An Iterative Method for Designing Separable Wiener Filters

Chang D. Yoo, Matthew M. Bace, and Jae S. Lim, *Fellow, IEEE*

***Abstract*—In this paper, we present a technique for designing the optimal (in the mean square error sense) separable 2-D filter for recovering a signal from a noise-corrupted signal when the joint statistics of the signal and noise are known. A set of nonlinear equations in the design parameters is derived, and an iterative algorithm to solve them is presented. The algorithm is shown to be nondivergent in theory and rapidly convergent in practice. The results of applying the separable filter design algorithm to several typical image recovery problems are given. In most cases, the performance of the resulting separable filter is similar to that of the optimal nonseparable filter with the same region of support.**

I. Introduction

ONE of the most common problems in signal processing is noise cancellation or reduction. It is well known that if the statistics of the undegraded signal and the noise are given, the optimal linear solution that minimizes the mean square error between the recovered signal and the undegraded original signal is the Wiener filter. In this paper, we impose the additional constraint of separability on the 2-D filter design problem. Is such a design possible, and if so, how does the separability constraint affect performance?

It is important to note that this design problem is not the same as trying to fit a separable filter $h_s(n_1, n_2) = h_1(n_1)h_2(n_2)$ to some ideal 2-D filter $h_I(n_1, n_2)$. That problem has been addressed already [1]–[3], [5], [8], [10], and it has been shown that one of the solution may be obtained by computing the singular value decomposition (SVD) of the ideal filter. What we describe in this paper is a method to go directly from the statistics of the signals to the coefficients of the separable filter without the intermediate step of computing the ideal unconstrained 2-D filter. This difference in design methodology is illustrated in Fig. 1. We have found that our separable filter outperforms the separable approximation of the ideal unconstrained 2-D Wiener filter. Performance comparison between the two filters is presented in Section V.

The motivation for developing such a design algorithm is clearly computational efficiency. A separable filter requires significantly fewer computations to implement than a nonseparable filter, particularly for large regions of support [7]. There may also be a large computational saving in the design of a separable filter as compared with the design of a nonseparable filter. Since most design techniques require the solution of a set of linear equations in the design parameters (generally an $O(n^3)$ procedure), the reduction in going from M_1M_2 parameters for an $M_1 \times M_2$ nonseparable filter to $M_1 + M_2 - 1$ parameters for a separable filter of the same size may also be significant.

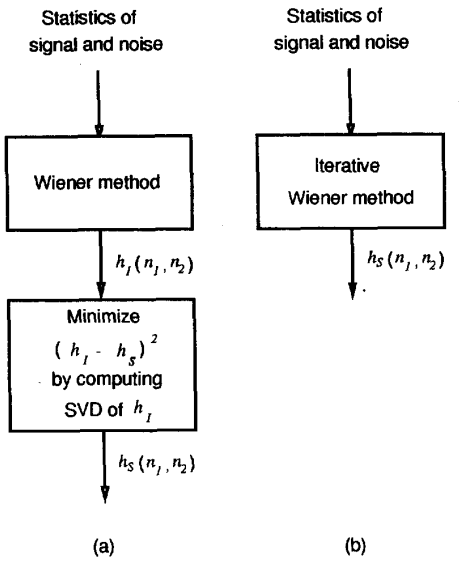


Fig. 1. Filter design methodologies: (a) Fitting a separable filter to an already-designed ideal 2-D filter; (b) directly designing a separable filter given the signal statistics.

As we will see later, minimizing the expected mean square error over the $M_1 + M_2 - 1$ design parameters requires solving a set of nonlinear equations. In Section II, we present a spatial domain algorithm for solving this set of equations. We show that the set of equations has a bilinear structure so that it may be solved in a straightforward manner by an iterative algorithm. The bilinear structure permits the coefficients of the 1-D filter to be optimized (locally) by solving a set of linear equations, whereas the coefficients of another 1-D filter are held constant. In Section III, we derive the optimization equations in the discrete Fourier transform domain and present an iterative algorithm for solving them. Because the decoupling of the parameters is more natural in the Fourier domain, the Fourier domain algorithm is more efficient than the spatial domain algorithm for large filter sizes. The convergence properties of both algorithms are discussed in Section V. The algorithm is nondivergent in theory and rapidly convergent in practice. Because of this rapid convergence, the

Manuscript received July 20, 1989; revised November 2, 1993. This work was supported by the National Science Foundation under Grant Nos. ECS-8407285 and MIP-8714969 and by the Advanced Television Research Program. The associate editor coordinating the review of this paper and approving it for publication was Prof. Jan Allenbach.

The authors are with the Research Laboratory of Electronics, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

IEEE Log Number 9407043.

separable filter design algorithm, despite its iterative nature, requires far fewer computations to reach a solution than any algorithm for designing optimal nonseparable filters. Finally, in Section V, we use several image restoration examples to compare the performance of the optimal separable filters with that of the optimal unconstrained Wiener filters. We show that the loss in SNR incurred by using a separable filter is small.

II. Spatial Domain Algorithm

In this section, we develop a set of nonlinear equations that results when we attempt to minimize the mean square error with respect to the separable filter design parameters. Consider a degraded signal $g(n_1, n_2)$, which is a signal $f(n_1, n_2)$ degraded by additive random noise $w(n_1, n_2)$, i.e.

$$g(n_1, n_2) = f(n_1, n_2) + w(n_1, n_2). \tag{1}$$

We assume that $f(n_1, n_2)$ and $w(n_1, n_2)$ are real-valued uncorrelated stationary processes with known autocorrelation functions $R_f(n_1, n_2)$ and $R_w(n_1, n_2)$ (it is possible to treat the cases of complex and correlated signals, but for the sake of simplicity, we make these assumptions). We wish to recover $f(n_1, n_2)$ from $g(n_1, n_2)$ by filtering it with a linear shift-invariant separable filter $h_s(n_1, n_2) = h_1(n_1)h_2(n_2)$. Furthermore, we assume that $h_1(n_1)$ and $h_2(n_2)$ have finite support since the purpose of the separability constraint is to make high-speed convolution possible in the spatial domain.

A. Optimization Equations

We wish to determine the optimal $M_1 \times M_2$ filter $h_s(n_1, n_2)$ that minimizes the expected value of the square of the error, where the error $e(n_1, n_2)$ is defined by

$$e(n_1, n_2) = \hat{f}(n_1, n_2) - f(n_1, n_2)$$
$$\hat{f}(n_1, n_2) = h_s(n_1, n_2) * g(n_1, n_2).$$

We will assume that $h_s(n_1, n_2)$ has the rectangular region of support

$$R_{M_1 \times M_2} = \{(n_1, n_2) | -N_1 \le n_1 \le N_1, -N_2 \le n_2 \le N_2\} \tag{2}$$

where $M_1 = 2N_1 + 1$ and $M_2 = 2N_2 + 1$. If we define our filter parameters as $h_1(l_1)$ and $h_2(l_2)$ for $(l_1, l_2) \in R_{M_1 \times M_2}$, then the expected error is given by [9]

$$\begin{aligned} E &= E[e^2(n_1, n_2)] \\ &= R_f(0,0) - 2\sum_{n_1}\sum_{n_2} h_1(n_1)h_2(n_2)R_f(n_1, n_2) \\ &\quad + \sum_{n_1}\sum_{n_2}\sum_{m_1}\sum_{m_2} h_1(n_1)h_2(n_2)h_1(m_1)h_2(m_2) \\ &\quad \times R_g(n_1 - m_1, n_2 - m_2) \end{aligned} \tag{3}$$

where n_1 and m_1 run from $-N_1$ to N_1, and n_2 and m_2 run from $-N_2$ to N_2 in each summation and

$$R_g(n_1, n_2) = R_f(n_1, n_2) + R_w(n_1, n_2). \tag{4}$$

Setting $\frac{\partial E}{\partial h_1(l_1)} = 0$ yields

$$\begin{aligned} &\sum_{n_2=-N_2}^{N_2} h_2(n_2)R_f(l_1, n_2) \\ &\quad = \sum_{n_2=-N_2}^{N_2}\sum_{m_1=-N_1}^{N_1}\sum_{m_2=-N_2}^{N_2} h_2(n_2)h_1(m_1)h_2(m_2) \\ &\qquad \times R'_g(l_1 - m_1, n_2 - m_2) \end{aligned} \tag{5}$$

where $R'_g(n_1, n_2)$ is the part of $R_g(n_1, n_2)$ that is symmetric about the n_1 axis and is given by

$$R'_g(n_1, n_2) = \frac{1}{2}(R_g(n_1, n_2) + R_g(-n_1, n_2)). \tag{6}$$

Similarly, setting $\frac{\partial E}{\partial h_2(l_2)} = 0$ yields

$$\begin{aligned} &\sum_{n_1=-N_1}^{N_1} h_1(n_1)R_f(n_1, l_2) \\ &\quad = \sum_{n_1=-N_1}^{N_1}\sum_{m_1=-N_1}^{N_1}\sum_{m_2=-N_2}^{N_2} h_1(n_1)h_1(m_1)h_2(m_2) \\ &\qquad \times R''_g(n_1 - m_1, l_2 - m_2) \end{aligned} \tag{7}$$

where $R''_g(n_1, n_2)$ is the part of $R_g(n_1, n_2)$ that is symmetric about the n_2 axis and is given by

$$R''_g(n_1, n_2) = \frac{1}{2}(R_g(n_1, n_2) + R_g(n_1, -n_2)). \tag{8}$$

Note that for real signals $R_g(n_1, n_2)$ is symmetric about the origin so that $R_g(-n_1, n_2) = R_g(n_1, -n_2)$. Thus, in the case of real signals, we have

$$R'_g(n_1, n_2) = R''_g(n_1, n_2) = R^s_g(n_1, n_2) \tag{9}$$

where $R^s_g(n_1, n_2)$ is symmetric about both axes.

B. Iterative Algorithm

By inspecting (5) and (7), one can easily see that solving for the filter coefficients is a nonlinear problem. Fortunately, though, the nonlinear equations can be docoupled into two sets of linear equations that may be solved iteratively.

Equations (5) and (7) serve as the basis for the iterative spatial domain algorithm. Note that if the $h_2(l_2)$ are specified for $-N_2 \le l_2 \le N_2$, then the M_1 equations given by (5) are linear in the $h_1(l_1)$. Similarly, specifying the $h_1(l_1)$ for $-N_1 \le l_1 \le N_1$ makes the M_2 equations given by (7) linear in the $h_2(l_2)$. If we write these two linear systems in matrix form, we have

$$\begin{bmatrix} A(-N_1, -N_1) & \cdots & A(-N_1, N_1) \\ \vdots & & \vdots \\ A(N_1, -N_1) & \cdots & A(N_1, N_1) \end{bmatrix} \begin{bmatrix} h_1(-N_1) \\ \vdots \\ h_1(N_1) \end{bmatrix} = \begin{bmatrix} B(-N_1) \\ \vdots \\ B(N_1) \end{bmatrix} \tag{10}$$

and

$$\begin{bmatrix} C(-N_2,-N_2) & \cdots & C(-N_2,N_2) \\ \vdots & & \vdots \\ C(N_2,-N_2) & \cdots & C(N_2,N_2) \end{bmatrix} \begin{bmatrix} h_2(-N_2) \\ \vdots \\ h_2(N_2) \end{bmatrix} = \begin{bmatrix} D(-N_2) \\ \vdots \\ D(N_2) \end{bmatrix} \tag{11}$$

where

$$A(n_1,m_1) = \sum_{n_2=-N_2}^{N_2} \sum_{m_2=-N_2}^{N_2} h_2(n_2)h_2(m_2) \times R_g^s(n_1-m_1,n_2-m_2) \tag{12a}$$

$$B(n_1) = \sum_{n_2=-N_2}^{N_2} h_2(n_2)R_f(n_1,n_2) \tag{12b}$$

$$C(n_2,m_2) = \sum_{n_1=-N_1}^{N_1} \sum_{m_1=-N_1}^{N_1} h_1(n_1)h_1(m_1) \times R_g^s(n_1-m_1,n_2-m_2) \tag{12c}$$

$$D(n_2) = \sum_{n_1=-N_1}^{N_1} h_1(n_1)R_f(n_1,n_2). \tag{12d}$$

We note further that the function $A(n_1,m_1)$ depends only on the difference n_1-m_1 and similarly that $C(n_2,m_2)$ depends only on the difference n_2-m_2. Combining this observation with the symmetry of $R_g^s(n_1,n_2)$, we can write

$$A(n_1,m_1) = A(m_1,n_1) = A(l_1)|_{l_1=|n_1-m_1|} = \sum_{n_2=-N_2}^{N_2} \sum_{m_2=-N_2}^{N_2} h_2(n_2)h_2(m_2)R_g^s(l_1,n_2-m_2) \tag{13a}$$

$$C(n_2,m_2) = C(m_2,n_2) = C(l_2)|_{l_2=|n_2-m_2|} = \sum_{n_1=-N_1}^{N_1} \sum_{m_1=-N_1}^{N_1} h_1(n_1)h_1(m_1)R_g^s(n_1-m_1,l_2). \tag{13b}$$

Making the above substitutions into (10) and (11), we have the $M_1 \times M_1$ symmetric positive-definite Toeplitz system

$$\begin{bmatrix} A(0) & \cdots & A(2N_1) \\ \vdots & & \vdots \\ A(2N_1) & \cdots & A(0) \end{bmatrix} \mathbf{h}_1 = \mathbf{b} \tag{14}$$

and the $M_2 \times M_2$ symmetric positive-definite Toeplitz system

$$\begin{bmatrix} C(0) & \cdots & C(2N_2) \\ \vdots & & \vdots \\ C(2N_2) & \cdots & C(0) \end{bmatrix} \mathbf{h}_2 = \mathbf{d} \tag{15}$$

where

$$\mathbf{h}_1 = \begin{bmatrix} h_1(-N_1) \\ \vdots \\ h_1(N_1) \end{bmatrix} \tag{16a}$$

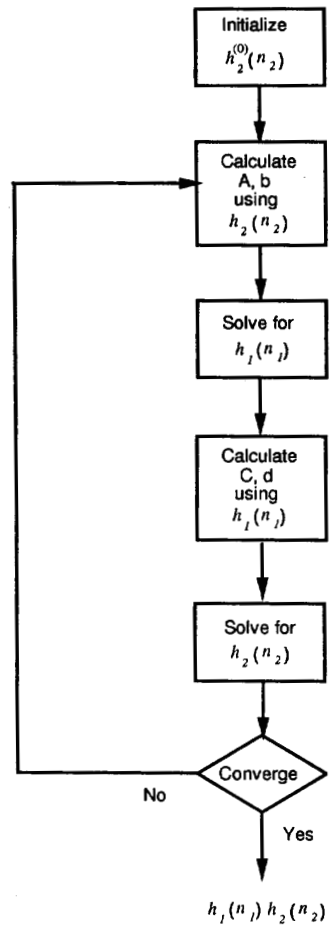


Fig. 2. Iterative spatial domain algorithm.

$$\mathbf{h}_2 = \begin{bmatrix} h_2(-N_2) \\ \vdots \\ h_2(N_2) \end{bmatrix} \tag{16b}$$

$$\mathbf{b} = \begin{bmatrix} B(-N_1) \\ \vdots \\ B(N_1) \end{bmatrix} \tag{16c}$$

$$\mathbf{d} = \begin{bmatrix} D(-N_2) \\ \vdots \\ D(N_2) \end{bmatrix}. \tag{16d}$$

Now, we summarize the steps involved in the spatial domain iterative algorithm for computing the optimal filter coefficients $h_1(n_1)$ and $h_2(n_2)$. We begin by initializing $h_2(n_2)$ to some arbitrary filter (e.g., $h_2^{(0)}(n_2) = \delta(n_2)$). Each iteration of the algorithm then proceeds as shown in Fig. 2 ($\mathbf{h}_1^{(i)}$ and $\mathbf{h}_2^{(i)}$ are the values of $\mathbf{h}_1$ and $\mathbf{h}_2$ after i iterations of the algorithm).

C. Implementation

During each iteration of the spatial domain algorithm, two symmetric positive-definite Toeplitz systems must be solved.

Usually, solving an $N \times N$ set of linear equations is an $O(N^3)$ process; however, in the special case of positive-definite Toeplitz matrices, only $O(N^2)$ computations are required. Through the application of the Levinson–Durbin recursion [4], an $N \times N$ symmetric positive-definite Toeplitz system can be solved in roughly $2N^2$ operations, which is a savings in computations by a factor of $O(N)$.

The most computationally expensive part of the spatial domain algorithm is the evaluation of the matrix elements $A(i,j)$ and $C(i,j)$ at each iteration. Despite the redundancy of the $A(i,j)$ and $C(i,j)$, about $2M_1M_2(M_1+M_2)$ operations per iteration are required to compute both of them for all values of i and j. For filters with large regions of support, these computations will clearly outweigh the computations involved in solving the linear systems (roughly $2(M_1^2+M_2^2)$ operations per iteration). However, despite this cubic dependence in the computational requirement, the spatial domain algorithm is generally more efficient than the Fourier domain algorithm discussed in Section III for most reasonable filter sizes ($M_1, M_2 \leq 15$). This efficiency arises from the fact that the spatial domain algorithm requires only real operations and works on a much smaller region of support than the Fourier domain algorithm.

In both the spatial domain algorithm and the Fourier domain algorithm presented later, there exists the need for rescaling. Suppose, for instance, that we have a set of filter coefficients satisfying (5) and (7):

$$h_1(n_1) = x_1(n_1) \tag{17a}$$

and

$$h_2(n_2) = x_2(n_2). \tag{17b}$$

Then, it is clear that the filters

$$h_1(n_1) = \alpha x_1(n_1) \tag{18a}$$

and

$$h_2(n_2) = \frac{1}{\alpha} x_2(n_2) \tag{18b}$$

will also satisfy the equations for all values of α.

Because of the scaling involved, there is a fundamental instability in the algorithm. If the iteration is allowed to continue indefinitely, then eventually, the magnitudes of the coefficients $h_1(n_1)$ will grow infinitely large, whereas the magnitudes of the coefficients $h_2(n_2)$ will become infinitesimally small, or vice versa. Thus, in order to correct for this instability, we need to rescale the coefficients to roughly the same order of magnitude after each iteration.

Essentially, the purpose of rescaling is to eliminate the extra degree of freedom we introduced by allowing an arbitrary scale factor in both $h_1(n_1)$ and $h_2(n_2)$. Note, for instance, that we could, without loss of generality, constrain $h_1(n_1)$ so that $h_1(0)$ is always equal to 1. A careful analysis of (5) and (7) shows that enforcing this constraint after each iteration is equivalent to setting $h_1(0) = 1$ from the beginning and reducing the number of parameters by one. However, this latter approach destroys much of the symmetry in (14) and (15).

III. Fourier Domain Algorithm

Since the filter we design is a linear shift-invariant system, all of the preceding developments naturally have Fourier domain analogues. In fact, the decoupling of the set of nonlinear equations mentioned above becomes much more apparent in the discrete Fourier transform domain.

A. Optimization Equations

We begin by transforming (5) and (7) into their discrete Fourier transform domain equivalents. Suppose that we have the following transform relations:

$$R_f(n_1,n_2) \Leftrightarrow S_f(k_1,k_2) \tag{19}$$

$$R_g(n_1,n_2) \Leftrightarrow S_g(k_1,k_2) \tag{20}$$

$$h_1(n_1) \Leftrightarrow H_1(k_1) \tag{21}$$

$$h_2(n_2) \Leftrightarrow H_2(k_2) \tag{22}$$

where we define the symmetric discrete Fourier transform $X(k_1,k_2)$ of a signal $x(n_1,n_2)$ as

$$X(k_1,k_2) = \sum_{n_1=-P_1}^{P_1} \sum_{n_2=-P_2}^{P_2} x(n_1,n_2) e^{-j\frac{2\pi}{Q_1}n_1k_1} e^{-j\frac{2\pi}{Q_2}n_2k_2}. \tag{23}$$

Consider the right-hand side of (5). The summations over m_1 and m_2 are simply computing the convolution of $R_g^s(n_1,n_2)$ with $h_s(n_1,n_2) = h_1(n_1)h_2(n_2)$, and therefore, it is possible to rewrite (5) as

$$\sum_{n_2=-N_2}^{N_2} h_2(n_2) R_f(l_1,n_2) = \sum_{n_2=-N_2}^{N_2} h_2(n_2) r(l_1,n_2) \tag{24}$$

where $r(n_1,n_2) = R_g^s(n_1,n_2) * h_s(n_1,n_2)$. We may similarly rewrite (7) as

$$\sum_{n_1=-N_1}^{N_1} h_1(n_1) R_f(n_1,l_2) = \sum_{n_1=-N_1}^{N_1} h_1(n_1) r(n_1,l_2). \tag{25}$$

From (24) and (25), it is clear that in order to compute the optimal $h_1(n_1)$ for $-N_1 \leq n_1 \leq N_1$ and $h_2(n_2)$ for $-N_2 \leq n_2 \leq N_2$, the values of $r(n_1,n_2)$ are needed only for points (n_1,n_2) in the region $R_{M_1 \times M_2}$. Thus, $R_f(n_1,n_2)$ and $R_g(n_1,n_2)$ must be known for $(n_1,n_2) \in R_{Q_1 \times Q_2}$, where

$$R_{Q_1 \times Q_2} = \{(n_1,n_2) | -P_1 \leq n_1 \leq P_1, -P_2 \leq n_2 \leq P_2\} \tag{26}$$

where the parameters P_1, P_2, Q_1, and Q_2 in (26) are given by

$$P_1 = 2N_1 \tag{27a}$$

$$P_2 = 2N_2 \tag{27b}$$

$$Q_1 = 2P_1 + 1 \tag{27c}$$

$$Q_2 = 2P_2 + 1 \tag{27d}$$

It follows, then, that in order to avoid the effects of aliasing, the discrete Fourier transforms in (19) and (20) must be computed on a grid of size at least $Q_1 \times Q_2$. Suppose we compute $r(n_1,n_2)$ by calculating the inverse discrete Fourier transform of the product $\Re e[S_g(k_1,k_2)]H_1(k_1)H_2(k_2)$ using a sample grid of $Q_1 \times Q_2$ points. The values of $r(n_1,n_2)$

for (n_1, n_2) outside of $R_{M_1 \times M_2}$ will be corrupted because we will be computing the circular, rather than linear, convolution of $R_g^s(n_1, n_2)$ and $h_s(n_1, n_2)$. However, since those values of $r(n_1, n_2)$ are not needed, we can use a grid of only $Q_1 \times Q_2$ points.

If we multiply both sides of (5) by $e^{-j\frac{2\pi}{Q_1}l_1 k_1}$ and then sum over l_1 from $-P_1$ to P_1, we get the "semi-Fourier" domain equation

$$H_1(k_1) \sum_{n_2=-N_2}^{N_2} \sum_{m_2=-N_2}^{N_2} h_2(n_2)h_2(m_2)(S_g(k_1, n_2 - m_2) + S_g^*(k_1, n_2 - m_2)) = 2 \sum_{n_2=-N_2}^{N_2} h_2(n_2) S_f(k_1, n_2) \tag{28}$$

where

$$S_f(k_1, n_2) = \sum_{n_1=-P_1}^{P_1} R_f(n_1, n_2) e^{-j\frac{2\pi}{Q_1}n_1 k_1} = \frac{1}{Q_2} \sum_{k_2=-P_2}^{P_2} S_f(k_1, k_2) e^{j\frac{2\pi}{Q_2}n_2 k_2} \tag{29}$$

and

$$S_g(k_1, n_2) = \sum_{n_1=-P_1}^{P_1} R_g(n_1, n_2) e^{-j\frac{2\pi}{Q_1}n_1 k_1} = \frac{1}{Q_2} \sum_{k_2=-P_2}^{P_2} S_g(k_1, k_2) e^{j\frac{2\pi}{Q_2}n_2 k_2}. \tag{30}$$

Using the expressions in (29) and (30)

$$H_1(k_1) = \frac{\sum_{k_2=-P_2}^{P_2} S_f(k_1, k_2) H_2^*(k_2)}{\sum_{k_2=-P_2}^{P_2} S_g(k_1, k_2) \|H_2(k_2)\|^2}. \tag{31}$$

It is possible to perform a similar transformation on (7) in order to arrive at

$$H_2(k_2) = \frac{\sum_{k_1=-P_1}^{P_1} S_f(k_1, k_2) H_1^*(k_1)}{\sum_{k_1=-P_1}^{P_1} S_g(k_1, k_2) \|H_1(k_1)\|^2}. \tag{32}$$

B. Iterative Algorithm

From (31) and (32), it is clear that solving for the filter coefficients is a nonlinear problem in the Fourier domain as well. However, as with the spatial domain algorithm, the nonlinear equations can be solved efficiently by a simple iteration.

In the Fourier domain, we see that if we are given $H_2(k_2)$ for $-P_2 \leq k_2 \leq P_2$, we can compute $H_1(k_1)$ for $-P_1 \leq k_1 \leq P_1$ in a straightforward manner by applying (31). Similarly, we can use (32) to compute $H_2(k_2)$ from $H_1(k_1)$. This observation naturally suggests an iterative approach to computing the optimal filter coefficients. Clearly, we could make some initial guess at $H_2(k_2)$ (e.g., $H_2(k_2) = 1$ for all values of k_2) and then alternately apply (31) and (32) in an iterative manner in the hope that the algorithm will eventually converge to the correct solution. This algorithm is shown in Fig. 3.

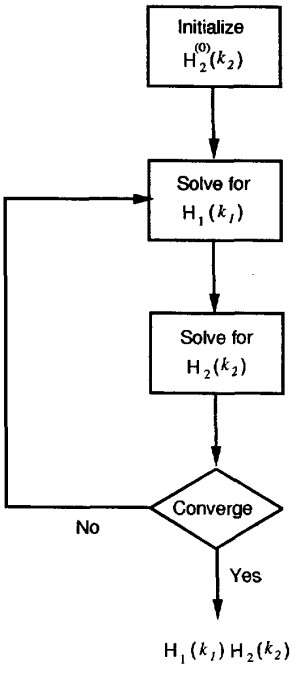


Fig. 3. Iterative Fourier domain algorithm.

C. Implementation

The Fourier domain implementation is more straightforward than its spatial domain counterpart. The computations involved in computing $H_1(k_1)$ from $H_2(K_2)$ or vice versa are essentially one complex vector inner product and two complex matrix/vector multiplications. The total number of complex operations (multiply/adds) per iteration is therefore roughly $4Q_1Q_2 + Q_1 + Q_2$, which is a quadratic dependence on the filter dimensions.

This quadratic dependence means that the Fourier domain implementation is more efficient than the spatial domain implementation. If we assume that each complex operation involves four real operations and we note that $Q_1 \approx 2M_1$ and $Q_2 \approx 2M_2$, then the total number of real operations per iteration of the Fourier domain algorithm is $64M_1M_2 + 8M_1 + 8M_2$. This means that for the Fourier domain algorithm to be more efficient than the spatial domain algorithm, the filter size must be fairly large $(M_1, M_2 \geq 16)$.

As with the spatial domain algorithm, rescaling is necessary after every iteration. There are many different ways to accomplish this rescaling, but the most obvious is to set $H_1(0)$ to 1 after each iteration and scale the remaining $H_1(k_1)$ and $H_2(k_2)$ accordingly. Of course, if one wanted to maintain compatibility with the spatial domain constraint $h_1(0) = 1$, then one could impose the constraint

$$\frac{1}{M_1} \sum_{k_1=-N_1}^{N_1} H_1(k_1) = 1 \tag{33}$$

after each iteration.

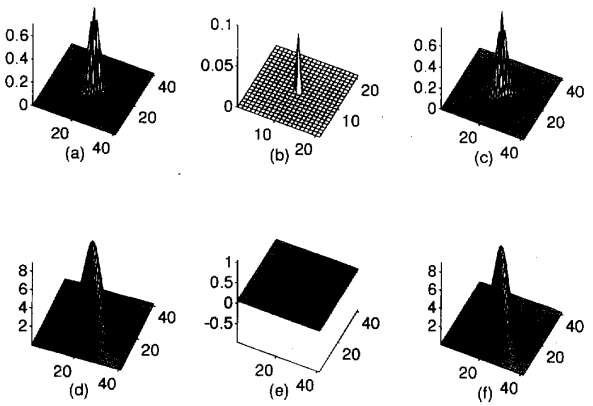


Fig. 4. Autocorrelations functions of (a) the uncorrupted signal $R_f(n_1, n_2)$; (b) white Gaussian noise $R_n(n_1, n_2)$; (c) corrupted signal $R_f(n_1, n_2) + R_n(n_1, n_2)$. Power spectra of (d) the uncorrupted signal $S_f(n_1, n_2)$; (e) white Gaussian noise $S_n(n_1, n_2)$; (f) corrupted signal $S_f(n_1, n_2) + S_n(n_1, n_2)$.

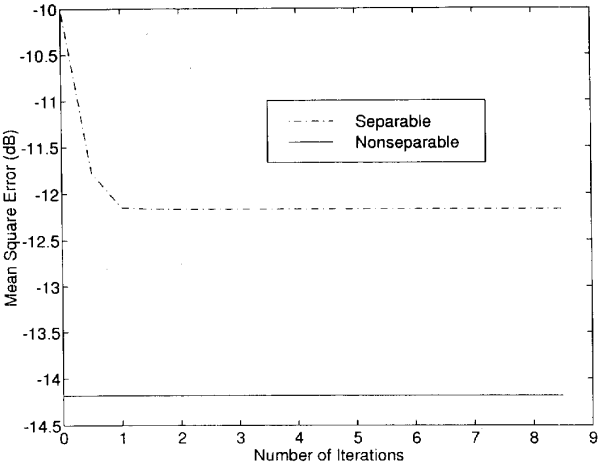


Fig. 5. Convergence of separable filter design algorithm for the case of a random 2-D signal with diagonally directed power spectrum.

IV. Convergence

In this section, we analyze the convergence properties of both the spatial domain and the Fourier domain optimization algorithms. Since the Fourier domain optimization equations are simply the transformed versions the spatial domain optimization equations, both algorithms solve the same set of equations. Thus, provided that the initial conditions are the same, after an equal number of iterations of both algorithms, $H_1(k_1)$ will be the transform of $h_1(n_1)$, and $H_2(k_2)$ will similarly correspond to $h_2(n_2)$. Clearly, then, if we can show that either algorithm converges, we have shown that both algorithms converge.

Since we began with an error criterion defined in the spatial domain, it is easier to analyze the convergence of the spatial domain algorithm. Suppose that after k iterations, the error is $E(\mathbf{h}_1^{(k)}, \mathbf{h}_2^{(k)})$. During the first half of iteration $k+1$, we solve the set of linear equations

$$\left.\frac{\partial E}{\partial \mathbf{h}_1}\right|_{h_2 = h_2^{(k)}} = \mathbf{0}. \tag{34}$$

It is important to note that $E(\mathbf{h}_1, \mathbf{h}_2)$ is a nonnegative quadratic function in $\mathbf{h}_1$ with $\mathbf{h}_2$ fixed so that $E(\mathbf{h}_1, \mathbf{h}_2)$ has a single extremal value which is a minimum, and furthermore, it is clear from (14) and (15) that there is a unique $\mathbf{h}_1$ that achieves this minimal point. In other words, we are minimizing $E(\mathbf{h}_1, \mathbf{h}_2)$ with respect to $\mathbf{h}_1$, whereas $\mathbf{h}_2$ is held constant at the value $\mathbf{h}_2^{(k)}$. Thus, the result of the first half of the iteration is to compute

$$\mathbf{h}_1^{(k+1)} = \arg\min_{h_1} E(\mathbf{h}_1, \mathbf{h}_2^{(k)}). \tag{35}$$

In going from $\mathbf{h}_1^{(k)}$ to $\mathbf{h}_1^{(k+1)}$, we have reduced the error from $E(\mathbf{h}_1^{(k)}, \mathbf{h}_2^{(k)})$ to $E(\mathbf{h}_1^{(k+1)}, \mathbf{h}_2^{(k)})$. We use the word "reduced" loosely here since it is possible that the error does not decrease even though it cannot increase. Similarly, during the second half of iteration $k+1$, we reduce the error by altering $\mathbf{h}_2$, whereas $\mathbf{h}_1$ is held constant at the value $\mathbf{h}_1^{(k+1)}$. The end result of the second half of the iteration is therefore

$$\mathbf{h}_2^{(k+1)} = \arg\min_{h_2} E(\mathbf{h}_1^{(k+1)}, \mathbf{h}_2). \tag{36}$$

As we noted above, in each half of the iteration, the error either remains the same or decreases. Therefore

$$\begin{aligned} E^{(k+1)} &= E(\mathbf{h}_1^{(k+1)}, \mathbf{h}_2^{(k+1)}) \\ &\le E(\mathbf{h}_1^{(k+1)}, \mathbf{h}_2^{(k)}) \\ &\le E(\mathbf{h}_1^{(k)}, \mathbf{h}_2^{(k)}) \\ &= E^{(k)}. \end{aligned} \tag{37}$$

From (37), the expected value of error variance is monotonically nonincreasing. In practice (see the examples below), geometric convergence to the optimal filter has been observed, as is discussed in the next section.

V. Performance Evaluation

The performance of the filters designed by these optimal separable filter design algorithms was evaluated by comparing them with the optimal unconstrained (nonseparable) Wiener filters and to the separable approximation of the 2-D Wiener filter with similar regions of support. Several 2-D random signal recovery problems and image recovery problems were used to carry out this comparison.

The first comparison was carried out on 2-D random signals. In order to best determine the limitations of separable filtering, a signal with a diagonally directional power spectrum (which is very difficult to approximate in a separable fashion) was chosen as the undegraded signal to be recovered. This signal was corrupted with white Gaussian noise of various power levels.The autocorrelation functions and power spectra for the case where the SNR was 10 dB are shown in Fig. 4.

The optimal separable and unconstrained 11×11 filters were computed. We found that the expected mean square error with the separable filter is about 50% more than it would be with the unconstrained filter. Fig. 5 shows that the convergence of the separable filter design algorithm is almost

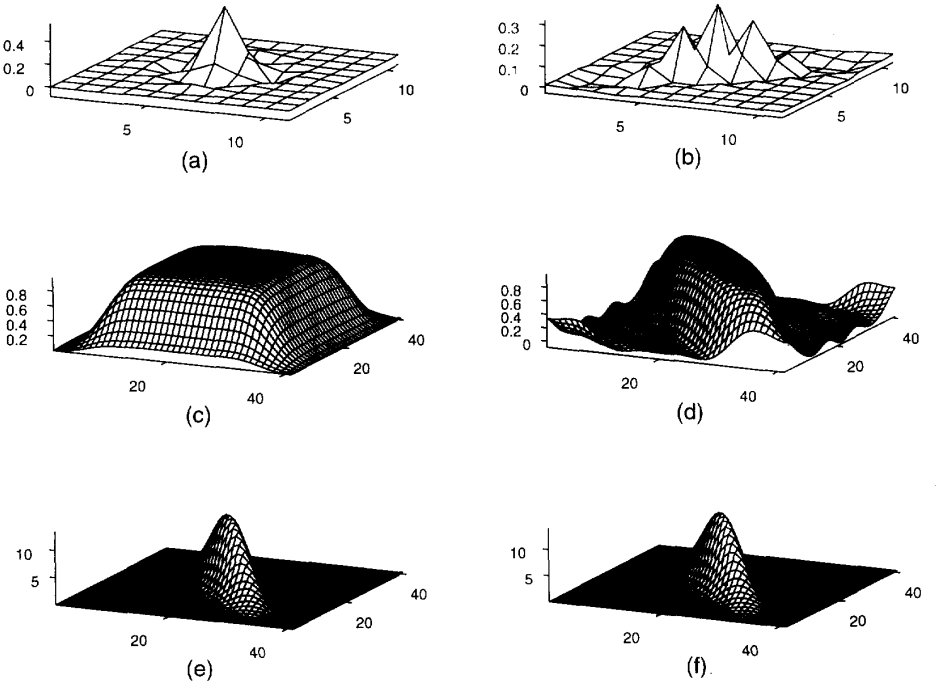


Fig. 6. (a) Optimal 11×11 separable filter; (b) optimal 11×11 nonseparable filter. Squared frequency response of (c) optimal separable filter; (d) optimal nonseparable filter. Power spectra of signal recovered by (e) optimal separable filter; (f) optimal nonseparable filter.

exactly geometric (until the precision of the computer prevents any further progress). The optimal separable and unconstrained filters and their frequency responses are shown in Fig. 6. The power spectra of the recovered signals are also shown. Note that at the frequencies where the uncorrupted signal has a high power level, the gain of the separable filter is somewhat less than that of the unconstrained filter.

Next, several images were degraded by adding a white Gaussian noise at an SNR of 12 dB. The optimal separable 11×11 filter was computed by the iterative spatial domain algorithm, and the optimal nonseparable 11×11 filter was computed by using the autocorrelation to set up a block-Toeplitz linear system in the filter coefficients. The autocorrelation function $R_f(n_1, n_2)$ used in these filter computations was obtained by averaging the deterministics autocorrelation functions of several "typical" images. We note here that only five iterations were required to compute the optimal filter to within six decimal places of accuracy. This convergence is demonstrated by the graph in Fig. 7. The graph shows the expected mean square error in decibels for every half iteration.

The results of image restoration are shown in Fig. 8. In Fig. 8(a), we have the original SHUTTLE image of 512×512 pixels. The noise-degraded image is shown in Fig. 8(b). The result of applying the optimal unconstrained filter to the image in Fig. 8(b) is shown in Fig. 8(c). Similarly, the image in Fig. 8(d) is the results of applying the optimal separable filter to the image in Fig. 8(b). Note that although there are noticeable differences between the two recovered images, the amount of noise reduction appears to be roughly the same in both. In fact, the residual error between the original SHUTTLE

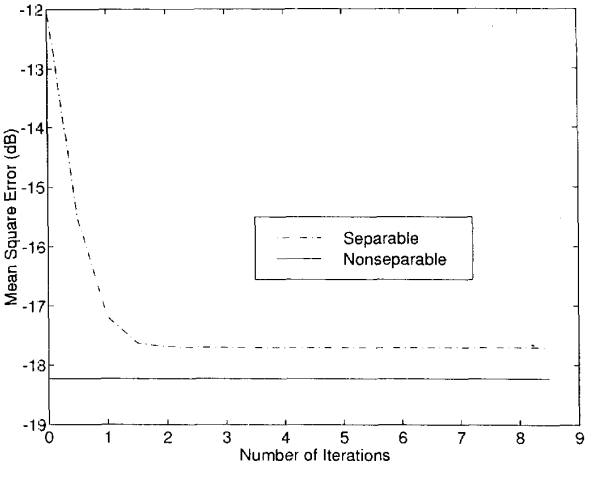


Fig. 7. Convergence of the iterative separable filter design algorithm for the case of an image degraded by white noise.

image and the image recovered by the separable filter is -17.73 dB, which is less than 0.5 dB higher than the residual error of -18.23 dB for the image recovered with the optimal unconstrained filter.

In the next example, a nonwhite Gaussian noise term was added to each image. Specifically, the noise was formed by convolving white Gaussian noise with the impulse response $\delta(n_1, n_2) + \delta(n_1 - 3, n_2 - 1)$ so that the variance of the resulting nonwhite noise would be 12 dB down from the mean square pixel value of the image. As in the white noise example,

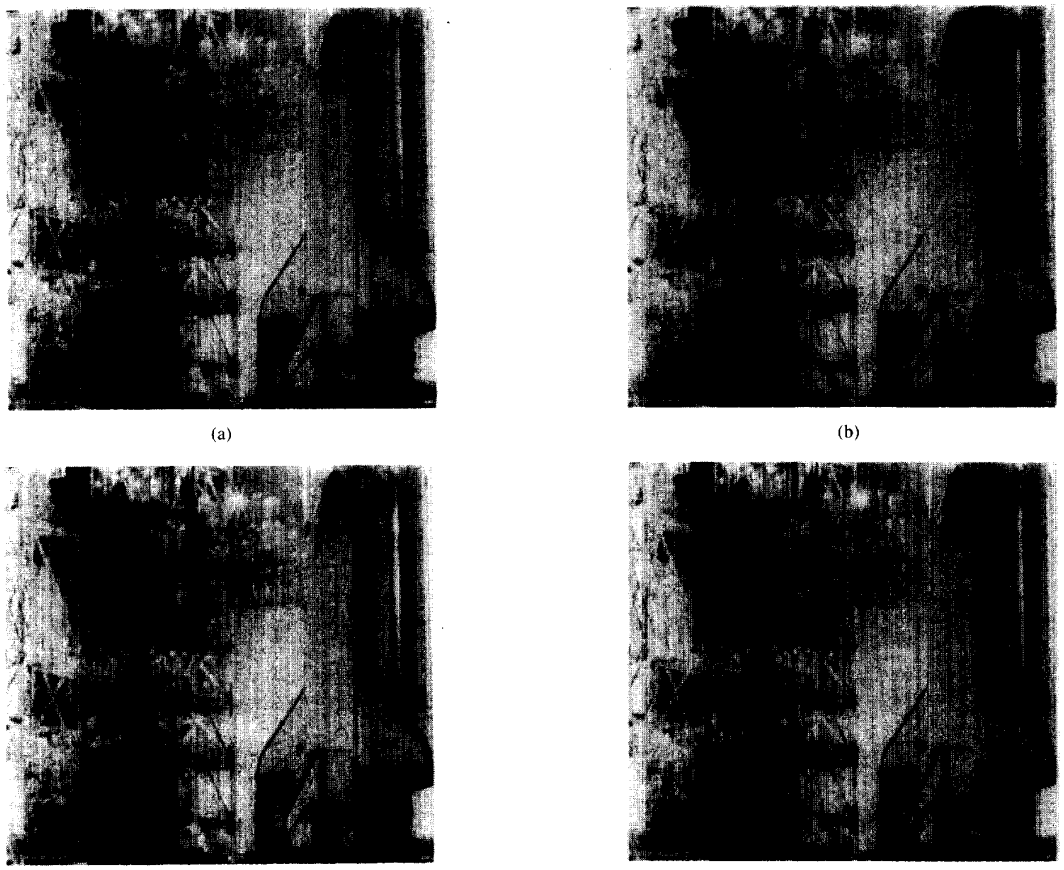


Fig. 8. (a) Original SHUTTLE image (512 × 512 pixels); (b) image degraded by additive white Gaussian noise (SNR = 12 dB). Images recovered by (c) optimal nonseparable filter; (d) optimal separable filter.

the optimal separable and nonseparable 11 × 11 filters were computed using the averaged "typical" image autocorrelation function. Again, as can be seen in Fig. 9, convergence was extremely rapid.

The results can be seen in Fig. 10. The original SHUTTLE image is shown in Fig. 10(a), and the corresponding noise-degraded image is shown in Fig. 10(b). In Fig. 10(c), we see the result of applying the optimal nonseparable filter to the degraded image in Fig. 10(b). The MSE between the recovered image in Fig. 10(c) and the original is -18.67 dB. This number is only 0.8 dB less than the MSE of -17.89 dB between the original image and the one resulting from the application of the optimal separable filter to the degraded image, which is shown in Fig. 10(d). Again, the level of noise reduction appears to be similar, although the unconstrained nonseparable filter does appear to do slightly better in areas with a high degree of detail or sharp edges.

Computing the SVD of the ideal Wiener filter allows us to decompose the ideal Wiener filter into the sum of several separable filters. It was found that one separable approximation (FIR filter) to the unconstrained Wiener filter was not sufficient, although this separable approximation was

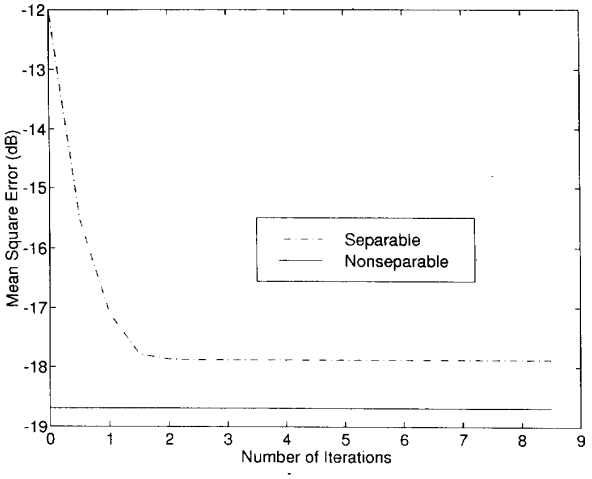
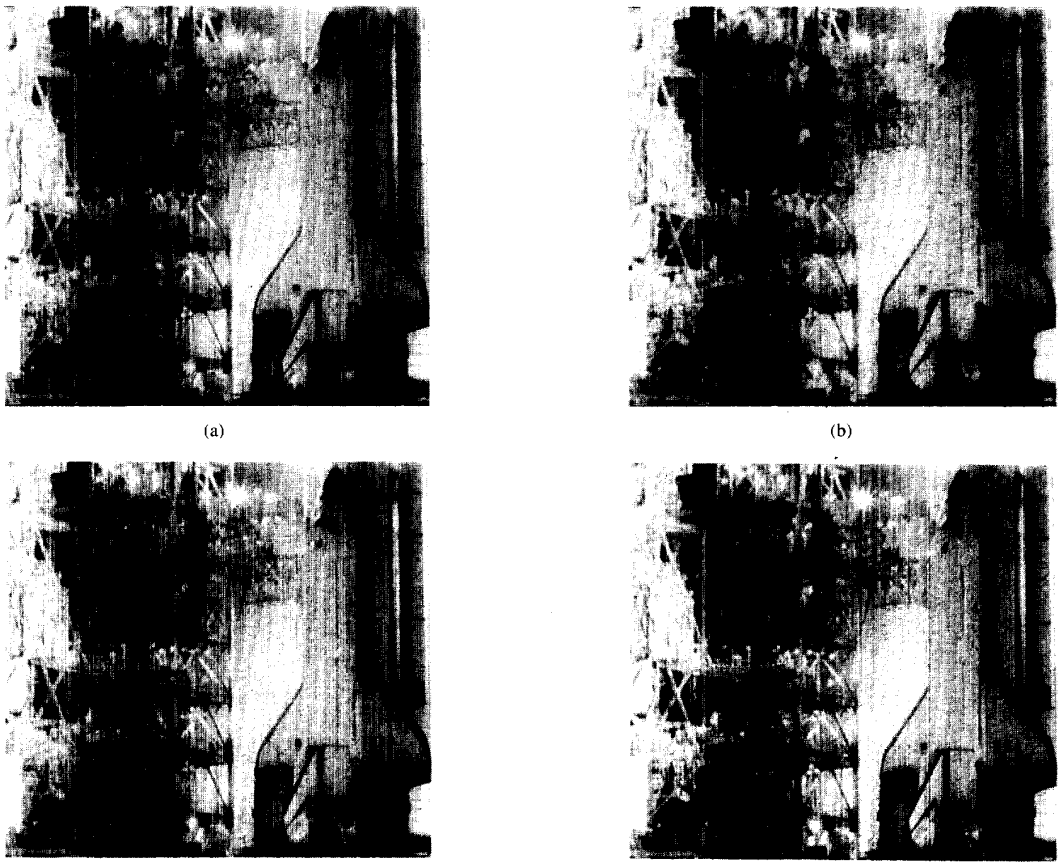


Fig. 9. Convergence of the iterative separable filter design algorithm for the case of an image degraded by nonwhite noise.

the best in the least square sense. Typically, it would require the application of at least two separable filters to achieve the level of performance of the unconstrained Wiener filter or



(a) (b) (c) (d)

Fig. 10. (a) Original SHUTTLE image (512 × 512 pixels); (b) image degraded by additive nonwhite Gaussian noise (SNR = 12 dB). Images recovered by (c) optimal nonseparable filter; (d) optimal separable filter.

TABLE I
MEAN SQUARE ERROR AFTER SUCCESSIVE SEPARABLE FILTER

Number of Separable	MSE (dB)
1	-12.43528
2	-17.84063
3	-17.84234
4	-17.89286
11	-18.22923

the optimal separable filter. This is shown in Table I. The table shows the mean square error after successive separable approximation is applied to the SHUTTLE image degraded by additive white Gaussian noise at an SNR of 12 dB.

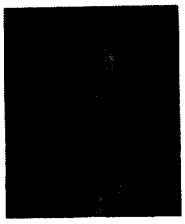
VI. CONCLUSION

We have derived the system of nonlinear equations that must be solved in order to determine the coefficients of the optimal separable finite impulse response filter. This nonlinear system may be solved by an iterative algorithm in either the spatial or Fourier domain. In both domains, the mean square error was proven to be nonincreasing. Finally, by applying the resulting separable filters to several typical noise reduction problems, we have shown that the use of separable filters drastically reduces the computational requirements while maintaining a level of performance similar to that of the unconstrained nonseparable filters.

REFERENCES

[1] H. C. Andrews and C. L. Patterson, "Outer product expansions and their uses in digital image processing," *Amer. Math. Monthly*, pp. 1–13, Jan. 1975.
[2] ——, "Singular value decomposition image coding," *IEEE Trans. Commun.*, pp. 425–432, Apr. 1976.
[3] A. Antoniou and W.-S. Lu, "Design of 2-D digital filters by using the singular value decomposition," *IEEE Trans. Circ. Syst.*, vol. CAS-34, no. 10, pp. 1191–1198, Oct. 1987.
[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*,. Baltimore: Johns Hopkins University Press, 1983, ch. 5.7.
[5] T. S. Huang, W. F. Schreiber, and O. J. Tretiak, "Image processing," *Proc. IEEE*, vol. 59. pp. 1586–1609, Nov. 1971.
[6] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[7] J. S. Lim, *2-D Signal and Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[8] S. Treitel and J. L. Shanks, "The design of multistage separable planar filters," *IEEE Trans. Geosci. Electron.*, vol. GE-9, pp. 10–27, Jan. 1971.
[9] H. L. Van Trees, *Detection, Estimation, and Modulation Theory Part I: Detection, Estimation, and Linear Modulation Theory.* New York: Wiley, 1968.
[10] P. A. Wintz, "Transform picture coding," *Proc. IEEE*, vol. 60, no. 7, pp. 809–820, July 1972.

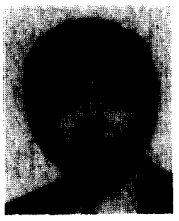


Chang D. Yoo received the B.S. degree from the California Institue of Technology, Pasadena, in 1986 and the M.S. degree from Cornell University, Ithaca, NY, in 1988.

He is currently pursuing the Ph.D. degree at the Massachusetts Institute of Technology, Cambridge, in the area of speech processing.

Mr. Yoo is a member of Tau Beta Pi.

Matthew M. Bace, photograph and biography not available at time of publication.



Jae S. Lim (F'86) received the S.B., S.M., E.E., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1974, 1975, 1978, and 1978, respectively.

He joined the faculty at MIT in 1978 as an Assistant Professor. He is currently a Professor in the Department of Electrical Engineering and Computer Science and is Director of the Advanced Telecommunications Research Program. His research interests include digital signal processing and its applications to image, video, and speech processing. He has contributed more than 100 articles to journals and conference proceedings. He is the author of a reprint book entitled *Speech Enhancement* (Prentice-Hall, 1983) and a coeditor of the reference book *Advanced Topics in Signal Processing* (Prentice-Hall, 1987). He has also authored the textbook entitled *Two-Dimensional Signal and Image Processing* (Prentice-Hall, 1990).

Dr. Lim has won three prize paper awards: one from the Boston Chapter of the Acoustical Society of America in 1976 and two from the IEEE Acoustics, Speech, and Signal Processing Society, in 1979 and 1985. He received the 1984 MIT Graduate Student Council's EECS Department Teaching Award. He was also corecipient of the 1984 Harold E. Edgerton Faculty Achievement Award.