Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 8 (7): 1159-1168, 2008
ISSN 1812-5654

Increasing Accuracy of Tracking Loop for the Rosette Scanning Seeker Using Improved ISODATA and Intelligent Center of Gravity

Hadi Soltanizadeh and Shahriar Baradaran Shokouhi
Department of Electrical Engineering, Iran University of Science and Technology,
Narmak, 16846, Tehran, Iran

Abstract: Rosette scan infrared seeker offers the imaging information of target to the processing unit. In the processing unit of the missile, all of the received samples are clustered, classified and then the center of each class is determined. The conventional clustering techniques on rosette pattern are unable to classify all samples correctly. A new clustering method, with improved classification ability, is proposed in this research. Also, a new technique to compute the centroid of each class is introduced. The method is robust against the variation of class radius and more precise in comparison with previous methods. Exploiting the proposed clustering method and features, real target is discriminated from flares and missile tracks the target in a correct trajectory. Also an effective simulator is used to evaluate the performance of rosette scanning seekers, decrease the cost of design and construction. Using this simulation, we can test the system in several cases with different algorithms. Hence the results can be readily verified by simulations within this framework. In this study, the rosette pattern along with the target and flares is simulated and then the clustering and classification methods including K-Mean's and ISODATA are used to distinguish the target. It is revealed that the tracking loop presents the best results when using the proposed improved ISODATA method to detect the target and Error Back Propagation (EBP) learning method for computing centroids.

Key words: Simulator, rosette pattern seeker, classification, central gravity, tracking loop

INTRODUCTION

A Rosette Scanning Infrared Seeker (RSIS) is the device mounted on the infrared guided missile. It offers the positions and images of target to missile's servo system by scanning a Total Field of View (TFOV) in a rosette pattern with a single detector. An Instantaneous Field of View (IFOV) is a diameter of the detector moving along the path of the rosette pattern. The IFOV has the property that its smaller size provides the less interference of background signals and detector noise.

The rosette pattern of the RSIS can be achieved by means of two counter-rotating optical elements such as prisms, tilted mirrors or off-centered lenses (Jahng *et al*., 1998). It offers the imaging information of target to the processing unit. Planes keep themselves safe against the thermal tracking missiles by discharging flares. In the processing unit of the missile, all of the received samples are clustered, classified and the center of each class calculated.

In order to distinguish the target from false targets, the previous studies based on image processing algorithms such as K-Mean Algorithm (KMA), Iterative Self Organizing Data Analysis Technique (ISODATA) and Moment Techniques were proposed.

In the moment technique (Jahng *et al*., 1999a), the target is distinguished from the flare by setting the detection threshold equal to the average intensity of the previously detected target signal. The flare may have a similar intensity level to the target signal, since its intensity varies with time. So the RSIS cannot distinguish the target from the flare. In the KMA (Jahng *et al*., 1999b), the pixels of the detected image are divided into two classes: the target and the flare. Then the RSIS tracks only the centroid of the target class. The clustering result that the KMA generates depends on the seed point of an initial class. Furthermore the number of clusters must be determined prior to the initialization. Thus, if a target discharges a random number of flares, the multiple flares maybe recognized as a single one, therefore the RSIS fails to track the target. In the ISODATA algorithm, unlike the KMA, the number of classes is not fixed (Jahng *et al*., 2000). Since RSIS does not know how many classes are in the TFOV, the ISODATA nominates any of the detected pixels as an initial class. These classes are merged and split through ISODATA algorithm. Because of the large

Corresponding Author: Hadi Soltanizadeh, Department of Electrical Engineering, Iran University of Science and Technology, Narmak, 16846, Tehran, Iran

number of initial classes, the relevant technique has a considerable processing time and requires parameters` modification during the tracking procedure.

To solve the mentioned problems of the conventional clustering methods, we propose to define the continuous data on each rosette cluster as an initial class. The initial classes are merged if they are too close. Iterating the lumping process, the proposed algorithm can classify all of the pixels in the TFOV without the help of the seed points, split of parameters and the number of cluster centroids. The convergence of the algorithm is fast.

In order to determine the central gravity of target, the mean and weighted mean can be applied. Also two types of weighted mean are Distribution Function (DF) and Error Back Propagation Learning methods. At last the tracking loop of the RSIS is simulated in a real time mode and results will be evaluated. The tracking loop presents the best results when using the proposed ISODATA method (Jahng *et al*., 1999a) to detect the target and EBP learning algorithm in computing of center of gravity.

In order to evaluate the performance of IR tracking seekers, usually Hardware in the Loop Simulation (HILS) is used (Pooly and Collin, 2001). The great interest for HILS makes it important to gather knowledge about its field of application and its limitations. A general step in this process is designing a model that can be used for computer simulation. This study describes an effective HILS model. The model is implemented as a part of a real-time framework along with numerous rosette pattern types, target shapes, countermeasures, reconstruction image, clustering and classification. We also simulate the tracking loop in the rosette scanning infrared seeker. In addition to decrease in cost, flexibility is one of its outstanding features and it can be experimented in several cases with different algorithms.

In this study we introduce a user friendly graphical environment for introducing targets and flares with different shapes and sizes. Then we can have IR signals and reconstructed images. Also by applying mentioned clustering methods in the HILS, we can evaluate the accuracy and processing time of each method.

GENERAL PROPERTIES OF ROSETTE PATTERN

The rosette pattern of RSIS is formed by two optical elements rotating in opposite directions. If rotational frequencies for two optical elements are f_1 and f_2, the loci of the rosette pattern at an arbitrary time t, in the Cartesian coordinates can be expressed with the Eq. 1 (Jahng *et al*., 2000).

$$x(t) = \frac{\delta}{2}(\cos 2\pi f_1 t + \cos 2\pi f_2 t)$$
$$y(t) = \frac{\delta}{2}(\sin 2\pi f_1 t - \sin 2\pi f_2 t) \quad (1)$$

where, δ is the refractive index of the prism.

The values of rotating elements which spinning with frequencies f_1 and f_2 determine the rosette pattern parameters such as the scan speed, total number of petals and the petal width. If f_2/f_1 is a rational number and f_1 and f_2 have the greatest common divisor f such that $N_1 = f_1/f$ and $N_2 = f_2/f$ are both positive integers, the pattern will be closed. Moreover N_1 and N_2 are the smallest integers satisfying.

All the images used in the evaluation of the algorithms were synthetically generated. The minimum size required for the background image to include the total rosette pattern is 400 pixels. Therefore, target image resolutions of 400*400 pixels were used with the homogeneous background. The target image can be selected with different shapes and sizes where the shapes of the flares are defined sphere.

The equations of rosette pattern in the polar system are as follows:

$$\rho_t = \mathrm{Cos}\left(\pi(f_1 + f_2)\, t/T_t\right)$$
$$\theta_t = \pi(f_1 - f_2)\, t/T_t \quad (2)$$

It can be rewritten in Cartesian form,

$$x_t = \rho\mathrm{Cos}(\theta_t), \quad y_t = \rho\mathrm{Sin}(\theta_t) \quad (3)$$

To map the rosette pattern in background image we have:

$$X(t) = (L + L \times x_t)$$
$$Y(t) = (L - L \times y_t)$$
$$L = D/2 \quad (4)$$

where, L is the half of the background image height and is equal to 200 pixels. Figure 1 shows the rosette patterns with different parameters. Figure 1a demonstrates a simple rosette pattern without overlapping of the petals when the parameters are $f_1 = 350$ and $f_2 = 250$ or $N_1 = 7$ and $N_2 = 5$. With this pattern only a fixed area of FOV can be scanned and the resolution is related to the number of petals. The processing algorithms for target detection of the simple patterns were proposed previously by Jahng *et al.* (2000).

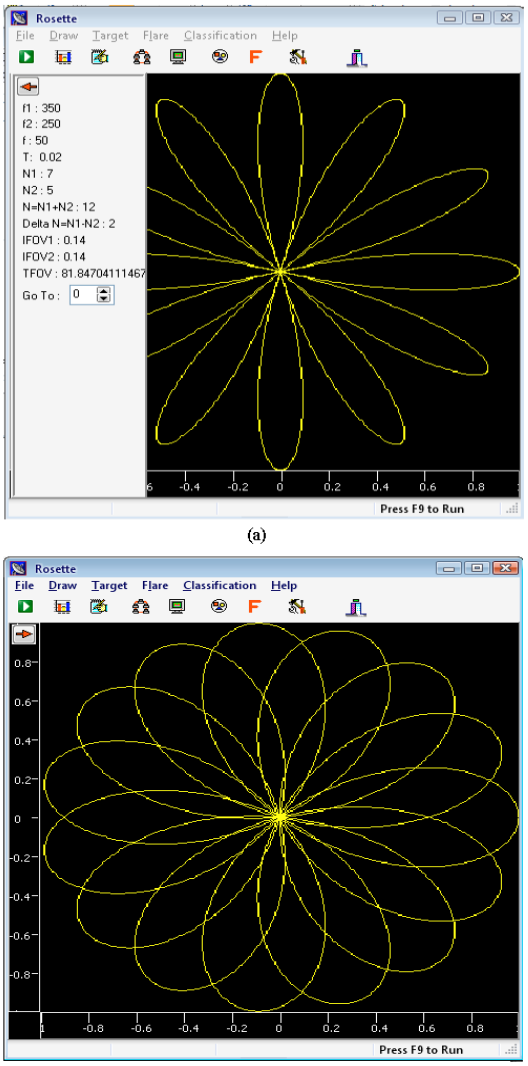


Fig. 1: The graphical environment for generating rosette pattern and all the processing algorithms. The generated rosette pattern in the background image for a) a simple rosette pattern without overlapping of the petals with $f_1 = 350$ and $f_2 = 250$ or $N_1 = 7$ and $N_2 = 5$, b) an overlapped rosette pattern with $f_1 = 275$ and $f_2 = 100$ or $N_1 = 11$, $N_2 = 4$

In this research, an overlapped rosette pattern is chosen with the parameters of: $f_1 = 275$ and $f_2 = 100$ or $N_1 = 11$, $N_2 = 4$ as shown in Fig. 1b. By choosing an overlapped pattern, we can increase the resolution of scanning area of FOV and also decrease blind spots of the pattern. But we have to spend more for complexity of the reconstruction and classification algorithms.

In the designed simulator we can enter values for N_1 and N_2 without any restriction to have overlapped or non-overlapped pattern. The simulator is able to process all the tasks for the entered pattern successfully. In the presented study, we have selected the overlapped pattern with the mentioned parameters because of its complexity and will show that the results of processing will be reasonable.

T is considered as the rosette scanning period time which is calculated by:

$$T = \frac{1}{F} = \frac{N_1}{f_1} = \frac{N_2}{f_2} \quad (5)$$

for $f_1 = 275$ and $f_2 = 100$, we have $F = 25$ and T is equal 0.04 sec.

Finally obtained IR signal is sampled with F_s and will be sent to the Processing unit. In this approach sampling frequency is considered 100 KHz. Therefore we have 4000 samples for each rosette scanning period.

TARGET AND FLARES

A background image is considered for the main environment so the simulator can show the target and flares with enough resolution and intensity for processing. The user can draw targets and flares with different shapes and sizes in the background image when they are in the TFOV. The minimum size of target is IFOV when missile is ready to launch and it has maximum size when missile is close to the target. Flare radiates non-constantly during burning 0. The intensity of the target varies between 11-32% of the flare intensity as shown in Fig. 2. Hence, if we consider the maximum intensity of flare equal to 255, the target intensity will be 80. Each flare burns completely about 3.5 sec. The flare intensity varies with time according to the Eq. 6.

$$I(t) = -0.0024t^7 + 0.0086t^6 + 0.1260t^5 - 1.0314t^4 + 3.1351t^3 - 4.7867t^2 + 3.5501t - 0.0004, 0 \le t \le 3.5(s) \quad (6)$$

where, I(t) is the IR intensity of flare.

A target and some flares are shown in Fig. 2. (a) Target is considered as a circle with IFOV area around it. Flares are usually smaller than a real target (1/10) and are shown. (b) IR signal pulses, which is generated by target and flares. (c) Reconstructed Image of target and flares in TFOV by using generated IR signal.

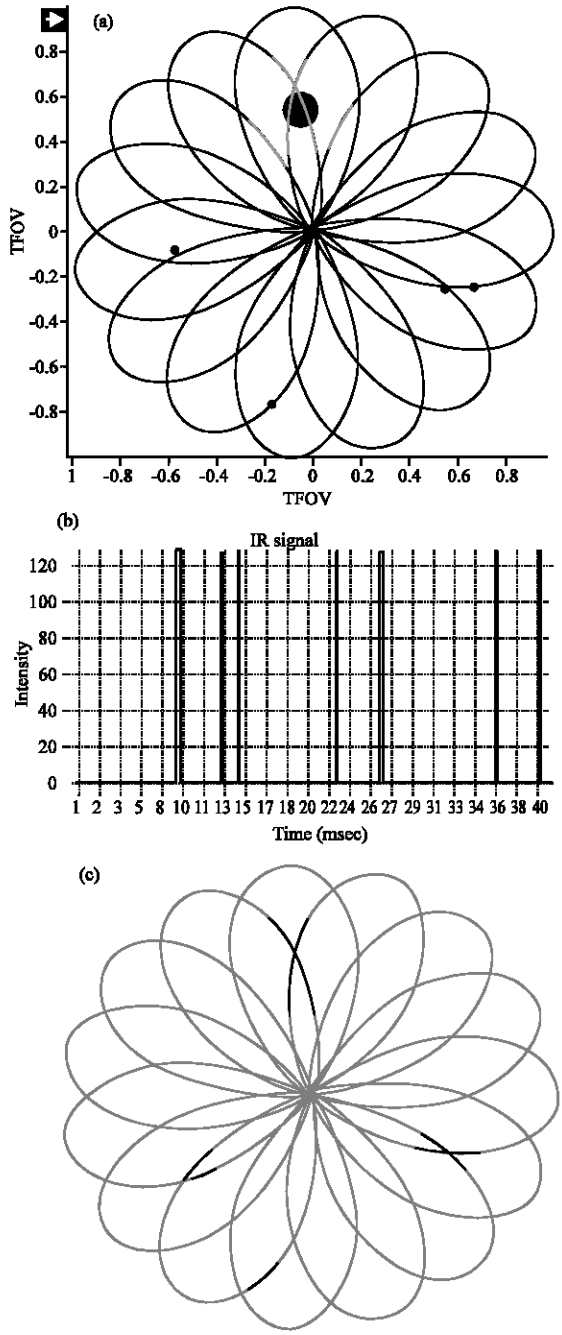


Fig. 2: (a) Target and three groups of flares in the rosette pattern, (b) IR pulses and (c) reconstructed images of target and flares

CLASSIFICATION

Different methods of classification have been proposed for rosette patterns (Jahng *et al.*, 1999b). Classification is applied to discriminate target from flares. Classifiers are ranked with their requisite processing time. If they cannot discriminate the target and flares in a real processing time, missile will miss the target. We will discuss the general methods for clustering and classification of objects in the rosette pattern, also our proposed method, named as Improved ISODATA will be presented (Shokouhi *et al.*, 2005).

Moment: In this method, classification is based on the radiation density. By considering this fact that the radiation amplitude of flare is greater than the target's, we can compare the signal levels. Signals with amplitudes greater than the target signal level are ignored. This method cannot detect the target correctly when the flare intensity is equal to the target and missile will be perverted.

K-Mean: In this method all of samples in the reconstructed image are classified into N classes: target and flares. Then central gravity of each class will be calculated. As we know target size is greater than flares, so the target can be distinguished.

The value of N (number of classes in the reconstructed image) should be determined initially. This method fails to perform well if the number of flares varies or the number of defined classes is different from the source number.

Improved ISODATA: This method exploits ISODATA algorithm to cluster samples in the reconstructed image. Clustering is based on minimizing the squared error. Error is the sum of squared distances between samples and central gravity of each class. Therefore clustering pixels of target and flares in the reconstructed image depends on some parameters like the standard deviation of samples from the central gravity, the distance between centers and number of classes.

In the ISODATA algorithm, the value of K (the number of clusters) is not defined by the user as KMA, rather K changes when the algorithm runs. At first, this algorithm, remove groups with low number of samples. Two clusters should be merged if the number of clusters is greater than K or if two clusters have close centers. Also a cluster should be split into two clusters if the number of clusters is smaller than K or the standard deviation of samples in a cluster is too high. There are two procedures for implementing of ISODATA algorithm:

- At first, all samples considered in one big cluster. Then split this big cluster into the clusters and each of new clusters splits to some other smaller clusters based on existing parameters. The splitting procedure would be repeated until any cluster couldn't be split.

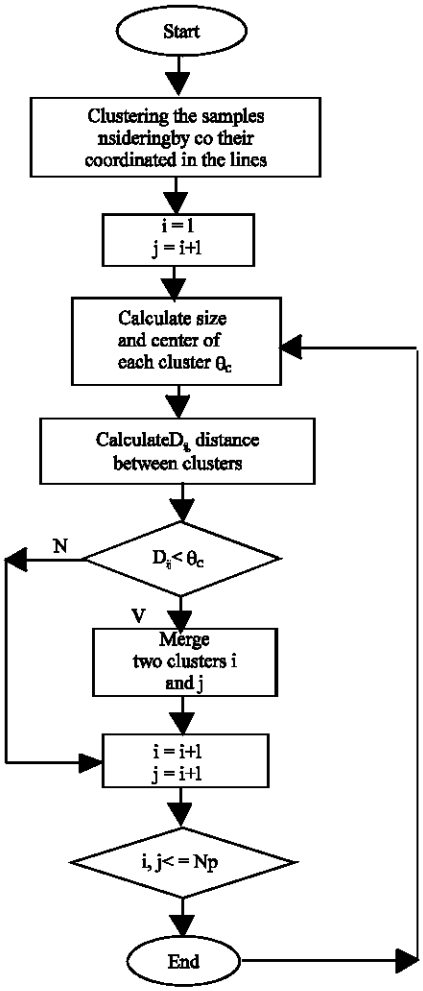


Fig. 3: The block diagram of the improved ISODATA clustering method

- At first, each sample is considered as a cluster. Then samples are grouped by merging. The processing time of this method is slower than the previous one.
- In the Improved ISODATA (Shokouhi *et al*., 2005), all samples lied in the same line, are considered as a single cluster (target or flares). Then new clusters are merged together with the proposed algorithm.

With this method, two classes will be merged if Euclidean distance between centers of classes is smaller than the size of each class. So improved ISODATA classifies data without any initial parameters therefore it is an autonomous algorithm. The block diagram of the proposed method is shown in Fig. 3.

In the block diagram θ_c is the size of cluster, D_{ij} is the Euclidean distance between clusters i and j and finally N_p is the number of clusters. In this diagram the distance between all the clusters must be checked and the algorithm will be stopped if there are no clusters which can be merged. A target and three groups of flares in Fig. 2a are clustered and four recognized clusters by employing the mentioned algorithm, one target and three flare groups, are depicted in Fig. 2c.

CENTRAL GRAVITY

An IR seeker sends information of target positioning to the missile processing unit. The IR rosette scanning seeker sweeps a small instantaneous field of view in the total field of view. Since the rosette pattern is nonlinear, the reconstructed images of objects in the rosette pattern depend on the object location in the pattern. Non-linearity of the pattern is the source of error in computing of target's central gravity. In the target tracking loop, system tracks the central gravity of the target; hence errors in computed central gravity leads to target lose. Therefore precise computation of central gravity is of utmost importance. After introducing some methods of computing central gravity, we propose a new method using error back propagation learning algorithm.

Mean method: In this simple method, the positions of samples in a period are saved. When sweeping the field of view ended, the mean value of all positions will be computed. Equation 7 calculates the mean value (i.e., the central gravity of target):

$$\hat{x} = \frac{\sum_{i=1}^{m} x_i}{m}, \quad \hat{y} = \frac{\sum_{i=1}^{m} y_i}{m} \tag{7}$$

Lines in the central parts of the rosette pattern are denser than the marginal parts. So the mean value of samples in target class is near to the center of the rosette pattern. As shown in Fig. 4, the real central gravity and computed central gravity for a sample target are different.

Weighted mean method: Density of lines which sweeping the target is not equal in all TFOV; it is greater in the center of the rosette pattern. As shown in the Mean method, the computed center is near to the center of rosette pattern, this makes the real and computed center a little different. In order to compensate this error, a weight would be assigned to each sample in the rosette pattern and weighted mean is to be computed as the target central gravity. Weights in this method are set by weight function or the EBP method.

Weighting with the distribution function: In this method, for each sample in rosette pattern, weight function will be

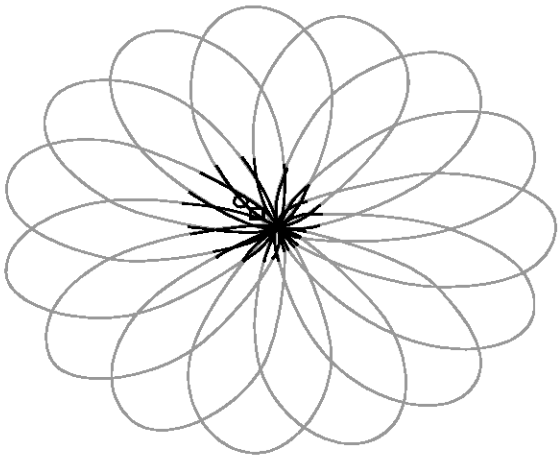


Fig. 4: The circle point shows real center and the square shows the computed central gravity of the target

set by using line density (Jahng *et al*., 2002). The weight function value would be greater where the line density is less. Equation 8 is used to compute central gravity of objects.

$$\hat{x} = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i}, \quad \hat{y} = \frac{\sum_{i=1}^{m} w_i y_i}{\sum_{i=1}^{m} w_i} \tag{8}$$

where, w_i and m are the value of ith element of weight matrix and length of weight matrix, respectively. Distribution function describes the line density in the rosette pattern. It is used to compute the weight function. At the first step, the distribution function of pixels' summation for a point target is computed according to the position of target in TFOV.

In the next step, the weight function is determined by inverting the values of distribution function. Dimension of the weight matrix is L×N, where N is half number of samples in a petal. Also the distance between two petals would be divided into L slices. Therefore equation of the kth line is:

$$y_k = \left(\frac{k}{L}\tan\left(\frac{2p}{N}\right)\right).x \tag{9}$$

Figure 5 shows partitioning two adjacent petals in a rosette pattern using the lines of Eq. 9.

To compute the distribution function, a target's point P(r, θ) moves along the lines of Eq. 8 with a defined step from r = 0 to r = 1, where, $(x^2+y^2)^{1/2}$ and $\theta = \tan^{-1}(y/x)$. For example if L=4 and r_{step} = 0.01, then the area between two adjacent petals of the rosette pattern is divided into 4×100 slices. To compute TNOP (Total number of points), we initialize the weight matrix and sweep IFOV through

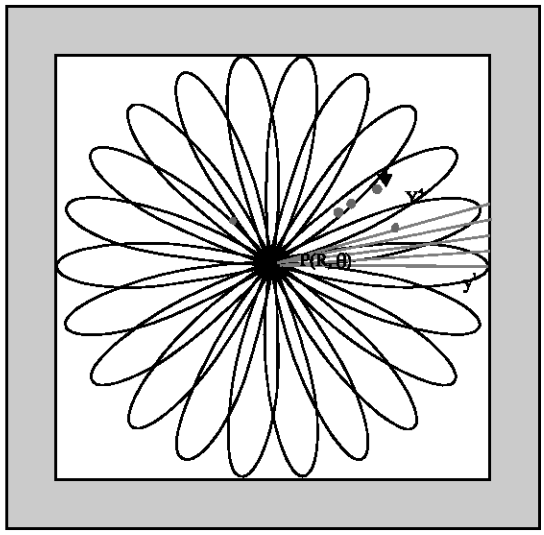


Fig. 5: Dividing a petal into four slices to compute the weight function

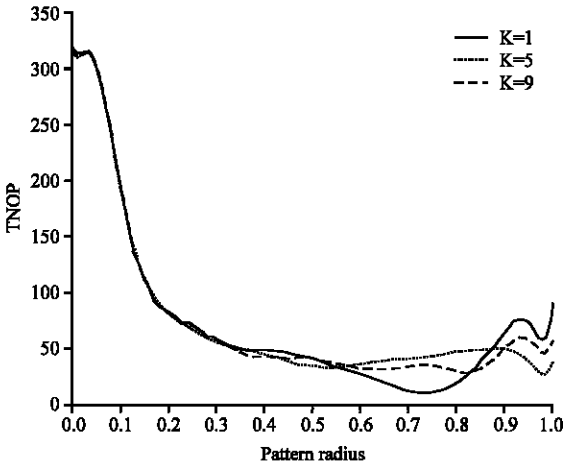


Fig. 6: Distribution function of TNOP estimated with the LMS[1] method

TFOV. When IFOV is going over each area, the value of TNOP for this area depends on the number of IFOV points. The whole TNOP matrix will be computed when seeker scans all the TFOV. Figure 6 shows TNOP for the rosette pattern with N_1 = 13, N_2 = 9, L = 10 and several values of k.

To compute the weight function, value of the distribution function for each point should be reversed. It is mentioned that the distribution and weighted functions have angular periodicity with the period of a petal angle. Therefore it is enough to compute weights in a period. Figure 7 shows the weight function computed by the distribution function.

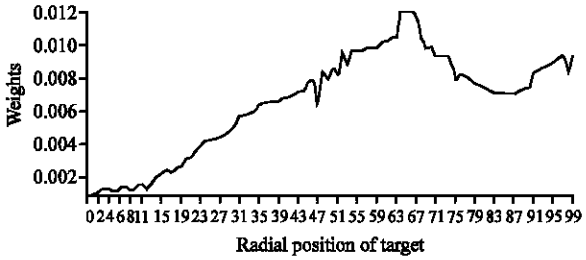


Fig. 7: Computed weight function for the rosette pattern

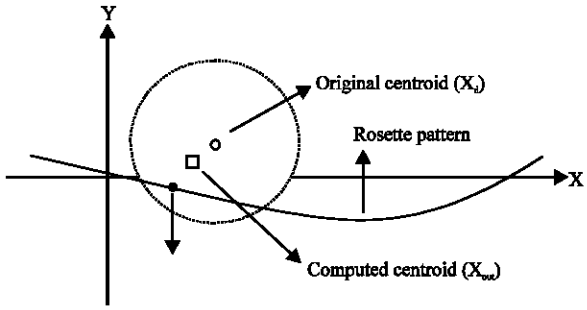


Fig. 8: Rosette pattern and the related parameters

Weighting with the EBP method: In this method, the area between two neighboring petals is divided into L radius parts and N angle directions. Therefore there are L´N points in the area for which a weight function is considered. For N = 10 and L = 100, the petal is divided into 1000 points. Figure 5 shows division of a petal into 4 angle directions.

At first, the weights are put in a 10x100 matrix. The wisely selected initial values help rapid convergence to the correct answer. In the training stage, a target would be placed on the rosette pattern and then its central gravity is computed employing Eq. 8. So computed center (x_{out}) would be compared with real target central gravity (x_d):

$$e_x = x_{out}-x_d \quad (10)$$

For simplicity the relations are considered only in one dimension. The weights should be changed to reduce the comparison error Eq. 11 in the next iteration.

If $w_x(n)$ is the weight for position x at stage n, the weight at stage n+1 is updated by:

$$w_x(n+1) = w_x(n)+\Delta w_x(n) \quad (11)$$

For both directions x and y changes to:

$$w(n+1) = w(n)+\Delta w_x+\Delta w_y \quad (12)$$

where, Δw_x is the adjustment term represented by:

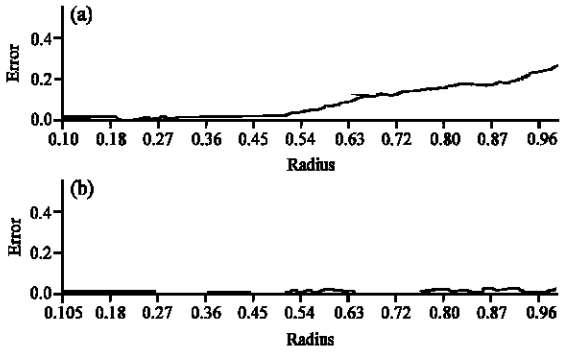


Fig. 9: Variation of error according to the variation of the target radius in (a) distribution function and (b) EBP methods

$$\Delta w_x(n) = \eta e_x (x-x_{out}) \quad (13)$$

where, η is the learning rate.

In the next step, center of class is set to the next point along the lines represented in Fig. 5 and the entire calculations are repeated. This process will be repeated until achieving an acceptable minimization of central gravity computational error. Target size should be different in all steps of the process from 0.1 to 1 [TFOV]. One iteration of the process is finished when the target size equals to 1 [TFOV]. To reduce calculation, we have exploited the periodicity property of the weights; they are computed only for the range between two neighboring petal's tips.

Figure 9 shows the computing error (Euclidean distance between real and computed central gravity) versus target radius in both methods; the distribution function method and the proposed method in an asymmetric pattern. As an example we put target center in position (0.1, 0.1) and changed target radius from 0.1 to 1 [TFOV]. Figure 10 shows the computational error rate for a target with radius 0.15[TFOV] in different positions from the rosette pattern center to the borders of TFOV using both mentioned methods.

The center of the class is set to the r axis and moves from r = 0 to r = 1 in steps of size 0.01 [TFOV]. When the class moves toward the petal's tip, the error increases because some parts of the class do not lie in the rosette pattern i.e., they are out of TFOV.

TRACKING LOOP

In simulation of the tracking loop, a target with 0.1[TFOV] radius would be used which diffused two flares with radius 0.02[TFOV]. Tracking takes about 4 sec and flares are diffused in 2 sec. All of the target and

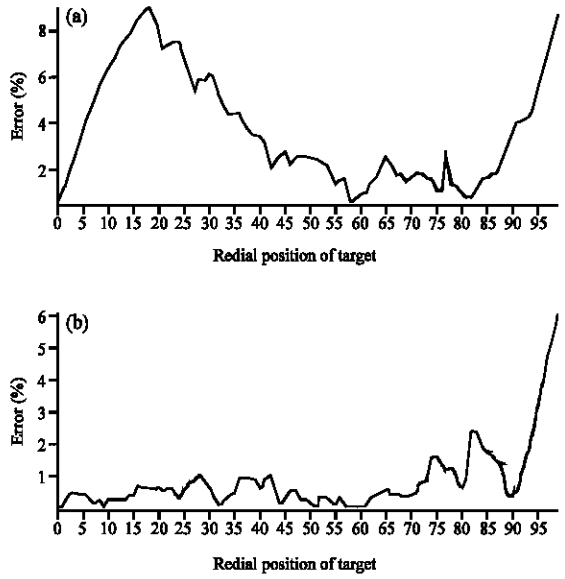


Fig. 10: Comparison of RMS error between (a) distribution function and (b) EBP methods

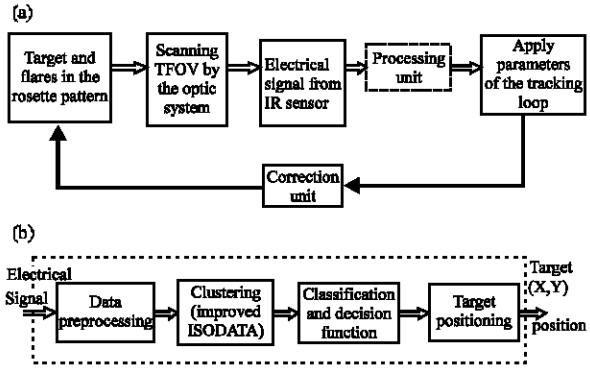


Fig. 11: (a) Schematic block diagram of the tracking loop, (b) stages involved for implementing the processing unit

flares are considered to have circular shape. Target speed is considered about 1[TFOV/Sec] and that of flares is about 0.1[TFOV/Sec].

General block diagram of the tracking loop system is shown in Fig. 11. As shown in this figure, target and flares simulation is accomplished. Then all of the TFOV is scanned by the rosette pattern, electrical signal pulses are produced relate to the received IR ray while IFOV is crossing over the target and flares. In the processing block, electrical IR signals are sampled, reconstructed and classified and eventually the target is discriminated from other classes. In the next block, the value of deviation for gyro, in order to set gyro in the direction of detected target central gravity would be computed. This value is delivered to the missile control unit.

Tracking error in each period is the distance between the rosette pattern centers and computed central gravity of the target class. In simulations, Eq. 13 is used to determine the requisite deviation for the next period.

$$\text{movement}(x,y)\text{-mean}_n(x,y) \quad (14)$$

In Eq. 14, the variables $\text{Center}_n(x,y)$, $\text{movement}(x,y)$, $\text{mean}_n(x,y)$, $\text{Center}_{n+1}(x,y)$ are the class center in the previous frame, the target movement speed in [TFOV/Period time], the computed center of the target class in the previous frame and the updated target center in the current frame, respectively.

RESULTS AND DISCUSSION

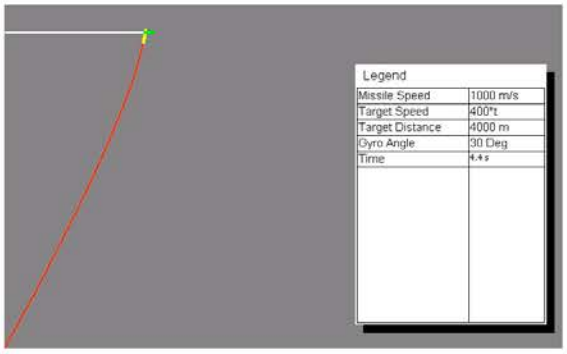
The evaluation of the algorithms was performed with a 2200 MHZ Pentium IV system with 512 MB of RAM. The image matrix size is chosen 400 by 400 and the sampling frequency is 100 KHz. Pattern with $f_1 = 275$, $f_2 = 100$ is just used in all simulation and test. We can apply targets and flares with different shapes and size. Also the number, launching time and speed of flares are variable parameters and should be selected by user.

In accomplished simulations, some parameters are considered as the inputs, such as: target speed, missile angle and target distance when tracking loop is applied. Figure 12 shows the results for different input parameters. The considered parameters are selected appropriately for Fig. 12a and b therefore the missile can track the target successfully.

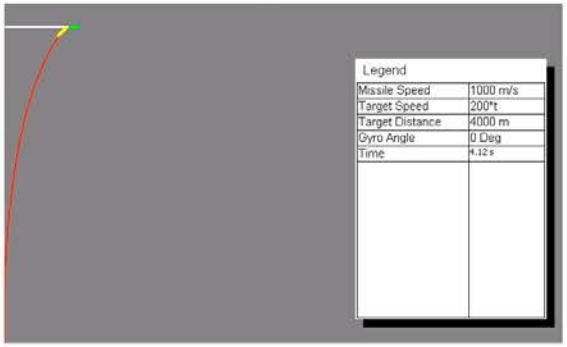
The simulation times achieved using two different algorithms for one circular target and different number of flares are shown in Table 1. All methods are tested in the same condition of the loci of flares and target. The proposed method, ISODATA based on clustering, has process time average about 30 msec compare of 0.7 sec for classic ISODATA.

Table 1: Comparing the processing times of the classical and proposed ISODATA

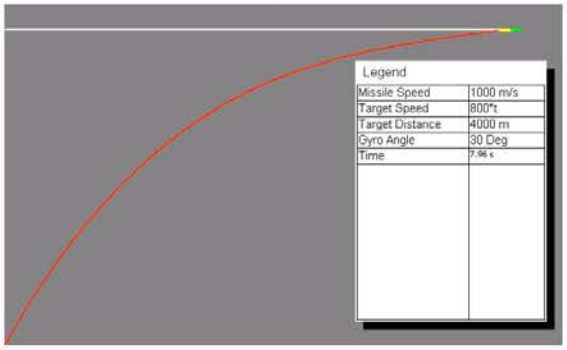
	Method	
Processing time	Classical ISODATA	Improved ISODATA
Data clustering	-	1 msec
Data classification	0.7 sec	24 msec
Calculation of central gravity	5.0 msec	5 msec
Total processing time	0.7 sec	30 msec



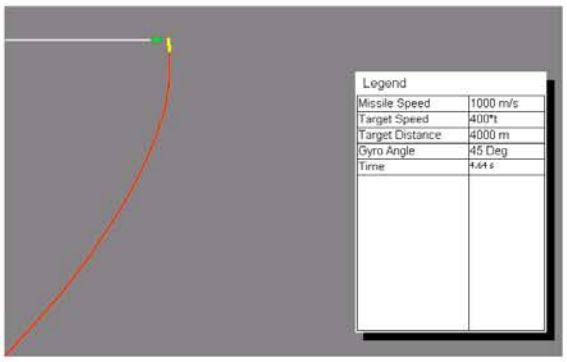
(a)



(b)



(c)



(d)

Fig. 12: The results of tracking for different parameters: target speed, target distance and missile angle, (the processing time table is embedded in the figures)

CONCLUSIONS

This study is described an effective simulator for the rosette pattern, target and flares, electrical signals, clustering and classification, center of gravity and finally tracking loop. For data clustering and classification the improved ISODATA method has been used. Central gravity of target is computed using an intelligent method which can consider shape and size of target. The computed central gravity is delivered to the tracking loop. We can put a target on the TFOV and all the setting parameters before starting the processing. In hardware in the loop simulation environment, where real time interaction between the seeker system and the image generation system is required, the transfer of the new target images to the simulation system might cause a bottleneck. In the tracking loop, the practical gyro parameters and latency have been considered to avoid from unreal simulation and also the generation of the next image to be used for the rosette simulation.

REFERENCES

Jahng, S.G., H.K. Hong, S.H. Han and J.S. Choi, 1998. Design and analysis of improved instantaneous field of view of rosette scanning infrared seeker. Proceedings SPIE., 3365: 158-168.

Jahng, S.G., H.K. Hong, S.H. Han and J.S. Choi, 1999a. Dynamic simulation of the rosette scanning infrared seeker and an IRCCM using the moment technique. Opt. Eng., 38: 921-928.

Jahng, S.G., H.K. Hong and J.S. Choi, 1999b. Simulation of rosette infrared seeker and counter-countermeasure using K-means algorithm. IEICE. Trans. Fundam. Elect. Commun. Comput. Sci., E82-A: 987-993.

Jahng, S.G., H.K. Hong, D.S. Seo and J.S. Choi, 2000. New infrared counter-countermeasure technique using an iterative self-organizing data algorithm for the rosette scanning infrared seeker. Opt. Eng., 39: 2397-2404.

Jahng, S.G., H.K. Hong, K.S. Doo, J.S. Oh, J.S. Choi and D.S. Seo, 2002. A New Clustering Algorithm for the Scanned Infrared Image of the Rosette Seeker. Patent Application Publication, US 2002/0037106 A1.

Pooly, J.U. and F.G. Collin, 2001. HILS testing: The use of PC for real time IR reticle simulation. Proceedings SPIE., 4366: 417-424.

Shokouhi, Sh.B., A.K. Momtaz and H. Soltanizadeh, 2005. A new weighting and clustering methods for discrimination of objects on the rosette pattern. WSEAS. Trans. Inform. Sci. Appl., 2: 1250-1257.