An Effective Simulator for the Rosette Scanning Infrared Seeker

Hadi Soltanizadeh
email: h_soltanizadeh@iust.ac.ir

Shahriar B. Shokouhi
email: bshokouhi@iust.ac.ir

Iran University of Science and Technology, Department of Electrical Engineering, Tehran, Iran

Keywords: Rosette Pattern Seeker, Classification, Central gravity, Tracking Loop, Simulator

ABSTRACT

Hardware in the loop simulation (HILS) 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 paper, the rosette pattern along with the target and flares is simulated and then the clustering and classification method, ISODATA is used to distinguish the target. Also it is used the mean and weighted mean methods in order to compute target central gravity. It is revealed that the tracking loop presents the best results when using the proposed ISODATA method to detect the target and Error Back Propagation (EBP) learning method for computing centroids.

I. 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 missiles' 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 [1].

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. It offers the imaging information of target to the processing unit. Planes keep themselves safe against the thermal tracking missiles by discharging flares. The flares are false targets released in different periods of time in discontinuous format to misguide the seeker. 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 evaluate the performance of IR tracking seekers, HILS[1] is used. The great interest for HILS makes it important to gather knowledge about its field of application and its limitations. A natural step in this process is designing a model that can be used for computer simulations. This paper describes such a HILS model. The model is implemented as a part of a real-time framework along with numerous rosette pattern types, target shapes, countermeasures and so on. We 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 [2].

This paper is organized as follows; in the next section, rosette pattern, along with the target and flares is simulated. Then the variety types of clustering and classification methods such as Moment, K-Mean and ISODATA used to distinguish the target, are described. In order to determine the central gravity of target, the mean and weighted mean have been applied. 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 and the results are evaluated. The tracking loop presents the best results when using the proposed ISODATA method to detect the target and EBP learning in computing the target central gravity [3].

II. General Properties of the 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 equation (1) [4].

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

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 $N1=f_1/f$ and $N2=f_2/f$ are both positive integers, the pattern will be

[1] **Hardware In Loop Simulation**

closed. Moreover N_1 and N_2 are the smallest integers satisfying.

2L

2L

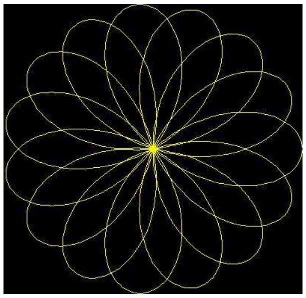


Figure 1. The simulated rosette pattern in the background image

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 equation of rosette pattern in the polar system is:

$$\rho_t = Cos\left(\pi(f_1+f_2)\,{}^{t}/_{T_t}\right), \theta_t = \pi(f_1-f_2)\,{}^{t}/_{T_t} \qquad \textbf{(2)}$$

It can be rewritten in Cartesian form,

$$x_t = \rho Cos\left(\theta_t\right),\ y_t = \rho Sin\left(\theta_t\right) \qquad \textbf{(3)}$$

To map the rosette pattern in background image we have:

$$X\left(t\right) = \left(L + L \times x_t\right)$$
$$Y\left(t\right) = \left(L - L \times y_t\right) \qquad \textbf{(4)}$$
$$L = {}^{D}/_{2}$$

Where L is the half of background image height and is equal to 200 pixels. In this research, the rosette pattern parameter values have been chosen as: $f_1=275$, $f_2=100$ and $N_1=11$, $N_2=4$.

III. Target and Flares

A background image for the target and flares is considered a black image the same size of main background image. Then the target and flares images will be drawn in the background image. Therefore we can draw all kind of targets and flares in the background image as they are in TFOV.

Flares radiate non-constant IR[1] intensity during burning [1]. The intensity of the target varies between 11-32% of flare intensity. Hence, if the maximum intensity of flare takes the value of 255, the target intensity equals 80. Each flare burns completely about 3.5 seconds. In simulations we consider maximum flare IR intensity, about 3 times greater than maximum target IR intensity. Also flare IR intensity varies with time according to equation (5).

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

Where *I(t)* is the IR intensity for flare.

IV. The Classifiers of the rosette pattern

There are different methods of classification in the rosette pattern. Classification is applied to discriminate target from flares. Classifiers are ranked with their requisite processing time. If they can not discriminate the target and flares in time, the missile will miss the target. Now we will discuss the general methods for clustering and classification in the rosette pattern, also the proposed method will be discussed [5].

A target in presence of some flares and classification results are shown in figure 2. The methods of classification are as follows;

ISODATA Method

In the Moment method, classification is based on radiation density. Considering the fact that the amplitude of flare radiation signal is greater than the target's, signal levels are compared. Signals with the amplitudes greater than the target signal level are ignored [1]. Flare radiation intensity is varying during time Therefore this method cannot detect the target correctly when the flare intensity is equal to the target and missile will be perverted.

In the K-Mean method, all the 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, target can be determined. In this method, 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 [3].

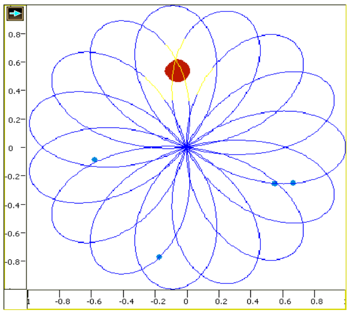
ISODATA method exploits ISODATA algorithm to cluster samples in the reconstructed image. Clustering is based on minimizing the squared error. Error is the sum of squared distances between samples and central gravity of each class. Therefore clustering pixels of target and flares in the reconstructed 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 is changed as 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 [4].

There are two kinds of improved ISODATA algorithm [6]:

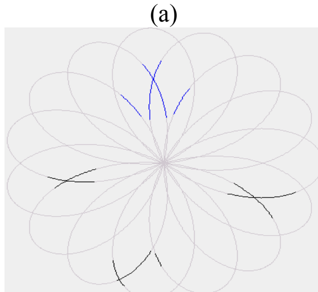
a) At first, all samples considered in one big cluster. Then split this big cluster into the clusters and each of new clusters splits to some other smaller clusters based on existing parameters. This splitting would be repeated until any cluster couldn't be split.

[1] Infrared

b) First of all, each sample is considered as a cluster. Then samples are clustered by merging. The process time of this method is slower than the previous one.



(a)



(b)

Figure 2. a) Target and three groups of flares in the rosette pattern. b) Reconstructed image of target and flares by considering IFOV=0.15.

In the proposed approach, all samples lied in the same line, are considered as a single cluster (target or flares). Then new clusters are merged together with ISODATA algorithm. Thus speed and precision of data clustering will be improved. [6]

A target and three flares in the rosette pattern are demonstrated in figure 2(a). Four recognized clusters employing the mentioned algorithm (one target and three flares) are depicted in figure 2(b).

V. Central gravity of the objects

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 objects' central gravity. In the target tracking, system tracks the central gravity of the target; hence errors in computed central gravity leads to target lose. Therefore precise computation of the targets' central gravity is of utmost importance. In this section, after introducing some methods of computing central gravity, we propose an improved method [7].

Weighted mean method

As shown in figure 3 the real central gravity and computed central gravity with simple mean method, for a sample target are different.

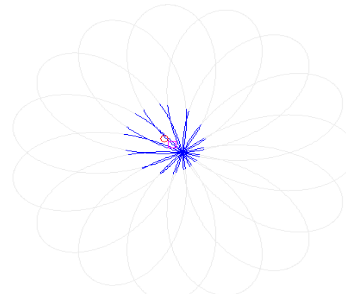


Figure 3. The circle point is shown real center and the square point is shown computed central gravity of the target.

Density of lines 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 set using line density. The weight function value would be greater where the line density is less. Equation (7) is used to compute the central gravity of the objects.

$$\hat{x} = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i}, \ \hat{y} = \frac{\sum_{i=1}^{m} w_i y_i}{\sum_{i=1}^{m} w_i} \qquad \textbf{(6)}$$

Where w_i and m are the value of i^{th} element of weight matrix and length of weight matrix respectively.

Distribution function describes the line density in 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 the target in TFOV. In the next step, the weight function is determined inverting distribution function values. Dimension of weight matrixes is $L \times N$, so any samples in rosette pattern have the weight.

Where N is the half of samples for each rosette petal. Also the distance between two petals would be divided into L slices. Therefore k^{th} line equation is:

$$y_k = \left(\frac{k}{L}\tan(\frac{2\pi}{N})\right).x \qquad (7)$$

Figure 4 shows partitioning two adjacent petals in a rosette pattern using the lines of equation (8).

To compute the distribution function, a point target $P(r,\theta)$ is moved along the lines of equation (6) with a defined step from r=0 to r=1, where $r = (x^2 + y^2)^{1/2}$

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

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 10*100 matrix. The wisely selected initial values help rapid convergence to the correct answer. In the training stage, a target would be placed on rosette pattern, and then its central gravity is computed employing equation (6). So the computed center (x_{out}) would be compared with real target central gravity (x_d):

$$e_x = x_{out} - x_d \quad \textbf{(8)}$$

For simplicity the relations are considered only in one dimension. The weights should be changed to reduce the comparison error (equation 8) 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

$$w_x(n+1) = w_x(n) + \Delta w_x(n) \quad \textbf{(9)}$$

For both directions x and y (9) changes to:

$$w(n+1) = w(n) + \Delta w_x + \Delta w_y \quad \textbf{(10)}$$

Where □w_x is the adjustment term represented by:

$$\Delta w_x(n) = \eta e_x (x - x_{out}) \quad \textbf{(11)}$$

Where η is learning rate.

In the next step, the class center is set to the next point along the lines represented in Figure 4, and the entire calculations are repeated. This process continues until 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 1*[TFOV]*.To decrease calculation, we have exploited the periodicity property of the weights; they are computed only for the range between two neighboring petal's tips.[6][7].

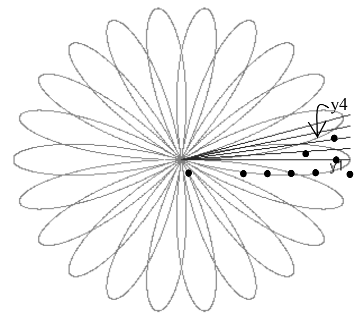


Figure 4. Division of two neighboring petals

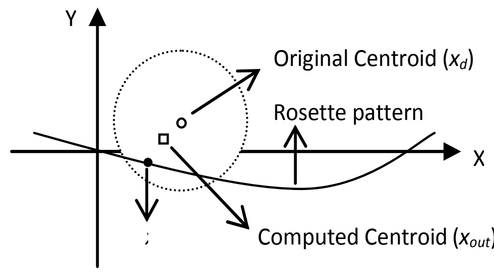


Figure 5. Rosette pattern and the related parameters

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.

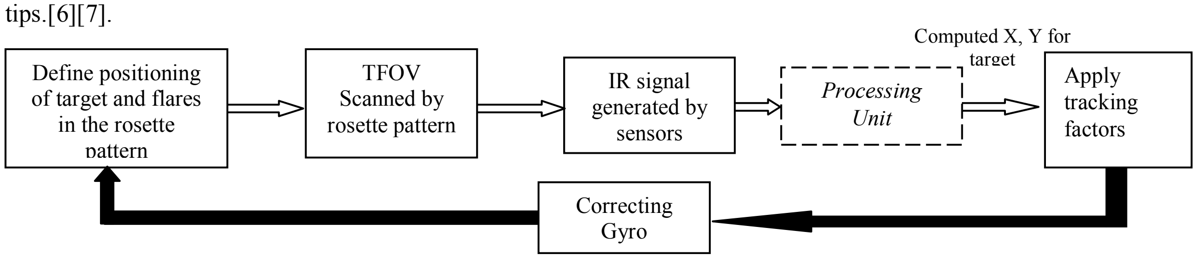
VI. Results

The image matrix size is chosen 400 by 400 and the sampling frequency is 100 KHz. 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.

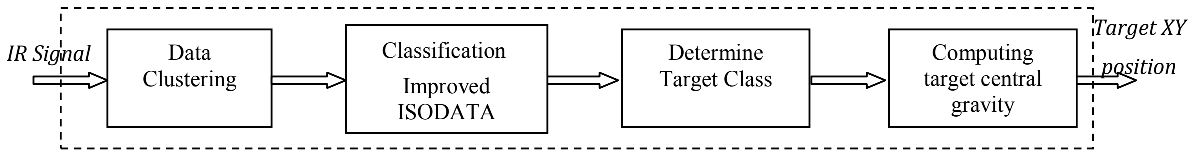
In simulation of the tracking loop, a target with 0.1[TFOV] radius would be used which diffused two flares with radius 0.02[TFOV]. General block diagram of the tracking loop system is shown in figure 6.

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 7 shows the results for different input parameters. The considered parameters are selected appropriately for figures 7 a, b and c therefore the missile has tracked the target successfully. But as shown in figure 7 d, tracking is unsuccessful because of the unsuitable parameters.

The simulation times achieved using two different algorithms for one circular target and different number of flares. The proposed method, ISODATA based on clustering, is processed about 30ms which is less than 40ms, one period of the rosette scanning time. The results for different groups of parameters also can be compared.

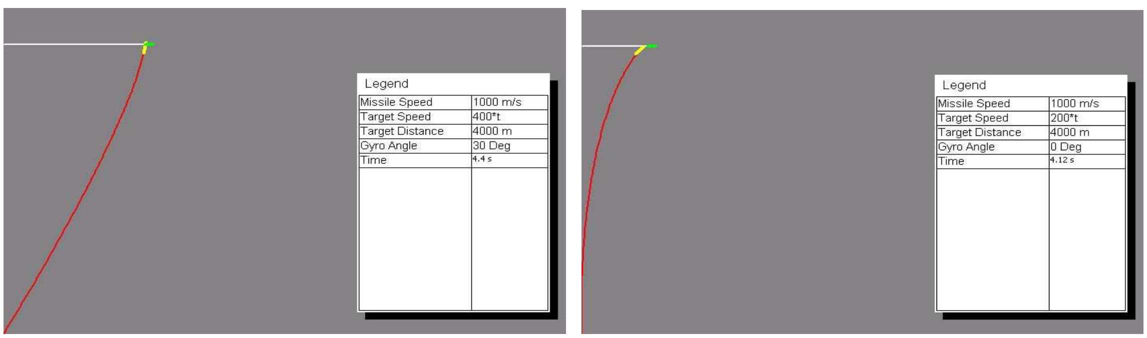


(a)



(b)

Figure. 6 (a) Schematic block diagram of tracking loop (b) Process block in tracking system RSIS



(a) (b)

Figure 7. The results of tracking for different parameters: target speed, target distance and missile angle, (the processing time table is embedded in the figures)

VII. Conclusion

This paper is described simulation of the rosette pattern, target and flares and IR signals. For data clustering and classification the effective ISODATA method has been used. Central gravity of target is computed using an intelligent method which can consider shape and size. The computed central gravity is delivered to the tracking loop. The target images were loaded into the system's RAM before the evaluation started. In a HIL 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 reticle 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

[1]. S.G. Jahng, H.K. Hong, S. H. Han, and J. S. Choi, “Dynamic simulation of the rosette scanning infrared seeker and an IRCCM using the moment technique,” *Opt. Eng*., Vol. 38, No. 5, pp. 921-928, 1999.

[2]. J. U. Pooly and F. G. Collin, "HILS testing: the use of PC for real time IR reticle simulation", Proc. SPIE, Vol. 4366, pp 417-424, 2001.

[3]. G. Jahng, H.K. Hong, and J.S. Choi, “Simulation of Rosette Infrared Seeker and Counter-countermeasure Using K-Means Algorithm,” *IEICE Tran. on Fundamentals of Electronics, Communications and Computer Science*s, Vol. E82-A, No. 6, pp. 987-993, 1999.

[4]. S.G. Jahng, H.K. Hong, D.S. Seo, and J.S. Choi, “New infrared counter-countermeasure technique using an iterative self-organizing data algorithm for the rosette scanning infrared seeker,” *Opt. Eng*., Vol. 39, No. 9, pp. 2397-2404, 2000.

[5]. S. G. Jahng, H. k. Hong, K. S. Doo, J. S. Oh, J. S. Choi, and D. S. Seo, “A new Clustering Algorithm for the Scanned Infrared Image of the Rosette Seeker,” *Patent Application Publication*, US 2002/0037106 A1, 2002.

[6]. A. K. Momtaz, Sh. B. Shokouhi, H. Soltanizadeh, “Classification and Detection Target and Flares in Rosette Pattern Based on Clustering”, Proceedings of 10th Conference of Iranian Computer Institute, Iranian Communication Research Center, pp. 433-441, 2002.

[7]. Sh. B. Shokouhi, A. K. Momtaz and H. Soltanizadeh, “A New Weighting and Clustering Methods for Discrimination of Objects on the Rosette Pattern,” *WSEAS TRANS. ON INFORMATION SCIENCE & APPLICATIONS*, Issue 9, Vol. 2, pp. 1250-1257, Sep. 2005.