A Class of Iterative Signal Restoration Algorithms

AGGELOS K. KATSAGGELOS, MEMBER, IEEE, AND SERAFIM N. EFSTRATIADIS, STUDENT MEMBER, IEEE

Abstract—**In this paper, a class of iterative signal restoration algorithms is derived based on a representation theorem for the generalized inverse of a matrix. These algorithms exhibit a first or higher order of convergence, and some of them consist of an on-line and an off-line computational part. The conditions for convergence, the rate of convergence of these algorithms, and the computational load required to achieve the same restoration results are derived. A new iterative algorithm is also presented which exhibits a higher rate of convergence than the standard quadratic algorithm with no extra computational load. These algorithms can be applied to the restoration of signals of any dimensionality. Iterative restoration algorithms that have appeared in the literature represent special cases of the class of algorithms described here. Therefore, the approach presented here unifies a large number of iterative restoration algorithms. Furthermore, based on the convergence properties of these algorithms, combined algorithms are proposed that incorporate *a priori* knowledge about the solution in the form of constraints and converge faster than the previously used algorithms.**

I. Introduction

THE recovery or restoration of a signal that has been distorted is one of the most important problems in signal processing applications [1], [18]. More specifically, the following degradation model is considered:

$$y = Dx, \tag{1}$$

where the vectors y and x represent, respectively, lexicographically ordered blurred and original signals. The matrix D represents a linear deterministic distortion which may be space varying or space invariant. When y and x represent images, then the distortion may be due to motion between the camera and the scene or due to atmospheric turbulence. The signal restoration problem is then to invert (1) or to find a signal as close as possible to the original one, subject to a suitable optimality criterion given y and D. Equation (1) also represents the more general degradation model where an additive noise term is considered. In this case, the restoration problem takes again the form of solving (1) for x, where D is replaced by a square well-conditioned matrix and y by $D^T y$, where ^{T} denotes the transpose of a matrix or vector. This case will be separately studied in Section III, since computationally simpler algorithms can be used.

Iterative algorithms are used in our work in solving the signal restoration problem. Iterative restoration algorithms have a number of advantages over direct or recursive restoration techniques, and they have been used extensively in the literature [18]. Most of these algorithms have a linear or first-order convergence rate. Singh *et al.* [19] derived an iterative restoration algorithm with a quadratic rate of convergence, when the matrix D in (1) is invertible. Morris *et al.* [14]–[16] and Lagendijk *et al.* [13] generalized this algorithm for higher orders of convergence. In their derivation, the matrix D in (1) was invertible. In [14]–[16] it was further assumed that D represents a convolution operator.

In this paper, we extend the results in [13]–[16] and [19] by showing that when D is singular, the higher order algorithms converge to the minimum norm solution of (1), provided that a solution exists. This is a very important result because for a large number of distortions of practical interest (motion, out-of-focus), the matrix D is singular. Furthermore, we derive iterative algorithms with linear and higher order convergence rates for the general case when D in (1) is a rectangular matrix. In this case, the limiting solution of these algorithms is the minimum norm least-squares (MNLS) solution of (1). The derivation of these algorithms is based on a representation theorem for the generalized inverse D^+ of the matrix D. Iterative restoration algorithms benefit a great deal from the use of constraints which incorporate properties of the solution into the restoration process. However, the direct use of constraints with the higher order algorithms may result in divergence or meaningless results. We propose techniques which allow us to effectively use constraints with a combination of linear and higher order iterative algorithms.

The derivation of the linear and higher order algorithms obtaining the MNLS solution of (1) is presented in Section II. Computationally simpler higher order algorithms solving for the minimum norm solution of (1), when D is a square, positive semidefinite matrix, are presented in Section III. Such a situation may result, for example, when a noise term is added to (1). Then, after regularization, the restoration problem is again the solution of a set of linear equations analogous to (1), where D and y are replaced by another matrix A and a vector b, respectively. These algorithms extend the results reported in [13]–[16] and [19]. In Section IV, the algorithms are compared with respect to their computational load. The incorporation of constraints are discussed in Section V, and a number of experimental results are presented in Section VI. Finally, conclusions are presented in Section VII.

Manuscript received August 2, 1988; revised June 16, 1989. This work was supported in part by the National Science Foundation under Grant MIP-8614217.

The authors are with the Department of Electrical Engineering and Computer Science, Northwestern University, The Technological Institute, Evanston, IL 60208-3118.

IEEE Log Number 9034417.

II. Minimum Norm Least-Squares Solution

In this section we assume that the matrix D in (1) is an $m \times n$ matrix, where $m \geq n$. That is, $D \in L(\boldsymbol{R}^n, \boldsymbol{R}^m)$, $x \in \boldsymbol{R}^n$ and $y \in \boldsymbol{R}^m$, where $L(\boldsymbol{R}^n, \boldsymbol{R}^m)$ is the set of matrices that map $\boldsymbol{R}^n$ into $\boldsymbol{R}^m$, the n-dimensional and m-dimensional Euclidean spaces, respectively. Let $\mathcal{R}(D)$ and $\mathcal{N}(D)$ denote, respectively, the range and the null space of D and let $\dim(S)$ denote the dimensionality of the subspace S [20]. If $\dim(\mathcal{R}(D)) = r$, then since $\dim(\mathcal{R}(D^T)) = r$, we get that $\dim(\mathcal{N}(D)) = n - r$ and $\dim(\mathcal{N}(D^T)) = m - r$. Equation (1) has at most one solution if and only if $r = n$, and we get no solution if $y \in \mathcal{N}(D^T)$. The degradation model of (1) can be modified so that D is a square matrix ($m = n$), by increasing the size of x, by adding zeros, or by reducing the size of y. Even in this case, however, for a large number of common distortions (motion, out-of-focus), the distortion matrix is singular, that is, $r < n$. Since in both cases (square and rectangular D) it cannot be guaranteed that $y \in \mathcal{R}(D)$, a least-squares (LS) solution is sought (the case when D is square and $y \in \mathcal{R}(D)$ will be studied in Section III). Such a solution minimizes the Euclidean norm $\| Dx - y \|$. The LS solution satisfies the normal equations

$$D^TDx = D^Ty. \tag{2}$$

The set of x's that satisfy (2) forms a closed convex set which contains a unique vector of minimum norm [5]. Then the generalized inverse $D^+ \in L(\boldsymbol{R}^m, \boldsymbol{R}^n)$ is defined by $D^+y = x^+$, where x^+ is the minimum norm least-squares (MNLS) solution of (1). A general theorem representing D^+ as the limit of a sequence of matrices, presented in Groetch [5], is presented next without proof, due to its significance.

A. Representation of D^+ and x^+

Theorem 1: Suppose $D \in L(\boldsymbol{R}^n, \boldsymbol{R}^m)$ and let $D^* = D^TD/\mathcal{R}(D^T)$. If Ω is an open set with $\sigma(D^*) \subset \Omega \subset (0, \infty)$ and $\{f_k(z)\}$ is a family of continuous real valued functions on Ω with $\lim_k f_k(z) = z^{-1}$ uniformly on $\sigma(D^*)$, then

$$D^+ = \lim_k f_k(D^*)D^T, \tag{3}$$

where the convergence is in the uniform topology for $L(\boldsymbol{R}^m, \boldsymbol{R}^n)$. Furthermore,

$$\left\| f_k(D^*)D^T - D^+ \right\| \leq \sup\left\{ \left| 1 - zf_k(z) \right| \right\} \cdot \left\| D^+ \right\|, \tag{4}$$

where the supremum is taken over all $z \in \sigma(D^*)$. □

Some of the notation used above is as follows. The spectral radius of a square matrix T and the restriction of T to a subspace S of $\boldsymbol{R}^n$ are, respectively, denoted by $\sigma(T)$ and T/S [20]. Clearly, the matrix D^* is symmetric and positive definite. Therefore, its spectral radius is a subset of the set $(0, \infty)$.

Theorem 1 is very powerful because it provides us with a general expression (3) for representing and iteratively computing the generalized inverse of a matrix. Furthermore, it provides us with a measure of the rate of convergence (4). Therefore, it unifies a large number of iterative restoration techniques. Any family of functions $\{f_k\}$ with the properties stated by Theorem 1 can result in a new representation for D^+. Clearly, some of these families of functions result in more attractive representations, from a computational point of view. It is noted here that Theorem 1 holds not only for matrices but for any linear operator with a closed range [5].

In signal restoration we are primarily interested in solving for x^+. Expressions for x^+ instead of D^+ are derived as follows. The convergence of the sequence $\{f_k(D^*)D^T\}$ to D^T is in the uniform topology for $L(\boldsymbol{R}^m, \boldsymbol{R}^n)$, which means that [5]

$$\lim_k \left\| f_k(D^*)D^T - D^+ \right\| = 0. \tag{5}$$

The uniform convergence of $\{f_k(D^*)D^T\}$ to D^+ implies strong convergence of $\{f_k(D^*)D^T\}$ to D^+, which means that for each $y \in \boldsymbol{R}^m$

$$\lim_k \left\| f_k(D^*)D^Ty - D^+y \right\| = 0. \tag{6}$$

Therefore, (3) and (4) are written, respectively, as [3], [10]

$$x^+ = \lim_k f_k(D^*)D^T \tag{7}$$

and

$$\left\| f_k(D^*)D^Ty - x^+ \right\| \leq \sup\left\{ \left| 1 - zf_k(z) \right| \right\} \cdot \left\| x^+ \right\|, \tag{8}$$

where the supremum is again over all $z \in \sigma(D^*)$. Therefore, Theorem 1 can be restated with (7) and (8) replacing (3) and (4), respectively. In the following section, different iterative restoration algorithms will be derived, corresponding to different choices of $\{f_k(z)\}$, by using (7) and (8).

B. A Linear Algorithm

Consider the sequence of functions $\{f_k(z)\}$ defined by

$$f_0(z) = \beta > 0$$

$$f_k(z) = f_0(z) \sum_{i=0}^{k} (1 - zf_0(z))^i. \tag{9}$$

It is shown [3], [5] that $\lim_{k \to \infty} f_k(z) = z^{-1}$ uniformly on compact subsets of the set

$$\Omega_\beta = \{z : |1 - \beta z| < 1\} = \left\{ z : 0 < z < \frac{2}{\beta} \right\}. \tag{10}$$

According to Theorem 1 and (9), by setting $x_k = f_k(D^*)D^Ty$, we get the iteration

$$x_0 = \beta D^Ty$$

$$\begin{aligned} x_{k+1} &= x_k + \beta D^T(y - Dx_k) \\ &= (I - \beta D^TD)x_k + \beta D^Ty, \end{aligned} \tag{11}$$

which converges to x^+ for

$$0 < \beta < 2 \cdot \|D\|^{-2}. \tag{12}$$

Iteration (11) also results from a successive approximations approach to the solution of the normal equations (2). It has been studied and used extensively for signal restoration [7], [18]. According to (8), the rate of convergence of iteration (11) is linear and it is characterized by the relation

$$\frac{\|x_k - x^+\|}{\|x^+\|} \leq c^{k+1}, \tag{13}$$

where [5]

$$c = \max\left\{\left|1 - \beta \cdot \|D\|^2\right|, \left|1 - \beta \cdot \|D^+\|^{-2}\right|\right\}. \tag{14}$$

An equivalent way of describing the linear rate of convergence of iteration (11) is with the use of the residual error at step k of iteration (9) [3]. It is defined as

$$r_k = 1 - zf_k(z) \tag{15}$$

and it represents the residual error associated with each eigenvalue of D^*, since $z \in \sigma(D^*)$. Then, according to iteration (9),

$$r_{k+1} = r_0 r_k. \tag{16}$$

Equation (16) represents a straight line on the $r_k r_{k+1}$-plane.

C. *Higher Order Algorithms*

Consider the sequence of functions $\{f_k(z)\}$ for an integer $p \geq 2$

$$f_0(z) = \beta > 0,$$

$$f_{k+1}(z) = f_k(z) \sum_{i=0}^{p-1} (1 - zf_k(z))^i. \tag{17}$$

The sequence defined by (17) converges uniformly to z^{-1} on compact subsets of Ω_β (10) [3], [5]. Application of Theorem 1 results in the algorithm [3], [5], [8]–[10]

$$D_0 = \beta D^T D, \qquad x_0 = \beta D^T y, \tag{18a}$$

$$\Phi_k = \sum_{i=0}^{p-1} (I - D_k)^i, \tag{18b}$$

$$D_{k+1} = \Phi_k D_k, \qquad x_{k+1} = \Phi_k x_k. \tag{18c}$$

An advantage of iteration (18) is that the matrix sequence $\{\Phi_k\}$ or $\{D_k\}$ can be computed in advance or *off-line*, although for a general D this may result in excessive storage. The solution sequence $\{x_k\}$ is then computed *on-line* after the distorted data y are available. As observed from (18), the limit of D_k is the projection onto the row space of D. This projection is equal to the identity matrix when D is invertible. That is, the distortion matrix is also updated. This means that if x_k is interpreted as the observed distorted signal at each iteration, then the distortion operator, which maps the original signal into x_k, is approaching the identity operator (if the inverse exists) as the iteration number increases.

Algorithm (18) exhibits pth order of convergence. That is, according to relation (8) [3], [6],

$$\frac{\|x_k - x^+\|}{\|x^+\|} \leq c^{p^k}, \tag{19}$$

where the convergence factor c is given by (14). Equivalently, it is easily shown that [3]

$$r_{k+1} = r_k^p, \tag{20}$$

where r_k is defined by (15). Equation (20) represents a pth-order curve on the $r_k r_{k+1}$-plane. Certain of these curves for $p = 2, 3, 4, 9, 20$ are shown in Fig. 1. The curve representing the rate of convergence of the linear algorithm (16) is also shown. Clearly, as p increases, the residual error for most of its values tends to go to zero in one iteration. Notice that the values -1 and 1 are excluded from the range of values that r_k takes.

D. *A New Iterative Algorithm*

Let us regard z^{-1} as the root of the function $f(u) = (u^{-1} - z)^\eta$, where $\eta > 0$. If the Newton–Raphson method is used in approximating this root, then the sequence $\{u_k\}$ is generated according to [3]

$$u_{k+1} = u_k \left[1 + \frac{1}{\eta}(1 - zu_k)\right], \tag{21}$$

for a suitable u_0. Suppose that for $\beta > 0$, a sequence of functions $\{f_k(z)\}$ is defined by

$$f_0(z) = \beta$$

$$f_{k+1}(z) = f_k(z)\left[1 + \frac{1}{\eta}(1 - zf_k(z))\right]. \tag{22}$$

The convergence and the rate of convergence of this algorithm can be described by considering r_k defined by (15). That is, it is found in a straightforward way that

$$r_{k+1} = \frac{1}{\eta} r_k(r_k + \eta - 1). \tag{23}$$

Note that for $\eta = 1$, this algorithm becomes the quadratic algorithm ($p = 2$) of (17). The curves described by (23) for different values of η are shown in Fig. 2. The lines $r_{k+1} = r_k$ and $r_{k+1} = -r_k$, also shown in this figure, are dividing the space into the regions I and II, defined, respectively, by $|r_{k+1}| < |r_k|$ and $|r_{k+1}| \geq |r_k|$. Clearly, if part of the curve represented by (23) for a certain η lies in region I, and if $|r_0| < 1$, iteration (22) converges; otherwise, it may not converge. For example, for $\eta > 1$ and $|r_0| < 1$, iteration (22) converges to z^{-1} on compact subsets of Ω_β, although the convergence rate of the algorithm may be slower than that of the quadratic.

On the other hand, for $0.5 < \eta < 1$, the part of the curve (23) for which $r_k \leq -\eta$ lies in the region II. Therefore, we need to restrict the residuals to satisfy $r_k > -\eta$,

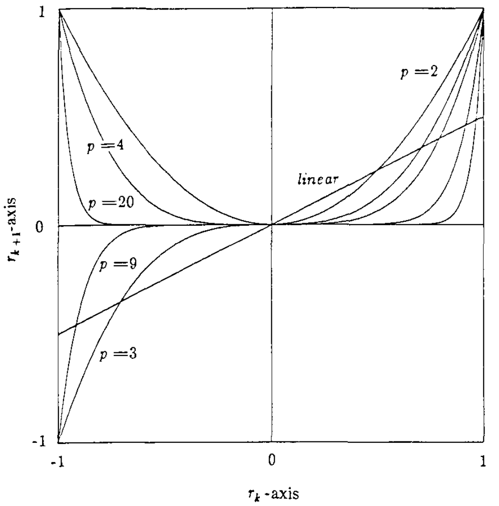


Fig. 1. Representation of the residual error of (16) and (20), respectively, for various values of the order p.

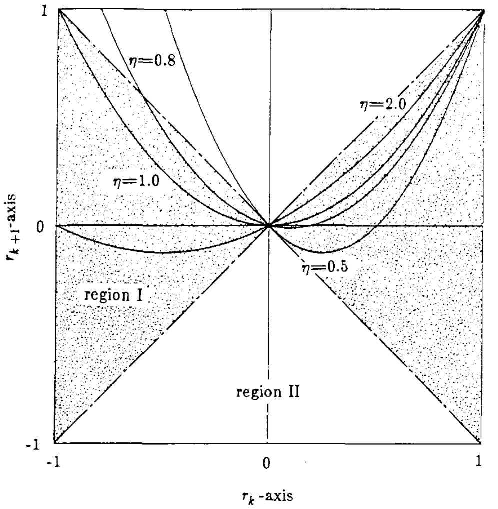


Fig. 2. Representation of the residual error of (23), for various values of η.

or $r_k \geq 0$. One way to accomplish this is by using $\eta = 1$ in evaluating $r_1 (k = 0)$ and then changing η to any value such that $0.5 < \eta < 1$. The rate of convergence of iteration (22) is pictorially represented by the slope of the curves shown in Fig. 2. For example, for $r_k \geq 0.2$, iteration (22) converges faster with $\eta = 0.8$ than with $\eta = 1.0$. With the conditions on η imposed according to Fig. 2 in mind, application of Theorem 1 results in the iteration [3]

$$D_0 = \beta D^T D, \qquad x_0 = \beta D^T y, \tag{24a}$$

$$\Phi_k = I + \frac{1}{\eta}(I - D_k), \tag{24b}$$

$$D_{k+1} = \Phi_k D_k, \qquad x_{k+1} = \Phi_k x_k, \tag{24c}$$

where $\eta = 1$ for $k = 0$ and $0.5 < \eta < 1$ for $k \geq 1$. In general, the rate of convergence of iteration (24) depends on the distribution of the eigenvalues of the matrix D^* defined by Theorem 1.

III. Minimum Norm Solution

In this section we consider the solution of

$$Ax = b, \tag{25}$$

where A is a square positive semidefinite matrix and $b \in \mathcal{R}(A)$. This is a case of special interest. Equation (25) may be the degradation model of (1), when, for example, $D = A$ represents the degradation due to atmospheric turbulence. Equation (25) may also result from the regularization of the ill-posed signal restoration problem. More specifically, the following degradation model is considered.

$$y = Dx + w, \tag{26}$$

where y and x represent, respectively, the lexicographically ordered distorted and original signals, and w denotes the additive noise. According to a regularization approach presented in [7] and [11], the solution of (26) is replaced by the solution of the well-conditioned system of equations

$$(D^T D + \alpha C^T C)x = D^T y. \tag{27}$$

The matrix C represents a high-pass filter and its role is to restrict the energy of the restored signal at high frequencies, due primarily to the amplified noise. The regularization parameter α is a function of the signal-to-noise ratio [7]. Therefore, the presence of additive noise in the degradation model does not alter the form of the iterative algorithms presented in Section II, since (1) is now replaced by (25).

Clearly, (25) can be solved by using any of the algorithms presented in Section II. A key difference, however, between (1) and (25) is that although matrix D is in general a rectangular matrix, matrix A is always square, positive definite, or positive semidefinite. Therefore, (25) might have a solution, which means that $b \in \mathcal{R}(A)$. As a matter of fact, the constraint C can be designed in such a way that $b \in \mathcal{R}(A)$ [7]. In this case, the minimum norm solution can be found with fewer computations than those required by the least-squares approach, as is shown next.

An iteration due to Bialy [2] with linear rate of convergence, suitable for finding the solution of (25), is presented by the following theorem.

Theorem 2: Let $A : \boldsymbol{R}^n \rightarrow \boldsymbol{R}^n$ be a positive semidefinite matrix. For $b \in \boldsymbol{R}^n$, $x_0 \in \boldsymbol{R}^n$ consider the iterative process

$$x_{k+1} = x_k + \beta(b - Ax_k), \tag{28}$$

where $0 < \beta < 2 \cdot \|A\|^{-1}$. Then, the sequence $\{x_k, k \geq 0\}$ converges to $x^* = \hat{x} + \boldsymbol{P}_{\mathfrak{N}(A)}\{x_0\}$, where $\hat{x}$ is the minimum norm solution of $Ax = b$ and $\boldsymbol{P}_{\mathfrak{N}(A)}\{x_0\}$ is the projection of x_0 onto the null space of A, if and only if $b \in \mathcal{R}(A)$. □

We can think of iterations (11) and (28) as forming a pair, since they both have a linear rate of convergence. Iteration (11) successively approximates the solution to the normal equations (2), while iteration (28) successively approximates the solution to (25). In extending the abovementioned correspondence between the linear algorithms (11) and (28) to the higher order algorithms, we present the following theorem [3], [8].

Theorem 3: Let $A : \boldsymbol{R}^n \rightarrow \boldsymbol{R}^n$ be a positive semidefinite matrix. For a given integer $p \geq 2$ and $\beta > 0$, consider the iterative process

$$A_0 = \beta A^T, \qquad x_0 = \beta b, \tag{29a}$$

$$\Phi_k = \sum_{i=0}^{p-1} (I - A_k)^i, \tag{29b}$$

$$A_{k+1} = \Phi_k A_k, \qquad x_{k+1} = \Phi_k x_k, \tag{29c}$$

where $0 < \beta < 2 \cdot \|A\|^{-1}$. Then the sequence $\{x_k, k \geq 0\}$ converges to $x^* = \hat{x}$, where $\hat{x}$ is the minimum norm solution of $Ax = b$, if and only if $b \in \mathcal{R}(A)$. □

The proof of Theorem 3 is presented in the Appendix. Algorithm (29) with $p = 2$ was proposed by Singh *et al.* [19] for the case that $\|I - A\| < 1$, and by Morris *et al.* [14] for the case that A is positive definite and represents a linear space invariant system (convolution case). Algorithm (29) for any $p \geq 2$ was proposed by Morris *et al.* [14], [15] and by Lagendijk *et al.* [13] for the case that A is positive definite. Therefore, Theorem 3 extends the previously reported results.

IV. Comparison Based on the Computational Load

The question we address in this section is the following. For a specific restoration problem, which of the iterative algorithms presented in Sections II (B, C, and D) and III should one use? We answer this question by considering the amount of computation required by each algorithm in obtaining the same solution point or in satisfying the same error criterion.

Clearly, algorithms (28) and (29), if applicable, should be used, since they require fewer computations than their counterparts, iterations (11) and (18), respectively. Additionally, iteration (24) should be used over iteration (18) for $p = 2$, if η is chosen according to the discussion in Section II-D, since the former requires the same number of computations as the latter, with the exception of an additional multiplication by the scalar $1/\eta$. Therefore, in the following, the algorithms of Section II-B and C will be compared. The same comparison holds true for the algorithms of Section III.

Iterative algorithms give the exact solution as $k \rightarrow \infty$, but in practice the iterative process is terminated after a finite number of iterations. Since the distortion operator is known, c in (14) is known, therefore, the number of iterations required by the algorithms to reach an approximate solution can be computed. More specifically, let us denote by k_1 and k_p the iteration steps of the first and pth-order algorithms, respectively. Let us also suppose that m_p iterations of the pth-order algorithm are run, that is, $k_p = 1, \cdots, m_p$. Then, according to (13) and (19), the k_pth iteration step of algorithm (18) is equivalent to $N(k_p)$ iterations of the linear algorithm, where

$$N(k_p) = p^{k_p} - p^{k_p - 1}. \tag{30}$$

That is, had the k_pth iteration step of algorithm (18) been replaced by $N(k_p)$ iteration steps of algorithm (11), the restoration results would have been the same. Now, the total number of iteration steps of algorithm (11) denoted by m_1, which are equivalent to m_p iteration steps of algorithm (18), are given by the expression

$$m_1 = \sum_{k_p=1}^{m_p} N(k_p) = p^{m_p} - 1. \tag{31}$$

According to (31), due to the exponential relation between m_1 and m_p, a tremendous number of iterations may be required by the linear algorithm in obtaining the same result with a higher order algorithm. For example, if $p = 5$ and $m_5 = 10$, then $m_1 = 9\,765\,624$. However, the relation between the computational load required by the linear and pth-order algorithm in running, respectively, m_1 and m_p iterations, is not exponential, as explained below.

In the general case, let us assume that matrix D has dimensions $m \times n$; then D^* is an $n \times n$ square matrix. Thus, the computational load for the linear algorithm after m_1 iterations is $\boldsymbol{M}_1 = mn^2 + (m_1 + 1)mn$ multiplies and $\boldsymbol{A}_1 = n^2(m - 1) + (m_1 + 1)mn$ additions, with a total of $\boldsymbol{C}_1 = n^2(2m - 1) + 2(m_1 + 1)mn$ operations. On the other hand, m_p iterations of the pth-order algorithm require $\boldsymbol{M}_p = nm + m_p[nm + (p - 1)n^2m]$ multiplies and $\boldsymbol{A}_p = n(m - 1) + m_p[n(m - n) + (p - 1)n^2m]$ additions, with a total of $\boldsymbol{C}_p = n(2m - 1) + m_p[n(2m - n) + 2(p - 1)n^2m]$ operations. The efficiency of the higher order algorithms over the linear depends on the order chosen, the dimensions m and n of the matrix D, and the number of iterations required. Table I shows the smallest number of iterations which the quadratic algorithm ($p = 2$) must run in order to be computationally more efficient than the linear algorithm, as a function of the dimensions of the matrix D. In this case, matrix D is considered to be square ($m = n$) and multiplies and additions are assumed to require the same amount of computation. According to Table I, although the required number of computations per iteration is greater for the higher order algorithms, the overall computational load is indeed less than that required by the linear algorithm, after a small number of iterations. The latter is due to the fact that the error for a given p decreases exponentially with a factor p, whereas the number of computations increases linearly with the same factor.

The computational savings with the use of the higher order algorithms increases when the distortion matrix D has a special form. For example, consider the common case when D is circulant. Then the algorithms are implemented using the Discrete Fourier Transform (DFT). For the linear algorithm, the number of computations after m_1

TABLE I
SMALLEST NUMBER OF ITERATIONS (m_2) FOR WHICH THE QUADRATIC ALGORITHM IS COMPUTATIONALLY MORE EFFICIENT THAN THE LINEAR ALGORITHM, AS A FUNCTION OF THE DIMENSION n OF THE SQUARE MATRIX D

n	m_2
2	3
3.4	4
5-7	5
8-11	6
12-20	7
21-35	8
36-63	9
64-113	10
114-204	11
. . .	. . .

TABLE II
COMPUTATION OF THE REQUIRED NUMBER OF ITERATIONS AND THE COMPUTATIONAL LOAD OF THE VARIOUS ALGORITHMS, GIVEN c AND ϵ (D CIRCULANT). THE MINIMUM C_p/N_F INDICATES THE OPTIMUM ORDER p_{opt}

$c = 0.9$	p	1	2	3	4	5	6	7	8	9	10
$\epsilon = 10^{-3}$	m_p	65	7	4	4	3	3	3	3	2	2
	C_p/N_F	133	22	21	29	28	34	40	46	35	39
$\epsilon = 10^{-6}$	m_p	131	8	5	4	4	3	3	3	3	3
	C_p/N_F	265	25	26	29	37	34	40	46	52	58

iterations is $\boldsymbol{M}_1 = (m_1 + 2)N_F$ complex multiplies and $\boldsymbol{A}_1 = (m_1 + 1)N_F$ complex additions, with a total of $\boldsymbol{C}_1 = (2m_1 + 3)N_F$ complex operations, where N_F is the extent of the DFT. For the pth-order algorithm, the number of computations after m_p iterations is $\boldsymbol{M}_p = (m_p p + 1)N_F$ complex multiplies and $\boldsymbol{A}_p = m_p(p - 1)N_F$ complex additions, with a total of $\boldsymbol{C}_p = [m_p(2p - 1) + 1]N_F$ complex operations. Clearly, since $\boldsymbol{C}_1$ and $\boldsymbol{C}_p$ depend linearly on m_1 and m_p, respectively, while the relation between m_1 and m_p is exponential, according to (31), $\boldsymbol{C}_p$ decreases relatively to $\boldsymbol{C}_1$, as the order p and iteration number m_p increase. For example, consider the case when $p = 2$ and $m_2 = 8$; then $\boldsymbol{C}_2 = 25N_F$. According to (31), the equivalent number of iterations for the linear algorithm is $m_1 = 255$ and $\boldsymbol{C}_1 = 513N_F$. If $p = 3$ and $m_3 = 8$, then $\boldsymbol{C}_3 = 41N_F$. In this case, the linear algorithm requires $m_1 = 6560$ and $\boldsymbol{C}_1 = 13\ 123N_F$ complex operations.

The analysis of the required computational load can be carried out from a different point of view, if we assume that an error threshold ϵ is determined in advance in terminating the iteration. Then, we are interested in finding the smallest m_1 or m_p, and of course that choice of the order p which minimizes the total number of computations. By using (13), m_1 is determined by $m_1 = \lceil \log(\epsilon/c)/\log(c) \rceil$, where $\lceil x \rceil$ is the smallest integer which is greater than or equal to x. For the higher order algorithms, m_p is given by

$$m_p = \left\lceil \frac{\log(\log(\epsilon)/\log(c))}{\log(p)} \right\rceil, \qquad (32)$$

and the optimum order p_{opt} minimizes C_p/N_F. Two examples with $c = 0.9$, are given in Table II. In the first example, $\epsilon = 10^{-3}$ and $p_{opt} = 3$, $m_3 = 4$, and $\boldsymbol{C}_3 = 21N_F$. In the second example, $\epsilon = 10^{-6}$ and $p_{opt} = 2$, $m_2 = 8$, and $\boldsymbol{C}_2 = 25N_F$. Note that in the last example, the linear algorithm would require $m_1 = 131$ iterations and $\boldsymbol{C}_1 = 265N_F$ complex operations in order to meet the same error criterion.

In conclusion, the computational load required by the pth-order algorithm is indeed smaller when compared to the computational load required by the linear algorithm. This statement is further amplified if the order p is a composite number. Then arithmetic computations are reduced dramatically, due to the decomposition of the pth-order algorithm into lower order algorithms, as was discussed by Morris *et al.* [16].

V. Combined Algorithms

An attractive feature of the linear iterative algorithms of (11) and (28) is the possibility of incorporating prior knowledge about the solution into the restoration process, in the form of constraints [18]. Among the different constraints, the nonlinear positivity constraint has been shown to be very powerful and useful [18]. However, according to our experimental evidence, when the positivity constraint is used with the higher order algorithms, it generally leads to erroneous results or causes divergence. The qualitative explanation we offer at this point is that this behavior is due to the decoupling of the computation of D_k from the computation of x_k in (18), (24), and (29). That is, there is no adjustment mechanism in the higher order algorithm as with the linear algorithm via the error term $(y - Dx_k)$ in (11) or the error term $(b - Ax_k)$ in (28). Therefore, the development of constrained higher order algorithms is an open research topic. A first step toward this direction is an iterative algorithm which makes use of both the linear and the pth-order algorithms along with the application of constraints, as discussed next for the algorithm in Section II [3], [10].

Let us denote by k_1 and k_p the iteration numbers of the first and pth order algorithms, respectively. According to (30) and (31), a combination of these algorithms can produce the same restoration results as each algorithm alone. More specifically, given a positive number ϵ, the required total number of iterations m_1 and m_p for algorithms (11) and (18), respectively, are determined as discussed in Section IV. If m_p is even(odd), then the pth order algorithm updates the solution only at its odd(even) iteration steps except at the last one, while its even(odd) iteration steps are replaced by $N(k_p)$ equivalent iterations of algorithm (13). (The opposite occurs for m_p odd.) The last iteration of the pth order algorithm is replaced by $K = m_1$

$- p^{m_p - 1}$ iterations of algorithm (13). For example, if m_p is even, then for $k_p = 1, 3, \cdots, m_p - 1$, we have $k_1 = 0$, while for $k_p = 2, 4, \cdots, m_p - 2$, we have $k_1 = 1, 2, \cdots, N(k_p)$, and for $k_p = m_p$, we have $k_1 = 1, 2, \cdots, K$. In general, if we denote by k the iteration number of the combined algorithm, then

$$k = \left\lceil \frac{k_p}{2} \right\rceil - \text{mod}\,(m_p, 2) + \sum_{l=1}^{k_p - 1} N(k_p) \cdot \text{mod}\,(l \cdot m_p, 2) + k_1, \qquad (33)$$

where mod $(i, 2)$ represents the modulo operation. When the combined algorithm is used, the proper deterministic constraint(s) can be imposed whenever algorithm (11) is applied. Note that, since after the incorporation of constraints (30) does not hold as is, the range of k_1 can be smaller than $N(k_p)$.

Adaptive regularized iterative image restoration algorithms have also appeared in the literature [6], [11], based on iterations (11) and (28). We have proposed a combined adaptive iterative algorithm based on iterations (18), (24), and (29) [4]. The same idea is used as the one described above. That is, one iteration of the pth-order algorithm (18), for example, is combined with $N(k_p)$ iterations (30) of the linear adaptive algorithm, in forming a combined adaptive iteration step.

VI. Experimental Results

Certain experimental results which demonstrate some of the basic ideas of the previous sections are described in this section. A synthetic signal of length 64 samples consisting of two impulses, $x(n) = \delta(n - 30) + \delta(n - 35)$, is used in our experiments. The simulated distortion is due to motion over 11 samples. The impulse response of such a distorting system is a rectangle, resulting in a singular matrix D. The normalized residual error [left-hand side of conditions (13) and (19)] is shown in Fig. 3, resulting respectively from the application of iterations (11) and (18) for different values of p. In our simulations, the value of x^+ was substituted by the available signal x_{or}. The normalized error is shown again in Fig. 4 with the application of the positivity constraint. The combined algorithm described in Section V for m_p even is implemented in this case for the higher order algorithms. It is observed in this case that the smaller the parameter p, the higher the convergence rate. This is due to the fact that the smaller the parameter p, the more often the higher order algorithm is applied. Due to this observation, the linear algorithms combined with the algorithm proposed in Section II-D is not shown in Fig. 4, since its performance is very similar with the performance of the quadratic algorithm.

Finally, the algorithm with quadratic convergence is compared to the algorithm proposed in Section II-D. The distortion is the same as before, while an image line is used as a test signal. The normalized error is shown in Fig. 5. The faster convergence of the new algorithm over the quadratic algorithm is obtained with no extra computational load.

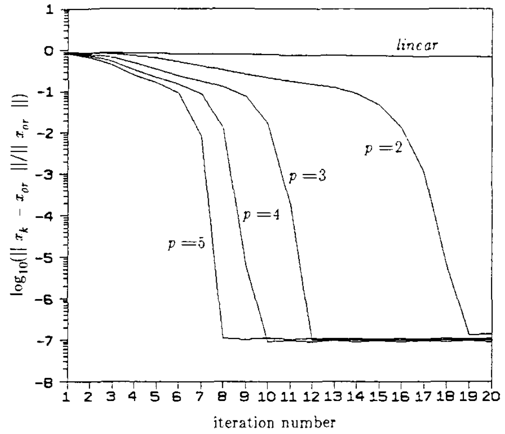


Fig. 3. Normalized residual error versus number of iteration for algorithms (11) and (18), for various values of p.

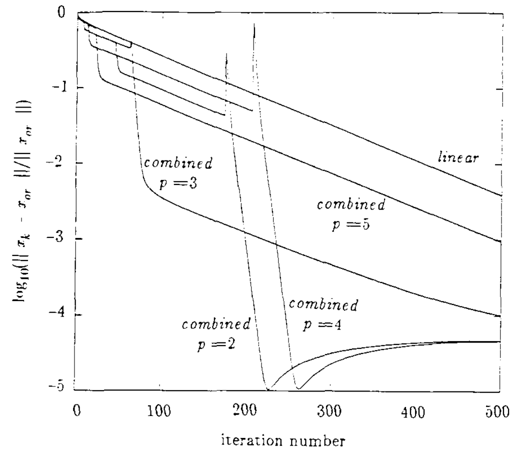


Fig. 4. Normalized residual error versus iteration number for the linear and combined algorithms of Section V, for various values of p, with the incorporation of the positivity constraint.

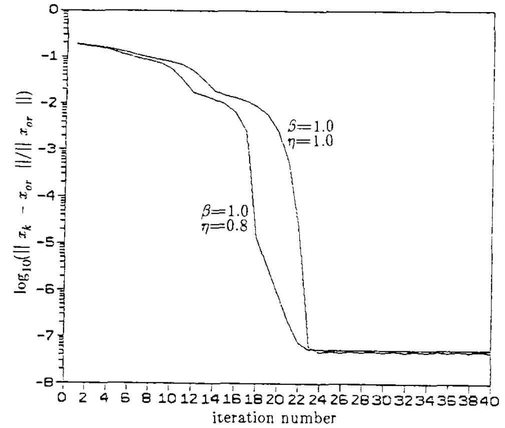


Fig. 5. Normalized residual error versus iteration number for algorithm (24) for various values of η ($\eta = 1.0$ corresponds to the quadratic algorithm).

VII. Discussion and Conclusions

A number of iterative signal restoration algorithms have been derived based on a representation theorem for the generalized inverse of a matrix. Some of these algorithms have appeared in the literature and some are new. An algorithm relating to the method of stochastic approximations can be also derived based on Theorem 1 [3], [5], [12]. Therefore, the approach followed here unifies the derivation of a large number of iterative restoration algorithms. These algorithms are applicable to the general case when additive noise is considered in the distortion model. The restoration approach is the same since the solution of (1) is replaced by the solution of (25). According to the analysis of Section IV, the application of the higher order algorithms is more advantageous due to the computational savings. In addition, due to the fact that they require a smaller number of iterations to converge, truncation or roundoff errors may be less pronounced.

One of the attractive properties of the linear restoration algorithms is the possibility of incorporating constraints in the iteration, which express *a priori* knowledge about the solution. Although the straightforward incorporation of constraints in the higher order algorithms results in undesirable results, we have proposed an algorithm which combines the constrained linear and the pth-order iterations. This combined algorithm converges faster than the constrained linear algorithm and with less overall computational load.

The algorithms presented can be used for the restoration of signals of any dimensionality as well as for the solution of any type of inverse problem which accepts the formulation of (1) or (25). The application of the algorithms to band-limited signal extrapolation is currently under investigation. Since the approach presented here in deriving iterative restoration algorithms is general, the use of other families of functions $f_k(z)$ which satisfy Theorem 1, and may lead to useful iterative restoration algorithms, is also currently under investigation.

Appendix
Proof of Theorem 3

Denote by λ_i, $i = 1, \cdots, n$ the eigenvalues of A. Since A is positive semidefinite, $\lambda_i > 0$ for $i = 1, \cdots, r$ and $\lambda_i = 0$ for $i = r + 1, \cdots, n$, where r is the rank of the matrix. Since A is symmetric, it has a complete set of orthonormal eigenvectors $u_1, \cdots, u_n$, where $(u_i, u_j) = \delta_{ij}$. That is, A can be written as

$$A = U\Lambda U^T = [u_1 \cdots u_n] \begin{bmatrix} \lambda_1 & & \varnothing \\ & \ddots & \\ \varnothing & & \lambda_n \end{bmatrix} \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}. \quad \text{(A-1)}$$

If we define

$$T = (I - \beta A) = U(I - \beta\Lambda)U^T, \quad \text{(A-2)}$$

and $T_k = I - A_k$, then the iterative algorithm (29) can be written as

$$T_0 = T, \quad x_t = \beta b,$$

$$T_{k+1} = T_k^p, \quad x_{k+1} = \left\{ \sum_{i=0}^{p-1} T_k^i \right\} x_k. \quad \text{(A-3)}$$

Solving for x_k, we obtain the following formula:

$$x_k = \left\{ \sum_{i=0}^{p^k-1} T^i \right\} \beta b, \quad \text{(A-4)}$$

or by using (A-2),

$$x_k = \beta \sum_{i=0}^{p^k-1} U(I - \beta\Lambda)^i U^T b. \quad \text{(A-5)}$$

Since A is symmetric, $\boldsymbol{R}^n = \mathfrak{N}(A) \oplus \mathfrak{R}(A^T) = \mathfrak{N}(A^T) \oplus R(A)$. The $(n - r)$ eigenvectors that correspond to the zero eigenvalues of A span $\mathfrak{N}(A)$ since

$$Au_i = \lambda_i u_i = 0, \quad \text{or} \quad u_i \in N(A)$$
$$\text{for } i = r + 1, \cdots, n$$
$$\text{and } (u_i, u_j) = \delta_{ij}$$
$$\text{for } i, j \in (r + 1, n).$$

Then $\mathfrak{R}(A) = \mathfrak{R}(A^T) = \text{span}\{u_1, \cdots, u_r\}$ and b can be written as

$$b = Uc \quad \text{or } U^T b = c, \quad \text{(A-6)}$$

where $c = [c_1, \cdots, c_n]^T$ is the coefficient column vector. From (A-5) and (A-6) we get

$$x_k = \beta \sum_{i=0}^{p^k-1} U(I - \beta\Lambda)^i c$$
$$= \beta \sum_{j=1}^{n} \left\{ c_j u_j \sum_{i=0}^{p^k-1} (1 - \beta\lambda_j)^i \right\}. \quad \text{(A-7)}$$

Since A is positive semidefinite, $0 < \lambda_{\max} \leq \|A\|$. In fact, $\lambda_{\max} = \|A\|_2$. It is assumed that

$$0 < \beta \cdot \|A\| < 2 \quad \text{or} \quad 0 < \beta\lambda_{\max} < 2$$
$$\text{or} \quad |1 - \beta\lambda_{\max}| < 1. \quad \text{(A-8)}$$

Therefore, since $\lambda_i > 0$ for $i = 1, \cdots, r$

$$|1 - \beta\lambda_i| < 1 \quad \text{(A-9)}$$

and

$$(1 - \beta\lambda_i)^{\nu(k)} \rightarrow 0 \quad \text{for } k \rightarrow \infty, \quad \text{(A-10)}$$

where ν is a positive, strictly increasing function of k such that $\nu(1) \geq 1$. Now, if $b \in \mathfrak{R}(A)$

$$c_{r+1} = \cdots = c_n = 0. \quad \text{(A-11)}$$

Finally, from (A-7) we get

$$x_k = \sum_{j=1}^{r} c_j \lambda_j^{-1} [1 - (1 - \beta\lambda_j)^{p^k}] u_j \quad \text{(A-12)}$$

and for $k \rightarrow \infty$, due to (A-10),

$$\hat{x} = x_\infty = \sum_{j=1}^{r} c_j \lambda_j^{-1} u_j, \quad \text{(A-13)}$$

where $\hat{x}$ is the minimum norm solution, since $A\hat{x} = b$ and the infinite set of solutions is equal to $x = \hat{x} + \bar{x}$, where $\bar{x} \in \mathfrak{N}(A)$. Now if $b \notin \mathfrak{R}(A)$, from (A-7) and (A-10), we get

$$x_k = \beta \sum_{j=1}^{r} \left\{ c_j u_j \sum_{i=0}^{p^k-1} (1 - \beta\lambda_j)^i \right\} + \beta p^k \sum_{j=r+1}^{n} c_j u_j \tag{A-14}$$

and for $k \rightarrow \infty$, $x_k \rightarrow \infty$ since at least one of the c_i, where $i = r + 1, \cdots, n$, is different from zero. Q.E.D.

References

[1] H. C. Andrews and B. R. Hunt, *Digital Image Restoration.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

[2] H. Bialy, "Iterative Behandlung Linearen Funktionalgleichungen," *Arch. Ration. Mech. Anal. 4*, pp. 166-176, July 1959.

[3] S. N. Efstratiadis, "Fast iterative signal restoration algorithms," M.S. thesis, Northwestern Univ., Dep. Elec. Eng., Comput. Sci., June 1988.

[4] S. N. Efstratiadis and A. K. Katsaggelos, "Fast adaptive iterative image restoration algorithms," in *Proc. SPIE Symp. Visual Commun. Image Processing '88*, Cambridge, MA, Nov. 1988, pp. 10-18.

[5] C. W. Groetch, *Generalized Inverses of Linear Operators.* New York: Marcel Dekker, 1977.

[6] A. K. Katsaggelos, J. Biemond, R. M. Mersereau, and R. W. Schafer, "Nonstationary iterative image restoration," in *Proc. 1985 Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 696-699.

[7] ——, "A general formulation of constrained iterative restoration algorithms," in *Proc. 1985 Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 700-703.

[8] A. K. Katsaggelos, "A unified approach to iterative image restoration," in *Proc. SPIE Symp. Visual Commun. Image Processing '87*, Cambridge, MA, Oct. 1987, pp. 163-167.

[9] A. K. Katsaggelos and S. N. Efstratiadis, "Fast iterative image restoration algorithms," in *Proc. 25th Annu. Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, Sept. 1987, pp. 493-502.

[10] ——, "A unified approach to iterative signal restoration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, Apr. 1988, pp. 1028-1031.

[11] A. K. Katsaggelos, "Iterative image restoration algorithms," *Opt. Eng.*, vol. 28, no. 7, pp. 735-748, July 1989.

[12] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* New York: Springer-Verlag, Appl. Math. Sci., vol. 26, 1978.

[13] R. L. Lagendijk, R. M. Mersereau, and J. Biemond, "On increasing the convergence rate of regularized image restoration algorithms," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Dallas, TX, Apr. 1987, pp. 28.2.1-4.

[14] C. E. Morris, M. A. Richards, and M. H. Hayes, "Iterative deconvolution algorithm with quadratic convergence," *J. Opt. Soc. Amer. A.*, vol. 4, no. 1, pp. 200-207, Jan. 1987.

[15] ——, "A generalized fast iterative deconvolution algorithm," in *Proc. 1987 Int. Conf. Acoust., Speech, Signal Processing*, Dallas, TX, Apr. 1987, pp. 36.1.1-36.1.4.

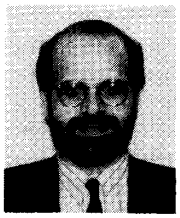
[16] ——, "Fast reconstruction of linearly distorted signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1017-1025, July 1988.

[17] A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*, 2nd ed. New York: Springer-Verlag, 1982.

[18] R. W. Schafer, R. M. Mersereau, and M. A. Richards, "Constrained iterative restoration algorithms," *Proc. IEEE*, vol. 60, pp. 432-450, Apr. 1981.

[19] S. Singh, S. N. Tandon, and H. M. Gupta, "An iterative restoration technique," *Signal Processing 11*, pp. 1-11, 1986.

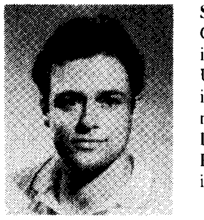
[20] G. Strang, *Linear Algebra and Its Applications*, 2nd ed. New York: Academic, 1980.



Aggelos K. Katsaggelos (S'80-M'85) was born in Arnea, Greece, on April 17, 1956. He received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1979. He received the M.S. and Ph.D. degrees, both in electrical engineering, from the Georgia Institute of Technology, Atlanta, in 1981 and 1985, respectively.

From 1980 to 1985 he was a Research Assistant at the Digital Signal Processing Laboratory of the Electrical Engineering School of Georgia Tech, where he was engaged in research on image restoration. He is currently an Assistant Professor in the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL. During the 1986-1987 academic year he was an Assistant Professor at Polytechnic University, Department of Electrical Engineering and Computer Science, Brooklyn, NY. His current research interests include signal and image processing, processing of moving images, computational vision, and VLSI implementation of signal processing algorithms. He is the Editor of the book *Digital Image Restoration* (New York: Springer-Verlag).

Dr. Katsaggelos is a member of SPIE, the IEEE-CAS Technical Committee on Visual Signal Processing and Communications, the Technical Chamber of Commerce of Greece, and Sigma Xi.



Serafim N. Efstratiadis (S'89) was born in Greece in 1964. He received the Diploma degree in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1986 and the M.S. degree in electrical engineering from Northwestern University, Evanston, IL, in 1988. He is currently working toward the Ph.D. degree in the area of motion compensated image sequence restoration.

He has been a Research Assistant, and currently he is a Teaching Assistant, in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include multidimensional signal processing, image modeling, identification, restoration, and video communications.

Mr. Efstratiadis is a member of the Technical Chamber of Commerce of Greece.