Modeling Clutter and Context for Target Detection in Infrared Images

Songnian Rong and Bir Bhanu
College of Engineering
University of California
Riverside, CA 92521

Abstract

In order to reduce false alarms and to improve the target detection performance of an automatic target detection and recognition system operating in a cluttered environment, it is important to develop the models not only for man-made targets but also of natural background clutters. Because of the high complexity of natural clutters, this clutter model can only be reliably built through learning from real examples. If available, contextual information that characterizes each training example can be used to further improve the learned clutter model. In this paper, we present such a clutter model aided target detection system. Emphases are placed on two topics: (1) learning the background clutter model from sensory data through a self-organizing process, (2) reinforcing the learned clutter model using contextual information.

1 Introduction

Automatic Target Detection and Recognition (ATD/R) is a process that takes as input a collection of sensory data (normally in the form of images and associated auxiliary data) from one or multiple sensors, preprocesses the data, detects all potential target areas in the data, and finally recognizes the identity and pose of the real target(s) [2, 8]. It has been recognized that ATD/R is a challenging application for the general techniques developed by image processing and computer vision communities [1]. There are several reasons that contribute to this challenge: (a) a target may appear in many different backgrounds and it tends to be mixed up with its surroundings, (b) signatures of a target strongly depends upon the background surrounding the target and environmental conditions, and (c) signatures of a target are generally not repeatable. Since the ATD/R algorithms are commonly used in a sequential manner, any target we fail to detect during the detection stage will be lost forever. As a result, in the detection stage, it is desired to single out every suspicious target area (region-of-interest) in the image, even at the cost that it may include some false target areas by doing so. Then it is the responsibility of the following recognition stage to verify the identity of each real target and to filter out the false targets. An ideal ATD/R system is the one that (a) does not miss any potential target area in the detection stage, and (b) does not verify any non-target area as target in the recognition stage.

To achieve the goal of high detection probability and simultaneous low false-alarm rate, we present a new strategy called Background Model Aided Target Detection and Recognition (BMATDR). The main idea of BMATDR is to use explicit background clutter models, as well as target models, throughout the ATD/R process. The background clutter model in the BMATDR system is represented by a bank of statistical models, each of which is constructed using a different image feature group. In the rest of this paper, we will use *Background Model Bank (BMB)* and *background model* interchangeably with the former emphasizing the background clutter model in the BMATDR system.

Another way to improve the performance of an ATD/R or more generally, a computer vision system is to use contextual information and other knowledge in the recognition process [12]. In a typical ATD/R mission, there are many variables that can affect the performance of an ATD/R system. Sherman et al. [8] categorized 41 such variables into five classes — background parameters, target parameters, platform dynamics, atmospherics and sensor parameters. In this paper, these variables are referred to as *contextual parameters* because they provide non-imagery information associated with each sensed image. A certain setting of these contextual parameters is referred to as a *contextual condition.* We use a reinforcement learning-based approach to associate context with clutter models to improve target detection performance.

2 Feature groups for characterizing natural background clutters

Previous work has shown that image features can be used to address two different ATD/R tasks: (a) quantify the clutter in infrared images and use the clutter measure to understand how the clutter affects the detection performance in a target detection system with man-in-the-loop [10, 13], (b) build statistical models for different natural backgrounds, and use these models in an ATD/R system [6]. Our BMATDR approach was developed to accomplish the second task. Since a natural background only makes sense to human eyes when a sufficient area of that background has been seen, region-based features are more suitable for characterizing a natural background than global or pixel

1063-6919/96 $5.00 © 1996 IEEE

based features. To facilitate the discussion in this section, we introduce two terms, **feature cell** and **feature cell size**, which will be used in the following discussions.

Definition: *A feature cell is a rectangular region within the image from which an image feature is computed. Feature cell size measures the two dimensions of this rectangular region in pixels.*

2.1 Gabor transform-based features

Features in the frequency domain have been widely used for accomplishing tasks like texture segmentation and texture classification. A special difficulty associated with ATD/R applications is that in most situations, a target of interest constitutes only a small part of the sensed image. Therefore, if a feature is based on global transforms like the Fourier transform, it is very likely that the existence of the target may not affect the spectrum to an extent that it is clearly discernible. But if the spectrum can be localized to an area whose size is comparable to the size of a target, the spectrum would have a recognizable variation when the "attention window" is placed on top of the target. One approach to achieve these localized spectra is by using the Gabor transform. Gabor transform can be considered as a special case of short-range Fourier transform. It decomposes an input image into basis functions which are localized both in spatial and frequency domains. This property is particularly desirable for ATD/R application. To compute the discrete Gabor transform of an image, we implemented an algorithm which makes the otherwise complex computation more efficient [14]. The 2-D Gabor Elementary Functions (GEF) are defined in (1),

$$g_{mnrs}(x,y) = g(x-mM, y-nM)\cdot e^{A(x,y,r,s)} \quad (1)$$

$$A(x,y,r,s) = 2\pi i\left[\frac{h(r)h(x)}{M} + \frac{h(s)h(y)}{M}\right] \quad (2)$$

where $h(t) = t - \frac{M-1}{2}$, and M is the feature cell size which determines the spatial resolution of the transform. Function $g(u,v)$ is the window function that localizes the GEF in spatial domain. For a $KM \times KM$ square image $Im(x,y)$, $x = 0,1,\cdots,KM-1$, and $y = 0,1,\cdots,KM-1$, the 2-D discrete Gabor transform can be expressed as

$$Im(x,y) = \sum_{m=0}^{K-1}\sum_{n=0}^{K-1}\sum_{r=0}^{M-1}\sum_{s=0}^{M-1} a_{mnrs}\, g_{mnrs}(x,y) \quad (3)$$

where a_{mnrs} is the Gabor coefficient corresponding to $g_{mnrs}(x,y)$. If $g_{mnrs}(x,y)$ is separable, which is the case when we use 2-D Gaussian function as the window function, the Gabor decomposition can be written in matrix form as $Im = G\cdot A\cdot G^T$, where A is the $KM\times KM$ Gabor coefficient matrix. Operator matrix G and its transpose can be precalculated based on the selected spatial resolution. To compute the transform, we have

$$A = G^{-1}\cdot Im\cdot (G^T)^{-1} \quad (4)$$

Each element of matrix A is complex and represented by its real part ar_{rs}^{mn} and imaginary part ai_{rs}^{mn}. Here m and n are the two spatial indexes that locate the element in spatial domain, while r and s are the two frequency indexes that locate the element in frequency domain. The amplitude spectrum of the transform is given by

$$a_{rs}^{mn} = \left((ar_{rs}^{mn})^2 + (ai_{rs}^{mn})^2\right)^{1/2} \quad (5)$$

Two feature groups are constructed directly from the Gabor amplitude spectrum:

Two-level mean amplitudes. The *first* group consists of two features — *MA*1 and *MA*2, which are the first and second level mean amplitudes of a feature cell. To compute them, the discrete Gabor amplitudes are first sorted in a decreasing order. If $\hat{a}(i)$ denotes the sorted Gabor amplitudes of a $M\times M$ feature cell, $i = 0,1,\cdots,N = M^2-1$, then

$$MA1 = \frac{2}{N+1}\sum_{i=0}^{N/2}\hat{a}(i),\quad MA2 = \frac{4}{N+1}\sum_{i=0}^{N/4}\hat{a}(i) \quad (6)$$

Moments of the Gabor amplitude spectrum. The *second* group consists of two features — the first order moments of the Gabor amplitude spectrum with respect to ω_x and ω_y axis.

2.2 Self-similarity in natural scenes

It has been recognized that images of natural scenes exhibit statistical self-similarity at different scales [9]. Field proposed a feature which relates this fractal phenomenon in an image to the amplitude spectrum of the image [3, 4]. His conclusion is that if a natural scene exhibits self-similarity, its amplitude spectrum will fall off in a rate of $1/k$ with respect to the spatial frequency k. If we plot the logarithmic amplitude vs. logarithmic frequency, the plot will be a straight line with a slope of -1.

Since one of the major functions of our background model is to discriminate natural backgrounds from man-made objects, this feature can be very useful if images of the man-made objects do not show this -1 slope. Again, since targets may constitute only a small part of the image, we use Gabor transform instead of the Fourier transform as the basis for computing this feature. To compute the slope feature, we first need to compute the average Gabor amplitude at each frequency over all the available orientations. Let $\bar{a}_k$ denote the average Gabor amplitude at spatial frequency f_k ($k = 1,2,\cdots,M-1$), a least-square method is used to fit a straight line to the data pairs of $(\log\bar{a}_k, \log f_k)$. The slope of the fitted line (*SLP*) and the maximum error of this linearization (*ERR*) are then computed as features using (7) and (8),

$$SLP = \frac{S_f\cdot S_a - (M-1)S_{fa}}{{S_f}^2 - (M-1)S_{f2}} \quad (7)$$

$$ERR = \left\{\max_{k=1}^{M-1} W_k - \min_{k=1}^{M-1} W_k\right\}\cos\left(\tan^{-1}(SLP)\right) \quad (8)$$

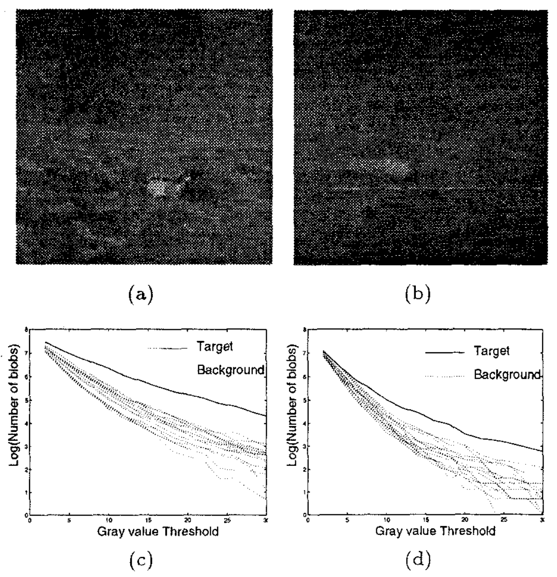

Figure 1: Two examples of using the local statistics of geometric elements (*LSGE*) feature to detect man-made targets in natural background. (a), (b) 200×200 FLIR images. (c), (d) The *LSGE* feature computed using 50 × 50 feature cell size.

where

$$S_f = \sum_{k=1}^{M-1} \log(f_k) \quad S_a = \sum_{k=1}^{M-1} \log(\bar{a}_k) \tag{9}$$

$$S_{fa} = \sum_{k=1}^{M-1} \log(f_k) \cdot \log(\bar{a}_k) \tag{10}$$

$$S_{f2} = \sum_{k=1}^{M-1} \log(f_k)^2 \tag{11}$$

$$W_k = SLP \cdot f_k + B - \bar{a}_k \tag{12}$$

$$B = \frac{S_a}{M-1} - SLP \cdot S_f \tag{13}$$

2.3 Local statistics of geometric elements

In FLIR images, a target normally appears as one or several blobs close to each other, but, due to varying contrast, the target-background border is not always distinct. This observation inspired the idea of using the number of blobs and the average size of blobs in a feature cell as new features. To find out the blobs in a selected feature cell, we first run an adaptive region growing algorithm over the image with different thresholds. Statistics of the size, shape, and population of blobs in a feature cell are then computed as features. We have found in our experiments that the change of these local statistics with respect to different thresholding values reveals some important characteristics of natural clutters. In this paper, we are focusing on the *number of the blobs* vs. the *gray value threshold*. When we plot these two variables on a semi-logarithmic plot, it can be seen that the lines corresponding to the target regions exhibit smaller slope than the lines corresponding to the background regions. Two examples using the second generation FLIR images are shown in Figure 1. The slope can be computed according to (14),

$$SLP_b = \frac{S_k \cdot S_b - N \cdot S_{kb}}{S_k^2 - N \cdot S_{k2}} \tag{14}$$

where

$$\begin{aligned} S_k &= \textstyle\sum_{k=1}^{N} k & S_b &= \textstyle\sum_{k=1}^{N} \log(n_k) \\ S_{kb} &= \textstyle\sum_{k=1}^{N} k \cdot \log(n_k) & s_{k2} &= \textstyle\sum_{k=1}^{N} k^2 \end{aligned} \tag{15}$$

N is the highest threshold used, and n_k denotes the number of blobs in a feature cell under threshold k. Two feature groups have been designed based on the local statistics of geometric elements (*LSGE*). The first one is called the ***absolute*** *LSGE*, which contains the SLP_b in (14) and the 10-blob threshold K_{10}, the threshold that allows the adaptive-region-growing algorithm to generate exactly 10 blobs in the given feature cell. The number 10 used in this feature is selected since we can "observe" on the average 10 blobs on a man-made target in the training images. The second feature is the *relative LSGE*, which is applied only when the sensory image contains more than one feature cells.

3 Learning the background model through a self-organizing process

Although many papers in the literature have used known statistical distributions in their analysis of natural clutters in FLIR images, there is no strong evidence that thermal natural clutters possess a certain statistical distribution [10]. Instead of artificially assigning a distribution model to background models, we construct our BMB from real images through a supervised learning process. Since reliable statistical models can only be obtained through analysis of a large population of samples, space and time complexities of algorithms become a major concern when selecting a learning scheme. In our approach, each BMB member is represented by a self-organizing map (SOM), which captures the distribution of the training data. By controlling the size of the SOM, we can easily control the space and time complexity of the learning process.

3.1 Self-organizing and near-miss injection algorithm

Kohonen's self-organizing map algorithm makes it possible to discover the distribution of the training data as well as the cluster center of the data, enabling us to find a certain class in the feature space. This information can be used to construct a more sophisticated statistical classifier, which would be more powerful than the conventional K-means algorithm. However, when we applied this algorithm to our target detection problem, we encountered the feature overlapping problem. Because of the complexity of natural background, statistical features extracted from a target area sometimes overlap with those extracted from the background areas. To build an applicable background model that can classify these two classes (background and target), we developed an algorithm that combines Kohonen's SOM algorithm with a novel near-miss injection algorithm [11]. This algorithm has been found effective in our experiments for solving the feature overlapping problem.

The first step of our learning algorithm is to use the Kohonen's algorithm to train the SOM by using positive training examples. By positive examples we mean pure background images that have no target embedded in them. To make the learning process autonomous, i.e. without the need for humans intervention, a metric reflecting the SOM's ordering is needed so that the algorithm can determine how well the SOM has been trained, and thus determine whether it is time to terminate the learning process. In our research, we developed two metrics to describe the ordering of a SOM. The first one is based on the proved asymptotic convergence property of the SOM, and the second one is based on a direct analysis of the distortion of the SOM grid.

When the disorder index is below a pre-selected threshold the SOM is in a well ordered state, and a conventional SOM algorithm can terminate its learning process. In our approach, at this time the learning process will go into the second stage — refining those ambiguous regions in the SOM by using the near-miss injection algorithm and negative examples. By ambiguous regions we mean regions where features of different classes (e.g. background and man-made targets) overlap. The near-miss injection algorithm runs inside a loop which contains the following two steps: (a) given a negative training vector $\mathbf{y}$, search the "hitting" neuron h within the $N \times N$ SOM using (16).

$$||\mathbf{y} - \mathbf{w}_h|| = \min_i ||\mathbf{y} - \mathbf{w}_i|| \tag{16}$$

where $\mathbf{w}_x$ is the weight vector of neuron x, and $i = (r, l)$ while $r, l = 1, 2, \cdots, N$.
(b) update the weight vectors according to

$$\mathbf{w}_i^{t+1} = \begin{cases} \mathbf{w}_i^t + \frac{\beta^t \mathbf{u}}{(||\mathbf{y}-\mathbf{w}_i^t||+1)^2} & \text{for } \mathbf{i} \in N_h \\ \mathbf{w}_i^t & \text{otherwise} \end{cases} \tag{17}$$

$$\mathbf{u} = \begin{cases} \frac{\mathbf{y}-\mathbf{w}_i^t}{||\mathbf{y}-\mathbf{w}_i^t||} & \text{if } ||\mathbf{y} - \mathbf{w}_i^t|| \neq 0 \\ \frac{\mathbf{y}-\bar{\mathbf{w}}_i^t}{||\mathbf{y}-\bar{\mathbf{w}}_i^t||} & \text{if } ||\mathbf{y} - \mathbf{w}_i^t|| = 0 \end{cases} \tag{18}$$

$$\bar{\mathbf{w}}_i = \frac{1}{4}\sum_j \mathbf{w}_j, \text{ j} \in \text{4-neighbor of i} \tag{19}$$

where β^t is the learning rate for this near-miss injection algorithm, it should decay with time, and we use the same decay function that is used in Kohonen's algorithm. N_h is the predefined neighborhood of the hitting neuron.

3.2 The classification criterion

Since the target detection process has been formulated as a 2-class classification problem (natural background vs. man-made target) in our approach, a classification criterion is needed to label a testing feature cell according to the learned background model. The criterion used in our experiments is based on the computation of two confidence values : $Conf_B$, the confidence value for a testing feature cell to be background and $Conf_T$, the confidence value for it to be man-made target. These two confidence value are computed by comparing the *Four Neighbor Average Distance (FNAD)* of a testing feature cell to $\bar{R}_P$, the average *FNAD* of all the positive training examples, and $\bar{R}_N$, the average *FNAD* of all the negative training examples. To compute the *FNAD* of a feature cell, the feature vector is first projected into the learned background model which is represented by a SOM. The hitting neuron is found next, and the four neighbors of this hitting neuron are identified. The *FNAD* can then be calculated according to

$$FNAD = \frac{\sum_{i=1}^4 r_i}{4} \tag{20}$$

where r_i is the Euclidean distance between the testing feature vector and one of the 4-Neighbor neurons. From (20) we have

$$\bar{R}_P = \frac{\sum_{n=1}^{N_P}\sum_{i=1}^4 r_i^n}{4N_P} \tag{21}$$

$$\bar{R}_N = \frac{\sum_{n=1}^{N_N}\sum_{i=1}^4 r_i^n}{4N_N} \tag{22}$$

where N_P is the number of positive training examples used in constructing the background model, and N_N is the number of near-misses used. The two confidence values are

$$Conf_B = \max\left(1, \frac{4\bar{R}_P}{\sum_{i=1}^4 r_i}\right) \tag{23}$$

$$Conf_T = \max\left(1, \frac{\sum_{i=1}^4 r_i}{4\bar{R}_N}\right) \tag{24}$$

Given $Conf_B$ and $Conf_T$, the classification of the testing feature cell is obtained as,

$$l = \begin{cases} \textit{Background} & \text{if } Conf_B > Conf_T \\ \textit{Target} & \text{Otherwise} \end{cases} \tag{25}$$

The classification confidence C_C is assigned the value of $Conf_B$ or $Conf_T$ accordingly.

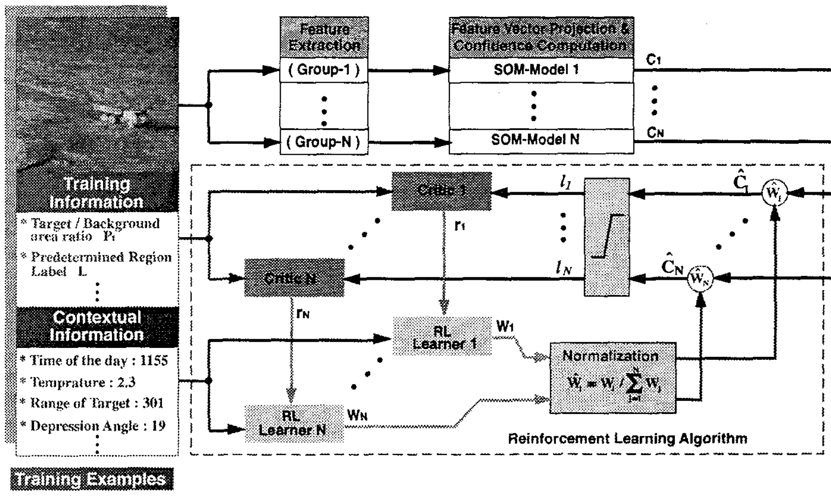

Figure 2: Context reinforced background model.

4 Reinforcement of the background clutter model using context

To be practically applicable, an ATD/R system must be able to detect targets under different contextual conditions. One way to achieve this goal is to use a learning technique to associate contextual parameters with the performance of each feature group in the background model bank. The rational behind this association is that if a feature group can effectively detect man-made objects under a given contextual condition, then it tends to be effective for images taken under similar contextual conditions. Since a human supervisor cannot provide any assistance to the ATD/R system in finding this association, except telling the system whether it is doing a "good job" with respect to a specific testing image, the most suitable learning technique for this task is the *reinforcement learning* scheme [7]. Figure 2 shows how this reinforcement learning subsystem fits into the background modeling process. If a feature group has a good performance under a certain contextual condition, its detection result deserves a larger weight in the background model bank under similar contextual conditions. So, the *context - performance* relationship can be replaced by a *context - weight* relationship, which is more complaint for being integrated into an automatic learning system. To facilitate the discussion, we define the following terms, which are used to formulate the following stochastic reinforcement learning algorithm. The superscript i ($i = 1, 2, \cdots, n_C$) refers to the i-th contextual parameter available to our BMATDR system. The subscript j ($j = 1, 2, \cdots, n_F$) refers to the j-th feature group in the background model bank.

- *Contextual Parameter* (c^i) is a scalar that quantifies a specific aspect of a contextual condition, it can be defined over continuous or discrete values.
- *Contextual vector* ($\mathbf{C}$) is a vector with each element being c^i, a contextual parameter.
- *Weight Vector* ($\mathbf{W}$) is a real value vector with each element being w_j, the weight of the j-th feature group.

Our learning problem can then be defined as:
Given a set of training images that cover the entire range of available contextual parameters, with the background model bank having been built as a collection of SOM's, associate with each BMB member SOM_j a stochastic transform function T_j, such that $w_j = T_j(\mathbf{C})$.

The transform function T_j is stochastic because the Context($\mathbf{C}$) - Weight($\mathbf{W}$) relationship cannot be described by a deterministic function, there are always exceptional cases due to the high complexity of the real-world.

4.1 Stochastic reinforcement learning algorithm

The reinforcement learning algorithm we selected for learning the context-performance relationship of each BMB member is shown in Figure 3. It is based on the stochastic real valued reinforcement learning algorithm, which is developed by Gullapalli for training a single actor using reinforcement as feedback [5]. This algorithm allows the system to learn outputs that take on real values. Since the performance of a feature group is best described as a real number, normally between 0 and 1, with 1(0) representing the best (worst) performance, this algorithm meets our requirement very well. However, since we want to use this algorithm to cooperate the actions of n_F BMB members to achieve a better performance in target detection, we need to make changes to the algorithm, so that it can handle multiple actors.

In the stochastic reinforcement learning algorithm, T_j is implemented as a random number generator according to the normal distribution. The mean μ_j, and standard deviation σ_j are determined by two internal vectors, $\mathbf{\Phi}_j$ and $\mathbf{\Psi}_j$ according to the following formula:

$$\mu_{j,n} = \mathbf{\Phi}^t_{j,n} \cdot \mathbf{C}_n \qquad (26)$$

where n denotes the n-th iteration of the learning process.

$$\sigma_{j,n} = 1 - \hat{r}_{j,n} \qquad (27)$$

where $\hat{r}_{j,n}$ is the estimated reinforcement feedback for the j-th feature group after n iterations of learning. It can be estimated using the following formula:

$$\hat{r}_{j,n} = 1 - f(\mathbf{\Psi}^t_j \cdot \mathbf{C}_n) \qquad (28)$$

where function $f(\cdot)$ often takes the form of

$$f(a) = \frac{1}{1+e^{-a}} \qquad (29)$$

which maps the real line onto the interval $(0,1)$. Once the two parameters (μ_j, σ_j) are available, the weight for the j-th feature group can be computed by passing μ_j and σ_j to a random number generator:

$$w_j \sim N(\mu_j, \sigma_j) \qquad (30)$$

So, in the learning system, each context-to-weight transform function T_j is actually "remembered" as two internal vectors $\mathbf{\Phi}_j$ and $\mathbf{\Psi}_j$. Starting with randomly selected initial values, these two internal vectors learn to represent the Context($\mathbf{C}$) - Weight($\mathbf{W}$) relationship by updating themselves according to (31) and (32).

$$\mathbf{\Phi}_{j,n+1} = \mathbf{\Phi}_{j,n} + \sigma_{j,n}(r_{j,n} - \hat{r}_{j,n})(w_{j,n} - \mu_{j,n})\mathbf{C}_n \quad (31)$$

$$\mathbf{\Psi}_{j,n+1} = \mathbf{\Psi}_{j,n} + \rho_n(r_{j,n} - \hat{r}_{j,n})\mathbf{C}_n \qquad (32)$$

where $r_{j,n} = g(\mathbf{P}_{j,n})$ is the reinforcement provided by a critic function $g(\cdot)$ for the the j-th feature group in the n-th detection trial. Vector $\mathbf{P}$ is the performance vector defined below. It is used by the *critic* to judge the performance of the system after a detection trial. Figure 3 shows the data flow path of the modified stochastic reinforcement learning algorithm. Gullapalli has provided a convergence proof of the single actor reinforcement learning algorithm [5]. The proof for the convergence of the multiple actor case can be performed in a similar manner.

4.2 Implementation concerns

To utilize this complex learning scheme to solve the previously defined Context($\mathbf{C}$) - Weight($\mathbf{W}$) problem, we have to make several implementation decisions:

Selection of contextual parameters. It is difficult to deal with 41 contextual parameters at the same time without organizing context in some hierarchical manner. One simple practical approach is to select a subset of important contextual parameters from the available context. In our implementation, we selected 4 parameters to form the contextual vector. These parameters are

- t : *Time of the day.*
- d : *Depression angle.*
- s : *Range to the target.*
- p : *Air temperature.*

In order to make the inner product in (26) and (28) meaningful, we used relative values of the contextual parameters in constructing the contextual vector. The relative value of d, for example, is $\frac{d}{d_{max}-d_{min}}$, where d_{max} and d_{min} are the maximum and minimum depression angle encountered in the training images.

Performance vector P. Since all our features are region based features, given a testing image, the image is first divided into feature cells based on the *range-to-target* information. The detection result is a label vector $\mathbf{l}$, with each of its element being the label of a feature cell. The label of the top-left feature cell is the first element in $\mathbf{l}$, and the label of rest of the feature cells are entered in a row-first manner. The easiest way to describe the performance of the detection is to compare $\mathbf{l}$ with the label vector $\mathbf{L}$ given by the learning supervisor. Thus, the performance vector $\mathbf{P}$ can simply be $\mathbf{P} = \mathbf{l} - \mathbf{L}$.

Selection of the critic function. Since we are dealing with a two class classification problem, both $\mathbf{l}$ and $\mathbf{L}$ can be a bit vector. A simple metric for the detection performance is obtained by examining the number of bits being set to 1 in $\mathbf{P}$. So, the reinforcement feedback can be computed as follows:

$$r_{j,n} = \sum_{k=1}^{Nb} p^k_{j,n}/Nb \qquad (33)$$

where Nb is the total number of feature cells within the testing image. $p^k_{j,n}$ denotes the k-th element of vector $\mathbf{P}_{j,n}$ which, in turn, describes the performance of the j-th feature group in the n-th detection trial.

4.3 Experimental results

In this experiment, we compare the detection performances of two background model banks learned by using the SOM and near-miss injection algorithm and 20 FLIR training images. In the *first* background model bank, no contextual information is involved. Thus all the BMB members (each represented by a 5×5 self-organizing map) in the background model bank are treated as equally important. Since we have five BMB members in our background model bank, the weight of each BMB member is 0.2. The *second* background model bank contains the same BMB members that constitute the first background model bank. In addition, contextual information is used to reinforce this background model bank. By applying the stochastic reinforcement learning algorithm given in Section 4.1 for 200 iterations, a relationship is set up between the importance of each BMB member and the underlying contextual condition. This relationship is represented by the two internal vectors Φ and Ψ associated with each BMB member.

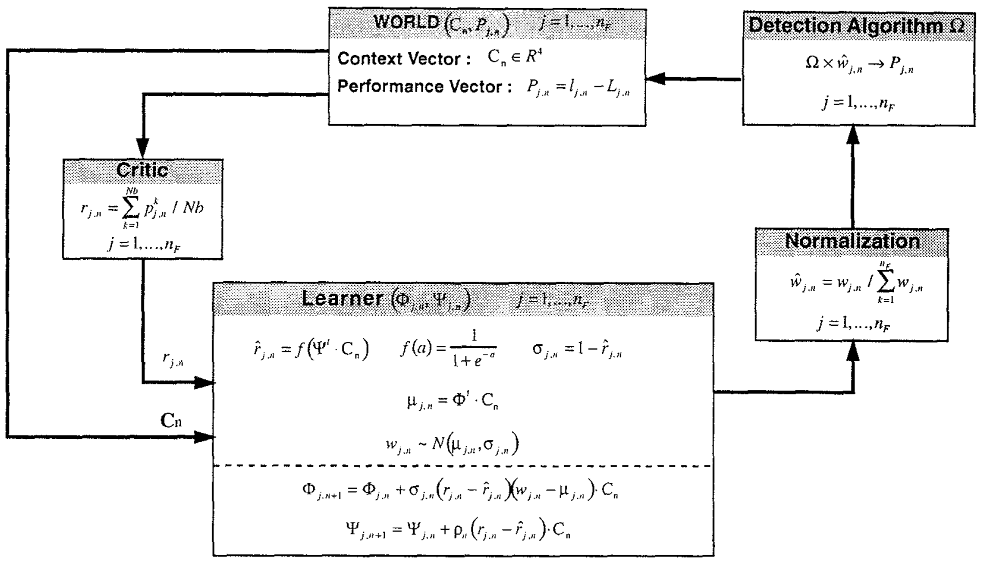
WORLD $(C_n, P_{j,n})$ $j=1,\dots,n_F$
Context Vector : $C_n \in R^4$
Performance Vector : $P_{j,n} = l_{j,n} - L_{j,n}$
Detection Algorithm Ω
$\Omega \times \hat{w}_{j,n} \rightarrow P_{j,n}$
$j=1,\dots,n_F$
Critic
$r_{j,n} = \sum_{k=1}^{Nb} p_{j,n}^k / Nb$
$j=1,\dots,n_F$
Normalization
$\hat{w}_{j,n} = w_{j,n} / \sum_{k=1}^{n_F} w_{j,n}$
$j=1,\dots,n_F$
Learner $(\Phi_{j,n}, \Psi_{j,n})$ $j=1,\dots,n_F$
$\hat{r}_{j,n} = f(\Psi^t \cdot C_n)$ $f(a) = \frac{1}{1+e^{-a}}$ $\sigma_{j,n} = 1 - \hat{r}_{j,n}$
$\mu_{j,n} = \Phi^t \cdot C_n$
$w_{j,n} \sim N(\mu_{j,n}, \sigma_{j,n})$
$\Phi_{j,n+1} = \Phi_{j,n} + \sigma_{j,n}(r_{j,n} - \hat{r}_{j,n})(w_{j,n} - \mu_{j,n}) \cdot C_n$
$\Psi_{j,n+1} = \Psi_{j,n} + \rho_n(r_{j,n} - \hat{r}_{j,n}) \cdot C_n$
$r_{j,n}$
Cn

Figure 3: The stochastic reinforcement learning algorithm.

After the construction of these two background model banks, another 20 second generation FLIR images are used as testing images. For each testing image, an accompanied image is built by removing rows and columns, equal to one-half the size of the selected feature cell, from all the four sides. By using the *first* background model bank and the classification criterion given in Section 3.2, out of the 217 feature cells in the 20 testing images, we achieved a 100% detection rate and a 12% false alarm. These 20 testing images and their accompanied images are then classified by using the *second* background model bank. The detection rate obtained is 100% and false alarm decreased by 2%. The detection confidence values of the correctly classified feature cells in both experiments are shown in Figure 4(a). Figure 4(b) shows the confidence values of the misclassified feature cells in both experiments. From these two figures we can see that by reinforcing the background model bank using contextual information, the confidence values of the correctly classified feature cells increase, and the confidence values of the misclassified feature cells decrease. The final effect is an improved detection performance. In Figure 4(c), the result is presented in a different way. The confidence values of the misclassified feature cells are represented by negative values, as before, the confidence values of the correctly classified feature cells are still represented as positive values. It can be seen that the context reinforced background model makes the curve of confidence values shifting upward, which produces a better detection result. We expect that further reduction in false alarms are possible by increasing the size of the training image set, which would expose the background model bank to more contextual conditions. In our experiments, because of the availability of limited experimental images, the contextual conditions of several testing images are significantly different from those of the training images.

5 Conclusions

Our work on the construction and enhancement of natural background clutter models has shown that, by introducing learning capabilities into an ATD/R system, we can build a model for the complex natural background from real images and improve it as we train the system with more examples. Contextual parameters which contain non-imagery information of the training examples can be used to enhance the background models. How beneficial the background models can be to ATD/R process depends on two factors, (a) the effectiveness of the selected features, and (b) better understanding of each feature's importance with respect to contextual conditions.

Acknowledgments

This work was supported by ARPA, AFOSR grants F49620-93-1-0624, F49620-95-1-0424 and MDA972-93-1-0010. The contents of the information do not necessarily reflect the position or the policy of the U.S. Government.

References

[1] B. Bhanu. Automatic target recognition: State of the art survey. *IEEE Trans. Aerospace and Electronic Systems*, AES-22(4):364–379, 1986.

[2] B. Bhanu and T.L. Jones. Image understanding research for automatic target recognition. *IEEE*

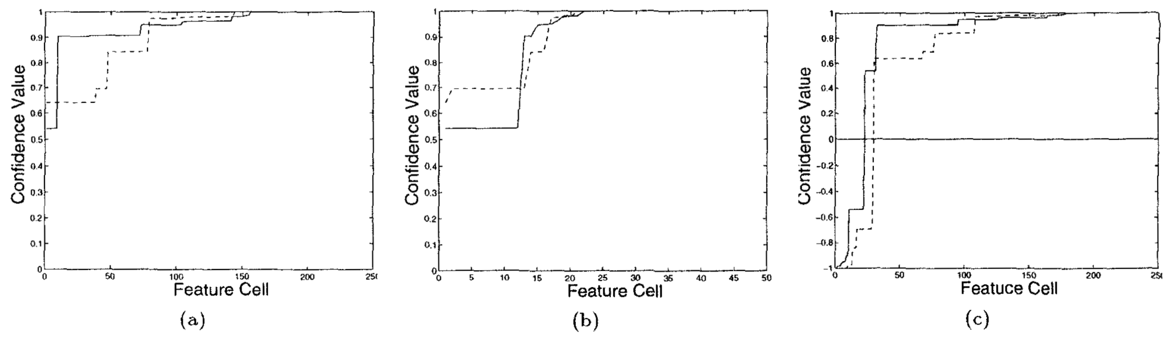

Figure 4: The improved detection performance after context reinforcement of the background model. The dashed line shows the result of using the *first* background model bank, which is not reinforced by the contextual information. The solid line represents the result of using the context reinforced background model bank. (a) correctly classified feature cells (b) misclassified feature cells (c) all the feature cells in the testing images.

Aerospace and Electronic Systems, 8(10):532–536, 1993.

[3] D.J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of Optical Society of America*, 4(12):2379–2394, 1987.

[4] D.J. Field. Scale-invariance and self-similar 'wavelet' transforms: an analysis of natural scenes and mammalian visual systems. In J.C.R. Hunt M. Farge and J.C. Vassilicos, editors, *Wavelets, Fractals, and Fourier Transforms*, pages 151–194, Oxford, 1993. Clarendon Press.

[5] V. Gullapalli. Associate reinforcement learning of real-valued functions. Technical report, Department of Computer and Information Science, University of Massachusetts, Amherst, May, 1993.

[6] M.A. Noah, P.V. Noah, J. Schroeder, B.V. Kessler, and J. Chernick. Background characterization techniques for pattern recognition applications. *Proceedings SPIE*, 1098:55–70, 1989.

[7] M.M. Kokar and S.A. Reveliotis. Reinforcement learning: architectures and algorithms. *International Journal of Intelligent Systems*, 8(8):875–894, 1993.

[8] J.W. Sherman, D.N. Spector, C.W. "Ron" Swonger, L.G. Clark, E.G. Zelnio, M.J. Lahart, and T.L. Jones. Automatic target recognition systems. In L. Shumaker, editor, *The Infrared and Electro-optical Systems Handbook*, pages 343–402. SPIE Optical Engineering Press, 1993.

[9] B.B. Mondelbrot. *The Fractal Geometry of Nature*. Freeman, 1983.

[10] W.R. Reynolds. Toward quantifying infrared clutter. *Proceedings SPIE*, 1311:232–240, 1990.

[11] S. Rong and B. Bhanu. Enhancing a self-organizing map through near-miss injection. In *Proc. of the 1995 World Congress On Neural Networks*, pages I552–I556, Washington DC, July 17-21 1995.

[12] T.M. Strat and M.A. Fischler. Context-based vision: Recognition of natural scenes. *Proceedings of 23rd Asilomar Conference on Signals, Systems and Computers*, pages 532–536, 1989.

[13] S.R. Rotman, G. Tidhar and M.L. Kowalczyk. Clutter metrics for target detection system. *IEEE Trans. Aerospace and Electronic Systems*, 30(1):81–91, 1994.

[14] J. Yao. Complete Gabor transformation for signal representation. *IEEE Trans. Image Processing*, 2(2):152–159, April 1993.