Pattern Recognition Letters 3 (1985) 65–69
North-Holland
January 1985

Noisy image restoration by cost function minimization

Eli HAROUCHE, Shmuel PELEG*, Haim SHVAYTSER*, Larry S. DAVIS
University of Maryland, Center for Automation Research, College Park, MD 20742, USA

Received 19 June 1984

Abstract: Implementation of a new approach to the nonlinear noisy image restoration problem, which is feasible for large images, is described. The approach is based on a vector space representation of images and on using the conjugate gradient algorithm to solve a least squares minimization problem. Computer simulations yield good results with relatively little computational cost.

Key words: Image restoration, nonlinear restoration, picture vector space.

1. Introduction

Image restoration is often defined as a process of recovering an original image from a degraded version. Whenever exact restoration is not feasible (this happens, for example, when random noise is involved in the distorting process), the restoration problem becomes an approximation problem. This paper treats the case where the degraded image is the sum of the original image and stationary white noise.

When the original image is unknown, a cost function to be minimized can be $f(x)=\|F(X)-Y\|^2$ where Y is the observed blurred image, X is the restored image, and F is the blurring function. In [2,4] Peleg and Shvaytser proposed a method for restoring images based on the above minimization function using the conjugate gradient method as the minimization technique, when images are represented in the log-ratio space [3]. The above minimization function is not valid when noise is present, and its use actually amplifies the noise. We suggest generalizing the objective function by adding a term whose minimization would suppress the noise effect.

Out of the multitude of existing restoration techniques, Weiner filtering [5,6,7] is outstanding in being optimal in the mean-square sense. However, its computation involves division by coefficients that can be zero, and uses assumptions about the blur being linear and uniform, and the noise being additive and uncorrelated. The methods proposed here avoid these difficulties.

The support of the National Science Foundation under grant MCS-82-18408 is gratefully acknowledged.

* Permanent address: Dept. of Computer Science, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel. These authors were also supported by a grant from the Israel Academy of Sciences.

2. Representation of the restoration problem

Regarding pictures as vectors the deblurring problem can be formulated as follows: Find a picture X such that

$$F(X)=Y \qquad (1)$$

where Y is the blurred picture and F is the blurring operator. In [2,4] there is a discussion regarding the problem of finding $X=F^{-1}(Y)$; the difficulties include:

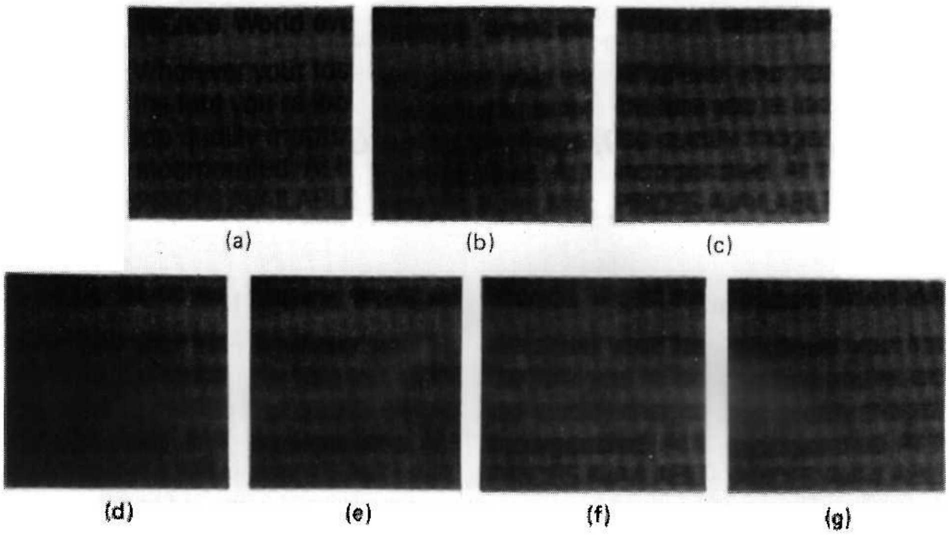


Figure 1. Test images. (a) Original pictures. (b) Image blurred by Gaussian blur, $\sigma = 20$, 5×5 convolution matrix. (c) Image blurred by motion blur (5×1 convolution matrix). (d) Gaussian blur with noise, SNR 50:1. (e) Gaussian blur with noise, SNR 10:1. (f) Motion blur with noise, SNR 50:1. (g) Motion blur with noise, SNR 10:1.

(a) Nonuniqueness: In general F is not 1-1.

(b) Noise, which is always present.

(c) Inaccurate knowledge of the blurring operator.

(d) High dimensionality, where pictures of size 512×512, for example, belong to a vector space of dimension 2^{18}.

(e) Out of range values: Digital pictures are represented by numerical values in a finite range $[0, M]$. Solving equation (1) without introducing range constraints results in out of range values in the solution.

Regarding out of range values, projection back into the picture range might destroy the solution. Two widely used types of transformations to bring matrices into the picture domain are: Truncating values greater than M and smaller than 0, and linear transformation into the range $[0, M]$. However, some difficulties arise when applying either one of the above transformations. Truncation will results in a significant loss of accuracy which is especially crucial in iterative approaches. Linear transformation cannot be used together with the vector space representation, since the matrix addition and scale multiplication do not obey the laws of a vector space.

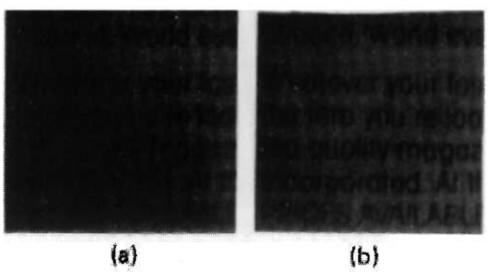


Figure 2. Restoration without added noise (15 iterations). (a) Restoration of (1.b). (b) Restoration of (1.c).

2.1. Log-ratio space representation

In [2, 3, 4] Shvaytser and Peleg proposed a representation of picture values, called the log-ratio representation, that overcomes the out-of-range difficulty. Given a grey level g in the range $(0, M)$ the ratio $g/(M-g)$ is in the range $(0, \infty)$ and $\log(g/M-g))$ is in the range $(-\infty, \infty)$. The resulting transformation is

$$x_{ij} = \Psi(X_{ij}) = \log\left(\frac{X_{ij}}{M - X_{ij}}\right)$$

and the reverse transformation is

$$X_{ij} = \Psi^{-1}(x_{ij}) = M\,\frac{e^{x_{ij}}}{1 + e^{x_{ij}}},$$

where X_{ij} is the pixel's grey level, and x_{ij} is the value in log-ratio space. Basic picture operations for the log-ratio space are defined as follows: Let

$$X = (X_{ij}), \qquad Y = (Y_{ij}),$$

then

$$Z = X + Y \quad \text{iff} \quad Z_{ij} = \Psi^{-1}(\Psi(X_{ij}) + \Psi(Y_{ij})),$$

$$Z = \alpha X \quad \text{iff} \quad Z_{ij} = \Psi^{-1}(\alpha \Psi(X_{ij})).$$

Properties of the log-ratio transformation can be found in [3].

3. Restoring noisy images

When restoring blurred noisy images, there is a trade-off between sharpness and noise level. In the presence of noise, when we tried to minimize

$$f(X) = \|F(X) - Y\|^2$$

we obtained X such that $F(X) \approx Y$; however, X was too noisy. A minimization function that will take into account the presence of noise should be introduced:

$$\text{Min} \quad f(X) = \|F(X) - Y\|^2 + \alpha \|\Omega(X)\|^2$$

over all X, where $\Omega(X)$ is an operator that measures the magnitude of the noise. Several possibilities for the choice of Ω are discussed in the sequel. The conjugate gradient algorithm [9] was very efficient for our optimization problems.

When applying gradient methods like the conjugate gradient method, the gradient $f(x)$ has to be computed. We assume that the operator $F(X) = Y$ can be described locally, i.e. each pixel Y_{ij} is obtained from X_{ij} and its neighbors. Let the operator on X_{ij} and its neighbors be L_{ij}, say

$$L_{ij} = g(X_{11}, X_{12}, \ldots, X_{mn})$$

and the operation $F(X) = Y$ transforms (X_{ij}) to (L_{ij}). After the log-ratio transformation $(x_{ij}) = (\Psi(X_{ij}))$ is transformed into (l_{ij}) where

$$l_{ij} = \Psi(g(\Psi^{-1}(x_{11}), \Psi^{-1}(x_{12}), \ldots, \Psi^{-1}(x_{mn})).$$

The detailed computation of the gradient is given in [1] and [2].

3.1. Laplacian

A known operator for noise indication is the Laplacian on a 3×3 neighborhood: $\Omega(X) = \nabla^2 X$ where

$$[\nabla]_{ij} = X_{i,j} - \frac{X_{i+1,j} + X_{i-1,j} + X_{i,j+1} + X_{i,j-1}}{4}$$

and

$$\|\Omega(X)\|^2 = \|\nabla^2 X\|^2 = \sum_{i,j=1,1}^{m,n} |\nabla^2 X|_{i,j}^2$$

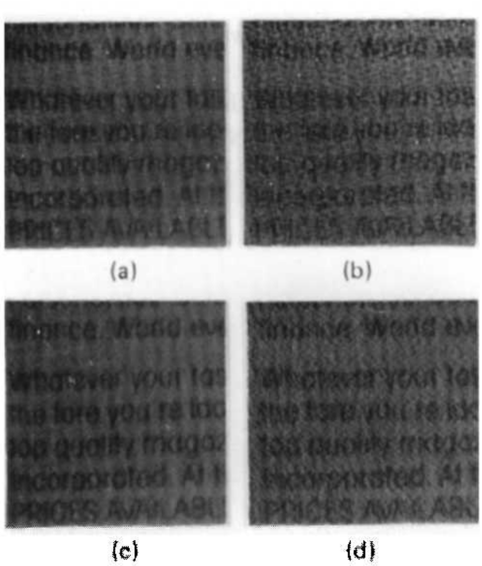


Figure 3. Noisy picture restoration with no constraint (15 iterations). (a) Restoration of (1.d). (b) Restoration of (1.e). (c) Restoration of (1.f). (d) Restoration of (1.g).

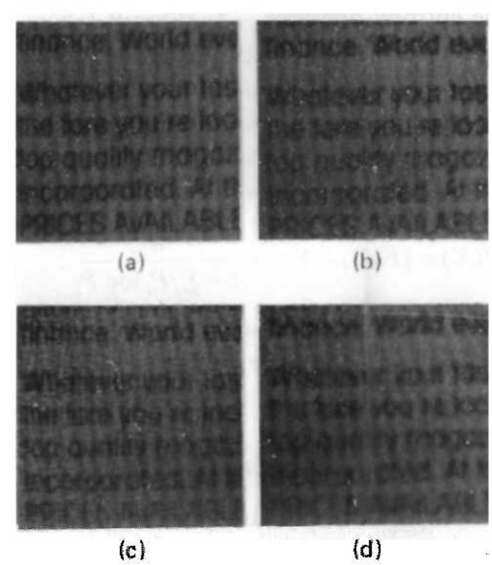


Figure 4. Using the Laplacian. (a) (1.f), $\alpha = 0.005$, 8 iterations. (b) (1.g), $\alpha = 0.001$, 15 iterations. (c) (1.e), $\alpha = 0.001$, 8 iterations. (d) (1.e), $\alpha = 0.001$, 8 iterations.

where m,n are the dimensions of the restored picture.

The Laplacian can be computed either in log-ratio space or in the grey level representation. Applying the Laplacian in log-ratio space will result in placing higher emphasis on the noise at the grey level bounds 0 and M, and lower emphasis around $M/2$, which agrees with visual perception.

3.2. *The product noise sensitive operator*

Another 3×3 noise sensitive operator is

$$\Omega(X_{ij}) = \left(X_{ij} - \frac{X_{i-1,j}+X_{i+1,j}}{2}\right) \times \left(X_{ij} - \frac{X_{i,j-1}+X_{i,j+1}}{2}\right).$$

This operator is zero for any pixel that is the average of either its vertical or horizontal neighbors. Compared to the Laplacian, this product operator yields high values for the 'noisy' pixels and smaller values for edges, and will preserve edges better. For this operator, as with the Laplacian operator, $\|\Omega(X)\|^2$ will be computed in the log-ratio space since it has better visual interpretation.

3.3. *Maximum entropy*

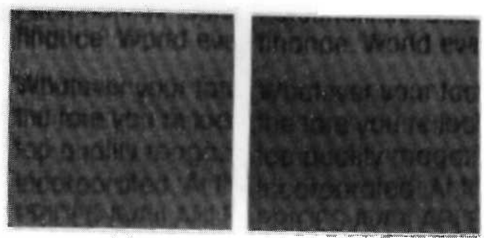
The entropy operator $-\sum_i P_i \log P_i$ where $P_i = X_i/\sum_n X_n$ and X_i is the pixel value at i, is often used to measure the smoothness of an image. It yields a maximum for an image with all pixels equal. See [8] for maximum entropy restoration methods. With entropy we minimize the function

$$f(X) = \|F(X) - Y\|^2 + \frac{\alpha}{\varepsilon - \sum_i P_i \log P_i}.$$

Unlike the previous constraint operators, the entropy is a global operator, and its gradient can be computed faster.

3.4. *Computation speedup*

To speed up the computation of ψ, ψ^{-1} and ψ' we use tables. See [1] for details. As an initial guess the zero picture in the log-ratio sense, or the original blurred picture, could be taken. All examples in this paper use the latter guess.



(a) (b)

Figure 5. Using the product operator. (a) (1.d), $\alpha=0.0005$, 8 iterations. (b) (1.f), $\alpha=0.0005$, 8 iterations.

4. Time complexity

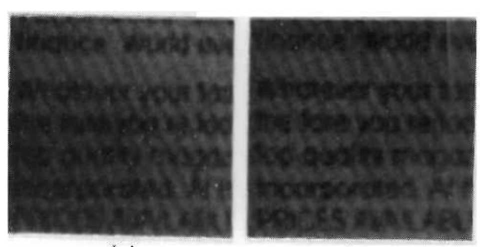
Assuming convergence in a constant number of iterations, the asymptotic behavior of the method is $O(mn^2)$, where m is the size of the matrix for the blurring operator, and n is the width of the image. These results are comparable even to methods that do not use range constraints. In our experiments on 120×120 images, implemented on a VAX 11/780, each iteration took approximately one minute when applying any approach, and the number of iterations that was needed for reasonable results was 5–8.

5. Simulation results

The following points were tested:

(1) Sensitivity to noise. White noise was added to the blurred image with SNR 10:1, SNR 50:1, and SNR 100:1 where SNR is defined as

$$\frac{\mathrm{Var(signal)}}{\mathrm{Var(noise)}}.$$



(a) (b)

Figure 6. Using the maximum entropy. (a) (1.d), $\alpha=0.5$, 15 iterations. (b) (1.f), $\alpha=0.5$, 15 iterations.

(2) Sensitivity to various kinds of blurring functions. Functions that were tested were:

(a) Uniform motion blur, simulated by giving each pixel a directed average of its neighbors.

(b) Gaussian blur with PSF

$$g(x, y, \alpha, \beta) = \exp\left(- \frac{(x-\alpha)^2 + (y-\beta)^2}{\sigma^2} \right).$$

6. Concluding remarks

In all our experiments the method always converged with reasonable accuracy in a *small* and *fixed* number of iterations, making the computation time small and proportional to the picture size. When restoring blurred noisy images, no knowledge about the noise is used. These points, and the fact that the restored images are of a good quality, give our method some advantage over other methods when the PSF is known but no a priori knowledge about the noise is given. However, a way of finding the optimal weight α for each approach is desired.

References

[1] Harouche, E. (1984). Noisy image restoration by cost function minimization. TR-1363, Center for Automation Research, University of Maryland, MD.

[2] Shvaytser, H. and S. Peleg (1983). Image restoration using vector space representation. *IPA Conference on Image Processing, Graphics, an Pattern Recognition,* Beersheva, pp. 1–14.

[3] Shvayster, A. and S. Peleg (1983). Pictures as elements in a vector space, *IEEE Conference on Computer Vision and Pattern Recognition,* Washington D.C., pp. 442–446.

[4] Shvaytser, H. and S. Peleg (1983). Inversion of picture operators. *IEEE International Conference on Systems, Man and Cybernetics,* Bombay, pp. 202–206.

[5] Rosenfeld, A. and A.C. Kak (1982). *Digital Picture Processing.* Academic Press, New York–London.

[6] Pratt, W.K. (1970). *Digital Image Processing.* Wiley, New York.

[7] Pratt, W.K. and F. Davarian (1977). Fast computational techniques for pseudo-inverse and Wiener image restoration, *IEEE Trans. Comput.* 27, 571–580.

[8] Burch, S.F., S.F. Gull and J. Skilling (1983). Image restoration by a powerfull maximum entropy method. *Computer Vision, Graphics, and Image Processing* 23, 113–28.