FAST IMAGE RESTORATION WITH THE HUBER-MARKOV PRIOR MODEL

Stéphane Pelletier and Jeremy R. Cooperstock

McGill University
Department of Electrical and Computer Engineering
Montréal, H3A 2A7, Canada

ABSTRACT

Image restoration is an ill-posed problem that must be regularized in order to reduce noise amplification in the restored image. Although quadratic penalty terms allow for fast restoration algorithms based on the Fast Fourier Transform (FFT), they often lead to images whose discontinuities are not well preserved. On the other hand, edge-preserving penalty terms can produce better results at the expense of computational efficiency. A restoration technique exploiting the Woodbury matrix identity was recently presented [5]. However, its performance decreases when the number of discontinuities become significant. To overcome this problem, we propose the use of a nonlinear conjugate gradient method in conjunction with a circulant preconditioner that can be updated quickly at each iteration. Experiments with simulated and real data are employed to demonstrate the effectiveness of our approach.

***Index Terms*—** Image restoration, edge-preserving regularization, Huber prior, preconditioning

1. INTRODUCTION

High quality images are desired and often required in various applications since they can provide details that are critical to the success of certain tasks. Image restoration attempts to reverse the degradation process undergone by a blurred and noisy image in order to recover the original image [1]. Since this task is *ill-posed* at worst and *ill-conditioned* at best, it must be reformulated into a better conditioned problem to reduce noise amplification in the restored image. Although quadratic penalty terms allow for fast algorithms based on the Fast Fourier Transform (FFT), they often yield images whose discontinuities are not well preserved [2]. To overcome this difficulty, edge-preserving penalty terms are often employed [3][4]. However, the problem is then shift-variant and the computational efficiency of the FFT cannot be exploited.

A restoration technique exploiting the Woodbury matrix identity was recently presented [5], in which the restored image is obtained by adding the results of two independent restorations problems; one produces an image that comes directly from a FFT restoration, whereas the other involves solving a system of linear equations whose size is proportional to the number of edges in the image. In some imaging scenarios, the number of edges, and in turn, the computational complexity, can be sufficiently high to preclude the use of this method.

As a solution to this problem, we show that a simple nonlinear conjugate gradient method with a circulant preconditioner, updated at each iteration, can provide significant speed improvement. An edge-preserving convex penalty term based on the Huber function [6] is employed. Furthermore, the computational efficiency and memory requirements of the proposed method do not depend on the number of edges in the restored image.

2. BACKGROUND

2.1. Problem formulation

The problem consists of restoring an $N \times N$ image from a blurred and possibly noisy image of the same size. The image formation model, which expresses the relationship between the degraded image and the original image, is often assumed to be linear and can be expressed in matrix form as:

$$\underline{y} = \mathbf{G}\underline{x} + \underline{n}, \tag{1}$$

where $\underline{y} \in \mathbb{R}^{N^2}$ is a vector whose elements are the observed pixels, $\mathbf{G}$ is a matrix describing the imaging geometry, $\underline{x} \in \mathbb{R}^{N^2}$ is a vector representing the (unknown) original image and $\underline{n}$ is an additive Gaussian noise vector. One way of finding a regularized solution to Equation (1) consists of minimizing the penalized least-squares cost function $\Phi : \mathbb{R}^{N^2} \rightarrow \mathbb{R}$, which is defined as:

$$\Phi(\underline{x}) = \frac{1}{2}[\underline{y} - \mathbf{G}\underline{x}]^T[\underline{y} - \mathbf{G}\underline{x}] + \lambda r(\underline{x}), \tag{2}$$

where $r : \mathbb{R}^{N^2} \rightarrow \mathbb{R}$ is a regularizing penalty function [7] and λ is a parameter that controls the tradeoff between consistency with the observed data and agreement with *a priori* knowledge about the solution.

In order to facilitate the minimization process, convex functions are often employed to define the regularizing term

This work was generously supported by a Precarn scholarship.

r. One useful edge-preserving convex penalty term is based on the Huber function [6], which we define as

$$\psi_T(x) = \begin{cases} \frac{x^2}{2}, & |x| \leq T \\ T|x| - \frac{T^2}{2}, & |x| > T \end{cases} . \quad (3)$$

This function quadratically penalizes small discontinuities in the image, which are often associated with noise, whereas large discontinuities (actual edges) are linearly penalized. Finite difference approximations to second-order derivatives are used as the image smoothness measure at pixel $x_{k,l}$, which are defined as:

$$\begin{aligned} d_{k,l,0}^T \underline{x} &= x_{k,l+1} - 2x_{k,l} + x_{k,l-1} \\ d_{k,l,1}^T \underline{x} &= \frac{1}{2}(x_{k-1,l+1} - 2x_{k,l} + x_{k+1,l-1}) \\ d_{k,l,2}^T \underline{x} &= x_{k-1,l} - 2x_{k,l} + x_{k+1,l} \\ d_{k,l,3}^T \underline{x} &= \frac{1}{2}(x_{k-1,l-1} - 2x_{k,l} + x_{k+1,l+1}). \end{aligned} \quad (4)$$

The regularizing term r can then be expressed as

$$r(\underline{x}) = \sum_{m=0}^{3} \sum_{k=1}^{N^2} \psi_T([\mathbf{D}_m \underline{x}]_k), \quad (5)$$

where $\mathbf{D}_m$ is a $N^2 \times N^2$ matrix representing the convolution operator corresponding to $\sum_{k,l} d_{k,l,m}^T$.

2.2. Limits of Pan and Reeves's method

Pan and Reeves propose an iterative scheme in which the objective function (2) is majorized at each iteration by a quadratic function whose minimum determines the majorizing function in the next iteration. By repeating this step several times, the iterative process is guaranteed to converge to the minimum of Φ; see [5] for more details. At each iteration, minimizing the quadratic function is achieved by solving a system of linear equations of the form:

$$\Big(\mathbf{G}^T \mathbf{G} + \lambda \sum_{m=0}^{3} \mathbf{D}_m^T \Gamma_m \mathbf{D}_m\Big) \underline{x} = \mathbf{G}^T \underline{y}, \quad (6)$$

where Γ_m is an $N^2 \times N^2$ diagonal matrix whose k^{th} element is:

$$\Gamma_m^{(k)} = \begin{cases} 1, & |[\mathbf{D}_m \underline{x}]_k| \leq T \\ \frac{T}{|[\mathbf{D}_m \underline{x}]_k|}, & otherwise \end{cases} . \quad (7)$$

The Woodbury identity is employed to accelerate the computation of the solution of Equation (6). The bottleneck of this approach is the solution to a dense system of K unknowns, where K is the number of elements in $\{\Gamma_m\}_{m=0}^3$ that are not equal to 1. Therefore, the effectiveness of this optimization depends on the magnitude of K, which is assumed to be of the order of N.

3. PROPOSED ALGORITHM

Since the cost function (2) is convex, its minimization is equivalent to finding the zero of its gradient:

$$-\nabla\Phi(\underline{x}) = \mathbf{G}^T[\underline{y} - \mathbf{G}\underline{x}] - \lambda \sum_{m=0}^{3} \mathbf{D}_m^T \underline{z}^{(m)}(\underline{x}), \quad (8)$$

where $\underline{z}^{(m)} : \mathbb{R}^{N^2} \to \mathbb{R}^{N^2}$ is defined by

$$z_k^{(m)}(\underline{x}) \triangleq \dot{\psi}_T([\mathbf{D}_m \underline{x}]_k).$$

We propose to use the preconditioned nonlinear conjugate gradient (PNCG) method with the Fletcher-Reeves update rule [8], in which each iteration involves the following operations:

$$\begin{aligned} \underline{r}^n &= -\nabla\Phi(\underline{x}^n) \text{(gradient; see (8))} \\ \underline{p}^n &= \mathbf{M}^{-1} \underline{r}^n \text{(preconditioner)} \\ \beta_n &= \frac{< \underline{r}^n, \underline{p}^n >}{< \underline{r}^{n-1}, \underline{p}^{n-1} >} \text{(Fletcher-Reeves rule)} \\ \underline{d}^n &= \underline{p}^n + \beta_n \underline{d}^{n-1} \text{(search direction)} \\ \alpha_n &= \operatorname*{argmin}_{\alpha} \Phi(\underline{x}^n + \alpha \underline{d}^n) \text{(line search)} \\ \underline{x}^{n+1} &= \underline{x}^n + \alpha_n \underline{d}^n \text{(solution update)} \end{aligned}$$

3.1. Preconditioner $\mathbf{M}$

The preconditioner $\mathbf{M}$ should normally approximate the Hessian matrix of the objective function at x^n, i.e., the current iterate [3]. Since the Huber function is not twice continuously differentiable, this matrix is not defined everywhere. To circumvent this difficulty, we approximate it by the Hessian matrix of the quadratic majorizing function, which is given by the coefficient matrix of Equation (6). We then define the preconditioner as

$$\mathbf{M} = \mathbf{G}^T \mathbf{G} + \lambda\beta \sum_{m=0}^{3} \mathbf{D}_m^T \mathbf{D}_m, \quad (9)$$

where β is the average of the diagonal elements of $\{\Gamma_m\}_{m=0}^3$. Although one might consider adapting the combined diagonal/circulant preconditioner proposed by Fessler and Booth [3] to cope with the diagonal matrices $\{\Gamma_m\}_{m=0}^3$, we performed a few experiments that did not produce satisfying results. Finally, we remark that since $\mathbf{G}$ and $\mathbf{D}$ have a block circulant with circulant block structure, the proposed preconditioner can be inverted efficiently using the FFT.

3.2. Line search

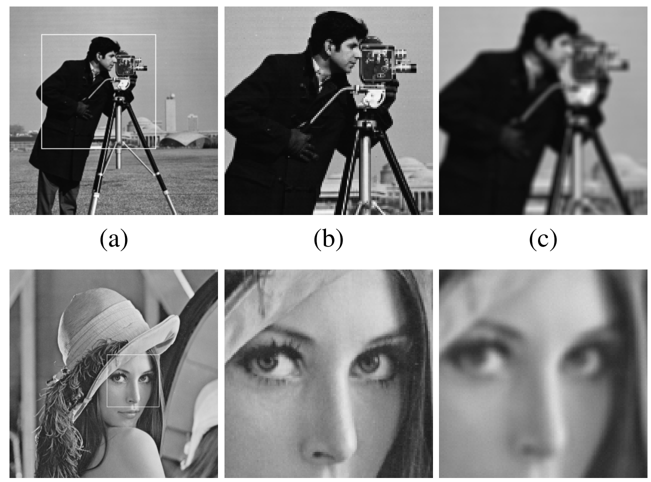
At each PNCG iteration, a *line search* is employed to find the minimum of Φ along the current search direction $\underline{d}^n$. Although there exist general-purpose methods for doing so, we use an approach that exploits the fact that the cost function (2) is a sum of quadratic functions and Huber functions. More specifically, we apply a technique similar to the iterative majorizing approach used by Pan and Reeves; the main difference is that in our case, the majorization is performed in one dimension only, namely along the current search direction. Typically, only three or four iterations are necessary to minimize Φ.

4. EXPERIMENTS

The effectiveness of the proposed preconditioner for accelerating the minimization of the cost function (2) is demonstrated using both simulated and real image data. The images shown in Fig. 1(a) and Fig. 1(d) are employed as ground-truth images in the simulation part. To facilitate later comparison with restored images, the regions inside the boxes are enlarged and shown in Fig. 1(b) and Fig. 1(e). Two degraded images are obtained by blurring the ground-truth images using a 5×5 Gaussian blur kernel with variance 2; noise is added to the second blurred image (Lena) such as to obtain a signal-to-noise ratio (SNR) of 40 dB. Fig. 1(c) and Fig. 1(f) show the regions in the degraded images corresponding to Fig. 1(b) and Fig. 1(e) respectively. Fig. 2(a) shows a real blurred image captured with a camera that was deliberately adjusted to be slightly out of focus; the region identified by the box is enlarged and shown in Fig. 2(b). The unknown camera PSF was determined roughly by trial and error and is assumed to be a 12×12 Gaussian blur kernel with variance 1.8.

Fig. 3 presents six restoration experiments whose inputs are the degraded images of Fig. 1 and Fig. 2. All computations are done using Matlab on a Dual-Core AMD Opteron Processor 2216 machine. In each experiment, the same degraded image is restored twice; once with the nonlinear conjugate gradient (NCG) algorithm, then using the PNCG algorithm with the proposed preconditioner. Since both algorithms employ the same values of λ and T, their restored images are *almost* identical and only one of them is shown per restoration scenario. The actual difference between the methods is the speed at which the restorations are performed; to demonstrate this, plots showing the value of the cost function versus time using both approaches are provided. Also, the number of unknowns K of the dense matrix that would need to be solved with Pan and Reeve's method is indicated, along with the space in memory required to store it. Clearly, the amount of memory required would certainly render inefficient their method in the experiments considered here.

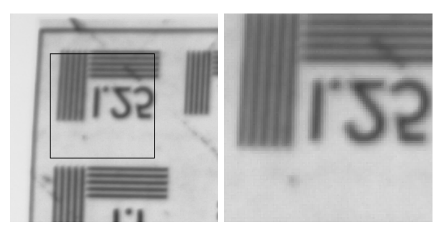
In all six scenarios, one can see that the proposed preconditioning approach provides an improvement in the convergence rate over the unpreconditioned algorithm. However, one can observe that for higher values of λ, the difference in performance between NCG and PNCG diminishes. This is due to the fact that increasing the weight of the regularization prior in the cost function makes the initial problem better conditioned, which reduces the need for preconditioning. However, this tends to produce smoother images, which might not always be desirable. On the other hand, reducing the value of λ too much can amplify noise in the restored image. It is important to mention that preconditioning does not make the restored image look better or worse, it just accelerates the computation of the solution. The values of the parameters must be chosen by taking into account the level of noise in the observed data, a problem that is not addressed in this paper.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Simulated degraded images. (a) First ideal image (256×256). (b) Region indicated in a). (c) First degraded image (5×5 Gaussian blur with var. 2). (d) Second ideal image (512×512). (e) Region indicated in d). (f) Second degraded image (5×5 Gaussian blur with var. 2 + 40dB SNR Gaussian noise).

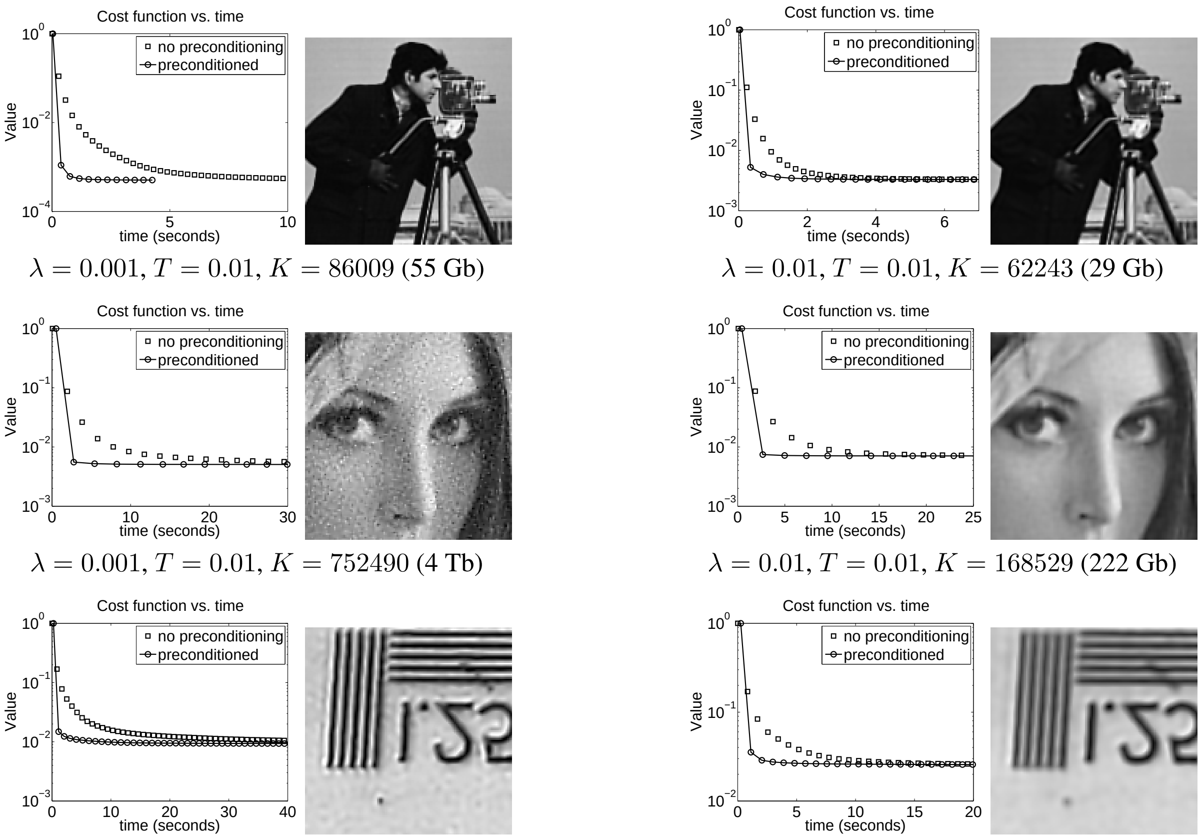


(a) (b)

Fig. 2. Degraded image from a real camera. (a) Captured image (256×256). (b) Region within the box in a).

5. CONCLUSIONS

In this paper, a method for accelerating the computation of image restoration problems involving an edge-preserving



$\lambda = 0.001, T = 0.01, K = 86009$ (55 Gb)

$\lambda = 0.01, T = 0.01, K = 62243$ (29 Gb)

$\lambda = 0.001, T = 0.01, K = 752490$ (4 Tb)

$\lambda = 0.01, T = 0.01, K = 168529$ (222 Gb)

$\lambda = 0.001, T = 0.005, K = 162848$ (198 Gb)

$\lambda = 0.01, T = 0.01, K = 55856$ (23 Gb)

Fig. 3. Effect of proposed preconditioner on the convergence rate of six restoration scenarios. For each experiment, the value of the cost function versus time (left) and the restored image (right) are shown, along with the values of λ and T. The number of unknowns K in the dense matrix that would need to be solved with Pan and Reeve's method is also indicated.

Huber-Markov prior was presented. It was shown that this technique can be effective when the number of edges is high.

6. REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, 2002.

[2] S. J. Reeves, "Optimal space-varying regularization in iterative image restoration," *IEEE Trans. Image Processing*, vol. 3, no. 3, pp. 319–324, May 1994.

[3] J. A. Fessler and S. D. Booth, "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction," *IEEE Trans. Image Processing*, vol. 8, no. 5, pp. 688–699, May 1999.

[4] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 298–311, Feb. 1997.

[5] R. Pan and S. J. Reeves, "Efficient Huber-Markov edge-preserving image restoration," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3728–3735, Dec. 2006.

[6] P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[7] J. Fessler, "Penalized weighted least squares image reconstruction for positron emission tomography," *IEEE Trans. Med. Imaging*, vol. 13, pp. 290–300, June 1994.

[8] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Tech. Rep., Carnegie Mellon University, Pittsburgh, PA, USA, 1994.