Advances in Adaptive Data Analysis
Vol. 3, No. 1 (2011) 1–15
© World Scientific Publishing Company
DOI: 10.1142/S1793536911000817

POSITIVELY CONSTRAINED TOTAL VARIATION PENALIZED IMAGE RESTORATION

RAYMOND H. CHAN[*,‡], HAI-XIA LIANG[*,§] and JUN MA[†,¶]

[*]*Department of Mathematics,*
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong, P. R. China

[†]*Department of Statistics,*
Macquarie University, Australia
[‡]*rchan@math.cuhk.edu.hk*
[§]*hxliang@math.cuhk.edu.hk*
[¶]*jun.ma@mq.edu.au*

The total variation (TV) minimization models are widely used in image processing, mainly due to their remarkable ability in preserving edges. There are many methods for solving the TV model. These methods, however, seldom consider the positivity constraint one should impose on image-processing problems. In this paper we develop and implement a new approach for TV image restoration. Our method is based on the multiplicative iterative algorithm originally developed for tomographic image reconstruction. The advantages of our algorithm are that it is very easy to derive and implement under different image noise models and it respects the positivity constraint. Our method can be applied to various noise models commonly used in image restoration, such as the Gaussian noise model, the Poisson noise model, and the impulsive noise model. In the numerical tests, we apply our algorithm to deblur images corrupted by Gaussian noise. The results show that our method give better restored images than the forward–backward splitting algorithm.

Keywords: Total variation; maximum penalized likelihood; multiplicative iterative algorithms; positivity constraint.

1. Introduction

The total variation (TV) image restoration is an important image-processing method due to its ability in preserving sharp edges in the restored image [Chambolle and Lions (1997); Vogel and Oman (1998); Strong and Chan (2003)]. The method is usually formulated as a penalized least squares with the TV penalty function [Vogel and Oman (1996, 1998)]. The existing numerical methods for solving this TV optimization problem include: partial differential equation (PDE) [Rudin *et al.* (1992)], primal-dual method using either Newton [Chan *et al.* (1996)] or conjugate gradient [Vogel and Oman (1998)] optimizations, etc. However, these methods do not consider the positivity constraint imposed on the restored image, and usually the pixel values are guaranteed to be nonnegative only in the last iteration by a simple projection or a scaling. In this paper, we propose a new algorithm for solving TV penalized, and positively constrained, image restoration problems. Our approach uses the multiplicative iterative (MI) algorithm originally developed for tomographic image reconstruction [Ma (2010)] under quadratic penalty terms. Here, we modify it to solve the TV penalized image restoration problem. Moreover, we extend the error probability model from the traditional Gaussian distribution to other probability distributions, such as Poisson and Laplace (equivalent to L_1 norm) distributions.

The following notations are used throughout this paper. Let $\mathcal{U} \in \mathbb{R}_+^{p\times q}$ be the unknown image, which requires to be estimated and $\mathcal{Y} \in \mathbb{R}_+^{p\times q}$ be the observed blurry image, where $\mathbb{R}_+$ denotes the positive orthant of $\mathbb{R}$. To simplify, images $\mathcal{U}$ and $\mathcal{Y}$ are lexicographically ordered into vectors, and we let $\mathbf{u} = (u_1, \ldots, u_n)^T$ denote the vectorized $\mathcal{U}$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$ denote the vectorized $\mathcal{Y}$, where superscript T represents matrix transpose and $n = p \cdot q$. We use these two notations of an image (i.e., two-dimensional (2D) $\mathcal{U}$ and its vectorized $\mathbf{u}$) interchangeably when there is no confusion; an operation on $\mathbf{u}$ can be defined using $\mathcal{U}$, and vice versa. For a function $g(\mathbf{u})$ we use $\nabla g(\mathbf{u})$ to denote derivative of g with respect to $\mathbf{u}$ and $\nabla_j g(\mathbf{u})$ the derivative of g with respect to u_j.

In this paper, we consider the statistical image restoration. Suppose the true (unobserved) image $\mathbf{u}$ is distorted by a blurring mechanism (such as a point spread function), which is denoted by an $n \times n$ matrix A. The expected observed image, denoted by an n-vector $\boldsymbol{\mu}$, is given by

$$\boldsymbol{\mu} = A\mathbf{u}.$$

However, due to noise contamination, we cannot observe $\boldsymbol{\mu}$ directly; instead, we observe the blurred noisy image $\mathbf{y}$. Our aim is to restore $\mathbf{u}$ from $\mathbf{y}$.

Statistical image restoration depends on the assumed probability model for the observed image and the penalty function (also known as the log prior density function). The penalty is used to restrict the restored image so that it follows certain local smoothness patterns. Assume that y_i are independent (given $\mathbf{u}$) and each follows a probability model, i.e.

$$y_i \sim p_i(y_i \,|\, \mu_i),$$

where $p_i(\cdot)$ represents the probability density function (PDF) of y_i and $\mu_i = A_i\mathbf{u}$ with A_i being the ith row of A. The penalized negative log-likelihood objective function $\Psi(\mathbf{u})$ for recovering $\mathbf{u}$ is given by

$$\Psi(\mathbf{u}) = -\sum_{i=1}^{n} l_i(\mu_i) + \lambda J(\mathbf{u}), \tag{1}$$

where $l_i(\mu_i) = \log p_i(y_i \,|\, \mu_i)$, $\lambda > 0$ is the smoothing parameter and $J(\mathbf{u})$ is the penalty function. The restored image $\hat{\mathbf{u}}$ is given as the minimizer of $\Psi(\mathbf{u})$ subject to $\mathbf{u} \geq 0$, namely

$$\hat{\mathbf{u}} = \operatorname*{argmin}_{\mathbf{u}\geq 0} \Psi(\mathbf{u}). \tag{2}$$

The first term of Eq. (1) represents the negative log-likelihood and it measures fidelity of the restored blurry image $\hat{\boldsymbol{\mu}} = A\hat{\mathbf{u}}$ to the observed image $\mathbf{y}$. Its second term, on the other hand, measures smoothness of the restored image $\hat{\mathbf{u}}$. The smoothing parameter λ is included for the purpose of controlling the amount of smoothness of the restored image. A good smoothing parameter should balance well these two conflicting targets, namely data fidelity and smoothness. This paper will not discuss how to select an optimal smoothing value for λ. Our focus is on how to compute efficiently the solution to problem (2) and how to obtain a good approximation to the target image.

The form of data-fitting term in Eq. (1) depends on the statistical noise model. Three kinds of common noise models are as follows:

(1) Gaussian noise model: observed image intensities $y_i \sim N(\mu, \sigma^2)$. In this model, after combining variance σ^2 with λ to form a new smoothing parameter, we have

$$l_i(\mu_i) = -\frac{1}{2}(y_i - \mu_i)^2. \tag{3}$$

(2) Poisson noise model: observed image intensities $y_i \sim \text{Poisson}(\mu_i)$. This noise model gives

$$l_i(\mu_i) = -\mu_i + y_i \log \mu_i. \tag{4}$$

(3) Impulsive noise model: observed image intensities $y_i \sim \text{Laplace}(\mu_i, \sigma^2)$. Similar to the Gaussian noise model example, after combining σ^2 with λ we can write

$$l_i(\mu_i) = -|y_i - \mu_i|. \tag{5}$$

In this paper, we consider the regularization function $J(\cdot)$ being the TV penalty function. Let $\|\cdot\|_2$ be the Euclidean norm of the relevant space and Ω be the domain of image $\mathcal{U}$. According to [Vogel and Oman (1998)], the TV penalty $J(\mathbf{u})$ can be written as

$$J(\mathbf{u}) = \int_{(\xi_1,\xi_2)\in\Omega} \|\nabla\mathcal{U}\|_2 d\xi_1 d\xi_2 \approx \sum_{j=1}^{n} \sqrt{(R_j\mathbf{u})^2 + (C_j\mathbf{u})^2 + \beta}, \tag{6}$$

where R_j and C_j are respectively the jth row of the $n \times n$ matrices R and C. The entries of $R\mathbf{u}$ and $C\mathbf{u}$ represent the first-order differences of $\mathcal{U}$ along the row and column directions, respectively. If $\mathbf{u}_j$ in $\mathbf{u}$ corresponds to $\mathcal{U}_{s,t}$ in $\mathcal{U}$, then

$$R_j\mathbf{u} = \begin{cases} \mathcal{U}_{s+1,t} - \mathcal{U}_{s,t} & \text{for } 1 \le s \le m-1 \\ \mathcal{U}_{1,t} - \mathcal{U}_{m,t} & \text{for } s = m \end{cases}$$

$$C_j\mathbf{u} = \begin{cases} \mathcal{U}_{s,t+1} - \mathcal{U}_{s,t} & \text{for } 1 \le t \le m-1 \\ \mathcal{U}_{s,1} - \mathcal{U}_{s,m} & \text{for } t = m \end{cases}$$

In Eq. (6), parameter $\beta > 0$ is included to avoid degenerate derivative of $J(\mathbf{u})$.

We introduce a multiplicative iterative (MI) algorithm to solve the constrained optimization problem (2) where the penalty function $J(\mathbf{u})$ is TV. We call this new method the MITV algorithm. The advantages of MITV are that it is very easy to derive and implement under different image noise models, such as Gaussian, Poisson and impulsive, and it respects the positivity constraint. In the numerical tests, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method gives better restored images than the forward–backward splitting (FBS) algorithm.

The rest of this paper is arranged as follows. Section 2 reviews the FBS algorithm for TV image restoration. It is the algorithm that we use to compare with MITV. Section 3 develops our new MITV image restoration algorithm. Two test images with Gaussian noises are used to compare MITV with FBS, and the results are given in Sec. 4. Finally, concluding remarks are provided in Sec. 5.

2. The FBS Algorithm

In this section, we explain the FBS algorithm (see e.g. [Combettes and Wajs (2005); Bredies (2009)]) for solving problem (2) for the Gaussian noise model. This algorithm was designed without considering the positivity constraint. To obtain a positive solution, however, one usually projects the iteration results into the nonnegative half space at every iteration or at the last iteration.

Let $H(\mathbf{u}) = -\sum_{i=1}^{n} l_i(\mu_i) = \frac{1}{2}\|A\mathbf{u} - \mathbf{y}\|_2^2$. We begin with the definition of the proximity operator. For any proper, convex and semicontinuous function $\phi(\cdot)$ with range $(-\infty, +\infty]$, its proximity operator is defined by

$$\operatorname{prox}_{\delta\phi} : \mathbf{u} \to \operatorname*{argmin}_{\mathbf{v}} \left\{ \frac{1}{2\delta}\|\mathbf{u} - \mathbf{v}\|_2^2 + \phi(\mathbf{v}) \right\}. \tag{7}$$

The FBS algorithm can be used to solve the minimization problem of the following form

$$\min_{\mathbf{u}} \{F_1(\mathbf{u}) + F_2(\mathbf{u})\}, \tag{8}$$

where F_1 is a proper, convex, lower semicontinuous function and F_2 is a convex, $1/\gamma$-Lipschitz continuous differentiable function. The FBS iteration formula for solving Eq. (8) is given as follows

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\delta F_1}(\mathbf{u}^k - \delta\nabla F_2(\mathbf{u}^k)). \tag{9}$$

In [Combettes and Wajs (2005)], the authors show that this FBS algorithm converges to the solution of Eq. (8) when $0 < \delta < 2\gamma$.

We can apply the FBS algorithm to solve problem (2) with the Gaussian noise model, where the penalty term $F_1(\mathbf{u}) = \lambda J(\mathbf{u})$ and the data fidelity term $F_2(\mathbf{u}) = H(\mathbf{u})$, respectively. In this case, the Lipschitz constant $\gamma = 1/\|A^T A\|_2$. The gradient of the data fidelity term is: $\nabla H(\mathbf{u}) = A^T(A\mathbf{u} - \mathbf{y})$. The FBS iteration formula (9) is then given by

$$\mathbf{u}^{k+1} = \mathrm{prox}_{\delta(\lambda J)}(\mathbf{u}^k - \delta \nabla H(\mathbf{u}^k)) \tag{10}$$

from any initial $\mathbf{u}^0$. By definition of the proximity operator (7), the iteration formula (10) is equivalent to

$$\mathbf{u}^{k+1} = \operatorname*{argmin}_{\mathbf{u}} \left\{ \lambda J(\mathbf{u}) + \frac{1}{2\delta} \|\mathbf{u} - (\mathbf{u}^k - \delta \nabla H(\mathbf{u}^k))\|_2^2 \right\}. \tag{11}$$

The FBS algorithm (11) converges to the solution of (2) for Gaussian noise removal if $\delta \in (0, \frac{2}{\|A^T A\|_2})$. Recall that $J(\mathbf{u})$ is the TV penalty function, so we can apply the Chambolle's denoising algorithm [Chambolle (2004)] to obtain the minimizer of Eq. (11) at each step. Interested readers can consult [Combettes and Wajs (2005)] for more details of the FBS algorithm [Chambolle (2004)] for general idea of the Chambolle's denoising algorithm, and [Hiriart-Urruty and Lemaréchal (1996); Moreau (1962, 1995)] for more background knowledge of convex analysis.

3. MITV Penalized Image Restoration

3.1. *Derivation of the algorithm*

In image processing, pixel values should be nonnegative numbers. In [Hager *et al.* (2009); Morini *et al.* (2010)], the authors discussed the box-constrained minimization problem based on the Karush–Kuhn–Tucker (KKT) condition. The interested readers can consult [Hager *et al.* (2009)] for the affine-scaling interior-point cyclic [Barzilai and Borwein (1988)] method for box-constrained minimization problem and [Morini *et al.* (2010)] for an a reduced Newton method for box-constrained linear least-squares problems. In [Ma (2010)], the author proposed a MI algorithm for problems in tomographic reconstructions. There, the penalty function is taken to be a quadratic one. In this section, we develop the MI algorithm for TV penalized and positively constrained image restoration. The algorithm is flexible and can be applied to different image noise models. It also begins with the KKT necessary conditions.

We first introduce some notations needed for the derivation of the MITV algorithm below. For any vector function $g(\mathbf{u})$, it can be separated into positive part vector $[g(\mathbf{u})]^+$ and negative part vector $[g(\mathbf{u})]^-$ with their jth entry defined by

$$[g(\mathbf{u})]_j^+ = \max\{[g(\mathbf{u})]_j, 0\}, \quad [g(\mathbf{u})]_j^- = \max\{-[g(\mathbf{u})]_j, 0\}, \tag{12}$$

where $[g(\mathbf{u})]_j$ is the jth component of $g(\mathbf{u})$. By Eq. (12)

$$g(\mathbf{u}) = [g(\mathbf{u})]^+ - [g(\mathbf{u})]^-. \tag{13}$$

Obviously $[g(\mathbf{u})]^+ \geq \mathbf{0}$ and $[g(\mathbf{u})]^- \geq \mathbf{0}$. We call this separation **Type 1** separation that we use in the following part. If $g(\mathbf{u})$ has an explicit separation form, i.e.

$g(\mathbf{u}) = \mathbf{a} - \mathbf{b}$ with $\mathbf{a} \geq \mathbf{0}$ and $\mathbf{b} \geq \mathbf{0}$, we just take $[g(\mathbf{u})]^+ = \mathbf{a}$ and $[g(\mathbf{u})]^- = \mathbf{b}$. We call this separation **Type 2** separation in the following part.

If $\hat{\mathbf{u}}$ solves optimization problem (2), then $\hat{\mathbf{u}}$ satisfies the KKT conditions:

$$\nabla_j \Psi(\hat{\mathbf{u}}) = 0, \quad \text{if } \hat{u}_j > 0$$

$$\nabla_j \Psi(\hat{\mathbf{u}}) \geq 0, \quad \text{if } \hat{u}_j = 0$$

for $j = 1, \ldots, n$. Equivalently, $\hat{\mathbf{u}}$ solves the following linear system

$$D(\mathbf{u}) \nabla \Psi(\mathbf{u}) = \mathbf{0}, \tag{14}$$

where $D(\mathbf{u}) = \operatorname{diag}(d_1(\mathbf{u}), \ldots, d_n(\mathbf{u}))$ with

$$d_j(\mathbf{u}) = \begin{cases} u_j, & \nabla_j \Psi(\mathbf{u}) = 0 \\ 0. & \nabla_j \Psi(\mathbf{u}) > 0 \end{cases}$$

Using expression (1), Eq. (14) is a linear system of

$$u_j \left(-\sum_{i=1}^{n} a_{ij} \nabla l_i(\mu_i) + \lambda \nabla_j J(\mathbf{u}) \right) = 0, \quad j = 1, \ldots, n \tag{15}$$

where $a_{ij} \geq 0$ represents the (i,j)th element of matrix A. First, we apply the separation (13) to $\nabla l_i(\mu_i)$ and $\nabla_i J(\mathbf{u})$, to get $\nabla l_i(\mu_i) = [\nabla l_i(\mu_i)]^+ - [\nabla l_i(\mu_i)]^-$ and $\nabla_j J(\mathbf{u}) = [\nabla_j J(\mathbf{u})]^+ - [\nabla_j J(\mathbf{u})]^-$. Then, we rewrite Eq. (15) as

$$u_j \left(\sum_{i=1}^{n} a_{ij} [\nabla l_i(\mu_i)]^- + \lambda [\nabla_j J(\mathbf{u})]^+ \right) = u_j \left(\sum_{i=1}^{n} a_{ij} [\nabla l_i(\mu_i)]^+ + \lambda [\nabla_j J(\mathbf{u})]^- \right), \tag{16}$$

where both sides of this equation are now nonnegative.

Equation (16) naturally suggests an iterative scheme for solving Eq. (15), and that is

$$u_j^{k+1/2} = u_j^k \frac{\delta_{1j}^k}{\delta_{2j}^k}, \qquad j = 1, \ldots, n \tag{17}$$

where

$$\delta_{1j}^k = \sum_{i=1}^{n} a_{ij} [\nabla l_i(\mu_i^k)]^+ + \lambda [\nabla_j J(\mathbf{u}^k)]^-$$

$$\delta_{2j}^k = \sum_{i=1}^{n} a_{ij} [\nabla l_i(\mu_i^k)]^- + \lambda [\nabla_j J(\mathbf{u}^k)]^+,$$

and

$$\delta_{2j}^k - \delta_{1j}^k = \nabla_j \Psi(\mathbf{u}). \tag{18}$$

Here, $\mu_i^k = A_i\mathbf{u}^k$ and $\nabla_j J(\mathbf{u}^k)$ denotes $\nabla_j J(\mathbf{u})$ evaluated at $\mathbf{u}^k$. In Eq. (17) the iteration index $k + 1/2$ explains that this is merely a temporary updating, and further improvements are necessary to give $\mathbf{u}^{k+1}$ by the following rule

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k \mathbf{d}^k.$$

Here, $\alpha^k \in (0, 1]$ is a positive step size and the direction vector $\mathbf{d}^k$ is given componentwise by

$$d_j^k = u_j^{k+1/2} - u_j^k.$$

Substituting Eqs. (17) and (18) into the above equality, we have

$$d_j^k = -\frac{u_j^k}{\delta_{2j}} \nabla_j \Psi(\mathbf{u}). \tag{19}$$

Hence, $\mathbf{d}^k$ is a down-hill direction for $\Psi(\mathbf{u})$ when the denominator δ_{2j}^k is nonzero for all j.

In order to facilitate Eq. (17) we must provide $\nabla_j J(\mathbf{u})$ for the TV penalty function $J(\mathbf{u})$ given in Eq. (6). It is not difficult to derive that

$$\nabla_j J(\mathbf{u}) = \sum_{t=1}^{n} \frac{r_{tj} R_t\mathbf{u} + c_{tj} C_t\mathbf{u}}{\sqrt{(R_t\mathbf{u})^2 + (C_t\mathbf{u})^2 + \beta}}. \tag{20}$$

To ensure that the iteration scheme (17) is well defined, we have to explain how to handle the possibility of $\delta_{2j}^k = 0$. This problem can be rectified simply by replacing any zero δ_{2j}^k with a constant ε (such as $\varepsilon = 10^{-2}$). In this case, however, the corresponding numerator of Eq. (17) must also be altered by $\delta_{1j}^k + \varepsilon$ so that the estimating Eq. (15) is still maintained.

It is possible that $\mathbf{u}^{k+1/2}$ by Eq. (17) does not decrease $\Psi(\mathbf{u})$, i.e., $\Psi(\mathbf{u}^{k+1/2}) \geq \Psi(\mathbf{u}^k)$. In this case we need a line search step to improve $\mathbf{u}^{k+1/2}$ such that the final update decreases the objective function $\Psi(\mathbf{u})$. In this line search, we must find a $0 < \alpha^k \leq 1$ such that

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k), \tag{21}$$

and $\mathbf{u}^{k+1}$ satisfies $\Psi(\mathbf{u}^{k+1}) < \Psi(\mathbf{u}^k)$. Here, α^k can be obtained efficiently by approaches such as step half or Armijo rule [Luenberger (1984)]. From Eqs. (17) and (19), we know that Eq. (21) maintain all the zero components of $\mathbf{u}^k$ in $\mathbf{u}^{k+1}$.

We call the iterative scheme defined by Eqs. (21) and (17), where the penalty function $J(\mathbf{u})$ and its derivative $\nabla J(\mathbf{u})$ are given by, respectively, Eqs. (6) and (20), the MITV algorithm. This algorithm is very easy to be implemented in TV image restoration tasks, and it can handle any image noise model. A very attractive feature is that this algorithm respects the positivity constraint usually imposed on image restoration problems. In fact, from any initial guess $\mathbf{u}^0 \geq 0$, we have $\mathbf{u}^{1/2} > 0$ by

Eq. (17). We then immediately obtain that $\mathbf{u}^1 \geq 0$ since $\mathbf{u}^1 = (1-\alpha^0)\mathbf{u}^0 + \alpha^0\mathbf{u}^1$ and $0 < \alpha^0 \leq 1$. By induction, we know that $\mathbf{u}^k$ is always nonnegative if initial guess $\mathbf{u}^0 \geq 0$. From the study reported in Sec. 4, the MITV algorithm is very competitive with the existing FBS TV denoising algorithm.

Under certain regularity conditions, the MITV algorithm is convergent, and moreover, it converges to the solution satisfying the KKT necessary conditions. The proof of the general convergence result is available in [Ma (2010)].

3.2. *MITV under different image noise models*

In this section, we provide the details of the MITV algorithms when the observed image contains Gaussian, Poisson, or impulse noises. From Eq. (17), these MITV algorithms differ only due to the fact that the derivatives of the log density functions, i.e., $\nabla l_i(\mu_i)$, are different for different noise models.

3.2.1. *Gaussian noise model*

For the Gaussian noise model, according to the data-fitting term (3), then the gradient $\nabla l_i(\mu_i) = f_i - \mu_i$, where $f_i \geq 0$ and $\mu_i > 0$. Use **Type 2** separation to $\nabla_i(\mu_i)$, we have $[\nabla l_i(\mu_i)]^+ = f_i$ and $[\nabla l_i(\mu_i)]^- = \mu_i$. In addition, we use **Type 1** separation to $\nabla J(\mathbf{u})$. Then iteration (17) becomes

$$u_j^{(k+1/2)} = u_j^{(k)} \frac{\sum_{i=1}^n b_{ij} f_i + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^-}{\sum_{i=1}^n b_{ij}\mu_i + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^+}, \tag{22}$$

for $j = 1, \ldots, n$. If the denominator of Eq. (22) is zero for a pixel j, then we add a threshold ε to both numerator and denominator of Eq. (22). Once $\mathbf{u}^{k+1/2}$ is obtained, we move to $\mathbf{u}^{k+1}$ as follows. If $\Psi(\mathbf{u}^{k+1/2}) < \Psi(\mathbf{u}^k)$, then $\mathbf{u}^{k+1} = \mathbf{u}^{k+1/2}$; otherwise, using a line search to find a $0 < \alpha^k < 1$ such that

$$\Psi(\mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k)) < \Psi(\mathbf{u}^k),$$

and then set $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k)$.

3.2.2. *Poisson noise*

For the Poisson noise model, from Eq. (4), the gradient $\nabla l_i(\mu_i) = -1 + f_i/\mu_i$. Any zero μ_i will make both $l_i(\mu_i)$ and $\nabla l_i(\mu_i)$ not well defined. If $\mu_i^k = 0$ happens in iterations we will replace μ_i^k in denominator by a small constant $\theta > 0$. Hence, we can assume all $\mu_i \neq 0$ without loss of generality. Using **Type 2** separation to $\nabla l_i(\mu_i)$, we have $[\nabla l_i(\mu_i)]^+ = f_i/\mu_i$ and $[\nabla l_i(\mu_i)]^- = 1$. In addition, we use **Type 1** separation to $\nabla J(\mathbf{u})$. Then iteration formula (17) now becomes

$$u_j^{(k+1/2)} = u_j^{(k)} \frac{\sum_{i=1}^n b_{ij} f_i/\mu_i^k + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^-}{\sum_{i=1}^n b_{ij} + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^+}, \tag{23}$$

for $j = 1, \ldots, n$. As in the Gaussian noise above, if the denominator of Eq. (23) is zero for a pixel j, then we add a threshold ε to both numerator and denominator of Eq. (23). Then, calculate $\mathbf{u}^{k+1}$ from $\mathbf{u}^{k+1/2}$ by line search.

3.2.3. *Impulsive noise*

We use the Laplace distribution to model impulsive noises so that $l_i(\mu_i) = -|f_i - \mu_i|$ according to Eq. (5). A problem with this model is that $l_i(\mu_i)$ is not differentiable at $\mu_i = f_i$. However, adopting the idea of Huber function [Huber (1973)], we may define the derivative of $l_i(\mu_i)$ to be

$$\begin{aligned}\nabla l_i(\mu_i) &= \begin{cases} 1 & \text{if } \mu_i < f_i - \gamma \\ -1 & \text{if } \mu_i > f_i + \gamma \\ \dfrac{f_i - \mu_i}{\gamma} & \text{if } f_i - \gamma \le \mu_i \le f_i + \gamma \end{cases} \\ &= I_{\frac{f_i-\mu_i}{\gamma}>1} + \frac{f_i - \mu_i}{\gamma} I_{0 \le \frac{f_i-\mu_i}{\gamma} \le 1} - I_{\frac{f_i-\mu_i}{\gamma}<-1} + \frac{f_i - \mu_i}{\gamma} I_{-1 \le \frac{f_i-\mu_i}{\gamma} < 0},\end{aligned}$$

We apply **Type 1** separation to $\nabla l_i(\mu_i)$, then we have $[\nabla l_i(\mu_i)]^+ = I_{\frac{f_i-\mu_i}{\gamma}>1} + \frac{f_i-\mu_i}{\gamma} I_{0 \le \frac{f_i-\mu_i}{\gamma} \le 1}$ and $[\nabla l_i(\mu_i)]^- = I_{\frac{f_i-\mu_i}{\gamma}<-1} + \frac{\mu_i - f_i}{\gamma} I_{-1 \le \frac{f_i-\mu_i}{\gamma} < 0}$, where $I_{\mathcal{A}}$ is an indicator function for event $\mathcal{A}$: $I_{\mathcal{A}} = 1$ if $\mathcal{A}$ is true, and $I_{\mathcal{A}} = 0$ otherwise. The separation for $\nabla J(\mathbf{u})$ is still the **Type 1** separation. The updating formula (17) now becomes

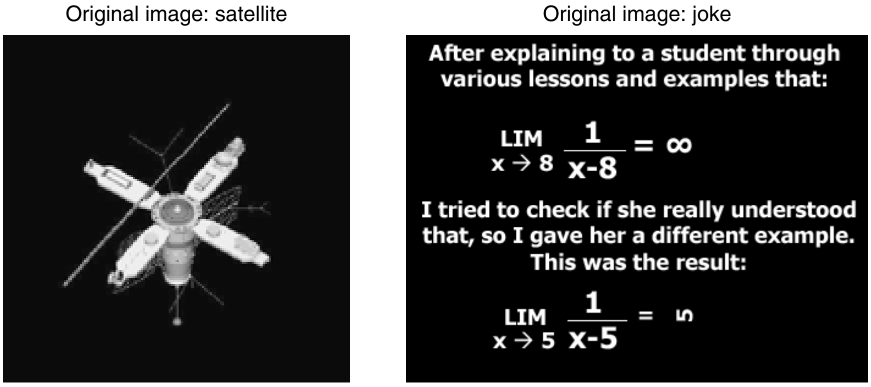
$$u_j^{k+1/2} = u_j^k \frac{\sum_{i=1}^n b_{ij} \left(I_{\frac{f_i-\mu_i^k}{\gamma}>1} + \frac{f_i-\mu_i^k}{\gamma} I_{0 \le \frac{f_i-\mu_i^k}{\gamma} \le 1} \right) + \lambda [\nabla_j J(\mathbf{u}^k)]^-}{\sum_{i=1}^n b_{ij} \left(I_{\frac{f_i-\mu_i^k}{\gamma}<-1} + \frac{\mu_i^k - f_i}{\gamma} I_{-1 \le \frac{f_i-\mu_i^k}{\gamma} < 0} \right) + \lambda [\nabla_j J(\mathbf{u}^k)]^+}, \tag{24}$$

for $j = 1, \ldots .n$. It is possible that at an iteration, the denominator of Eq. (24) is zero. In this case, we add a quantity $\varepsilon > 0$ to both numerator and denominator of Eq. (24). The update $\mathbf{u}^{k+1}$ is obtained from $\mathbf{u}^{k+1/2}$ by line search; see the Gaussian noise model example above.

4. Numerical Results

In this section, we apply our MITV algorithm and the FBS algorithm to deblur images which are corrupted by Gaussian noises. For the Poisson and impulsive noise models, the augmented Lagrangian method (ALM) in [Wu *et al.* (2009)] is a fast and efficient method. The interested readers can consult [Wu *et al.* (2009)] for more details. Comparison of MITV with ALM on non-Gaussian noise models will be reported elsewhere.

In our tests, we use two images as shown in Fig. 1. The satellite image has a size of 176 × 176, and the joke image has a size of 284 × 378. We choose these images because they both contain large zero backgrounds, so we expect that the positivity



Original image: satellite

Original image: joke

After explaining to a student through various lessons and examples that:

$$\lim_{x \to 8} \frac{1}{x-8} = \infty$$

I tried to check if she really understood that, so I gave her a different example. This was the result:

$$\lim_{x \to 5} \frac{1}{x-5} = \text{ທ}$$

Fig. 1. Original test images.

constraint is strongly informative in both examples. In the satellite image, there are 76.72% pixels equal to 0. In the joke image, this percentage is 86.50%. Due to such features, it is easier to test the effect of our algorithm in ensuring the positivity constraint.

In all the tests, matrix A is the blurring matrix corresponding to motion blur and is generated by the MATLAB command

```
fspecial('motion', 15, 30).
```

Gaussian noise is added to the blurry image to obtain the observed one. Denote the noise level by σ, which represents the standard deviation of the Gaussian noise. We test two cases: one has noise level $\sigma = 5$, and the other one has noise level $\sigma = 10$.

Considering the good property of MITV in preserving positivity and the feature of our test images that most of the pixels equal to zero, we propose a projection in each MITV iteration:

$$u_j^{k+1} = \begin{cases} 0, & u_j^{k+1} < \eta \\ u_j^{k+1}, & \text{otherwise} \end{cases} \tag{25}$$

for $j = 1, \ldots, n$. We call this projection the lower projection, where η is called the lower projection parameter. This projection will help to further improve the restoration from our MITV algorithm. We call MITV with lower projection the PMITV algorithm. The optimal choice for η in all the tests is obtained experimentally. By trial and error, we find that the value of η almost increases in direct proportion to σ. Morini *et al.* [2010], also discuss the box-constrained image restoration problems with projections. The interested readers can consult [Morini *et al.* (2010)] for more details.

By trial and error, we find that for the satellite image the best smoothing parameter $\lambda = 0.4$ at noise level $\sigma = 5$, and the best smoothing parameter $\lambda = 1$ at noise level $\sigma = 10$. For the joke image, the best smoothing parameter $\lambda = 0.2$ at noise level $\sigma = 5$, and the best smoothing parameter $\lambda = 0.4$ at noise level $\sigma = 10$.

We use the peak signal-to-noise ratio (PSNR) to measure the quality of the restored images, which is defined by

$$\mathrm{PSNR} := 10\log_{10}\frac{255^2}{\frac{1}{n}\|\hat{\mathbf{u}}-\mathbf{u}\|_2^2}(\mathrm{dB}).$$

Here, $\mathbf{u}$ and $\hat{\mathbf{u}}$ denote the original and restored images respectively, and n is the total pixel number of the image.

We show the comparison results of the PMITV algorithm and the FBS algorithm in Figs. 2–5. Figures 2 and 3 show the observed and the recovered images by the PMITV and FBS algorithms. In PMITV, at noise level $\sigma = 5$, we choose $\eta = 4$; at noise level $\sigma = 10$, we choose $\eta = 9$. At each of these ηs, the corresponding PMITV algorithm reaches the highest PSNR value among all the trials that we conducted. The PFBS algorithm is the FBS algorithm with a lower projection, see Eq. (25) at every iteration. Also from the trials, we find that zero is the best choice for η in PFBS.

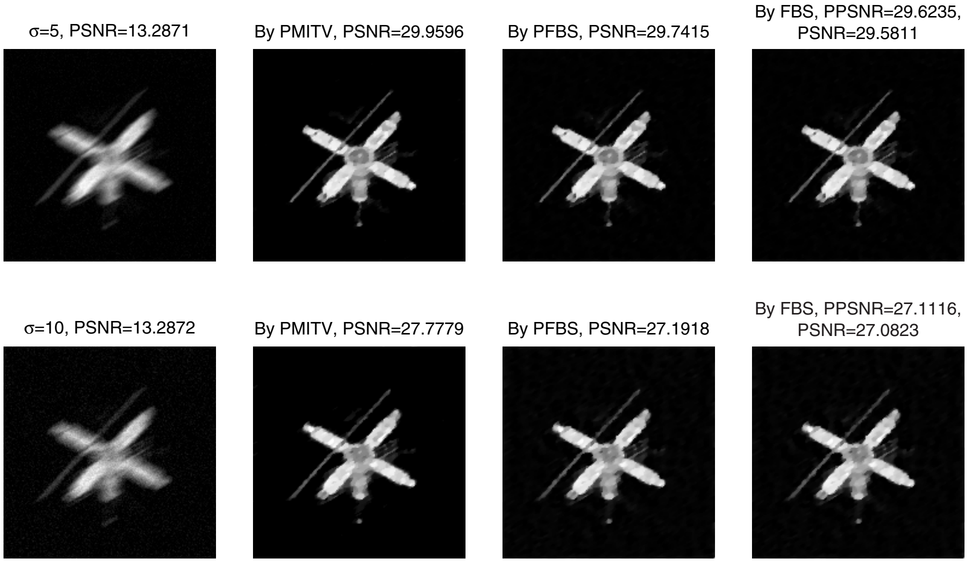


Fig. 2. The figure shows the results of MITV algorithm and FBS algorithm when restoring blurred noisy image “satellite.” From left to right, the first column shows the blurred noisy images (row 1: noise level $\sigma = 5$, $\mathrm{PSNR} = 13.2871$; row 2: noise level $\sigma = 10$, $\mathrm{PSNR} = 13.2872$). The right three columns show the restored results by the MITV algorithm and the FBS algorithm for restoring the corresponding degraded image in the first column on the same row. PMITV is the MITV (row 1: $\lambda = 0.4$; row 2: $\lambda = 1$) algorithm with a lower projection, see Eq. (25) (row 1: $\eta = 4$; row 2: $\eta = 9$) at every iteration; PFBS is the FBS algorithm with a lower projection, see Eq. (25) (row 1 and row 2: $\eta = 0$) at every iteration. In the last column, PPSNR denotes the PSNR value of the recovered image by the FBS algorithm only with a lower projection, see Eq. (25) ($\eta = 0$) at the final step; PSNR denotes the PSNR value without any projection. All the results shown in the figure are obtained with an additional upper projection, see Eq. (26) at the final step.

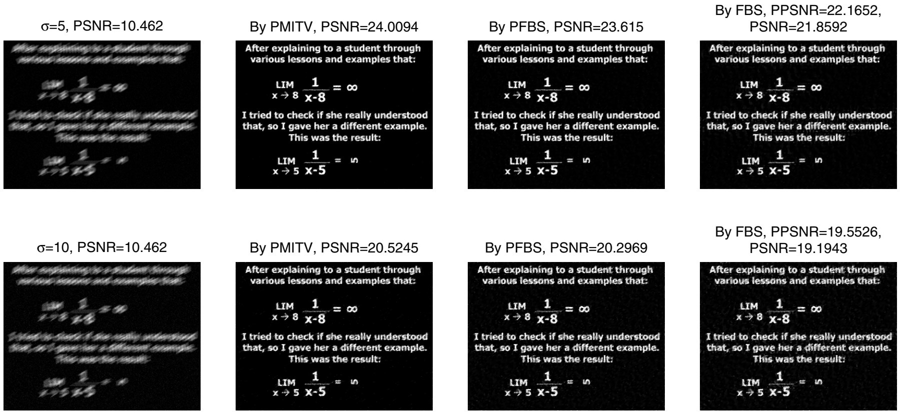

σ=5, PSNR=10.462
By PMITV, PSNR=24.0094
By PFBS, PSNR=23.615
By FBS, PPSNR=22.1652, PSNR=21.8592
σ=10, PSNR=10.462
By PMITV, PSNR=20.5245
By PFBS, PSNR=20.2969
By FBS, PPSNR=19.5526, PSNR=19.1943
After explaining to a student through various lessons and examples that:
LIM x→8 1/x-8 = ∞
I tried to check if she really understood that, so I gave her a different example. This was the result:
LIM x→5 1/x-5 = ∽

Fig. 3. The figure shows the results of MITV algorithm and FBS algorithm when restoring blurred noisy image "joke." From left to right, the first column shows the blurred noisy images (row 1: noise level $\sigma = 5$, PSNR $= 10.462$; row 2: noise level $\sigma = 10$, PSNR $= 10.462$). The right three columns show the restored results by the MITV algorithm and the FBS algorithm for restoring the corresponding degraded image in the first column on the same row. PMITV is the MITV (row 1: $\lambda = 0.2$; row 2: $\lambda = 0.4$) algorithm with a lower projection, see Eq. (25) (row 1: $\eta = 4$; row 2: $\eta = 9$) at every iteration; PFBS is the FBS algorithm with a lower projection, see Eq. (25) (row 1 and row 2: $\eta = 0$) at every iteration. In the last column, PPSNR denotes the PSNR value of the recovered image by the FBS algorithm only with a lower projection, see Eq. (25) ($\eta = 0$) at the final step; PSNR denotes the PSNR value without any projection. All the results shown in the figure are obtained with an additional upper projection, see Eq. (26) at the final step.

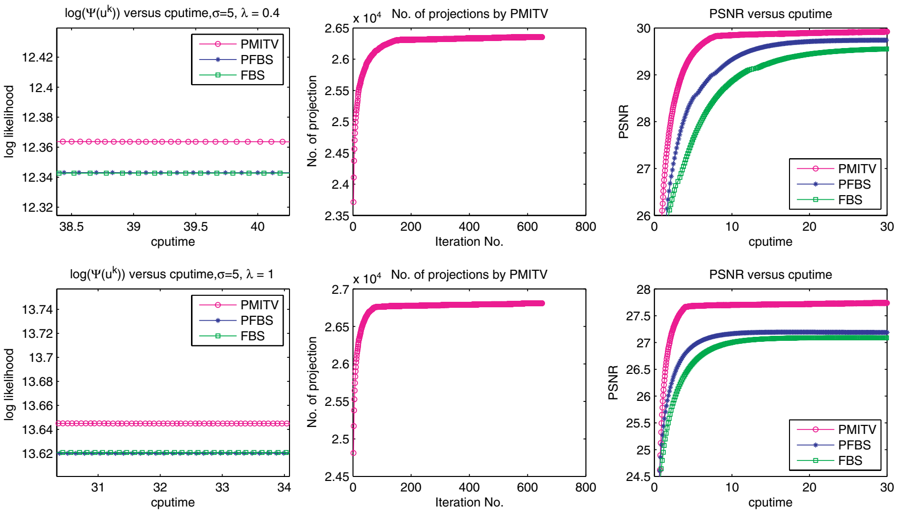


Fig. 4. The figure shows the accuracy and the time comparison of the MITV algorithm and FBS algorithm for satellite image. PMITV algorithm is the MITV algorithm with a lower projection. The first row is for noise level $\sigma = 5$, in PMITV, we take $\lambda = 0.4$, $\eta = 4$; the second row is for noise level $\sigma = 10$, in PMITV, we take $\lambda = 1, \eta = 9$. From left to right, the first column describes the decreasing of the energy value in time; the second column describes the number of projections at each iteration in PMITV; the third column shows the timing comparison of the PSNR values. From the second column, we see that the number of projections tends to a constant as time increasing. The third column shows us clearly that the MITV algorithm can reach a higher PSNR value faster than PFBS with projection and FBS without projection.

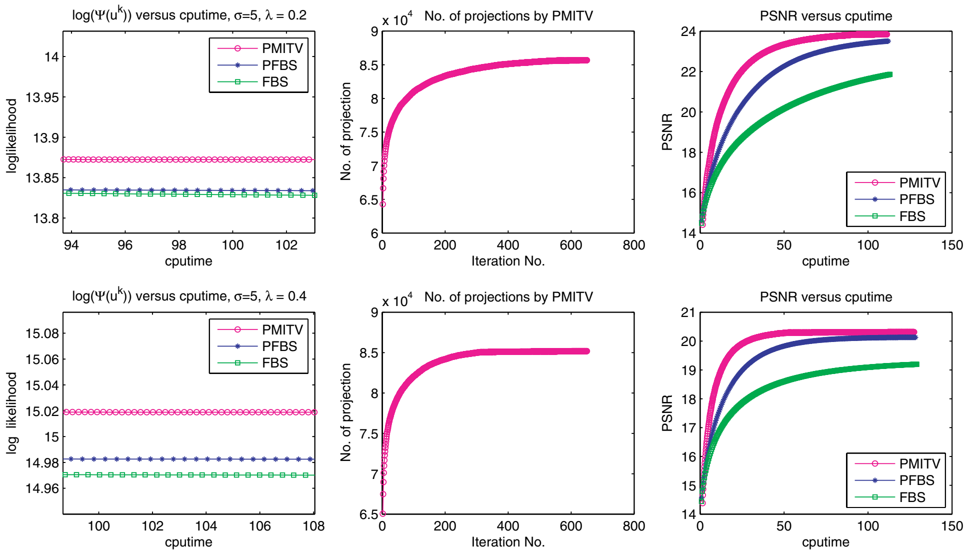


Fig. 5. The figure shows the accuracy and the time comparison of the MITV algorithm and FBS algorithm for joke image. PMITV algorithm is the MITV algorithm with a lower projection. The first row is for noise level $\sigma = 5$, in PMITV, we take $\lambda = 0.2, \eta = 4$; the second row is for noise level $\sigma = 10$, in PMITV, we take $\lambda = 0.4, \eta = 9$. From left to right, the first column describes the decreasing of the energy value in time; the second column describes the number of projections at each iteration in PMITV; the third column shows the timing comparison of the PSNR values. From the second column, we see that the number of projections tends to a constant as time increasing. The third column shows us clearly that the MITV algorithm can reach a higher PSNR value faster than PFBS with projection and FBS without projection.

In the subfigures for the FBS algorithm in Figs. 2 and 3, the PSNR records the PSNR value of the final iteration of the FBS algorithm without any projection, while the PPSNR records the PSNR value of FBS algorithm with a lower projection, see Eq. (25) only at the end of all the iterations, where the lower projection parameter is zero. At the final step in all the test algorithms, we take an upper projection

$$u_j^{\text{rec}} = \begin{cases} 255, & u_j^{\text{rec}} > 255 \\ u_j^{\text{rec}}, & \text{otherwise} \end{cases} \tag{26}$$

such that the recovered image is in $[0, 255]$. From Figs. 2 and 3, we see that the PMITV always reach a higher PSNR value than PFBS (i.e., with projection) and FBS (without projection).

Figures 4 and 5 give computational time comparisons of the MITV algorithm and the FBS algorithm. From left to right, the first columns of these figures describe the time comparison of the energy value sequence $\{\Psi(\mathbf{u}^k)\}$. The second columns give the number of the pixels that are projected to zero at each iteration in the PMITV algorithm. We observe that the number of pixels that are projected tends to stabilize as the iteration number increases. The third columns show the time comparison in relation to the PSNR value. We find that the PMITV always reaches higher PSNR value faster than the PFBS and FBS algorithms.

5. Conclusions

In this paper, we develop and implement a new approach for total variation image restoration. Our method is based on the multiplicative iterative algorithm originally developed for tomographic image reconstruction. The advantages of our algorithm are that it is very easy to derive and implement under different noise models and it respects the positivity constraint. We discuss in the paper how to apply this method to Gaussian, Poisson, and impulsive noise models. In the numerical test, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method gives better restored images than the FBS algorithms.

Acknowledgment

The research was supported in part by HKRGC Grant CUHK 400508 and CUHK DAG 2060257.

References

Barzilai, J. and Borwein, J. (1988). Two point step size gradient methods. *IMA J. Numer. Anal.*, **8**: 141–148.

Bredies, K. (2009). A forwardbackward splitting algorithm for the minimization of non-smooth convex functionals in Banach space. *Inverse Problems*, **25**: 1–20.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *J. Math. Imag. Ves.*, **20**: 89–97.

Chambolle, A. and Lions, P. L. (1997). Image recovery via total variation minimization and related problems, *Numer. Math.*, **72**: 167–188.

Chan, T., Golub, G. H. and Mulet, P. (1996). A non-linear primal-dual method for TV-based image restoration, *Lecture Notes in Control and Information Science No. 219*, (Eds.) M. Berger *et al.*, pp. 241–252.

Combettes, P. L. and Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, **4**: 1168–1200.

Hager, W. Mair, B. and Zhang, H. (2009). An affine-scaling interior-point CBB method for box-constrained optimization. *Math. Program.*, **119**: 1–32.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (1996). *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, I.

Huber, P. J. (1973). Robust regression: asymptotics, conjectures, and Monte Carlo. *Ann. Statist.*, **1**: 799–821.

Luenberger, D. (1984). *Linear and Nonlinear Programming* (2nd Edition), J. Wiley.

Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Trans. Nuc. Sci.*, **57**: 181–192.

Moreau, J. J. (1962). Function convexes duales et points proximaux dans un espace Hibertien. *C. R. Acad. Sci. Paris Sér. A Math.*, **255**: 1897–2899.

Moreau, J. J. (1995). Proximité et dualité dans un espace hibertien. *Bull. Soc. Math. France*, **93**: 273–299.

Morini, B. Porcelli, M. and Chan, R. H. (2010). A reduced Newton method for constrained linear least-squares problems. *J. Comp. Appl. Math.*, **233**: 2200–2212.

Rudin, L. Osher, S. and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, **60**: 259–268.

Strong, D. and Chan, T. (2003). Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, **19**: 165–187.

Vogel, C. R. and Oman, M. E. (1998). Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Process.*, **7**: 813–824.

Vogel, C. R. and Oman, M. E. (1996). Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, **17**: 227–238.

Wu, C. Zhang, J. and Tai, X. (2009). Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. *UCLA CAM Report 09-82*.