Delay Estimation Based on the ML Criterion

In the last chapter, we developed an estimation scheme based on the MAP criterion and demonstrated its feasibility by simulation experiments on three sets of experiments. In this chapter, a gradient-based algorithm is developed and implemented. The estimator is based on the maximum likelihood (ML) criterion and is referred to as the generalized maximum likelihood (GML) algorithm.

This chapter is organized as follows. In Section 1, we present the proof of the general GML algorithm for demodulation of a waveform. Section 2 is devoted to the implementation of the GML algorithm for variable time delay estimation. Section 3 is focused on the simulation experiments.

1. THE GENERALIZED MAXIMUM LIKELIHOOD ALGORITHM (GML)\(^1\)

The objective of this section is to derive a new estimator based on the maximum likelihood criterion.

To help us derive the estimator equation, consider

\[
    r(t) = s(t, a(t)) + n(t) \tag{5.1}
\]

where \(n(t)\) is assumed to be a sample function of a zero mean white Gaussian noise process with variance \(N_0/2\). The preliminary steps of derivation are parallel to [3]. We represent \(a(t)\) by a finite term orthogonal expansion

\[
    a_i(t) = \sum_{i=1}^{I} a_i \psi_i(t) \tag{5.2}
\]

and then let \(I \to \infty\).

\(^1\)The proof of the GML algorithm was developed by J.A. Stuller.
In (5.2) $\psi_l(t)$ is the eigenfunction solution of
\[ \mu_i \psi_l(t) = \int_{T_l}^{T_f} K_u(t, u) \psi_l(u) du. \] (5.3)

$\mu_i$ is the corresponding eigenvalue, $K_u(t, u)$ is the covariance function, and
\[ a_i = \int_{T_l}^{T_f} a(t) \psi_l(t) dt. \] (5.4)

Then, $s(t, a(t))$ is equivalent to $s(t, a)$ in which $a = (a_1, a_2, \ldots, a_I)^T$. The maximum likelihood estimate of $a$ is the vector $A = [A_1, A_2, \ldots, A_I]^T$ maximizing the log-likelihood function [3]:
\[ \ell(n \Lambda_1[r(t), A]) = \frac{2}{N_0} \int_{T_l}^{T_f} r(t) s(t, A) dt - \frac{1}{N_0} \int_{T_l}^{T_f} s^2(t, A) dt \] (5.5)

Application of the steepest ascent algorithm to (5.5) yields
\[ \dot{A}_i(n) = \dot{A}_i(n-1) + \frac{\epsilon}{2} \frac{\partial}{\partial A_i} \ell(n \Lambda_1[r(t), A]) \bigg|_{A_i(n-1)} \] (5.6)

where $i = 1, 2, \ldots, I$, and the parameter $\epsilon_i$ is chosen to influence the rate of convergence. The substitution of (5.5) in (5.6) yields
\[ \dot{A}_i(n) = \dot{A}_i(n-1) + \frac{\epsilon}{N_0} \int_{T_l}^{T_f} [r(z) - s(z, A_i(n-1))] \frac{\partial s(z, A)}{\partial A_i} \bigg|_{A_i(n-1)} dz \] (5.7)

or equivalently,
\[ \dot{A}_i(n) = \dot{A}_i(n-1) + \frac{\epsilon}{N_0} \int_{T_l}^{T_f} [r(z) - s(z, A_i(n-1))] \frac{\partial s(z, a_1(z))}{\partial a_1(z)} \frac{\partial a_1(z, n-1) \psi_i(z) dz}{\partial a_1(z, n-1)} \] (5.8)

in which
\[ \dot{a}_i(t, n) = \sum_{i=1}^{I} \dot{A}_i(n) \psi_i(t). \] (5.9)
Further combination of Equations (5.8) and (5.9) with \( \epsilon_i = \epsilon \mu_i \) yields
\[
\hat{a}_i(t, n) = \hat{a}_i(t, n - 1) +
\]
\[
\frac{\epsilon}{N_0} \int_{t_i}^{T_f} \left[ r(z) - s(z, \hat{a}_i(z, n - 1)) \right] \frac{\partial s(z, a(z))}{\partial a(z)} \Bigg|_{a(z, n-1)} \sum_{i=1}^{I} \mu_i \psi_i(t) \psi_i(z) dz.
\]
(5.10)

Taking the limit \( I \to \infty \) in (5.10) with
\[
\hat{a}(t, n) = \lim_{I \to \infty} \hat{a}_i(t, n)
\]
(5.11)

and invoking Mercer’s Theorem [3]
\[
K_a(t, z) = \lim_{I \to \infty} \sum_{i=1}^{I} \mu_i \psi_i(t) \psi_i(z)
\]
(5.12)

the Equation (5.10), finally, becomes
\[
\hat{a}(t, n) = \hat{a}(t, n - 1) + \frac{\epsilon}{N_0} \int_{t_i}^{T_f} K_a(t, z) \frac{\partial s(z, a(z))}{\partial a(z)} \Bigg|_{a(z, n-1)} \left[ r(z) - s(z, \hat{a}(z, n - 1)) \right] dz.
\]
(5.13)

Note that the Equation (5.13) is not the ML estimator in the conventional sense because it uses the knowledge of the covariance function \( K_a(t, z) \). However, it makes no additional assumption regarding the statistics of \( a(t) \). Since it uses the ML estimates of the Karhunen-Loeve expansion coefficients, the Equation (5.13) is based on the ML criterion and is referred to as the generalized maximum likelihood (GML) estimator.

In the following section, we specialize the GML algorithm (5.13) to the problem of variable time delay estimation.

2. IMPLEMENTATION USING THE GML ALGORITHM

Equation (5.13) can be readily related to the delay estimation problem by proper choice of functions. For convenience, we rewrite the receiver equations (1.2)
\[
r_i(t) = s(t) + n_i(t)
\]
\[
r_j(t) = s(t - D(t)) + n_j(t)
\]
(1.2)
Modulation with Memory

\[ x(t) = \int_{T_i}^{T_f} h(t, u) a(u) \, du \]

\[ r(t) = s(t, x(t)) + n(t) \]

\[ a(t) = \sum_{i=1}^{K} a_i \psi_i(t) \]

\[
\begin{align*}
  x(t) &= \int_{T_i}^{T_f} h(t, u) \left( \sum_{i=1}^{K} a_i \psi_i(u) \right) \, du \\
  &= \sum_{i=1}^{K} a_i \int_{T_i}^{T_f} h(t, u) \psi_i(u) \, du
\end{align*}
\]

\[ s(t, x(t)) = s(t, A) \]

\[ r(t) = s(t, A) + n(t) \]
Recall

Multiple Parameter Estimation:

\[ \hat{A}_r = M_r \int_{T_i}^{T_f} \left( \frac{\cos(z, A_r)}{\rho A_r} \right) \left[ r(g(z)) - g(z) \right] \, dz, \quad r = 1, 2, \ldots, k \]

\[ A = \hat{A} \]

When

\[ M_r = \text{Var}(e_r) \]

\[ r_g(z) - g(z) = \int_{T_i}^{T_f} \rho A_i(z, \hat{u}) \left[ r(\hat{u}) - s(y, \hat{a}) \right] \, du \]

Looking at (1) we find

\[ \frac{\cos(z, A_r)}{\rho A_i} = \frac{\cos(z, x(z))}{\rho x(z)} = \frac{\cos(z, x(z))}{\rho x(z)} \cdot \frac{\rho x(z)}{\rho A_i(x(z))} \]

\[ (1) = \int_{T_i}^{T_f} h(z, u) \psi_i(u) \, du \]
\[ \frac{\cos(z,A)}{\cos A_i} = \frac{\cos(z,x(z))}{\cos x(z)} \cdot \int_{T_i}^{T_f} h(z,u) \psi_i(u) \, du. \] (2)

(2) \quad \Rightarrow \quad (x) = b

\[ \hat{a}_i = M_i \int_{T_i}^{T_f} \left[ \frac{\cos(z,x(z))}{\cos x(z)} \int_{T_i}^{T_f} h(z,u) \psi_i(u) \, du \right] \left( r_g(z) - g(z) \right) \, dz \] (3)

Where

\[ \hat{x}(t) = \int_{T_i}^{T_f} h(t,u) \hat{a}(u) \, du. \]

As before

\[ \hat{a}(t) = \sum_{i=1}^{K} \hat{a}_i \psi_i(t) \] (4)
\[
\hat{a}_k(t) = \sum_{i=1}^{k} \left\{ \mu_i \int_{T_i}^{T_f} \left[ \frac{\cos(z, \tilde{\chi}(z))}{\tilde{\chi}(z)} \right] h(z, u) \Psi_i(u) du \right\} (r_g(z) - g(z)) \, dz \psi_i(t)
\]

\[
= \int_{T_i}^{T_f} \frac{\cos(z, \tilde{\chi}(z))}{\tilde{\chi}(z)} h(z, u) \left[ \sum_{i=1}^{k} \mu_i \psi_i(u) \psi_i(t) \right] (r_g(z) - g(z)) \, dz \, du
\]

As \( k \to \infty \),

\[
\hat{a}(t) = \int_{T_i}^{T_f} \frac{\cos(z, \tilde{\chi}(z))}{\tilde{\chi}(z)} h(z, u) K_a(t, u) \left[ r_g(z) - g(z) \right] \, dz \, du.
\]

If we let \( \int_{T_i}^{T_f} h(z, u) K_a(t, u) \, du = h_a(z, t) \),

Then,

\[
\hat{a}(t) = \int_{T_i}^{T_f} \frac{\cos(z, \tilde{\chi}(z))}{\tilde{\chi}(z)} h_a(z, t) \left[ r_g(z) - g(z) \right] \, dz.
\]
When noise is white, \( r(t) = 0 \),

\[
\hat{a}(t) = \frac{2}{N_0} \int_{T_i}^{T_f} \frac{\rho s(z, \tilde{x}(z))}{\rho \tilde{x}(z)} h_a(z, t) \left[ r(z) - s(z, \tilde{x}(z)) \right] dz.
\]

\[
Q_n(t, u) = \frac{2}{N_0} \delta(t-u)
\]
1. FM is an example of modulation with memory. Here,

\[ x(t) = x_f \int_0^t a(u) \, du \]

\[ \uparrow \quad T_i \]

\[ \text{the deviation} \]

\[ s(t, x(t)) = \sqrt{2} P \sin \left( \omega_c t + \frac{df}{T_i} \int_0^t a(u) \, du \right). \]

2. \[ \hat{x}(t) = \hat{x}(t). \]

\[ a(t) \]

\[ \hat{a}(t) \]

\[ h(t, u) \]

\[ x(t) \]

\[ \hat{x}(t) = \hat{x}(t). \]

wish to know that if a linear operation on a MAP estimate of a finite random process equals the MAP estimate of the output of the linear operation.
Note that

$$\tilde{x}(t) = \int_{T_i}^{T_f} h(t, u) \tilde{a}(u) \, du$$  \hspace{1cm} (1)$$

and

$$\tilde{a}(t) = \int_{T_i}^{T_f} \int_{\mathbb{R}} \frac{\cos(z, \tilde{x}(z))}{\sqrt{\tilde{x}(z)}} \, h(z, y) \, k_a(t, y) \left[ r_g(z) - g(z) \right] \, dy \, dz \hspace{1cm} (2)$$

$$n(2) \Rightarrow (1) = 1$$

$$\tilde{x}(t) = \int_{T_i}^{T_f} h(t, u) \left\{ \int_{T_i}^{T_f} \frac{\cos(z, \tilde{x}(z))}{\sqrt{\tilde{x}(z)}} \, h(z, y) \, k_a(t, y) \left[ r_g(z) - g(z) \right] \, dy \, dz \right\} \, du$$

$$= \int_{T_i}^{T_f} \frac{\cos(z, \tilde{x}(z))}{\sqrt{\tilde{x}(z)}} \left[ r_g(z) - g(z) \right] \left\{ \int_{T_i}^{T_f} h(t, u) \, h(z, y) \, k_a(t, y) \, du \right\} \, dy \, dz \hspace{1cm} (3)$$
We note that
\[ r(t) = s(t, x(t)) + n(t) \]

with
\[ \hat{x}(t) = \int_{T_i}^{T_f} \frac{\cos(z, \hat{x}(z))}{\cos \hat{z}(z)} K_x(t, z) \left[ r_g(z) - g(z) \right] dz \]  

(4)

Hence, (3) and (4) would be the same if

\[ \int_{T_i}^{T_f} \int_{T_i}^{T_f} h(t, u) h(z, y) K_a(u, y) \, du \, dy = K_x(t, z) \]  

(5)

but,
\[ K_x(t, z) = E \left\{ x(t) x(z) \right\} = E \left\{ \int_{T_i}^{T_f} \int_{T_i}^{T_f} h(t, u) a(u) \, du \int_{T_i}^{T_f} h(z, y) a(y) \, dy \right\} \]

\[ = \int_{T_i}^{T_f} \int_{T_i}^{T_f} h(t, u) h(z, y) K_a(u, y) \, du \, dy \]

which validates (5).

\[ \tilde{x}(t) = \hat{x}(t) \]
A NEW ITERATIVE ALGORITHM FOR IMAGE RESTORATION BASED ON MAXIMUM LIKELIHOOD PRINCIPLE

Nader M. Namazi and Chieh M. Fan
Department of Electrical Engineering
The Catholic University of America
Washington, DC 20064


ABSTRACT
This paper is concerned with the development and implementation of a new gradient-based algorithm for image restoration. The algorithm assumes that the original intensity signal \( s(x) \) has been affected by a (known) linear, but not necessarily space-invariant, point spread function (PSF) \( h(x,u) \) in an additive white Gaussian noise environment. It is assumed that the covariance function of \( s(x) \) is known \textit{a priori}. Based on these assumptions, the algorithm tends toward the maximum likelihood estimate of \( s(x) \) using the steepest ascent routine. It is shown that the algorithm is reduced to the least squares error (LSE) restoration scheme reported by Angel and Jain [1] in the absence of noise when the covariance function of \( s(x) \) is an impulse function. Simulation experiments are presented for the algorithm with actual and modeled statistics and comparisons are made with the LSE algorithm.

1. INTRODUCTION
This paper is focused on the development and implementation of a new gradient-based iterative algorithm for image restoration. Image restoration has found use in many areas such as meteorology, medicine, geophysics, and astronomy. For example, in astronomy, the star images are often blurred due to atmospheric turbulence or misfocus [2]. Therefore, image restoration is needed to retrieve the original star images.

The degraded image model is commonly described by the following image formation system [3]:

\[
\begin{align*}
    r(x) &= b(x) + w(x) \\
    b(x) &= \int_{R} h(x,u)s(u)du
\end{align*}
\]

where

\[
x = [x_1, x_2]^T
\]

is a spatial vector representing the pixel location, \( r(x) \) is the monochrome intensity function at pixel \( x \) of the noisy image, \( s(x) \) is the monochrome intensity function of the original unblurred image, \( w(x) \) is additive white Gaussian noise with zero-mean and four-sided spectral density \( N/4 \). \( h(x,u) \) represents the point spread function (PSF) of the blurring system, and \( R \) denotes the support of the PSF. The objective is to estimate the original image \( s(x) \) from \( r(x) \).

The literature includes a large body of methods for image restoration [4]. In this paper, we present a new iterative method that considers the statistical aspects of the data and is able to deal with space-varying PSF’s in the presence of noise. The scheme utilizes the steepest ascent routine to search for the maximum likelihood estimate of the Karhunen-Loeve coefficients of \( s(x) \).

2. ESTIMATION BASED ON THE MAXIMUM LIKELIHOOD (ML) CRITERION
In this section, we confine our attention to the derivation of an iterative algorithm that seeks the maximum likelihood estimation of \( s(x) \). To proceed, expand \( s(x) \) using the Karhunen-Loeve (KL) series; that is,

\[
s(x) = \sum_{k=1}^{\infty} \phi_k(x) \lambda_k
\]

where \( \{\phi_k(x)\} \) is a complete orthonormal set that satisfies the integral equation:

\[
\int_{R} \phi_k(u) K(x,u)du = \lambda_k \phi_k(x)
\]

wherein \( K(x,u) \) is the covariance function of \( s(x) \) and \( \lambda_k \) denotes the eigenvalue of the covariance function.

To overcome the convergence ambiguity, define

\[
s(k)(x) = \sum_{i=1}^{k} \sum_{j=1}^{k} s_{ij} \phi_i(x) \phi_j(x)
\]

Then, define

\[
\int_{R} h(x,u)s(k)(u)du = \delta(x,5)
\]

where \( S = [s_{11}, s_{12}, \ldots, s_{kk}]^T \). Consequently, (1) becomes

\[
r(k)(x) = h(x,5) + w(x)
\]

where \( r(k)(x) \) approaches \( r(x) \) as \( k \) goes to infinity. It is well known that [5] the maximum likelihood estimate of \( S \) is the vector \( S = [s_{11}, s_{12}, \ldots, s_{kk}] \) that maximizes the log-likelihood function. Because the noise is white and Gaussian, the log-likelihood function can be formed as follows:

\[
\ln \Lambda[r^{(k)}(x),S] = -\frac{1}{2N} \int_{R} r^{(k)}(x)\delta(x,5)dx
\]

\[
- \frac{1}{2} \int_{R} \delta^2(x,5)dx
\]

Using the steepest ascent algorithm, we get
\[ \tilde{S}_j^{*} = \tilde{S}_j - \frac{\alpha_j}{\beta_j} \int \int \frac{\partial}{\partial S_j} \delta(x,S_j) \left[ \tilde{r}_j^{(k)}(x) - \delta(x,S_j) \right] dx \]

where, \( \alpha_j > 0, j = 1,2, \ldots, k \), are convergence parameters and \( \tilde{r}_j^{(k)} \) and \( \delta(x,S_j) \) signify the present and old estimates, respectively. Application of (7) in (8) yields

\[ \tilde{S}_j^{*} = \tilde{S}_j^{i} \]

\[ + \frac{\alpha_j}{\beta_j} \int \int \frac{\partial}{\partial S_j} \delta(x,S) \left[ \tilde{r}_j^{(k)}(x) - \delta(x,S) \right] dx \]

It follows from (4) and (5) that

\[ \frac{\partial}{\partial S_j} \delta(x,S) = \int h(x,u) \phi_{i,j}(u) du \]

We define the ML estimate of \( \tilde{r}_j^{(k)}(x) \) as

\[ \tilde{r}_j^{(k)}(x) = \sum_{i=1}^{k} \sum_{j=1}^{k} \delta_{i,j} \phi_{i,j}(x) \]

Henceforth, (9)-(11) renders

\[ \tilde{S}_j^{(k)}(x) = \tilde{r}_j^{(k)}(x) \]

\[ + \frac{1}{N_0} \int \int \int h(x,u) \left[ \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_{i,j} \phi_{i,j}(u) \right] dx \]

\[ \left[ \tilde{r}_j^{(k)}(x) - \delta(x,S) \right] \int \frac{\partial}{\partial S_j} \delta(x,S) dx du \]

(13)

where \( \tilde{S}_j^{(k)}(x) \) gains upon \( \tilde{r}_j^{(k)}(x) \) as \( k \) moves toward infinity. The expression (14) implies that \( \tilde{r}_j^{(k)}(x) \) is the output of the blurring filter \( h(x,u) \) when the input is \( s(x) \). Further combination of (12)-(14), with \( k \rightarrow \omega \), and using the Mercer's theorem [5], results in the following estimator:

\[ \tilde{s}_j^{*} = \tilde{s}_j^{i} + \tilde{r}_j^{(k)}(x) + \gamma \int \int h(x,u) \tilde{r}_j^{(k)}(u) du \]

(15)

in which \( \gamma = \frac{\alpha_j}{\beta_j} \) and \( \tilde{r}_j^{(k)}(x) \) denotes the cross-covariance function between \( s(x) \) and \( b(x) \). In (15) we have used the fact that

\[ K_{ab}(x,z) = \int \int h(x,u)K_{ss}(x,u) du \]

where \( K_{ss}(x,u) \) signifies the covariance function of \( s(x) \). Since \( K_{ab}(x,z) \) is attained by passing the covariance function of the original intensity signal \( K_{ss}(x,u) \) through the blurring filter, it can be viewed as the blurred covariance of \( s(x) \).

Note that the algorithm (15) is not a maximum likelihood (ML) estimator in the strict sense since it utilizes the a priori knowledge of the cross-covariance function \( K_{ab}(x,z) \). However, due to the fact that it

searches for the ML estimates of the KL coefficients of \( s(x) \), we refer to it as the generalized maximum likelihood (GML) algorithm.

The update term of the GML algorithm (15) includes the blurred error term \( b(x) - \tilde{b}(x) \) which is a linear function of the estimation error of the original intensity signal \( s(x) - \tilde{s}(x) \). This useful error function tends to drive the algorithms into lock. The convergence of the algorithms at each iteration is governed by the cross-covariance function \( K_{ab}(x,z) \) and the positive constant \( \gamma \). In addition, in the absence of any blur, \( h(x,z) = \delta(x-z) \), where \( \delta(x) \) denotes the two-dimensional impulse function; consequently, it follows from (14) and (16) that \( \tilde{b}(x) = \tilde{s}(x) \) and \( K_{ab}(x,z) = K_{ss}(x,z) \). In this event, the GML algorithm (15) becomes the GML smoothing filter [6-7].

For digital computations, we can approximate (15) over an \( m \times J \) block of pixels; that is,

\[ \tilde{s}_{i,j}^{*} = \tilde{s}_{i,j}^{i} \]

\[ + \gamma \sum_{t,m} K_{ab}(i-1,j-1,t,m) [\delta_{t,m} - \tilde{b}_{t,m}] \]

where \( i = 1,2, \ldots, I; j = 1,2, \ldots, J \) are the pixel locations and all the appropriate constants are absorbed in \( \gamma \). Similarly, (14), with \( k \rightarrow \omega \), can be written as

\[ \tilde{b}_{i,j} = \sum_{t,m} h_{t,m} s_{t,m} \]

(18)

Accordingly, (15) becomes

\[ K_{ab}(i-1,j-1,t,m) = \sum_{n,s,n'} h_{t,m,n,n'} K_{ss}(i-1,j-1,n,n') \]

(19)

It can be shown that when the PSF is space-invariant and \( s_{i,j} \) is a sample function from a stationary random field, then \( K_{ab}(i,j) = K_{ab}(i-1,j-1) \). In this event, the summations in (17) and (18) are convolution relationships. Henceforth, combination of (17) and (18) in the frequency domain becomes:

\[ \tilde{S}_{i,j}^{\omega_1,\omega_2} = \left[ 1 - \gamma K_{ab}(\omega_1,\omega_2) H(\omega_1,\omega_2) \right] \tilde{S}_{i,j}^{\omega_1,\omega_2} \]

\[ + \gamma K_{ss}(\omega_1,\omega_2) R(\omega_1,\omega_2) \]

(20)

where \( \langle \omega_1,\omega_2 \rangle \) is a spatial frequency vector, \( \tilde{S}_{i,j}^{\omega_1,\omega_2} - \mathcal{F}[s_{i,j}] \), \( \mathcal{F}[\cdot] \) signifies the discrete Fourier transformation, \( K_{ab}(\omega_1,\omega_2) = \mathcal{F}[K_{ab}(\cdot,\cdot)] \), \( H(\omega_1,\omega_2) = \mathcal{F}[H(\cdot,\cdot)] \), and \( R(\omega_1,\omega_2) = \mathcal{F}[R(\cdot,\cdot)] \). It can be demonstrated that when the expression (20) converges, the solution \( s_{i,j}^{\omega_1,\omega_2} \) approaches \( K_{ss}(\omega_1,\omega_2) R(\omega_1,\omega_2) \). In other words, in the steady state, the algorithm (20) acts as an inverse filter [4].

Finally, for \( v(\cdot,\cdot) = 0 \) \( \\forall \cdot \cdot \) over an \( m \times J \) block of pixels, the least square error (LSE) algorithm reported by Angel and Jain [1] is shown to be:

\[ \tilde{s}_{i,j}^{*} = \tilde{s}_{i,j}^{i} \]

\[ + \alpha \sum_{t,m} h_{t,m} i,j [\delta_{t,m} - \tilde{b}_{t,m}] \]

(21)

where \( \tilde{b}_{t,m} \) is defined by (18), \( i = 1,2, \ldots, I; j = 1,2, \ldots, J \), and \( \alpha \) denotes the convergence parameter of the algorithm. Comparison of the GML algorithm (17) and the LSE scheme (21) indicates that similar estimates are obtained when

\[ K_{ab}(i,j,t,m) = h(t,m;i,j) \]

(22)
In addition, in view of the expressions (19) and (21), it is seen that the LSE algorithm (21) assumes that
\[ K_{ss}(i,j;n,n') = \sigma_s^2 \delta(i-n,j-n') \]  
(23)
where \( \sigma_s^2 \) is the variance of \( s(i,j) \) and \( \delta(\cdot,\cdot) \) is the two-dimensional Kronecker impulse function. Consequently, the CML and LSE algorithms are equivalent if noise is zero and if all the samples \( s(i,j) \) are uncorrelated, \( \forall i, j \). Notice that the LSE algorithm (21) is most useful when little (or nothing) is known about the statistical properties of the data.

3. CONVERGENCE OF THE CML ALGORITHM

This section is devoted to the convergence analysis of the CML algorithm (17). Similar to (18), (1b) can be written as
\[ b(i,j) = \sum_{f=1}^{I} \sum_{m=1}^{J} h(i,j;f,m)s(f,m) \]  
(24)
Thus, mapping (1a) into a discrete domain and ordering its samples lexicographically renders
\[ x = Hs + w \]  
(25)
where \( \tau = [a(1), a(2), \ldots, a(1)]^T, a(i) = [r(i,1), r(i,2), \ldots, r(i,J)], e = [a(1), a(2), \ldots, a(I)]^T, s = [s(1), s(2), \ldots, s(I)]^T, w = [w(1), w(2), \ldots, w(I)]^T \) in which \( w(i) = [w(i,1), w(i,2), \ldots, w(i,J)] \) and \( H \) is an \( I \times I \) matrix described as
\[
H = \begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1I} \\
A_{21} & A_{22} & \cdots & A_{2I} \\
\vdots & \vdots & \ddots & \vdots \\
A_{I1} & A_{I2} & \cdots & A_{II}
\end{bmatrix}
\]
where
\[
A_{ij} = \begin{bmatrix}
h(1,i;1,i) & h(1,i;1,2) & \cdots & h(1,i;1,J) \\
h(1,i;2,1) & h(1,i;2,2) & \cdots & h(1,i;2,J) \\
\vdots & \vdots & \ddots & \vdots \\
h(1,i;J,1) & h(1,i;J,2) & \cdots & h(1,i;J,J)
\end{bmatrix}
\]
Combination of (17)–(19), (24), and (25), yields
\[ \tilde{z} = s + \gamma K_{ss} [H (s - \hat{s}) + w] \]  
(26)
in which \( \tilde{s} = [s(1), s(2), \ldots, s(I)]^T, \hat{s}(i) = [s(1,i), s(2,i), \ldots, s(I,i)]^T, w(i) = [w(i,1), w(i,2), \ldots, w(i,J)] \). The matrix \( K_{ss} \) is the \( I \times I \) cross-covariance matrix between \( s \) and \( b \) where \( b = [b(1), b(2), \ldots, b(I)]^T, b(i) = [b(1,i), b(2,i), \ldots, b(I,i)]^T \). The matrix \( K_{ss} \) is characterized as
\[
K_{ss} = \begin{bmatrix}
X_{11}X_{12} & \cdots & X_{1I} \\
X_{21}X_{22} & \cdots & X_{2I} \\
\vdots & \vdots & \ddots & \vdots \\
X_{I1}X_{I2} & \cdots & X_{II}
\end{bmatrix}
\]
where
\[
K_{ss} = K_{ss}(1,1;1,1)K_{ss}(1,2;1,1)K_{ss}(1,3;1,1) \cdots K_{ss}(1,I;1,1) \\
X_{11} = K_{ss}(1,1;1,2)K_{ss}(1,2;1,2)K_{ss}(1,3;1,2) \cdots K_{ss}(1,I;1,2) \\
X_{12} = K_{ss}(1,1;1,3)K_{ss}(1,2;1,3)K_{ss}(1,3;1,3) \cdots K_{ss}(1,I;1,3) \\
\vdots & \vdots & \ddots & \vdots \\
X_{I1} = K_{ss}(1,1;1,I)K_{ss}(1,2;1,I)K_{ss}(1,3;1,I) \cdots K_{ss}(1,I;1,I)
\]
and \( X_{ij} = s(i) \tilde{s}(j)^T, i = 1, 2, \ldots, I, j = 1, 2, \ldots, I \). It follows from (26) that
\[ e' = (I - \gamma K_{ss} H) e' - \gamma K_{ss} w \]  
(27)
in which \( e' = e - s \) is the IE matrix.

We now take the expected value of both sides of (27):
\[ E(e') = (I - \gamma K_{ss} H) E(e') \]  
(28)
in which the fact that the noise vector \( w \) is zero-mean has been utilized. The expression (28) reveals that the algorithm provides unbiased estimates at every iteration if the initial estimate is unbiased. This can be easily achieved by setting \( \bar{s} = \bar{E}(s) \), where \( \bar{s} \) denotes the initial estimate of \( s \).

It is well known that (28) is convergent if, and only if, every eigenvalue of the matrix \( (I - \gamma H) K_{ss} H \) lies within the unit circle; in other words,
\[ |\lambda (I - \gamma H) K_{ss} H | < 1 \]  
(29)
where \( \lambda(\cdot) \) signifies the eigenvalue of its argument. It follows from (29) that
\[ |1 - \gamma \lambda (K_{ss} H H)| < 1 \]  
(30)
Using (16), it can be shown that
\[ K_{ss} = K_{ss} H H \]  
(31)
where \( K_{ss} \) is the covariance matrix of \( s \). Therefore,
\[ \lambda(K_{ss} H H) > 0 \]  
(32)
Because \( K_{ss} \) is a covariance matrix, it is positive-definite. In addition, the matrix \( H H \) is also positive-definite. Henceforth, the eigenvalues of the matrix \( K_{ss} H H \) are all real and positive; that is
\[ \lambda_{\max}(K_{ss} H H) > 0 \]  
(33)
Consequently, from (30), the convergence condition is
\[ 0 < \gamma < \frac{2}{\lambda_{\max}(K_{ss} H H)} \]  
(34)
The estimation of the maximum eigenvalue of the matrix \( K_{ss} H H \), in general, impractical; we utilize the well-known fact that it is over bounded by the trace:
\[ \lambda_{\max}(K_{ss} H H) \leq \text{Tr}(K_{ss} H H) \]  
(35)
Thus, a convenient range of values for \( \gamma \) is
\[ 0 < \gamma < \frac{2}{\text{Tr}(K_{ss} H H)} \]  
(36)
It follows from the definitions of \( K_{ss} \) and \( H \) that
\[ \text{Tr}(K_{ss} H H) = \sum_{i=1}^{I} \sum_{j=1}^{I} K_{ss}(i,j;i,j)h(i,j;f,m) \]  
(37)
which upon substitution in (36) leads to the desired result.

4. IMPLEMENTATION AND COMPARISON OF THE CML AND LSE ALGORITHMS

Simulation of the algorithms was performed in MATLAB [8], a very complete and powerful matrix programming environment. To demonstrate the convergence and performance of the algorithms, we define a normalized mean-squared error (NMSE) function at every iteration:
\[ \text{NMSE} = \frac{1}{N} \sum_{i=1}^{I} \sum_{j=1}^{J} [s(i,j) - s(i,j)]^2 \]  
(38)
where \( N \) is a normalizing constant to make \( \text{NMSE} = 1 \) at the first iteration. Figure 1 illustrates \( s(i,j) \) for \( i,j = 1, 32 \). It should be mentioned that for the simulations, \( s(i,j) \) was made zero mean such that the algorithms be initialized with zero.

A signal-to-noise ratio in dB is defined as follows:
\[ \text{SNR} = 10 \log \left( \frac{s^2}{w^2} \right) \]  
(39)
\[ \text{Note that we have implicitly assumed that neither of the matrices } K_{ss} \text{ or } H \text{ is singular; otherwise, the inequality } (33) \text{ reads } \lambda(K_{ss} H H) \geq 0 \text{ and, as a result, (28) can become neutrally stable.} \]
where \( \sigma_f^2 \) is the average intensity variance and \( \sigma_n^2 \) is the variance of the additive noise. In all the experiments, we assume that SNR = 20 dB.

Experiments were performed for three types of common space-invariant blur and one type of space-varying blur. The space-invariant blurs are defined as [9-10]:

(a) Uniform motion blur:
\[
h(i,j) = \frac{1}{2d} \text{for} \quad 0 \leq i \leq d\quad (40)
\]

(b) Out of focus blur:
\[
h(i,j) = \frac{1}{\pi R^2} \text{for} \quad i^2 + j^2 \leq R^2\quad (41)
\]

(c) Truncated Gaussian blur:
\[
h(i,j) = A \exp \left(-\frac{i^2 + j^2}{2\sigma^2}\right) \text{for} \quad i^2 + j^2 \leq R^2\quad (42)
\]

where
\[
A = \frac{1}{\sqrt{2\pi \sigma^2} \left(1 - \exp\left(-\frac{R^2}{\sigma^2}\right)\right)^{1/2}}
\]

For the motion blur experiments, \( d = 2 \), for out of focus blur experiments \( R = 2 \), and for truncated Gaussian blur experiments \( \sigma^2 = 10 \) and \( R = 2 \). For the space-varying blur, we assume that the PSF is separable [4]; that is,
\[
h(1,i\cdot|\cdot, m) = h(1,i\cdot) h(j\cdot, m)\quad (43)
\]
where \( h(1,i\cdot) = h(1,j\cdot) \) and \( h(j\cdot, m) \)

\[
h(1,j\cdot) = \exp(-\beta_1 (i1)^2)\quad (44)
\]

with
\[
\beta_1 = 2 \cdot \frac{1916 - 1}{150}\quad (45)
\]

Figures 2a-2d show the noisy blurred images due to the aforementioned PSF's.

**Experiment 1. The GML Algorithm with Actual Statistics**

To investigate the performance of the GML algorithm under near-ideal conditions, we estimated the cross-covariance function \( K_s(1,i\cdot, t\cdot, m) \) directly from \( s(1,i\cdot) \) and \( b(1,j\cdot) \).

It is assumed that \( s(1,i\cdot) \) is stationary. Therefore, for the space-invariance blurs (40)-(42) we have
\[
K_s(1,i\cdot, t\cdot, m) = K_s(1,i\cdot, j\cdot, m)\quad (46)
\]

It is observed that, in this event, the summation part of (17) is a convolution process that can be calculated efficiently using the fast Fourier transform (FFT). To facilitate the computation, we assumed that the expression (46) is also satisfied for the space-varying blur (43). Figures 3a-3d exhibit the restored images after 500 iterations and Figure 4 illustrates the FMSE function (38) for the GML algorithm with actual cross-covariance function.

**Experiment 2. The GML Algorithm with Modeled Statistics**

We now consider the GML algorithm under a more realistic condition in which the actual cross-covariance function is not available. The problem is approximated by modeling \( s(1,i\cdot) \) as a sample function from a separable Markov-2 field [4]:
\[
K_s(1,i\cdot, t\cdot, m) = \sigma_f^2 \left[ 1 - |\rho_1| \right] \left[ 1 - |\rho_2| \right] \quad (47)
\]

where \( 0 < \rho_1 < 1 \) and \( 0 < \rho_2 < 1 \) are the correlation coefficients of the field. The cross-covariance function \( K_s(1,i\cdot, t\cdot, m) \) is then calculated from (47) and (19). In all the experiments we assumed that \( \rho_1 = \rho_2 = \rho \). It was shown, by simulation experiments, that for the image of Figure 1, the respective values of \( \rho = 0.35 \) and \( \rho = 0.25 \) results in the minimum FMSE function for the space-invariance and space-varying blurs previously mentioned. Figures 5a-5d show the restored images for the model-based GML after 500 iterations and Figure 6 presents the FMSE function for this scenario.

**Experiment 3. The LSE Algorithm**

It is interesting to note that the GML algorithm (17) with the Markovian assumption (47), and in the absence of noise, approaches the LSE algorithm (21) as \( \rho_1 \to \rho \).

Figures 7a-7d depict the restored images using the LSE algorithm (21) after 500 iterations and Figure 8 shows the FMSE function (38) for this case.

Figure 9 highlights the FMSE functions averaged over the four aforementioned blurs for each algorithm. This figure reveals that the performance of the model-based GML is comparable with the GML algorithm with the actual cross-covariance function. In addition, both of the GML algorithms outperform the LSE scheme. This is expected since the LSE algorithm does not account for the effect of noise and does not exploit the statistical attributes of the data.

**5. SUMMARY AND CONCLUSIONS**

This paper has developed and implemented a new iterative algorithm (GML) for image restoration. The algorithms searches for the ML estimates of the KL coefficients of the intensity signal using the steepest ascent routine. The GML algorithm presented in this paper assumes the a priori knowledge of the covariance function \( K_s(1,i\cdot, t\cdot, m) \) is available. Estimation of this function is, in general, a difficult problem. Hence, we implemented the GML algorithm under the assumption that \( s(1,i\cdot) \) is a separable Markov-2 field. The simulation experiments demonstrated that, based on the image presented in the Figure 1, the model-based GML algorithm performs better than the LSE and functions nearly the same as the GML algorithm with actual statistics. Overall, the model-based GML algorithm appears to be the best choice since it is a compromise between the other two with realistic statistical assumptions and reasonable performance.

**REFERENCES**


Figure 1. The original image $s(i,j)$.

Figure 2. The degraded images with SNR = 20 dB. 

(a) Uniform motion blur with $d = 2$. 
(b) Out of focus blur with $R = 2$. 
(c) Truncated Gaussian blur with $R = 2$ and $\sigma^2 = 10$. 
(d) Space-varying blur (43)-(45).

Figure 3. The restored images of the patterns shown in Figure 2 after 500 iterations with SNR = 20 dB using the GML algorithm with actual statistics.

(a) The restored image of Figure 2a with uniform motion blur.
(b) The restored image of Figure 2b with out of focus blur.
(c) The restored image of Figure 2c with truncated Gaussian blur.
(d) The restored image of Figure 2d with space-varying blur.

Figure 4. The NMSE function for SNR = 20 dB using the GML algorithm with actual statistics.
Figure 5. The restored images of the patterns shown in Figure 2 after 500 iterations with SNR = 20 dB using the GML algorithm with modeled statistics. 

(a) The restored image of Figure 2a with uniform motion blur. 
(b) The restored image of Figure 2b with out of focus blur. 
(c) The restored image of Figure 2c with truncated Gaussian blur. 
(d) The restored image of Figure 2d with space-varying blur.

Figure 6. The NMSE function for SNR = 20 dB using the GML algorithm with modeled statistics.

Figure 7. The restored images of the patterns shown in Figure 2 after 500 iterations with SNR = 20 dB using the LSE algorithm. 

(a) The restored image of Figure 2a with uniform motion blur. 
(b) The restored image of Figure 2b with out of focus blur. 
(c) The restored image of Figure 2c with truncated Gaussian blur. 
(d) The restored image of Figure 2d with space-varying blur.

Figure 8. The NMSE function for SNR = 20 dB using the LSE algorithm.

Figure 9. The NMSE function averaged over the blur functions.