Two-Dimensional Block Kalman Filtering for Image Restoration

MAHMOOD R. AZIMI-SADJADI, MEMBER, IEEE, AND PING WAH WONG, STUDENT MEMBER, IEEE

Abstract—This paper is concerned with developing an efficient two-dimensional (2-D) block Kalman filtering for digital image restoration. A new 2-D multiinput, multioutput (MIMO) state-space structure for modeling the image generation process is introduced. This structure is derived by arranging a vector autoregressive (AR) model with a causal quarter-plane region of support in block form. This model takes into account the correlations of the image data in successive neighboring blocks and, as a result, reduces the edge effects prominent in the available Kalman strip filtering techniques. The degradation model for an infinite extent linear space invariant (LSI) blur and white Gaussian (WG) noise is also modeled by an MIMO block state-space equation stemmed from a single-input single-output (SISO) 2-D state-space structure. The image generation model and the degradation model are combined to yield a composite block-state dynamic structure. The block Kalman filtering equations are obtained for this dynamic structure and then used to compute the suboptimal filter estimates of a noisy and blurred image.

I. Introduction

The model of the image formation system for a blurred and noisy image is

$$y(i, j) = \sum_k \sum_l h(i, j; k, l) f(k, l) + v(i, j) \qquad i = 0, 1, \cdots, N_1 \qquad j = 0, 1, \cdots, N_2 \tag{1}$$

where $h(i, j; k, l)$ is the spatial operator (point spread function or PSF) representing the blurring effect which may be caused by such phenomena as atmospheric turbulence, relative motion between the camera and the object being photographed, and defocusing; $f(i, j)$ represents the uncorrupted image of size $N_1 \times N_2$; $y(i, j)$ is the observed image; and $v(i, j)$ represents the noise in the digitized image which may be due to sensors, quantization effects, and transmission media, etc. The noise which is a stochastic phenomenon, in most practical situations, may be considered to be white Gaussian. For LSI blur, the PSF $h(i, j; k, l)$ reduces to $h(i - k, j - l)$ and, hence,

$$y(i, j) = \sum_k \sum_l h(i - k, j - l) \cdot f(k, l) + v(i, j). \tag{2}$$

The aim of image restoration is to estimate $f(i, j)$ from the observed image $y(i, j)$ given the PSF of the LSI blur and some statistical knowledge of the noise.

Recursive estimation techniques (Kalman filters) have been very useful in 1-D digital signal processing. The extension of Kalman filtering to the 2-D case, and its application to image restoration, has been receiving a great deal of attention in recent years. The initial attempt to model images by state-space techniques was reported by Nahi and Assefi [1]. Their modeling procedure exhibits several difficulties in representing 2-D random fields by 1-D models. Aboutalib and Silverman [2] have considered the case of images that are degraded by linear motion blur and additive noise. The original image is modeled as the output of a line scanner, and the blurring process is modeled by a 1-D linear dynamic model. However, the periodic nature of the scanning procedure gives rise to nonstationarity of the output image. Later, this approach was extended to the general motion blur [3]. The main problems in extending the standard 1-D recursive filtering techniques to the 2-D case are not only due to the difficulty in establishing a suitable 2-D recursive model, but also the high dimensionality of the resulting state vectors. Woods and Radewan [4] have proposed two 2-D Kalman processors known as the Kalman Strip Filter and Reduced Update Kalman Filter (RUKF), which use vector and scalar scanning schemes, respectively. The RUKF scheme was shown to offer significant reduction in the total computational load, and hence overcome the computational problems that have precluded the use of 2-D Kalman-like processors. The computational saving in this method was accomplished by limiting the updating process in the Kalman filter to a certain region in the vicinity of the point currently being processed. Later, Woods and Ingle [5] extended the RUKF to the case of degradation due to both blur and random noise. Over the past few years, several other authors have proposed different new 2-D Kalman filtering schemes for restoration of images degraded by both blur and noise [6]–[8]. The Kalman window approach proposed by Dikshit [6] is shown to be nonoptimal because of the erroneous assumption of the white noise model and inadequate description of the dynamic model representing the original image [9]. In [7] a set of low-order Kalman filters for nearly optimal recursive image restoration is derived which is suitable for parallel processing in the Fourier domain. The Kalman strip filtering proposed by Suresh and Shenoi [8] is particularly inter-

Manuscript received August 6, 1985; revised July 2, 1987.

M. R. Azimi-Sadjadi is with the Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523.

P. W. Wong is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

IEEE Log Number 8716986.

0096-3518/87/1200-1736$01.00 © 1987 IEEE

esting since it is concerned with modeling the blur by a 2-D state-space structure. However, there are some problems associated with this technique. This method fails to consider optimally the correlation between the pixels along the boundaries of successive neighboring strips. Moreover, the local state in this method propagates only in one dimension, i.e., during the intrastrip recursion and not during the interstrip recursion. This consequently results in a processor which is not a true 2-D recursive estimator.

In this paper, a generalized 2-D block Kalman filtering scheme for restoration of images degraded by LSI blur and additive WG noise is presented. In Section II, a vector AR model with a quarter-plane region of support is introduced which linearly relates a column of pixels in a strip to other columns in the same strip and also to those in the previous strip. A new 2-D block state-space structure is developed to model the image generation process [10]. Using this structure, each block of image is related to the three previously processed blocks and a block of random noise input. The effect of infinite extent LSI blur can be described by a discrete 2-D state-space equation [11]. In Section III, the SISO 2-D state-space model introduced by Roesser [12] is used to generate an MIMO block state-space structure for modeling the blur. It is shown that the horizontal and vertical states in this model propagate in different directions and, as a result, an efficient recursive implementation scheme can be devised. The two block state-space structures are combined in Section IV, and an observation noise is added to the model to form a composite block dynamic structure which accounts for both degradations. The block Kalman filter equations are developed for this new block dynamic model. Now, by moving the prediction window and applying the Kalman estimator at each stage, the block estimates can be computed. The comparison of the relative efficiency of the 2-D block Kalman filtering to those of the other 2-D Kalman strip processors indicates significant reduction in the total computational cost and storage requirements. The effectiveness of the proposed block Kalman filtering scheme is examined on a real world image using a VAX 11/780 computer.

II. Image Generation Model

The image generation model proposed by Suresh and Shenoi [8] is based upon partitioning an image into nonoverlapping strips and defining a vector AR process in each strip. This model incorporates both horizontal correlation (limited to the order of the AR model) and the vertical correlation (restricted to the width of each strip). Using this method, each strip is processed independently, and consequently distortion occurs at the edges of the strips because the correlations of pixels in neighboring strips are neglected. In this section, a new model is derived which takes into account the correlations of pixels within a strip as well as those between adjacent strips. In what follows, a 2-D dynamic state-space equation is developed for processing the images in blocks.

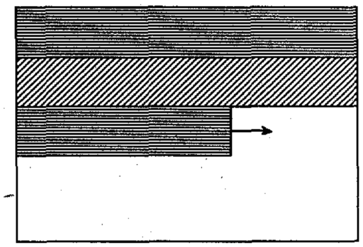


Fig. 1. The strip processing of images and vector scanning scheme.

Consider an image which is partitioned into strips as shown in Fig. 1. The image process starts from the upper-left-hand corner of the image and then proceeds horizontally along a strip. At the end of the strip, the process continues from the left-hand side of the next strip until the entire image has been processed. Note that the assignment of orientation and the direction of the vector scan is purely arbitrary and does not affect the generality of the model. Let us start with an $N \times N$ image which is sectioned into strips of width M. It is assumed that N is exactly divisible by M, i.e., $N = nM$ where n is an integer. The image is assumed to be represented by a vector Markovian random field with zero mean. We define an $M \times 1$ vector $Z(i, j)$ which denotes a column of pixels in the kth strip as follows:

$$Z(i,j) = \left[z(i,j)\, z(i+1,j) \cdots z(i+M-1,j)\right]^T \qquad (i,j) \in \mathcal{R} \tag{3}$$

$$\mathcal{R} = \left\{(i,j): i = (k-1)M;\ j = 0, 1, \cdots, N-1\right\}, \quad k = 1, 2, \cdots, n \tag{4}$$

where $z(i, j)$ is the pixel element at (i, j)th position and T denotes matrix transpose. Let the image process be modeled by a causal quarter-plane vector AR process of order M, which incorporates the correlation of pixels in a strip with those in the previous strip. Then, we have

$$\begin{aligned} Z(i,j) = {} & \Phi_1 Z(i,j-1) + \Phi_2 Z(i,j-2) \\ & + \cdots + \Phi_M Z(i,j-M) \\ & + \Theta_0 Z(i-M,j) + \Theta_1 Z(i-M,j-1) \\ & + \cdots + \Theta_M Z(i-M,j-M) + U(i,j) \end{aligned} \tag{5}$$

where Φ_i's and Θ_i's are the coefficient matrices of the AR process. $U(i, j)$ can be viewed as the input which drives the process or the error in generating $Z(i, j)$ from a linear combination of the past vectors $Z(i - k, j - l)$. This $M \times 1$ vector is defined by

$$U(i,j) = \left[u(i,j)\, u(i+1,j) \cdots u(i+M-1,j)\right]^T \qquad (i,j) \in \mathcal{R}. \tag{6}$$

It can be shown (Appendix A) that $U(i, j)$ is an uncorrelated random vector which satisfies

$$\mathcal{E}[U(i,j)\cdot U^T(i-p,j-q)] = Q_U\delta(p,q)$$
$$(i,j)\in\mathcal{R} \qquad (p,q)\in\mathcal{S} \tag{7}$$

where the set $\mathcal{S}$ is defined in (A8); Q_U is the correlation matrix; $\delta(p,q)$ is the Kronecker delta; and $\mathcal{E}$ is the expectation. Note that (7) does not imply that the components of the vector $U(i,j)$ are mutually uncorrelated. The image process (5) can be assumed to be "column wide sense stationary." Now, define the following $2M \times 1$ vectors (Fig. 2) as:

$$\begin{aligned}\zeta(i,j) &= [Z^T(i-M,j)\, Z^T(i,j)]^T\\ &= [z(i-M,j)\, z(i-M+1,j)\\ &\quad \cdots z(i+M-1,j)]^T\\ &\qquad (i,j)\in\mathcal{R}.\end{aligned} \tag{8}$$

Then, (5) can be written as

$$\begin{aligned}\zeta(i,j) &= \Psi_0\zeta(i,j) + \Psi_1\zeta(i,j-1) + \Psi_2\zeta(i,j-2)\\ &\quad + \cdots + \Psi_M\zeta(i,j-M) + \Omega U(i,j)\end{aligned} \tag{9}$$

where

$$\Psi_0 = \begin{bmatrix} I_M & 0\\ \Theta_0 & 0\end{bmatrix}_{2M\times 2M} \tag{10a}$$

$$\Psi_i = \begin{bmatrix} 0 & 0\\ \Theta_i & \Phi_i\end{bmatrix}_{2M\times 2M} \qquad i = 1, 2, \cdots, M \tag{10b}$$

$$\Omega = [0 \;\; I_M]^T \qquad \text{size } 2M\times M \tag{11}$$

and I_M is the identity matrix of order M.

To apply the Kalman filtering, the image process in (9) should be arranged in the state-space form with an appropriate state vector. In order to construct such a model, the coefficient matrices of the AR process must be computed. Let us define the autocorrelation matrix of $Z(i,j)$ as

$$\rho_{p,q} = \mathcal{E}[Z(i-p,j-q)\cdot Z^T(i,j)]$$
$$(i,j)\in\mathcal{R} \qquad (p,q)\in\mathcal{S}. \tag{12}$$

Now, taking the transpose of both sides of (5), premultiplying by the vector $Z(i-k,j-l)$, and then taking the expectation yields

$$\begin{aligned}\rho_{k,l} &= \rho_{k,l-1}\Phi_1^T + \rho_{k,l-2}\Phi_2^T\\ &\quad + \cdots + \rho_{k,l-M}\Phi_M^T\\ &\quad + \rho_{k-M,l}\Theta_0^T + \rho_{k-M,l-1}\Theta_1^T\\ &\quad + \cdots + \rho_{k-M,l-M}\Theta_M^T\\ &\quad + \mathcal{E}[Z(i-k,j-l)\,U^T(i,j)].\end{aligned} \tag{13}$$

Putting the values of

$$(k,l) = \{(0,1), (0,2), \cdots, (0,M)\\ (M,0), (M,1), \cdots, (M,M)\}$$

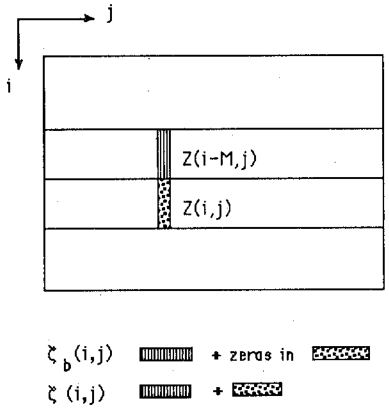


Fig. 2. The structur of the state vectors $\zeta(i,j)$, $\zeta_b(i,j)$, and $Z(i,j)$ in two adjacent strips.

into (13) and making use of the properties (see Appendix A)

$$\mathcal{E}[U(i,j)\,Z^T(i-k,j-l)] = 0 \qquad (k,l)\neq(0,0) \tag{14}$$

and

$$\rho_{k,l} = \rho^T_{-k,-l}, \tag{15}$$

the following system of equations can be obtained which may be used to evaluate Φ_i's and Θ_i's given the autocorrelation matrices $\rho_{k,l}$'s:

$$\begin{bmatrix} \boldsymbol{R}_0' & \boldsymbol{R}_1'^T & \boldsymbol{R}_2'^T & \cdots & \boldsymbol{R}_M'^T\\ \boldsymbol{R}_1' & \boldsymbol{R}_0 & \boldsymbol{R}_1^T & & \boldsymbol{R}_{M-1}^T\\ \boldsymbol{R}_2' & \boldsymbol{R}_1 & \boldsymbol{R}_0 & & \boldsymbol{R}_{M-2}^T\\ \vdots & & & & \vdots\\ & & & & \boldsymbol{R}_1^T\\ \boldsymbol{R}_M' & \boldsymbol{R}_{M-1} & \boldsymbol{R}_{M-2} & \cdots & \boldsymbol{R}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_0'\\ \boldsymbol{S}_1\\ \boldsymbol{S}_2\\ \vdots\\ \boldsymbol{S}_{M-1}\\ \boldsymbol{S}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_0'\\ \boldsymbol{T}_1\\ \boldsymbol{T}_2\\ \vdots\\ \boldsymbol{T}_{M-1}\\ \boldsymbol{T}_M \end{bmatrix} \tag{16}$$

where

$$\boldsymbol{R}_k = \begin{bmatrix} \rho_{0,k} & \rho_{M,k}\\ \rho^T_{M,-k} & \rho_{0,k}\end{bmatrix} \qquad k = 0, 1, \cdots, M-1 \tag{17a}$$

$$\boldsymbol{R}_k' = [\rho^T_{0,k} \;\; \rho_{M,-k}]^T \qquad k = 1, 2, \ldots, M \tag{17b}$$

$$\boldsymbol{R}_0' = \rho_{0,0} \tag{17c}$$

$$\boldsymbol{S}_k = [\Theta_k \;\; \Phi_k]^T \qquad k = 1, 2, \cdots, M \tag{18a}$$

$$\boldsymbol{S}_0' = \Theta_0^T \tag{18b}$$

$$\boldsymbol{T}_k = [\rho^T_{M,k} \;\; \rho^T_{0,k}]^T \qquad k = 1, 2, \cdots, M \tag{19a}$$

$$\boldsymbol{T}_0' = \rho_{M,0} \tag{19b}$$

Now, if we further assume that the image process satisfies an ergodic theorem, good estimates for the autocorrelation matrices $\rho_{p,q}$ can be obtained by

$$\rho_{p,q} = \frac{1}{\| \mathcal{R} \|} \sum\sum_{(i,j)\in\mathcal{R}} Z(i-p, j-q)\, Z^T(i,j) \qquad (p,q) \in \mathcal{S} \tag{20}$$

where $\| \mathcal{R} \|$ is the size of $\mathcal{R}$.

In order to put (9) into the state-space form, let us define a state vector $\chi(i,j)$ as

$$\chi(i,j) = \left[\zeta^T(i,j-M) \cdots \zeta^T(i,j-1)\, \zeta^T(i,j)\right]^T \qquad \text{size } 2M(M+1) \times 1. \tag{21}$$

Let $\chi_b(i,j)$ be a vector such that the top M elements (each element is a vector of size $2M \times 1$) are identical to those in $\chi(i,j)$, and the last element is vector $\zeta_b(i,j)$ (Figs. 2 and 3). Thus,

$$\chi_b(i,j) = \left[\zeta^T(i,j-M) \cdots \zeta^T(i,j-1)\, \zeta_b^T(i,j)\right]^T \qquad \text{size } 2M(M+1) \times 1 \tag{22}$$

where

$$\zeta_b(i,j) = \left[Z^T(i-M,j)\; 0\right]^T \qquad (i,j) \in \mathcal{R}. \tag{23}$$

At the (i,j)th iteration, the vector $\chi(i,j)$ is evaluated from the vector $\chi_b(i,j)$ using the following state-space equations:

$$\chi_b(i,j) = \tilde{G}\chi(i,j-1) + \tilde{H}\chi(i-M,j) \tag{24a}$$

$$\chi(i,j) = \tilde{A}_1\chi_b(i,j) + \tilde{B}_1 U(i,j) \tag{24b}$$

$$\gamma(i,j) = \tilde{C}_1\chi(i,j) \tag{24c}$$

where

$$\tilde{G} = \begin{bmatrix} 0 & I_{2M} & & & \bigcirc \\ & & I_{2M} & & \\ & & & \ddots & \\ & \bigcirc & & & I_{2M} \\ & & & & 0 \end{bmatrix}_{2M(M+1)\times 2M(M+1)} \tag{25a}$$

$$\tilde{H} = \begin{bmatrix} 0 & \cdots & \cdots & & 0 \\ \vdots & & \bigcirc & & \vdots \\ 0 & \cdots & \cdots & & 0 \\ 0 & \cdots & \cdots & 0 & I_M \\ 0 & \cdots & \cdots & & 0 \end{bmatrix}_{2M(M+1)\times 2M(M+1)} \tag{25b}$$

$$\tilde{A}_1 = \begin{bmatrix} I_{2M} & & & & \bigcirc \\ & I_{2M} & & & \\ & & \ddots & & \\ \bigcirc & & & I_{2M} & 0 \\ \Psi_M & \cdots & \cdots & \Psi_1 & \Psi_0 \end{bmatrix}_{2M(M+1)\times 2M(M+1)} \tag{25c}$$

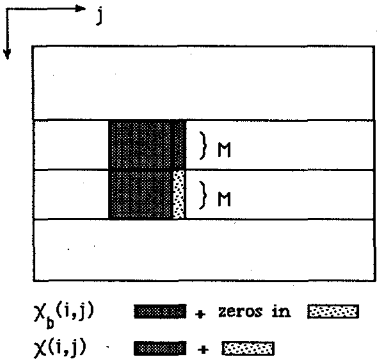


Fig. 3. The structure of the state vectors $\chi(i,j)$ and $\chi_b(i,j)$ in two adjacent strips.

$$\tilde{B}_1 = \left[0\; 0 \cdots 0\; I_M\right]^T \qquad \text{size } 2M(M+1) \times M \tag{25d}$$

$$\tilde{C}_1 = \left[0\; 0 \cdots 0\; I_M\right] \qquad \text{size } M \times 2M(M+1) \tag{25e}$$

$$\gamma(i,j) = \left[f(i,j) f(i+1,j) \cdots f(i+M-1,j)\right]^T \qquad \text{size } M \times 1. \tag{26}$$

Equation (24a) generates the vector $\chi_b(i,j)$ from the state vectors $\chi(i,j-1)$ and $\chi(i-M,j)$ and prepares it for processing. It essentially performs a shift operation on the vector $\chi(i,j-1)$ and appends a portion of $\chi(i-M,j)$ to the appropriate location in $\chi_b(i,j)$. As a result, the operations in this equation can be done by data translations and appropriate addressing which require no computations. Equation (24b) is obtained by rearranging the AR process of (5). One column of image pixels $\gamma(i,j)$ is evaluated from the state vector by (24c). The output elements of the system, $f(i,j)$, have the same statistics as the process of which our ideal image forms a sample. The state-space model (24) takes an uncorrelated random vector as its input and generates one column of pixels of size M in one iteration. This model can be extended such that the image is processed in blocks of size $M \times M$ (Fig. 4). Let us define the following block states, input and output vectors as:

$$x_1(i,j) = \left[\zeta^T(i,j-M)\, \zeta^T(i,j-M+1) \cdots \zeta^T(i,j-1)\, \zeta^T(i,j)\, \zeta^T(i,j+1) \cdots \zeta^T(i,j+M-1)\right]^T \qquad \text{size } 4M^2 \times 1 \tag{27}$$

$$x_{1b}(i,j) = \left[\zeta^T(i,j-M)\, \zeta^T(i,j-M+1) \cdots \zeta^T(i,j-1)\, \zeta_b^T(i,j)\, \zeta_b^T(i,j+1) \cdots \zeta_b^T(i,j+M-1)\right]^T \qquad \text{size } 4M^2 \times 1 \tag{28}$$

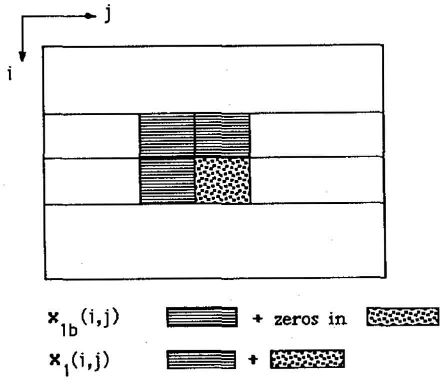


Fig. 4. The structure of the block state vectors $x_1(i, j)$ and $x_{1b}(i, j)$ in the image generation model.

$$f(i, j) = \left[\gamma^T(i, j)\, \gamma^T(i, j + 1) \cdots \gamma^T(i, j + M - 1)\right]^T \quad \text{size } M^2 \times 1 \tag{29}$$

$$u(i, j) = \left[U^T(i, j)\, U^T(i, j + 1) \cdots U^T(i, j + M - 1)\right]^T \quad \text{size } M^2 \times 1 \quad (i, j) \in \mathcal{R}; \quad j \in \mathcal{R}^- \tag{30}$$

where

$$\mathcal{R}^- = \left\{j: j = (l - 1)M\right\}, \qquad l = 1, 2, \cdots, n \tag{31a}$$

and

$$\mathcal{E}\left[u(i - k, j - l) \cdot u^T(i, j)\right] = Q_u \delta(k, l) \tag{32a}$$

where Q_u is a correlation matrix of size $M^2 \times M^2$ given by

$$Q_u = I_M \otimes Q_U \tag{32b}$$

where $\otimes$ denotes Kronecker product operation.

The block state-space model becomes

$$x_{1b}(i, j) = Gx_1(i, j - M) + Hx_1(i - M, j) \tag{33a}$$

$$x_1(i, j) = A_1 x_{1b}(i, j) + B_1 u(i, j) \tag{33b}$$

$$f(i, j) = C_1 x_1(i, j) \qquad (i, j) \in \mathcal{R}; j \in \mathcal{R}^- \tag{33c}$$

where

$$G = \begin{bmatrix} 0 & I_{2M^2} \\ 0 & 0 \end{bmatrix} \tag{34a}$$

$$H = \begin{bmatrix} \bigcirc & & & & \bigcirc \\ & 0 & I_M & & \\ & 0 & 0 & & \\ & & & \ddots & & \\ & & & & 0 & I_M \\ \bigcirc & & & & 0 & 0 \end{bmatrix}_{4M^2 \times 4M^2} \tag{34b}$$

$$A_1 = \begin{bmatrix} I_{2M} & & & & & \bigcirc \\ \bigcirc & \ddots & & & & \\ & & I_{2M} & & & \\ \alpha_{11} & \alpha_{12} \cdots & \alpha_{1M} & \lambda_1 & & \\ \alpha_{21} & \cdots & \alpha_{2M} & \lambda_2 & \lambda_1 & \\ \vdots & & \vdots & \vdots & & \ddots \\ \alpha_{M1} & \cdots & \alpha_{MM} & \lambda_M & \cdots & \lambda_1 \end{bmatrix}_{4M^2 \times 4M^2} \tag{34c}$$

$$B_1 = \begin{bmatrix} \beta_1 & & \bigcirc \\ \beta_2 & \beta_1 & \\ \vdots & & \ddots \\ \beta_M & \cdots & \beta_1 \end{bmatrix}_{4M^2 \times 4M^2} \tag{34d}$$

$$C_1 = \begin{bmatrix} 0 & I_M & & & & & & \bigcirc \\ & & 0 & I_M & & & & \\ & & & & 0 & I_M & & \\ \bigcirc & & & & & & \ddots & \\ & & & & & & 0 & I_M \end{bmatrix}_{M^2 \times 4M^2} \tag{34e}$$

where

$$\alpha_{1j} = \Psi_{M+1-j} \qquad j = 1, 2, \cdots, M \tag{35a}$$

Ψ_i's are defined in (10) and $\Psi_k = 0$ for $k < 0$ or $k > M$.

$$\alpha_{ij} = \Psi_{M+i-j} + \Psi_1 \alpha_{i-1,j} + \Psi_2 \alpha_{i-2,j} + \cdots + \Psi_{i-1} \alpha_{1j} \qquad j = 1, 2, \cdots, M \quad i = 2, 3, \cdots, M \tag{35b}$$

$$\lambda_1 = \Psi_0 \tag{35c}$$

$$\lambda_i = \Psi_1 \lambda_{i-1} + \Psi_2 \lambda_{i-1} + \cdots + \Psi_{i-1} \lambda_1 \qquad i = 2, 3, \cdots, M \tag{35d}$$

$$\beta_1 = \left[0 \quad I_M\right]^T \tag{35e}$$

$$\beta_i = \Psi_1 \beta_{i-1} + \Psi_2 \beta_{i-2} + \cdots + \Psi_{i-1} \beta_1 \qquad i = 2, 3, \cdots, M. \tag{35f}$$

It can be shown (Appendix B) that α_{ij}'s, λ_i's, and β_i's for $i = 2, 3, \cdots, M$ can be evaluated recursively by

$$\alpha_{ij} = \alpha_{i-1,j-1} + \alpha_{i-1,M} \alpha_{1j} \qquad j = 1, 2, \cdots, M \quad i = 2, 3, \cdots, M \tag{36a}$$

$$\lambda_i = \alpha_{i-1,M} \lambda_1 \qquad i = 2, 3, \cdots, M \tag{36b}$$

$$\beta_i = \alpha_{i-1,M} \beta_1 \qquad i = 2, 3, \cdots, M. \tag{36c}$$

In the block dynamic model (33), (33a) represents a shift operation on the vector $x_1(i, j - M)$ plus the extraction of elements from the previous state vector $x_1(i - M, j)$. These operations can be done with indirect addressing and essentially no computation. After the completion of this step, vector $x_{1b}(i, j)$, which consists of three previously processed blocks and zero entries in the present block, is evaluated. Then (33b) is performed to advance the state vector $x_1(i, j)$ and estimate a block of image data. Therefore, (33b) and (33c) generate the present block of the estimated image from three past processed blocks and a block of uncorrelated random vectors. The state vector $x_1(i, j)$ then propagates to the next iteration for subsequent processing. Moreover, part of the elements in this vector are saved which will be recalled when the processing window is moved to the adjacent block in the following strip. This procedure is accomplished simply by writing the new required elements over the old elements in the storage. Thus, while the local state consists of only two vectors, $x_1(i, j - M)$ and $x_1(i - M, j)$, the global state has to include all the vectors $x_1(i - M, j)$, $\forall j \in \mathcal{R}^-$ in the previous strip when the processing is moved to a new strip. A pointer is used to keep track of the head of the global state vector. To implement this process more efficiently, only those elements in $x_1(i - M, j)$ which will be recalled by the recursive process need to be stored in the global state. This is called reduced updating of the global state [4].

III. LSI Blur Model

The state-space model derived by Suresh and Shenoi [8] for an infinite extent LSI blur takes a column of the input in a strip and generates a column of output in conformity with the strip filtering model. This structure was originated from the Roesser [12] 2-D SISO model. In this section, the problem of modeling the blur by a 2-D block state-space structure is considered. A new version of the block state-space structure derived in [13] is developed. Because the block state vectors in this MIMO structure propagate in different directions, an efficient recursive implementation scheme is suggested.

Let $f(i, j)$ be a single pixel element at the (i, j)th position generated by the image process in the previous section, $y(i, j)$ be the blurred pixel at the same location. Define an $n_1 \times 1$ vertical state vector $R(i, j)$ and an $n_2 \times 1$ horizontal state vector $S(i, j)$. Then, Roesser's SISO state-space model is given by

$$\begin{bmatrix} R(i+1, j) \\ S(i, j+1) \end{bmatrix} = \begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_2 \\ \mathcal{E}_3 & \mathcal{E}_4 \end{bmatrix} \begin{bmatrix} R(i, j) \\ S(i, j) \end{bmatrix} + \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \end{bmatrix} f(i, j) \tag{37a}$$

$$y(i, j) = [\mathcal{K}_1 \;\; \mathcal{K}_2] \begin{bmatrix} R(i, j) \\ S(i, j) \end{bmatrix} + \mathcal{L} f(i, j) \tag{37b}$$

where $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, $\mathcal{E}_4$, $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{L}$ are matrices of appropriate dimensions. Note that the global state in this model consists of initial conditions $R(0, j)$, $\forall j \geq 0$ and $S(i, 0)$, $\forall i \geq 0$, which refer to the state along the boundaries. Thus, it provides the information about the entire past history of the image. The local state, on the other hand, consists of vectors $R(i, j)$ and $S(i, j)$ which propagate during the state recursion. Let us define the following "block state" vectors:

$$R_r(i, j) = [R^T(i, j)\, R^T(i, j+1) \cdots R^T(i, j+M-1)]^T \quad \text{size } Mn_1 \times 1 \tag{38}$$

$$S_c(i, j) = [S^T(i, j)\, S^T(i+1, j) \cdots S^T(i+M-1, j)]^T \quad \text{size } Mn_2 \times 1 \tag{39}$$

where $R_r(i, j)$ and $S_c(i, j)$ denote the states associated with the boundary elements of the (i, j)th block along the horizontal and vertical directions, respectively (Fig. 5). Using these two vectors, the block state-space model can be completely characterized. Recursive application of (37a) and (37b) for each element of the block state vectors yields the following 2-D block state-space model [11], [13]:

$$\begin{bmatrix} R_r(i+M, j) \\ S_c(i, j+M) \end{bmatrix} = \begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix} \begin{bmatrix} R_r(i, j) \\ S_c(i, j) \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} f(i, j) \tag{40a}$$

$$y(i, j) = [K_1 \;\; K_2] \begin{bmatrix} R_r(i, j) \\ S_c(i, j) \end{bmatrix} + L f(i, j)$$

$$(i, j) \in \mathcal{R}; \quad j \in \mathcal{R}^- \tag{40b}$$

where $f(i, j)$ is defined in (29), and

$$y(i, j) = [Y^T(i, j)\, Y^T(i, j+1) \cdots Y^T(i, j+M-1)]^T \quad (i, j) \in \mathcal{R}; \quad j \in \mathcal{R}^- \tag{41}$$

denotes a block of blurred pixels (size $M^2 \times 1$)

$$E_1 = \begin{bmatrix} \tilde{E}_1 & & & \\ \tilde{E}_2\tilde{E}_3 & \ddots & & \bigcirc \\ \tilde{E}_2\tilde{E}_4\tilde{E}_3 & & & \\ \vdots & & \ddots & \\ \tilde{E}_2\tilde{E}_4^{M-2}\tilde{E}_3 & \cdots & \tilde{E}_2\tilde{E}_3 & \tilde{E}_1 \end{bmatrix}_{Mn_1 \times Mn_1} \tag{42a}$$

$$E_2 = \begin{bmatrix} \tilde{E}_2 \\ \tilde{E}_2\tilde{E}_4 \\ \vdots \\ \tilde{E}_2\tilde{E}_4^{M-1} \end{bmatrix}_{Mn_1 \times Mn_2} \tag{42b}$$

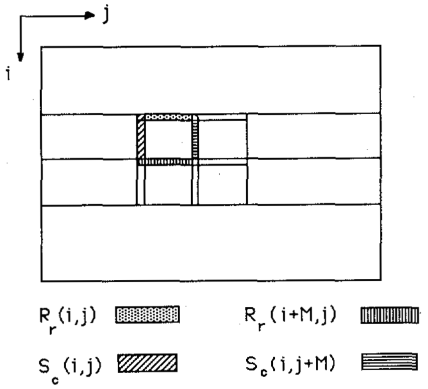


Fig. 5. The structure of block state vectors $R_r(i,j)$, $S_c(i,j)$, $R_r(i+M, j)$, and $S_c(i,j+M)$ in the blur model.

$$E_3 = [\tilde{E}_4^{M-1}\tilde{E}_3 \;\; \tilde{E}_4^{M-2}\tilde{E}_3 \cdots \tilde{E}_3] \qquad \text{size } Mn_2 \times Mn_1 \tag{42c}$$

$$E_4 = \tilde{E}_4^M \qquad \text{size } Mn_2 \times Mn_2 \tag{42d}$$

$$F_1 = \begin{bmatrix} \tilde{F}_1 & & & \bigcirc \\ \tilde{E}_2\tilde{F}_2 & \ddots & & \\ \tilde{E}_2\tilde{E}_4\tilde{F}_2 & & \ddots & \\ \vdots & & & \ddots \\ \tilde{E}_2\tilde{E}_4^{M-2}\tilde{F}_2 & \cdots & \tilde{E}_2\tilde{F}_2 & \tilde{F}_1 \end{bmatrix}_{Mn_1 \times M^2} \tag{43a}$$

$$F_2 = [\tilde{E}_4^{M-1}\tilde{F}_2 \;\; \tilde{E}_4^{M-2}\tilde{F}_2 \cdots \tilde{F}_2] \qquad \text{size } Mn_2 \times M^2 \tag{43b}$$

$$K_1 = \begin{bmatrix} \tilde{K}_1 & & & \bigcirc \\ \tilde{K}_2\tilde{E}_3 & \ddots & & \\ \tilde{K}_2\tilde{E}_4\tilde{E}_3 & & \ddots & \\ \vdots & & & \ddots \\ \tilde{K}_2\tilde{E}_4^{M-2}\tilde{E}_3 & \cdots & \tilde{K}_2\tilde{E}_3 & \tilde{K}_1 \end{bmatrix}_{M^2 \times Mn_1} \tag{44a}$$

$$K_2 = \begin{bmatrix} \tilde{K}_2 \\ \tilde{K}_2\tilde{E}_4 \\ \vdots \\ \tilde{K}_2\tilde{E}_4^{M-1} \end{bmatrix}_{M^2 \times Mn_2} \tag{44b}$$

$$L = \begin{bmatrix} \tilde{L} & & & \bigcirc \\ \tilde{K}_2\tilde{F}_2 & \ddots & & \\ \tilde{K}_2\tilde{E}_4\tilde{F}_2 & & \ddots & \\ \vdots & & & \ddots \\ \tilde{K}_2\tilde{E}_4^{M-2}\tilde{F}_2 & \cdots & \tilde{K}_2\tilde{F}_2 & \tilde{L} \end{bmatrix}_{M^2 \times M^2} \tag{45}$$

where

$$Y(i,j) = [y(i,j)\, y(i+1,j) \cdots y(i+M-1,j)]^T \qquad (i,j) \in \mathcal{R} \tag{46}$$

$$\tilde{E}_1 = \mathcal{E}_1^M \qquad \text{size } n_1 \times n_1 \tag{47a}$$

$$\tilde{E}_2 = [\mathcal{E}_1^{M-1}\mathcal{E}_2 \;\; \mathcal{E}_1^{M-2}\mathcal{E}_2 \cdots \mathcal{E}_2] \qquad \text{size } n_1 \times Mn_2 \tag{47b}$$

$$\tilde{E}_3 = \begin{bmatrix} \mathcal{E}_3 \\ \mathcal{E}_3\mathcal{E}_1 \\ \vdots \\ \mathcal{E}_3\mathcal{E}_1^{M-1} \end{bmatrix}_{Mn_2 \times n_1} \tag{47c}$$

$$\tilde{E}_4 = \begin{bmatrix} \mathcal{E}_4 & & & \bigcirc \\ \mathcal{E}_3\mathcal{E}_2 & \ddots & & \\ \mathcal{E}_3\mathcal{E}_1\mathcal{E}_2 & & \ddots & \\ \vdots & & & \ddots \\ \mathcal{E}_3\mathcal{E}_1^{M-2}\mathcal{E}_2 & \cdots & \mathcal{E}_3\mathcal{E}_2 & \mathcal{E}_4 \end{bmatrix}_{Mn_2 \times Mn_2} \tag{47d}$$

$$\tilde{F}_1 = [\mathcal{E}_1^{M-1}\mathcal{F}_1 \;\; \mathcal{E}_1^{M-2}\mathcal{F}_1 \cdots \mathcal{F}_1] \qquad \text{size } n_1 \times M \tag{48a}$$

$$\tilde{F}_2 = \begin{bmatrix} \mathcal{F}_2 & & & \bigcirc \\ \mathcal{E}_3\mathcal{F}_1 & \ddots & & \\ \mathcal{E}_3\mathcal{E}_1\mathcal{F}_1 & & \ddots & \\ \vdots & & & \ddots \\ \mathcal{E}_3\mathcal{E}_1^{M-2}\mathcal{F}_1 & \cdots & \mathcal{E}_3\mathcal{F}_1 & \mathcal{F}_2 \end{bmatrix}_{Mn_2 \times M} \tag{48b}$$

$$\tilde{K}_1 = \begin{bmatrix} \mathcal{K}_1 \\ \mathcal{K}_1\mathcal{E}_1 \\ \vdots \\ \mathcal{K}_1\mathcal{E}_1^{M-1} \end{bmatrix}_{M \times n_1} \tag{49a}$$

$$\tilde{K}_2 = \begin{bmatrix} \mathcal{K}_2 & & & \bigcirc \\ \mathcal{K}_1\mathcal{E}_2 & \ddots & & \\ \mathcal{K}_1\mathcal{E}_1\mathcal{E}_2 & & \ddots & \\ \vdots & & & \ddots \\ \mathcal{K}_1\mathcal{E}_1^{M-2}\mathcal{E}_2 & \cdots & \mathcal{K}_1\mathcal{E}_2 & \mathcal{K}_2 \end{bmatrix}_{M \times Mn_2} \tag{49b}$$

$$\tilde{L} = \begin{bmatrix} \mathcal{L} & & & \bigcirc \\ \mathcal{K}_1\mathcal{F}_1 & \ddots & & \\ \mathcal{K}_1\mathcal{E}_1\mathcal{F}_1 & & \ddots & \\ \vdots & & & \ddots \\ \mathcal{K}_1\mathcal{E}_1^{M-2}\mathcal{F}_1 & \cdots & \mathcal{K}_1\mathcal{F}_1 & \mathcal{L} \end{bmatrix}_{M \times M}. \tag{50}$$

It is observed that the 2-D MIMO block state-space model in (40) has a form similar to its SISO 2-D state-space counterpart. The block state vectors $R_r(i, j)$ and $S_c(i, j)$ propagate in vertical and horizontal directions, respectively, to generate the vectors $R_r(i + M, j)$ and $S_c(i, j + M)$ which correspond to the states associated with the horizontal and vertical boundary elements of $(i + M, j)$th and $(i, j + M)$th blocks, respectively (Fig. 5). These latter vectors are computed recursively using (40a, b), while the output block at the (i, j)th position is being evaluated. The block state vector $R_r(i + M, j)$ will not be needed until the processing has been moved to the next strip at the $(i + M, j)$th position. The block state vector $R_r(i, j + M)$, which is required for processing the block at the $(i, j + M)$th position and is computed in the previous strip (the result of processing the $(i - M, j + M)$th block), will be recalled from memory. Thus, for the recursive process to work, all the state vectors, $R_r(i + M, j)$, $\forall j \in \mathcal{R}^-$, that are evaluated when processing the kth strip must be stored as they will be required for processing the next strip. This indicates that the global state consists of the vectors $R_r(i, j)$, $\forall j \in \mathcal{R}^-$ while only the local state is involved in each iteration. Note that the vectors $R_r(0, j)$ and $S_c(i, 0)$ are available externally for the first stage of the recursion process.

IV. Block State-Space Dynamic Model and Kalman Filter Equations

The image generation model in (33) and the blur model in (40) can be combined, and the observation noise may be added to form the following composite block dynamic model which accounts for both of the above processes.

$$x_{1b}(i, j) = Gx_1(i, j - M) + Hx_1(i - M, j) \tag{51a}$$

$$\begin{bmatrix} x_1(i, j) \\ R_r(i + M, j) \\ S_c(i, j + M) \end{bmatrix} = A \begin{bmatrix} x_{1b}(i, j) \\ R_r(i, j) \\ S_c(i, j) \end{bmatrix} + Bu(i, j) \tag{51b}$$

$$y(i, j) = C \begin{bmatrix} x_1(i, j) \\ R_r(i, j) \\ S_c(i, j) \end{bmatrix} + v(i, j)$$

$$(i, j) \in \mathcal{R}; \quad j \in \mathcal{R}^- \tag{51c}$$

where

$$A = \begin{bmatrix} A_1 & 0 & 0 \\ F_1C_1A_1 & E_1 & E_2 \\ F_2C_1A_1 & E_3 & E_4 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ F_1C_1B_1 \\ F_2C_1B_1 \end{bmatrix},$$

$$C = [LC_1 \;\; K_1 \;\; K_2] \tag{51d}$$

and $v(i, j)$ is the observation noise vector of size $M^2 \times 1$ with

$$\mathcal{E}\left[v(i - k, j - l) \cdot v^T(i, j)\right] = Q_v\delta(k, l) \tag{52}$$

where Q_v is the correlation matrix of size $M^2 \times M^2$ defined in a manner similar to Q_u in (32b). Note that the subscript (i, j) in each vector represents the location of the first pixel at the beginning of each block. Now if this is mapped in accordance with the nature of the vector scan such that the kth block corresponds to the block with the first pixel at the (i, j)th position, then we have

$$w(i, j) = w(k) \tag{53a}$$

$$w(i, j + M) = w(k + 1) \tag{53b}$$

$$w(i + M, j) = w(k + n) \tag{53c}$$

where $k = (in + j)/M + 1$, $n = N/M$, and w can be any block vector in the model. Using this mapping, (51) can be rewritten as

$$x_{1b}(k) = Gx_1(k - 1) + Hx_1(k - n) \tag{54a}$$

$$\begin{bmatrix} x_1(k) \\ R_r(k + n) \\ S_c(k + 1) \end{bmatrix} = A \begin{bmatrix} x_{1b}(k) \\ R_r(k) \\ S_c(k) \end{bmatrix} + Bu(k) \tag{54b}$$

$$y(k) = C \begin{bmatrix} x_1(k) \\ R_r(k) \\ S_c(k) \end{bmatrix} + v(k). \tag{54c}$$

The model in (54) is a linear dynamic model, with the state vectors $x_1(k)$, $R_r(k)$, and $S_c(k)$, that propagates differently. The state vector $x_{1b}(k)$ is generated from the state vectors $x_1(k - 1)$ and $x_1(k - n)$ using only shift operation and data translation. Then, the state vector $x_1(k)$ is evaluated from $x_{1b}(k)$ using the least square prediction equation. The state vector $R_r(k)$ will be computed as described in Section III. The state vector $S_c(k)$ is evaluated along the direction of the vector scan, i.e., it propagates directly from the kth iteration to the $(k + 1)$th iteration except at the boundary of the image where the boundary condition is applied. The Kalman filtering equations (with an extra step) for the block dynamic model (54) are

$$\hat{x}_{1b}(k) = G\hat{x}_1(k - 1) + H\hat{x}_1(k - n) \tag{55a}$$

$$\begin{bmatrix} \hat{x}_1(k) \\ \hat{R}_r(k + n) \\ \hat{S}_c(k + 1) \end{bmatrix} = A \begin{bmatrix} \hat{x}_{1b}(k) \\ \hat{R}_r(k) \\ \hat{S}_c(k) \end{bmatrix} \tag{55b}$$

$$P_b(k) = AP_a(k - 1)A^T + BQ_uB^T \tag{55c}$$

$$K(k) = P_b(k)C'^T\left\{C'P_b(k)C'^T + Q_v\right\}^{-1} \tag{55d}$$

$$\begin{bmatrix} \hat{\hat{\boldsymbol{x}}}_1(k) \\ \hat{\hat{R}}_r(k+n) \\ \hat{\hat{S}}_c(k+1) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_1(k) \\ \hat{R}_r(k+n) \\ \hat{S}_c(k+1) \end{bmatrix} + \boldsymbol{K}(k) \left\{ \boldsymbol{y}(k) - \boldsymbol{C} \begin{bmatrix} \hat{\boldsymbol{x}}_1(k) \\ \hat{\hat{R}}_r(k) \\ \hat{\hat{S}}_c(k) \end{bmatrix} \right\} \tag{55e}$$

$$\boldsymbol{P}_a(k) = \left\{ I - \boldsymbol{K}(k)\, \boldsymbol{C}' \right\} \boldsymbol{P}_b(k) \tag{55f}$$

$$\hat{\hat{f}}(k) = [C_1 \quad 0 \quad 0] \begin{bmatrix} \hat{\hat{\boldsymbol{x}}}_1(k) \\ \hat{\hat{R}}_r(k+n) \\ \hat{\hat{S}}_c(k+1) \end{bmatrix} \tag{55g}$$

where ‘‘$\hat{\ }$’’ and ‘‘$\hat{\hat{\ }}$’’ at the space vectors denote the estimates before and after updating, respectively; and $\hat{\hat{f}}(k)$ is the suboptimal estimate of the original image; and matrix C' is defined as

$$\boldsymbol{C}' = \boldsymbol{C}\boldsymbol{A}'^{-1} \tag{56a}$$

where the nonsingular matrix A' is given by

$$A' = \begin{bmatrix} I & 0 & 0 \\ F_1C_1 & E_1 & E_2 \\ F_2C_1 & E_3 & E_4 \end{bmatrix}. \tag{56b}$$

The error covariance matrices of the state vectors before and after updating at the kth iteration, $\boldsymbol{P}_b(k)$ and $\boldsymbol{P}_a(k)$, are, respectively,

$$\boldsymbol{P}_b(k) \triangleq \mathcal{E}\left[(\boldsymbol{X}(k) - \hat{\boldsymbol{X}}(k))\, (\boldsymbol{X}(k) - \hat{\boldsymbol{X}}(k))^T \right] \tag{57a}$$

$$\boldsymbol{P}_a(k) \triangleq \mathcal{E}\left[(\boldsymbol{X}(k) - \hat{\hat{\boldsymbol{X}}}(k))\, (\boldsymbol{X}(k) - \hat{\hat{\boldsymbol{X}}}(k))^T \right] \tag{57b}$$

where

$$\boldsymbol{X}(k) \triangleq \begin{bmatrix} \boldsymbol{x}_1(k) \\ R_r(k+n) \\ S_c(k+1) \end{bmatrix}. \tag{57c}$$

The above Kalman filtering equations can be divided into two parts [14]:

i) the state prediction and update part: (55a), (55b), (55e), and (55g);

ii) the Kalman gain evaluation part: (55c), (55d), and (55f).

The Kalman gain evaluation in part ii) is determined by the model parameters and the initial estimate of the mean of the state vector. Note that parts of $\boldsymbol{P}_a(k)$ generated in (55f) will not be needed when the processing is moved to the $k + 1$th stage. These parts correspond to two blocks of image data estimated at stages $(k - 1)$ and $(k - n - 1)$ and the state vector $R_r(k + n)$ which is not used until the processing is moved to the $k + n$th stage. These parts will be replaced by the corresponding parts of previously stored $\boldsymbol{P}_a$'s, i.e., those of the image block and the state $R_r(k + 1)$ estimated at $k - n + 1$th stage. This newly formed $\boldsymbol{P}_a(k)$ is now used in (55c) to generate $\boldsymbol{P}_b(k + 1)$. However, since the Kalman gain matrix converges to a constant matrix after a certain number of iterations that depends on a particular model, the Kalman gain part can be evaluated off-line until it converges within a predefined tolerance factor and then is used in the state estimation in part i). Kalman filtering is applied to the image through the above model such that at each iteration, a block of noisy and blurred pixels of size $M \times M$ is taken as input and a block of smoothed pixels is produced as its output. When compared to the SISO scheme, a delay at the output is expected because the output is available only after the entire block has been read and processed.

Remark 1: If we consider the relations

$$\boldsymbol{A}'^{-1}\boldsymbol{A} = A_1 \oplus I$$

and

$$\boldsymbol{A}'^{-1}\boldsymbol{B} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$$

where $\oplus$ represents the direct sum operation, and denote

$$\boldsymbol{P}_c(k) \triangleq \boldsymbol{A}'^{-1}\boldsymbol{P}_b(k)\,(\boldsymbol{A}'^{-1})^T \tag{58a}$$

where

$$\boldsymbol{P}_c(k) \triangleq \mathcal{E}\left[(\boldsymbol{X}_c(k) - \hat{\boldsymbol{X}}_c(k))(\boldsymbol{X}_c(k) - \hat{\boldsymbol{X}}_c(k))^T \right] \tag{58b}$$

and

$$\boldsymbol{X}_c(k) \triangleq \begin{bmatrix} \boldsymbol{x}_1(k) \\ \hat{R}_r(k) \\ \hat{S}_c(k) \end{bmatrix}. \tag{58c}$$

Now using these relations, (55c), (55d), and (55f) become

$$\boldsymbol{P}_c(k) = [A_1 \oplus I](\boldsymbol{P}_a(k-1)\,[A_1 \oplus I]^T + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \boldsymbol{Q}_u [B_1^T\ 0] \tag{59a}$$

$$\boldsymbol{K}(k) = \boldsymbol{A}'\boldsymbol{P}_c(k)\,\boldsymbol{C}^T\{\boldsymbol{C}\boldsymbol{P}_c(k)\,\boldsymbol{C}^T + \boldsymbol{Q}_v\}^{-1} \tag{59b}$$

$$\boldsymbol{P}_a(k) = \boldsymbol{A}'\boldsymbol{P}_c(k)\,\boldsymbol{A}'^T - \boldsymbol{K}(k)\,\boldsymbol{C}\boldsymbol{P}_c(k)\,\boldsymbol{A}'^T. \tag{59c}$$

Closer investigation reveals that the above equations represent a combination of filtering and one-step prediction equations. This is due to the fact that the states R_r and S_c are evaluated one step ahead based upon the given information in the current block $\boldsymbol{y}(k)$. Thus, the parts of the above equations which correspond to these states are similar to those of one-step predictor [14]. This is also evident by the presence of matrix A' in (59b) and (59c) and its special structure. As far as x_1 is concerned, the formulations show filtering process, because the relevant term corresponding to x_1 in A' is an identity matrix and thus the equations for this part are analogous to those of filtering equations.

Remark 2: It is observed that the dimension of the state vector in (54) is $M(4M + n_1 + n_2)$. If M is chosen to be greater than n_1 and n_2, the application of Kalman filtering equations would result in order of $\mathcal{O}(M^6)$ computations. Since a block of the output pixels of size $M \times M$ is produced at each iteration, the order of computations per each output pixel is $\mathcal{O}(M^4)$. Subsections of $x_1(k)$'s ($M \times M$ elements) and $R_r(k)$'s ($M \times n_1$ elements) of a strip have to be stored for use in processing the following strip. If the image is stored row by row on sequential-row-access devices, sufficient memory must be reserved so that the image can be read and written M rows at a time. The memory requirement for the data storage in the block Kalman filtering process is $n \times M \times (2M + n_1)$ or $N \times (2M + n_1)$. However, since the matrices in the Kalman filter equations require $\mathcal{O}(M^4)$ storage, the overall memory requirement of the block Kalman filtering will be $\mathcal{O}(M^4)$.

The order of computations required for the RUKF [4] per each output pixel is found to be $\mathcal{O}(M^3N)$ where M in this reference is the order of the NSHP recursive model and N is the width of the image. For practical applications, an approximate RUKF is suggested in [4] which further reduces the order of computations to $\mathcal{O}(M^4)$, i.e., independent of N. This saving is achieved by confining the region where the error covariance and gain are evaluated to a fixed region which encloses the region of support for the local state. As a result, the block Kalman filtering requires the same order of computations as with the approximate RUKF. However, the memory requirement for the block Kalman filtering is smaller than that of the approximate RUKF [$\mathcal{O}(M^3N)$]. The strip filtering scheme in [8] requires an order of computations of $\mathcal{O}(M^2L^3)$ per each output pixel, where M in this reference is the strip width and L is the order of the AR process. If L is chosen to be equal to M (as in our block state-space model), the order of computations becomes $\mathcal{O}(M^5)$ per each output pixel which is larger than that of the block Kalman filter.

V. Implementation

The proposed block Kalman processor has been implemented to restore the "FRISBEE image" corrupted by both WG noise and LSI blur. Fig. 6 shows the original FRISBEE image which contains fine details. This image has a resolution of 512 × 512 pixels and the number of gray levels which is 256. There are basically three processes involved in our simulations: a) the model building process; b) the corruption process; and c) the restoration process. In the model builder program, first, the image model is constructed using the method in Section II; second, the blur model is generated based upon the structures derived in Section III. The LSI blur with PSF of

$$h(i, j) = e^{-0.8i} e^{-0.8j} \qquad i, j > 0 \tag{60}$$

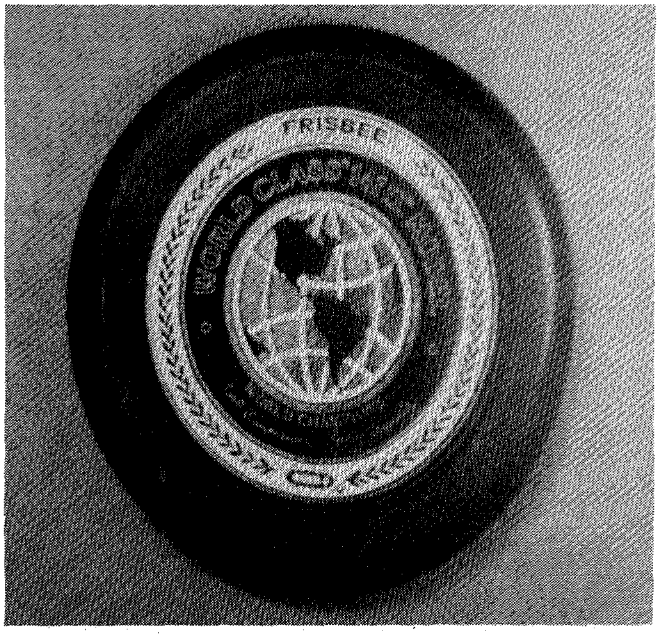


Fig. 6. Original FRISBEE image.

is used. This has been realized by the following 2-D SISCO state-space equations:

$$R(i + 1, j) = e^{-0.8} R(i, j) + e^{-0.8} S(i, j) + f(i, j) \tag{61a}$$

$$S(i, j + 1) = e^{-0.8} S(i, j) + f(i, j) \tag{61b}$$

$$y(i, j) = e^{-0.8} R(i, j) + e^{-0.8} S(i, j) + f(i, j). \tag{61c}$$

Note that both n_1 and n_2 are equal to 1 for this model. These equations are used to generate the corresponding 2-D block state-space equations. The composite block dynamic model in (54) is then formed which will be used in the restoration process. In the corruption program, the image is first blurred with the above PSF which has been implemented using the following difference equations:

$$q(i, j) = e^{-0.8} q(i, j - 1) + f(i, j) \tag{62a}$$

$$y(i, j) = e^{-0.8} y(i - 1, j) + q(i, j). \tag{62b}$$

WG noise of zero mean is then added to the image and the variance of this noise is chosen to provide signal-to-noise ratios (SNR) of 0 and 15 dB. The degraded images for these cases are shown in Figs. 7 and 8, respectively. In the restoration process, the Kalman gain is first pre-evaluated and the approximate steady state is obtained in 18 iterations. Then the block Kalman filtering equations developed in Section IV are implemented to estimate a block of the image. A block size of 4 × 4 is chosen ($M = 4$). Since the AR process in (5) takes into account the correlations of pixels in neighboring strips, the effective correlation distance of the block state-space model in (33) is 8 × 8. The matrices in the model of (51) contain many identity and zero elements, thus, efficient algorithms which take advantage of the specific structure of

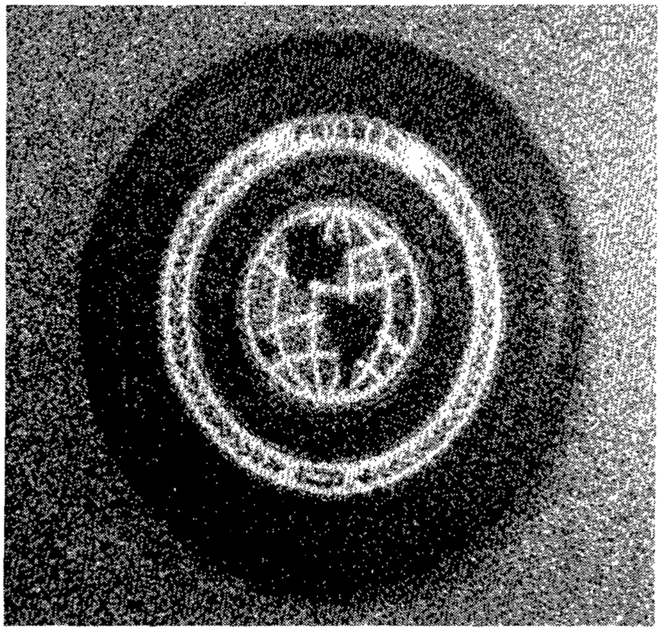


Fig. 7. Degraded image (SNR = 0 dB).

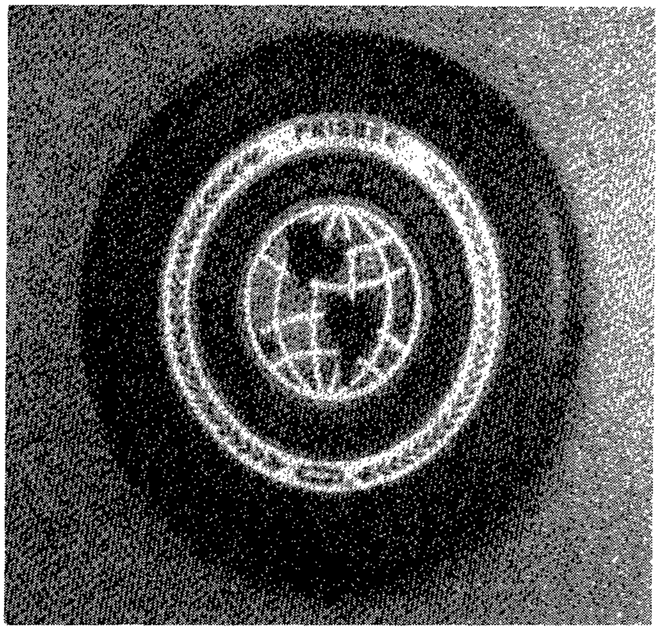


Fig. 8. Degraded image (SNR = 15 dB).

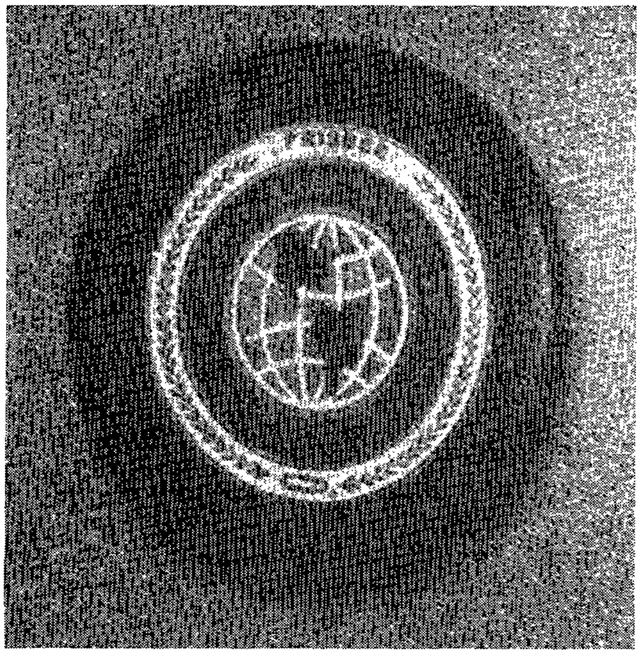


Fig. 9. Restored image (SNR = 9.2 dB).

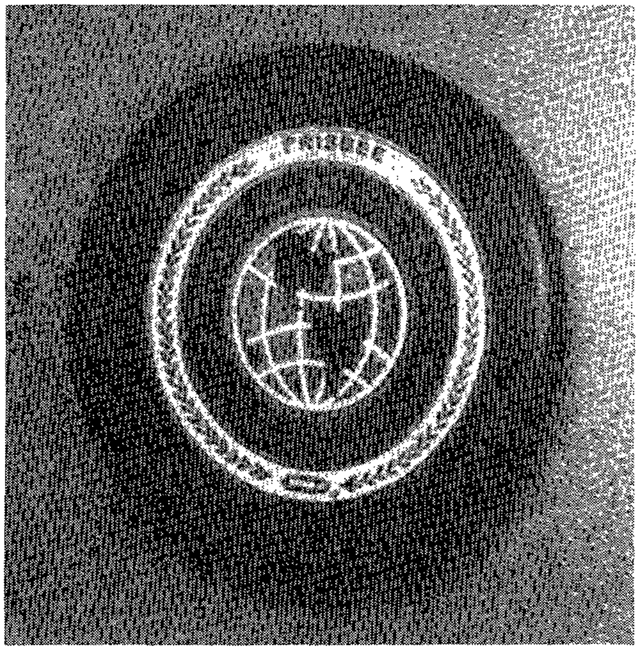


Fig. 10. Restored image (SNR = 21.6 dB).

these sparse matrices have been developed to enhance the computational speed further. The restoration process is performed on a VAX 11/780 computer and takes approximately 150 s for the entire image. The signal-to-noise ratios of the restored images are measured to be 9.2 and 21.6 dB which show considerable improvement in the quality of the processed images. The restored images using block Kalman filtering method developed in this paper are shown in Figs. 9 and 10, respectively.

VI. Conclusion

The block Kalman processing method proposed in this paper is based on developing a vector AR model in a strip which relates a column of pixels to a specified set of columns in the same strip (determined by the order of AR model) and also to those of the previous strip. This model incorporates the correlations between the adjacent strips and, as a result, circumvents the edge effects that are a drawback in conventional strip filtering schemes. The vector AR model is arranged in form of a block state-space equation in order to estimate a block of the degraded image at a time. A recursive formula is derived to solve for the matrix elements in this image generation model. The effect of an infinite extent LSI blur is modeled by a 2-D MIMO block state-space structure derived from a 2-D SISO state-space model. It has been shown [15] that for modally controllable and modally observable 2-D systems, a "modal controller form" or a near canonical realization can be obtained. Without loss of generality, in this paper we have implemented our model using a separable blur for which a canonical realization is directly available if the system is separately locally controllable and observable [16]. It must be mentioned that the separable blur does indeed occur in practical cases such as diffraction limited blur and atmospheric turbulence blur with long exposure recording. The two block state-space models are then combined to form a composite block dynamic model which accounts for both degradations. Block Kalman filtering equations are then derived for this block dynamic model. The amount of computation and the memory requirements for this scheme have been obtained and then compared to those of the other 2-D Kalman processors. This comparison indicates that the order of computation for the block Kalman filter is equivalent to that of approximate RUKF [4], whereas it is reduced by a factor of M (M being the width of a strip) when compared to that of the Kalman strip filter [8]. As a consequence, the 2-D block Kalman filtering technique proposed in this paper provides a powerful method for finding the suboptimal estimates of noisy and blurred images when a vector scanning scheme is considered.

Appendix A

The image model (5) is a system of M linear equations in z and u which is to be evaluated at $(i, j) \in \mathcal{R}$ as defined in (4). Each element of the vector $Z(i, j)$ can be written as

$$z(i + m, j) = \sum_{(k,l)\in\mathcal{W}}\sum a^m_{k,1}\, z(i - k, j - l) + u(i + m, j)\; m = 0, 1, \cdots, M - 1 \tag{A1}$$

where

$$\mathcal{W} = \{(k, 1): (-M + 1 \le k \le M,\; 1 \le l \le M) \cup (1 \le k \le M, \quad l = 0)\}$$

$$a^m_{k,l} = \begin{cases} \varphi_1(m + 1, 1 - k) \quad -M + 1 \le k \le 0, \\ \quad 1 \le l \le M & \text{(A2)} \\ \theta_l(m + 1, M + 1 - k) \quad 1 \le k \le M, \\ \quad 0 \le l \le M & \text{(A3)} \end{cases}$$

$\varphi_l(i, j)$ and $\theta_l(i, j)$ are the (i, j)th entry of the matrices Φ_l and Θ_l, respectively. The signal $u(i + m, j)$ can be viewed as the error in generating $z(i + m, j)$ from a linear combination of $z(i - k, j - l)$, $\forall\, (k, l) \in \mathcal{W}$. By the orthogonality principle, the mean square error $\mathcal{E}[u(i + m, j)^2]$, $m = 0, 1, \cdots, M - 1$, is minimized when

$$\mathcal{E}[u(i + m, j)\, z(i - k, j - l)] = 0 \qquad m = 0, 1, \cdots, M - 1 \quad (k, l) \in \mathcal{W} \qquad (i, j) \in \mathcal{R}. \tag{A4}$$

As can be seen, the scalar model in (A1) is a semicausal minimum variance representation (MVR) driven by colored noise [17]–[19]. In fact, $\{u(i + m, j)\}$ is a moving average and correlated in the noncausal direction "i" and white in the causal direction "j." However, the stationary vector Markovian representation in (5) is a causal vector counterpart of (A1) and possesses the following properties [17]. Arranging (A4) in vector form yields

$$\mathcal{E}[U(i, j)\, Z^T(i - p, j - q)] = 0 \qquad (i, j) \in \mathcal{R}$$
$$(p, q) \in \{(0, 1), (0, 2), \cdots, (0, M), (M, 0), (M, 1), \cdots, (M, M)\}. \tag{A5}$$

Making use of (5), we get

$$\begin{aligned}\mathcal{E}[U(i, j)\, U^T(i - p, j - q)] &= \mathcal{E}[U(i, j)\, Z^T(i - p, j - q)] \\ &\quad - \mathcal{E}[U(i, j)\, Z^T(i - p, j - q - 1)]\, \Phi_1^T \\ &\quad - \cdots - \mathcal{E}[U(i, j)\, Z^T(i - M - p, \\ &\quad j - M - q)]\, \Theta_M^T \qquad (i, j) \in \mathcal{R}.\end{aligned} \tag{A6}$$

By vector Markovian assumption, the symmetrical property of autocorrelation, and (A5), we can deduce from (A6) that

$$\mathcal{E}[U(i, j)\, U^T(i - p, j - q)] = Q_U\, \delta(p, q) \qquad (i, j) \in \mathcal{R} \qquad (p, q) \in \mathcal{S} \tag{A7}$$

where the set $\mathcal{S}$ specifies the entire past of the model and is given by

$$\mathcal{S} = \{(p, q): p = 0, M, 2M, \cdots; q \in [-N, N]\}. \tag{A8}$$

Q_U is the autocorrelation matrix and $\delta(k, l)$ is the Kronecker delta.

Appendix B

Lemma: Given a sequence of matrices Ψ_i, $i = 0, 1, \cdots, M$, and $\Psi_k = 0$ for $k < 0$ or $k > M$, and let us define the elements α_{ij}'s, λ_i's, and β_i's by

$$\alpha_{1j} = \Psi_{M+1-j} \qquad j = 1, 2, \cdots, M \tag{B1}$$

$$\alpha_{ij} = \Psi_{M+i-j} + \Psi_1\alpha_{i-1,j} + \Psi_2\alpha_{i-2,j} + \cdots + \Psi_{i-1}\,\alpha_{1j} \qquad j = 1, 2, \cdots, M \quad i = 2, 3, \cdots, M \tag{B2}$$

$$\lambda_1 = \Psi_0 \tag{B3}$$

$$\lambda_i = \Psi_1\lambda_{i-1} + \Psi_2\lambda_{i-2} + \cdots + \Psi_{i-1}\lambda_1 \qquad i = 2, 3, \cdots, M \tag{B4}$$

$$\beta_i = I_{2M} \tag{B5}$$

$$\beta_i = \Psi_1 \beta_{i-1} + \Psi_2 \beta_{i-2} + \cdots + \Psi_{i-1}\beta_1 \qquad i = 2, 3, \cdots M \tag{B6}$$

then the elements α_{ij}'s, λ_i's, and β_i's for $i = 2, 3, \cdots, M$ can be evaluated recursively by

$$\alpha_{ij} = \alpha_{i-1,j-1} + \alpha_{i-1,M}\,\alpha_{1j} \qquad j = 1, 2, \cdots, M \qquad i = 2, 3, \cdots, M \tag{B7}$$

$$\lambda_i = \alpha_{i-1,M}\lambda_1 \qquad i = 2, 3, \cdots, M \tag{B8}$$

$$\beta_i = \alpha_{i-1,M}\,\beta_1 = \alpha_{i-1,M} \qquad i = 2, 3, \cdots, M. \tag{B9}$$

Proof: The above recursive equations can be proved by induction. To obtain (B7), let $i = 2$, then from (B2) and (B1) we have

$$\begin{aligned}\alpha_{2j} &= \Psi_{M+2-j} + \Psi_1\alpha_{1j} \\ &= \Psi_{M+1-(j-1)} + \Psi_{M+1-M}\,\alpha_{1j} \\ &= \alpha_{1,j-1} + \alpha_{1M}\,\alpha_{1,j} \qquad j = 1, 2, \cdots, M,\end{aligned} \tag{B10}$$

i.e., (B7) holds for $i = 2$.

Assume (B7) is true for $i = 2, 3, \cdots, k$ where $k < M$, i.e.,

$$\alpha_{kj} = \alpha_{k-1,j-1} + \alpha_{k-1,M}\,\alpha_{1j} \qquad j = 1, 2, \cdots, M. \tag{B11}$$

For $i = k + 1$, (B2) yields

$$\begin{aligned}\alpha_{k+1,j} = {} & \Psi_{M+k+1-j} + \Psi_1\alpha_{kj} \\ & + \Psi_2\alpha_{k-1,j} + \cdots + \Psi_k\,\alpha_{1j} \\ = {} & \Psi_{M+k+1-j} + \Psi_1\alpha_{k-1,j-1} \\ & + \Psi_1\alpha_{k-1,M}\,\alpha_{1j} + \Psi_2\alpha_{k-2,j-1} \\ & + \Psi_2\alpha_{k-2,M}\,\alpha_{1j} + \cdots + \Psi_{k-1}\,\alpha_{1,j-1} \\ & + \Psi_{k-1}\,\alpha_{1M}\,\alpha_{1j} + \Psi_k\,\alpha_{1j} \\ = {} & \{\Psi_{M+k-(j-1)} + \Psi_1\alpha_{k-1,j-1} \\ & + \Psi_2\alpha_{k-2,j-1} + \cdots + \Psi_{k-1}\,\alpha_{1,j-1}\} \\ & + \{\Psi_{M+k-M} + \Psi_1\alpha_{k-1,M} + \Psi_2\alpha_{k-2,M} \\ & + \cdots + \Psi_{k-1}\alpha_{1M}\}\,\alpha_{1j} \\ = {} & \alpha_{k,j-1} + \alpha_{kM}\,\alpha_{1j}\end{aligned} \tag{B12}$$

which is (B7) for $i = k + 1$. Thus, (B7) holds for all i. The structures of (B4) and (B6) are identical with a difference in the initial conditions given in (B3) and (B5). Therefore, if (B8) is proved, (B9) can also be proved by the same manner. Now, let us obtain (B8) by induction. Letting $i = 2$ in (B4) gives

$$\lambda_2 = \Psi_1\lambda_1 = \Psi_{M+1-M}\lambda_1 = \alpha_{1M}\lambda_1, \tag{B13}$$

i.e., (B8) holds for $i = 2$.

Assume (B8) is true for $i = 2, 3, \cdots, k$ where $k < M$, i.e.,

$$\lambda_k = \alpha_{k-1,M}\lambda_1. \tag{B14}$$

For $i = k + 1$, (B4) gives

$$\begin{aligned}\lambda_{k+1} &= \Psi_1\lambda_k + \Psi_2\lambda_{k-1} + \cdots + \Psi_k\lambda_1 \\ &= \Psi_1\alpha_{k-1,M}\lambda_1 + \Psi_2\alpha_{k-2,M}\lambda_1 \\ &\quad + \cdots + \Psi_{k-1}\alpha_{1M}\lambda_1 + \Psi_k\lambda_1 \\ &= \{\Psi_{M+k-M} + \Psi_1\alpha_{k-1,M} \\ &\quad + \cdots + \Psi_{k-1}\alpha_{1M}\}\lambda_1 \\ &= \alpha_{kM}\lambda_1\end{aligned} \tag{B15}$$

which is (B8) for $i = k + 1$. Thus, (B8) is valid for all i. Q.E.D.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments and suggestions which have greatly improved the quality of the paper.

References

[1] N. E. Nahi and T. Assefi, "Bayesian recursive image estimation," *IEEE Trans. Comput.*, vol. C-21, pp. 734–738, July 1972.

[2] A. O. Aboutalib and L. M. Silverman, "Restoration of motion degraded images," *IEEE Trans. Circuits. Syst.*, vol. CAS-22, pp. 278–286, Mar. 1975.

[3] A. O. Aboutalib, M. S. Murphy, and L. M. Silverman, "Digital restoration of images degraded by general motion blur," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 294–302, June 1977.

[4] J. W. Woods and C. H. Radewan, "Kalman filtering in two dimensions," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 473–482, July 1977.

[5] J. W. Woods and V. K. Ingle, "Kalman filtering in two dimensions: Further results," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 188–197, Apr. 1981.

[6] S. S. Dikshit, "A recursive Kalman window approach to image restoration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 125–140, Apr. 1982.

[7] J. Biemond, J. Rieske, and J. J. Gerbrands, "A fast Kalman filter for images degraded by both blur and noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1248–1256, Oct. 1983.

[8] B. R. Suresh and B. A. Shenoi, "New results in two-dimensional Kalman filtering with applications to image restoration," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp., 307–319, Apr. 1981.

[9] J. Biemond and R. H. Plompen, "Comments on a recursive Kalman window approach to image restoration," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-31, Dec. 1983.

[10] M. R. Azimi-Sadjadi and P. W. Wong, "An image generation model for 2-D block Kalman filtering," in *Proc. ISCAS'86*, San Jose, CA, May 1986, pp. 411–415.

[11] —, "A 2-D multi-input, multi-output structure for modeling blur in image restoration," in *Proc. 28th Midwest, Symp. Circuits Syst.*, Louisville, KY, Aug. 1985, pp. 458–460.

[12] R. P. Roesser, "A discrete state-space model for linear image processing," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 1–10, Feb. 1975.

[13] M. R. Azimi-Sadjadi and R. A. King, "Two-dimensional block processors—Structures and implementations," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 42–50, Jan. 1986.

[14] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communicating and Control*. New York: McGraw-Hill, 1971, pp. 253–272.

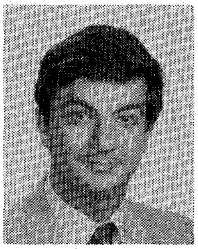
[15] S. Y. Kung, B. C. Levy, M. Morf, and T. Kailath, "New results in 2-D systems theory, Part II: 2-D state-space models—Realization and the notions of controllability, observability and minimality," *Proc. IEEE*, vol. 65, pp. 945–961, June 1977.

[16] T. Hinamoto, "Realization of a state-space model from two-dimensional input-output map," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 36–44, Jan. 1980.

[17] A. K. Jain and J. R. Jain, "Partial differential equations and finite difference methods in image processing—Part II: Image restoration," *IEEE Trans. Automat. Contr.*, vol. AC-23, pp. 817–834, Oct. 1978.

[18] A. K. Jain, "Advances in mathematical models for image processing," *Proc. IEEE*, vol. 69, pp. 502–528, May 1981.

[19] R. L. Kashyap, "Characterization and estimation of two-dimensional ARMA models," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 736–745, Sept. 1984.



Mahmood R. Azimi-Sadjadi (S'81–M'81) was born in Tehran, Iran, in 1952. He received the B.Sc. degree from University of Tehran, Iran, in 1977, the M.Sc. 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, as an Assistant Professor of Electrical Engineering. His main research interests are in the areas of digital signal/image processing and multidimensional system theory and analysis.

Dr. Azimi-Sadjadi is the recipient of the 1984 DOW Chemical Outstanding Young Faculty Award of the American Society for Engineering Education.

Ping Wah Wong (S'84) was born on October 9, 1954. He received the B.Sc. (Eng.) degree from the University of Hong Kong in 1977, and the M.S.E.E. degree from the University of Michigan–Dearborn in 1985.

He is currently a student at the Information Systems Laboratory, Stanford University, Stanford, CA. From 1977 to 1981 he was an Electrical Engineer at Coronet Industries Limited, Hong Kong, where he was involved in designing radio frequency circuits. From 1981 to 1983 he worked on automatic train control systems at Mass Transit Railway Corporation, Hong Kong. His research interest is in digital signal processing, communication, and oversampled analog-to-digital conversion.