A VERSATILE ALGORITHM FOR TWO-DIMENSIONAL SYMMETRIC NONCAUSAL MODELING

George-Othon Glentis, Cornelis H. Slump and Otto E. Herrmann

University of Twente
Department of Electrical Engineering
Control, Systems and Computer Engineering Group (BSC),
Laboratory for Network Theory and VLSI-Design,
P.O. Box 217, 7500 AE Enschede, The Netherlands.

ABSTRACT

In this paper a novel algorithm is presented for the efficient Two-Dimensional (2-D) symmetric noncausal Finite Impulse Response (FIR) filtering and Autoregressive (AR) modeling. Symmetric filter masks of general boundaries are allowed. The proposed algorithm offers the greatest maneuverability in the 2-D index space in a computational efficient way. This flexibility can be taken into advantage if the shape of the 2-D mask is not a priori known and has to be dynamically configured.

1. PROBLEM FORMULATION

Let $x(n_1, n_2)$ be the input of a linear, space invariant 2-D FIR filter. The filter's output $y(n_1, n_2)$ is a linear combination of past input values $x(n_1 - i_1, n_2 - i_2)$ weighted by the *filter coefficients* c_{i_1,i_2} over a convex support region, or *filter mask*, $\mathcal{M}$

$$y(n_1, n_2) = - \sum_{(i_1,i_2)\in\mathcal{M}} c_{i_1,i_2} x(n_1 - i_1, n_2 - i_2) \quad (1)$$

The filter is restricted to be linear-phase. Thus, the following conditions should be satisfied, [1],

$$\begin{array}{lll} \text{Mask symmetry} & \forall (i_1, i_2) \in \mathcal{M}, & \exists\, (-i_1, -i_2) \in \mathcal{M} \\ \text{Coeff. symmetry} & & c_{i_1,i_2} = c_{-i_1,-i_2} \end{array}$$

Given an input 2-D signal $x(n_1, n_2)$ and a desired response 2-D signal $z(n_1, n_2)$ the optimal Mean Squared Error (MSE) 2-D FIR filter is obtained by minimizing the cost function

$$\mathcal{E}\Big[(z(n_1, n_2) - y(n_1, n_2))^2\Big] \quad (2)$$

$\mathcal{E}[.]$ is the expectation operator. MSE 2-D linear prediction can be handled as a special case of filtering, setting $z(n_1, n_2) = x(n_1, n_2)$ and excluding the origin $\{(0,0)\}$ from the filter mask, i.e., $(i_1, i_2) \in \mathcal{M} - \{(0,0)\}$.

2. 2-D SYMMETRIC SUPPORT REGION

Consider the support region depicted on Figure 1. More precisely $\mathcal{M}$ consists of a union of intervals

$$\mathcal{M} = \cup_{i_1=-k_1}^{k_1} \mathbf{m}(i_1)$$

$$\mathbf{m}(i_1) = \{(i_1, i_2) : -k_2(-i_1) \le i_2 \le k_2(i_1)\}$$

Clearly,

$$k_1 = \max\{i_1 : (i_1, i_2) \in \mathcal{M}\}$$

$$k_2(i_1) = \max\{i_2 : (i_1, i_2) \in \mathbf{m}(i_1)\}$$

Then, eq. (1) takes the form

$$y(n_1, n_2) = - \sum_{i_1=-k_1}^{k_1} \sum_{i_2=-k_2(-i_1)}^{k_2(i_1)} c_{i_1,i_2} x(n_1 - i_1, n_2 - i_2)$$

The above equation can be written as a linear regression

$$y(n_1, n_2) = -\mathcal{X}^t_{\mathcal{M}}(n_1, n_2)\mathcal{C}_{\mathcal{M}} \quad (3)$$

where the regressor (data vector) and the filter coefficients vector are defined by

$$\mathcal{X}_{\mathcal{M}}(n_1, n_2) = [\ \mathbf{x}^t_{m(-k_1)}(n_1, n_2)\ \mathbf{x}^t_{m(-k_1+1)}(n_1, n_2) \ldots \mathbf{x}^t_{m(k_1-1)}(n_1, n_2)\ \mathbf{x}^t_{m(k_1)}(n_1, n_2)\ ]^t \quad (4)$$

$$\mathcal{C}_{\mathcal{M}} = \Big[\mathbf{c}^t_{m(-k_1)}\ \mathbf{c}^t_{m(-k_1+1)} \cdots \mathbf{c}^t_{m(k_1-1)}\ \mathbf{c}^t_{m(k_1)}\Big]^t \quad (5)$$

The work of Dr. Glentis is supported by the European Community, Human Capital Mobility, *EUROCHIP* program.

0-8186-6950-0/94 $4.00 © 1994 IEEE

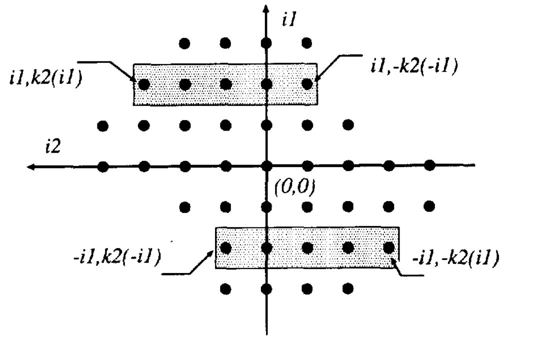

Figure 1: Symmetric support region

and

$$\mathbf{x}_{m(i_1)}(n_1,n_2) = \Big[x(n_1-i_1,n_2+k_2(-i_1))$$

$$\ldots\ x(n_1-i_1,n_2-k_2(i_1)+1)\ \ x(n_1-i_1,n_2-k_2(i_1))\Big]^t$$

$$\mathbf{c}_{m(i_1)} = \Big[c_{i_1,-k_2(-i_1)}\ \cdots\ c_{i_1,k_2(i_1)}\Big]^t$$

The filter coefficients symmetry implies that

$$\mathcal{C}_{\mathcal{M}} = \mathcal{J}\mathcal{C}_{\mathcal{M}} \tag{6}$$

where $\mathcal{J}$ is a matrix with ones in the antidiagonal, zeros elsewhere. Clearly, $\mathcal{J}\mathcal{J} = \mathbf{I}$. Minimization of (2), subject to the symmetry constraint (6), leads to the following linear system of equations, [3]

$$\big(\mathcal{R}_{\mathcal{M}} + \mathcal{J}\mathcal{R}_{\mathcal{M}}\mathcal{J}\big)\,\mathcal{C}_{\mathcal{M}} = -\big(\mathcal{D}_{\mathcal{M}} + \mathcal{J}\mathcal{D}_{\mathcal{M}}\mathcal{J}\big) \tag{7}$$

where $\mathcal{R}_{\mathcal{M}} = \mathcal{E}\big[\mathcal{X}_{\mathcal{M}}(n_1,n_2)\mathcal{X}^t_{\mathcal{M}}(n_1,n_2)\big]$ is the input signal autocorrelation matrix, and $\mathcal{D}_{\mathcal{M}} = \mathcal{E}\big[\mathcal{X}_{\mathcal{M}}(n_1,n_2)z(n_1,n_2)\big]$ is the crosscorrelation vector between the input and the desired response signal.

In the sequel, real and homogeneous, random, wide-sense stationary 2-D signals will be considered. This implies that the autocorrelation between two samples depends on the difference of their coordinates

$$\mathcal{E}\Big[x(n_1-i_1,n_2-i_2)\ x(n_1-j_1,n_2-j_2)\Big] =$$

$$\rho(i_1-j_1,i_2-j_2),\quad \rho(i,j) = \rho(-i,-j) \tag{8}$$

The autocorrelation matrix $\mathcal{R}_{\mathcal{M}}$ is a block matrix of block order $2k_1+1$

$$\mathcal{R}_{\mathcal{M}} = \Big[\mathbf{R}(i_1,j_1)\Big]_{\substack{i_1=-k_1\ldots k_1\\ j_1=-k_1\ldots k_1}}$$

with entry Toeplitz matrices of the form

$$\mathbf{R}(i_1,j_1) = \mathcal{E}\Big[\mathbf{x}_{m(i_1)}(n_1,n_2)\mathbf{x}^t_{m(j_1)}(n_1,n_2)\Big] =$$

$$\Big[\rho(i_1-j_1,i_2-j_2)\Big]_{\substack{i_2=-k_2(-i_1)\ldots k_2(i_1)\\ j_2=-k_2(-j_1)\ldots k_2(j_1)}}$$

$\mathcal{D}_{\mathcal{M}}$ is a block vector $\mathcal{D}_{\mathcal{M}} = \big[\mathbf{d}(i_1)\big]_{i_1=-k_1\ldots k_1}$ with entry subvectors $\mathbf{d}(i_1) = \big[d(i_1,i_2)\big]_{i_2=-k_2(-i_1)\ldots k_2(i_1)}$ where $d(i_1,i_2) = \mathcal{E}[x(n_1-i_1,n_2-i_2)z(n_1,n_2)]$

In addition to the block Toeplitz structure, the autocorrelation matrix is perisymmetric, as it follows from the symmetry of the support region and eq. (8), i.e.,

$$\mathcal{R}_{\mathcal{M}} = \mathcal{J}\mathcal{R}_{\mathcal{M}}\mathcal{J} \tag{9}$$

Thus, the normal equations (7) take the form

$$\mathcal{R}_{\mathcal{M}}\mathcal{C}_{\mathcal{M}} = -\mathcal{D}^s_{\mathcal{M}} \tag{10}$$

where $\mathcal{D}^s_{\mathcal{M}} = 1/2\big(\mathcal{D}_{\mathcal{M}} + \mathcal{J}\mathcal{D}_{\mathcal{M}}\big)$

3. DATA PARTITIONS

Efficient order recursive algorithms for 1-D, as well as for 2-D, MSE filtering are based on suitable partitions of the data parameters that utilize time, or spatial, shift invariance properties, [2]-[6].

Let us consider the increased order support region

$$\mathcal{M}^{i_1} = \mathcal{M} \cup \{(i_1,k_2(i_1)+1)\} \cup \{(-i_1,-k_2(i_1)-1)\}$$

The passage from $\mathcal{M}$ to the increased order mask $\mathcal{M}^{i_1}$ can be accomplished either by a double step as described above, or by a sequence of two single steps as

$$\mathcal{M} + L(i_1) = \mathcal{M} \cup \{(i_1,k_2(i_1)+1)\}$$

$$\mathcal{M}^{i_1} = \mathcal{M} + L(i_1) + R(-i_1) =$$

$$\Big(\mathcal{M} + L(i_1)\Big) \cup \{(-i_1,-k_2(i_1)-1)\}$$

$L(i_1)$ stand for a single step increment at the left hand side of the i_1-th row. $R(-i_1)$ stand for a single step increment at the right hand side of the $-i_1$-th row.

Let us consider the increased order data vector corresponding to the augmented support region $\mathcal{M}^{i_1}$. It is partitioned as

$$\mathcal{X}_{\mathcal{M}^{i_1}}(n_1,n_2) = \mathcal{W}^t_{i_1}\begin{bmatrix} x(n_1+i_1,n_2+k_2(i_1)+1) \\ \mathcal{X}_{\mathcal{M}}(n_1,n_2) \\ x(n_1-i_1,n_2-k_2(i_1)-1) \end{bmatrix} \tag{11}$$

Alternatively, it can be partitioned in two steps as

$$\boldsymbol{\mathcal{X}}_{\mathcal{M}^{i_1}}(n_1,n_2) = \mathcal{T}^t_{R(-i_1)} \begin{bmatrix} x(n_1+i_1, n_2+k_2(i_1)+1) \\ \boldsymbol{\mathcal{X}}_{\mathcal{M}+L(i_1)}(n_1,n_2) \end{bmatrix} \tag{12}$$

$$\boldsymbol{\mathcal{X}}_{\mathcal{M}+L(i_1)}(n_1,n_2) =$$

$$\mathcal{S}^t_{L(i_1)} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\mathcal{M}}(n_1,n_2) \\ x(n_1-i_1, n_2-k_2(i_1)-1) \end{bmatrix} \tag{13}$$

W_{i_1}, $\mathcal{T}_{R(-i_1)}$ and $\mathcal{S}_{L(i_1)}$ are suitable permutation matrices. The increased order masks considered so far, correspond to a single order increment of the filter configuration, along the i_1-th row of the filter mask. It is necessary to consider block step increments of the filter mask. Indeed, let

$$\boldsymbol{\mathcal{M}} + \mathbf{L} = \boldsymbol{\mathcal{M}} \cup_{i_1=-k_1}^{k_1} \{(i_1, k_2(i_1)+1)\} \tag{14}$$

The corresponding data vector is partitioned as

$$\boldsymbol{\mathcal{X}}_{\mathcal{M}+L}(n_1,n_2) = \mathcal{T}^t_L \begin{bmatrix} \mathbf{x}^f(n_1,n_2-1) \\ \boldsymbol{\mathcal{X}}_{\mathcal{M}}(n_1,n_2-1) \end{bmatrix} =$$

$$\boldsymbol{\mathcal{S}}^t_L \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\mathcal{M}+L(i_1)}(n_1,n_2) \\ \mathbf{x}^b_{i_1}(n_1,n_2) \end{bmatrix} \tag{15}$$

Vectors $\mathbf{x}^f(n_1,n_2)$ and $\mathbf{x}^b_{i_1}(n_1,n_2)$ are defined as

$$\mathbf{x}^f(n_1,n_2) = \Big[x(n_1+\ell, n_2+k_2(-\ell))\Big]_{\ell=-k_1\ldots k_1}$$

$$\mathbf{x}^b_{i_1}(n_1,n_2) = \Big[x(n_1-\ell, n_2-k_2(\ell)-1)\Big]_{\substack{\ell=-k_1\ldots k_1 \\ \ell\neq i_1}}$$

$\boldsymbol{\mathcal{T}}_L$ and $\boldsymbol{\mathcal{S}}_L$ are suitable permutation matrices.

4. THE ALGORITHM

Consider the increased order system (10) corresponding to the augmented support region $\boldsymbol{\mathcal{M}}^{i_1}$,

$$\boldsymbol{\mathcal{R}}_{\mathcal{M}^{i_1}} \boldsymbol{\mathcal{C}}_{\mathcal{M}^{i_1}} = -\boldsymbol{\mathcal{D}}^s_{\mathcal{M}^{i_1}}$$

Taking into account the data partition (11), we obtain

$$\boldsymbol{\mathcal{R}}_{\mathcal{M}^{i_1}} = \mathcal{W}^t_{i_1} \begin{bmatrix} r^{fo} & \widehat{\mathbf{r}}^{i_1 t}_{\mathcal{M}} & r^{bo} \\ \widehat{\mathbf{r}}^{i_1}_{\mathcal{M}} & \boldsymbol{\mathcal{R}}_{\mathcal{M}} & \mathbf{r}^{i_1}_{\mathcal{M}} \\ r^{bo} & \mathbf{r}^{i_1 t}_{\mathcal{M}} & r^{fo} \end{bmatrix} \mathcal{W}_{i_1} \tag{16}$$

where $r^{fo} = \rho(0,0)$, $r^{bo} = \rho(2i_1, 2k_2(i_1)+2)$, and

$$\mathbf{r}^{i_1}_{\mathcal{M}} = \mathcal{E}\Big[\boldsymbol{\mathcal{X}}_{\mathcal{M}}(n_1,n_2)x(n_1-i_1, n_2-k_2(i_1)-1)\Big] \tag{17}$$

$$\widehat{\mathbf{r}}^{i_1}_{\mathcal{M}} = \mathcal{E}\Big[\boldsymbol{\mathcal{X}}_{\mathcal{M}}(n_1,n_2)x(n_1+i_1, n_2+k_2(i_1)+1)\Big] \tag{18}$$

Clearly, $\widehat{\mathbf{r}}^{i_1}_{\mathcal{M}} = \mathcal{J}\mathbf{r}^{i_1}_{\mathcal{M}}$. Moreover,

$$\boldsymbol{\mathcal{D}}^s_{\mathcal{M}^{i_1}} = \mathcal{W}^t_{i_1} \begin{bmatrix} d^s(i_1, k_2(i_1)+1) \\ \boldsymbol{\mathcal{D}}^s_{\mathcal{M}} \\ d^s(i_1, k_2(i_1)+1) \end{bmatrix} \tag{19}$$

where $d^s(i_1, k_2(i_1)+1) = 1/2(d(i_1, k_2(i_1)+1) + d(-i_1, -k_2(i_1)-1))$

Application of the matrix inversion lemma for partitioned matrices leads to a recursive estimation of the increased order filter, eqs. (1)-(5) of Table 1. Auxiliary vector $\mathbf{q}^{i_1}_{\mathcal{M}}$ is defined as

$$\boldsymbol{\mathcal{R}}_{\mathcal{M}} \mathbf{q}^{i_1}_{\mathcal{M}} = -(\mathbf{r}^{i_1}_{\mathcal{M}} + \mathcal{J}\mathbf{r}^{i_1}_{\mathcal{M}}) \tag{20}$$

or

$$\mathbf{q}^{i_1}_{\mathcal{M}} = \mathbf{b}^{i_1}_{\mathcal{M}} + \mathcal{J}\mathbf{b}^{i_1}_{\mathcal{M}} \quad \boldsymbol{\mathcal{R}}_{\mathcal{M}} \mathbf{b}^{i_1}_{\mathcal{M}} = -\mathbf{r}^{i_1}_{\mathcal{M}} \tag{21}$$

To be able to develop an order recursive algorithm for the determination of the optimum filter $\boldsymbol{\mathcal{C}}_{\mathcal{M}}$, recursions for updating $\mathbf{q}^{\ell}_{\mathcal{M}}$, or equivalently $\mathbf{b}^{\ell}_{\mathcal{M}}$, $\ell = -k_1 \ldots k_1$, are required.

Let us consider the increased order linear system corresponding to the augmented mask $\boldsymbol{\mathcal{M}}^{i_1}$

$$\boldsymbol{\mathcal{R}}_{\mathcal{M}^{i_1}} \mathbf{b}^{\ell}_{\mathcal{M}^{i_1}} = -\mathbf{r}^{\ell}_{\mathcal{M}^{i_1}}, \quad \ell = -k_1 \ldots k_1$$

where $\mathbf{r}^{\ell}_{\mathcal{M}^{i_1}} =$

$$\mathcal{E}\Big[\boldsymbol{\mathcal{X}}_{\mathcal{M}^{i_1}}(n_1,n_2)x(n_1-\ell, n_2-k_2(\ell)-1-\delta(\ell-i_1))\Big]$$

$\delta(n)$ is the discrete time Dirac function, i.e., $\delta(n) = 1$ for $n = 0$ and $\delta(n) = 0$ for $n \neq 0$.

Parameters can be partitioned using eq. (12) as

$$\boldsymbol{\mathcal{R}}_{\mathcal{M}^{i_1}} = \mathcal{T}^t_{R(-i_1)} \begin{bmatrix} \rho(0,0) & \widehat{\mathbf{r}}^{i_1 t}_{\mathcal{M}+L(i_1)} \\ \widehat{\mathbf{r}}^{i_1}_{\mathcal{M}+L(i_1)} & \boldsymbol{\mathcal{R}}_{\mathcal{M}+L(i_1)} \end{bmatrix} \mathcal{T}_{R(-i_1)}$$

$$\mathbf{r}^{\ell}_{\mathcal{M}^{i_1}} = \mathcal{T}^t_{R(-i_1)}$$

$$\begin{bmatrix} \rho(i_1+\ell, k_2(i_1)+k_2(\ell)+2+\delta(\ell-i_1)) \\ \mathbf{r}^{\ell}_{\mathcal{M}+L(i_1)} \end{bmatrix}$$

where $\widehat{\mathbf{r}}^{i_1}_{\mathcal{M}+L(i_1)} =$

$$\mathcal{E}\Big[\boldsymbol{\mathcal{X}}_{\mathcal{M}+L(i_1)}(n_1,n_2)x(n_1+i_1, n_2+k_2(i_1)+1\Big]$$

$$\mathbf{r}^{\ell}_{\mathcal{M}+L(i_1)} =$$

$$\mathcal{E}\Big[\boldsymbol{\mathcal{X}}_{\mathcal{M}+L(i_1)}(n_1,n_2)x(n_1-\ell, n_2-k_2(\ell)-1-\delta(\ell-i_1))\Big]$$

Application of the matrix inversion lemma results to the eqs. (19)- (21) of Table 1.

Moreover,

$$\mathcal{R}_{\mathcal{M}+L(i_1)} = \mathcal{S}^t_{L(i_1)} \begin{bmatrix} \mathcal{R}_{\mathcal{M}} & \mathbf{r}^{i_1}_{\mathcal{M}+L(i_1)} \\ \mathbf{r}^{i_1 t}_{\mathcal{M}+L(i_1)} & \rho(0,0) \end{bmatrix} \mathcal{S}_{L(i_1)}$$

and

$$\widehat{\mathbf{r}}^{i_1}_{\mathcal{M}+L(i_1)} = \mathcal{S}^t_{L(i_1)} \begin{bmatrix} \mathcal{J}\mathbf{r}^{i_1}_{\mathcal{M}} \\ \rho(2i_1, 2k_2(i_1)+2) \end{bmatrix}$$

and for $\ell \neq i_1$,

$$\mathbf{r}^{\ell}_{\mathcal{M}+L(i_1)} = \mathcal{S}^t_{L(i_1)} \begin{bmatrix} \mathbf{r}^{\ell}_{\mathcal{M}} \\ \rho(i_1-\ell, k_2(i_1)-k_2(\ell)) \end{bmatrix}$$

Thus, eqs. (15)- (16) and (7)- (8) of Table 1 are obtained.

The case $\ell = i_1$ is treated separately. Indeed, using (15) we obtain

$$\mathcal{R}_{\mathcal{M}+L} = \mathcal{T}^t_L \begin{bmatrix} R^{fo} & \widehat{\mathbf{R}}^t_{\mathcal{M}} \\ \widehat{\mathbf{R}}_{\mathcal{M}} & \mathcal{R}_{\mathcal{M}} \end{bmatrix} \mathcal{T}_L =$$

$$\mathcal{S}^t_L \begin{bmatrix} \mathcal{R}_{\mathcal{M}+L(i_1)} & \mathbf{R}^{i_1}_{\mathcal{M}} \\ \mathbf{R}^{i_1 t}_{\mathcal{M}} & R^{bo}_{i_1} \end{bmatrix} \mathcal{S}_L$$

$$\mathbf{r}^{i_1}_{\mathcal{M}+L} = \mathcal{T}^t_L \begin{bmatrix} \boldsymbol{\rho}^f \\ \mathbf{r}^{i_1}_{\mathcal{M}} \end{bmatrix} = \mathcal{S}^t_L \begin{bmatrix} \mathbf{r}^{i_1}_{\mathcal{M}+L(i_1)} \\ \boldsymbol{\rho}^{b(i_1)} \end{bmatrix}$$

where

$$R^{fo} = \Big[\rho(\ell_1-\ell_1, k_2(-\ell_1)-k_2(-\ell_2))\Big]_{\substack{\ell_1=-k_1\ldots k_1 \\ \ell_2=-k_1\ldots k_1}}$$

$$\widehat{\mathbf{R}}_{\mathcal{M}} = \Big[\mathcal{J}\mathbf{r}^{\ell}_{\mathcal{M}}\Big]_{\ell=-k_1\ldots k_1}$$

$$\mathbf{R}_{\mathcal{M}+L(i_1)} = \Big[\mathbf{r}^{\ell}_{\mathcal{M}+L(i_1)}\Big]_{\substack{\ell=-k_1\ldots k_1 \\ \ell\neq i_1}}$$

$$\boldsymbol{\rho}^{b(i_1)} = \Big[\rho(\ell-i_1, k_2(\ell)-k_2(i_1)-2)\Big]_{\substack{\ell=-k_1\ldots k_1 \\ \ell\neq i_1}}$$

$$\boldsymbol{\rho}^{f} = \Big[\rho(\ell-i_1, -k_2(-\ell)-k_2(i_1)-1)\Big]_{\substack{\ell=-k_1\ldots k_1 \\ \ell\neq i_1}}$$

The above partitions yield eqs. (10)- (17) of Table 1.

The order recursive equations developed so far, are tight together to form a powerful *true order recursive* 2-D algorithm for symmetric filtering and linear prediction. Indeed, let $\mathcal{M}^{fin}$ be the support region within the search for the optimum mask will be conducted. Let $k_1^{fin} = \max\{i_1 : (i_1, i_2) \in \mathcal{M}^{fin}\}$. Then, for all $i_1 \in [-k_1, k_1], k_1 \leq k_1^{fin}$, anyone of the increased order filters corresponding to a symmetric increment along the i_1 and the $-i_1$ rows of $\mathcal{M}$ can be estimated as $\mathbf{C}_{\mathcal{M}} \rightarrow \mathbf{C}_{\mathcal{M}^{i_1}}$. The update of parameters to a mask that contains extra rows, i.e. going from $[-k_1, k_1] \rightarrow [-k_1-1, k_1+1]$, can be accomplished only for the points laying across the vertical axis. Once the increased order filter that corresponds to $\mathcal{M} \cup \{(0, -k_1-1)\} \cup \{(0, k_1+1)\}$ is determined, further recursions along that row can be performed.

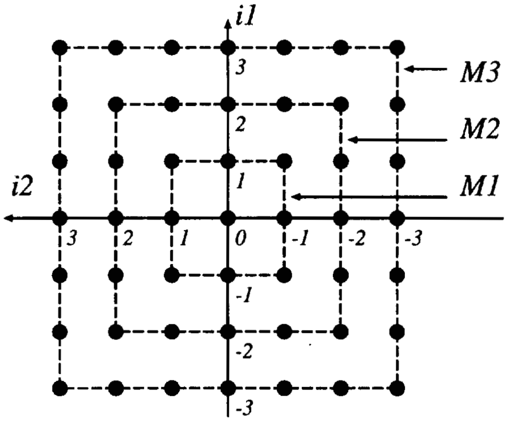

Figure 2: Example

The computational complexity of the algorithm is $O(2k_1P)$ operations per recursion, where $P =$

$\dim(\mathcal{C}_{\mathcal{M}}) = \sum_{i_1=0}^{k_1} 2k_2(i_1) + 1$. For a 2-D filter of a final mask shape $\mathcal{M}^{fin}$, $O\Big(k_1^{fin}(P^{fin})^2\Big)$ operations are required. For the special case of rectangular shaped masks this amount equals to the complexity of the multichannel LWR algorithm, [3]. The non-rectangular mask case cannot be handled by the LWR algorithm, unless overparametrization is utilized. A great advantage the proposed algorithm offer against conventional LWR based counterparts, is the accommodation of masks of general boundaries and the estimation of lower order parameters. Thus, it is established that in all cases the proposed highly efficient order recursive 2-D algorithm performs better than any existing scheme.

The order updating procedure is illustrated using a simple but important support region depicted in Figure 2. Suppose that the MSE filter corresponding to $\mathcal{M}_1$ is known. The estimation of the increased order MSE filter corresponding to $\mathcal{M}_2$ is accomplished following the updating scheme

$$
\begin{gathered}
M1 \cup \{(2,0)\} \cup \{(-2,0)\} \\
\cup \{(2,1)\} \cup \{(-2,-1)\} \\
\cup \{(2,2)\} \cup \{(-2,-2)\} \\
\cup \{(1,2)\} \cup \{(-1,-2)\} \\
\cup \{(0,2)\} \cup \{(0,-2)\} \\
\cup \{(-1,2)\} \cup \{(1,-2)\} \\
\cup \{(-2,2)\} \cup \{(2,-2)\} \\
\cup \{(-2,1)\} \cup \{(2,-1)\}
\end{gathered}
$$

5. CONCLUSIONS

A highly efficient, order recursive algorithm for symmetric 2-D FIR filtering and 2-D system identification has been developed. Symmetric support regions with arbitrary convex shape can be handled. The proposed algorithm offers the greatest possible maneuverability in the 2-D index space. It allows for recursive estimation of the 2-D filter mask shape. The implicit flexibility of the algorithm enables for a dynamical reconfiguration of the mask shape in a computational efficient way.

6. REFERENCES

[1] J. Lim, Two-Dimensional signal and image processing, Prentice-Hall, 1990.

[2] N. Kalouptsidis, 'Efficient computation of multichannel Wiener filters with linear phase,' IEEE Trans. on Circuits and Systems, vol. CAS-35, pp. 433-438, April 1988.

[3] A. Kok, D. Manolakis, and V. Ingle, 'Symmetric noncausal spatial models for 2-D signals with applications in stochastic texture modeling,' Multidimensional Systems and Sig. Proc., pp. 125-149, 1993

[4] G. Glentis and N. Kalouptsidis, 'Fast algorithms for 2-D least squares FIR filtering,' IEEE Int. Conf.-ASSP Toronto, 1991.

[5] G. Glentis and N. Kalouptsidis, 'Efficient Order Recursive Algorithms for Multichannel Least Squares Filtering,' IEEE Transactions on Signal Processing, vol. 40, pp.1354-1374, June 1992.

[6] G. Glentis and N. Kalouptsidis, 'Efficient multichannel FIR filtering using a single step versatile order recursive algorithm,' Signal Processing, vol. 37, pp. 437-462, 1994.

```
$\mathbf{q}^{i_1}_{\mathcal{M}} = \mathbf{b}^{i_1}_{\mathcal{M}} + \mathcal{J}\mathbf{b}^{i_1}_{\mathcal{M}}$    (1)
$\beta_{i_1} = 1/2(d(i_1, k_2(i_1)+1) + d(-i_1, -k_2(i_1)-1)) + \mathbf{r}^{i_1 t}_{\mathcal{M}}\mathcal{C}_{\mathcal{M}}$    (2)
$\alpha^{s}_{i_1} = \rho(0,0) + \rho(2i_1, 2k_2(i_1)+2) + \mathbf{r}^{i_1 t}_{\mathcal{M}}\mathbf{q}^{i_1}_{\mathcal{M}}$    (3)
$k_{i_1} = -\beta_{i_1}/\alpha^{s}_{i_1}$    (4)
$\mathcal{W}_{i_1}\mathcal{C}_{\mathcal{M}^{i_1}} = \begin{bmatrix} 0 \\ \mathcal{C}_{\mathcal{M}} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ \mathbf{q}^{i_1}_{\mathcal{M}} \\ 1 \end{bmatrix} k_{i_1}$    (5)
$\alpha_{i_1} = \rho(0,0) + \mathbf{r}^{i_1}_{\mathcal{M}}\mathbf{b}^{i_1}_{\mathcal{M}}$    (6)
FOR $\ell = -k_1$ TO $k_1$, AND $\ell \neq i_1$, DO
    $\beta^{b\ell}_{i_1} = \rho(i_1 - \ell, k_2(i_1) - k_2(\ell)) + \mathbf{r}^{i_1}_{\mathcal{M}}\mathbf{b}^{\ell}_{\mathcal{M}}$    (7)
    $k^{b\ell}_{i_1} = -\beta^{b\ell}_{i_1}/\alpha_{i_1}$    (8)
    $\mathcal{S}_{i_1}\mathbf{b}^{\ell}_{\mathcal{M}+L(i_1)} = \begin{bmatrix} \mathbf{b}^{\ell}_{\mathcal{M}} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}^{i_1}_{\mathcal{M}} \\ 1 \end{bmatrix} k^{b\ell}_{i_1}$    (9)
ENDFOR $\ell$
$\boldsymbol{\mathcal{A}} = \left[\mathcal{J}\mathbf{b}^{\ell}_{\mathcal{M}}\right]_{\ell=-k_1\ldots k_1}, \boldsymbol{\mathcal{B}}^{i_1} = \left[\mathbf{b}^{\ell}_{\mathcal{M}+L(i_1)}\right]_{\substack{\ell=-k_1\ldots k_1 \\ \ell\neq i_1}}$
$\beta^{b(i_1)}_{L} = \boldsymbol{\rho}^{b(i_1)}_{L} + \boldsymbol{\mathcal{A}}^{t}_{\mathcal{M}}\mathbf{r}^{i_1}_{\mathcal{M}}$    (10)
$\boldsymbol{\alpha}_L = R^{fo} + \widehat{\mathbf{R}}^{t}_{\mathcal{M}}\boldsymbol{\mathcal{A}}_{\mathcal{M}}$    (11)
$K^{b(i_1)}_{L} = -\boldsymbol{\alpha}^{-1}_{L}\beta^{b(i_1)}_{L}$    (12)
$\boldsymbol{\mathcal{T}}_L\mathbf{b}^{i_1}_{\mathcal{M}+L} = \begin{pmatrix} \mathbf{0} \\ \mathbf{b}^{i_1}_{\mathcal{M}} \end{pmatrix} + \begin{pmatrix} \mathbf{I} \\ \boldsymbol{\mathcal{A}}_{\mathcal{M}} \end{pmatrix} K^{b(i_1)}_{L}$    (13)
$\begin{pmatrix} \mathbf{b}^{i_1}_{\mathcal{M}+L(i_1)} \\ \mathbf{0} \end{pmatrix} = \boldsymbol{\mathcal{S}}_R\mathbf{b}^{i_1}_{\mathcal{M}+L} - \begin{pmatrix} \boldsymbol{\mathcal{B}}^{i_1}_{\mathcal{M}+L(i_1)} \\ \mathbf{I} \end{pmatrix} \widehat{K}^{b(i_1)}_{L}$    (14)
$\beta^{f}_{i_1} = \rho(2i_1, 2k_2(i_1)+2) + \mathbf{r}^{i_1}_{\mathcal{M}}\mathcal{J}\mathbf{b}^{i_1}_{\mathcal{M}}$    (15)
$k^{f}_{i_1} = -\beta^{f}_{i_1}/\alpha_{i_1}$    (16)
$\mathcal{S}_{i_1}\mathbf{a}^{i_1}_{\mathcal{M}+L(i_1)} = \begin{bmatrix} \mathcal{J}\mathbf{b}^{i_1}_{\mathcal{M}} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}^{i_1}_{\mathcal{M}} \\ 1 \end{bmatrix} k^{f}_{i_1}$    (17)
$\alpha_{-i_1} = \rho(0,0) + \widehat{\mathbf{r}}^{\ell}_{\mathcal{M}+L(i_1)}\mathbf{a}^{i_1}_{\mathcal{M}+L(i_1)}$    (18)
FOR $\ell = -k_1$ TO $k_1$, DO
    $\beta^{b\ell}_{-i_1} = \rho(i_1 + \ell, k_2(i_1) + k_2(\ell) + 2 + \delta(\ell - i_1)) + \widehat{\mathbf{r}}^{i_1}_{\mathcal{M}+L(i_1)}\mathbf{b}^{i_1}_{\mathcal{M}+L(i_1)}$    (19)
    $k^{b\ell}_{-i_1} = \beta^{b\ell}_{-i_1}/\alpha_{-i_1}$    (20)
    $\mathcal{T}_{R(-i)}\mathbf{b}^{\ell}_{\mathcal{M}^{i_1}} = \begin{bmatrix} 0 \\ \mathbf{b}^{\ell}_{\mathcal{M}+L(i_1)} \end{bmatrix} + \begin{bmatrix} 1 \\ \mathbf{a}^{i_1}_{\mathcal{M}+L(i_1)} \end{bmatrix} k^{b\ell}_{-i_1}$    (21)
ENDFOR $\ell$
```

Table 1 The Algorithm