Background Clutter Rejection Using Generalized Regression Neural Networks

Charles Ralph Waters and Tony Sommese
Photon Research Associates, Inc.
170 North Country Road, Suite 1
Port Jefferson, New York 11777
(516) 331-6322
and Capt. Brian Hibbeln
National Air Intelligence Center
WPAFB, OH

Abstract—Advances in focal plane array technology has led to the development of "staring" sensors for a number of remote sensing application. Here the sensor line-of-sight (LOS) is fixed to a background point and stares at that point while radiometric measurements are collected. Inadvertent motions of the LOS result in unwanted time signals (clutter) that corrupt the measurements. This paper develops a technique that estimates these signals in the output of each focal plane detector by employing a Generalized Regression Neural Network (GRNN). The GRNN is an optimal estimator that is based on the well-known statistical concept of conditional probability. Two implementations are evaluated for removing the background. The first technique estimates the clutter signal in each detectors output based on the previous measurements from that detector. The second method trains the GRNN with the measurements from the surrounding spatial pixels on the current data frame. Both techniques were evaluated using measurement sets from an existing staring space sensor. The results show the GRNN estimates and follows the clutter signal very well with a rms error < 3% which is within the variation of the random sensor noise.

TABLE OF CONTENTS

1.0 INTRODUCTION

In a number of remote sensing applications, the primary mission of the sensor is to detect and track specific man made targets. Therefore, natural backgrounds represent a corrupting signal source of clutter noise that needs to be suppressed. Classically, the processing approaches have involved simple frame subtraction, bandpass filtering and more recently signal subspace techniques that determine the correlated portions of the measured signals which result from the background. In this paper, we investigate the feasibility of using Generalized Regression Neural Networks (GRNN) to estimate and remove the corrupting background.

Considered is the processing of staring sensor array for which the LOS of the sensor is held fixed to a point in the background scene. Because of orbital drift and structural vibrations, the sensor pointing is not maintained resulting in motion of the detector array across the radiant background. To first order, the change in the intensity for a given detector over one time sample is the product of its local background gradient and the total displacement. Therefore, the combination of the LOS motion and the local cloud gradients cause a significant noise component in the temporal intensity history for each of the detectors in the array, and it is the suppression of this background noise that is demonstrated here using a GRNN.

A GRNN is an optimal statistical estimator and is, therefore, applicable to a wide range of prediction problems. Our investigation considered a measured cloud data set from a spacebased infrared satellite. Two feature sets for the GRNN are used and contrasted. Since the detector motions for this sensor system are fairly periodic, the detector temporal intensity history has the same frequencies as the LOS motion. The first feature set exploits this periodically by using time lags in the temporal intensity history of the detector to estimate the detector intensity at future times. The idea is that if the intensity signal is periodic then a few lags (or even a single lag in the absence of random detector noise) should be sufficient to predict future values of the time series. Of course the disadvantage of this approach is that if the frequencies in the control system change and are different from what was observed during the training period then the training patterns are no longer reliable for use in predicting future intensities. Another disadvantage is that other control systems associated with sensors different from the one used here, may produce a much more random LOS history making it impossible to predict future intensity values with any number of time lags.

The second feature set, which is more generic, uses spatial correlations between detectors on the same temporal data frame and is insensitive to the frequency content of the LOS motions. Here the intensities of neighboring detectors are used to estimate on the same frame the intensity of the pixel under test. Note that no temporal information is processed making this method attractive when the motions are totally random. The underlying principle for this approach is that similar LOS displacements produce the same spatial intensity pattern in the array. Therefore, spatial patterns observed over the training frames represent those that would be seen on other frames as long as the displacement amplitudes are similar.

It should be mentioned that both feature sets are sensitive to the amplitude of the LOS displacements. The performance obtained using both sets degrades substantially if the amplitude of the displacement increases outside the training interval, since larger amplitudes produce patterns that were not previously observed, and this causes the GRNN to extrapolate which is not something it does well.

The GRNN is trained using three different methods. The first method combines Conjugate Gradient and Line Minimization techniques. If the error surface is close to quadratic then this method is the most efficient in finding the minimum square error since it uses derivative information. The second method used is Differential Evolution (DE) which performs a global stochastic search using a population of solutions. It does not use any derivative information and as a result is able to avoid getting stuck at a local minimum. The third method is a hybrid method that combines DE with Line Minimization. Here DE is used to perform a global search picking out promising regions in solution space and then Line Minimization is applied to efficiently search those regions.

All training methods produced networks that performed well both inside and outside the training region indicating that the GRNN is a viable algorithm for this application. Of the two feature sets, the spatial feature set performed better and seemed to whiten the signals eliminating all background induced noise. This is because the training period for the time lag technique did not include enough time to observe all of the lower frequency components of the signal. Since training using the temporal lags is sensitive to the signal frequencies, this approach suffered because there were temporal patterns observed in the validation region that were not previously seen in the training region.

The paper is organized as follows: in Section 2.0 a detailed description of the problem is given. Two different applications involving background removal are stated. The measured data set used in the analysis is also described here, and an example of a clutter time signal is given. Also more standard techniques that are normally used to eliminate the background are summarized for comparison purposes. Section 3.0 provides a brief description of both the theory and application of a GRNN. It also gives a detailed description of the feature sets used as well as some of their limitations. Performance results are given in Section 4.0. The results given are taken from the runs using the Conjugate Gradient training algorithm since all training algorithms provided similar results. Also, issues are listed that need to be addressed before the implementation used here can be extended to an operational algorithm. Section 5.0 provides a brief description of the training methods and the training results. Efficiency is measured in terms of the number of function evaluations and is provided for each of the feature sets and training methods along with the performance that is obtained. The network standard deviations produced from each training method are listed and compared. The final Section 6.0 summarizes the results and conclusions.

2.0 Problem Description

A major thrust in the development of remote sensing system is the utilization of staring two-dimensional arrays of detectors. Errors in controlling the attitude of the array result in each detector in the array moving across its local background gradient. This creates a temporal intensity signal in each detector that interferes with the target signals if the target is present. The major component of detector motion is translation in which all of the detectors in the array experience the same motion and therefore, their temporal background signals are correlated. The correlation between detectors is something that is exploitable by background suppression algorithms.

Background noise poises two problems to a sensor system. The first is it interferes with the ability of the sensor to pick out the detectors (or locations) that contain targets. This is the classic detection function. The second problem is that even after targets are detected, the noise interferes with the ability of the sensor to extract key intensity information needed to characterize targets. This is the typing or identification function, and the information that is desired is a highly accurate estimate of the intensity of the target.

Background suppression may be accomplished differently for each application. Classical detection algorithms typically distort the target signal making it difficult (if not impossible) to estimate the target intensity from the filtered signal. Therefore, many times after targets are detected and their locations known a different algorithm is used to clean up the signal so that a good estimate of the target intensity can be produced.

Historically, a simple background suppression technique is frame differencing. Here the detector intensity on the previous frames is subtracted from the current frame. If the control system keeps the LOS constant over the two frames then differencing removes the background. Furthermore, if the target moves or changes intensity then the associated change in the detector signal will come through the differencing filter and the target will be detected. While this approach is relatively simple and therefore, adaptable to real-time processing it is susceptible LOS motion between frames. Also, this approach distorts the target signal removing much of the low frequency components.

More sophisticated techniques involve processing a batch of frames and calculating a temporal correlation matrix. Since the detectors see the same motions the correlation matrix is calculated by averaging over spatial locations. Previous frames are weighted based upon how correlated they are with current frame. This approach has the desirable effect of minimizing the RMS filtered output, which is equivalent to minimizing the mean squared prediction error. However, this technique has the drawback that the correlations calculated are average values and are not necessarily representative of the temporal correlations for any given detector. Also like frame differencing, this processing distorts the target signal in an unpredictable manner.

To evaluate and compare detection algorithm approaches we used a data set from a representative space-based sensor. The data is collected in an infrared band and an intensity gray scale image of the first frame in the data set is shown in Figure 1. The image (11 x 8 detectors) is subset of the full array. There are 400 sequential frames that are included in the data set. For reference purposes the detector at the lower left-hand corner of the image is designated Detector 0,0. Present in all of the frames is a cloud background. The transition between the higher intensity values (lighter detectors) in the lower right of the image and the less intense (darker) detectors in the upper left is a cloud edge. It is the motion of a detector across this cloud edge that causes significant clutter signals over time.

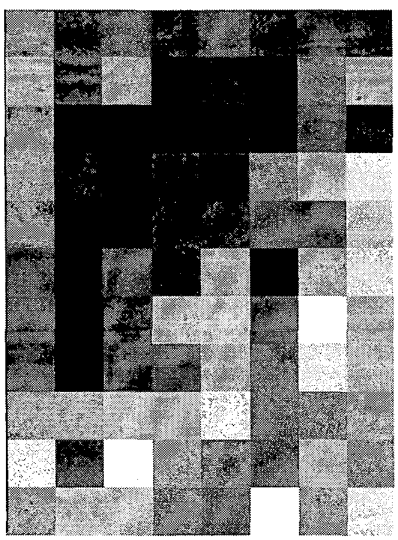


Figure 1 Sensor Cloud Image

The time, intensity profile for Detector 5,4 (detector in the center of the image) is shown in Figure 2. This time history used as a test case for the GRNN and shown here to illustrate some of the problems encountered. The intensity (measured in counts) variation over time results from the detector moving back and fourth across the cloud edge. The frequencies of the control system are observable in the time signal and it is seen that there is a frequency component whose period is about 30 frames. Also seen is a lower frequency component since the signal amplitude in each of the cycles is changing slowly with time. Finally, at the end of the sequence the transition of the control system to a new operating regime is observed through the dramatic and sudden change in the detector intensity.

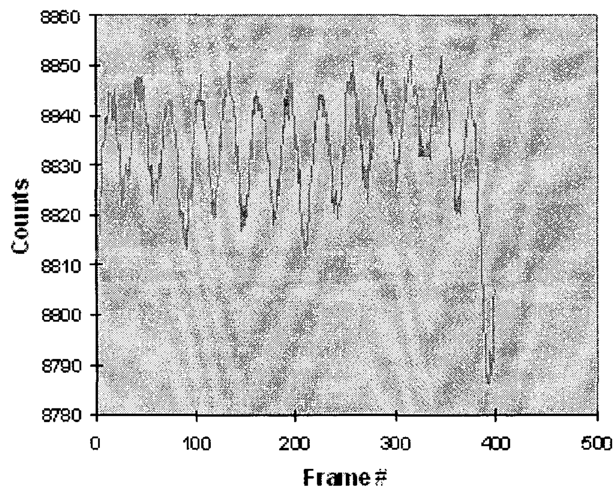


Figure 2 Typical Time Signal (Pixel 5,4)

The next section explains how the GRNN operates and how it is trained over a limited number of frames to predict the detector intensity at latter times. Two feature sets are proposed and described.

3.0 ALGORITHM DESCRIPTION

The background suppression processing performed in this analysis is described here. First, a brief description of the GRNN is given along with some of the underlying theory. Next, a description of both of the feature sets is given as well as some of the training assumptions.

3.1 GRNN Description

The GRNN is an optimal estimator that implements the well-known statistical concept of conditional proability. If a joint probability distribution between a set of independent variables ($\underline{X}$) and a dependent variable (Y) exists, then it is well known that the best estimate of Y given a particular realization of $\underline{X}$ ($\underline{X}_r$) is given by:

$$E\{Y|\underline{X}_r\} = \int_{-\infty}^{\infty} Y\, f(Y,\underline{X}_r)dy \Big/ \int_{-\infty}^{\infty} f(Y,\underline{X}_r)dy \quad (1)$$

The numerator in the equations says the best estimate of Y is the mean of the marginal distribution. Note that the denominator is nothing more than a scaling term that ensures that the marginal distribution integrates to one.

The problem now is how to develop a joint distribution from a set of training patterns. Here a non-parameteric technique developed by Parzen is used. Parzen developed a window function and showed that if you convolved it with the training patterns that this would produce an estimate of the joint distribution. As the number of training points approaches infinity, the technique produces an unbiased and consistent estimate (meaning that the predicted distribution approaches the actual one). Specifically:

$$f_s(\underline{Z}) = \int_{-\infty}^{\infty} \hat{f}(\underline{Z}) W(\underline{Z} - \underline{Z}') d\underline{Z}' \quad (2)$$

and

$$\hat{f}(\underline{Z}) = \frac{1}{N} \sum_{i=1}^{n} \delta(\underline{Z} - \underline{Z}_i^{*})$$

Note that n is the number of training points, δ is the Kronecker delta function, $\underline{Z}^T = \{Y, \underline{X}^T\}$ and $\underline{Z}_i^T$ is the i[th] training pattern. The only constraint on the window function, W, is that it integrates to one. It is easily seen from the above equation that $\hat{f}_s$ is the smooth version of the sampled training set.

If the window function is made a gaussian, it is easy to show from Equations (1) and (2) that the best estimate of Y is given by:

$$\hat{Y} = \sum f_i Y_i^{*} \Big| \sum f_i \quad (3)$$

and

$$f_i = \prod_j e^{-1/2\left(\frac{x - x_{ij}^{*}}{\sigma_j}\right)^2}$$

The standard deviations of the gaussian function are free parameters and can be determined from the training data itself. They control the width of the window function in each dimension and should approach zero as the number of patterns approaches infinity. Their best values for a fixed number of training points are obtained by minimizing the squared error function of all the patterns in the training set.

As seen from Equation (3) the best estimate of Y can be obtained by performing a number of parallel calculations. Therefore, Equation (3) can be efficiently implemented as a neural network with a generic implementation given in Figure 3. From the figure it is seen that there are four layers. The first layer just distributes input vector to each of the pattern nodes. The pattern nodes each produce a weight based upon how close the input vector is to the associated pattern. The third layer consists of two summations. The first summation takes a weighted sum of the pattern dependent variables with the weights being taken from the outputs of the pattern nodes. The other summation simply sums the weights produced by the pattern node. Finally, the output is the division of the two summations.

Equation (3) is intuitive in that is says that the best estimate is just a weighted sum of all the training values. The weight that is placed on each of the values is determined by how close the input vector is to the training pattern. In this respect the GRNN is nothing more then an interplator.

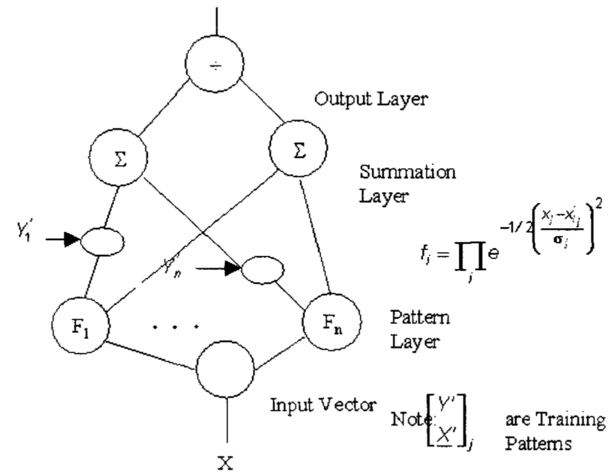


Figure 3 GRNN Architecture

The GRNN implemented here is based on one given by Masters [1]. The only difference is that this implementation uses the -1/2 term in the exponent as listed in Equation (3). It is felt that the inclusion of this term cuts down on the chance of under flowing the exponent, which if it happens often leads to numerical errors. Masters does not use the term. Of course, all of the derivatives used in minimizing the error function are also changed slightly from the ones given in [1] with this implementation.

3.2 Feature Sets

Two feature sets are identified here. The first set is commonly used in a variety of applications and is described briefly. Here the intensities at specific time lags are used to predict future values in the time series of a single detector. If the control system is close to periodic then a few lags are sufficient to predict future values. For this case five time lags are used. They are the intensity of the signal at lags of 24, 28, 29, 30 and 34 frames. These values are chosen because they bracket the dominant frequency of the control system that has a period of about 30 frames. If the current background intensity could be predicted by a fixed number of lags then it could be removed and what signal is left could be thresholded to determine whether it is due to a target or random detector noise. This assumes that the target is not present in the training interval and is also the reason why smaller lags are not used. Smaller lags may be more correlated with the current frame but since targets take a few frames to move through a detector there is a possibility that outside the training region the smaller lags could be corrupted by the target signal.

To train the GRNN network with this set of features, the first 150 test cases are used. The first frame in the data set is designated 0 and since the largest lag is 34 frames, the first test pattern occurs at frame 34 and the last at frame 184. Five cycles of the control system are observed during this training period and it is thought that this should be enough to characterize its higher frequencies. No attempt is made to optimize the training interval or the number of lags or the lag values. With more analysis better results may be obtainable.

The second feature set uses spatial correlations between detectors to predict the intensity in the pixel under test. Since no time lags are used, its performance is dependent on the number of translations of the array that it observes during training and not on the frequencies of the control system. The feature set is illustrated in Figure 4. Here the presence or absence of a target in the middle detector is being determined. As a result of the blur of the optical system and the fact that the target can be at a number of different phasings within the middle detector, the nearest 8 surrounding detectors are ignored since they could be corrupted by target signal. Therefore, the intensities in the outer ring of 16 detectors are used to predict the intensity in the pixel under test. After the intensity is predicted it is subtracted and the residual signal is thresholded to check for the existence of a target. Note that again it is assumed that there are no targets present in the training period, which means that the estimate obtained for the center detector is due to the background only.

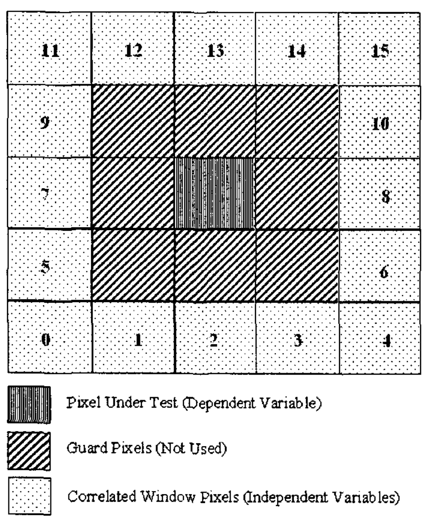


Figure 4 Spatial Window Features

As with the first feature set, the first 150 frames are used as the training patterns. However, in this case the first training frame is at time 0 since there are no time lags involved. Again, no optimization of the number of test patterns or window size has been performed.

4.0 PERFORMANCE

The performance associated with both of the feature sets is given here. For both cases, the results using the Conjugate Gradient training algorithm are used. The training is described in detail in the next section. The Conjugate Gradient algorithm found the lowest minimum of all the training algorithms considered and it did it more efficiently, and therefore, its results are presented. However, it should be noted that all of the training algorithms produced similar results.

Figure 5 shows the results using Detector 5,4 and the temporal lag features. The top most chart gives both the measured and the predicted signal as a function of frame number. The bottom most chart shows the error or residual signal which is the measured signal minus the predicted. It is this residual signal that thresholded in an operational system to determine if a target is in the pixel. The results show excellent agreement between the measured and predicted signals inside the training interval, with good agreement outside of it.

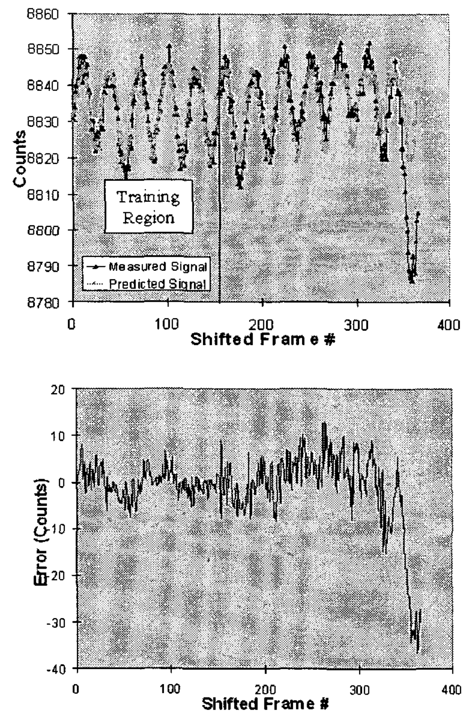
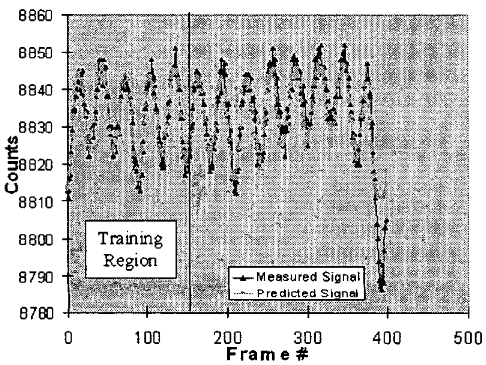


Figure 5 Background Estimation & Removal (Temporal Lag Features)

There are many interesting points to make. The first is that with the exception at the very end, the errors grow only slightly outside the training interval. At the end, the control system is moving into a regime of operation, which causes a large motion, which was not observed during training. This causes the GRNN to try extrapolating which it can not do well, and the result is large errors.

Also seen in the Figure is a low frequency component to the residual signal. If things worked perfectly all of the frequencies of the background should be removed and all that should be left is the random detector noise, which is temporally uncorrelated. The low frequencies of the residual signal indicate that the observation time used in training was not long enough to capture all of the lower frequencies in the patterns. As mentioned in previous sections, we expect this approach to be sensitive to new signal frequencies outside of the training interval so this is not an unexpected result. Using a larger observation interval for training could mitigate both effects.

Similar results using the spatial feature technique are given in Figure 6. The results are excellent both inside and outside the training interval until the last couple of frames where the control system provides a large array displacement. The performance degradation observed at the end is not unexpected since a LOS displacement of that size was not present in the training interval. The residuals obtained throughput the rest of the frames indicate the desirable result that the signal is whitened and that all background frequencies are removed.



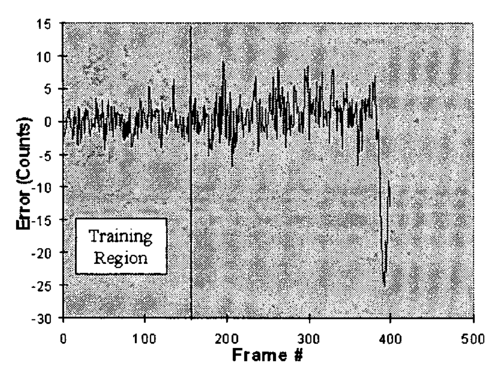


Figure 6 Background Estimation Removal (Spatial Window Features)

To show that the spatial features are robust and work well for other pixel locations in the background, similar results are provided in Figure 7 for five other detector time histories. In all cases, the resulting residual signal is whitened. Finally, the amount of degradation in performance outside of the training interval is characterized in Figure 8, which provides the RMS values of the residuals. In all cases, the performance degrades by less then a factor of two outside of the training interval. The RMS represents the power of the signal that would be thresholded in operational systems. Note that for the RMS calculation the last 50 values are ignored because that is where a large unexpected displacement of the array occurred.

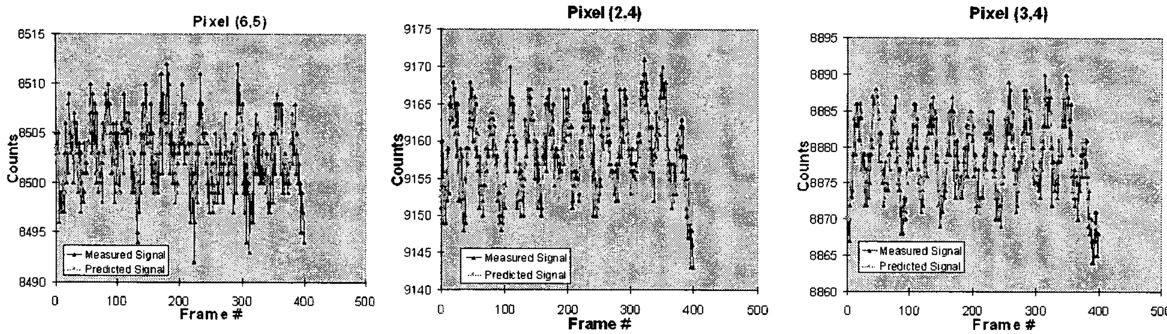
5.0 Training

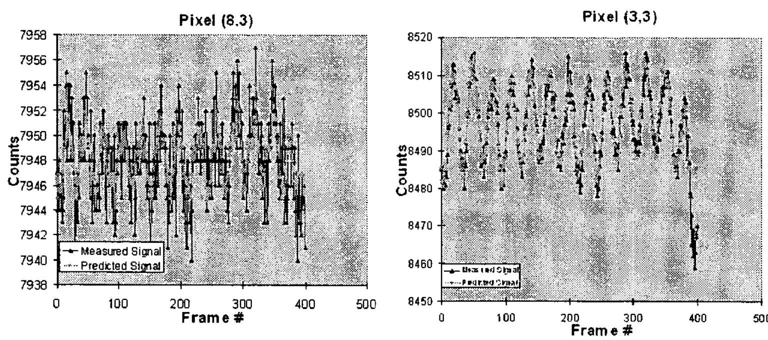
Three training methods are investigated in this analysis. The first involves using Conjugate Gradient combined with Line Minimization to find the minimum of the error function. The implementation of this algorithm is based on the ones given by Masters [1] and Press[2]. For a multidimensional function, Conjugate Gradient determines the best search direction along a line. The direction it chooses is orthogonal to all directions that it has chosen in the past and therefore, the search along the new direction does not interfere with any progress made previously. If the function is quadratic, the Conjugate Gradient method is proven to find the minimum in n line minimizations where n is the dimension of the function.

The line minimization that is implemented involves a global search where a trio of points that bound the minimum are found followed by a refinement stage which uses second order information to zoom in on the absolute minimum. The global method breaks the interval up into a fixed number of equal segments and then evaluates the function at each segment endpoint until the global minimum is found. Refinement is accomplished using Brent's method, which fits parabolas through the trio of points and solves for the minimum. If the function is not well behaved then Brent's method uses Golden Sections to constrict the trio of points until the minimum is identified.

The second training method is Differential Evolution which is a global search algorithm that uses a population of solutions and applies mutation and crossover operations over time to evolve them towards better and fitter solutions. The mutation operation adjusts each of the parameters in the solution to the natural scaling of the problem. The crossover operator takes the best from two solutions and combines them. The algorithm does not use any derivative information and as a result has less of a chance in getting stuck at a local minimum. For the training performed, the crossover coefficient is set at 0.4 (60% of the time the parameter is taken from the primary parent) and the mutation coefficient is set at 0.4.

The last algorithm is hybrid that uses DE to perform the global search to identifying promising areas in solution space and then occasionally uses Line Minimization to search those areas efficiently. After a child is generated in DE, a random yes/no decision is made to perform a single line minimization on its parameters. The line minimization is performed 10% of the time.





Pixel	RMS Error (Training)	RMS Error (Evaluation)
5,4	1.8	3.1
6,5	2.1	2.9
8,3	1.8	2.3
2,4	1.9	2.9
3,3	1.2	2.9
3,4	1.8	2.3

Figure 7 Background Removal (Spatial Window Features)

All three algorithms are compared in Figure 8. Here the squared error for the best solution is shown versus the number of function evaluations. It is seen from the figure that the Conjugate Gradient Algorithm performs the best. It reaches a lower minimum with less function evaluations. This can only occur if the error surface is close to quadratic. Under this assumption using derivative information is very helpful and the global methods, which do not, generally perform many unneeded calculations. The hybrid algorithm has the slowest convergence rate. A possible reason for this is that initially there are low probabilities that line minimization is performed on the best solution. Also, since only a single line minimization is performed, the benefit of searching in conjugate directions is not obtained until the population converges. Finally, it is noted from the figure that all algorithms converged to roughly the same minimum.

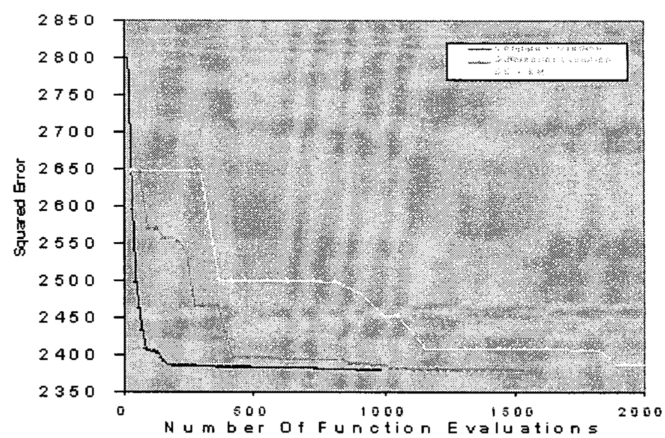


Figure 8 Training Summary (Spatial Window Features)

6.0 CONCLUSIONS

The feasibility of using a Generalized Regression Neural Network (GRNN) for clutter suppression and background estimation was investigated. Two feature sets were considered in processing:

(1) previous time samples from the same pixel that occurred before a specified lag
(2) the surrounding spatial pixels on the same temporal frame

Three training methods were evaluated:

(1) Conjugate Gradient/Line Minimization
(2) Differential Evaluation
(3) Hybrid combination of (1) and (2)

A measured dataset from a spacebased infrared staring sensor was used to evaluate both techniques.

The results showed that both techniques provided excellent ability to follow the time varying signals out of each detector in the focal plane. The RMS errors in the processed signal were on the order of 3 counts when the signals had mean values of 8800 and sinusoidal variations on the order of 20 counts. The 3 count RMS errors where only slightly greater than the RMS training errors which were on the order of 2 counts. All 3 training methods proved the same minimum. They also demonstrated a broad error surface for this dataset.

This analysis showed that the feature sets and GRNN implementation could be used as is, without any modification, to eliminate background signals in a post-detection mode. For this application, the target locations would be known beforehand, and the GRNN could use training data taken from times both before and after the target was in a detector and use it to predict the background signals when the target is actually in the detector. This is an excellent application for the GRNN since it provides optimal use of the data, each new data point being just another node in the network. Note that for this case, filtering does not need to be causal so the number of test patterns can be quite large. Also since this is post-detection processing, it typically does not need to be performed in real time.

The GRNN could be used in real time detection process but there are a number of issues that need to be addressed before it is practical. First, a way needs to be developed to initialize the GRNN so that training is fast. The optimal values of the standard deviations used in the networks must be related to some physical parameters such as the gradients in the background. This exact relationship needs to be worked out so that the standard deviations can be obtained with little or no training. The results indicate that the error function is broad so that if an efficient way of initializing the training procedure can be developed, the training time may be able to be made insignificant.

Also, there are slowly changing features in the background that need to be accommodated in an operational design. This suggests that training needs to be a continuous process and therefore, an efficient way of adding new patterns to the network needs to be developed. As a corollary, tests to decide if a pattern is new also needs to be worked out. Also for computational purposes, old patterns that are not longer relevant probably need to be eliminated over time.

7.0 REFERENCES

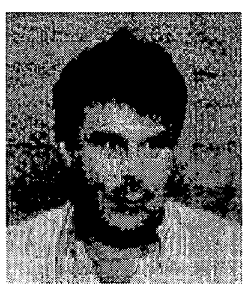
[1] Master, (1995) *Advanced Algorithms for Neural Networks,* John Wiley & Sons, N.Y.

[2] Press, Teukolsky, Vetterling & Flannery, (1992), *Numerical Recipes in Fortran,* Cambridge University Press

8.0 BIOGRAPHY

C. Ralph Waters (Member, IEEE) received BES, MES and PhD in Electrical Engineering from SUNY at Stony Brook in 1967, 1968 and 1975, respectively. From 1969 to 1985 he was with the Research Department of Grumman Aerospace where he worked on developing estimation and processing concepts for advanced space sensor concepts. During this time he was also Adjunct Associate Professor at SUNY at Stony Brook teaching courses in Estimation and Control Theory.

Since 1985 Dr. Waters has been Vice President and Director of Signal/Data Processing for Photon Research Associates. His recent work has concentrated on developing multidimensional estimation and exploitation algorithms for hyperspectral sensor concepts.



Tony Sommese received his BS and MS from SUNY at Stony Brook in 1982 and 1985, respectively. Since 1985 he has worked at Photon Research Associates developing detection, estimation and pattern recognition algorithms. Previously he had worked at Grumman Aerospace designing on-board processing systems for space based surveillance sensors.

Captain Brian A. Hibbeln graduated from the USAF Academy with a bachelor of Physics and was named the outstanding cadet in Physics research in May of 1991. He then was selected to attend the Air Force Institute of Technology at Wright-Patterson Air Force Base where he earned his MS in Space Physics in December of 1992. His thesis work led to a patent for a high-voltage, high-current semiconductor switch for space and pulsed power applications. He then served at the National Air Intelligence center as the Program Manager of an advanced space-based infrared satellite program until September of 1996. He currently supports the National Air Intelligence Center in Washington DC as a developer and manager of advanced space based satellite sensors, ground systems, and advanced processing algorithms.