IMAGE RESTORATION USING NEURAL NETWORKS

M. Teles de Figueiredo *José M. N. Leitão*

Centro de Análise e Processamento de Sinais - Instituto Nacional de Investigação Científica
Departamento de Engenharia Electrotécnica e de Computadores - Instituto Superior Técnico
Lisboa - PORTUGAL

ABSTRACT

In this paper, two neural algorithms for image restoration are proposed. The image is considered degraded by linear blur and additive white Gaussian noise. *Maximum a posteriori* estimation and regularization theory applied to this problem lead to the same high dimension optimization problem. The developed schemes, one having a sequential updating schedule and the other being fully parallel, implement iterative minimization algorithms which are proved to converge. The robustness of these algorithms with respect to finite numerical precision is studied. Examples with real images are presented.

1 INTRODUCTION

Restoration of blurred and noisy images is an *ill-conditioned* or even *ill-posed* problem [1], [2], [3]. Bayesian estimation and regularization theory are two of the classical formulations where *a priori* information is incorporated in order to overcome this difficulty. However, both lead to very large dimension optimization problems [1], [2], [3], [4], [5].

Consider the observed image $\mathbf{y}$ modelled as a linearly blurred (LB) and noisy version of the original image $\mathbf{x}$,

$$\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{n}, \qquad (1)$$

where $\mathbf{B}$ is the $MN \times MN$ blur matrix, and $\mathbf{n}$ is a white Gaussian noise (WGN) vector with $MN \times MN$ covariance matrix $\sigma^2\mathbf{I}$. In (1), $M \times N$ pixels images are represented by MN-dimensional vectors

The original image $\mathbf{x}$ is considered as a sample of a zero mean Gauss-Markov random field (ZMGMRF), with covariance matrix $\mathbf{A}$.

The *maximum a posteriori* (MAP) estimation criterion, corresponding to the above assumptions, yields the following minimization problem:

$$\hat{\mathbf{x}}_{\mathrm{MAP}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} - \mathbf{b}^T\mathbf{x} \right\}. \qquad (2)$$

In (2), $\mathbf{C}$ is a $MN \times MN$ symmetric positive definite matrix (SPDM) with positive diagonal elements [5], and $\mathbf{b}$ is a MN-dimensional vector given by

$$\mathbf{C} = \mathbf{A}^{-1} + \frac{1}{2\sigma^2}\mathbf{B}^T\mathbf{B}, \quad \text{and} \quad \mathbf{b} = \frac{1}{2\sigma^2}\mathbf{B}^T\mathbf{y}. \qquad (3)$$

Application of the Tikhonov-Miller regularization approach would involve a minimization problem equivalent to (2) [3], [5].

Neural-like structures have been proposed to solve large scale optimization problems. The mapping of the function to be minimized into the energy of a given network is the commonly adopted strategy. In the field of image restoration, some recent publications have suggested Hopfield-like neural networks [6], [7]. The standard Hopfield network proposed in [6] violates the nonzero autoconnection assumption, which prevents convergence from being guaranteed. In [7], a modified Hopfield network with nonzero autoconnections and two new updating schemes (one sequential and the other parallel) are introduced. However, the convergence proof for the parallel scheme is based on an almost never satisfied condition.

In this paper, a different approach is taken: instead of mapping the function to be minimized into the energy of a predefined network, neural implementations of previously developed iterative restoration schemes are introduced. These algorithms are suitable for distributed implementation, the key feature being that the updating of each element depends only on local information [5].

In steepest descent methods, optimal step size depends on, and has to be known by all sites. Although presenting better convergence rate, they are not suitable for fully parallel implementation by neural-like networks [8], [10].

Throughout this paper the following notation will be used

Euclidean vector norm:	$\| \mathbf{v} \|_2$
Maximum vector norm:	$\| \mathbf{v} \|_\infty = \max_i\{\lvert v_i \rvert\}$.

We we will also use the following induced matrix norm

$$\| \mathbf{A} \|_2 = \rho(\mathbf{A}) = \max_i\{|\lambda_i(\mathbf{A})|\} \tag{4}$$

where $\rho(\mathbf{A})$ is the *spectral radius* of $\mathbf{A}$ (assumed symmetric), and $\{\lambda_i(\mathbf{A})\}$ is the set of its eigenvalues [9], [10].

The following facts, for any MN-dimensional vector $\mathbf{v}$, and SPDM $\mathbf{A}$, will also be invoked

$$\| \mathbf{v} \|_2 \leq \sqrt{MN} \| \mathbf{v} \|_\infty \tag{5}$$

$$\| \mathbf{A} \|_2 = \max_{\|\mathbf{u}\|_2 = 1} \{\mathbf{u}^T \mathbf{A} \mathbf{u}\}. \tag{6}$$

The *condition number* of a SPDM is defined as

$$\kappa(\mathbf{A}) = \frac{\rho(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \geq 1 \tag{7}$$

where $\lambda_{\min}(\mathbf{A})$ stands for the minimum eigenvalue of $\mathbf{A}$. A matrix is said well conditioned if this number is not too large [9], [10].

2 PROPOSED ALGORITHMS

Under the assumptions of the previous Section, image restoration involves the minimization of a huge dimension quadratic form, which can be achieved by solving $\mathbf{Cx} = \mathbf{b}$. The dimension of this system demands iterative schemes [3], [5].

Splitting matrix $\mathbf{C}$ as $\mathbf{C} = \mathbf{G} - \mathbf{H}$ leads to the equivalent system $\mathbf{Gx} = \mathbf{Hx} + \mathbf{b}$ which suggests the iteration

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{G}^{-1}(\mathbf{Hx}(t) + \mathbf{b}) \\ &= \mathbf{x}(t) - \mathbf{G}^{-1}(\mathbf{Cx}(t) - \mathbf{b}) \end{aligned} \tag{8}$$

with some initial condition $\mathbf{x}(0)$ [9], [10]. Matrix $\mathbf{G}$ has to be easily invertible (e.g. diagonal or triangular). Defining the error vector $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{C}^{-1}\mathbf{b}$, it follows that $\mathbf{e}(t) = (\mathbf{G}^{-1}\mathbf{H})^t \mathbf{e}(0)$; therefore, iteration (8) converges if and only if matrix $\mathbf{M} \equiv \mathbf{G}^{-1}\mathbf{H}$ is *convergent* [9], i.e.

$$\lim_{t \to \infty} (\mathbf{G}^{-1}\mathbf{H})^t = 0 \quad \Leftarrow \quad \| \mathbf{G}^{-1}\mathbf{H} \| < 1, \tag{9}$$

for some matrix norm $\| \cdot \|$. Matrix $\mathbf{M}$ is convergent if and only if $\rho(\mathbf{M}) < 1$. The rate of convergence of such algorithms is defined as

$$R = -\log \rho(\mathbf{G}^{-1}\mathbf{H}) \tag{10}$$

and verifies $\log \| \mathbf{e}(t) \| \leq \log \| \mathbf{e}(0) \| - t\, R$.

2.1 Sequential Update

Taking matrix $\mathbf{G}$ as the lower triangular part of $\mathbf{C}$ yields the Gauss-Seidel algorithm [9], [10]. To implement it, the following neural structure and updating scheme are proposed.

Neural Algorithm 1 *Define a network of MN continuous valued elements $\{x_i,\ i = 1, 2, \ldots MN\}$ assigned to each image pixel. Let $W_{ij} = -C_{ij}/C_{ii}$ be the interconnection strength between elements i and j, and $I_i = b_i/C_{ii}$ be a bias input to each element. Let a cyclic sequential visiting schedule to the elements be given, and the network be initialized with any finite state. At time t an element x_i, chosen according to the visiting schedule, updates its state as*

$$x_i(t+1) = x_i(t) + \sum_{j=1}^{MN} W_{ij} x_j(t) + I_i.$$

It can be shown that this algorithm (Gauss-Seidel) converges if the system matrix is positive definite and has positive diagonal elements [9], [10], as is the case.

Note: We proved in [5] that the *iterated conditional modes* (ICM) algorithm [4], under the ZMGMRF-LB-WGN assumptions, is equivalent to the Gauss-Seidel scheme. A relation between ICM and the iterative solution of a system of equations was already recognized in [4], although no use was made of it to prove convergence.

2.2 Simultaneous Update

The distributed structure of neural networks will be fully explored if, instead of updating just one element at each step, all of them change state simultaneously.

Let

$$\mathbf{G} = \text{diag}\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{MN}\}. \tag{11}$$

Convergence conditions on the parameters ε_i are given by the following theorem:

Theorem 1 *Let $\mathbf{Cx} = \mathbf{b}$ be the system to be solved. Iteration (8), with $\mathbf{G}$ given by (11) converges if*

$$\varepsilon_i > \frac{1}{2} \sum_{j=1}^{MN} |C_{ij}|, \quad \forall_{i=1,2\ldots,MN}. \tag{12}$$

See proof in Appendix. If conditions (12) were satisfied for the diagonal elements C_{ii},

$$C_{ii} > \frac{1}{2} \sum_{j=1}^{MN} |C_j|, \tag{13}$$

or

$$C_{ii} > \sum_{j=1, j\neq i}^{MN} |C_{ij}|, \tag{14}$$

then, the corresponding Jacobi algorithm ($\mathbf{G}$ = diag($\mathbf{C}$), see [9] or [10]) would converge. But (14) is a diagonal dominance condition on $\mathbf{C}$ which can not be guaranteed *a priori*.

Iterative scheme (8), with $\mathbf{G}$ given by (11), is neurally implemented as follows.

Neural Algorithm 2 *Define a network of MN continuous valued elements $\{x_i,\ i = 1, 2, \ldots MN\}$ assigned to each image pixel. Let $H_{ij} = -C_{ij}/\varepsilon_i$ be the interconnection strength between elements i and j, and $J_i = b_i/\varepsilon_i$ be a bias input to each element i. Let all the ε_i satisfy condition (12). At each iteration t all the elements update their states simultaneously, according to*

$$x_i(t+1) = x_i(t) + \sum_{j=1}^{MN} H_{ij}x_j(t) \ + \ J_i.$$

To study the effect of parameters ε_i on the convergence rate, let us consider the simpler case $\varepsilon_i = \varepsilon, i = 1, 2, \ldots, MN$, for which convergence condition (12) reduces to $\varepsilon > \rho(\mathbf{C})/2$. It can be shown (see [9]) that the best convergence rate is obtained with

$$\varepsilon = \varepsilon_{\text{opt}} \equiv \frac{\rho(C) + \lambda_{\min}(\mathbf{C})}{2} = \frac{\rho(\mathbf{C})}{2}\left(\frac{\kappa(\mathbf{C})+1}{\kappa(\mathbf{C})}\right).$$

Conclusion: if $\mathbf{C}$ is well conditioned, i.e. $\lambda_{\min}(C) \simeq \rho(\mathbf{C})$, it is advantageous to increase ε up to $\varepsilon_{\text{opt}} \simeq \rho(\mathbf{C})$. If not, i.e. $\lambda_{\min}(C) \ll \rho(\mathbf{C})$, then $\varepsilon_{\text{opt}} \simeq \rho(\mathbf{C})/2$ and the convergence rate can not be much increased.

Note: In [7], a parallel algorithm and its convergence proof are presented. The main condition is $C_{ii} \geq \rho(\mathbf{C})$, for all diagonal elements C_{ii} of matrix $\mathbf{C}$. But $C_{ii} \leq \rho(\mathbf{C})$, for $i = 1, 2, \ldots, MN$, is a simple corollary of (6). Note that, if $\mathbf{C}$ is a SPDM, then

$$\rho(\mathbf{C}) = \max_{\|\mathbf{x}\|_2=1} \{\mathbf{x}^T\mathbf{C}\mathbf{x}\} \geq \mathbf{e}_i^T\mathbf{C}\mathbf{e}_i = C_{ii} \qquad (15)$$

for any $i = 1, 2, \ldots, MN$, where $\mathbf{e}_i$ is a vector with a 1 in position i and zero elsewhere, which clearly satisfies $\| \mathbf{e}_i \|_2 = 1$.

2.3 The effect of finite numerical precision

Hardware implementations or computer simulations of neural networks for image processing purposes use a small number of bits per pixel (typically 8 or 10 bits integers). We now examine the effect of finite numerical precision on the proposed algorithms. Consider that intermediate computations can be made with infinite precision but that each neuron (pixel) can only store integer values. Both schemes can be written as

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \mathcal{R}\left(\mathbf{G}^{-1}(\mathbf{C}\mathbf{x}(t) - \mathbf{b})\right) \qquad (16)$$

where $\mathcal{R}(\cdot)$ is an operator that rounds each coordinate of its vector argument to the nearest integer. Note that any $\mathbf{u} = \mathcal{R}(\mathbf{v})$ always has a strictly positive projection on $\mathbf{v}$, unless $\mathbf{u} = 0$, that is if $\| \mathbf{v} \|_\infty < \frac{1}{2}$. This fact implies that the updated vector $\mathbf{x}(t+1)$ moves closer to the solution $\mathbf{x}^* \equiv \mathbf{C}^{-1}\mathbf{b}$, unless

$$\| \mathbf{G}^{-1}(\mathbf{C}\mathbf{x}(t) - \mathbf{b}) \|_\infty < \frac{1}{2}. \qquad (17)$$

This condition defines the fixed points of iteration (16). The important issue is to find the distance between these fixed points and the solution $\mathbf{x}^*$. From (17) and (5) one can conclude that

$$\| \mathbf{G}^{-1}\mathbf{C}(\mathbf{x} - \mathbf{x}^*) \|_2 < \frac{\sqrt{MN}}{2},$$

or

$$\| \mathbf{x} - \mathbf{x}^* \|_2 < \frac{\sqrt{MN}}{2} \| \mathbf{G}\mathbf{C}^{-1} \|_2 . \qquad (18)$$

- In Neural Algorithm 1 (Gauss-Seidel), matrix $\mathbf{G}\mathbf{C}^{-1}$ is completely determined by matrix $\mathbf{C}$, and so are the convergence rate and the error bound (18). For this scheme, the error bound can be rewritten as

  $$\| \mathbf{x} - \mathbf{x}^* \|_2 < \frac{\sqrt{MN}}{2}\frac{\rho(\mathbf{C})}{\lambda_{\min}(\mathbf{C})} = \frac{\sqrt{MN}}{2}\kappa(\mathbf{C})$$

  since matrix $\mathbf{G}$, being the lower triangular part of $\mathbf{C}$, satisfies $\rho(\mathbf{G}) = \max\{C_{ii}\} \leq \rho(\mathbf{C})$. Also note that $\| \mathbf{C}^{-1} \|_2 = 1/\lambda_{\min}(\mathbf{C})$. In conclusion: the error bound is proportional to the condition number of matrix $\mathbf{C}$.

- In Neural Algorithm 2 the parameters ε_i can be adjusted. Let us again consider the simpler case $\varepsilon_i = \varepsilon, i = 1, 2, \ldots, MN$. The error bound (18), depends on ε, and can be written as

  $$\| \mathbf{x} - \mathbf{x}^* \|_2 < \frac{\varepsilon\sqrt{MN}}{2\,\lambda_{\min}(\mathbf{C})} = \frac{\varepsilon\sqrt{MN}}{2\rho(\mathbf{C})}\kappa(\mathbf{C}).$$

  Again, as with the sequential algorithm, the error bound depends on the condition number of matrix $\mathbf{C}$; the better it is conditioned, the smaller the error. Also, when ε in increased to ε_{opt} the error bound increases, i.e. there is a tradeoff between convergence rate and error bound.

3 EXAMPLES

In the examples of Figures 1 and 2 a first order ZMGMRF was adopted. Since the sequential update algorithm is equivalent to ICM (of which there are many examples in the literature) only results of Neural Algorithm 2 are presented. It was simulated on a conventional computer using 512×512 pixels, 8 bits/pixel, images. The intermediate values are stored as 8 bits integers. Both figures show restored windows of artificially degraded images.

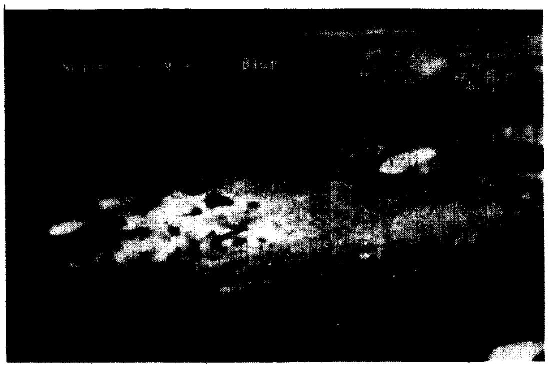

Figure 1: Blur: 9 × 9 uniform low pass. Although no noise was added, $\sigma^2 = 1$ was assumed.

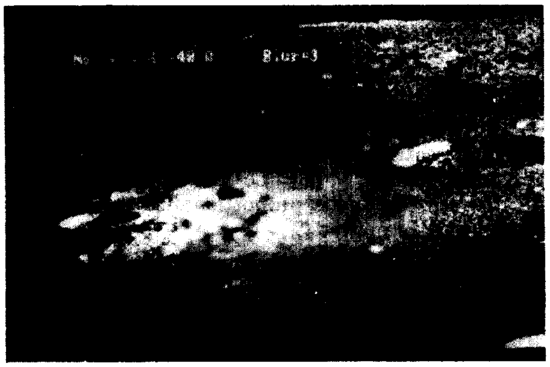

Figure 2: Blur: 3 × 3 uniform low pass. Noise variance $\sigma^2 = 40^2$.

APPENDIX

The proof of Theorem 1 is based on the following theorem (see [9]):
Theorem 2 *Let* $\mathbf{G}$ *be a non-singular and symmetric matrix, and* $\mathbf{C} = \mathbf{G} - \mathbf{H}$ *be positive definite. Then,* $\mathbf{M} = \mathbf{G}^{-1}\mathbf{H}$ *is convergent if and only if* $\mathbf{Q} \equiv \mathbf{G} + \mathbf{H} = 2\mathbf{G} - \mathbf{C}$ *is positive definite.*

According to Theorem 2, $\mathbf{M}$ is convergent if and only if $\mathbf{Q}$ is positive definite. A sufficient condition is diagonal dominance. Since the elements of $\mathbf{Q}$ are

$$Q_{ij} = \begin{cases} 2\varepsilon_i - C_{ii} & \Leftarrow \quad i = j \\ -C_{ij} & \Leftarrow \quad i \neq j, \end{cases}$$

the diagonal dominance condition on matrix $\mathbf{Q}$ is

$$2\varepsilon_i - C_{ii} > \sum_{j=1, j\neq i}^{MN} |C_{ij}|$$

or

$$\varepsilon_i > \frac{1}{2}\sum_{j=1}^{MN} |C_{ij}|,$$

since $C_{ii} > 0$. This concludes the proof of Theorem 1.

References

[1] H. C. Andrews and B. R. Hunt. *Digital Image Restoration*. Prentice Hall, New Jersey, 1977.

[2] M. Bertero, T. Poggio, and V. Torre. "Ill-posed problems in early vision". *Proc. of the IEEE*, vol. 76(8):869–889, August 1988.

[3] A. Katsaggelos, J. Biemond, R. Schafer, and R. Mersereau. "A regularized iterative image restoration algorithm". *IEEE Trans. on SP*, vol. 39, pp. 914–929, 1991.

[4] J. Besag. "On the statistical analysis of dirty pictures". *Journal of the Royal Statistical Socieiy B*, vol. 48(3):259–302, 1986.

[5] M. Teles de Figueiredo and José M. N. Leitão. "Sequential and parallel iterative image restoration". In *Proc. of the* 13^{th} *GRETSI Symposium on Signal and Image Processing*, Juan-les-Pins, France, 1991.

[6] Y. Zhou, R. Chellappa, A. Vaid, and B. Jenkins. "Image restoration using a neural network". *IEEE Trans. on ASSP*, vol. ASSP-36, pp. 1141–1151, July 1988.

[7] J. Paik and A. Katsaggelos. "Image restoration using the Hopfield network with nonzero autoconnections". In *Proc. of the ICASSP*, pp. 1909–1912, 1990.

[8] T. Simchony, R. Chellappa, and Z. Lichtenstein. "Pyramid implementation of optimal-step conjugate-search algorithms for some low-level vision problems". *IEEE Trans. on SMC*, vol. SMC-20, pp. 1408–1425, 1989.

[9] E. Isaacson and H. Keller. *Analysis of Numerical Methods*. John Wiley & Sons, New York, 1966.

[10] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation. Numerical Methods*. Prentice Hall, New Jersey, 1989.