

Dim Target Detection Using High Order Correlation Method

REN-JEAN LIOU, Student Member, IEEE

MAHMOOD R. AZIMI-SADJADI, Senior Member, IEEE
Colorado State University

This paper presents a new method for clutter rejection and dim target track detection from infrared (IR) satellite data using neural networks. A high order correlation method is developed which recursively computes the spatio-temporal cross-correlations between data of several consecutive scans. The implementation of this scheme using a connectionist network is also presented. Several important properties of the high order correlation method are established which indicate that the resultant filtered images capture all the target information. The simulation results using this approach show at least 93% clutter rejection. Further improvement in the clutter rejection rate is achieved by modifying the high order correlation method to incorporate the target motion dynamics. The implementation of this modified high order correlation using a high order neural network architecture is demonstrated. The simulation results indicate at least 97% clutter rejection rate for this method. A comparison is also made between the methods developed here and the conventional frequency domain three-dimensional (3-D) filtering scheme, and the simulation results are provided.

Manuscript received November 8, 1991; revised May 1, 1992.

IEEE Log No. T-AES/29/3/07973.

This work was supported by IBM Corporation, Federal Sector Division, Boulder, CO 80301.

Authors' address: Dept. of Electrical Engineering, Colorado State University, Fort Collins, CO 80523.

0018-9251/93/\$3.00 © 1993 IEEE

I. INTRODUCTION

This paper is concerned with the problem of detecting dim moving targets from infrared (IR) data and identifying their tracks using artificial neural networks. A number of automatic target detection schemes have been developed over the past years [1-7]. Among these schemes are spatio-temporal filtering [2, 6], maximum likelihood (ML) estimation [3] and recursive Kalman filtering [7]. However, in practice, these methods have been only partially successful and have been shown to produce high false-alarm rate. Some contributing factors for this are : extremely low signal-to-noise/clutter ratio, nonrepeatability of the target signature, competing clutter forming returns with the same shapes as those of the actual targets, obscuration of targets in noise and lack of a priori information about the initial conditions and the signal statistics. In addition, the conventional approaches lose their accuracy when multiple targets are present in the scene, and the targets move arbitrarily in any direction and form a track with any shape.

Neural networks offer potentially powerful, robust and adaptive means of detecting and recognizing targets in high cluttered background. More recently, two different neural network-based approaches have been developed [8-10] to reject the background clutter and identify the target tracks in IR, radar or sonar data. In [9] the optimum post-detection target track receiver was implemented using a Hopfield network. The network is trained to identify straight line tracks in the image without taking into account the temporal behavior of the targets. The method in [10] uses a Hopfield network as a pre-processor and a multilayer Perceptron network as a detector for detecting target signals in underwater acoustic field. However, the application of these schemes are limited to certain known shaped patterns and they lose their validity if any arbitrary shape track is to be detected. In addition, the Hopfield network has two major limitations when used as a content addressable memory. First, the number of patterns that can be stored and accurately recalled is severely limited (15% of the total number of input nodes). Second, an exemplar pattern will be "unstable" if it shares many bits in common with another exemplar pattern. An exemplar is considered to be unstable when it is applied at time zero and the network converges to some other exemplar.

A novel clutter rejection scheme is developed here which circumvents all the shortcomings of the previous schemes. This method which is referred to as "high order correlation" [11, 12] exploits the consecutive temporal and spatial dependencies of the points on a track to discriminate them from background clutter. The implementation of this method using a feedforward connectionist network architecture is also presented. This method does not make any

assumption on the shape and the direction of tracks and it preserves the shape of the track in the resultant filtered images. This algorithm is tested on numerous target events and an average clutter rejection rate of 95% is achieved without losing any valid information pertaining to the target tracks. This method is then modified to improve the clutter rejection rate by considering the high order correlations between three consecutive scans. The improvement in the discrimination capability of this method is achieved by imposing a curvature constraint associated with real target tracks into the process.

An alternative approach for the target detection using 3-D spatio-temporal filtering [6] is also applied for comparison. This approach is based on directional filtering which uses a bank of filters for all the possible target directions and performs the filtering in the frequency domain. Simulation results for various events are obtained and a comparison on effectiveness of all the methods is also given.

The organization of this paper is as follows. In Section II, the problem of multi-scan target detection is described and a model for the process is formulated. In Section III, a basic spatio-temporal cross-correlation method is developed to examine the data between consecutive scans. This cross-correlation scheme is then further extended using high order correlations to improve the clutter rejection rate. The implementation of the high order correlation scheme on a connectionist network is presented and the process of generating the filtered images using this architecture is discussed. Some interesting properties of the high order correlation scheme are also derived. A modification of high order correlation method for achieving better clutter rejection rate is discussed in Section IV. In Section V, target detection using the 3-D filtering scheme is presented. Simulation results on several selected IR images provided by the IBM Corp. are presented in Section VI and a discussion on the effectiveness of the algorithms is also given. Finally, Section VII deals with concluding remarks.

II. MULTISCAN TARGET DETECTION PROBLEM AND MODELING

A. Detection Problem

A multi-scan image is actually obtained by stacking several two-dimensional (2-D) images collected at uniform time intervals by an IR sensor whose field of view is fixed with respect to the background. This yields a 3-D digital image in the discrete Cartesian coordinates (x, y, t_n) where (x, y) are the spatial variables and t_n represents the time or the scan number. A moving target in the sensed scene forms a track or a signature in this 3-D image.

The problem of target detection from a multi-scan data is stated as follows. Given a 3-D image as

described above, detect and identify the target tracks and remove the background clutter as much as possible. The following assumptions are made in this detection process.

- 1) The absolute range of target speeds is 4 pixels per scan.
- 2) The maximum degree of curvature from one scan to the next for a target track is $\pi/3$ rad/s.
- 3) The maximum number of missing target points due to the sensors in 10 scans is two.
- 4) There can be more than one target in the scene.

Since the target tracks extend only over small portions of the entire 3-D image, windowing during the detection process was adopted. Other benefits of windowing include suitability for parallel processing and reduction in the computational complexity. The determination of an appropriate window size is a compromise. It should be as small as possible so that less clutter is enclosed; on the other hand it should be big enough to include all the target points in n_0 scans where n_0 is the scan number at which the detection must be made.

B. Modeling of The Process

A scanning or a mosaic sensor on board a satellite provides a set of returns that contains target signal as well background clutter and additive sensor noise. Owing to the fact that the intensity of target is unknown and can vary substantially from one scan to the other, the detection of dim targets in clutter and noise using intensity-based schemes becomes totally impractical. Thus, methods such as thresholding naturally lead to a low detection rate or high false alarm rate. As a result, we consider the sensor returns as a binary image. If the sensor gives an intensity to a particular location, then the corresponding pixel is set to 1, otherwise is 0. Thus, the image model of one scan can be expressed as

$$F(x, y, t_n) = S(x, y, t_n) + [1 - S(x, y, t_n)]N(x, y, t_n) \quad (1)$$

where x, y are the locations on a discrete 2-D plane, t_n is the time unit or scan number, and F is the image consisting of the target S and clutter N . Note that F, S , and N are all binary arrays. Thus, $F = 1$ corresponds to either target or clutter.

III. DEVELOPMENT OF SPATIO-TEMPORAL HIGH ORDER CORRELATION METHOD

A. Basic Cross-Correlation Method

Let us for now consider only the target alone scenario, i.e., no background clutter. Since the

target moves along a certain direction and builds a time-dependent track, there exists correlations between different scans. Although the velocity of target is unknown, a reasonable assumption for the range of its variations can be made.

Let us assume that the target moves with maximum velocity of v pixels in both x and y directions at one time-unit, i.e., between two consecutive scans t_n and t_{n+1} , then the correlation between the target points can be determined by checking

$$R_{t_n, t_{n+1}}(v) = \sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} S(x_n, y_n, t_n) \times S(x_n + i, y_n + j, t_{n+1}) = 1 \quad (2)$$

where (x_n, y_n) is the target location at time t_n .

However, since the target location (x_n, y_n) is not known, we need to examine all the points within a window. Windows of size 30×30 are used which would be big enough to cover all the track points from scan 1 to scan 10. Then the spatio-temporal correlation can be evaluated using

$$R_{t_n, t_{n+1}}(v) = \sum_{x=x_L}^{x_L+29} \sum_{y=y_L}^{y_L+29} \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} S(x, y, t_n) \times S(x+i, y+j, t_{n+1}) \right] \quad (3)$$

where x_L, y_L correspond to lower left coordinates of the window. If a target is present in a window and within two consecutive scans then $R_{t_n, t_{n+1}}(v) = 1$, otherwise $R_{t_n, t_{n+1}}(v) = 0$.

To consider the effects of clutter in the formulations, (3) can be rewritten for the received image F as

$$R_{t_n, t_{n+1}}(v) = \sum_{x=x_L}^{x_L+29} \sum_{y=y_L}^{y_L+29} \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} F(x, y, t_n) \times F(x+i, y+j, t_{n+1}) \right] \quad (4)$$

Note that $R_{t_n, t_{n+1}}(v)$ in this case may be greater than 1 because clutter may also be correlated. To determine whether a window contains a possible target track $R_{t_n, t_{n+1}}(v)$ should be greater than zero for all $n \in [1, n_0]$, if there is no missing target point. In this case when n varies from 1 to $n_0 - 1$ there is consistent correlations at all times. Obviously, these correlations may be caused by either target points or clutter which makes a time-dependent false track. Since no a priori knowledge about the existence of the target is available, the correlation values may not convey any valid information to distinguish between target and clutter windows. Instead, the decision on the detection and clutter rejection may be made by examining the consistency in the correlations for all

$n \in [1, n_0 - 1]$ rather than using the actual correlation values. This can be achieved by using a hard limiter threshold function $g(\cdot)$ defined by

$$g(x) = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{if } x \leq 0. \end{cases} \quad (5)$$

Now, by applying $g(\cdot)$ to R , a decision rule can be generated to check the consistencies in the correlations and determine whether or not a window contains a possible target track. This is given by

$$U = \sum_{n=1}^{n_0-1} g(R_{t_n, t_{n+1}}) \quad (6)$$

Ideally for a window which covers all the target track points for n_0 scans, U should be equal to $n_0 - 1$. Nevertheless, it is possible that the target points may be missed by the sensor. In this case, we can choose a threshold less than but close to $n_0 - 1$ to take into account for the missing target points. If U is greater than this threshold then the window is a possible candidate, otherwise it is a nontarget window and hence its content can be removed.

By applying this method, the experimental results showed that about 80% of the nontarget windows can be removed. In what follows a new method is developed which yields significantly better clutter rejection rate.

B. High Order Correlation Method

The approach described in the previous section computes the spatio-temporal correlations between windows of two consecutive scans. Consequently, it only provides information on window correlations between two consecutive scans. However, these correlations may be caused by points located anywhere inside the 30×30 window which may not necessarily form a track for more than three consecutive scans.

The problem can be remedied by developing a method referred to as "high order correlations" which computes the correlations recursively. This method provides information on the correlations of several consecutive scans instead of just two scans as in the previous case. Equation (4) is modified to take into account the behavior of each individual point within a window from scan n to $n+1$. This yields the following equation after the thresholding function $g(\cdot)$ is applied.

$$Y(x, y, t_n) = g \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} F(x, y, t_n) F(x+i, y+j, t_{n+1}) \right] \quad (7)$$

where v is the maximum target speed range, and Y , which represents how the points of F at time t_n

are correlated to their neighboring points at t_{n+1} , is evaluated in a window of size 30×30 . Equation (7) performs a point oriented correlation which generates an array of values of Y with size 30×30 . To determine the correlation in more than two consecutive scans we continue this process and compute the correlations of Y . This then yields a recursive procedure with the general form of

$$Y^{(k)}(x, y, t_n) = g \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} Y^{(k-1)}(x, y, t_n) \times Y^{(k-1)}(x+i, y+j, t_{n+1}) \right] \quad (8)$$

where n varies from 1 to n_{\max} , and k is the order of the recursion. When the order k increases n_{\max} should decrease such that $k + n_{\max} = n_0$ for computing the correlations among n_0 scans. Note that at $k = 0$, $Y^{(0)}(x, y, t_n) = F(x, y, t_n)$. It will be shown that if $Y^{(k)}(x_n, y_n, t_n) = 1$, then there is a possible track extending from location (x_n, y_n) at scan t_n to a point at scan t_{n+k} . This track can have arbitrary shape and consistently gives correlations between two adjacent scans.

EXAMPLE 1 In order to illustrate this process let us consider a simple 1-D example. Shown below are 5 scans of data associated with a target and nontarget windows. It is assumed that the maximum target movement is 2 pixels/scan.

Target Window					Nontarget Window						
x	1	2	3	4	5	x	1	2	3	4	5
$Y^{(0)}(x, t_1)$	1	0	0	0	1	$Y^{(0)}(x, t_1)$	0	1	0	0	0
$Y^{(0)}(x, t_2)$	0	1	0	0	0	$Y^{(0)}(x, t_2)$	1	0	0	0	1
$Y^{(0)}(x, t_3)$	0	0	1	0	0	$Y^{(0)}(x, t_3)$	0	0	0	1	0
$Y^{(0)}(x, t_4)$	1	0	0	1	0	$Y^{(0)}(x, t_4)$	1	0	0	0	1
$Y^{(0)}(x, t_5)$	0	0	0	0	1	$Y^{(0)}(x, t_5)$	0	0	1	0	0

Using the 1-D version of (8) and setting $v = 2$, the correlations of the above sequences are given by

$$Y^{(1)}(x, t_n) = g \left[\sum_{i=-2}^2 Y^{(0)}(x, t_n) Y^{(0)}(x+i, t_{n+1}) \right], \quad n \in [1, 4] \quad (9)$$

where $0 < x + i \leq 5$. This yields the correlation sequences given below.

Target Window					Nontarget Window						
x	1	2	3	4	5	x	1	2	3	4	5
$Y^{(1)}(x, t_1)$	1	0	0	0	0	$Y^{(1)}(x, t_1)$	0	1	0	0	0
$Y^{(1)}(x, t_2)$	0	1	0	0	0	$Y^{(1)}(x, t_2)$	0	0	0	0	1
$Y^{(1)}(x, t_3)$	0	0	1	0	0	$Y^{(1)}(x, t_3)$	0	0	0	1	0
$Y^{(1)}(x, t_4)$	0	0	0	1	0	$Y^{(1)}(x, t_4)$	1	0	0	0	1

It is clearly observed that $Y^{(1)}(x, t_n)$ represents how $Y^{(0)}(x, t_n)$ are correlated to their neighboring points in $Y^{(0)}(x, t_{n+1})$. Interestingly enough if the method in Section III.A was used, the target and nontarget windows would generate the same $R_{Y_n} = \sum_x Y^{(1)}(x, t_n)$ values which indicate consistent correlations in every two consecutive scans. As a result, this method would not be capable of differentiating between the two cases since all R_{Y_n} are equal to 1. However, using the high order correlation method a sequence of new correlations between $Y^{(1)}(x, t_n)$ and $Y^{(1)}(x, t_{n+1})$ can be generated which provides better discrimination capability. This sequence is generated using

$$Y^{(2)}(x, t_n) = g \left[\sum_{i=-2}^2 Y^{(1)}(x, t_n) Y^{(1)}(x+i, t_{n+1}) \right], \quad n \in [1, 3] \quad (10)$$

where $0 < x + i \leq 5$. This operation yields the sequences shown below.

Target Window					Nontarget Window						
x	1	2	3	4	5	x	1	2	3	4	5
$Y^{(2)}(x, t_1)$	1	0	0	0	0	$Y^{(2)}(x, t_1)$	0	0	0	0	0
$Y^{(2)}(x, t_2)$	0	1	0	0	0	$Y^{(2)}(x, t_2)$	0	0	0	0	1
$Y^{(2)}(x, t_3)$	0	0	1	0	0	$Y^{(2)}(x, t_3)$	0	0	0	1	0

Note that $Y^{(2)}$ can be related to $Y^{(0)}$ by

$$Y^{(2)}(x, t_n) = g \left[Y^{(1)}(x, t_n) \sum_j Y^{(1)}(x+j, t_{n+1}) \right], \quad n \in [1, 3]$$

or

$$Y^{(2)}(x, t_n) = g \left[Y^{(0)}(x, t_n) \sum_i Y^{(0)}(x+i, t_{n+1}) \right] g \left[\sum_j Y^{(0)}(x+j, t_{n+1}) \left[\sum_i Y^{(0)}(x+i+j, t_{n+2}) \right] \right] \quad (11)$$

which clearly indicates the correlations between three consecutive scans t_n , t_{n+1} , and t_{n+2} .

The first term on the right side of (11) represents the correlations of $Y^{(0)}(x, t_n)$ in scan t_n with its neighboring points in $Y^{(0)}(x, t_{n+1})$ in scan t_{n+1} . The second term represents the correlations of the neighboring points of $Y^{(0)}(x, t_n)$ in scan t_{n+1} to all their neighboring points $Y^{(0)}(x, t_{n+2})$ in scan t_{n+2} . As a result, $Y^{(2)}(x, t_n)$ carries the correlational information between $\{Y^{(0)}(x, t_n)$ and $Y^{(0)}(x, t_{n+1})\}$, $\{Y^{(0)}(x, t_{n+1})$ and $Y^{(0)}(x, t_{n+2})\}$, and also $\{Y^{(0)}(x, t_n)$, $Y^{(0)}(x, t_{n+1})$ and $Y^{(0)}(x, t_{n+2})\}$. As far as the nontarget window is concerned, $Y^{(2)}(x, t_1) = 0$, $\forall x$ implies that $Y^{(0)}(x, t_n)$

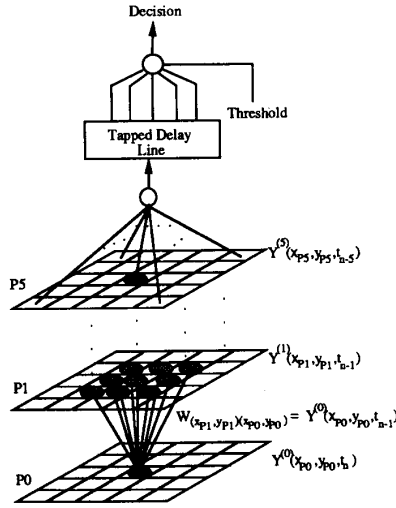


Fig. 1. Connectionist network for high order correlation.

sequence for $n = 1, 2$, and 3 does not form a possible track for all x ; whereas $Y^{(2)}(5, t_2) = 1$ may indicate that $\{Y^{(0)}(5, t_2) Y^{(0)}(4, t_3) Y^{(0)}(5, t_4)\}$ is a part of a possible track in three consecutive scans. Also, $Y^{(2)}(4, t_3) = 1$ may imply that $\{Y^{(0)}(4, t_3) Y^{(0)}(5, t_4) Y^{(0)}(3, t_5)\}$ is a part of a possible track in three consecutive scans. For the target window, however, $Y^{(2)}(1, t_1) = Y^{(2)}(2, t_2) = Y^{(2)}(3, t_3) = 1$ which shows consistent correlations in every three consecutive scans. Consequently, $Y^{(2)}(x, t_n)$ sequence can provide information on the existence of any possible track in three consecutive scans $n, n + 1$ and $n + 2$. To qualify as a complete track $R_{Y_n} = \sum_x Y^{(2)}(x, t_n)$, for every $n = 1, 2, 3$, should be greater or equal to unity. This condition is obviously satisfied for the target window but not for the nontarget window. The target window distinguishes itself from the nontarget window because of its spatio-temporal consistency inherent in high order correlations. The correlations of order higher than 2 may be used to determine consistency in correlations in more than three consecutive scans.

C. Connectionist High Order Correlation Network

This section presents the neural network implementation of the high order correlation method developed in Section III.B. It is noted that (8) is indeed analogous in the form to the equation used for computing the outputs of a feedforward neural network. In (8), the term $Y^{(k)}(x, y, t_n)$ can be regarded as the neuron output, $Y^{(k-1)}(x + i, y + j, t_{n+1})$'s as the inputs, $Y^{(k-1)}(x, y, t_n)$'s as the connection weights and $g(\cdot)$ as the neuron activation function. Based upon this analogy one can design a connectionist network structure for implementing the high order correlation scheme. Fig. 1 shows a sequential feedforward network which has 5 planar layers (excluding the inputs) to

compute the correlations up to the 5th order. All the planar layers have the same number of neurons and all these neurons perform hard limiter threshold function. The outputs of the n th planar layer represent the results of the n th-order correlation. The process is a pipeline operation in which the data is sequentially evolved through the network while the processors are operating simultaneously.

The data within a specific window at each scan is presented to the network inputs sequentially. At scan 1, the original data $Y^{(0)}(x_{p0}, y_{p0}, t_1)$ is directly used to set the connection weights between the planes $P0$ and $P1$, i.e., $w_{(x_{p1}, y_{p1})(x_{p0}, y_{p0})} = Y^{(0)}(x_{p0}, y_{p0}, t_1)$, where (x_{p0}, y_{p0}) and (x_{p1}, y_{p1}) are the row and column numbers at planes $P0$ and $P1$, respectively. At scan 2, the network computes the output of plane $P1$, i.e., $Y^{(1)}(x_{p1}, y_{p1}, t_1)$, using

$$\begin{aligned}
 Y^{(1)}(x_{p1}, y_{p1}, t_1) &= g \left[\sum_{i=-v}^v \sum_{j=-v}^v w_{(x_{p1}, y_{p1})(x_{p0}, y_{p0})} \right. \\
 &\quad \left. \times Y^{(0)}(x_{p1} + i, y_{p1} + j, t_2) \right] \\
 &= g \left[\sum_{i=-v}^v \sum_{j=-v}^v Y^{(0)}(x_{p1}, y_{p1}, t_1) \right. \\
 &\quad \left. \times Y^{(0)}(x_{p1} + i, y_{p1} + j, t_2) \right] \\
 &\quad \forall x_{p0}, y_{p0}, x_{p1}, y_{p1} \in [1, 30]. \quad (12)
 \end{aligned}$$

It should be noted that since we are only concerned with the correlations within the assumed maximum target movements, full connections between all the nodes of two layers are not needed. The number of weights connected to an individual output are fixed.

After the outputs of plane $P1$ are generated, $w_{(x_{p1}, y_{p1})(x_{p0}, y_{p0})}$ is reset so that $w_{(x_{p1}, y_{p1})(x_{p0}, y_{p0})} = Y^{(0)}(x_{p1}, y_{p1}, t_2)$ and the weights connecting planes $P2$ and $P1$, $w_{(x_{p2}, y_{p2})(x_{p1}, y_{p1})}$, are set to $w_{(x_{p2}, y_{p2})(x_{p1}, y_{p1})} = Y^{(1)}(x_{p1}, y_{p1}, t_2)$. At scan 3, $Y^{(1)}(x_{p1}, y_{p1}, t_2)$ are obtained at the output of plane $P1$ as shown in (12) with one time-unit increment, and at the same time the outputs of plane $P2$ can be computed using

$$\begin{aligned}
 Y^{(2)}(x_{p2}, y_{p2}, t_1) &= g \left[\sum_{i=-v}^v \sum_{j=-v}^v w_{(x_{p2}, y_{p2})(x_{p1}, y_{p1})} \right. \\
 &\quad \left. \times Y^{(1)}(x_{p2} + i, y_{p2} + j, t_2) \right] \\
 &= g \left[\sum_{i=-v}^v \sum_{j=-v}^v Y^{(1)}(x_{p2}, y_{p2}, t_1) \right. \\
 &\quad \left. \times Y^{(1)}(x_{p2} + i, y_{p2} + j, t_2) \right] \\
 &\quad \forall x_{p2}, y_{p2} \in [1, 30] \quad (13)
 \end{aligned}$$

where (x_{P2}, y_{P2}) are the row and column numbers at plane $P2$. This process is repeated for all the planes and all the future scans until at scan 6 the high order correlation (5th order) results become available at the output of plane $P5$. The results of high order correlation which are obtained sequentially are then stored for the next $n_0 - 6$ scans in the tapped delay line (TDL) in order to check the consistencies in the correlations. The decision is made after the TDL is full, i.e., after all n_0 scans are processed. The processing element for this decision layer also performs the hard limiter function. To determine whether or not a window contains a possible target track, the results (1 or 0) from the TDL are added for the entire time sequence and then compared with a chosen threshold. The threshold can be adjusted to accommodate the inconsistent correlations caused by missing target points. This should be a negative value so that when the sum of the TDL outputs is greater than the absolute value of the threshold then the output of the decision node is one indicating that this is a possible target window, otherwise it is a nontarget window and its content (clutter) can be removed.

The decision output in the architecture of Fig. 1 is primarily used to reject the noise and clutter as much as possible. However, any false track forming from clutter and/or noise which satisfies the high order correlation criterion is retained as a possible track.

In the previous discussion "clutter rejection" meant discarding all the windows which do not contain a possible track, while clutter inside a possible target window was remained unchanged. The results of high order correlations can be used as a mean to reduce clutter inside a possible target window.

From (8) it is clear that the value of the high order correlation is zero if the original data points do not lie on a possible track. The result of high order correlation in (8) which is still an array of size 30×30 can, therefore, be considered as a filtered image. This image which is referred to as "high order correlation image" provides more refined results as only the points that form possible tracks will remain in the image. Those data points $Y^{(0)}(x, y, t_n)$ for which the high order correlations $Y^{(k)}(x, y, t_n)$ are zero will, therefore, be removed. This image is directly available at the planar outputs of the high order correlation network in Fig. 1, i.e., the n th planar outputs generate the n th order correlation image.

In general, if one has a fixed number of scans, say n_0 , for detection, for the case of k th order correlation one would lose k filtered images in this process. For instance, suppose we have 10 scans of data from t_1 to t_{10} , when $k = 4$ the high order correlation results will

only contain six images, i.e.,

$$Y^{(4)}(x, y, t_n) = g \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} Y^{(3)}(x, y, t_n) \times Y^{(3)}(x + i, y + j, t_{n+1}) \right]$$

where $n \in [1, 6]$. Clearly if the data after scan n_0 are available one can continue generating the high order correlation images after $n_0 - k + 1$ scan. However, if this is not possible then one can continue calculating the high order correlations based upon the old data in reverse order until n_0 high order correlation images are generated to complete the filtering process. This can be accomplished using the backward process,

$$Y^{(k)}(x, y, t_n) = g \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} Y^{(k-1)}(x, y, t_n) \times Y^{(k-1)}(x + i, y + j, t_{n-1}) \right] \quad (14)$$

in which the subscripts of t has been changed.

The choice of the order k in this method presents a tradeoff between the clutter rejection capability and the sensitivity to missing target points. More specifically, if large order k is used more clutter can be removed in a target window, but the process becomes less robust to missing target points. For example, for a target point at (x, y, t_n) , if its next point at scan t_{n+1} was misdetected by the sensor, then (x, y, t_n) may be considered as clutter and removed according to the correlation criterion. If low order correlations are used, this may not create a problem at the decision making stage provided that an appropriate threshold is chosen. However, for large order case, the correlations at the outputs of TDL are no longer consistent. This may lead to misdetection if the threshold is not adjusted accordingly. Even if the target is detected the high order correlation images may contain less valid target information for further processing. In this case, the likelihood of misdetection depends on the robustness of the particular postprocessing scheme. The problem of missing target points is demonstrated in the following example.

EXAMPLE 2 Shown below is a target sequence $Y^{(0)}(n, t_n)$, where $n \in [1, 5]$, with a missing point at $Y^{(0)}(2, t_2)$ together with the results of the 1st-order and 2nd-order correlations. As can be seen, $Y^{(1)}(x, t_n)$ sequence contains more target information than $Y^{(2)}(x, t_n)$ sequence, even though the clutter rejection capability of the 2nd order process is definitely superior. In both cases, the original target information in scan 1, i.e., $Y^{(0)}(1, t_1)$, was not retained. The performance of the 1st and 2nd order correlation processes in retaining most of the original target

information depends of course on where the missing points are occurred. It should be noted that if the sequences $Y^{(1)}(x, t_5)$, $Y^{(2)}(x, t_4)$ and $Y^{(2)}(x, t_5)$ are generated using (14), both $Y^{(1)}(x, t_n)$ and $Y^{(2)}(x, t_n)$ would possess same target information.

Original Image	1st Order	2nd Order
x 1 2 3 4 5	x 1 2 3 4 5	x 1 2 3 4 5
$Y^{(0)}(x, t_1)$ 1 0 0 0 1	$Y^{(1)}(x, t_1)$ 0 0 0 0 0	$Y^{(2)}(x, t_1)$ 0 0 0 0 0
$Y^{(0)}(x, t_2)$ 0 0 0 0 0	$Y^{(1)}(x, t_2)$ 0 0 0 0 0	$Y^{(2)}(x, t_2)$ 0 0 0 0 0
$Y^{(0)}(x, t_3)$ 1 0 1 0 0	$Y^{(1)}(x, t_3)$ 1 0 1 0 0	$Y^{(2)}(x, t_3)$ 0 0 1 0 0
$Y^{(0)}(x, t_4)$ 1 0 0 1 0	$Y^{(1)}(x, t_4)$ 0 0 0 1 0	
$Y^{(0)}(x, t_5)$ 0 0 0 0 1		

D. Properties of High Order Correlation Method

In this section some interesting properties of the high order correlation method are established.

PROPOSITION 1 *Assuming that at scan 1, a target return is occurred at location (x_1, y_1) , i.e., $F(x_1, y_1, t_1) = 1$, then all the future target positions $F(x_n, y_n, t_n)$ can be expressed in terms of variables (x_1, y_1) with various shifts in the spatial coordinates.*

PROOF. Assuming the maximum velocity ν for the target, we can write

$$\begin{aligned}
 & F(x_2, y_2, t_2) \\
 &= g \left\{ \sum_{i_1=-\nu}^{\nu} \sum_{j_1=-\nu}^{\nu} F(x_1 + i_1, y_1 + j_1, t_1) \right\} \\
 & F(x_3, y_3, t_3) \\
 &= g \left\{ \sum_{i_2=-\nu}^{\nu} \sum_{j_2=-\nu}^{\nu} F(x_2 + i_2, y_2 + j_2, t_2) \right\} \\
 &= g \left\{ \sum_{i_2=-\nu}^{\nu} \sum_{j_2=-\nu}^{\nu} \sum_{i_1=-\nu}^{\nu} \sum_{j_1=-\nu}^{\nu} F(x_1 + i_1 + i_2, y_1 + j_1 + j_2, t_1) \right\} \\
 & \dots \quad \dots \quad \dots \\
 & \dots \quad \dots \quad \dots \\
 & \dots \quad \dots \quad \dots
 \end{aligned}$$

$$\begin{aligned}
 & F(x_n, y_n, t_n) \\
 &= g \left\{ \sum_{i_{n-1}=-\nu}^{\nu} \sum_{j_{n-1}=-\nu}^{\nu} F(x_{n-1} + i_{n-1}, y_{n-1} + j_{n-1}, t_{n-1}) \right\} \\
 &= g \left\{ \sum_{i_{n-1}=-\nu}^{\nu} \sum_{j_{n-1}=-\nu}^{\nu} \dots \sum_{i_1=-\nu}^{\nu} \sum_{j_1=-\nu}^{\nu} \right. \\
 & \quad \left. F(x_1 + i_1 + \dots + i_{n-1}, y_1 + \dots + j_{n-1}, t_1) \right\} \quad \square
 \end{aligned}$$

In the above expression since the movement is unknown, (i, j) s are varied within the velocity window to find the location of the target in the next scan. Since there may exist more than one point within the velocity window, the limiting function $g(\cdot)$ is used. With the above statement in mind, we can arrive at the following property.

PROPOSITION 2 *In relation to Proposition 1, it is also possible to express all the k th order correlations $Y^{(k)}(x_n, y_n, t_n)$ in terms of x_1, y_1 variables with various shifts in the spatial coordinates, i.e.,*

$$\begin{aligned}
 & Y^{(k)}(x_n, y_n, t_n) \\
 &= g \left\{ \sum_{i_{n-1}=-\nu}^{\nu} \sum_{j_{n-1}=-\nu}^{\nu} Y^{(k)}(x_{n-1} + i_{n-1}, y_{n-1} + j_{n-1}, t_{n-1}) \right\}. \quad (15)
 \end{aligned}$$

PROOF. The method of induction is used to prove this property. That is, we assume that (15) is valid for order $k-1$, then we prove that it will also be valid for order k . Now since it was valid for $k=0$ (see Proposition 1) it will be valid for $\forall k$. Using the definition of high order correlations and substituting for the relevant terms we have

$$\begin{aligned}
 & Y^{(k)}(x_n, y_n, t_n) \\
 &= g \left\{ Y^{(k-1)}(x_n, y_n, t_n) \sum_{i_n=-\nu}^{\nu} \sum_{j_n=-\nu}^{\nu} Y^{(k-1)}(x_n + i_n, y_n + j_n, t_{n+1}) \right\} \\
 &= g \left\{ \sum_{i_{n-1}=-\nu}^{\nu} \sum_{j_{n-1}=-\nu}^{\nu} Y^{(k-1)}(x_{n-1} + i_{n-1}, y_{n-1} + j_{n-1}, t_{n-1}) \right. \\
 & \quad \times \sum_{i_n=-\nu}^{\nu} \sum_{j_n=-\nu}^{\nu} \\
 & \quad \left. \times Y^{(k-1)}(x_{n-1} + i_{n-1} + i_n, y_{n-1} + j_{n-1} + j_n, t_{n+1}) \right\} \\
 &= g \left\{ \sum_{i_{n-1}=-\nu}^{\nu} \sum_{j_{n-1}=-\nu}^{\nu} Y^{(k)}(x_{n-1} + i_{n-1}, y_{n-1} + j_{n-1}, t_{n-1}) \right\} \quad \square
 \end{aligned}$$

This equation is a general form of the equation in Proposition 1. Now, we can use the result of this proposition to arrive at yet another important feature of the high order correlation scheme.

THEOREM 1 *After computing the high order correlations recursively, if the $n-1$ th-order correlation $Y^{(n-1)}(x_1, y_1, t_1)$ is equal to one, then*

$$\begin{aligned}
 & Y^{(n-1)}(x_1, y_1, t_1) = F(x_1, y_1, t_1) F(x_2, y_2, t_2) \\
 & \quad \times \dots \times F(x_n, y_n, t_n) = 1 \quad (16)
 \end{aligned}$$

i.e., there exists a track (or part of a track) which consists of n points extending from scan t_1 to scan t_n . These points are denoted by $F(x_1, y_1, t_1), F(x_2, y_2, t_2), \dots, F(x_n, y_n, t_n)$ with values 1 at the specific locations and scan numbers.

PROOF. For the first-order correlations, using the results in Propositions 1 and 2, we have

$$\begin{aligned} Y^{(1)}(x_1, y_1, t_1) &= Y^{(0)}(x_1, y_1, t_1)g \left\{ \sum_{i_1=-v}^v \sum_{j_1=-v}^v Y^{(0)}(x_1 + i_1, y_1 + j_1, t_2) \right\} \\ &= F(x_1, y_1, t_1)F(x_2, y_2, t_2). \end{aligned}$$

Therefore, when $Y^{(1)}(x_1, y_1, t_1) = 1$, the product of the two points, $F(x_1, y_1, t_1)$ and $F(x_2, y_2, t_2)$ in two consecutive scans t_1 and t_2 , is one which implies a two-point track. The distance between these two points in each direction is less than or equal to v pixels. If any of the two points is zero, then $Y^{(1)}(x_1, y_1, t_1) = 0$, which means that the two-point sequence does not exist. Intuitively, since the second-order correlations can be viewed as the product of two points of the first-order correlations, it contains information about three consecutive points in a track, i.e.,

$$\begin{aligned} Y^{(2)}(x_1, y_1, t_1) &= Y^{(1)}(x_1, y_1, t_1)g \left\{ \sum_{i_2=-v}^v \sum_{j_2=-v}^v Y^{(1)}(x_1 + i_2, y_1 + j_2, t_2) \right\} \\ &= Y^{(1)}(x_1, y_1, t_1)Y^{(1)}(x_2, y_2, t_2) \\ &= F(x_1, y_1, t_1)F(x_2, y_2, t_2)F(x_2, y_2, t_2)F(x_3, y_3, t_3) \\ &= F(x_1, y_1, t_1)F(x_2, y_2, t_2)F(x_3, y_3, t_3). \end{aligned}$$

By performing the computation recursively, similar equation can be obtained for the general form of high order correlations, i.e.,

$$\begin{aligned} Y^{(n-1)}(x_1, y_1, t_1) &= F(x_1, y_1, t_1)F(x_2, y_2, t_2) \cdots F(x_n, y_n, t_n). \end{aligned}$$

For an existing track, the target points from $F(x_1, y_1, t_1)$ to $F(x_n, y_n, t_n)$ should exist consistently. If $Y^{(n-1)}(x_1, y_1, t_1) = 0$ then it means that at least one point is missing and the points do not form a complete track. \square

REMARK 1 An interesting by-product of this Theorem is the fact that the high order correlation filtered image preserves the shape of the track.

IV. MODIFIED HIGH ORDER CORRELATION METHOD

The high order correlation method developed in Section III does not make any assumption pertaining to the target motion information, which includes direction of motion, velocity variations, and the target initial location. Although this is advantageous from one hand, on the other hand, no constraint such as the target curvature, velocity variations, etc., can be imposed. As a result, perfect clutter rejection rate may not be achieved as clutter with returns similar in shape to a jagged track would satisfy the high order spatio-temporal correlation criterion as well.

A. Development of Modified High Order Correlation Method

One way to remedy this problem is to apply directional constraints on the target movements. Since no a priori information on the target moving direction is available, it is not possible to accomplish this when correlations are computed between two consecutive scans. However, if we consider correlations between three consecutive scans, then each possible movement from the first to the second scan within the velocity window, can be used to determine the range of possible movements from the second to the third scan for a predetermined curvature constraint. This can be portrayed by using the following equation

$$\begin{aligned} Y^{(k)}(x, y, t_n) &= g \left[Y^{(k-1)}(x, y, t_n) \sum_{i_1} \sum_{j_1} Y^{(k-1)}(x + i_1, y + j_1, t_{n+1}) \right. \\ &\quad \left. \times \sum_{i_2} \sum_{j_2} Y^{(k-1)}(x + i_1 + i_2, y + j_1 + j_2, t_{n+2}) \right]. \end{aligned} \quad (17)$$

In (17), to compute the correlations between scans t_n and t_{n+1} , we consider all the possible movements and use the entire velocity window, i.e., $i_1, j_1 \in [-|v|, |v|]$. Given a curvature constraint, for each pair of i_1 and j_1 there is a set of possible i_2 and j_2 values that can be used to determine the size and the shape of the velocity window between t_{n+1} and t_{n+2} . For a given maximum moving curvature constraint this set of i_2 and j_2 values can be obtained from a look-up table.

The target bearing angle from scan t_n to scan t_{n+1} can be calculated using

$$\theta = \tan^{-1} \frac{j_1}{i_1}. \quad (18)$$

Now, if the maximum moving curvature is δ then the turning angle from scan t_{n+1} to scan t_{n+2} should be within a bounded region, i.e.,

$$\theta - \delta \leq \tan^{-1} \frac{j_2}{i_2} \leq \theta + \delta. \quad (19)$$

TABLE I
Curvatures and Movements

2.36	2.22	2.04	1.82	1.57	1.33	1.11	0.93	0.79
2.60	2.36	2.16	1.89	1.57	1.25	0.98	0.79	0.64
2.68	2.55	2.36	2.04	1.57	1.11	0.79	0.59	0.46
2.90	2.82	2.68	2.36	1.57	0.79	0.46	0.32	0.25
3.14	3.14	3.14	3.14	X	0.	0.	0.	0.
3.39	3.46	3.61	3.93	4.71	-0.79	-0.46	-0.32	-0.25
3.61	3.73	3.93	4.25	4.71	-1.11	-0.79	-0.59	-0.46
3.79	3.93	4.12	4.39	4.71	-1.25	-0.98	-0.79	-0.64
3.93	4.02	4.25	4.47	4.71	-1.33	-1.11	-0.93	-0.79

The range of possible values for i_2 and j_2 can be determined using (19). It should be noted that this variable size and shape window is not bigger than the velocity window for i_1 and j_1 . The size and shape of the window is dependent on the parameters θ and δ . Additionally, since maximum target speed is imposed, only limited pairs of i_2 and j_2 need to be considered. Table I shows all the possible cases of moving angles which can be used to obtain the range of (i_2, j_2) once θ and δ are determined. The "X" in the center is the location where the process is originated. The location for each box represents the movement from one scan to the next. The numbers in the boxes are the moving angles corresponding to the next possible movement. The unit is in rad/s.

We now elaborate on the implementation of (17) using a neural network structure. For simplicity, the 1-D case is considered first to demonstrate the concept.

B. Connectionist Modified High Order Correlation Network

The feed-forward network used in the original high order correlation method uses first-order units that are expressed as

$$z_i = g \left[\sum_{j=1}^N w_{ij} x_j \right] \quad (20)$$

where $\{w_{ij}\}$ represents a set of weights connecting inputs $\{x_j\}$ to the unit i and g is the neuron activation function. The output of this unit is denoted by z_i . In [13] it was claimed that great enhancement in learning, generalization, and knowledge representation can be achieved by crafting the network to reflect the high order correlational properties of the input environment in which it is designed to operate. The output of these high order units can be represented [13] by

$$z_i = g \left[w_{o,i} + \sum_j w_{1,ij} x_j + \sum_j \sum_m w_{2,ijm} x_j x_m + \dots \right] \\ = g [T_0(i) + T_1(i) + T_2(i) + \dots] \quad (21)$$

In contrast to the simple model in (20) which performs the weighted sum operation, (21) is a model

of multiplicative connections in which two or more inputs are weighted and multiplied together before entering into the sum. Such a multiplicative connection allows one input to gate the other. If one input of a multiplicative set is equal to zero, the other members of the set will have no effect. On the other hand, if one member of the set is 1, the others are passed to the receiving unit without change. These units are called "sigma-pi units" [14].

The 1-D version of (17) can be rewritten as

$$Y^{(k)}(x, t_n) = g \left[\sum_{i_1} \sum_{i_2} Y^{(k-1)}(x, t_n) \times Y^{(k-1)}(x + i_1, t_{n+1}) Y^{(k-1)}(x + i_1 + i_2, t_{n+2}) \right] \quad (22)$$

Comparison of $T_2(i)$ in (21) with the terms in (22) gives the following relations.

$$g[T_2(i)] = Y^{(k)}(x, t_n), \\ w_{2,ijk} = Y^{(k-1)}(x, t_n), \\ x_j = Y^{(k-1)}(x + i_1, t_{n+1}) \\ x_m = Y^{(k-1)}(x + i_1 + i_2, t_{n+2}), \\ \text{with } i_1 = j, \quad i_2 = m.$$

From this analogy it is realized that high order neural network architectures [13, 14] can be used to implement the modified high order correlation algorithm. The data at scan t_{n+2} are used to gate the data at scan t_{n+1} .

The neural network implementation of the modified high order correlation method can be obtained by rewriting (17) as

$$Y^{(k)}(x, y, t_n) = g \left[\sum_{i_1} \sum_{j_1} Y^{(k-1)}(x + i_1, y + j_1, t_{n+1}) \times \sum_{i_2} \sum_{j_2} Y^{(k-1)}(x, y, t_n) \times Y^{(k-1)}(x + i_1 + i_2, y + j_1 + j_2, t_{n+2}) \right] \\ = g \left[\sum_{i_1} \sum_{j_1} Y^{(k-1)}(x + i_1, y + j_1, t_{n+1}) U(i_1, j_1) \right] \quad (23)$$

where

$$U(i_1, j_1) = \sum_{i_2} \sum_{j_2} Y^{(k-1)}(x, y, t_n) \times Y^{(k-1)}(x + i_1 + i_2, y + j_1 + j_2, t_{n+2}). \quad (24)$$

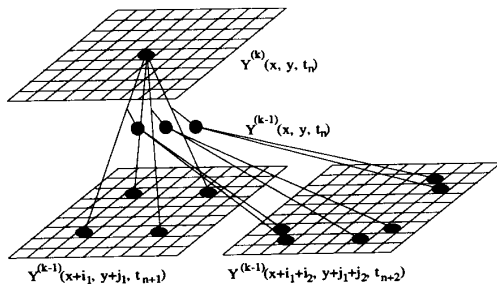


Fig. 2. Connectionist network for one layer modified high order correlation.

Equations (23) and (24) are analogous to those of a feed-forward neural network structure. In equation (23), $U(i_1, j_1)$ can be considered as the connection weights of a neural network with inputs $Y^{(k-1)}(x + i_1, y + j_1, t_{n+1})$ and output $Y^{(k)}(x, y, t_n)$ which is the result of the modified high order correlation. Moreover, $U(i_1, j_1)$ which is obtained from (24) can be generated using a similar network structure.

Fig. 2 shows the architecture of one layer of the modified high order correlation network which uses high order units. Depending on the order of correlations several of these first-order layers should be cascaded. Note that since this method computes the correlations in three consecutive scans, for detection from ten scans the maximum allowable order is 4 as opposed to 9 in the original high order correlation case. Consequently, the computational complexity of the modified high order correlation is comparable to its original version. The network is similar in structure to that of the original high order correlation network with the only difference that the output of the network on the right hand side of Fig. 2 is modulated with the connection weights of the network on the left side, hence generating the high order terms needed for computing the modified high order correlations.

V. THREE-DIMENSIONAL SPATIO-TEMPORAL FILTERING

This section describes a conventional approach [6] using 3-D spatio-temporal filtering in the frequency domain. To detect the presence of a target track, 3-D Fourier transform was first performed on the multiframe images. The transformed data was then passed through a set of directional filters to extract the possible track features. A directional filter is a filter designed specifically to give maximum enhancement to images having certain shape features, especially straight lines in a certain direction. A directional filter is not of low pass, high pass, or bandpass type filter. Its function is to pass all the significant components in the frequency domain that correspond to a straight line in the spatio-temporal coordinates. Since there are

unlimited possible cases for an arbitrary curved track, studies have been restricted to straight lines.

To demonstrate the process of designing the directional filters let us, for now, consider only the true target points. Assuming that the velocities along x and y directions are v_x and v_y , respectively, then at time t we have

$$S(x, y, t) = S(x_1 + v_x t, y_1 + v_y t, t) \quad (25)$$

where (x_1, y_1) is the location where the target originated. Note that the changes in v_x and v_y give line tracks in different directions. $S(x, y, t)$ can also be expressed using the Dirac delta function, i.e.,

$$S(x, y, t) = \delta(x - x_1 - v_x t) \delta(y - y_1 - v_y t).$$

The 3-D Fourier transform of $S(x, y, t)$ in the continuous case is given as

$$\begin{aligned} & \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x - x_1 - v_x t) \delta(y - y_1 - v_y t) \\ & \quad \times e^{-j(\omega_x x + \omega_y y + \omega_t t)} dx dy dt \\ & = e^{-j(\omega_x x_1 + \omega_y y_1)} \int_{-\infty}^{\infty} e^{-j(\omega_x v_x + \omega_y v_y + \omega_t) t} dt \\ & = 2\pi e^{-j(\omega_x x_1 + \omega_y y_1)} \delta(\omega_x v_x + \omega_y v_y + \omega_t). \end{aligned} \quad (26)$$

Equation (26) indicates that the transformed function is non-zero only on the inclined spectral plane defined by

$$\omega_x v_x + \omega_y v_y + \omega_t = 0. \quad (27)$$

Thus, the Fourier transform components are concentrated on a space passing through the origin in the coordinate system $(\omega_x, \omega_y, \omega_t)$. The magnitude response is uniform and the phase response is linear.

If clutter and background noise are added to the image, the 3-D Fourier transform does not show the regularity of mass distribution as in (27). To perform the filtering, those components with coordinates $(\omega_x, \omega_y, \omega_t)$ which do not satisfy (27) are set to zero, and the rest of the components are passed without change. An inverse 3-D transform on the filtered data generates an image with straight lines, which would satisfy (27) in the frequency domain.

Since the Fourier transform is shift invariant, one only needs to design filters for lines in different directions regardless of their locations. Each line with different direction requires its own directional filter. The filters were generated with lines rotated every 15° in all the four quadrants, hence giving 24 possible filters. Each line consists of 10 points for detection in 10 scans.

Problem arises when digital processing is performed. To explain this, let us assume that the image is of size $N \times N$, and the time variable is uniformly sampled, i.e., time units are t_1, t_2 , and so forth up to t_{n_0} , where n_0 is the number of scans used

for detection. Equation (27) is satisfied only when images are in continuous case or in discrete case with $n_0 = N$ and $v_x = v_y = 1$. If the data is discrete and does not meet these conditions, then sidelobes are created and the magnitude response is not uniform.

To design the directional filters in discrete space, thresholding on the magnitude response of the transformed discrete line is performed to generate a set of binary data. This binary data represents the response of the directional filters which passes frequency components corresponding to line features in the spatio-temporal domain. The choice of the threshold depends on the number of missing points allowed. That is, suppose we have a clean image consisting of 10 scans within a window of size 30×30 and 4 target points are allowed to be misdetected by the sensors, then the magnitude of the frequency response at the origin is $(10 - 4)/\sqrt{30 \times 30 \times 10} = 0.0632$ with zero phase. Thus, the threshold should be chosen less than this maximum component.

To detect the presence of a target track in a cluttered environment in the discrete case, similar processes as in the continuous case are performed. In frequency domain, the transformed data is multiplied by the frequency response of designed filters. An inverse 3-D Fourier transform on the filtered data gives an analog image, thus thresholding is used again to obtain a binary image.

The important features of the 3-D filtering method are shift invariant property, relative insensitivity to missing target points, and independency of computation on the size of windows. However, as may be noted from (27) v_x and v_y are assumed to be constants implying that the target moves with uniform velocities which is not true in practical situations. This is a serious drawback of 3-D filtering method. Different number of zeros between two consecutive target points represents different target speed. These zeros are considered as missing target points by the process which may be acceptable to a limited amount. It would be much more complicated to design directional filters to deal with variable speed and arbitrary curvature cases. The other shortcomings of this filtering scheme include the following.

- 1) Its detection capability is limited to the cases that the filters were designed for.
- 2) It is computationally intensive to pass the data through all the filters in the bank.
- 3) The procedure to design the filter bank is computationally laborious.
- 4) Filters are not flexible to deal with variable situations. For examples, if the size of the image or the number of scans used for detection are changed, the filters would not be able to deal with the new environment. In such cases a new set of filters needs to be designed.
- 5) The scan number at which the decision must be made is not flexible. That is, if 10 scans were used to

design the filters, then the decision cannot be made until all the data points for 10 scans are available, even though the characteristics of the track may be apparent before the 10th scan.

VI. RESULTS AND DISCUSSION

This section presents the results of applying the high order correlation methods and the 3-D filtering described in the previous sections to six data files containing different target events. We first present the results of the original and the modified high order correlation methods.

The total number of scans used for detection is $n_0 = 10$. Several experiments were conducted which indicated that the images digitized into matrices of 512×512 pixels captured all the target information. The correlations were evaluated in windows of size 30×30 pixels. Windows were swept across the image from bottom to top and left to right with 15-pixel overlap between the adjacent windows. Windows which lie in blank areas and contain no information were skipped. In addition, if the first half of the bottom row of the window was blank the window would be moved up by 1 pixel. Using this process only a maximum of 656 windows were used as opposed to 1100 windows which would have been needed if standard windowing had been used. This obviously improves the speed of operations significantly when implemented on a serial computer. Once the windows were selected, the spatio-temporal high order correlations were generated using (8). In this process k (order of recursion) was varied from 1 to 5 and thus n_{max} was changed from 9 to 5. It was assumed that the maximum velocity is $v = 4$ pixels/scan. At each step of the process of evaluating the high order correlations, a threshold was used to determine a possible target window. If the sum of high order correlations in a window was higher than the chosen threshold then the high order correlation image was generated recursively, otherwise the window was considered to be a nontarget window and its content can be removed as clutter. Usually the nontarget windows can be distinguished before the second order correlations are performed. The threshold was chosen to be $n_{max} - 2$ for different values of k which takes into account up to 2 missing points in 10 scans.

Table II summarizes the results on the chosen data files using the original high order correlation method. The second column in the table is the total number of windows for the image of the relevant target event. The third column is the number of windows that were considered to be possible target windows after the detection process. The target detection rates which indicate the percentage of the target information (windows) retained in the filtered results are shown in the fourth column. The clutter rejection rates which

TABLE II
High Order Correlation Results

Target event	Total windows	Possible target windows	Target detection rate	Clutter rejection rate
10001	372	26	100%	93.26%
20001	656	2	100%	99.84%
30001	582	4	100%	99.45%
40003	507	24	100%	95.26%
50001	593	11	100%	98.31%
60001	530	6	100%	99.24%

TABLE III
Modified High Order Correlation Results

Target event	Total windows	Possible target windows	Target detection rate	Clutter rejection rate
10001	372	8	100%	98.38%
20001	656	1	100%	100%
30001	582	1	100%	100%
40003	507	15	100%	97.23%
50001	593	4	100%	99.66%
60001	530	3	100%	99.81%

TABLE IV
3-D Filtering Results, Window Size = 30 x 30

Target event	Total windows	Possible target windows	Target detection rate	Clutter rejection rate
10001	360	90	100%	75.42%
20001	631	45	100%	93.16%
30001	565	79	100%	86.32%
40003	502	136	100%	73.20%
50001	586	122	0%	79.45%
60001	530	139	0%	74.05%

indicate the percentage of the nontarget windows removed are given in the fifth column. Note that there could be several windows that cover the same target track. Thus, the overall clutter rejection rates could be higher than those presented in this column.

The imperfect clutter rejection ratio in the results can be attributed primarily to two main factors: 1) clutter which has returns similar in shape to a jagged track and satisfy the spatio-temporal correlation criterion described before, and 2) clutter which persistently occur at the same location every scan.

The results of using the modified high order correlation method are shown in Table III. The maximum moving curvature was assumed to be $\pi/3$ rad/s. The maximum order of correlation was 3 in this case. The average central processing unit (CPU) time was approximately 15 min on an IBM RS/6000 machine. The performance, as expected, has been improved due to the incorporation of the target motion dynamics into the process. Thus, the first factor responsible for imperfect clutter rejection rate has been alleviated. The improvement in the clutter rejection rate for the modified high order correlation method is even more evident when the density of clutter is high.

For 3-D filtering, a bank of 24 directional filters were designed which allowed a maximum of 4 missing

target points. The results are shown in Table IV. The average CPU time required for 3-D filtering was about 30 min on the same computing environment.

The misdetection in 3-D filtering scheme was not only caused by missing points when windows were shifted, but also by the nonuniform target speeds. The images used to design each directional filter contained target movements in a specific direction with uniform speeds 1, 2, and 3 pixels per scan. As a result, each image had three target tracks along the same line but with different lengths in order to consider limited cases of varying target speed. Obviously, the filters which allowed 4 maximum missing points were more adaptive to varying target speeds but they also passed more clutter. The algorithm would perform better when the target moves at a uniform speed on a straight line track. Event 10001 was an example of such cases.

For some target events only part of the target points remained in the filtered results and the original curved shapes of the tracks were not retained. Those target points remained typically form straight line segments which pass the directional filters.

Experiments were also conducted using a different window size, i.e., 60×60 , the results of which are given in Table V. As can be seen, increasing the size of the window will not change the detection and clutter

TABLE V
3-D Filtering Results, Window Size = 60 x 60

Target event	Total windows	Possible target windows	Target retained	Clutter rejection rate
10001	126	47	100%	63.71%
20001	183	25	100%	87.36%
30001	165	46	100%	73.01%
40003	143	67	100%	53.90%
50001	161	68	0%	58.49%
60001	144	62	0%	57.75%

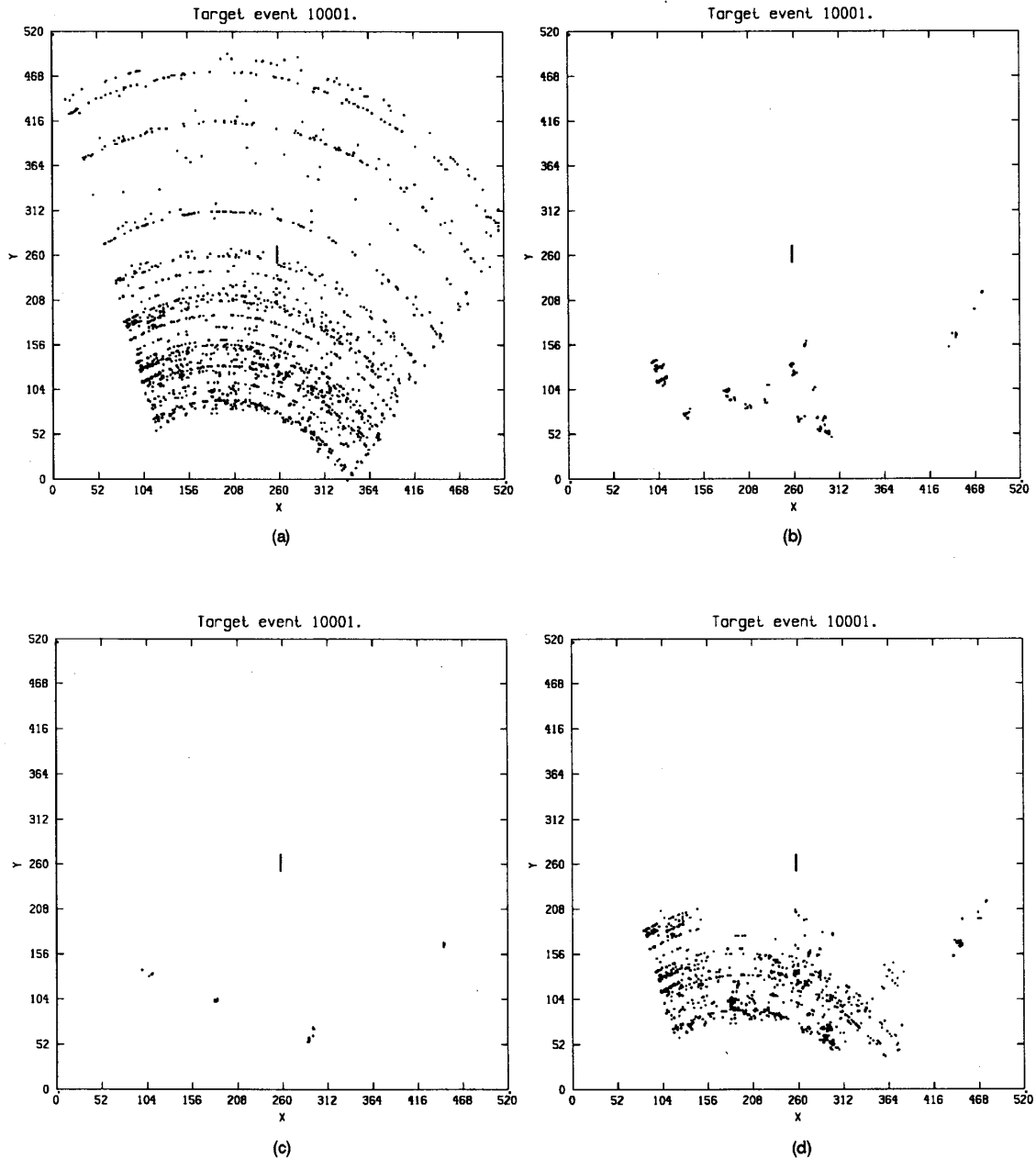


Fig. 3. (a) Original data event 10001 after 10 scans. (b) Processed data using high order correlations. (c) Processed data using modified high order correlations. (d) Processed data using 3-D filtering.

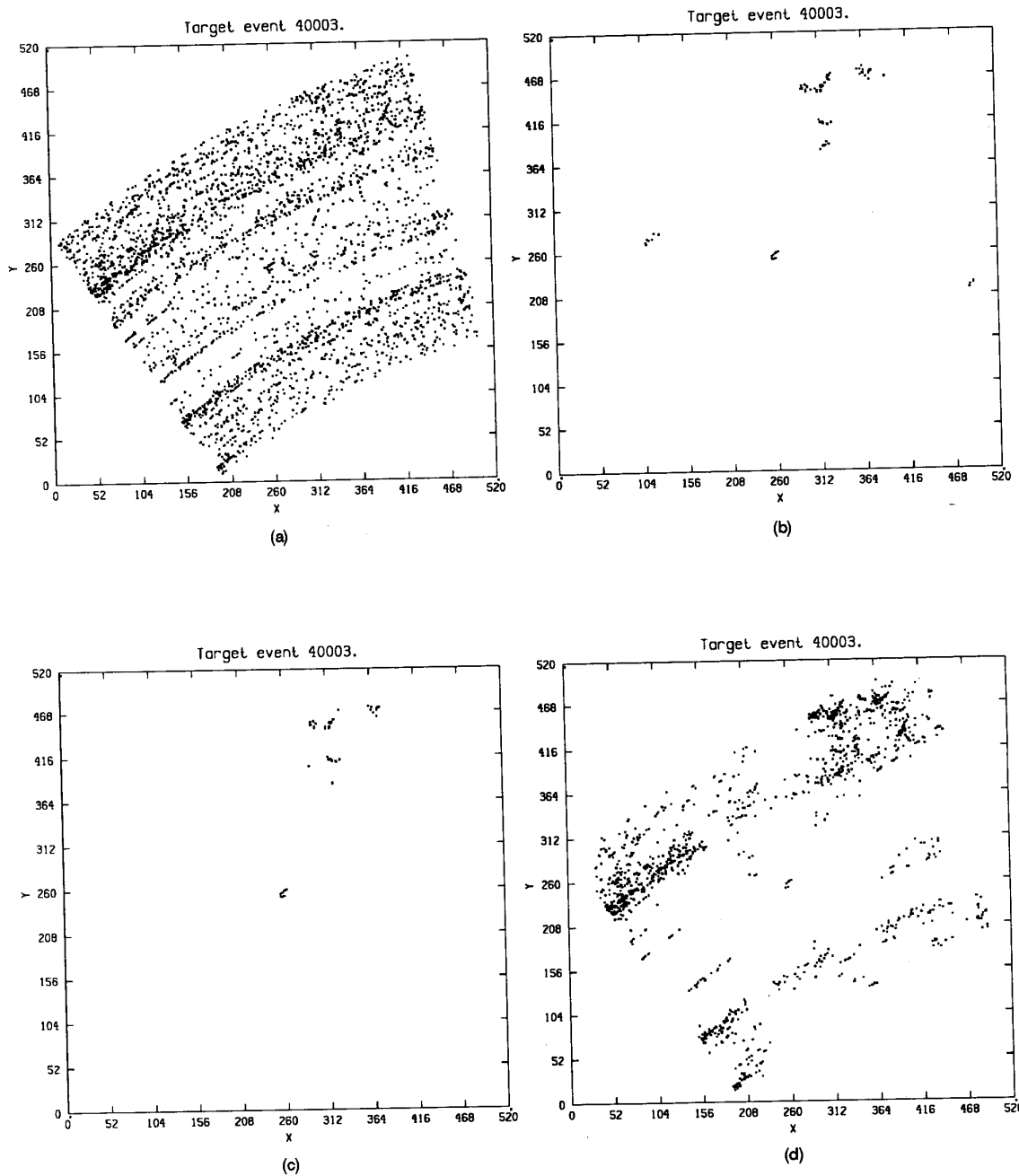


Fig. 4. (a) Original data event 40003 after 10 scans. (b) Processed data using high order correlations. (c) Processed data using modified high order correlations. (d) Processed data using 3-D filtering.

rejection rates significantly. Thus, in contrast to the high order correlation methods which compute the total correlations inside a window, 3-D filtering finds the desired features which exist in the image and hence the window size does not have an impact on the results.

The graphical representations of the original and the processed images using the original, modified high order correlation and 3-D filtering schemes for three selected events are shown in Figs. 3-5. All the targets are located in the center of the field. These images clearly indicate the effectiveness of the proposed schemes.

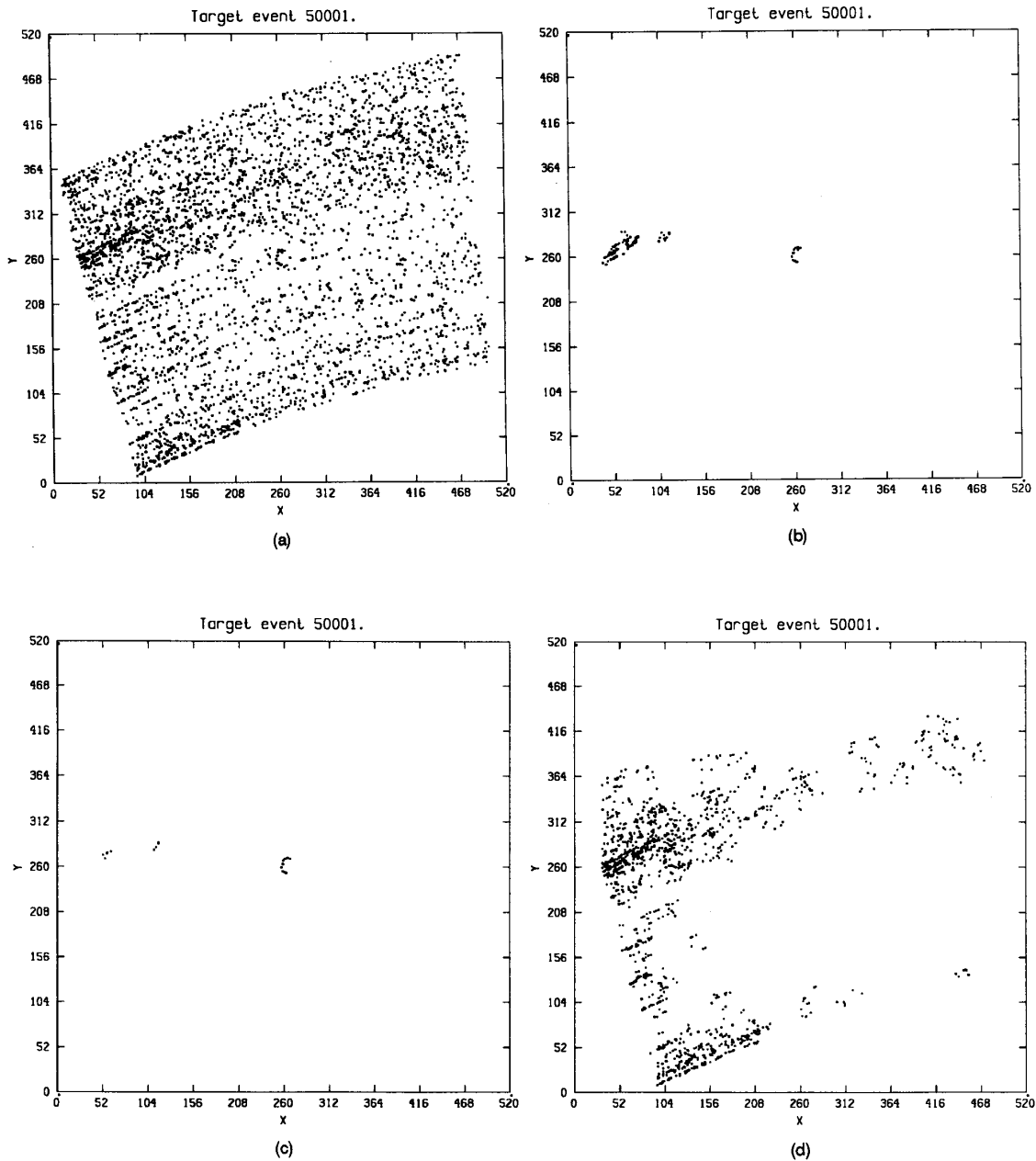


Fig. 5. (a) Original data event 50001 after 10 scans. (b) Processed data using high order correlations. (c) Processed data using modified high order correlations. (d) Processed data using 3-D filtering.

VII. CONCLUSIONS

In this paper a new method for dim moving target detection and clutter rejection is developed which uses high order spatio-temporal correlations to discriminate actual target tracks from false positives. The original high order correlation scheme provides an average clutter rejection rate of 95% without disturbing the target information. The important properties of this

method are also presented. This scheme can be implemented on a multiplanar connectionist network which generates the filtered images in parallel/pipeline fashion. The high order correlation is then modified to incorporate target motion dynamics. This modified high order correlation method which computes the correlations among three consecutive scans (rather than two in the original version) offers an average clutter rejection rate of 98%. The implementation of

clutter rejection rate of 98%. The implementation of this scheme using high order neural network paradigm is also presented. An alternative conventional method using 3-D filtering is also examined. Simulation results on six target events are provided which indicate the effectiveness of the proposed schemes in this paper.

REFERENCES

- [1] Bhanu, B. (1986)
Automatic target recognition: State of the art survey.
IEEE Transactions on Aerospace and Electronic Systems, 22, 4 (July 1986), 364-379.
- [2] Reed, I. S., Gagliardi, R. M., and Shao, H. M. (1983)
Application of three dimensional filtering to moving target detection.
IEEE Transactions on Aerospace and Electronic Systems, AES-19, 6 (Nov. 1983), 898-905.
- [3] Mohanty, N. C. (1981)
Computer tracking of moving targets in space.
IEEE Transactions on Pattern Analysis Machine Intelligence, PAMI-3, 5 (Sept. 1981), 606-611.
- [4] Chen, J. Y., and Reed, I. S. (1987)
A detection algorithm for optical targets in clutter.
IEEE Transactions on Aerospace and Electronic Systems, AES-23, 1 (Jan. 1987), 46-59.
- [5] Barniv, Y. (1985)
Dynamic programming solution for detecting dim moving targets.
IEEE Transactions on Aerospace and Electronic Systems, AES-21, 1 (Jan. 1985), 144-156.
- [6] Porat, B., and Friedlander, B. (1990)
A frequency domain algorithm for multiframe detection and estimation of dim targets.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-12, 4 (Apr. 1990), 398-401.
- [7] Bar-Shalom, Y., and Fortman, T. E. (1988)
Tracking and Data Association.
New York: Academic Press, 1988.
- [8] Roth, M. (1990)
Survey of neural network technology for automatic target recognition.
IEEE Transactions on Neural Networks, 1, 1 (Mar. 1990), 28-43.
- [9] Roth, M. (1989)
Neural networks for extraction of weak targets in high clutter environments.
IEEE Transactions on Systems, Man, and Cybernetics, 19, 10 (Oct. 1989), 1210-1217.
- [10] Khotanzad, A., Lu, J. H., and Srinath, M. D. (1990)
Target detection using a neural network based passive sonar system.
Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP'90, Apr. 1990, I-335-340.
- [11] Liou, R., Azimi-Sadjadi, M. R., and Dent, R. (1991)
Detection of dim targets in high cluttered background using high order correlation neural network.
In *Proceedings of the IJCNN* Seattle, July 1991, 701-706.
- [12] Liou, R. J., and Azimi-Sadjadi, M. R. (1992)
Dim target detection and clutter rejection using modified high order correlation neural network.
In *Proceedings of the IJCNN*, Baltimore, June 1992, IV-289-294.
- [13] Giles, C. L., and Maxwell, T. (1987)
Learning, invariance, and generalization in high-order neural networks.
Applied Optics, 26, 23 (Dec. 1987), 4972-4978.
- [14] Rumelhart, D. E., McClelland, J. L., and PDP Research Group (1986)
Parallel Distributed Processing.
Cambridge, MA: MIT Press, 1986.



Ren-Jean Liou (S'87) received a diploma in electrical engineering from Ming-Chi Institute of Technology, Taipei, Taiwan, R.O.C., in 1984, and the M.S. degree in electrical engineering from Marquette University, Milwaukee, WI, in 1989. At present, he is working toward the Ph.D. degree in electrical engineering at Colorado State University, Fort Collins, CO.

His research interests are in the areas of digital signal/image processing and neural networks.



Mahmood R. Azimi-Sadjadi (M'81—SM'89) was born in Tehran, Iran, in 1952. He received the B.S. degree from the University of Tehran, Iran, in 1977, and the M.S. and Ph.D. degrees from Imperial College, University of London, England, in 1978 and 1982, respectively, all in electrical engineering.

He served as an Assistant Professor in the Department of Electrical and Computer Engineering, University of Michigan-Dearborn. Since July 1986 he has been with the Department of Electrical Engineering, Colorado State University, where he is now an Associate Professor. His areas of interest are digital signal/image processing, multidimensional system theory and analysis, adaptive filtering, system identification, and neural networks.

Dr. Azimi-Sadjadi is a recipient of the 1990 BATTELLE Summer Faculty Fellowship award, and the 1984 DOW chemical Outstanding Young Faculty Award of the American Society for Engineering Education. He is co-author of the book, *Digital Filtering in One and Two Dimensions*, Plenum Press, 1989.