"Architecture and Algorithms for Digital Image Processing", Proc. SPIE 435, August 1983, pp. 90-97

Intelligent autocueing of tactical targets

B. Bhanu, A.S. Politopoulos and B.A. Parvin

Ford Aerospace & Communications Corporation
Newport Beach, California 92660

Abstract

In this paper we present a set of algorithms used to automatically detect, segment and classify tactical targets in FLIR (Forward Looking InfraRed) images. These algorithms are implemented in an Intelligent Automatic Target Cueing (IATC) system. Target localization and segmentation is carried out using an intelligent preprocessing step followed by relaxation or a modified double gate filter followed by difference operators. The techniques make use of range, intensity and edge density information. A set of robust features of the segmented targets is computed. These features are normalized and decorrelated. Feature selection is done using the Bhattacharrya measure. Classification techniques include a set of linear, quadratic classifiers, clustering algorithms, and an efficient K-nearest neighbor algorithm. Facilities exist to use structural information, to use feedback to obtain more refined boundaries of the targets and to adapt the cuer to the required mission. The IATC incorporating the above algorithms runs in an automatic mode. The results are shown on a FLIR data base consisting of 480, 512x512, 8 bit air-to-ground images.

I. Introduction

Automatic target recognition is crucial to the success of the deployment of future autonomous vehicles. With the availability of VLSI and VHSIC technology, it is feasible to implement target recognition algorithms in hardware and carry out target cueing in real time. In this paper we present a set of algorithms for the purpose of automatic detection, segmentation, feature extraction and classification of tactical targets in FLIR (Forward Looking InfraRed) images. These algorithms are implemented in an Intelligent Automatic Target Cuer (IATC). Fig. 1 shows the block diagram of the IATC system. Target localization and segmentation is carried out using two schemes. In the first a preprocessing step is carried out which makes use of range, intensity and edge density information. This step reduces the image area which is to be used to detect and segment the targets significantly depending upon the details present in the image. Segmentation is carried out by using a relaxation technique. In the second scheme a modified double gate filter which makes use of range and contrast information is used to localize the targets and segmentation is carried out by a combination of Laplacian and gradient operators. A set of robust features of the segmented targets is computed. These features are first normalized with respect to range (distance of the target from the center of the field of view of the sensor), angular extent of the field-of-view and dynamic range and then decorrelated. Feature selection is done using the Bhattacharrya measure. Classification techniques include a set of linear, quadratic classifiers, clustering algorithms, and an efficient K-nearest neighbor algorithm. Structural information may be used in the decision making process of target classification, feedback between the classifier and the detector to obtain more refined boundaries of the target and allow the cuer to be adaptive to the required mission. The IATC incorporating the above algorithms runs in an automatic mode. The results are shown on a FLIR data base. The data base consists of 480, 512x512, 8 bit air-to-ground images. For each of the images and targets present in the image ground truth information is known. These images include a variety of targets at different ranges (300 meters to 10,000 meters), targets with very poor contrast, targets with nonuniform intensity distribution, targets occluding each other, targets hidden in smoke and dust, targets in close proximity etc. In this paper we concentrate on the detection and classification algorithms applied on a single frame basis, interframe analysis is discussed in [1].

In section II we describe target detection and segmentation algorithms. Section III discusses feature computation, feature selection, classification algorithms and use of structural information. Finally section IV presents the results and general summary of the paper.

II. Target localization and segmentation

In the past several techniques such as spoke filter, superslice, contrast box, double window filter, Fisher linear discriminant, spatial stochastic models, simple operators utilizing edge and texture features, relaxation, mode seekers, pyramid approaches etc. have been used for target detection and segmentation [2-12]. Burton and Benning [2], Schachter [3] and Hartley et al. [4] present evaluation of some of these techniques. In this section we present two new algorithms for the localization and segmentation of targets in FLIR images. Unlike the previous work, we make use of range, whenever it can be used to reduce the amount of computation or adjust the parameters automatically.

(a) Algorithm (A). Preprocessing followed by a relaxation technique

Preprocessing : The objective behind the preprocessing step is to flag those areas in the image where a potential target could be present with a minimum amount of computation performed at a pixel level. The

assumptions made here are that the target objects to be detected in the image differ from the background in gray level so edges are present at their boundary, and there are more edges present with a target than in clutter since the target has typically more structure than natural clutter. Preprocessing makes use of intensity, edge and range information. Depending upon the angle of the field-of-view and the range to the center of the field-of-view, maximum and minimum expected sizes of the targets in pixels are roughly estimated. The preprocessing step requires the following procedure.

1) Compute edges in the image using a Sobel edge operator.
2) Threshold the edge magnitude image obtained in 1). In practice this threshold is automatically selected as a function of range.
3) Prepare a binary mask of the thresholded image in 2).
4) Pass a square window over the mask obtained in 3). The size of this window is twice the maximum size of the estimated target at the given range (see Fig. 2). This window is scanned over the mask in steps of half window sizes from left to right and top to bottom. If the number of edge points lying within the window at a particular location is greater than a certain threshold then create a mask image. This threshold is determined as a function of the minimum size of the estimated target and its boundary points. We compute the centroid of the edge points within the window, and then flag the mask image by adding a constant (say 60) to all the pixels within an area of the size of the window centered at the centroid of edge points.
5) Determine the new window center location and continue scanning the image.

Since we have taken the window size as twice the size of the target, there can be only four overlaps that a target may have at the end of the scan in the mask image (Fig. 2). The values in the mask image will be 0, 60, 120, 180 and 240 depending upon whether there have not been enough edge points or one, two, three and four overlaps of the target in the window.

Using the mask image we can select the desired region from the original image which should be further investigated for the possible targets. The higher the number of overlaps, the better the chances that a target is present in that region with fewer pixels to be subsequently processed during segmentation. However, it is possible that in the higher overlapped regions, there may not be entire target details since there is an intensity distribution across the target. In practice, we have found that one or two overlaps are sufficient to flag the potential targets successfully and they reduce the image area over which segmentation is to be carried by over 85 to 90% depending upon the details present in the image. As an example Fig. 3 shows the preprocessing step. Fig. 3(a) is a 512x512 FLIR image. Fig. 3(b) shows the thresholded Sobel magnitude image. Fig. 3(c) shows window overlaps and Fig. 3(d) shows the selected overlap superimposed on the original image. This is the only area of the image which needs to be considered subsequently for possible location of the target. Connected components are found in the mask image and then we use them to do the segmentation of the original image. A two class gradient relaxation technique [13] is used for the segmentation of each connected component. The technique provides the automatic selection of the threshold. The 3 required parameters in the segmentation process are set once for all the images in the data base. A brief description of the technique is given below.

Segmentation using a relaxation technique : Suppose we have a set of N pixels i= 1,2,...,N which fall into two classes λ_1 and λ_2 corresponding to the white (gray value = 255) and black (gray value = 0). The relaxation process is specified by choosing a model of interaction between pixels. We attach to every pixel i the set V_i of its 8 nearest neighbors. Assuming that objects of interest in the picture are continuous we will make like reinforce like and define a compatibility function c such that:

$$\begin{aligned} c(i,\lambda_k,j,\lambda_\ell) &= 0, \quad k\neq\ell, \quad j \text{ in } V_i \text{ for all } i \\ c(i,\lambda_k,j,\lambda_k) &= 1, \quad k=1,2 \quad j \text{ in } V_i \text{ for all } i \end{aligned} \tag{1}$$

The consistency vector $\vec{q}_i$ is then defined as

$$q_i(\lambda_k) = \frac{1}{8} \sum_{j\in V_i} \sum_{\ell=1}^{2} c(i,\lambda_k,j,\lambda_\ell)\, p_j(\lambda_\ell), \qquad \begin{matrix} k = 1,2 \\ i = 1,\ldots,N \end{matrix} \tag{2}$$

In effect $q_i(\lambda_k)$ is the mean neighborhood probability of the ith pixel for the class under consideration, i.e., $q_i(\lambda_k) = (1/8) \sum_{j\in V_i} p_j(\lambda_k)$.

Based upon the explicit use of consistency and ambiguity a global criterion is defined upon the set of pixels. It is given by the inner product of probability vector $\vec{p}_i$ and consistency vector $\vec{q}_i$ as,

$$C(\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N) = \sum_{i=1}^{N} \vec{p}_i \cdot \vec{q}_i \tag{3}$$

It is maximized using the gradient projection approach. The maximization of the global criterion (3) means that we are seeking a local maximum close to the initial labeling $\vec{p}_i^{(0)}$ $(i = 1,\ldots,N)$ subject to the constraints that $\vec{p}_i$'s are probability vectors. It results in a reduced inconsistency and ambiguity. Inconsistency is defined as the error between $\vec{p}_i$ and $\vec{q}_i$. Intuitively this means the discrepancy between what every pixel "thinks" about its own labeling ($\vec{p}_i$) and what its neighbors "think" about it ($\vec{q}_i$). Ambiguity

is measured by the quadratic entropy and results from the fact that initial labeling $\vec{p}_i^{(o)}$ is ambiguous ($\vec{p}_i^{(o)}$ are not unit vectors). We are therefore trying to align the vectors $\vec{p}_i$ and $\vec{q}_i$ while turning them into unit vectors. Indeed it can be easily seen that each term $\vec{p}_i \cdot \vec{q}_i$ is maximum for $\vec{p}_i = \vec{q}_i$ (maximum consistency) and $\vec{p}_i = \vec{q}_i$ = unit vector (maximum unambiguity).

The maximization of (3) results in iterative equations given by,

$$p_i^{(n+1)}(\lambda_k) = p_i^{(n)}(\lambda_k) + \rho_i^{(n)} \left[2q_i(\lambda_k)-1\right] \tag{4}$$

Where,

$$\rho_i^{(n)} = \begin{cases} \alpha_1 \rho_{iMax}^{(n)}, & \text{if } 2q_i(\lambda_1)-1 > 0 \\ \alpha_2 \rho_{iMax}^{(n)}, & \text{if } 2q_i(\lambda_1)-1 < 0 \end{cases} \tag{5}$$

and

$$\rho_{iMax}^{(n)} = \begin{cases} \left(\dfrac{1-p_i^{(n)}(\lambda_1)}{2q_i(\lambda_1)-1}\right), & \text{if } 2q_i(\lambda_1)-1 > 0 \end{cases} \tag{6}$$

$$\begin{cases} \left(\dfrac{p_i^{(n)}(\lambda_1)}{1-2q_i(\lambda_1)}\right), & \text{if } 2q_i(\lambda_1)-1 < 0 \end{cases} \tag{7}$$

where, k=1,2 and and are constants less than unity. In the actual implementation of the algorithm, we only need to evaluate (4) for k=1 or 2 only since $p_i^{(n+1)}(\lambda_1) + p_i^{(n+1)}(\lambda_2) = 1$.

The initial assignment of probabilities to every pixel is very important. It affects the convergence rate and the results of relaxation scheme. The initial assignment of probabilities has been obtained by,

$$p_i(\lambda_1) = FACT*\left(\frac{I(i)-\bar{I}}{255}\right)+0.5 \tag{8}$$

where I(i) is the intensity at the ith pixel and G (=256) the number of possible gray levels ($0 \le I(i) \le G-1$) and $\bar{I}$ is the mean of the image. When $I(i) < \bar{I}$, FACT has usually been taken between 0.7 and 1. Of course, if the first term of (8) happens to be greater than 0.5 or less than -0.5, then a probability of one or zero respectively is assigned to that pixel.

This method provides the automatic selection of threshold for the segmentation of images and the control over the relaxation process by choosing the α_1, α_2 and FACT parameters which can be tuned to obtain the desired segmentation results. The magnitude of α's controls the degree of smoothing at each iteration and their ratio the bias towards a class. Changing the values of α_1 and α_2 not only allows one to control where one wants to converge but also how fast one converges. The magnitude of FACT controls the initial assignment of probabilities.

Fig. 4 shows the segmentation results on the superimposed image shown in Fig. 3(d). Using this technique we were able to extract targets even when contrast was very poor (about 10%).

(b) Algorithm (B), Modified Double Gate Filter Followed by Difference Operators

Modified double gate filter : This filter is designed for the acquisition of contrasting compact subsets of pixels having the intuitive appearance of 'blobs'. Fig. 5 illustrates the geometry of the filter. An explanation of the basic filter properties is given in a one-dimensional setting shown in Fig. 6. Here, an idealized square target profile is superimposed on a flat background. Let $(a_1,p_1),(a_2,p_2),\ldots$, etc. denote average intensity values within the corresponding domains. The differences $(a_1-p_1),(a_2-p_2),\ldots$ etc. are evaluated and plotted as in Fig. 6(b). It can be seen from this plot that the target-to-background boundary is revealed by the location of the maximum difference a_i-p_i (in this case, a_6-p_6). Fig. 7 illustrates the same approach on an intensity distribution which, although still idealized, exhibits less abrupt transitions from object-to-background intensities. Again, as shown by Fig. 7(b), the maximization of the differences a_i-p_i occurs at the boundary between object and background intensities.

The key message conveyed by the one-dimensional formulations of the previous paragraph is that telescoping expansion of perimeter-to-target differences should be expected to peak at the target-to-background interface point, at least for distributions that show close approximation to the idealized examples considered above. It is obviously possible to readily defeat the generality of the deduced statement with intensity distributions which radically depart from the square pulse or triangular waveforms examined here. However, within the context of specific practical FLIR imagery and target configurations, enough similarity to this model was observed to make the perceived principle a viable practical approach.

The two-dimensional extension of the one-dimensional considerations is illustrated by Fig. 5. Here, a square mask configuration is partitioned into eight sectors. Each sector, in turn, is partitioned into a sequence of expanding triangular domains with each triangle accompanied by a corresponding perimeter bin, as shown in Fig. 5(b). The selection of the geometric form of the sectors, their number and their size can be

parametrically optimized for the specific problem under consideration. We have found that the eight triangular sectors shown in Fig. 5 with a two-pixel perimeter expansion strategy were adequate. Before the application of the filter the image is minified by a factor of 1,2,4 or 8 depending upon the range. Minification has two effects: first it reduces the amount of computation and second, it results in a more "blob-like" definition of the target silhouette. The filter is allowed to traverse the image on a pixel by pixel basis. At each pixel location each of the filter's eight sectors is telescopically expanded from some minimum size (say, 2x2 pixels) to some predetermined upper size, grossly estimated to accommodate the maximum expected target extent. In practice the slant range and the angle of field of view measures are used. For each sector's telescoping run the maximum difference $d_j = \max(a_i - p_i)$ $j=1,..8$ is noted and when all eight sectors have been considered, the filter response is set at

$$F(i) = \min(d_1, d_2, \ldots d_8)$$

where i denotes the ith pixel at which the filter is currently centered. As the filter is scanned across the image, a preselected number (say, 20) of top filter response locations are noted. At the end of the image scan these locations are the ones considered cued by the filter as containing contrasting compact object regions meriting closer scrutiny by a preclassifier. It is worth noting that to avoid repeated high responses from the same object a requirement is imposed dictating at least half a maximum gate size separation from previous top responses before adding it to the top response list. Also, the filter can be used for both hot and cold contrast region extraction by monitoring the sign of the eight differences d_j.

Segmentation : The threshold used to segment the target is found by making use of the fact that the interface points between background and object are locations of high second derivative values. A histogram is formed of the positive Laplacian values. A second order derivative discrete operator is then applied to this histogram. The actual threshold is selected as the value above the histogram's peak value where the seocnd-order derivative of the histogram has a maximum value. Since in this method a single threshold may not be the right threshold if there is a large intensity distribution across the target, before the preclassifier rejects an object as clutter, we vary threshold about the threshold value as obtained above. This variation is based on the fact that many edge or boundary points of the extracted object coincide with the thresholded Sobel. Thus the feedback between the classifier and segmentator helps to obtain more refined boundaries of the object.

Fig. 8 shows the target localization and segmentation results when we considered 10 top responses. Fig. 9 shows the number of detected targets vs. number of top responses for 76 targets. Most of the targets are found in the first 6 top responses. As an example at response 1, 28 targets, at response 2, 25 targets and so on.

III. Features computation, selection and classification

A few simple features (such as area, perimeter, length and width) of a target are computed as soon as it is segmented. These features are passed to the preclassifier which checks if these values are within some conservative thresholds. The preclassifier classifies the potential target either as a clutter area or a potential target. Once this test is passed, a set of 36 features of the target is computed during training. Features used are mainly shape, gray scale and moment features. Shape features also include four Fourier features which are obtained using only 7 Fourier coefficients [14] computed by using the Goertzel Algorithm [15]. The features are normalized so that they are invariant to rotation, translation and scale. The features can also be normalized with respect to their own dynamic range and mean and variance.

Feature selection is done on the decorrelated features so that the selected feature set does not have correlated features. The feature selection is performed by a K-L Transformation (multi-dimensional axis rotation). The total covariance is just an average of the individual class covariance matrices.

$$[\phi] = \sum_{k=1}^{K} P(S_k)\,[\phi_k]$$

where $P(S_1),\ldots,P(S_K)$ are a priori class probabilities for K classes and the covariance matrix for a class is given by

$$[\phi_k] = E\left[(y_m^{(k)} - \bar{y}_m^{(k)})\,(y_m^{(k)} - \bar{y}_m^{(k)})^t\right]$$

The expected value is approximated by the sample average. $y_m^{(k)}$ is the mth sample for class k. The diagonal elements of the covariance matrix are the feature variances. The matrix which diagonalizes the covariance matrix is computed as

$$A^T [\phi]\, A = \Lambda, \text{ where } \Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

Λ is diagonal having the eigenvalues of the covariance matrix as diagonal elements. The matrix A which accomplishes this diagonalization is the well known matrix of eigenvectors, $A = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N]$, where $\bar{x}_i$ is an eigenvector, i.e., $[\phi]\bar{x}_i = \lambda_i \bar{x}_i$. A new feature set is computed by multiplying every vector in the original space by A^T, i.e., $z_m = A^T y_m$. The covariance matrix in the rotated space is diagonal and it is given by

$$[\phi]_R = A^T [\phi]\, A = \Lambda$$

The rotated space of features is forwarded to the feature selection routine.

The criterion of optimality for the selection of a feature set is the probability of misclassification. Several measures have been proposed, which upper bound the misclassification of the sample. In particular for a Bayes symmetric cost function classifier and Gaussian data, the error rate has been shown to be upper bounded inversely as the Bhattacharrya measure [16]. The computationally simpler form of one-at-a-time (which

involves only scalar means and variances) is used for feature selection. It is given by,

$$B_n(S_1,S_2) = \frac{1}{4} \ln \left[\frac{1}{4}\left(\frac{\sigma_1^2(n)}{\sigma_2^2(n)} + \frac{\sigma_2^2(n)}{\sigma_1^2(n)} + 2 \right) \right] + \frac{1}{4} \left[\frac{(\mu_1(n)-\mu_2(n))^2}{\sigma_1^2(n)+\sigma_2^2(n)} \right]$$

where n refers to the nth dimension of the feature space. μ_1, μ_2, σ_1^2, σ_2^2, are the mean and variance of the nth feature for the classes S_1 and S_2. The equation implies that a large difference in mean accompanied by a small variance is a desirable quality in a feature for distinguishing between two classes. If the variances are significantly different, the feature is still considered of potential use in separating the classes.

Feature selection is done by keeping only those features which are above the average one-at-a-time Bhattacharrya measure and the remainder are discarded. For example if there are 4 classes, then 6 combinations of 2 classes are taken. The sum of the Bhattacharrya measure for a feature for these 6 combinations is computed. This is repeated for all the features. Now the average of the Bhattacharrya measure for all features is computed and only the features having a measure above the average are retained. Using this procedure we found the number of selected features to be 11. Since we compute the K-L transformation for decorrelating the features, we also made a comparison of the Bhattacharrya measure with the decorrelated features eigenvalues. We found that Bhattacharrya measure works better than the feature eigenvalues in identifying the best features.

An experiment to test the accuracy of feature computation and its sensitivity to classification, with respect to different segmentation results obtained when controlling the threshold in segmentation in the Algorithm (B) outlined above, is also done on the selected features. The selected features are found to be robust when the change in the gray level threshold was ± 5.

Mitchell and Lutton [9] make use of projection through the segmented object to derive classification features. Other studies such as [5,7] use moments and a K-nearest neighbor algorithm. In our study the classification algorithms include linear, quadratic training algorithms (Fisher discriminant, Ho-Kashyap method and other discriminant functions) and clustering techniques (K-means, Mean-squared error and hierarchical). The technique that gives the least classification error on the selected feature set is selected. For the results reported here we have used K-means clustering technique for the training of the samples. K-nearest technique is used for classification. It utilizes a tree based approach and reduces the number of distance calculations to determine the K nearest neighbors of a test sample. The classification results for the image of Fig. 3(a) are shown in Fig. 10.

In order to evaluate the clustering of features we find the clusters in the training set. A clustering fidelity criterion used is,

$$\beta = \mathrm{Tr}(S_b) * \mathrm{Tr}(S_w)$$

where:

$$S_b = \frac{1}{K} \sum_{k=1}^{K} (\mu_k - \mu_o)(\mu_k - \mu_o)^T$$

$$\mu_o = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad N = \text{total number objects}$$

$$S_w = \frac{1}{K} \sum_{k=1}^{K} E\left[(x - \mu_k)(x - \mu_k)^t \right]$$

$$E(\cdot) = \frac{1}{M_k} \sum_{x_i \in S_k} (x_i - \mu_k)(x_i - \mu_k)^t$$

S_b and S_w are between-cluster and within-cluster scatter matrices. μ_k is the mean of the kth cluster, M_k is the number of elements in the Kth cluster, x_i is an element in the kth cluster, K is the total number of clusters and μ_o is the overall mean vector of the entire mixture. The behavior of β is shown in Fig. 11. It passes through a maximum at the intrinsic number of clusters (L). When β is maximum the ratio of $\mathrm{Tr}(S_b)/\mathrm{Tr}(S_w)$ is exactly 1. The maximum of β can be determined by incrementing the number of clusters until a decrease is detected. This allows us to evaluate the number of clusters inherent in the data. Departure of the number of clusters so obtained from the known number of clusters tells us about the quality of features. To determine if clustering is really present and not an artifact of the data or clustering method, we used the criteria of compactness and isolation. If a cluster is formed early in the dendrogram corresponding to its size and lasts a relatively long time, it is considered a valid cluster. Compactness is measured by the birth size and isolation by the cluster's life time.

Performance of the classifier is measured by the "leaving-one-out" method. We remove one sample from the T total samples and use the remaining samples to derive a decision rule and test it on isolated sample. Repeat the process T times. Since we use K-nearest neighbor algorithm, we measure performance by counting the number of neighbors out of K about the test sample that belong to the correct classification of that sample. In this study K has been taken as 3.

When the range is small (100-550 meters) there are two options for target classification. We can either minify the image with respect to some standard range so that the inside structure of the target is not visible in the image or we can make use of the structural information present in the image. A simple example of structure utilization could be the localization of the wheels and turrets of the vehicles. Important classification parameters might be the size of the wheels and the relative position of the wheels. Circles corresponding to the wheels could be found by using least squares, RANSAC paradigm or Hough transform techniques. Mission oriented information and the resulting structural information can be used in the decision making. Thus we have two sets of classification results, one from a statistical classifier and the other from

a structural block. The final decision can be made on a simple set of rules based on range and the confidence of the classification decisions from these two sources. This is useful in missiles with fiber optics.

IV. Results and discussions

The training data consisted of 420 images and testing data of 60 images. The testing set was separate from the training set. The classes in the training set were tank, truck, jeep, APC and clutter. Since the number of nontank target types was not enough to obtain statistically meaningful results, we considered only two classes, the tank and the clutter class. Semantic information was not used and only the targets with a range greater than 550 meters were considered. Algorithm (B) was used for target detection and segmentation. Range R and the minification factor used are given below,

R(meters)	Minification	R(meters)	Minification
R <550	No minification	1340 $\leq$R <2700	2
550 $\leq$R <670	8		
670 $\leq$R <1340	4	2700 $\leq$R	1

Probability of target detection (classification) is defined as the ratio of total number of targets correctly detected (classified) in the testing set divided by the total number of targets in the testing set. False alarm per frame is defined as the total number of false alarms divided by the total number of images in the testing set. These three parameters give the overall performance of the auto cuer. Probability of detection was found to be 85%. If the centroid of the detected and segmented target lies within a 10x10 window of the known target location, the target is said to have been detected. Detection failures are attributed to several reasons: overlapping targets, targets which are partly hot and partly cold, targets obscured by dirt, smoke etc. The probability of correct classification was 80% on the detected targets. Thus the overall cuer classification accuracy was 68%. False alarms were slightly more than 1 per frame. The reasons for this are that the filter used provided a fixed number of responses (6) at every frame and the features were computed on the minified image. False alarms can be reduced using Algorithm (A). It is currently being tested. All the algorithms are integrated into a software package and the algorithms run in an automatic mode. CPU time to process each 512x512 image was 2 minutes on a VAX 11/780.

References

[1] D.C. Dorrough, V.F. Pizzurro, B. Bhanu and J.R. Pasek, "A Multi-Sensor Multi-Mode Tracker Approach to Missile Ship Targeting," Proc. Tri-Service Workshop on Missile Ship Targeting, Naval Research Laboratory, Washington, DC, August 10-12, 1982.

[2] M. Burton and C. Benning, "Comparison of Imaging Infrared Detection Algorithms," Infrared Technology for Target Detection and Classification, Proc. SPIE 302, pp. 26-32.

[3] B.J. Schachter, "A Survey and Evaluation of FLIR target Detection/Segmentation Algorithms," Proc. DARPA Image Understanding Workshop, Sept. 1982, pp. 49-57.

[4] R.L. Hartley et al., "Segmentation of FLIR Images: A Comparative Study,"Proc. DARPA Image Understanding Workshop, Sept. 1982, pp. 323-341.

[5] L.M. Rubin and D.Y. Tseng, "Automatic Target Cueing," Proc. 4th Int. Joint Conference on Pattern Recognition, Nov. 1978, pp. 774-779.

[6] G.E. Tisdale, "The AUTO-Q Digital Image Processor for Target Acquisition, tracking, and Handoff," Paper presented at IRIS'80, Seattle, Washington, May 1980.

[7] D.E. Soland and P.M. Narendra, "Prototype Automatic Target Screener," Smart Sensors, Proc. SPIE 178, pp. 175-184, April 1979.

[8] C.H. Chen and C. Yen, "Object Isolation in FLIR Images Using Fisher's Linear Discriminant," Pattern Recognition, Vol. 15, No. 3, pp. 153-159, 1982.

[9] O.R. Mitchell and S.M. Lutton, "Segmentation and Classification of Targets in FLIR Images," Image Understanding Systems and Industrial Applications, SPIE Vol. 155, pp. 83-90, 1978.

[10] T.S. Yu and K.S. Fu, "Recursive Contextual Classification Using a Spatial Stochastic Model," Pattern Recognition, Vol. 16, No. 1, pp. 89-108, 1983.

[11] L.G. Minor and J. Sklansky, "Detection and Segmentation of Blobs in Infrared Images," IEEE Trans. Systems, Man and Cyber., March 1981, pp. 216-232.

[12] D.L. Milgram and A. Rosenfeld, "Algorithms and Hardware Technology for Image Recognition," Final Report to U.S. Army Night Vision & Electro-Optics Lab., Fort Belvoir, Virginia, March 31, 1978.

[13] B. Bhanu and O.D. Faugeras, "Segmentation of Images Having Unimodal Distributions," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 4, July 1982, pp.408-419.

[14] G.H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. on Computers, Vol. C-21, Feb. 1972, pp. 195-201.

[15] G. Goertzel, "An Algorithm for the Evaluation of Finite Trigonometric Series," Amer. Math. Monthly, Vol. 65, Jan. 1958, pp. 34-35.

[16] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press, New York, 1972.

[17] R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," John Wiley & Sons, 1973.

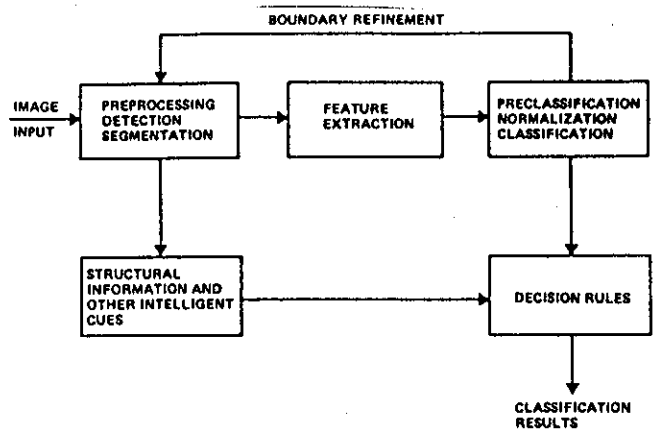


Fig. 1. Block diagram of the Intelligent Automatic Target Cuer (IATC)

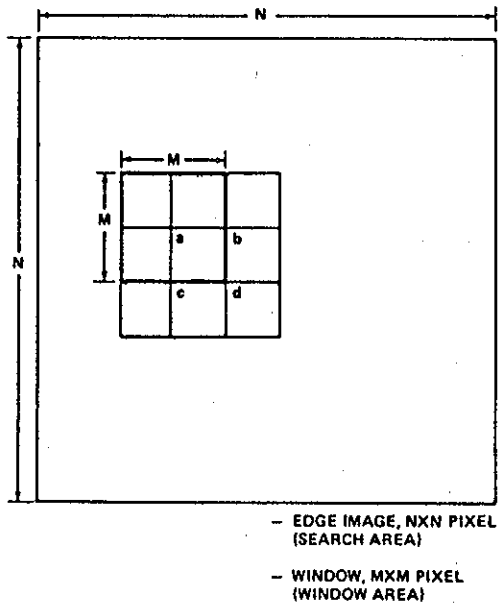
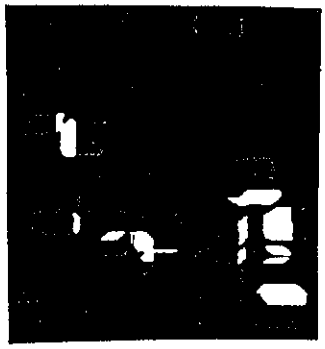


Fig. 3. Preprocessing step

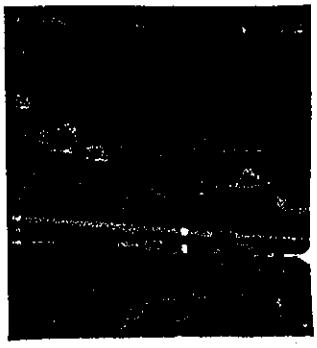
Fig. 2. Window definition of the preprocessing step

(a) A FLIR image

(b) Thresholded Sobel



(c) Mask image



(d) Selected overlap

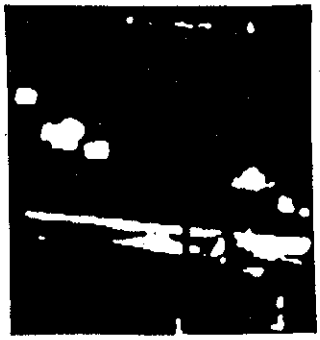


Fig. 4. Segmentation

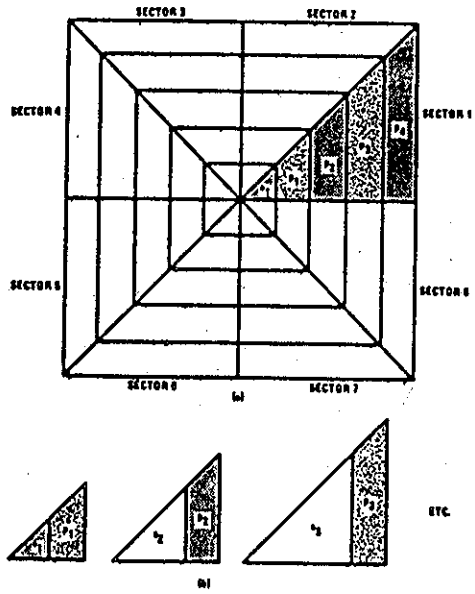


Fig. 5. (a) Filter mask definition
(b) Contrast function domain definitions for sector 1

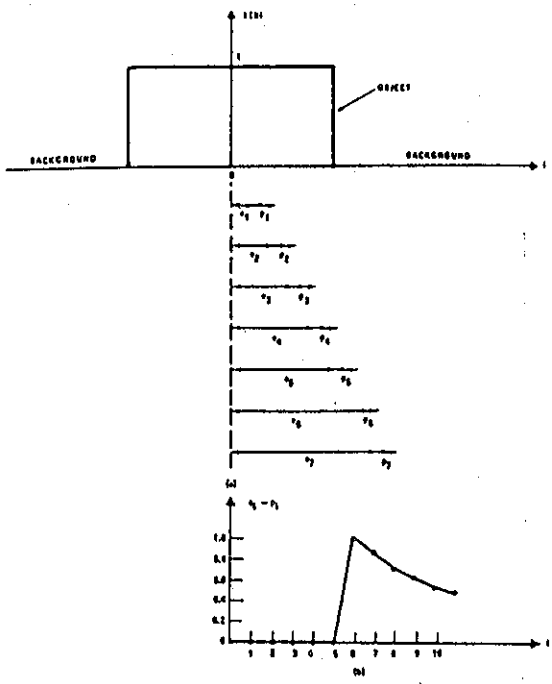


Fig. 6. (a) Object versus background distribution
(b) Contrast function $(a_i - p_i)$

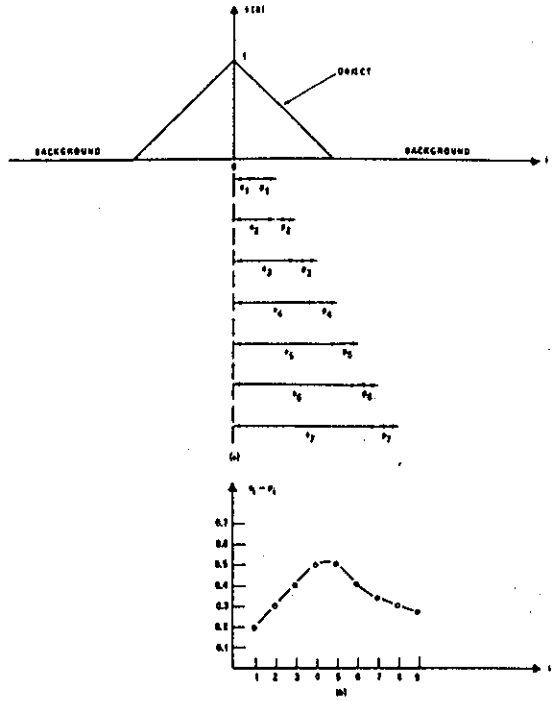


Fig. 7. (a) Object versus background distribution
(b) Contrast function $(a_i - p_i)$

(a) FLIR image (b) Target localization and segmentation

Fig. 8.

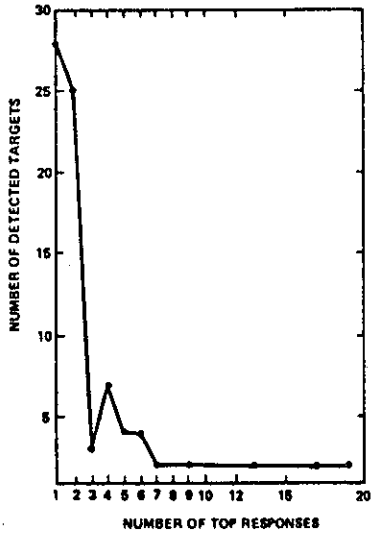


Fig 9. Number of detected targets versus number of top response

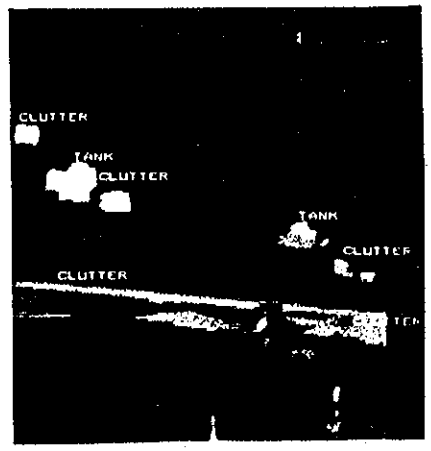


Fig. 10. Classification results for the image in fig. 3(a)

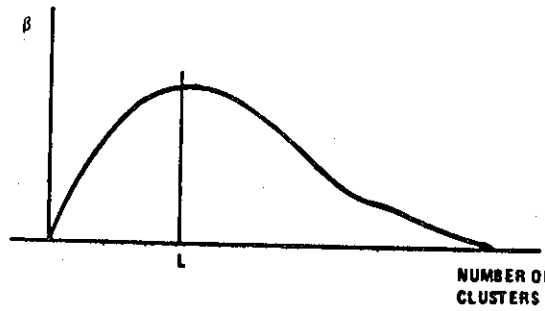


Fig. 11. A clustering quality measure