Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp167-172)

A New Weighting and Clustering method for Discrimination of Objects on the Rosette Pattern

Shahriar Baradaran Shokouhi, Amirkeyvan Momtaz, Hadi Soltanizadeh
Collage of Electrical Engineering
Iran University of Science and Technology
Narmak, Tehran
Iran

Abstract: -Rosette scan infrared seeker is a single or double band detector with rosette pattern which is mounted on the thermal tracking missiles. 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 then the center of each class is determined. The conventional clustering techniques on the rosette pattern are unable to classify all samples correctly. A new clustering method is proposed in this paper. This algorithm makes small groups from the neighborhood local features and then merges the reconstructed groups to the real clusters. 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.

Key-Words: -Rosette pattern, field of view, cluster, classification, centroid.

1 Introduction

A rosette scan infrared seeker (RSIS) scans a small instantaneous field of view (IFOV) across the total field of view (TFOV) and detects the heat radiated from the target. IFOV is the diameter of detector moving along the path of the rosette pattern, while TFOV is the inner area of a circle with the center of the rosette pattern and a radius equal to the length of each petal. For simplicity the size of TFOV is normalized to 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 [1]. In order to distinguish real target from false targets, the previous studies based on image processing algorithms such as k-means (KMA), Iterative Self Organizing Data Analysis Technique (ISODATA), and moment techniques.

In the moment technique, the target is distinguished from the flare by setting the detection threshold equal to the average intensity of the previously detected target signal [2]. 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, 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 [3]. The clustering result that the KMA generates depends on the seed point of an initial class [4]. 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 [5, 6]. 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 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.

To calculate the center of each class in the rosette pattern, a new method is introduced. In previous works, the class centroids are calculated precisely, considering a weight for each point of the rosette pattern. These weights are computed according to the distribution function of total number of target image pixels [6]. In other words, the distribution function, and relevant weights are not independent of the target size. Hence, variation of target radius deteriorates the result. This dependence to target size is considered in the proposed method so despite the variation of classes` radii the result will be still precise.

2 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 Cartesian coordinates can be expressed with the equation 1 [2].

$$x(t) = \frac{d}{2}(\cos 2pf_1t + \cos 2pf_2t)$$
$$y(t) = \frac{d}{2}(\sin 2pf_1t - \sin 2pf_2t) \qquad (1)$$

where d is the refractive index of the prism.

The values of rotating elements 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 is closed. Moreover N_1 and N_2 are the smallest integers satisfying

$$\frac{N_2}{N_1} = \frac{f_1}{f_2} \qquad (2)$$

The rosette period, T, is $1/f = N_1/f_1 = N_2/f_2$. The number of petals in the rosette pattern is represented by

$$N = N_1 + N_2 \qquad (3)$$

The parameter representing the width of the rosette pattern petals is

$$\Delta N = N_1 - N_2 \qquad (4)$$

The width of the petal increases with increase of ΔN .

To decrease the effect of detector noise and background signals, the size of IFOV should be chosen small, however it should be large enough to provide full scan coverage (FSC). The size of IFOV is defined as the distance between two points selected from the intersection area of two neighboring petals [7].

$$w = d\cos\left(\frac{p}{\Delta N}\right)\sqrt{2 - 2\cos\left(\frac{2p}{N}\right)}, \Delta N \geq 4 \qquad (5)$$

Figure 1 illustrates the detected images of the circular targets on the rosette pattern. The center coordinates and the radii of each target are A(x,y,r)=(0.1,0.2,0.1), B(x,y,r)=(0.1,-0.15,0.02) and C(x,y,r)=(0.5,0.45,0.01). The rosette pattern parameters are N_1=13, N_2=9 and IFOV=0.20126×TFOV. For simplicity the radius of rosette pattern (TFOV) is normalized to 1.

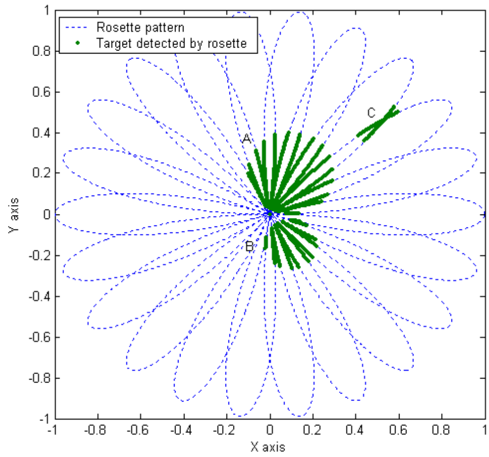


Fig. 1: The detected images according to the various positions and radii of the circular inputs

3 Calculation of the Centroid

To calculate the center of each class in the rosette pattern, two methods are proposed based on a distribution function and a neural network training scheme respectively.

In the averaging method, the positions of all samples of each class are stored in the memory. At the end of each period, the stored data is averaged and the result is set as the center of the class [7]. Because of the nonlinearity of the rosette pattern, the number of scan lines passing over the target is not uniform over the TFOV. There are more scan lines at the center than the extremes of the pattern. Consequently the computed centroid leans to the center of the rosette pattern.

To compensate these errors, the second method, specifies a weight for each point of the rosette pattern. Calculating the weight as a function of target position is based on approximating the distribution function of the total number of pixels (TNOP) for each class of TFOV.

Determining the TNOP, a class with the radius of 0.1×TFOV is set to the center of the rosette pattern. When one scan frame of the rosette pattern is finished, the TNOP for the relevant class in each position of TFOV is calculated. Then the weight function is defined as a reciprocal of the distribution function [6].

Figure 2 shows the distribution function of the TNOP of the corresponding class.

In simulations, when the radius of relevant class is equal to the radius of the class used for extracting the weight function (here radius=0.1×TFOV), the performance of the algorithm is acceptable.

However with the variation of the classes' radii, the performance declines.

We calculate the weight of each point of the rosette pattern according to variations of the radii of the classes inspired by the neural network.

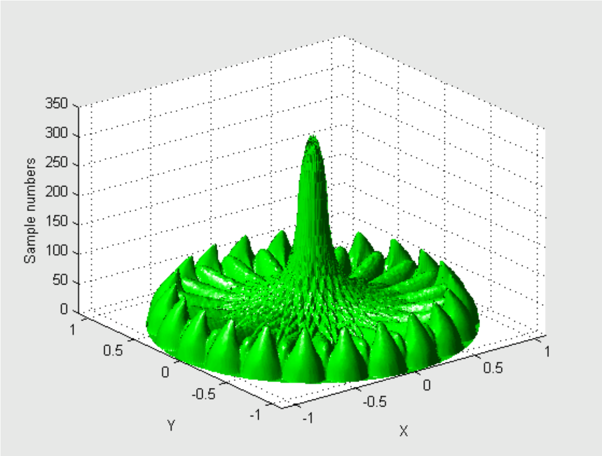


Fig. 2: Distribution of the TNOP of the class with radius of 0.1[TFOV]. The rosette pattern parameters are N_1=13, N_2=9

Also, the area between two neighboring petals is divided into 100 radius parts, and 10 angle directions. Therefore there are 100×10 points in the area for which a weight function is considered.

If the area is divided into *m* lines, the equation of *nth* line is obtained by

$$y_n = tg(\frac{2p}{N}).\frac{n}{m}.x \qquad (6)$$

Figure 3 shows the division of a petal into 4 angle directions.

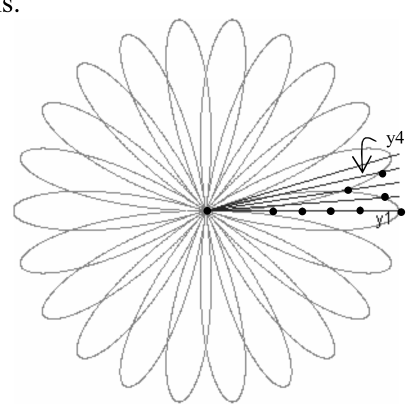


Fig. 3: Division of two neighboring petals. Rosette pattern parameters are: N_1=13, N_2=9.

The weights are put in a 10×100 matrix. The components of the matrix are initialized using the previous method. The wisely selected initial values help rapid convergence to the correct answer. In the training stage, a class with the radius of 0.1×TFOV is set to the center of the rosette pattern. The centroid of the class is calculated as follows:

$$\hat{x} = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i}, \hat{y} = \frac{\sum_{i=1}^{m} w_i y_i}{\sum_{i=1}^{m} w_i} \qquad (7)$$

where, w_i and m stand for the weight related to the point (x_i,y_i) of the class, and the total number of class samples, respectively. In the next stage, the computed result (x_{out}) is compared with the original center of the class (x_d).

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

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 [8]

$$w_x(n+1) = w_x(n) + \Delta w_x(n) \qquad (9)$$

For both directions x and y (9) changes to

$$w(n+1) = w(n) + \Delta w_x + \Delta w_y \qquad (10)$$

where Δw_x is the adjustment term represented by

$$\Delta w_x(n) = h e_x(x - x_{out}) \qquad (11)$$

h is the learning rate.

Figure 4 shows the class, and introduced parameters of a rosette pattern.

In the next step, the class center is set to the next point along the lines represented in Figure 3, and the entire calculations are repeated. This process continues until the center of the class remains unchanged for the whole points of the common area of two neighboring petals.

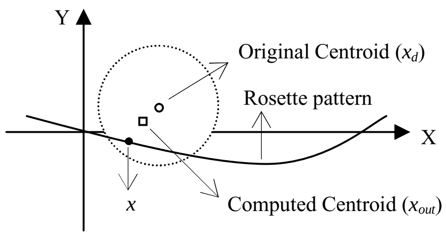
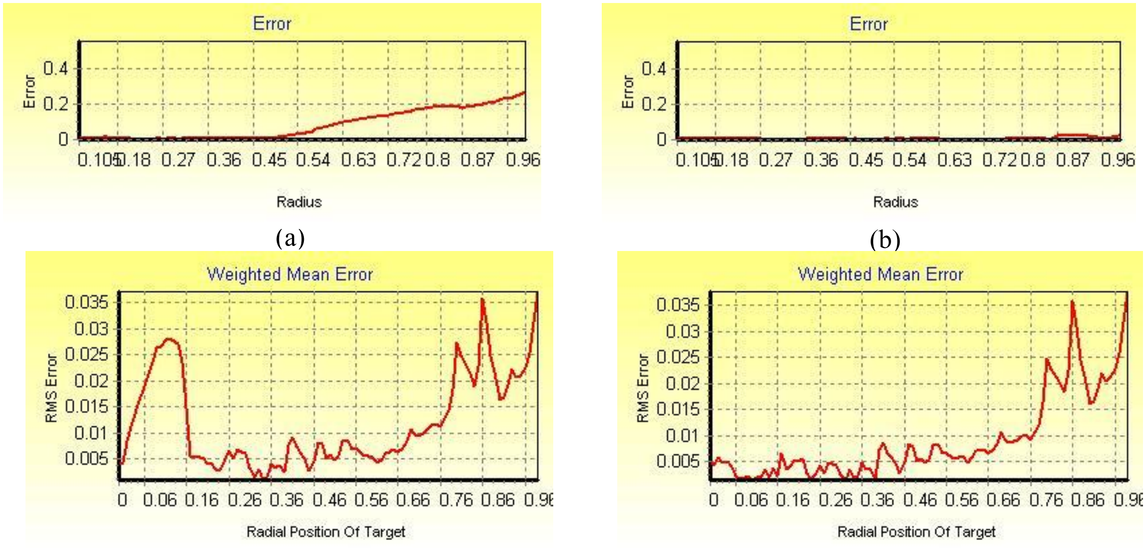


Fig. 4: Rosette pattern and the related parameters

Then the class radius is set to 0.2×TFOV, and the entire process is repeated. The increment of class radius continues till it reaches 1×TFOV. To decrease calculations, we have exploited the periodicity property of the weights; they are computed only for the range between two neighboring petal's tips.

Figure 5 part (a) and (b), show the error due to the variation of class radius in both the distribution function, and proposed method in an unsymmetrical pattern with N_1=11, N_2=4. The center of the class is set to (0.1, 0.1), and the class radius varies from 0.1×TFOV to 1×TFOV. In Figs. (c) and (d) show the difference between the original, and calculated centroid for a class of radius 0.15×TFOV for both the distribution function, and proposed method. The center of the class is set to the x axis (y=0) and moves from x =0 to x =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. The rosette pattern parameters are N_1=13, N_2=9.



(c) (d)

Fig. 5: Variation of error according to the variation of the target radius in (a) the distribution function method and (b) the proposed method. Comparison of the RMS error between (c) the distribution function method and (d) the proposed method

4 New Clustering Method Using Neighborhood Features

The conventional methods are unable to classify all input samples correctly.

In the moment technique, the target is discriminated from the flares by defining a threshold for the intensity. An interpolation for the flare intensity according to [6] has been calculated as in (12).

$$\begin{aligned} I = &-0.0024t^7 + 0.0086t^6 + 0.1260t^5 - 1.0314t^4 \\ &+ 3.1351t^3 - 4.7867t^2 + 3.5501t - 0.0004 \\ &, 0 \le t \le 3.5(s) \end{aligned} \quad (12)$$

where I is the flare intensity. The target intensity is selected 1 out of five of the maximum flare intensity.

According to relation 12, when the intensity of flare is equal to that of target the RSIS fails to track correctly. In k-means method, samples are divided into two classes: target and flare. At the beginning of the algorithm according to the number of classes, two spots are selected as seed points. The result of clustering in k-means algorithm depends on the seed points of initial cluster. If only one flare is discharged by the plane, the algorithm works properly, but when there is more than one flare in the TFOV, the result of clustering algorithm is not acceptable. In the ISODATA method, the samples are clustered according to the ISODATA algorithm. Before running the algorithm, it is necessary to determine some parameter values: 1) the desired number of classes 2) the lumping parameter 3) the splitting parameter 4) maximum number of cluster pairs allowed to be lumped 5) initial cluster centroid i.e. seed point 6) minimum number of samples in each class, and 7) the number of iterations.

While the algorithm is running, each sample is considered as a seed point. These points are the initial classes that should be merged if they are too close or split while containing very dissimilar pixels. Hence, the samples are clustered at the end of the algorithm regardless of their numbers. The algorithm has two main disadvantages: 1) varying the class size, the relevant parameters should be modified 2) since each point is initialized as a separate class, the processing is time-consuming. Therefore, this method is inappropriate for real time applications.

To solve the problems of the conventional ISODATA, all of the neighboring samples located in a cluster are initialized as one class. Then according to the lumping parameter determined by the user at the beginning of the algorithm, the relevant classes are merged together. When the results in two subsequent iterations are the same, classification stops, and classes with insufficient samples are discarded.

Figure 6 shows the flowchart of the proposed algorithm based on the ISODATA algorithm but instead of merging samples, the algorithm merges initial clusters.

Since each initial class, contains many samples, the proposed method has a better speed than the previous techniques. Unlike the ISODATA technique, in the proposed method, the lumping parameter, and minimum number of pixels in a cluster are the only parameters required to be specified at the beginning of the algorithm.

After classification of samples, the algorithm specifies the size of each class. Since the target size is larger than the flare's, the largest class is selected as a target.

Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp167-172)

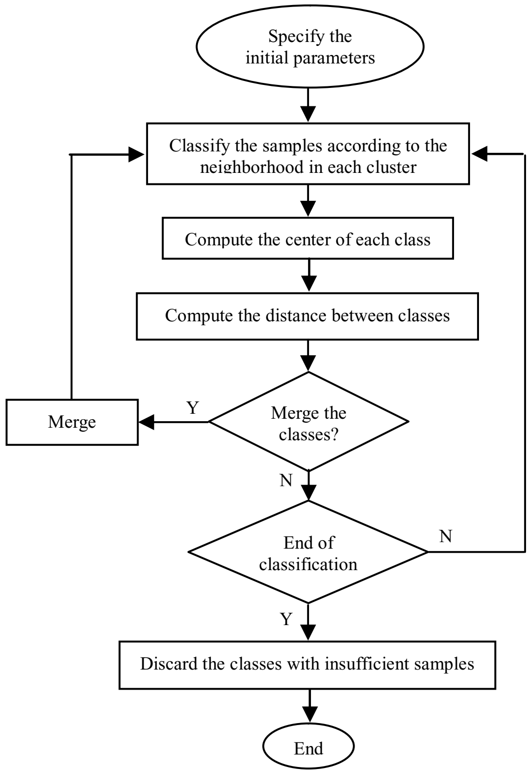


Fig. 6: The flowchart of proposed algorithm

Figure 7 shows the distance between the centers of initial classes situated in two different positions of the rosette pattern.

To determine the value of the lumping parameter some issues are to be noted; near the center of the rosette pattern, as shown in Figure 7, because of symmetry, the centers of initial classes are very close together. But at the tips of the petals, these distances increase. If the value of lumping parameter is determined low, when the target is located near the tips of the petals, each initial cluster of the target class may be considered as an independent class. The target class is the largest one among the existing classes; independent classes have also large sizes. So, one of these classes is selected as the target hence, the RSIS can correctly track the target.

5 Simulation Results in the Tracking Loop

The simulation is performed with the following assumptions: 1) both the target and flares have circular shapes. 2) The radii of the target and flares are assumed 0.1×TFOV, and 0.02×TFOV, respectively. 3) The radiant energies of atmosphere are not considered.

Figure 8 shows the target and flares trajectories in the tracking loop. The target moves with the velocity of 1×TFOV/Second in $(+x,+y)$ direction. But flare1 and flare2 fall with the velocity of 0.1×TFOV/Second in $(-x,-y)$ and $(+x,-y)$, respectively. The flares are discharged 0.2 s after the beginning of tracking loop. The intensity of flares is selected according to relation 12, and has the peak value 5 times higher than that of target.

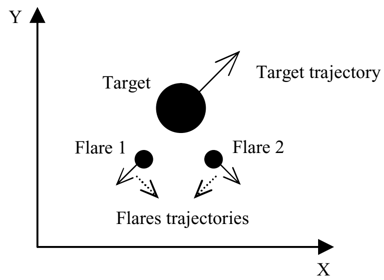
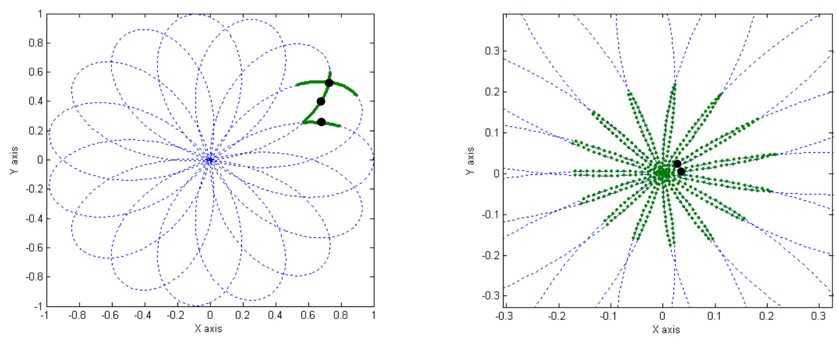


Fig 8: The considered target and flares trajectories

In Figure 9 tracking errors have been shown for the previous clustering methods; moment, k-means and ISODATA. In the moment technique because of equal intensities of target and flares at the beginning, the RSIS can not recognize the target correctly, so the tracking error increases. In the k-means method, all of the samples are divided into two clusters. When flares are released, both of them lie in one cluster and target is specified as other class. Since the flares recede each other, after some periods of time the size of flare cluster becomes bigger than the target and the tracking error suddenly jumps. In the ISODATA technique, the result is based on parameter selection.



(a) (b)

Fig 7: The distances of the initial cluster centers for a class with the radius of 0.2×TFOV in position of (a) (0.7, 0.4) and (b) (0.02, 0.02) with magnification. The rosette pattern parameters are N_1=11, N_2=4.

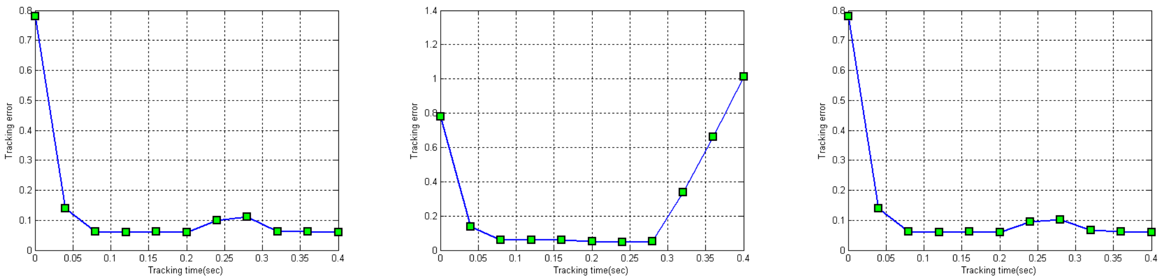


Fig. 9: The tracking error in conventional methods: (a) moment technique (b) k-means and (c) ISODATA. The initial target position is $(x, y) = (0.5, 0.6)$. The rosette pattern parameters are N_1=11, N_2=4.

If the parameters are not selected correctly or not modified during the execution of the algorithm, classification will not be acceptable. The lumping and splitting parameters are selected 0.2 and 0.1, respectively. According to Figure 9 part (c), samples are not classified correctly but when the intensity of flares increases enough, the RSIS can distinguish the target from flares.

Simulation is done under same conditions as before. The flare intensity is time varying and the two flares released 0.2 s after starting the target tracking. The lumping parameter is set to 0.1×TFOV. Despite the variation of the rosette parameters, and the size of IFOV, there is no need to change the value of lumping parameter. As shown in Figure 10 when flares with time varying intensities release at time 0.2 s, the tracking error does not change and RSIS distinguishes the target from flares.

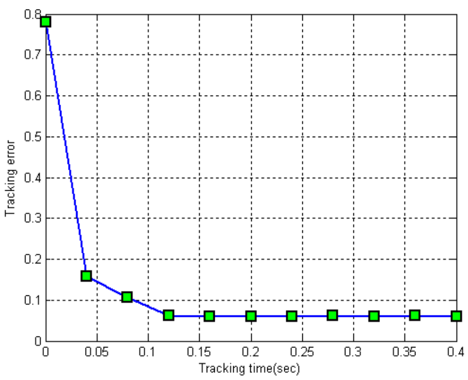


Fig. 10: Tracking error according to proposed algorithm. The initial target position is $(x, y) = (0.5, 0.6)$. The rosette pattern parameters are N_1=11, N_2=4.

6 Conclusion

In this paper, a new method is proposed to calculate the centroid of each class more precisely, regardless of the variation of class radius in the rosette pattern. For each point of the rosette pattern, a weight is determined. Weights are calculated according to the variation of the of the classes radii with the help of the neural network training algorithm. The conventional clustering methods for the rosette pattern like moment, k-means and ISODATA have been processed and their performances in tracking loop studied. For these methods, by variation of number of classes in the field of view, variation of flare intensity and improper initialization of parameters, the sensitivity and ability of the system for distinguishing targets from flares deteriorate. The new method overcomes these limitations, besides it converges faster, and hence, improves the tracking performance significantly.

Refrences:

[1] T. Tajima, S. Wakabayashi, M. Kondo, and T. Takei, “Rosette scan infrared sensor,” *Proc. SPIE*, Vol. 219, pp. 51-57, 1980.

[2] 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.

[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] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*, pp. 213-224, Prentice-Hall, 1996.

[5] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, pp. 94-104, Addision-Wesley Reading, MA, 1974.

[6] 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.

[7] S. G. Jahng, H. K. Hong, S. H. Han, and J. S. Choi, “Design and Analysis of Improved Instantaneous Field of View of Rosette Scanning Infrared Seeker,” *Proc. SPIE,* Vol. 3365, pp. 158-168, 1998.

[8] S. Haykin, “Neural Networks- A comprehensive foundation,” 2.nd edition, *Prentice Hall*, 1999.