International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.5 (2016), pp.389-398
http://dx.doi.org/10.14257/ijsip.2016.9.5.35

Split Bregman and Stationary Second-Degree Based Iterative Algorithm for Image Deconvolution

Su Xiao

School of Computer Science and Technology, Huaibei Normal University
Huaibei, 235000, China
csxiaosu@163.com

Abstract

This paper models image deconvolution as an l_2-l_1 minimization problem, which is an approach taken by many state-of-the-art image deconvolution algorithms. We present a novel iterative algorithm based on the split Bregman method and the stationary second-degree method, which efficiently addresses the classic convex minimization problem. The split Bregman method, which has been proven to be very efficient for non-differentiable minimization problems, decomposes the equivalent constrained version of the l_2-l_1 deconvolution problem into a series of sub-problems. These sub-problems are then individually solved using appropriate methods to obtain their closed-form solutions. Unlike the majority of other similar deconvolution algorithms, we use a modified stationary second-degree method to solve the l_2-l_1 denoising sub-problem, prompted by some recent work on the improvement of the iterative thresholding method. The presented algorithm can be categorized as a split Bregman method, so convergence of the solution can be guaranteed. In our experiment, the presented algorithm and the algorithms in references [6] and [8] are used to restore Gaussian-blurry and uniform-blurry images. The experimental results show that the presented algorithm is effective and it outperforms other algorithms in comparison.

Keywords: *Image deconvolution, l_2-l_1 minimization problems, split Bregman method, stationary second-degree method, denoising operators*

1. Introduction

Blur and noise are two adverse factors that degrade digital images; thus image deconvolution tasks must deblur and denoise the blurry images to restore sharp images. The degradation of sharp images can be modeled as $g=Hu+v$, where $g\in\mathrm{R}^M$ is the known blurry image in vector form; $H\in\mathrm{R}^{M\times N}$ denotes the linear operator; $u\in\mathrm{R}^N$ is the unknown sharp image or its sparse representation and $v\in\mathrm{R}^M$ is the additive Gaussian noise. H can generally not be inverted, so directly solving $g=Hu+v$ is impossible, especially when H is heavily ill-posed. To restore sharp images, image deconvolution is generally converted into a minimization problem $\min_u\{J(u)=\|g-Hu\|_2^2/2+\lambda\|Wu\|_1\}$ (we use **P0** to represent it below) which is also called an "l_2-l_1 deconvolution problem", where $J(u)$ is the objective function; $\|g-Hu\|_2^2/2$ is the data-fidelity term relative to the type of image noise; $\|\cdot\|_1$ denotes the l_1-norm; $\|Wu\|_1$ is the regularization term representing the prior knowledge about the sharp images; the constant $\lambda>0$ controls the balance between the data-fidelity and regularization; u is the sparse representation of the unknown sharp image; $W\in\mathrm{R}^{N\times N}$ is the diagonal weight matrix, *i.e.*, $W=\mathrm{diag}[w_1, w_2,\ldots, w_N]$ with $w_j>0$ for all js; and $\|g-Hu\|_2^2/2$ and $\lambda\|Wu\|_1$ are both convex, proper and lower semi-continuous. Thus $J(u)$ is a convex function, which means the set of solutions of problem **P0** is nonempty. Since u is a sparse representation of an unknown sharp image, **P0** belongs to the synthesis-based image deconvolution problem. In this case, $H=BD$, and B and D denote the blur operator and the

ISSN: 2005-4254 IJSIP

redundant dictionary, respectively. Once the minimizer u^* of problem **P0** has been calculated, the sharp image can be obtained by computing Du^*. For a convex minimization problem **P0**, Daubechies *et al.* [1] present an iterative thresholding algorithm (ITA) as $u^{n+1}=\Theta_{\lambda W}(u^n+H^T(g-Hu^n))$, where $\Theta_{\lambda W}$ is called the "denoising operator". When dealing with a denoising problem which has the form $0.5\|y-x\|_2^2+\lambda\|x\|_1$, where x and y denote the arbitrary signals (*e.g.*, images), ITA reduces to $x^{n+1}=\Theta_\lambda(y)$. The most common choice of denoising operator is perhaps the soft-thresholding function [2]. For example, Chan *et al.* [3] use the soft-thresholding function to solve one of their sub-problems. Due to its simplicity and guaranteed convergence [1], ITA is one of the most popular algorithms for the minimization problem **P0**. However, when H is heavily ill-posed and/or λ is too small or too large, it is difficult for ITA to achieve optimal results. To improve ITA, Wright *et al.* [4] use an adaptive continuation strategy [5] to simultaneously update the values of λ when iteratively reconstructing sparse signals. In contrast to the algorithm of Wright *et al.* which only employs one previous iteration in each subsequent iteration, the algorithm presented by Beck *et al.* [6] uses a linear combination of two previous iterations to compute u^{n+1}. To guarantee the convergence of their presented algorithm, Beck *et al.* use a well-designed expression for the linear combination coefficient and update this coefficient with each iteration. The experimental results show that compared to ITA, the algorithms of Wright *et al.* and Beck *et al.* and other variants of ITA truly have superior performance. Another popular algorithm for the minimization problem **P0** is forward-backward splitting (FBS) [7], which is typically suitable for minimization problems where the objective function is the sum of a differentiable term and a non-differentiable term. FBS addresses **P0** by iteratively computing: $u^{n+1}=u^n+\beta_n(\Phi_G(u^n+\alpha_n\nabla E(u^n))-u^n)$, where $E(u)=\|g-Hu\|_2^2/2$ and $G(u)=\lambda\|Wu\|_1$; Φ_G denotes the proximity operator; $\nabla E(u)$ is the gradient of $E(u)$; and α_n and β_n denote the step-size and relaxation parameter of the n^{th} iteration, respectively. If we let $\Theta_{\lambda W}$ and Φ_G represent the same operator and let $\alpha_n=1$ and $\beta_n=1$ for all ns, then the algorithm shown in above equation is the exact same as ITA, therefore FBS can be regarded as a generalized ITA. Therefore, for the minimization problem **P0**, ITA can be also expressed by a more general form $u^{n+1}=(1-\gamma)u^n+\gamma\Theta_{\lambda W}(u^n+H^T(g-Hu^n))$ with constant $\gamma>0$. As an operator splitting algorithm, the two main advantages of FBS are that it has proven convergence and it can decouple the differentiable terms and non-differentiable terms of the objective function. However, if the objective function of a minimization problem has more than one non- differentiable term, FBS cannot be applied to it. In addition to image deconvolution problems [8], FBS is also widely used to address some typical imaging inverse problems, such as image denoising [9], image inpainting [10] and so on.

To solve problem **P0** more efficiently, this paper presents a novel image deconvolution algorithm that is based on the split Bregman method (SBM) and the stationary second-degree method (SSDM) [11]. As a powerful tool, SBM is utilized to convert and split **P0** into a series of sub-problems. A modified version of SSDM is then applied to the denoising sub-problem to improve image deconvolution. The rest of the paper is organized as follows. Section 2 presents a brief review of SBM. The presented deconvolution algorithm and SSDM are described in Section 3. The experimental results are presented and discussed in Section 4. The last section provides the conclusions.

2. Review of SBM

SBM was first presented by Goldstein *et al.* [12] for image denoising and compressed sensing. Its applications were then expanded to other fields, such as image deconvolution [13], image segmentation [14] and so on. As the most popular member of the family of Bregman iteration methods, the major benefits of SBM are that it can be coded easily and it usually needs only several iterations to achieve convergence. To address the convex minimization problem

$$\min_{x} \; F_1(x) + F_2(Px) \,, \tag{1}$$

where $F_1(x)$ is differentiable, convex and proper; $F_2(Px)$ is non-differentiable, convex and proper; and P is an arbitrary linear operator, SBM first converts it into the equivalent constrained form

$$\begin{aligned} &\min_{x,y} \; F_1(x) + F_2(y) \\ &s.t. \quad y = Px \end{aligned} \,, \tag{2}$$

and then iteratively and alternately computes

$$\begin{cases} (x^{n+1}, y^{n+1}) = \arg\min_{x,y} \; F_1(x) + F_2(y) + \dfrac{\mu}{2} \| y - Px - d^n \|_2^2 \\ d^{n+1} = d^n + (Px^{n+1} - y^{n+1}) \end{cases} . \tag{3}$$

From Eq. (3), the optimality conditions of SBM can be obtained as follows

$$\begin{cases} 0 = \nabla F_1(x^{n+1}) + \mu P^{\mathrm{T}} (Px^{n+1} - y^n + d^n) \\ 0 = \nabla F_2(y^{n+1}) + \mu (y^{n+1} - Px^{n+1} - d^n) \\ d^{n+1} = d^n + (Px^{n+1} - y^{n+1}) \end{cases} . \tag{4}$$

It is has been proven that if the minimization problem (1) has at least one feasible solution x^*, then the following facts can be established

$$\begin{cases} \lim_{x \to \infty} F_1(x^n) + F_2(Px^n) = F_1(x^*) + F_2(Px^*) \\ \lim_{x \to \infty} \| x^n - x^* \| = 0 \end{cases} \tag{5}$$

which means that the convergence of SBM is guaranteed. For the sub-problems in Eq. (3), additional methods may be needed to obtain their closed-form solutions. As recently revealed by Setzer *et al.* [15], SBM can be interpreted as a Douglas-Rachford splitting algorithm applied to the dual of problem (1), and it can also be interpreted as an alternating direction method of multipliers (ADMM), which is a special case of the augmented Lagrangian method, applied to l_1-regularized minimization problems.

3. Presented Deconvolution Algorithm

To address the classic l_2-l_1 deconvolution problem **P0**, we first reformulate it as

$$\begin{aligned} &\min_{u,z} \; \frac{\| g - Hu \|_2^2}{2} + \lambda \| z \|_1 \\ &s.t. \quad z = Wu \end{aligned} \,, \tag{6}$$

and then use SBM to decouple the problem (6) into sub-problems

$$\begin{cases} u^{n+1} = \arg\min_{u} \| g - Hu \|_2^2 + \eta \| z^n - Wu - e^n \|_2^2 \\ z^{n+1} = \arg\min_{z} \lambda \| z \|_1 + \dfrac{\eta \| z - Wu^{n+1} - e^n \|_2^2}{2} \\ e^{n+1} = e^n + Wu^{n+1} - z^{n+1} \end{cases} \tag{7}$$

Since its objective function is quadratic, the sub-problem u^{n+1} in (7) can be directly solved to get

$$u^{n+1} = (H^{\mathrm{T}}H + \eta W^{\mathrm{T}}W)^{-1}[H^{\mathrm{T}}g + \eta W^{\mathrm{T}}(z^n - e^n)] . \tag{8}$$

To efficiently compute Eq. (8), we use the fast Fourier transform (FFT) to convert it into

$$u^{n+1} = F^{-1}\left(\frac{F(H)^* \otimes F(g) + \eta F(W)^* \otimes F(z^n - e^n)}{F(H)^* \otimes F(H) + \eta F(W)^* \otimes F(W)} \right), \tag{9}$$

where $F(\cdot)$ denotes the FFT operation, $F^{-1}(\cdot)$ denotes the inverse FFT operation; * denotes the complex conjugate; and $\otimes$ denotes the dot product. Since the time complexity of the FFT is $O(n\log n)$, any one of $F(H)^*\otimes F(g)$, $F(W)^*\otimes F(z^n-e^n)$, $F(H)^*\otimes F(H)$ and $F(W)^*\otimes F(W)$ can be implemented within the time complexity of $O(n\log n)$. After computing $H^{\mathrm{T}}g$, $W^{\mathrm{T}}(z^n-e^n)$, $H^{\mathrm{T}}H$ and $W^{\mathrm{T}}W$ by FFT, the division in Eq. (9) can also be completed within the time complexity of $O(n\log n)$. Therefore, using FFT to compute u^{n+1} is definitely more efficient than the direct computations in Eq. (8).

For the l_2-l_1 denoising sub-problem z^{n+1}, most image deconvolution algorithms solve it using ITA. However, prompted by recent studies on improving ITA, we use a modified version of SSDM to solve the subproblem z^{n+1} in order to obtain better restored results. Considering the linear system $Ax = b$, the unknown signal x is computed by

$$\begin{cases} x^1 = \gamma(Kx^0 + L) + (1-\gamma)x^0 \\ x^{i+1} = \rho[\gamma(Kx^i + L) + (1-\gamma)x^i] + (1-\rho)x^{i-1} \end{cases}, \tag{10}$$

where i denotes the i^{th} SSDM iteration; the constant $\rho>0$; $K=I-Q^{-1}A$, $L= Q^{-1}b$ and Q is a positive definite invertible matrix. Substituting $K=I-Q^{-1}A$ and $L= Q^{-1}b$ into Eq. (10), we get

$$\begin{cases} x^1 = x^0 + \gamma Q^{-1}(b - Ax^0) \\ x^{i+1} = (1-\rho)x^{i-1} + \rho x^i + \rho\gamma Q^{-1}(b - Ax^i) \end{cases}. \tag{11}$$

Let $b=S^{\mathrm{T}}f$, $Q=I+\sigma C^i$, $A=\sigma C^i+S^{\mathrm{T}}S$, and $x=z^{n+1}$, Eq. (11) can be expressed as

$$\begin{cases} z^{n+1,1} = (1-\gamma)z^{n+1,0} + \gamma Q^{-1}(z^{n+1,0} + S^{\mathrm{T}}(f - Sz^{n+1,0})) \\ z^{n+1,i+1} = (1-\rho)z^{n+1,i-1} + (\rho-\tau)z^{n+1,i} + \tau Q^{-1}(z^{n+1,i} + S^{\mathrm{T}}(f - Sz^{n+1,i})) \end{cases}, \tag{12}$$

where C^i is a diagonal matrix that depends on x^i and the regularizer; S and f denote an arbitrary matrix and vector, respectively; and the constant $\tau=\rho\gamma$. If we let $\rho=\gamma=1$, $S=I$ and $f=Wu^{n+1}+e^n$, and if multiplication by Q^{-1} is replaced by the denoising operator used in ITA, Eq. (12) is reduced to

$$z^{n+1,i+1} = \Theta_{\lambda W}(Wu^{n+1} + e^n) \tag{13}$$

which is equal to directly applying ITA to the sub-problem z^{n+1}. Therefore, the modified SSDM in Eq. (12) can be regarded as a variant of ITA. This fact motivates us to formulate the solution of the sub-problem z^{n+1} in Eq. (7) as

$$\begin{cases} z^{n+1,1} = (1-\gamma) z^{n+1,0} + \gamma \Omega_{\lambda/\eta}(Wu^{n+1} + e^n) \\ z^{n+1,i+1} = (1-\rho) z^{n+1,i-1} + (\rho - \tau) z^{n+1,i} + \tau \Omega_{\lambda/\eta}(Wu^{n+1} + e^n) \end{cases}, \tag{14}$$

where $\Omega_{\lambda/\eta}(Wu^{n+1}+e^n)$ is the soft-thresholding function, and it denotes the component-wise application of the function

$$(Wu^{n+1} + e^n)_j = \text{sign}((Wu^{n+1} + e^n)_j) \max(|(Wu^{n+1} + e^n)_j| - \lambda/\eta, 0) \tag{15}$$

with $(Wu^{n+1}+e^n)_j$ as the j^{th} (j=1, 2, … , N) component of $Wu^{n+1}+e^n$.

Using Eqs. (7), (9) and (14), we summarize the procedure for the presented image deconvolution algorithm below, where $\|\cdot\|_{\text{fro}}$ is the Frobenius norm. Since the convergences of SBM and SSDM have been proven, the convergence of the presented algorithm is guaranteed. Additionally, the set of solutions of the image deconvolution problem **P0** is not empty. Therefore, with proper parameter values and reasonable initialization, u^{n+1} will converge to a point in the solution set of **P0**.

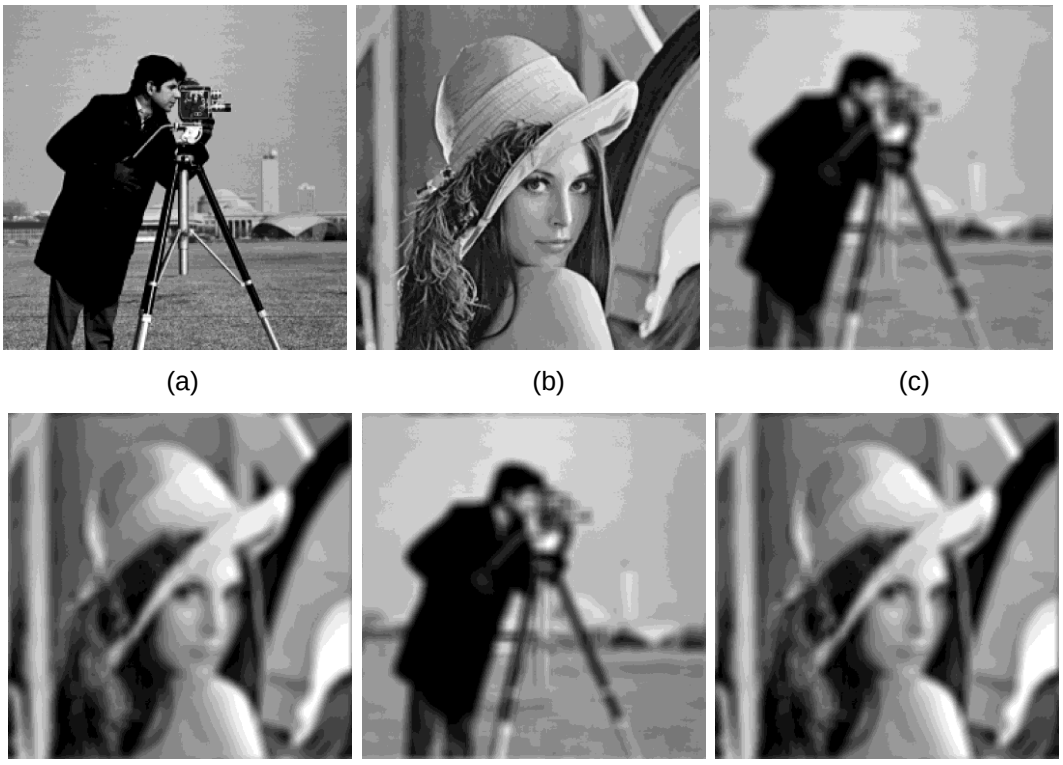
Presented Image Deconvolution Algorithm	**SSDM (stationary second-degree method)**

Input: λ and η; g, H and W; z^0 and e^0
Output: u^{n+1}
for n=0 to $n_{\max}$ **do**
compute u^{n+1} using Eq. (9);
if $\|u^{n+1}-u\|_{\text{fro}} > \|u^n-u\|_{\text{fro}}$ **then**
$u^{n+1}=u^n$;
stop iteration;
end if
compute $\tilde{z}^{n+1} = \Omega_{\lambda/\eta}(Wu^{n+1} + e^n)$;
compute z^{n+1} by SSDM;
$e^{n+1} = e^n + Wu^{n+1} - z^{n+1}$;
end for
	Input: γ, ρ and τ; $z^{n+1,0}$ and $\tilde{z}^{n+1}$
	Output: $z^{n+1,i+1}$ (i. e., z^{n+1})
	$z^{n+1,1} = (1-\gamma) z^{n+1,0} + \gamma \tilde{z}^{n+1}$;
	for i=1 to $i_{\max}$ **do**
	$z^{n+1,i+1} = (1-\rho) z^{n+1,i-1} + (\rho - \tau) z^{n+1,i} + \tau \tilde{z}^{n+1}$;
	end for

4. Experimental Results

In this section, a series of experiments are carried out to verify the effectiveness of the presented algorithm. We select the famous test gray images “Cameraman” and “Lena”, which are shown in Figure 1(a)-(b) as sharp images. To produce the blurry images shown in Figure 1 (c)-(f), the sharp images are first convolved by a Gaussian blur kernel and a uniform blur kernel, and then different levels of Gaussian noise are added to the blurry images. The Gaussian blur kernel and uniform blur kernel are generated using MATLAB functions *fspecial*(’gaussian’, [9 9], 9) and *fspecial*(’average’, [9 9]), respectively; the

BSNR (blurred signal-to-noise ratio) is defined as $10\times\log_{10}[\|u^{n+1}-u\|_2^2/N\delta^2]$ with δ as the standard deviation of the noise. To evaluate the performance of the presented algorithm, we also test the algorithms in references [6] and [8] to restore the same blurry images. The algorithm in [6] uses an improved ITA to address the sparse representation based image deconvolution problem, and the algorithm in [8] uses Bregman iteration plus FBS to address the total variation based image deconvolution problem. All algorithms are implemented on the following platform: Windows XP SP3, Intel Core 2 CPU T7200 @ 2.00GHZ, 4GB RAM and MATLAB R2012b. For the presented algorithm, the parameters are set as follows: η=0.00007, λ=0.07, ρ=1.95, τ=4, γ=2.05, and $n_{\max}$ and $i_{\max}$ are both set to be 20. For convenience and without loss of generality, w_j is set to 1 for all js. As for the algorithms in references [6] and [8], we employ their default settings.



(a) (b) (c)

(d) (e) (f)

Figure 1. Sharp Images and Blurry Images (a) Cameraman; (b) Lena; (c) Gaussian-Blurry Cameraman (GBC); (d) Gaussian-Blurry Lena (GBL); (e) Uniform-Blurry Cameraman (UBC); (f) Uniform-Blurry Lena (UBL). The Resolutions of All Images are 256×256, and the BSNR Values of all Blurry Images are 40 dB.

The experimental results are presented in Table 1 and Figures 2-3, where PSNR (peak signal-to-noise ratio) and MSE (mean square error) are defined as $10\times\log_{10}[N\times255^2/\|u^{n+1}-u\|_2^2]$ and $\|u^{n+1}-u\|_2^2/N$, respectively. From the PSNR values in Table 1 and the images in Figure 2, we conclude that the presented algorithm can effectively restore common types of blurry images. Compared with similar deconvolution algorithms in experiment, the presented algorithm illustrates better performance. Especially in term of visual effects, our algorithm preserves more details of restored images without the emergence of stair-step effects as shown in Figure 2(e)-(h). The underlying reasons for this fact are mainly that the presented algorithm has an excellent architecture and the modified SSDM outperforms the ITA used by the algorithms in references [6] and [8]. From the curves in Figure 3, we

observe that the MSE values decrease with each iteration, *i.e.*, u^{n+1} approaches sharp image u with each iteration, meaning that the presented algorithm is convergent. Figure 3 also shows that even without many iterations, the presented algorithm can obtain satisfactory solutions.

Table 1. PSNR Values of Restored Blurry Images in Figure 2

Algorithms	Restored images	PSNR values
algorithm in [6]	Figure 2(a)	26.01 dB
	Figure 2(b)	28.07 dB
	Figure 2(c)	26.02 dB
	Figure 2(d)	28.15 dB
algorithm in [8]	Figure 2(e)	25.66 dB
	Figure 2(f)	27.33 dB
	Figure 2(g)	25.82 dB
	Figure 2(h)	27.40 dB
presented algorithm	Figure 2(i)	27.01 dB
	Figure 2(j)	28.67 dB
	Figure 2(k)	26.95 dB
	Figure 2(l)	28.73 dB

5. Conclusions

We present a novel SBM and SSDM based algorithm which addresses the image deconvolution problem by solving the minimization problem **P0**. SBM decomposes **P0** into sub-problems and SSDM is used to solve the denoising sub-problem to improve the deconvolution results. In simulated experiment, the presented algorithm is used to restore two common types of blurry images and the results show its effectiveness. Compared with the algorithms in references [6] and [8], the presented algorithm obtains higher PSNR values and better visual effects. Finally, we also experimentally demonstrate the convergence of the presented algorithm with a variety of MSE values. Studies are conducted to extend our algorithm to other classic image deconvolution models, and the future work involves using a similar algorithm to solve the total variation model.

(a) (b) (c)

Figure 2. Restored Results of Blurry Images in Figure 1 (a) to (d) Results Obtained by the Algorithm in [6]; (e) to (h) Results Obtained by the Algorithm in [8]; (i) to (l) Results Obtained by the Presented Algorithm. For each Algorithm, the Images Respectively Denote the Restored GBC, GBL, UBC and UBL

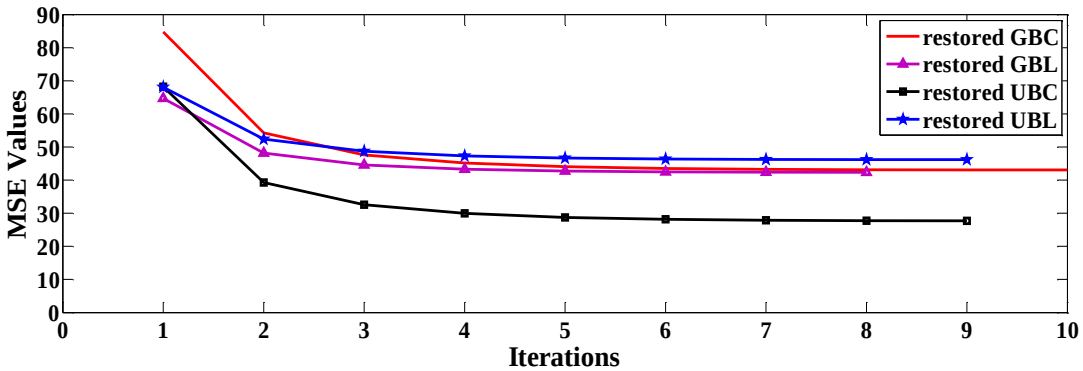


Figure 3. Variations of MSE Values of Images Restored by the Presented Algorithm

Acknowlegdment

This work is supported by Anhui Provincial Natural Science Foundation (Grant No. 1608085QF150).

References

[1] I. Daubechies, M. Defris and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", Communications on Pure and Applied Mathematics, vol. 57, no. 11, **(2004)**, pp. 1413-1457.
[2] D. L. Donoho, "De-noising by soft-thresholding", IEEE Transactions on Information Theory, vol. 41, no. 3, **(1995)**, pp. 613-627.
[3] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, An augmented Lagrangian method for total variation video restoration, IEEE Transactions on Image Processing, vol. 20, no. 11, **(2011)**, pp. 3097-3111.
[4] S. J.Wright, R. D. Nowak and M. A. Figueiredo, "Sparse reconstruction by separable approximation", IEEE Transactions on Signal Processing, vol. 57, no. 7, **(2009)**, pp. 2479-2493.
[5] E. T. Hale, W. Yin and Y. Zhang, "Fixed-point continuation for l1-minimization: methodology and convergence", SIAM Journal on Optimization, vol. 19, no. 3, **(2008)**, pp. 1107-1130.
[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", SIAM Journal on Imaging Sciences, vol. 2, no. 1, **(2009)**, pp. 183-202.
[7] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting", Multiscale Modeling & Simulation, vol. 4, no. 4, **(2005)**, pp. 1168-1200.
[8] X. Zhang, M. Burger, X. Bresson X, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction", SIAM Journal on Imaging Sciences, vol. 3, no. 3, **(2010)**, pp. 253-276.
[9] S. Durand, J. Fadili and M. Nikolova, "Multiplicative noise removal using L1 fidelity on frame coefficients", Journal of Mathematical Imaging and Vision, vol. 36, no. 3, **(2010)**, pp. 201-226.
[10] B. Dong, H. Ji, J. Li, Z. Shen and Y. Xu, "Wavelet frame based blind image inpainting", Applied and Computational Harmonic Analysis, vol. 32, no. 2, **(2012)**, pp. 268-279.
[11] D. R. Kincaid, "Stationary second-degree iterative methods", Applied Numerical Mathematics, vol. 16, no. 1, **(1994)**, pp. 227-237.
[12] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems", SIAM Journal on Imaging Sciences, vol. 2, no. 2, **(2009)**, pp. 323-343.
[13] S. Setzer, G. Steidl and T. Teuber, "Deblurring Poissonian images by split Bregman techniques", Journal of Visual Communication and Image Representation, vol. 21, no. 3, **(2010)**, pp. 193-199.
[14] T. Goldstein, X. Bresson and S. Osher, "Geometric applications of the split Bregman method: segmentation and surface reconstruction", Journal of Scientific Computing, vol. 45, no. 1-3, **(2010)**, pp. 272-293.
[15] Setzer S, "Operator splittings, Bregman methods and frame shrinkage in image processing", International Journal of Computer Vision, vol. 92, no. 3, **(2011)**, pp. 265-280.

