

영상신호의 고속처리를 위한 최적화 알고리즘에 대한 연구

正會員 權 起 弘*

A study of optimal algorithm for high-speed process of image signal

Kee Hong Kwon*

要 約

본 논문에서는 훼손된 영상을 처리하는 방법에 대하여 연구하였다. 기존의 처리방법은 특이점이나 악 조건일 경우 수렴 속도가 늦어진다든 점과 처리시간이 많이 소요되는 단점이 있다.

이러한 단점을 보완하기 위하여 Gauss-Seidel 방법으로 처리하는 방법이 있으나 이러한 경우 영상을 반복해서 처리해야 하므로 처리시간이 많이 소요된다.

이러한 단점(수렴 속도, 전체 처리시간)을 개선하기 위하여 본 논문에서는 기존의 신호처리(Gauss-Seidel)와 제안된 알고리즘을 적용시켜 비교하여 봄으로써 특이점 혹은 악조건일 경우에도 수렴속도를 고속화하여 기존의 Gauss-Seidel 신호 처리 방법보다 처리 시간을 단축할 수 있는 영상 복원 방법을 제시하였다.

제안된 최적화 알고리즘을 영상신호에 적용시켜 가속 상수에 따른 처리 신호의 simulation과 MSE(mean-square error)의 변화를 비교하여 봄으로써 처리정도를 알아보았다.

그리고, 본 알고리즘의 유효성을 입증하기 위하여 모든 가속상수의 변화에 대한 영상복원 결과와 처리시간을 측정하여 보았다.

ABSTRACT

In this paper, the method of processing a blurred noisy image has been researched.

The conventional method of processing signal has faults which are slow convergence speed and long time-consuming process at the singular point and/or in the ill condition.

There is the process, the Gauss-Seidel's method to remove these faults, but it takes too much time because it processed signal repeatedly.

For overcoming the faults, this paper shows a image restoration method which takes shorter than the Gauss-Seidel's by comparing the Gauss-Seidel's with proposed algorithm and accelerating convergence speed at the singular point and/or in the ill condition.

In this paper, the conventional process method(Gauss-Seidel) and proposed optimal algorithm were used to get a standard image(256×56×bits), and then the results are simulated and compared each other in order

* 신일전문대학 전자계산과
Dept. of Computer Science., Shiil Technical College
論文番號 : 94172
接受日字 : 1994年 6月 27日

to examine the variance of MSE(Mean Square Error) by the acceleration parameter in the proposed image restoration.

The result of the signal process and the process time was measured at all change of acceleration parameter in order to verify the effectiveness of the proposed algorithm.

I. 서 론

일반적인 신호처리 방법의 궁극적인 목적은 원래의 입력신호와 가장 근사한 같은 신호를 재현하는데 중점을 두고 있다.

신호는 시스템이 가지는 특성으로 인해 영상, 감쇠, 기록, 전송의 단계를 거치는 동안 여러원인에 의해 훼손된다.

이러한 훼손은 다음과 같이 표현된다.^[1]

$$y = H \cdot x + n \tag{1}$$

훼손된 신호 y 는 원래의 입력신호 x 가 시스템함수 H 에 의해 흐려지고, 잡음 n 이 더해진 형태로 나타난다.

식(1)과 같이 훼손된 신호는 보통 흐림훼손과 가산잡음에 의한 훼손으로 나뉘어져 분리 처리되고 있다.

즉, 흐림에 의한 훼손을 처리하는 역필터, Jacobi방법, Gauss-Seidel방법등과, 가산잡음에 의한 훼손을 처리하는 Wiener, Lee, CSW, 적응성 필터 등이 있으며, 흐림훼손과 가산잡음에 의한 훼손을 처리하는 Tikhonov-Miller방법등이 있다. 기존의 Jacobi방법이나 Gauss-Seidel방법은 수학적인 모델이 비교적 간단하고, 신호처리효과도 우수한 실용적인 방법이다.

그러나, 이러한 신호처리방법이 어디면에서 우수함에도 불구하고 고비용 신호처리에 적용될때 발생하는 가장 큰 단점은 신호처리시 항상 수립하지 않을 뿐 아니라 수립시 반복처리를 필요로 하여 많은 처리시간의 소요와 식(1)과 같이 잡음이 포함되어 있는 훼손의 경우에 이 방법을 적용하였을 경우 잡음의 증폭이 발생한다.^[2]

본 논문에서는 신호처리의 상점과 Tikhonov-Miller 방법의 상점을 이용한 방법에 가산 상수인 도입, 기존의 여러가지 단점을 보완하고, 신호처리장도를 증가시킬 뿐만 아니라, 기존의 방법에 있어서 가장 문제시되

는 처리시간을 현저히 줄일 수 있는 새로운 알고리즘을 제안하였다.

본 논문에서 제안하게 될 알고리즘은 수학적 해석과 신호처리기술에 바탕을 두며, 이로부터 유도해석된 최적화 상수를 이용하여 신호를 고속처리 가능케 할 것이다.

II. 기존 신호 처리 방법의 분석

1. Jacobi/Gauss-Seidel 처리 방법

신호가 식(1)과 같이 훼손되었을 경우, 신호처리시 역행렬이 존재하지 않으면 해를 구할 수 없다.

즉, H 역행렬 경우, H 가 특이점을 가지거나 약조건인 경우 역행렬 마법으로 처리가 불가능하다. 이러한 단점을 해결하기 위하여 Jacobi 반복 처리 방법이 있는데 이 방법은 H 이 아닌 대각선 요소를 갖는 행렬에 적용한다.

예를 들어 AX-B라는 식 (2)에서 Jacobi방법을 적용시킬 경우 다음과 같다.^[3]

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{2}$$

첫번째 반복시에는

$$\begin{aligned} x_1 &= (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \\ x_2 &= (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \\ x_3 &= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \end{aligned} \tag{3}$$

과 같은 식이 된다.

일반적으로 Jacobi방법의 알고리즘은

$$x_i^{(k+1)} = \frac{(b_i - \sum_{j=1}^{i-1} a_{ij} * x_j^{(k)} - \sum_{j=i+1}^n a_{ij} * x_j^{(k)})}{a_{ii}} \quad (4)$$

이다.⁽¹¹⁾

그런데 Jacobi방법은 $x_2^{(k+1)}$ 의 해를 구할 경우 최
단에 구하여진 해 $x_1^{(k+1)}$ 을 사용하지 않는다.

이러한 단점을 보완한 것이 Gauss-Seidel방법인⁽¹²⁾
이를 수식적으로 표현하면

For $i = 1, 2, 3, \dots, n$

$$x_i^{(k+1)} = \frac{(b_i - \sum_{j=1}^{i-1} a_{ij} * x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} * x_j^{(k)})}{a_{ii}} \quad (5)$$

여기에서 $x_1^{(k+1)}$ 가 계산과정에 포함되어 조금 더
근사한 값이 계산과정에 포함됨으로 처리 속도가 빠름
을 알 수가 있다.

위의 두가지 방법에 메모리 절약과 시간 단축을 위
하여 LDU행렬을 사용한다.

$$L = \begin{pmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ a_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ a_{n1} & \cdot & \cdot & \cdot & \cdot & a_{nm-1} & 0 \end{pmatrix} \quad (6)$$

$$D = \text{diag}(a_{11}, \dots, a_{nn}) \quad (7)$$

$$U = \begin{pmatrix} a_{12} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ & & & & & & a_{2n} \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & a_{1-m} \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{pmatrix} \quad (8)$$

L, D, U행렬을 jacobi방법에 도입하면

$$\begin{aligned} M_j \cdot X^{(k+1)} &= N_j \cdot X^{(k)} + b \\ M_j &= D \quad N_j = (L + D) \end{aligned} \quad (9)$$

Gauss-Seidel 방법은

$$M_j \cdot X^{(k+1)} = N_j \cdot X^{(k+1)} + b \quad (10)$$

여기서 $M_j = (D + L)$ $N_j = -U$ 가 된다.

윗 식에서 해가 얼마나 빨리 $X = A^{-1}B$ 에 수렴
할 수 있느냐가 중요하다. Gauss-Seidel에 의한 방법이
Jacobi의 방법보다 수렴 속도가 약간 빠를 뿐만 아니라
계산기에 적용할 경우 간단하다.⁽¹⁰⁾

2. 정칙화된 Tikhonov-Miller 방법의 분석

Gauss-Seidel방법으로 신호처리시 가산잡음이 포함
되어 있는 훼손신호에 적용하였을 경우 잡음의 증폭이
발생한다.

이러한 단점을 보완한 것이 Tikhonov-Miller 방법이
다.⁽⁹⁾

신호처리시 대개 훼손된 신호의 잡음항 norm의 가
능한 근사치를 사용하게 된다.⁽⁹⁾

가능한 해의 집합은 이 잡음 norm에 대한 개선도
($y - H\hat{x}$)의 norm을 제한함으로써 정의할 수 있다.

정칙화된 Tikhonov-Miller방법은 식(1)에서 개선도
의 norm을 구하면

$$\|y - H\hat{x}\|_R = [(y - H\hat{x})' R (y - H\hat{x})]^{1/2} \leq \epsilon \quad (11)$$

여기서 범위 ϵ 은 훼손된 신호 y 의 잡음항의 합과
관련된 변수이다.

필터를 통과한 신호 $L\hat{x}$ 의 norm은

$$\|L\hat{x}\|_S = [(L\hat{x})' S (L\hat{x})]^{1/2} \leq \epsilon \quad (12)$$

이고, 여기서 L 은 선형 정칙화 연산자이다.

Miller의 정칙화 연구에서는 식(11)와 (12)를 결합하
여

$$\Phi(\hat{x}) = \|y - H\hat{x}\|_R^2 + \alpha \|L\hat{x}\|_S^2 \leq 2\epsilon^2 \quad (13)$$

되고, 여기서 α 는 정착화변수로 고정된 값 $\alpha = (\frac{\epsilon}{E})^2$ 이다.

만약 해 \hat{x} 가 식(11)와 (12)의 범위를 만족한다면 식(13)을 만족한다.

역으로, 해 \hat{x} 가 식(13)을 만족한다면 식(11)와 (12)도 또한 만족한다. 식 (13)을 만족하는 해중에서 적절한 해는 \hat{x}_m 으로써 Miller의 정착화해라 부르며, $\Phi(\hat{x})$ 를 최소화한다.

이 최소화 해는 다음과 같이 주어진다.

$$(H'RH + \alpha L'L) \hat{x}_m = H'Ry \quad (14)$$

$$\hat{x}_m = \frac{H'Ry}{H'RH + \alpha L'L} \quad (15)$$

로 구해진다.

III. 제안된 알고리즘

Tikhonov Miller방법은 Gauss-Seidel방법을 수정 보완한것으로 반복실행치리에 따른 처리시간이 많이 소요된다.

즉, 신호처리를 기존의 방법으로 할 경우 반복치리에 따른 처리시간이 문제가 된다. 이러한 단점을 보완하기 위하여 어떠한 경우에도 수렴속도가 늦어지지 않으며, 가산잡음이 포함된 훼손신호에 적용함에 있어서도 전체 처리시간의 현저한 감소와 신호처리성도를 대폭 개선할 수 있는 새로운 알고리즘을 제안한다.

즉, Jacobi방법은

$$(X_i)^{(k+1)} = \frac{-1}{a_{ii}} [b_i - \sum_{j=1, j \neq i}^n a_{ij} \cdot X_j^{(k)}] \quad (16)$$

For $i=1, 2, 3, \dots, n$ 이다.

Gauss-Seidel방법은 위의 방법을 보완한 것으로

$$a_{ii} \cdot X_i^{(k+1)} = - \sum_{j=1}^{i-1} a_{ij} \cdot X_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \cdot X_j^{(k)} + b_i \quad (17)$$

For $i=1, 2, 3, \dots, n$

즉, Jacobi방법의 계산속도를 빠르게 한 것이다. Gauss-Seidel방법 역시 특이점(singular point)나 악 조건(ill condition)일 경우 수렴 속도가 매우 느려진다는 단점이 있다.

이것은 α 가 0인 (특이점)인 경우 식 (10)의 M_k 항이 작아져 수렴속도가 느려진다는 것이다.

이것은 식(10)에서

$$D \cdot X^{k+1} = -L \cdot X^{k+1} - U \cdot X^k + b \quad (18)$$

와 같이 된다. 여기서

- D : 대각선 행렬
- L : 하부 삼각 행렬
- U : 상부 삼각 행렬

식 (18)를 정리하면

$$X_i^{(k+1)} = \frac{1}{a_{ii}} [b_i - \sum_{j=1}^{i-1} d_{ij} \cdot X_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \cdot X_j^{(k)}] \quad (19)$$

For $i=1, 2, 3, \dots, n$

으로 쓸 수 있는데 (K+1)번째의 근사해 $X^{(k+1)}$ 에 대하여

$$A \cdot X^{(k+1)} = B^{(k+1)} \text{ 이라 하면} \quad (20)$$

$$R^{(k+1)} = B^{(k+1)} - A \cdot X^{(k)}$$

여기서 $R^{(k+1)}$ 은

$$R^{(k+1)} = [(r_1)^{k+1}, (r_2)^{k+1}, \dots, (r_n)^{k+1}]^T \quad (21)$$

과 같이 되는데 이것을 개선도 벡터(Residual vector)라고 한다.

$$R^{(k+1)} = B^{(k+1)} - AX^{(k)}$$

이 말의 의미는 $= B^{(k+1)} - B^{(k)}$ 이므로

(k+1)번째 신호가 (k)번째 신호보다 얼마나 개선되었는가를 나타낸다.

$R^{(k+1)} = 0$ 이면 $X^{(k+1)}$ 가 정해지지만 대개는 $R^{(k+1)} = 0$ 이 아니다.

특히 Gauss-Seidel 모델에서의 $(r_i)^{k+1}$ 은

$$(r_i)^{k+1} = b_i - \sum_{j=1}^{i-1} a_{ij} \cdot X_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \cdot X_j^k - a_{ii} \cdot X_i^k \quad (22)$$

이므로

$$a_{ii} \cdot X_i^k + (r_i)^{k+1} = b_i - \sum_{j=1}^{i-1} a_{ij} \cdot X_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \cdot X_j^k \quad (23)$$

(19)식과 (23)식에서

$$a_{ii} \cdot X_i^k + (r_i)^{k+1} = a_{ii} \cdot X_i^{k+1} \quad (24)$$

$$\text{즉, } X_i^{k+1} = X_i^k + \frac{(r_i)^{k+1}}{a_{ii}} \quad (25)$$

개선도 벡터가 0일 경우에 반복이 멈추어 진다. 그러나 보통 이 값은 0이 아니므로 수렴성의 확실성을 희박하게 한다. 그러므로 여기서 개선도 벡터(Residual vector) R^{k+1} 의 노름을 감소시키기 위하여 식 (25)을 수정하여

$$(X_i)^{k+1} = (x_i)^k + W \cdot \frac{(r_i)^{k+1}}{a_{ii}} \quad (26)$$

로 하여 W의 값에 따라 수렴속도를 조절할 수 있는 방법을 제시한다. W의 도입은 $(r_i)^{(k+1)}$ 의 개선도 vector가 실제로 0이 안됨으로 수렴성을 가속화하기 위한 상수값이다.

여기서 W값을 구하기 위하여

$$\rho(A) = \max \{ |\lambda_i| \} \quad (27)$$

와 같이 정의한다.

단, λ_i 는 A의 고유치이고 만약 $\lambda = a+bi$ 이면 $|\lambda_i| = \sqrt{a^2 + b^2}$ 으로 정의하면

n차 정방 행렬

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ a_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \quad \text{에러}$$

$$L = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ a_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & & & & & \\ & & & & & & & & \\ a_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{nn-1} & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} a_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ a_{22} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ 0 & \cdot & \cdot & \cdot & a_{n-1n} & \cdot & \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

$$A = L + D + U \quad (28)$$

$$T = D^{-1}(L + U) \text{라할때} \quad (29)$$

$$W = \frac{2}{1 + \sqrt{1 - \rho(T)^2}} \quad (30)$$

로 하면 수렴 속도가 가장 빠른것이 증명되어 있다.(11,12)

여기에서 $\rho(T)$ (행렬 T의 최대반경)를 구할시에는 실수 요소로 된 n차 정방행렬 A의 고유치 $(\lambda_i)_{i=1}^n$ 에 대하여

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \quad (31)$$

를 가정하고 A 는 1차 독립인 고유벡터 $(U_i)_{i=1}^n$ 을 갖는다고 하자.

또 n 차 열 벡터 X_0 를

$$X_0 = a_1 U_1 + a_2 U_2 + a_3 U_3 + \dots + a_n U_n \quad (32)$$

이라 하고

$$AU_i = \lambda_i U_i \quad \text{에서}$$

$$A(AU_i) = A\lambda_i U_i = \lambda_i(AU_i)$$

$$\text{즉, } A^2 U_i = (\lambda_i)^2 U_i \quad (33)$$

$$A^3 U_i = (\lambda_i)^3 U_i$$

⋮

$$A^k U_i = (\lambda_i)^k U_i \text{이며,}$$

다음과 같은 수열

$$X_0 = A$$

$$X_1 = AX_0$$

$$X_2 = AX_1, (X_2 = A^2 X_0) \quad (34)$$

⋮

$$X_k = AX_{k-1}, (X_k = A^k X_0)$$

를 만들면 식 (32), (33), (34)에서

$$\begin{aligned} X_k &= A^k \cdot X_0 \\ &= a_1 A^k U_1 + a_2 A^k U_2 + \dots + a_n A^k U_n \\ &= a_1 (\lambda_1)^k U_1 + a_2 (\lambda_2)^k U_2 + \dots + a_n (\lambda_n)^k U_n \\ &\quad \vdots \\ &= (\lambda_1)^k (a_1 U_1 + a_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k U_2 + \dots + a_n \left(\frac{\lambda_n}{\lambda_1}\right)^k U_n) \end{aligned} \quad (35)$$

가 된다.

식 (31)의 조건하에서 K 가 증가하면

$$X_k = (\lambda_1)^k a_1 \cdot U_1 \text{이 된다.}$$

$$\text{또 } X^{k+1} = (\lambda_1)^{k+1} a_1 U_1 \quad \text{또는} \quad X^{k+1} =$$

$$(\lambda_1)^k (\lambda_1) \cdot a_1 U_1 = (\lambda_1)^k X_k \text{이므로 이것을 이용하여 } \lambda_1 \text{을 구할 수 있다.}$$

$$\text{그런데, } (\lambda_i)^k = \frac{X^{k+1}}{X^k} \text{에 의하여 } \lambda_1 \text{을 구하려면} \quad (36)$$

매 단계마다 벡터계산이 필요하다. 이런 점을 제거하기 위하여 예를 들어,

$$X'_k \cdot X_{k+1} = (\lambda_1)^k \cdot X'_k \cdot X_k \text{이므로}$$

$$(X'_k \text{는 } X_k \text{의 전치벡터})$$

$$\lambda_1 = \frac{X'_k X_{k+1}}{X'_k X_k} \text{로써 구하면 scalar 계산이 된다.}$$

또 X_{k+1} 의 선대치 최대요소를 X_k 의 선대치 최대요소로 구하면 λ_1 을 구할수 있으나 이 방법을 매 단계마다 선대치 최대요소를 판별해서 구해야 하고 또 $|\lambda_1|$

1이면 k 가 증가 할수록 X_k 도 커지고 $|\lambda_1|$ 이 k 가 감소하여 자칫 컴퓨터의 계산범위를 벗어날 수도 있다.

이상의 문제점을 해결하기 위하여

$$X'_0 X_0 = 1 \text{ (즉 } \|X_0\| = 1) \text{인 초기치 } X_0$$

를 선정하고 매 단계마다 1-norm 을 이용한 정규화(normalization) 또는 scaling을 통하여 X_k 를 구하여 식 (36)으로써 scalar계산을 통하여 λ_1 을 구한다.

이 알고리즘은

$$(1) X'_0 X_0 = 1 \text{인 초기치 벡터 } X_0 \text{ 선정}$$

$$(2) i = 1$$

$$(3) Y_i = AX_{i-1}$$

$$(4) \beta_i = X'_{i-1} Y_i \text{ (고유치 } \lambda_i)$$

$$(5) \eta_i = \sqrt{Y'_i Y_i} \text{ (} l_2 \text{ norm)}$$

$$(6) X_i = \frac{Y_i}{\eta_i} \text{ (} X_i \text{는 } \|X_i\| = 1 \text{인 고유벡터)}$$

(7) 수렴 판정, λ_i 이 수렴하면 끝.

(8) $i = i + 1$ ($i + 1$)

(9) (3)부터 반복

알고리즘에서 계산되는 β_i 는

$$\beta_i = \frac{X_{i-1}^t Y_i}{X_{i-1}^t X_{i-1}} \text{ 이어야 한다.}$$

여기서 구한 β_i 값을 $\rho(T)$ 에 대입한다.

여기에서 구한 W 와 식 (26)와 식 (22)에서 신호처리에 적용될 최적화 알고리즘의 식을 구하면

$$(\hat{x})^{k+1} = (1 - W) \cdot (\hat{x})^k + W [\hat{x}^k - W_2 \cdot (y - H \hat{x}^k)] \quad (37)$$

이다.(여기서 W_2 위의 상수)

이 방법은 수렴 속도가 낮은 Gauss-Seidel방법을 개선하기 위하여 최적화 한 것이다.

(37)와 (17)을 비교하여 보면 $(1-W) \cdot (X_i)^k$ 와 W_1, W_2 가 사용되었다.

이 W 는 상수값으로 주어지므로 연산 속도에는 크게 미치지 못하는 못하나 $(I - W) \cdot (X_i)^k$ 때문에 같은 반복횟수에 비하여 처리속도가 가속화되고 특이점, 약조건 W 는 상수이므로 빨리 수렴하게 된다.

반복 Tikhonov-Miller방법의 알고리즘 역시 Gauss-Seidel방법을 수정보완한 것이므로 가속상수의 도입과 신호처리의 최적화가 가능하다.

즉, 반복 Tikhonov-Miller방법의 장점을 도입하기 위해 식 (15)를 연속적으로 근사화함으로써

$$\hat{x}_0 = H^t \cdot y \quad (38)$$

$$x^{(k+1)} = (I - \alpha L^t L) \hat{x}^k + \beta H^t (y - H \hat{x}^k) \quad (39)$$

와 같이 구해진다.

여기서 전체 처리시간과 기억 용량의 증가함이 없이 특이점이나 약조건일 경우에도 수렴성을 보장받고, 가산잡음에 대해서도 처리정도를 현저히 개선시킬 뿐만 아니라 처리시간상의 문제점을 해결하기 위하여 식(37)

과 (38)를 정리하여 최적화하면 다음과 같다.

$$x^{(k+1)} = (1 - W_1) \hat{x}^k + W_1 [(I - \alpha L^t L) \hat{x}^k + \beta H^t (y - H \hat{x}^k)] \quad (40)$$

여기서 β 값은 전체연산속도를 최적화하는 값을 사용하였다.

여기서 W_1 과 β 는 상수로써 전체 연산속도에는 영향을 주지 않는다.

IV. 시뮬레이션 결과 및 고찰

본 논문에서 제안한 알고리즘의 유효성을 확인하기 위하여 영상신호처리에 적용시켜 보았다.

사용된 영상신호는 256×256 해상도 및 256의 계조도를 갖는 표준 영상을 사용하였다.

영상을 처리할 경우, 영상처리장치의 불완전함에 의해 흐려지고 가산잡음에 의해 훼손되어 지는데, 가산잡음은 영상의 형성, 진단, 저장과정에 임의로 영상에 섞이게 된다.

본 논문에서는 표준 원 영상신호를 점확산 함수로써 (PSF)로써 흐려지게 하고 흐려진 영상에 가우스 분포를 갖는 백색 잡음(Gaussian White Noise)를 가산하였다. 이 때 흐려진 영상의 신호 대 잡음비 BSNR의 식은

$$\begin{aligned} BSNR &= \frac{\text{variance of the blurred imaged}}{\text{variance of the additive noise}} \\ &= \frac{\sigma^2 H \cdot x}{\sigma^2 n} \end{aligned} \quad (41)$$

과 같다.

본 논문에서 사용된 가산잡음은 20dB의 신호 대 잡음비를 가지는 백색잡음을 사용하였다.

다음과 같이 정의한다.

$$MSE = \frac{1}{M \cdot N} \sum_{i=1}^N \sum_{j=1}^M (X(i, j) - \hat{X}(i, j))^2 \quad (42)$$

훼손된 영상의 MSE는 268.42588이다.

표(1)과 그림(1)은 Tikhonov-Miller방법과 제안된 알고리즘의 평균 자승오차(MSE)를 시간에 대해 비교하여 보았다.

제안된 알고리즘에서 사용된 가속상수는 $W=0.5$
 $\beta = 9.0$ 값을 사용하였으며, IBM PC 486기종을 사용
 하여 시뮬레이션한 처리시간을 측정, 비교하였다.

보다 나은 처리속도를 가진 컴퓨터를 사용한다면,
 처리시간은 단축될 수 있을 것이다

표 1. 처리시간에 대한 평균자승오차(MSE)의 비교

TABLE 1. Compared MSE for processed time

반복횟수	기존의 방법 Tikhonov_miller	제안된 알고리즘 proposed	처리시간 (sec)
1	170.51861	153.79955	90
2	161.51880	162.10701	180
3	158.12446	166.82231	270
4	157.05715	170.49918	350
5	157.03906	173.55008	440
6	157.51382	176.21306	540
7	158.21780	178.62706	630
8	158.02305	180.91017	720
9	159.86165	183.00883	800
10	160.70044	185.21865	890

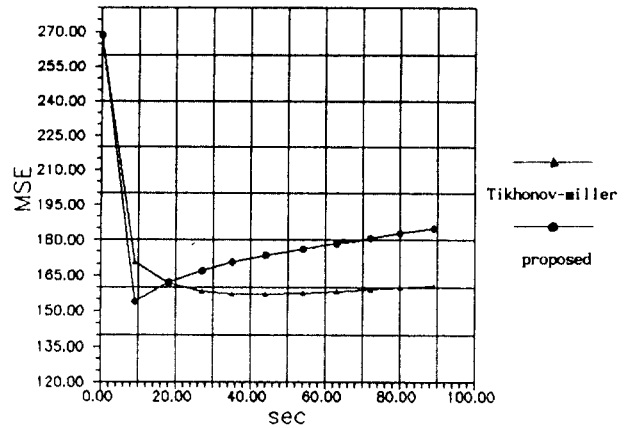


그림 1. 처리시간에 대한 평균자승오차(MSE)의 비교
 Fig. 1. Compared MSE for processed time

표 2. 기준시간 (9sec) 에서의 가속상수에 따른 영상신호
 호차리의 처리 오차(MSE) 비교

Table 2. Compared MSE of image signal process by
 the standard time (9sec)

가속상수	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0
0.5	186.89865	180.19897	174.84111	170.51864	167.01688	164.17286	161.86176	159.76537	158.46642	157.24340	156.28702	155.49749	154.82248	154.25664	154.13679	153.92280	153.93417	153.76695	153.79987	153.85628
1.0	180.13077	170.51864	164.17286	159.86237	157.24340	155.49749	154.45564	153.92280	153.76695	153.85628	154.25664	154.78540	155.42832	156.20756	157.08779	158.05681	159.10770	160.23910	161.45268	162.73122
1.5	174.84111	164.17286	158.46642	155.49749	154.13679	153.76695	154.04698	154.76541	155.80465	157.08779	158.57187	160.23810	162.07854	164.10152	166.33882	168.82286	171.58855	174.79569	178.59949	183.35735
2.0	170.51864	159.86237	155.49749	153.92281	153.82608	154.76541	156.20756	158.36611	160.23810	16.73122	168.36678	172.61038	177.24478	183.35735	192.94619	204.41056	218.22689	234.28965	252.50036	
2.5	167.01688	157.24340	154.13679	153.85608	155.08001	157.08779	159.85322	162.73122	165.33682	170.63382	175.94392	183.35735	196.60386	208.19405	226.76296	250.15836	281.61985	324.87504	380.50000	
3.0	161.72895	155.49749	153.76695	154.76541	157.08779	160.23810	164.10152	168.82255	174.76666	183.35735	201.96784	211.25854	222.25207	239.81950	267.25064	314.91805	385.26689	483.51146	628.85536	
3.5	161.86176	154.45564	154.04698	155.20756	159.85322	164.10152	169.70872	177.24789	189.82766	208.15405	236.80349	278.98920	340.24087	435.60036	584.56780	827.52780	1186.02100	1714.84810	2500.00000	
4.0	149.86176	153.82281	154.76541	159.06311	162.73122	168.82255	177.24789	192.94619	211.25854	242.50620	304.47474	374.12675	468.63320	603.59780	821.83336	1154.63284	1657.15070	2382.50000	3400.00000	
4.5	151.46642	153.76695	154.80465	160.23810	168.33682	174.76666	189.83095	211.25856	257.21332	347.25094	476.31622	662.12342	944.75483	1377.35282	2044.12254	3086.61678	4686.61678	7000.00000	10000.00000	
5.0	157.24340	153.85608	157.08779	162.73122	170.63382	183.35735	208.15405	250.67498	324.50009	441.61446	603.83336	821.83336	1124.64242	1574.18410	2244.67880	3244.67880	4744.67880	6744.67880	9244.67880	
5.5	156.28702	154.24336	158.54789	166.58573	175.98392	201.96784	266.83349	364.73474	516.31082	701.63446	973.26628	1347.08282	1917.34029	2707.31778	3817.30208	5267.30208	7167.30208	9667.30208	12967.30208	
6.0	154.82248	155.42812	162.07854	170.61038	196.60382	266.83349	364.73474	516.31082	701.63446	973.26628	1347.08282	1917.34029	2707.31778	3817.30208	5267.30208	7167.30208	9667.30208	12967.30208	17267.30208	
6.5	154.45564	156.20756	164.10152	177.24789	208.15405	283.96920	382.86008	507.76008	672.66008	907.56008	1212.46008	1607.36008	2102.26008	2797.16008	3692.06008	4886.96008	6381.86008	8276.76008	10671.66008	
7.0	154.13679	157.08779	166.33082	183.35735	224.76236	304.67236	403.58236	522.49236	671.40236	860.31236	1099.22236	1408.13236	1797.04236	2305.95236	2994.86236	3883.77236	5072.68236	6661.59236	8650.50236	
7.5	154.92281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	
8.0	154.82281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	
8.5	154.82281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	
9.0	154.82281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	
9.5	154.82281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	
10.0	154.82281	158.05681	168.82286	192.94619	250.15836	339.06836	457.97836	606.88836	795.79836	1034.70836	1343.61836	1752.52836	2261.43836	2970.34836	3879.25836	5068.16836	6557.07836	8445.98836	10734.89836	

가속상수와 반복처리시간에 따른 전체처리결과를 구하였으나 너무 방대한 양이므로 기준시간(9sec)에서의 가속변수에 따른 영상신호처리의 처리오차(Mean Square Error)를 그림(2)에서 도식화하고 표(2)에 자료를 정리하였다.

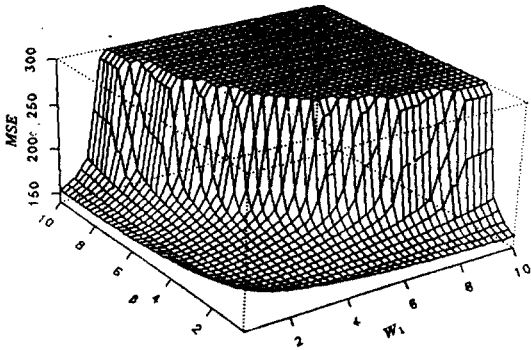


그림 2. 기준시간(9초)에서의 가속상수에 따른 영상신호처리의 처리오차(MSE)비교

Fig. 2. Compared MSE of image singal process by the standard time (9sec)

그림 [3]은 원영상과 훼손된 영상을 나타내었다.

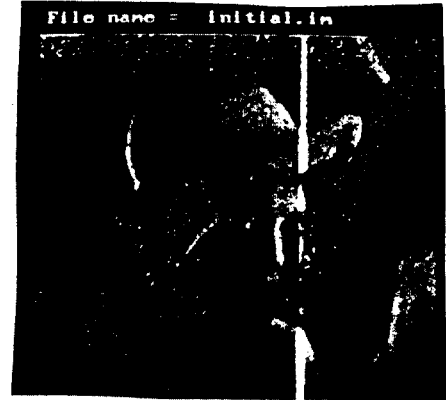
여기서 (c)는 Tikhonov-Miller의 초기근사해를 위해 초기화한 영상이다.



(a) original image



(b) oserved image(blurred-noisy image)



(c) initinal image

그림 3. 원 영상과 훼손된 영상

Fig. 3. Original image and noisy-blurred image for BSNR=20dB

그림 [4]는 기존의 반복 Tikhonov-Miller방법을 사용하여 처리영상을 반복횟수(K)에 따라 나타내었고 그림 [5]는 제안된 방법에 의해 처리된 영상을 나타내었다.



(a) k=1



(d) k=4



(b) k=2



(c) k=5



(e) k=3



(f) k=6



(g) k=7



(j) k=10

그림 4. 기존의 반복 Tikhonov-Miller 방법에 의한 복원영상(k 반복회수)

Fig. 4. Restored image by conventional(Tikhonov-Miller) iterative method (k iterative numbers)



(h) k=8



그림 5. 제안된 방법에 의한 복원영상

Fig. 5. Restored image by proposed method



(i) k=9

V. 결 론

신호는 그 형성, 감지, 기록, 전송의 단계를 거치는 동안 여러가지 원인에 의해 훼손되며, 훼손된 신호를 처리하여 원래의 입력신호와 가장 근사한 신호를 재현하는 기존의 많은 복원 방법은 여러가지 단점들이 발생하였다.

기준의 처리 방법들 중 가산잡음에 의해 훼손된 신호를 처리하는 여러가지 방법들을 선형성 호림에 의한 훼손을 처리하지 못함으로써 실제 응용에 많은 제약이 있으며, 호림에 의한 훼손을 처리하는 반복 복원방법들은 가산잡음이 있을 식 그 복원과정에서 가산잡음의 증폭을 초래하여 오히려 더 훼손된 결과를 가져온다.

순 호림 훼손과 가산잡음에 의한 훼손은 복원시 서로 상호적인 관계를 가진다. 또한, 호림 훼손 신호를 반복복원 할 때 약조건이나 특이점을 가진 경우 원상수준까지 않는 단점을 가지고 있으며, 이러한 약조건과 특이점 문제를 해결하기 위한 공정이 필요하다.

그러나, 이러한 공정은 전체 처리시간을 증가시키므로써 단점을 가진다. 본 논문에서 제안된 알고리즘은 호림훼손과 가산잡음이 있는 형태의 신호를 처리함에 있어 약조건이나 특이점에서도 수렴점인 Tikhonov-miller의 반복 복원 방법에 새로운 조건의 개념을 도입하여 반복처리를 하지 않고 직접적으로 수렴함으로써 다각된 신호처리시 각종 문제점이 되어 오 처리시간을 해결하였으며, 가산잡음에 의한 훼손을 원시화 감소시킨 방법이다. 본 논문에서 제시된 수학적 알고리즘은 기존의 알고리즘에 W_1 , β 의 값을 부가적으로 사용하였다.

이것은 기존의 알고리즘의 모든 장점들 가산잡음에 대한 처리, 특이점이나 약조건에서의 수렴성을 그대로 간직하되 정수만을 사용함으로써 메모리의 증가를 필요로 하지않고 반복처리시의 단점인 전체 처리시간을 고수화시킨 방법이다.

제안된 알고리즘과 Tikhonov-miller방법을 식스넷 함수에 의해 호리되고 20dB의 가산잡음이 있는 영상신호를 복원하는데 비교해 본 결과 전체 처리시간의 급격한 감소와 객관적인 평가 척도인 평균 제곱 오차(MSE)가 매우 줄인 유용한 방법임을 확인할 수 있었다.

참 고 문 헌

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

[2] J. Biemond, F. G. Von der Putten, and J. W. Woods, "Identification and restoration of images with symmetric non causal blur," IEEE Trans. Circuits Syst., vol. CAS 35, pp 385-394, 1988.

[3] P. H. Westerink, J. Biemond, and P. H. L. de Bruin, "Digital color image restoration," in Signal Processing III:Theories and Applications, I. T. Young et al., Eds. Amsterdam, The Netherlands:Elsevier North Hollands, 1986, pp. 761-764.

[4] A. L. Steven, W. Zucker, and A. Rossenfeld, "Iterative enhancement of noisy images,"IEEE Trans. Systems, Man and Cybernetics SMC 7, 1977, pp. 435-441.

[5] H. J. Trussell and M. R. Civanlar, "The feasible solution to non-blind restoration," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP 32, pp. 1001-1011, 1984.

[6] A. K. Katsaggelos, J. Biemond, R. M. Mersereau, and R. W. Schafer, "A general formulation of constrained iterative image restoration algorithms," in Proc. IEEE Int. conf. Acoust., Speech, Signal Processing 1985, Tampa, FL, 1985, pp. 700-703.

[7] K. Miller, "Least squares methods for ill posed problems with a prescribed bound," SIAM J. Math. Anal., vol. 1, pp. 52-74, 1970.

[8] Reginald L. Lagendijk and Jan Biemond, "Regularized iterative image restoration with ringing reduction," IEEE Trans. ASSP, vol. 36, NO. 12, December 1988.

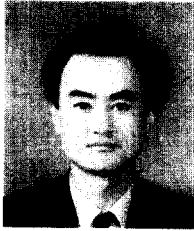
[9] A. N. Tikhonov and V. Y. Arsenin, solutions of ill posed problems, New York:Wiley, 1977.

[10] Lee W. Johnson, R. Dean Riess:"Numerical Analysis", Addison-Wesley Publishing Company, 1982.

[11] Richard L. Burden, J. Douglas Faires, Albert C. Reynolds:"Numerical Analysis", Prindle, Weber & Schmidt Boston, Massachusetts, 1981.

[12] Melvin J. Maron:"Numerical Analysis", Macmillan publishing Co., Inc. New York.

[13] Francis S. ed:"Numerical Analysis", Schaum's outline ser.



權起弘(Kee Hong Kwon) 정회원

1989년 : 영남대학교 공과대학
전자공학과 학사

1991년 : 영남대학교 공과대학
전자공학(전자통신
신공)석사

1991년 : 영남대학교 공과대학
전자공학(전자통신 전
공) 박사과정 입학

1991년 ~ 현 : 신일전문대학 전자계산과 전임강사 재직