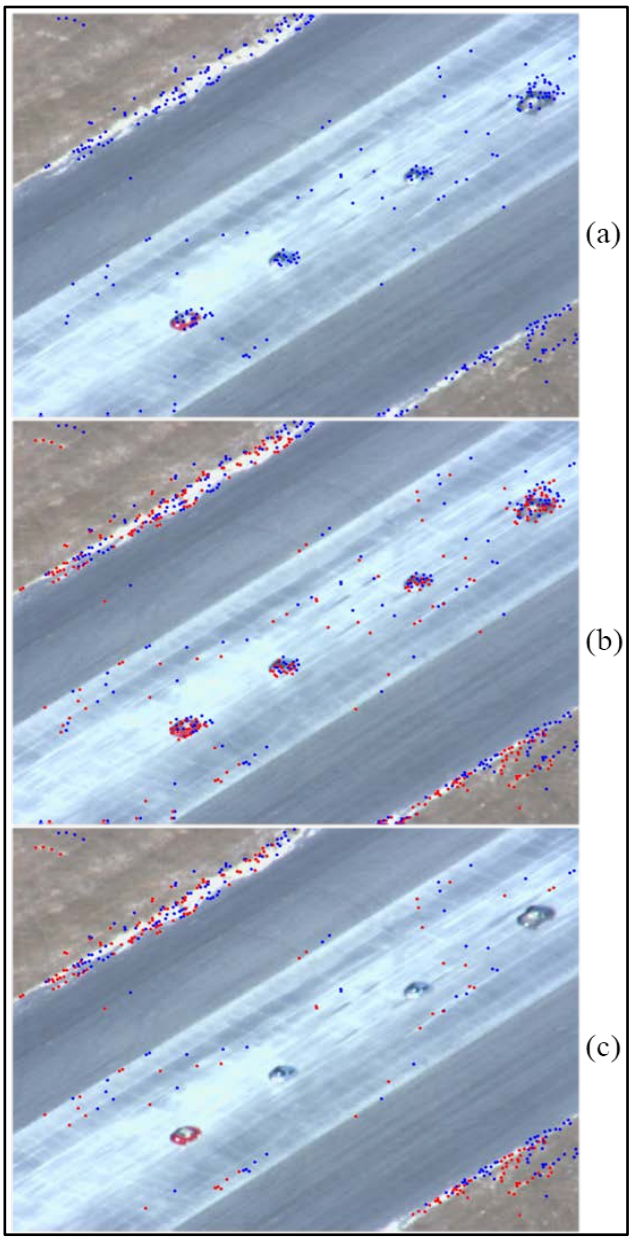


**Fig**.5 Harries Corner *HC* detection and estimation. (a) detecting *HC* in the whole image, (b) estimating *HC* in the next frame, (c) eliminating *HC* points belongs to the moving object.

Let $(x_i, y_i)$ denotes an $i^{th}$ *HC* point and $(y_i', y_i')$ for its correspondence one. The Euclidian distance *ED* between angle $\omega$ between them can be calculated using equation (8) and (9), respectively.

$$ED = \sqrt{\left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2} \quad (8)$$

$$\omega = \frac{y_i - y_i'}{x_i - x_i'} \quad (9)$$

After that, the k-means algorithm is applied on all *ED* and $\omega$ data to cluster them into two groups. The system assumes that the group containing the largest number of *HC* points is considered as background. The other group points are removed from the *HC* pair set as shown in Fig.5(c).

### 3.2.2 Ego-motion estimation

The ego-motion effect is resulted from the motion of the moving platform that holds the camera. So, the captured image contains moving background and moving targets. As illustrated above the proposed system applied k-means algorithm on all *HC* pair set and removed the targets points. So, the remaining *HC* points are representing the background only. After that the image frame is divided into sub-blocks. Each sub-block is assigned to all *HC* pair that located inside its borders as shown in Fig.6(a).

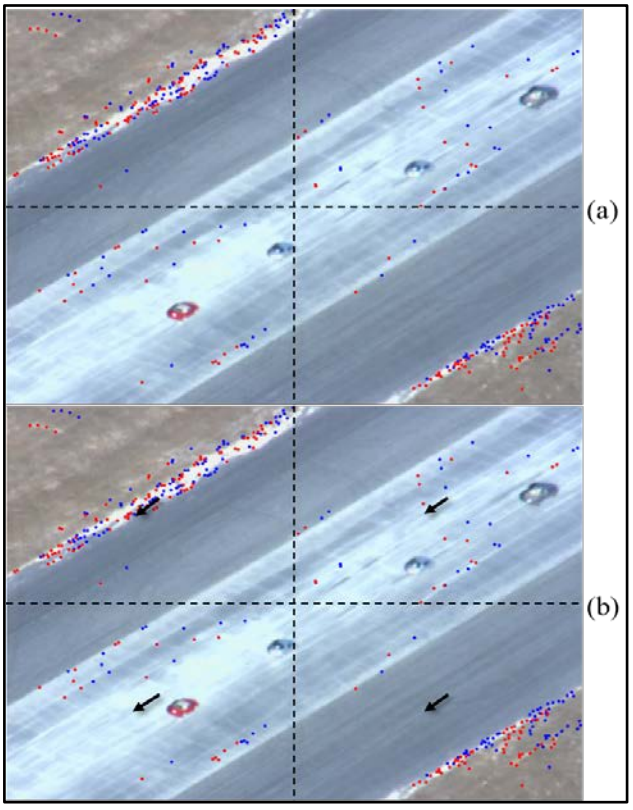


**Fig**.6 Harries Corner *HC* detection and estimation. (a) image sub-blocks, (b) estimating ego-motion magnitude and direction in each sub-block.

The affine geometric transformation is estimated for each block using MLESAC method as shown in Fig.6(b). MLESAC is a generalization of RANSAC estimator with a maximization of the likelihood rather than just the number of inliers [17]. Not all blocks have the same affine transformation matrix because of the possibilities of image rotation, skewing, and warping. So, the ego-motion transformation matrix *E* is calculated as the median of all affine transformation matrices that calculated for each block. It takes the form of 3x3 matrix which have 6 parameters as shown in equation (10).

$$E = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

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

The proposed system incorporates the ego-motion transformation matrix *E* with the auto-regression model to give optimized dynamic model. This model is applied on the *N* particles that represent all target candidates state to predict the state of the candidate targets in the next frame. Each candidate target can be represented using the state vector *S* that includes information about the target position, velocity and size as illustrated in the equation (11):

$$S = (x, y, \dot{x}, \dot{y}, w, h)^T \tag{11}$$

Where x and y are the left-top coordinate of the candidate rectangle and w and h are its width and height respectively. The target velocity information is represented by ẋ and ẏ components.

To reduce the calculations cost, the propose system uses the vectorization technique in all matrix manipulations instead of the traditional for-loop fashion. Thus, the predicted states for *N* particles is calculated using the following equations:

$$\boldsymbol{X_t} = \boldsymbol{X_{t-1}} + \dot{\boldsymbol{X}}_{\boldsymbol{t-1}} + \boldsymbol{Gx_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_x}) \tag{12}$$

$$\boldsymbol{Y_t} = \boldsymbol{Y_{t-1}} + \dot{\boldsymbol{Y}}_{\boldsymbol{t-1}} + \boldsymbol{Gy_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_y}) \tag{13}$$

$$\dot{\boldsymbol{X}}_{\boldsymbol{t}} = \dot{\boldsymbol{X}}_{\boldsymbol{t-1}} + \boldsymbol{G\dot{x}_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_{\dot{x}}}) \tag{14}$$

$$\dot{\boldsymbol{Y}}_{\boldsymbol{t}} = \dot{\boldsymbol{Y}}_{\boldsymbol{t-1}} + \boldsymbol{G\dot{y}_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_{\dot{y}}}) \tag{15}$$

$$\boldsymbol{W_t} = \boldsymbol{W_{t-1}} + \boldsymbol{Gw_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_w}) \tag{16}$$

$$\boldsymbol{H_t} = \boldsymbol{H_{t-1}} + \boldsymbol{Gh_{t-1}}(\mathbf{0}, \boldsymbol{\sigma_h}) \tag{17}$$

where $\boldsymbol{X_t}$, $\boldsymbol{Y_t}$, $\dot{\boldsymbol{X}}_{\boldsymbol{t}}$, $\dot{\boldsymbol{Y}}_{\boldsymbol{t}}$, $\boldsymbol{W_t}$ and $\boldsymbol{H_t}$ predicted state vectors for all *N* particles in the current time *t*. $G_*(*)$ 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 $G_*(*)$.

The proposed system takes only the first two rows in *E* and updates only the two position components in the state vector **X** (i.e. $x_t$ and $y_t$) as the following equation:

$$\begin{bmatrix} \boldsymbol{Xe_t} \\ \boldsymbol{Ye_t} \end{bmatrix}_{2\times N} = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}_{2\times 3} \times \begin{bmatrix} \boldsymbol{X_t} \\ \boldsymbol{Y_t} \\ \mathbf{1} \end{bmatrix}_{3\times N} \tag{18}$$

where $\boldsymbol{Xe_t}$ and $\boldsymbol{Ye_t}$ are the two updated position components of the target at frame *t*.

The components $a$ and $d$ in the ego-motion transformation matrix E replicate the scale change of the entire image due to the UAV altitude variations. The proposed system inspires these components to update the predicted size of the candidate targets. The following are used to calculate the new width and heights for all *N* particles states:

$$\boldsymbol{We_t} = \boldsymbol{W_t} * a \tag{19}$$

$$\boldsymbol{He_t} = \boldsymbol{H_t} * d \tag{20}$$

where $\boldsymbol{We_t}$ and $\boldsymbol{He_t}$ are the updated width and heights components of the target at frame *t*. The components $a$ and $d$ signifies the decreasing or increasing the image scale and therefore the target size. Finally, the predicted state of the particles can be expressed by the following equation:

$$\boldsymbol{S_t} = \left[\boldsymbol{Xe_t}, \boldsymbol{Ye_t}, \dot{\boldsymbol{X}}_{\boldsymbol{t}}, \dot{\boldsymbol{Y}}_{\boldsymbol{t}}, \boldsymbol{We_t}, \boldsymbol{He_t}\right] \tag{21}$$

## 3.3 Correction Phase

The correction phase consists of three main steps. First, *HR* and *EOH* features are extracted for all *N* particles by the same way as discussed before but using the vectorization method. Then the $N \times T$ matrix are created which represents the target observation features as shown in equation (22).

$$O = \begin{bmatrix} f_1^1 & f_2^1 & \dots & f_T^1 \\ f_1^2 & f_2^2 & \dots & f_T^2 \\ & & \dots & \\ f_1^N & f_2^N & \dots & f_T^N \end{bmatrix} \tag{22}$$

These observations are compared with the target template by computing the error between them using equation (23).

$$err = O - O_T \quad (23)$$

Thanks to the vectorization technique, all errors are calculated for all particles and all features in one step with high performance. Then the system calculates the likelihood $L$ of the particles based on their errors using the equation (24).

$$L = \frac{1}{\sqrt{2\pi\alpha^2}} e^{\left(-\frac{err}{2\alpha}\right)^2} \quad (24)$$

where $\alpha$ is a tuneable parameter. The exponential function is used to boost the lower error estimates to be compared with higher error values.

Initially, all the particles are given the same weights. Then they are re-weighted according to their likelihood $L$. based on the new weights, the particles are resembled to reject the particles having very low weight and to concentrate on the large weighted ones.

### 3.4 Post-Tracking Phase

The new state of the target is evaluated as the median of all candidate states represented by the re-sampled particles. The median filter is to eliminate any particles may have status positions relatively far from the object center. It achieves good results better than average filter due to its robust resistance to the noise.

Based on the estimated new position of the target, the new search area is determined for further tracking in the next frame. Finally, all the above procedures are repeated every $n^{th}$ frame.

## 4 Experimental Results and Analysis

To evaluate the proposed system, Numerous experiments are accomplished on VIVID database. All algorithms are implemented in MATLAB and executed on 2.6 G Hz processor and 4 GB RAM. The proposed system achieved 95.2% for successful target tracking. It can process more than 57 frames per second using 2000 particles which indicates more improved results. As illustrated in Table 1 both speed and accuracy are registered in comparison between the proposed system and tracking method proposed in [8].

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

| Tracking Method | Speed (fps) | Accuracy % |
|---|---|---|
| Proposed tracking system | 57 | 95.4 |
| Classical PF tracking framework | 13.1 | 95.1 |

### 4.1 Database

VIVID database is used to evaluate the proposed system with different scenarios [18]. It consists of numerous image sequences captured via UAV camera for different scenes (runways, roads, desert, forests, …). The images have sizes 640×480 pixels which provide good information details about target to be tracked. To simulate more challenging environments that usually happened during UAV flight, some modifications are performed on the captured image frame. The UAV vibrations or its camera shacking cause changing in image position, rotation and scale. This can be simulated by applying affine transformation with small random values on the new frames.

Generally, the VIVID includes civilian and/or military vehicles moving on a road and/or runway. The moving vehicles have various types of motions, varying speed and similar shape and color. Also, the scale of them are changing in the image due to the change of UAV altitude. The VIVID database offers a ground truth for certain vehicle, every 10 frame of its sequence, to compare with the resulted target position and size. To evaluate the proposed system the recall and precision are calculated by the same way as in [8].

### 4.2 Analysis and Discussion

In pre-tracking phase, the proposed system generated 20 random states around selected vehicle and 200 random states as backgrounds. After that, the extended *HR* and *EOH* features are extracted to provide 127 pool of features as the following:

- Normal *HR* features (1 - 105)
- *EOH* features (106 - 121)
- Oriented *HR* features (122 – 127)

Fig.7 illustrates the histogram of the best 32 features over 10 running times of the AdaBoost classifier. In Fig.7(a) tracked vehicle and background color are semi-close to each other. Conversely, the vehicle has distinguishable oriented edges. As a result, the

classifier designated *EOH* and oriented *HR* features more than *HR* ones as shown in Fig.7(b). As demonstrated in Fig.7(c), the vehicle color, edges and orientation edges can be distinguished easily. So, the classifier selected the best features from all of them approximately by the same degree as shown in Fig 7(d).

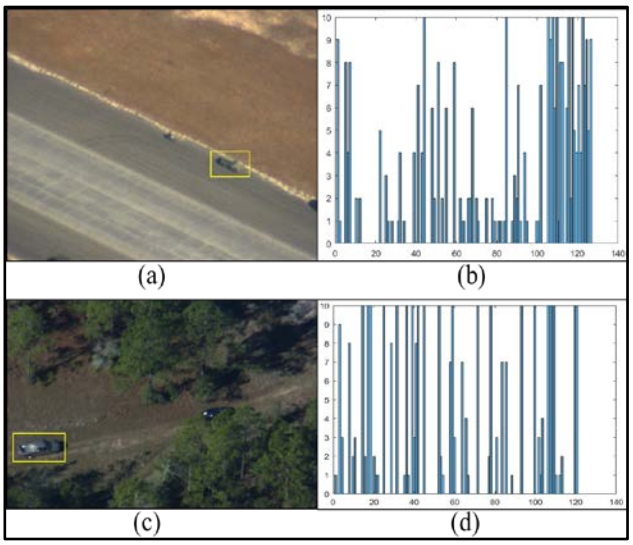


**Fig**.7 Selecting best features using Gentle AdaBoost Classifier. (a) selected target in "egtest03" scene, (b) best features histogram of (a), (c) selected target in "egtest05" scene, (d) best features histogram of (c).

The AdaBoost classification process consume more than 52 seconds to select the best 32 features, as an example, using the for-loop fashion. Thanks to the vectorization technique, the time elapsed for the same number of features reduced to less than 2 seconds. This time is needed only at the beginning of the system and not affect the tracking time. However, the number of features used to represent the target affects the tracking system speed and accuracy. The higher number of features, the higher accuracy and lower speed as shown in Fig.8 and Fig.9.

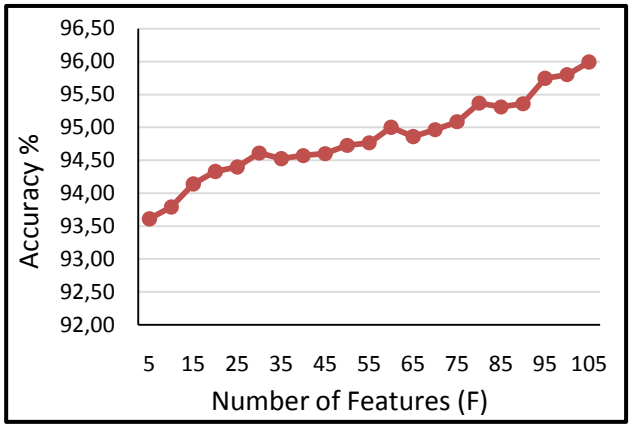


**Fig8**. Accuracy of the proposed system for different number of features.

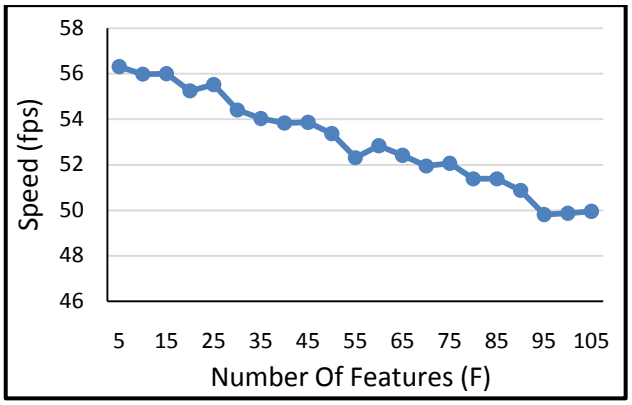


**Fig**.9 Speed of the proposed system for different number of features.

In ego-motion transformation step, the system tried homogenously to detect the strongest *HC* points over the entire image. Then, the image is divided into 2× 2 sub-block. All *HC* pairs assigned to a sub-block used to calculate its ego-motion transformation matrices. The number of *HC* points was affecting by the background smoothness. When the image background contained large smooth area, the number of required *HC* points increased to cover all image areas.

As mentioned before, the main purpose for calculating ego-motion transformation matrix *E* is to incorporate it with the target dynamic model. To prove that, the proposed system was tested 10 times on different scenarios in image sequence "egtest01". The recall and precision are calculated with and without the optimized dynamic model. At the beginning, the dynamic model works good but after frame #900 the vehicle accelerates making the particles lag its center. This means that the dynamic model does not guide the particles as better as before which reduces the accuracy as shown in Fig.10 (red curve). When system used the optimized dynamic model, the particles directed to the vehicle center which improved the accuracy (green curve).

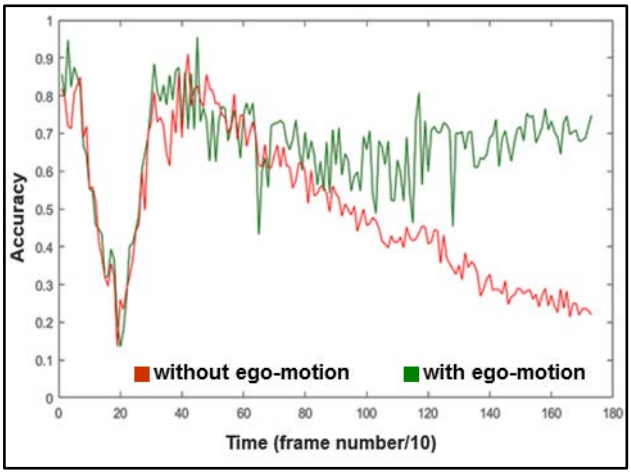


**Fig**. 10. The proposed tracking system accuracy.

After frame #1300, the UAV altitude started decreasing. This affected the image scale that started increasing. Accordingly, the vehicle size increased with the same ration as the image. Regarding this effect produced particles with states far from the real target candidate states in width and height. Therefore, the recall accuracy registered fewer values. So, the total accuracy reduced despite of the good precision values as shown in Fig. 11.

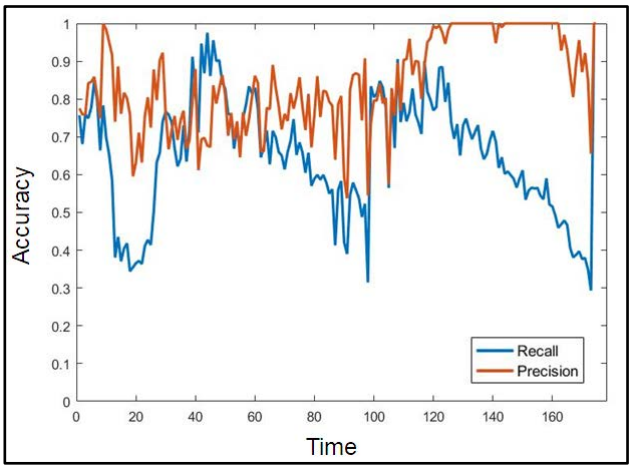


Fig.11 Recall and precision without using optimizes dynamic model.

As discussed before, the proposed system updated the target dynamic model using equations (19) and (20) to compensate the background scale change. This corrected the width and height of particles state. Hence, the recall and precision got higher values as shown in Fig. 12.

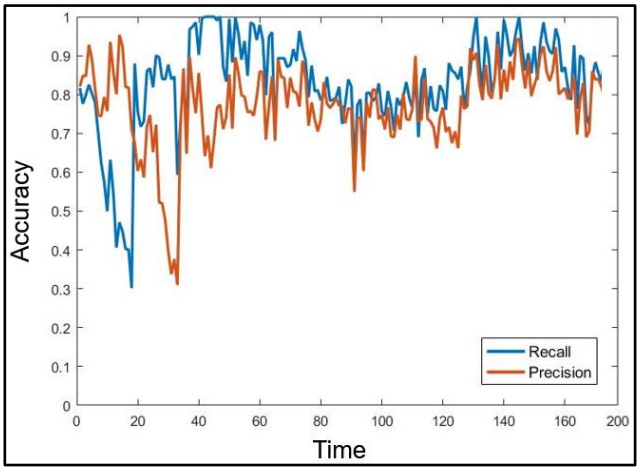


Fig.12 Recall and precision with optimizes dynamic model.

## 5 Conclusion

This paper proposed a real-time target tracking system based on UAV images. The optical flow and particle filter techniques were integrated together to build the optimized tracking system. The Lucas-Kanade and MLESAC algorithms were used to calculate the ego-motion transformation matrix. Then the system incorporated this matrix with the target dynamic model for the particle filter prediction step. The extended *HR* and *EOH* features were collaborated to build the particle filter observation model.

The incorporation of ego-motion transformation matrix with the first order auto-regression model produced an optimized target dynamic model. It compensated the effect of moving background and altitude changes on the target position inside the image. So, it can guide the particles towards the new estimated positions with better accuracy.

Thanks to Gentle AdaBoost classifier, only the best features conveying more information about the target than others were selected. This reduces number of features that required to achieve desired accuracy. So, the performance and speed of the tracking system were increased.

The use of integral image approach in features extractions had a great effect in the system speed and implementation simplicity. It reduced the calculation cost for any summation process to only four memory access. Also, the use of vectorization technique in matrix operations for large number of particles reduced the memory overhead and enhanced the tracking speed. The adaptive search area yielded the system to focus its calculations within only the effective area near the target. So, a great reduction in calculations time was achieved for single target tracking.

In the future, the proposed system will be implemented on an embedded system for onboard target tracking system.

*References:*

[1] Yilmaz, Alper, Omar Javed, and Mubarak Shah, "Object Tracking: A Survey," *Acm computing surveys (CSUR),* vol. 38, no. 4, p. 13, 2006.

[2] Kanellakis, Christoforos, and George Nikolakopoulos, "Survey on Computer Vision for UAVs: Current Developments and Trends," *Journal of Intelligent & Robotic Systems,* pp. 1-28, 2017.

[3] Saif, AFM Saifuddin, Anton Satria Prabuwono, Zainal Rasyid Mahayuddin, and Hendri T. Himawan, "A Review of Machine Vision based on Moving Objects Object Detection from UAV Aerial Images," *International Journal of Advancements in Computing Technology,* vol.

5, no. 15, p. 57, 2013.

[4] Burdziakowski, Paweł, Marek Przyborski, A. Janowski, and J. Szulwic, "A Vision-Based Unmanned Aerial Vehicle Navigation Method," *1st International Conference on Innovative Research and Maritime Applications of Space Technology,* 2015.

[5] Yang, Hanxuan, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song., "Recent advances and trends in visual tracking: A review," *Neurocomputing,* vol. 74, no. 18, p. 3823 – 3831, 2011.

[6] Sivaraman, Sayanan, and Mohan Manubhai Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE transactions on intelligent transportation systems,* vol. 14, no. 4, pp. 1773-1795, 2013.

[7] Coşkun, Musab, and Sencer Ünal, "Implementation of Tracking of a Moving Object Based on Camshift Approach with a UAV," *Procedia Technology,* vol. 22, pp. 556-561, 2016.

[8] Cao, Xianbin, Changcheng Gao, Jinhe Lan, Yuan Yuan, and Pingkun Yan, "Ego-motion guided particle filter for vehicle tracking in airborne videos," *Neurocomputing,* vol. 124, p. 168–177, 2014.

[9] Sefidgari, Bahram Lavi, "Feed-Back Method Based on Image Processing for Detecting Human Body Via Flying Robot," *International Journal of Artificial Intelligence & Applications,* vol. 4, no. 6, p. 35, 2013.

[10] Gerónimo, David, A. Sappa, Antonio López, and Daniel Ponsa, "Adaptive image sampling and windows classification for on-board pedestrian detection," *Proceedings of the International Conference on Computer Vision Systems,* vol. 39, 2007.

[11] Joshi, Mukul, Rajashri Madri, and Madhuri Joshi, "Real time motion tracking algorithm for search and rescue robots," *TENCON 2013-2013 IEEE Region 10 Conference (31194),* pp. 1-4, 2013.

[12] Abdelwahab, Mohamed A., and Moataz M. Abdelwahab, "A Novel Algorithm for Vehicle Detection and Tracking in Airborne Videos," *Multimedia (ISM), 2015 IEEE International Symposium on,* pp. 65-68, 2015.

[13] Motai, Yuichi, Sumit Kumar Jha, and Daniel Kruse, "Human tracking from a mobile agent: Optical flow and Kalman filter arbitration," *Signal Processing: Image Communication,* vol. 27, no. 1, pp. 83-95, 2012.

[14] Shantaiya, Sanjivani, Kesari Verma, and Kamal Mehta, "Multiple Object Tracking Using Kalman Filter And Optical Flow," *European Journal of Advances in Engineering and Technology,* vol. 2, no. 2, pp. 34-39, 2015.

[15] Viola, Paul, and Michael Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on,* vol. 1, pp. I-I, 2001.

[16] Shi, Jianbo, "Good Features to Track," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on,* pp. 593-600, 1994.

[17] Torr, Philip HS, and Andrew Zisserman, "MLESAC: A New Robust Estimator with Application to estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, no. 1, pp. 138-156, 2000.

[18] Collins, Robert, Xuhui Zhou, and Seng Keat Teh, "An Open Source Tracking Testbed and Evaluation Web Site," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,* vol. 35, 2005.