AN ALGORITHM FOR BLIND RESTORATION OF BLURRED AND NOISY IMAGES

Nader Moayeri and Konstantinos Konstantinides
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304-1120
moayeri,konstant@hpl.hp.com

Abstract

This paper presents a technique for deblurring noisy images. It includes two processing blocks, one for denoising and another for blind image restoration. The denoising step is based on the theories of singular value decomposition and compression-based filtering. The deblurring step is based on a double-regularization technique. Experimental results show that the combination of these techniques is quite effective in restoring severely blurred and noise-corrupted images, without prior knowledge of either the noise or image characteristics.

1 Introduction

Image deblurring is a difficult problem even when there is exact knowledge of the blur degradation. In this paper, we consider a linear space-invariant model for the blur, where the blur point spread function (PSF) and the noise strength are assumed to be unknown. The most effective deblurring technique known to the authors for this framework is the one recently proposed by You and Kaveh [1]. This is an iterative technique which simultaneously estimates the blur PSF and restores the image. In [1], the performance of this technique is shown for a few cases including one where the blur PSF is a uniform function on a 3 × 3 support and the signal-to-noise ratio (SNR) is 30dB. In this paper, we consider the case of a uniform 5 × 5 blur PSF and SNR's as low as only 10dB. Our approach for restoring such severely degraded images is still based on You and Kaveh's technique. However, we show that the performance of this technique is enhanced if it is preceded by an explicit denoising step. This is in addition to the implicit denoising operation inherent in You and Kaveh's technique.

For the denoising step, we use a block-based, non-linear filtering algorithm based on the theories of singular value decomposition (SVD) and compression-based filtering [3, 4]. The algorithm also employs an efficient method for estimating the noise power from the input data with no additional a priori information. A major characteristic of both the image deblurring and the denoising algorithms is their ability to preserve edge details.

In Sections II and III we provide brief descriptions of the deblurring and denoising algorithms employed in our proposed system. In Section IV we present experimental results and we compare our scheme with that of You and Kaveh's on images that are both severely blurred and noise corrupted. Some concluding remarks are given in Section V.

2 The Deblurring Component of Our System

In this paper, we assume an image degradation model of the form

$$g(\underline{n}) = (d \star f)(\underline{n}) + w(\underline{n}) \text{ ; for all } \underline{n} \in \Omega \ , \tag{1}$$

where $\underline{n} = (n_1, n_2)$ denotes the discrete coordinates for image pixels, f is the original image, d is the unknown blur PSF, $\star$ denotes the two-dimensional convolution, g is the observed (blurred) image, w denotes the additive noise present in that image, and Ω denotes the rectangular support both f and g. It is also assumed that the additive noise w is i.i.d. Gaussian and independent of both f and d. In addition, since there is no loss of energy in the imaging system, it is further assumed that the components of d are nonnegative real numbers that add up to one. That is,

$$\sum_{\underline{n} \in \mathcal{D}} d(\underline{n}) = 1 \ , \tag{2}$$

where $\mathcal{D}$ denotes the unknown support for the blur PSF.

The goal in blind image deblurring is to find good estimates for both the blur PSF d and the original image f, based on the observed image g in (1). Assuming that d is known, a solution to this problem is given in [2]. In [1], You and Kaveh extend that technique to the blind restoration case formulated above. In summary, their double regularization technique tries to minimize some cost function for estimating f and d as $\hat{f}$ and $\hat{d}$, drawn from a "reasonable" set of solutions. This set of solutions is defined by a regularization constraint that accounts for the fact that images are generally low bandwidth signals which do not exhibit much high frequency content, except in the vicinity of edges and in textured regions. This assumption is particularly valid for photographic blurs, where edges are absent. Hence, they use a second regularization constraint for the blur PSF d. The imposition of these regularization constraints also reduces the search space for solutions. Specifically, You and Kaveh's technique tries to minimize the cost functional

$$L(\hat{d},\hat{f}) = \frac{1}{2}\sum_{\underline{n}\in\Omega}\left[g(\underline{n}) - (\hat{d}\star\hat{f})(\underline{n})\right]^2 + \frac{\lambda}{2}\sum_{\underline{n}\in\Omega} W(\underline{n})\left[(c\star\hat{f})(\underline{n})\right]^2 + \frac{\gamma}{2}\sum_{\underline{n}\in\mathcal{D}}\left[(a\star\hat{d})(\underline{n})\right]^2 \ , \quad (3)$$

where a and c are the highpass filter regularization operators, W is a weight function, and λ and γ are the Lagrangian multipliers, better known as the regularization parameters. If we ignore the effect of W for the moment, the goal in the above minimization problem is to find an estimate $\hat{f}$ which makes the mean squared estimation error (the first term in (3)) small and yet would not allow $\hat{f}$ and $\hat{d}$ to have too much high frequency content. In other words, the second and the third summations in (3) are penalty terms for high frequency content in $\hat{f}$ and $\hat{d}$, respectively. The weight function W makes it possible to allow high frequency content in $\hat{f}$ in the high activity (edge and texture) regions and to heavily penalize such content in low activity (smooth) regions.

The cost functional defined in (3) is quartic in its variables, the samples of $\hat{f}$ and $\hat{d}$. To avoid this complication, an alternate minimization approach was used in [1]. In each iteration, $\hat{f}$ is first kept fixed and $L(\hat{d},\hat{f})$ is minimized over $\hat{d}$. Then, $\hat{d}$ is kept fixed at the value just found and $L(\hat{d},\hat{f})$ is minimized over $\hat{f}$. Each of the two subproblems solved in each iteration is quadratic in its variables and is solved with the conjugate gradient method, which is iterative itself. It is also necessary to impose the non-negativity and range constraints on both $\hat{d}$ and $\hat{f}$, as well as the constraint (2) on $\hat{d}$. These constraints are enforced after each subproblem has been solved, and not within the iterations of the conjugate gradient method. Yet another aspect of the problem is the uncertainty about the blur PSF support $\mathcal{D}$. You and Kaveh start with a conservative large support. As the algorithm goes through its iterations, one observes that the samples of $\hat{d}$ on its outer layers gradually vanish, and one can prune $\mathcal{D}$ accordingly. For pruning, we used a strategy similar to the one described in [1].

The method described above works reasonably well, except for two well known problems; (a) convergence to local minima and (b) lack of a good way of determining the values of the regularization parameters λ and γ. You and Kaveh propose a way of setting these parameters [1]. However, as they have pointed out in their paper, these values are meant to be used as guidelines only and not as exact values. The values we used are described in more detail in Section IV.

3 The Denoising Component of Our System

You and Kaveh's method includes an implicit denoising scheme, however, under severe noise degradation, we expect that an explicit noise filtering scheme should improve the overall performance of their algorithm. One of the major drawbacks of conventional filtering techniques is that they tend to blur the original image or they require some prior knowledge of the image or noise characteristics. In [3], the authors describe an efficient non-linear filtering scheme, based on the theories of singular value decomposition and compression-based filtering [4].

The main steps of this SVD-based filtering algorithm are:

1. Divide the noisy image g into non-overlapping blocks.

2. Perform SVD on each block, set to zero the singular values that are smaller than a threshold ϵ, and reconstruct a filtered-version of the original block using these new singular values.

Given an $m \times n$ block B, the SVD of B is given by

$$B = U_B \mathbf{\Sigma}_B V_B^t, \tag{4}$$

where U_B and V_B are orthogonal matrices and

$$\mathbf{\Sigma}_B = diag(\beta_1, \beta_2, ..., \beta_n) \tag{5}$$

is a diagonal matrix. The diagonal elements of $\mathbf{\Sigma}_B$ are called the singular values of B. Given a threshold ϵ, let $\mathbf{\Sigma}_B^{'}$ = $diag(\beta_1^{'}, \beta_2^{'}, ..., \beta_n^{'})$, where $\beta_i^{'}$ = β_i if $\beta_i > \epsilon$ and $\beta_i^{'}$ = 0 if $\beta_i \leq \epsilon$. Then, the filtered block $B^{'}$ is defined as $B^{'} = U_B \mathbf{\Sigma}_B^{'} V_B^t$.

The effectiveness of this algorithm depends on the accuracy of the estimate of the threshold ϵ. In this paper, we estimate ϵ using the following compression-based calibration scheme [3, 4].

1. Apply the filtering algorithm for various values of the threshold ϵ.
2. Compress each filtered image using a lossless compression algorithm, such as lossless JPEG.
3. Plot the compressed size as a function of ϵ. Let ϵ^* be the knee-point of the plot. Use $\epsilon = \epsilon^*$ as the threshold on the SVD filtering algorithm. If S denotes the compressed size of a filtered image, the knee-point is defined as the point at which the second derivative $\frac{d^2 S}{dlog(\epsilon)^2}$ is maximum.

Note that for a set of noisy images obtained under the same noise conditions, the calibration scheme has to be performed only once for a typical image in that set. A more detailed example is given in the next section.

4 Experimental Results

We performed two sets of experiments with two blurred and noise-corrupted images involved in each set. First, we tried to deblur the images with the deblurring algorithm of Section II alone. Then, we deblurred the same images by first passing them through the denoising algorithm of Section III and then applying the deblurring algorithm of Section II to the denoised images. The blurred images were obtained by artificially blurring the Cameraman image shown in Figure 1. This is a 256 × 256, 8-bit, gray-scale image, and it is commonly used in image deblurring experiments. We used the blur PSF

$$d(\underline{n}) = \begin{cases} 0.04 & ; \text{ if } |n_1| \leq 2, |n_2| \leq 2 \\ 0 & ; \text{ otherwise} \end{cases} \quad ,$$

and we further corrupted the blurred images with additive white Gaussian noise. The signal-to-noise ratio (SNR) for the two test images was 20dB and 10dB. The SNR is defined as

$$\text{SNR} = 10 \log_{10} \frac{\sigma_{d \star f}^2}{\sigma_w^2} \ ,$$

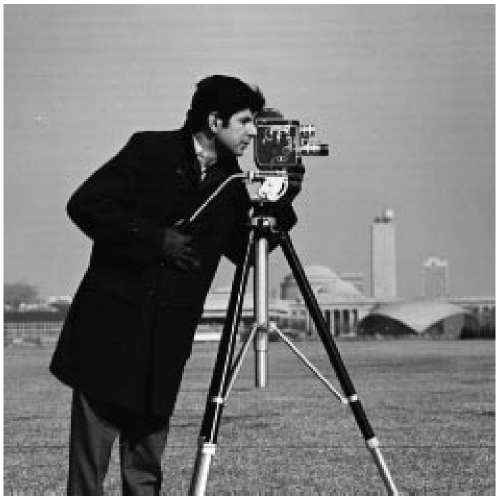


Figure 1: The original Cameraman image.

where σ_w^2 denotes the variance of the noise and $\sigma_{d\star f}^2$ denotes the variance of the blurred image.

Following the calibration scheme described in the previous section, each of the blurred and noisy images was filtered for various values of the threshold ϵ between 0 and 200. In all cases, we used an 8×8 block size. For SNR = 20 db, Figure 2 shows a plot of the normalized size of the compressed filtered image as a function of ϵ. Also shown is the second derivative of the curve with respect to $\log(\epsilon)$.

From Fig. 2, there are two knee-points; one at ϵ = 35 and one at ϵ = 125. The first one was expected from the theory of compression-based filtering [3], and denotes the point by which most of the noise has been removed. After this point, the increase in compression is less obvious, indicating the start of removing image information. We believe that the second knee-point is due to the blur in the original image. At this point, there is enough loss of image information so that the original blur plays no additional role in the compressibility of the filtered image. Therefore, we ignored the second knee-point and selected ϵ^* = 35 as the filtering threshold. Using a similar approach, for SNR=10 dB, we selected a filtering threshold of ϵ^* = 100.

In all cases, the deblurring algorithm was ran with an initial support $\mathcal{D} = \{-4, \ldots, 4\} \times \{-4, \ldots, 4\}$ for the blur PSF. This is larger than the actual support size for the blur PSF, since, in practice, the original size is considered unknown. A first estimate of the size of $\mathcal{D}$ can be obtained by inspecting the blurred image in the vicinity of an edge separating two regions of almost constant pixel intensities. If computations are not too excessive, then it is advisable to start the algorithm with a slightly larger support size than this first estimate.

For both regularization operators (the highpass filters c and a) we used the Laplace filter

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} .$$

This is actually one of the two filters suggested for this purpose in [1]. Following [2, 1], we used a

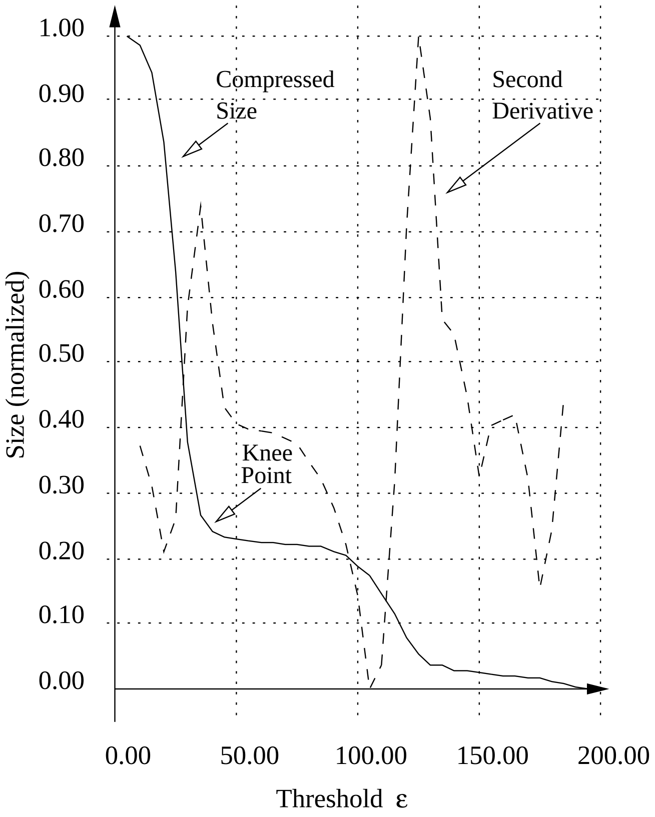


Figure 2: Size of compressed filtered image versus ϵ, and its second derivative. SNR = 20db.

weight function of the form

$$W(\underline{n}) = \frac{1}{1 + \alpha\sigma^2(\underline{n})} \; ; \text{ for all } \underline{n} \in \mathbf{\Omega} \, , \tag{6}$$

where $\sigma^2(\underline{n})$ is the local variance of the blurred image g at pixel $\underline{n}$. This variance is high in the edge and texture regions and low in smooth regions.

In (6), You and Kaveh use $\alpha = 1000/\sigma^2_{\text{max}}$, where σ^2_{max} is the maximum value for the local variance $\sigma^2(\underline{n})$. We found experimentally the α value at which $\frac{\max(W(\underline{n}))}{\min(W(\underline{n}))}$ = 2,000 is a good choice. This choice yields a restored image that has fairly sharp edges and yet it is not grainy in areas where it is supposed to be smooth. The local variance at pixel location $\underline{n}$ was computed over a 3 × 3 window centered at $\underline{n}$.

The results of our experiments are shown in Figures 3 and 4 for the two cases of 20dB SNR and 10dB SNR, respectively. Each of these figures shows: (a) the original blurred and noisy image, (b) the image obtained by applying only the denoising algorithm to the image in (a), (c) the image obtained by applying only the deblurring algorithm to the image in (a), and (d) the image obtained by applying the deblurring algorithm to the denoised version of the blurred image; that is, the image in (b). In all cases, the deblurring algorithm was run for 50 iterations, even though the results were equally good after 20 iterations. In the conjugate gradient algorithm, in order to find the best $\hat{f}$ for a given $\hat{d}$, we restricted the number of iterations to ten. To find the best $\hat{d}$ for a given $\hat{f}$, instead of using the conjugate gradient method, we used the direct method of (a) differentiating the cost function $L(\hat{d}, \hat{f})$ with respect to $\hat{d}$, (b) setting these derivatives to zero, and (c) solving the resulting system of linear equations for $\hat{d}$. Table 1 shows the values of the regularization parameters used to obtain each deblurred image. As expected, we had to use larger λ's (i.e., larger degrees of regularization) for noisier images.

Image in Figure	Regularization Parameters	
	λ	γ
2(c)	10^3	10^8
2(d)	2	10^8
3(c)	10^4	3×10^8
3(d)	10	3×10^8

Table 1: Regularization parameters used for obtaining the four deblurred images.

In all cases, the deblurring algorithm was able to estimate the blur PSF very accurately and the pruning of the blur PSF support $\mathcal{D}$ took place as predicted. In fact, the algorithm converges to the right size for $\mathcal{D}$ very fast. Regarding the output images from the deblurring algorithm, from Figs. 3 and 4, images 3(d) and 4(d) are perceptually better than the images 3(c) and 4(c), respectively. Therefore, we conclude that denoising does indeed improve the deblurring process when the blurred images are very noisy.

5 Conclusions

Noise in images may significantly affect the performance of even the best of the deblurring algorithms. In this paper, we presented an algorithm for the blind restoration of blurred and noisy images that combines the deblurring efficiency of You and Kaveh's algorithm with the filtering efficiency of an SVD-based filter. Our scheme requires no prior knowledge of either the image or noise characteristics and performs well even at low SNRs.

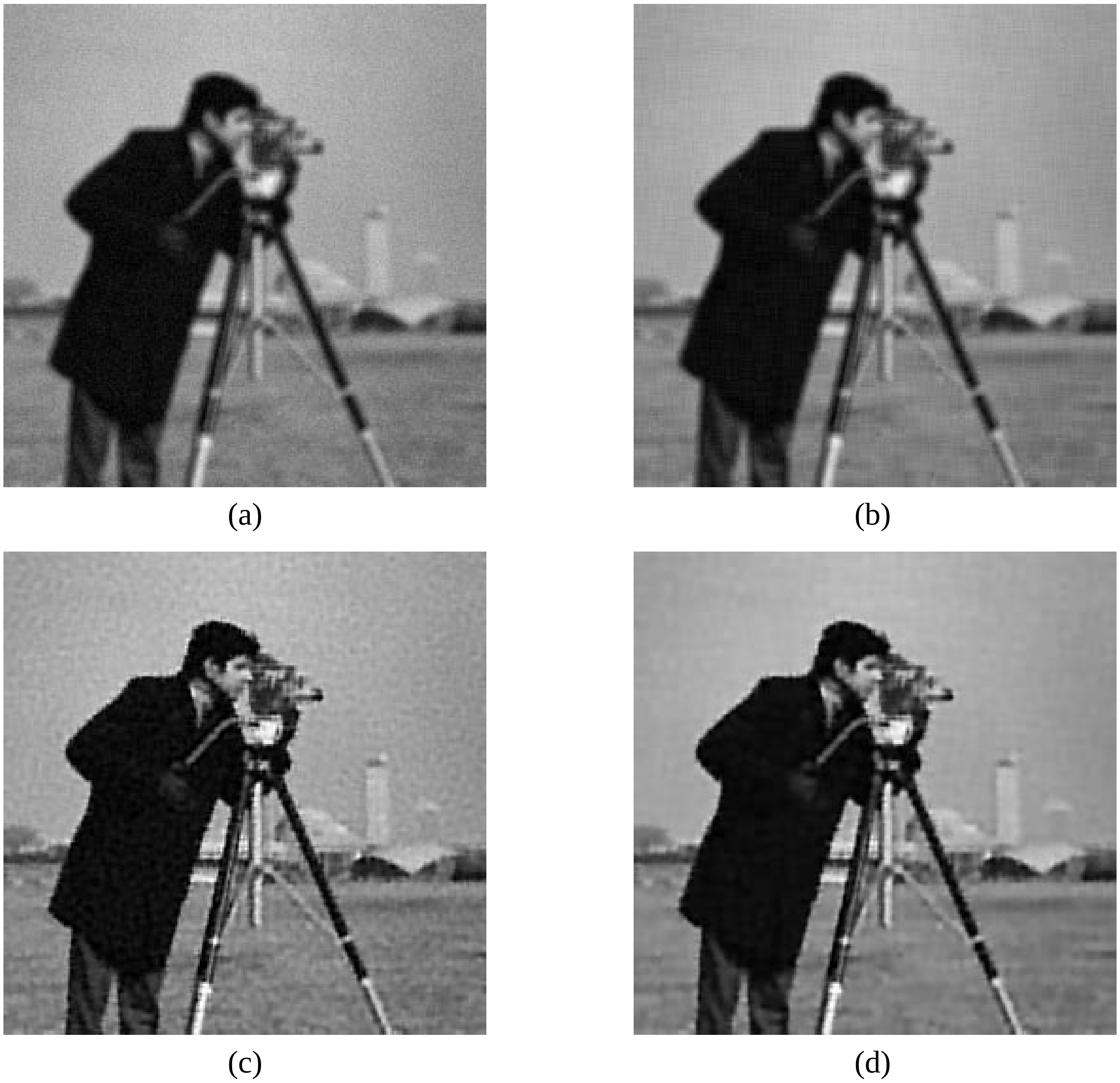
6 References

[1] Y.-L. You and M. Kaveh, “A regularization approach to joint blur identification and image restoration,” *IEEE Trans. Image Processing,* vol. IP-5, pp. 416-428, March 1996.

[2] R.L. Lagendijk, J. Biemond, and D.E. Boekee, “Regularized iterative image restoration with ringing reduction,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1874-1887, Dec. 1988.

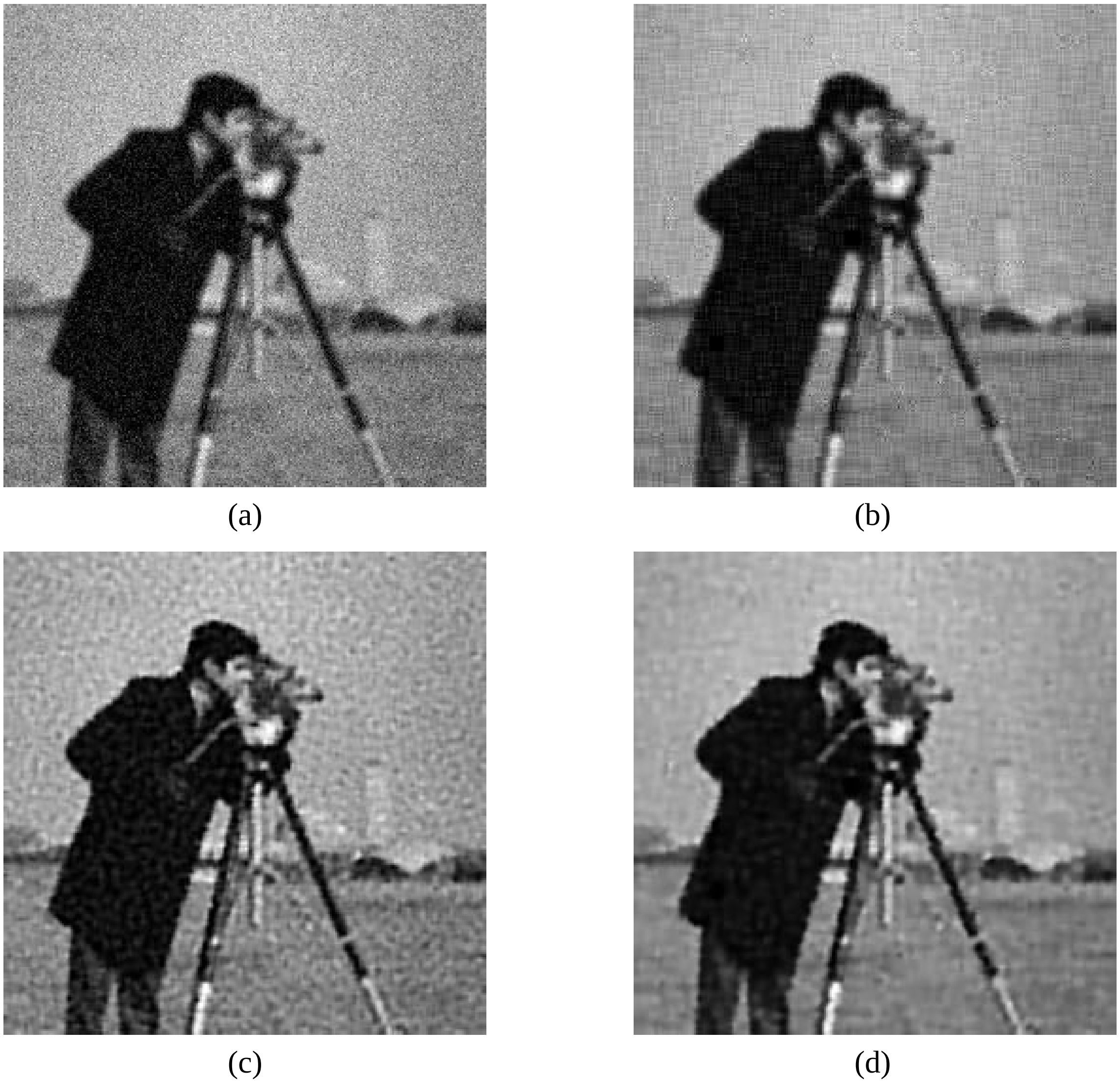
[3] K. Konstantinides and G.S. Yovanof, “Improved compression performance using SVD-based filters for still images,” *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science and Technology,* San Jose, CA, vol. 2418, pp. 100-106, Feb. 1995.

[4] B.K. Natarajan, “Filtering random noise from deterministic signals via data compression,” *IEEE Trans. Signal Processing*, vol. SP-43, pp. 2595-2604, Nov. 1995.



(a) (b) (c) (d)

Figure 3: (a) Image blurred with the 5 × 5 uniform blur PSF, SNR=20dB, (b) result of applying the denoising algorithm to the image in (a), (c) result of applying the deblurring algorithm to the image in (a), (d) result of applying the deblurring algorithm to the image in (b).



(a)

(b)

(c)

(d)

Figure 4: (a) Image blurred with the 5 × 5 uniform blur PSF, SNR=10dB, (b) result of applying the denoising algorithm to the image in (a), (c) result of applying the deblurring algorithm to the image in (a), (d) result of applying the deblurring algorithm to the image in (b).