Partial Convolution for Total Variation Deblurring and Denoising by New Linearized Alternating Direction Method of Multipliers with Extension Step

Yuan Shen[†] [1] **Lei Ji**[†]

School of Applied Mathematics, Nanjing University of Finance & Economics, Nanjing, 210023, P. R. China.[†]

Abstract: In this paper, we propose a partial convolution model for image delburring and denoising. We also devise a new linearized alternating direction method of multipliers (ADMM) with extension step. On one hand, the computation of its subproblem is dominated by several FFTs, hence its per-iteration cost is low, on the other hand, the relaxed parameter condition together with the extra extension step inspired by Ye and Yuan's ADMM enables faster convergence than the original linearized ADMM. Preliminary experimental results show that our algorithm can produce a result with better quality than some existing efficient algorithms while the computation time is competitive. Specially, the advantage of our algorithm can be more evident when the noise ratio is high.

Keywords: convex optimization, proximal point algorithm, augmented Lagrangian, total variation, deblur and denoise, partial convolution.

AMS classification(2010): 90C30, 65K05, 94A08

1 Introduction

In this paper, we consider recovering image from its observation degraded by blurring and impulsive noise. Let $\bar{x} \in \mathbb{R}^{mn}$ be the original $m \times n$ image by stacking its the columns, $K \in \mathbb{R}^{mn \times mn}$ represents a blurring (or convolution) matrix, $\omega \in \mathbb{R}^{mn}$ be additive noise, and $f \in \mathbb{R}^{mn}$ be an observation which satisfies the relationship

$$f = K\bar{x} + \omega. \tag{1.1}$$

For given K, the underlying problem is to recover $\bar{x}$ from the blurry and noisy observation f.

Due to the large condition number of K, recovering $\bar{\mathbf{x}}$ from f by directly inverting (1.1) is unstable and produces useless results because the solution is highly sensitive to the inevitably additive noise ω and the round-off error. Worse still, the dimension of (1.1) can be notably large, *e.g.*, even a small-scale image like 256×256 image can lead to a $K \in \mathbb{R}^{65536 \times 65536}$ which is difficult to handle on a normal PC.

To stabilize the recovery of $\bar{x}$, regularization strategy has then been introduced. Traditional regularization techniques include the Tikhonov regularization [46], Mumford-Shah regularization [37], and the total variation (TV) regularization [45, 44]. All of them have been well studied in the literatures.In which, the ROF TVL2 model with TV regularizer was introduced by Rudin, Osher and Fatemi [45].

$$\textbf{ROF TVL2 model} \quad \min_{x} \sum_{i=1}^{m \times n} \|D_i x\|_2 + \mu \|Kx - f\|_2^2, \tag{1.2}$$

where the summation term is sum of discrete difference at all pixels.

The main advantage of TV regularization is that it can preserve the sharp boundaries better than Tikhonov regularizer [1], so it yields high-quality result for natural image, especially for those with piecewise smooth objects but without complicated and minor textures *e.g.*, see [52]. However, for the non-Gaussian type of noise

[1] Email: ocsiban@126.com. Research supported by National Natural Science Foundation of China under grant 11401295 and by Jiangsu Provincial Natural Science Foundation under grant BK20141007 and by Major Program of the National Social Science Foundation of China under Grant 12&ZD114 and by National Social Science Foundation of China under Grant 15BGL58 and by Jiangsu Provincial Social Science Foundation under Grant 14EUA001 and by Qinglan Project of Jiangsu Province.

like impulsive noise where the values on the corrupted entries are significantly changed, the result obtained by the ROF model is of very poor quality.

In order to handle the impulsive noise, the nonsmooth fidelity term $\|Kx - f\|_1$ was then introduced [2]. By replacing $\|Kx - f\|_2^2$ with $\|Kx - f\|_1$, we obtain the ROF TVL1 model [2, 39]:

$$\textbf{ROF TVL1 model} \quad \min_x \sum_{i=1}^{m\times n} \|D_i x\|_2 + \mu\|Kx - f\|_1, \tag{1.3}$$

which has attracted attention due to a number of advantages [13], including its superior performance with non-Gaussian noise such as impulsive noise. Beyond these, TVL1 model also has some other interesting applications, *e.g.*, see [13, 18, 54]. Throughout this paper, we concentrate on the TVL1 model, but the proposed algorithms are also designed for, but not limited to, TVL1 model.

Although the TVL1 model is quite effective for image restoration, and a bunch of efficient algorithms have been proposed for solving it, e.g., see [49, 35], however, it still has limitations. For instance, the TVL1 model can not handle the case when too many entries are corrupted since the L^1 fidelity term can only tolerate a small portion of outliers. Therefore, as the noise ratio increases, the result obtained by algorithms based on TVL1 model will become worse. When over 60% pixels are corrupted, the TVL1 model can no longer produce a satisfactory result [49]. What makes thing worse is that, when the noise is "biased", *i.e.*, the entries corrupted by *light* noise and *dark* noise are not equal, the result of TVL1 model would also be "biased" even when the ratio of noisy entries is not that high.

In this paper, we put emphasize on improving the TVL1 model as well as the algorithms. To this end, we propose a partial convolution TVL1 model which only make use of uncorrupted entries. Its major advantage lies in that it can tolerate higher noise ratio than some existing algorithms. For solving this model, we propose a new linearized alternating direction method of multipliers. Preliminary experimental results shows that it can obtain a result with much better quality than that obtained by algorithms based on standard TVL1 model, and the performance gap can be more evident when the noise ratio is high.

The remainder of the present paper is organized as follows. In Section 2, we briefly review the existing algorithms for TVL1 model. In Section 3, we derive our proposed method, establish its convergence results, and illustrate its implementation for TVL1 problem. In Section 4, we present extensive numerical results on the performance of our proposed algorithm in comparison to some state-of-the-art algorithms. Finally, we conclude the paper in Section 5.

2 Brief review on existing algorithms

Earlier methods for TV-based deblurring models (1.2) includes lagged diffusivity (LD) algorithm [47], primal-dual (PD) algorithm [14], iterative shrinkage/thresholding (IST) [19]. On one hand, these earlier methods are slow in terms of computing time; on the other hand, they are only designed for solving TVL2 model which can only deal with i.i.d. Gaussian noise.

In recent years, there have been abundance works on exploring efficient and versatile algorithms for image restoration, e.g., fast total variation de-convolution algorithm (FTVD) [49], TV minimization by augmented Lagrangian and alternating direction algorithms (TVAL3) [35]. The FTVD consists of solvers for both TVL2 and TVL1 models. For solving the TVL1 model, it uses the celebrated alternating direction method of multipliers (ADMM) [24, 23] to solve an equivalent form of TVL1 model as follows:

$$\begin{aligned} \textbf{ROF TVL1 model} \quad & \min_{x,z,w_1,\ldots,w_{m\times n}} \sum_{i=1}^{m\times n} \|w_i\| + \mu\|z\|_1 \\ & \text{s.t.} \quad D_i x = w_i, \quad i = 1, 2, ..., m \times n, \\ & \qquad Kx - z = f. \end{aligned} \tag{2.4}$$

Let us briefly review on the ADMM, and start from a generic constrained optimization with two blocks of

variables in the form of:

$$\begin{aligned} \min_{x,y} \quad & f(x)+g(y) \\ \text{s.t.} \quad & Ax+By=b, \\ & x\in\mathcal{X},\ y\in\mathcal{Y}, \end{aligned} \tag{2.5}$$

where $\mathcal{X}\subset\Re^{n_1\times 1}$ and $\mathcal{Y}\subset\Re^{n_2\times 1}$ are given closed and convex sets, $f(x):\Re^{n_1\times 1}\to\Re$ and $g(y):\Re^{n_2\times 1}\to\Re$ are closed proper convex functions, and $A\in\Re^{m\times n_1}$, $x\in\Re^{n_1\times 1}$, $B\in\Re^{m\times n_2}$, $y\in\Re^{n_2\times 1}$, $b\in\Re^{m\times 1}$. We always assume that the solution set of (2.5) is nonempty.

Let the augmented Lagrangian function of (2.5) be

$$L_s(x,y,\lambda)=f(x)+g(y)-\lambda^T(Ax+By-b)+\frac{1}{2s}\|Ax+By-b\|^2, \tag{2.6}$$

where $s>0$ be a penalty parameter; and λ be the Lagrange multiplier associated with constraints $Ax+By=b$. The augmented Lagrangian method (ALM) in [32] and [42] can be applied to solve (2.6). With given λ^k, the iterative scheme of ALM for (2.5) is as follows:

$$\begin{cases} (x^{k+1},y^{k+1})=\arg\min\{L_s(x,y,\lambda^k)\mid x\in\mathcal{X},\ y\in\mathcal{Y}\}, \\ \lambda^{k+1}=\lambda^k-\gamma\frac{1}{s}(Ax^{k+1}+By^{k+1}-b). \end{cases} \tag{2.7}$$

It has been well studied in the literature (see e.g. [43]) that the ALM is an application of the proximal point algorithm (PPA) to the dual problem of (2.5). Hence, the convergence of ALM can be derived in terms of the dual variable, see e.g. [6].

The scheme (2.7) is the direct application of the ALM to (2.5), and it requires to minimize the variables x and y simultaneously at each iteration. This is not efficient when the functions $f(x)$ and $g(y)$ have particular properties such as separability. When the minimization problem in (2.7) is decomposed into two separable ones and they are minimized consecutively, the resulting alternating direction method of multipliers (ADMM) [24, 26] iterates as follows:

$$\begin{cases} x^{k+1}=\arg\min\{L_s(x,y^k,\lambda^k)\mid x\in\mathcal{X}\}, \\ y^{k+1}=\arg\min\{L_s(x^{k+1},y,\lambda^k)\mid y\in\mathcal{Y}\}, \\ \lambda^{k+1}=\lambda^k-\gamma\frac{1}{s}(Ax^{k+1}+By^{k+1}-b). \end{cases} \tag{2.8}$$

The ADMM can be seen as a splitting version or a Gauss-Seidel version of the ALM. It was proposed by Gabay [23], Gabay and Mercier [24], Glowinski and Marrocco [26], and its convergence analysis can be found in [25], or in [30] for a simpler proof. It has been extensively studied in the theoretical frameworks of both Lagrangian functions [22] and maximal monotone operators [20]. It has also been shown [36] that the method is an instance of the Douglas-Rachford splitting [20] for finding a zero point of a maximal monotone operator. Since there have been abundant works about its theoretical analysis, improvements, and applications, e.g., see [21, 31, 29] and the cited works therein. Due to the simple algorithm framework and satisfactory numerical performance, the ADMM has been successfully applied to solve problems arising from various practical applications, e.g., see [8] for a complete review. Compared to the original problem (2.7), the subproblems in (2.8) are of lower dimensions and thus often easier to solve. Furthermore, by taking the advantage of special structure of $f(x)$ and $g(y)$, the subproblems in (2.8) are often easy enough to have closed-form solutions. This fact makes the application of ADMM particularly efficient for a wide range of problems. In FTVD, the ADMM is applied to solve (2.4), the subproblem involving variables $\{w,z\}$ (referred as $\{w,z\}$-subproblem) can be computed by a soft shrinkage whose cost is negligible. The x-subproblem is an unconstrained quadratic programming which can be solved by various existing algorithms. Furthermore, by exploiting the circular structure of the constraint matrices, the FTVD applies FFT to enable faster computation. The satisfactory performance of ADMM together with the cheap per-iteration cost (dominated by several FFTs) make the FTVD stunningly faster than those earlier methods. It is reported to be orders of magnitude faster than those earlier methods like the LD method [47], or Newton based method such as [14]. It can recover a normal-size image on a normal PC within tolerable time. The FTVD is also easy to implement due to its simple iterating scheme. Furthermore, the FTVD can be used to solve both TVL1 and TVL2 models, as well as multichannel (color) model, hence, it is also a versatile algorithm.

The FTVD is quite efficient for solving TVL1 model, however, it still has some limitations. For instance, it can not handle the high noise ratio case since the L^1 fidelity term can only tolerate a small portion of outliers. Therefore, as the noise ratio increases, the produced result will become worse. It is reported in [49] that when over 60% pixels are corrupted, the TVL1 model can no longer produce a satisfactory result. What makes thing worse is that when the noise is "biased", *i.e.*, the numbers of "light" and "dark" corrupted entries are not roughly equal, the result of TVL1 model would also be "biased" even when the noise ratio is not high.

Unlike Gaussian noise, the entries corrupted by impulsive noise does not contain any useful information, hence, a simple and straightforward remedy is to detect those corrupted pixels, and only make use of the rest part of pixels which are presumed to be uncorrupted. Based on this idea, a two-stage approach was proposed in [9, 11]. Its algorithm framework consists of two stages. In the first stage, it gets an indices set matrix Ω by some heuristic method such as median-type filter [11], where Ω contains the indices of entries which are presumed to be uncorrupted. In the second stage of [9], it uses the partial observation of corrupted image, and considers a model involving Mumford-Shah regularization function [37] which can produce a result as good as that of TV-regularizer based model. This model is especially attractive to render restoration with fine structures in image, such as neat edges, textures. However, the non-convexity of the objective makes it difficult to handle, thus an approximation model was considered therein. The approximation model can be solved by variational approaches. However, the computational effort on doing that is time-consuming, *e.g.*, see [47]. This algorithm is slow, but its excellent quality is still attractive due to the removal of corrupted pixels.

Inspired by the idea of the two-stage approach, we propose a similar two-stage approach. We follow the first stage in [9]. In the second stage, we use a partial TVL1 model as follows:

$$\textbf{Partial TVL1 model} \quad \begin{aligned} \min_{x,z,w_1,\dots w_{m\times n}} & \sum_{i=1}^{m\times n} \|w_i\| + \mu\|z\|_1 \\ \text{s.t.} \quad & D_i x - w_i = 0, \quad i = 1,2,...,m\times n, \\ & P_\Omega(Kx - z) = P_\Omega(f), \end{aligned} \tag{2.9}$$

where $Kx - f$ is replaced by $P_\Omega(Kx - f)$.

Specially, when the indices matrix Ω is free of corrupted entries, the model (2.9) reduces to the following partial noiseless model:

$$\textbf{Partial noiseless model} \quad \begin{aligned} \min_{x,w_1,\dots w_{m\times n}} & \sum_{i=1}^{m\times n} \|w_i\| \\ \text{s.t.} \quad & D_i x - w_i = 0, \quad i = 1,2,...,m\times n, \\ & P_\Omega(Kx) = P_\Omega(f). \end{aligned} \tag{2.10}$$

However, the matrix $P_\Omega(K)$ does not possess circular structure, hence, the FFT is not applicable for solving the x-subproblem in the ADMM scheme. As a consequence, the FTVD is no longer suitable for handling partial models such as (2.9) or (2.10).

For solving the partial TVL1 model, Huang et al. [34] first proposed a fast method by considering an approximation model whose solution approaches that of (2.9) asymptotically. The numerical results in [34] illustrated that their algorithm is much more efficient than the algorithm proposed in [9]. Furthermore, [50] improved its efficiency further by introducing some more splitting variables, and formulating it into a constrained problem with three blocks of variables, and the resulting subproblems are tractably tackled with closed-form solutions. In general, the convergence of ADMM for more than two blocks of variables does not hold (see [16] for counter examples), although researchers have established its convergence with more than two blocks of variables under additional assumptions [27, 33, 17]. Some researchers use correction step such as Gaussian back substitution technique [28] to derive the convergence without additional assumptions, however, this technique can slow down the convergence of ADMM.

Alternatively, a simple and straightforward remedy is allowing the inexact computation for solving the x-subproblem of ADMM. For instance, the TV minimization by augmented Lagrangian and alternating direction algorithm (TVAL3) [35] uses the gradient descent method with Barzilai-Borwein (BB) [3] step size to solve the x-subproblem inexactly. It is reported to more preferable from the aspect of application although its

theoretical result is limited due to the utilization of inexact minimization. In addition, the application range of TVAL3 is restricted since it can only handle the partial noiseless model (2.10). Another class of algorithm for solving TVL1 model is the primal-dual algorithm [10] which solves a reformulated TVL1 model. Other competitive algorithms include NESTA [5] which uses the Nesterov's accelerating technique [38], TwIST [7] which implements a two-step IST method, and FISTA [4] which has better convergence rate than the classical IST. However, their application range can be restricted too. For instance, the NESTA requires the constraint matrix to be row-orthonormal which does not hold in (2.9) and (2.10), the TwIST can only handle noiseless model (2.10) like TVAL3.

As suggested in [51, 48], for solving a generic model (2.5), we can linearize the quadratic term in one subproblem of ADMM (e.g., y-subproblem in (2.5)), and add a proximal term $\frac{r}{2}\|y-y^k\|^2$, which suggests the following procedures:

$$\begin{cases} x^{k+1} = \arg\min\{(f(x) - {\lambda^k}^T Ax + \frac{1}{2s}\|Ax+By^k-b\|^2 \mid x\in\mathcal{X}\},\\ y^{k+1} = \arg\min\{(g(y) + \frac{1}{s}B^T[(Ax^{k+1}+By^k-b) - s\lambda^k])^T(y-y^k) + \frac{r}{2}\|y-y^k\|^2 \mid y\in\mathcal{Y}\},\\ \lambda^{k+1} = \lambda^k - \frac{1}{s}(Ax^{k+1}+By^{k+1}-b), \end{cases} \tag{2.11}$$

where the parameter $r>0$ controls the proximity to y^k.

The above method was entitled linearized ADMM (LADMM) [48], PADM [51], Proximal ADMM [15], or linearized Bregman iterative method (in a more general framework) [40]. It has been studied from the aspects of both theory and application. In [51], it was proved that the iterating sequence generated by (2.11) is convergent providing the condition $\|B^TB\| < rs$ holds, and it was reported to perform well for solving compressed sensing (CS) problem [51]. When $g(y)$ has some favourable property such as separability, the computation of y-subproblem can be done by a closed form solution. This can greatly reduce the cost of solving y-subproblem, hence although the LADMM converges slower than the ADMM, it is still preferable due to its low per-iteration cost, thus suitable for solving partial TVL1 model (2.9).

To overcome the issue of slow convergence, we propose a new LADMM with faster convergence, and present it in the next section.

3 Our method

As elaborated in literatures, (e.g., [51]), the LADMM is convergent when $\|B^TB\| < rs$ holds. However, this condition is conservative, and this limits the performance of LADMM. In fact, there have been abundant works on improving the performance of the LADMM, e.g., see [12, 41]. Inspired by Ye-Yuan's modified ADMM (YYADMM) [53], we take the solution of (2.11) as a predicting point instead of a new iterate, and then do an extension on variables $u \triangleq \begin{pmatrix} y \\ \lambda \end{pmatrix}$.

The iterative scheme of our new LADMM for solving the generic optimization model (2.5) consists of the following procedures:

$$\begin{cases} x^{k+1} = \arg\min\{(f(x) - {\lambda^k}^T Ax + \frac{1}{2s}\|Ax+By^k-b\|^2 \mid x\in\mathcal{X}\},\\ \tilde{y}^k = \arg\min\{(g(y) + \frac{1}{s}B^T[(Ax^{k+1}+By^k-b) - s\lambda^k])^T(y-y^k) + \frac{r}{2}\|y-y^k\|^2 \mid y\in\mathcal{Y}\},\\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^{k+1}+B\tilde{y}^k-b),\\ u^{k+1} = u^k - \gamma\alpha_k^*(u^k-\tilde{u}^k), \end{cases} \tag{3.12}$$

where $\gamma\in(0,2)$, and α_k^* is an optimal step size.

By introducing some notations:

$$\mathrm{dy} \triangleq r(y^k-\tilde{y}^k),\quad \mathrm{d}\lambda \triangleq s(\lambda^k-\tilde{\lambda}^k),\quad \mathrm{du} \triangleq \begin{pmatrix} \mathrm{dy} \\ \mathrm{d}\lambda \end{pmatrix},\quad G \triangleq \begin{pmatrix} rI & 0 \\ 0 & sI \end{pmatrix},\quad G-\mathrm{norm}:\ \|u\|_G = \sqrt{u^TGu}, \tag{3.13}$$

the procedures described in (3.12) can be simplified to be

$$u^{k+1} = u^k - \gamma\alpha_k^* G^{-1}\mathrm{du},$$

where α_k^* is explicitly given by

$$\alpha^* = \frac{\|G^{-1}\mathrm{du}\|_G^2 + \mathrm{d}\lambda^T B\mathrm{dy}}{\|G^{-1}\mathrm{du}\|_G^2}.$$

In addition, we let $u^* = \begin{pmatrix} y^* \\ \lambda^* \end{pmatrix}$ denotes an arbitrary solution of (2.5), and Ω^* denotes the solution set of (2.5).

To summarize, the new algorithm is described as follows:

Algorithm 1: New Linearized ADMM

Input $A \in \Re^{m\times n1}$, $B \in \Re^{m\times n2}$, $b \in \Re^m$, $r > 0$ and $s > 0$.
Initialize y^0, λ^0 and set $k = 0$.
while *not "converged"* **do**
  $x^{k+1} = \arg\min\{(f(x) - \lambda^{k^T} Ax + \frac{1}{2s}\|Ax + By^k - b\|^2 \mid x \in \mathcal{X}\}$,
  $\tilde{y}^k = \arg\min\{(g(y) + \frac{1}{s}B^T[(Ax^{k+1} + By^k - b) - s\lambda^k])^T(y - y^k) + \frac{r}{2}\|y - y^k\|^2 \mid y \in \mathcal{Y}\}$,
  $\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^{k+1} + B\tilde{y}^k - b)$,
  $u^{k+1} = u^k - \gamma\alpha_k^*(u^k - \tilde{u}^k)$ with $\alpha^* = \frac{\|G^{-1}\mathrm{du}\|_G^2 + \mathrm{d}\lambda^T B\mathrm{dy}}{\|G^{-1}\mathrm{du}\|_G^2}$,
  Increment k and continue.

The advantage of our new approach is two-fold: we can derive weaker parameter condition than $\|B^TB\| < rs$ which allows more aggressive choices of the parameters; the convergence of LADMM is accelerated by adopting the extension step on variable u.

To derive the contractive property of the iterative sequence generated by our new method, the following lemma and theorem are useful.

Lemma 3.1 *Let $\tilde{u}^k$ be generated by (3.12), when $\|B^TB\| < 4rs$ holds, we have*

$$(\tilde{\lambda}^k - \lambda^k)^T B(\tilde{y}^k - y^k) > (c-1)(s\|\lambda^k - \tilde{\lambda}^k\|^2 + r\|y^k - \tilde{y}^k\|^2), \quad \textit{for some} \quad c > 0. \tag{3.14}$$

Proof: When $\|B^TB\| < 4rs$ holds, there exists a constant $c > 0$ such that

$$\|B^TB\| < 4(1-c)^2 rs,$$

then we have

$$\|B(\tilde{y}^k - y^k)\|^2 \le (2-2c)^2 rs\|\tilde{y}^k - y^k\|^2. \tag{3.15}$$

Using the the following inequality

$$\frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 + a^Tb \ge 0, \tag{3.16}$$

we have (setting $a := \sqrt{(2-2c)s}(\tilde{\lambda}^k - \lambda^k)$ and $b := \frac{1}{\sqrt{(2-2c)s}}B(\tilde{y}^k - y^k)$)

$$\begin{aligned}(\tilde{\lambda}^k - \lambda^k)^T(B(\tilde{y}^k - y^k)) &\ge (c-1)s\|\lambda^k - \tilde{\lambda}^k\|^2 + \frac{1}{2(2c-2)s}\|B(y^k - \tilde{y}^k)\|^2, \\ &\ge (c-1)(s\|\lambda^k - \tilde{\lambda}^k\|^2 + r\|y^k - \tilde{y}^k\|^2),\end{aligned} \tag{3.17}$$

where the second inequality follows from (3.15), thus the assertion is proved. □

Theorem 3.1 *When (3.14) holds, there exists a step size $\alpha_k > 0$ such that, when we take $u^{k+1} = u^k - \alpha_k G^{-1}\mathrm{du}$, it holds that*

$$\|u^k - u^*\|_G^2 - \|u^{k+1} - u^*\|_G^2 \ge \kappa\|G^{-1}\mathrm{du}\|_G^2 \qquad \textit{for some } \kappa > 0. \tag{3.18}$$

Proof: Invoking the optimality condition of x^{k+1} and $\tilde{y}^k$, we obtain

$$\begin{aligned}(x^* - x^{k+1})^T(\nabla f(x^{k+1}) - A^T\tilde{\lambda}^k + \frac{1}{s}A^TB(y^k - \tilde{y}^k)) \ge 0, \\ (y^* - \tilde{y}^k)^T(\nabla g(\tilde{y}^k) - B^T\tilde{\lambda}^k + (rI - \frac{1}{s}B^TB)(\tilde{y}^k - y^k)) \ge 0.\end{aligned} \tag{3.19}$$

On the other hand, the solution of (2.5) satisfies

$$\begin{aligned}(x^{k+1}-x^*)^T(\nabla f(x^*)-A^T\lambda^*)\geq 0,\\ (\tilde{y}^k-y^*)^T(\nabla g(y^*)-B^T\lambda^*)\geq 0.\end{aligned} \tag{3.20}$$

Adding (3.19), (3.20) and using the convexity of f and g, we get

$$\begin{aligned}(x^{k+1}-x^*)^T(A^T(\tilde{\lambda}^k-\lambda^*)+\frac{1}{s}A^TB(\tilde{y}^k-y^k))\geq 0,\\ (\tilde{y}^k-y^*)^T(B^T(\tilde{\lambda}^k-\lambda^*)-(rI-\frac{1}{s}B^TB)(\tilde{y}^k-y^k))\geq 0.\end{aligned} \tag{3.21}$$

We do a sum for the two inequalities in (3.21), and obtain

$$(\tilde{\lambda}^k-\lambda^*)^T(Ax^{k+1}+B\tilde{y}^k-b)+(x^{k+1}-x^*)^TA^T(\frac{1}{s}B(\tilde{y}^k-y^k))-(\tilde{y}^k-y^*)^T(rI-\frac{1}{s}B^TB)(\tilde{y}^k-y^k)\geq 0. \tag{3.22}$$

Invoking

$$Ax^{k+1}+B\tilde{y}^k-b=s(\lambda^k-\tilde{\lambda}^k), \tag{3.23}$$

(3.22) can be rearranged to be

$$s(\lambda^k-\lambda^*)^T(\lambda^k-\tilde{\lambda}^k)+r(y^k-y^*)^T(y^k-\tilde{y}^k)\geq r\|y^k-\tilde{y}^k\|_2^2+(\lambda^k-\tilde{\lambda}^k)^TB(y^k-\tilde{y}^k)+s\|\lambda^k-\tilde{\lambda}^k\|_2^2. \tag{3.24}$$

Here $u^{k+1}=u^k-\alpha_kG^{-1}\mathrm{du}$, so we have

$$\begin{aligned}\|u^k-u^*\|_G^2-\|u^{k+1}-u^*\|_G^2 &= 2\alpha_k(u^k-u^*)^TG\mathrm{du}-\alpha_k^2\|G^{-1}\mathrm{du}\|_G^2,\\ &\geq (2\alpha_kc-\alpha_k^2)\|G^{-1}\mathrm{du}\|_G^2,\end{aligned} \tag{3.25}$$

where the inequality follows from (3.14) and (3.24). When we take $\alpha_k=c$, it holds that

$$\|u^k-u^*\|_G^2-\|u^{k+1}-u^*\|_G^2\geq c^2\|G^{-1}\mathrm{du}\|_G^2,$$

where $\kappa=c^2>0$, thus the assertion is proved. □

The results of Lemma 3.1 and Theorem 3.1 show that, when $\|B^TB\|<4rs$ holds, $G^{-1}\mathrm{d}u$ is a descent direction of the merit function $\|u-u^*\|_G^2$ at u^k under G-norm.

Lemma 3.2 *Let u^* be an arbitrary solution of (2.5). When the iterating sequence generated by (3.12) satisfies (3.18) for each k, we can obtain $\|u^k-u^*\|_G^2\to 0$ when $k\to\infty$.*

Proof: By defining a "residual":

$$e(u^k)\triangleq u^k-\tilde{u}^k,$$

solving (2.5) is equivalent to finding a zero point of $e(u)$, *i.e.*, finding a u such that

$$\|e(u)\|=0.$$

By using (3.18), and assuming u^k is not a solution, then the sequence $\{\|u^k-u^*\|_G^2\}$ is strictly monotonic decreasing. Otherwise we can obtain $\|G^{-1}\mathrm{d}u\|_G^2=0$, i.e., $\|e(u^k)\|=0$, which yields contradiction with the assumption that u^k is not a solution.

Summing the inequalities (3.18) over $k=1,...,\infty$, we obtain

$$\|u^1-u^*\|_G^2\geq c\sum_{k=1}^{\infty}\|G^{-1}\mathrm{du}\|_G^2, \tag{3.26}$$

thus

$$\|G^{-1}\mathrm{d}u\|_G^2=r\|y^k-\tilde{y}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2\to 0, \tag{3.27}$$

or equivalently

$$\|y^k-\tilde{y}^k\|^2\to 0\quad\text{and}\quad\|\lambda^k-\tilde{\lambda}^k\|^2\to 0. \tag{3.28}$$

Using the notation $e(u)$, the above results can be reformulated as

$$\|e(u^k)\| \to 0. \tag{3.29}$$

Since $\|u^k - u^*\|_G^2$ is strictly monotonic decreasing, the sequence $\{u^k\}$ is bounded. Then there exists a cluster point, says $\bar{u}$, and a subsequence u^{k^j}, such that $u^{k^j} \Rightarrow \bar{u}$ when $j \to \infty$. Note (3.29) implies $\|e(u^{k^j})\| \to 0$ as $j \to \infty$, together with the continuity of $e(\cdot)$ and the fact $u^{k^j} \to \bar{u}$, we obtain $\|e(\bar{u})\| = 0$.

As a consequence, we obtain $\bar{u} \in \Omega^*$, or equivalently $u^{k^j} \Rightarrow u^*$. By using the monotonicity of sequence $\{\|u^k - u^*\|_G^2\}$, we arrive at the assertion $\|u^k - u^*\|_G \to 0$ as $k \to \infty$. □

By combining the result of Lemma 3.1, 3.2, and Theorem 3.1, we obtain the final result as follows:

Theorem 3.2 *When $\|B^T B\| < 4rs$ holds, the sequence generated by (3.12) satisfies $\|u^k - u^*\|_G^2 \to 0$.*

Compared with the classic LADMM which converges providing $\|B^T B\| < rs$ holds, our new LADMM has relaxed condition $\|B^T B\| < 4rs$, this gives us more freedom in choosing parameters r and s, which could help improving the speed performance of LADMM.

4 Numerical Experiments

4.1 Preparation

In this section, we test our new algorithm by doing experiments on image deblurring and denoising problem. Most of our experiments were done on a notebook computer with Intel Core2 Duo CPU at 2.53GHz, 4GB memory, Windows Vista, and Matlab R2009a, while the experiments in section 4.3 and 4.4 were done on a DELL workstation with Intel Xeon E5506 CPU at 2.13GHz(×8), 10GB memory, Ubuntu 9.10 and Matlab 2009b.

The implementation of our new LADMM for solving partial TVL1 model (2.9) can be obtained by taking $A = \begin{pmatrix} D^1 \\ D^2 \end{pmatrix}$, $B = P_\Omega(K)$, and applying a simple variable substitution. Additionally, in order to make better use of the structure of constraint matrix, we can specify different settings of s to two blocks of multipliers respectively, which allows more flexibility in parameter settings. Some default parameter settings are as follows: $\nu = 1.5$, $c = 10^{-6}$, $\gamma = 1.2$, $s^1 = 0.1$, $s^2 = 0.005$.

Since there is no dedicated solver package for (2.9) at present time to the best of present authors' knowledge, we include FTVD and TVAL3 in our comparison although they are not designed for (2.9). As a consequence, this comparison is not a pure algorithm comparison since all algorithms are designed for different models which yield different solutions. However, this kind of comparison is still reasonable from the practical aspect.

We use the latest version of TVAL3 (v2.2) and did a few necessary revisions to fit our tests. All the parameters settings are left to the default values. In the latest version of FTVD (v4.1), the convolution operation is done by the more efficient FFT instead of `imfilter`. For fairness, we can modify our algorithm and TVAL3 by implementing the same technique, however, for simplicity, we use the second latest version of FTVD(v4.0), and retain the `imfilter` in all the implementations. Additionally, we follow the default settings of the implementation of FTVD v4.0.

We generate the testing examples by the the following procedures: (1) inputting the original image, converting the original integer data into double-precision real data, scaling the values of all entries into the interval [0,1]; (2) blurring it by some convolution (default: `average` convolution kernel with region size `7`) with some boundary condition (default: `circular`); (3) adding impulsive noise such as salt and pepper noise, where the ratio between noisy entries and total entries is entitled by "Noisy Ratio" and denoted by `NR`.

Since our algorithm and TVAL3 are two-stage algorithms for partial model, we need to derive an indices set matrix Ω in the first stage. We use the popular "median filter" algorithm coded by the present author to generate Ω. According to our limited numerical experience, the median filter can remove most corrupted entries, i.e., the accuracy of Ω is very high. To make the problems more challenging and practical, we deteriorate the quality of Ω by randomly choosing some corrupted entries and putting their indices into Ω. To measure the quality of the "contaminated" Ω, we define the ratio between the numbers of corrupted entries and total entries in Ω as `RC` which stands for "Ratio of Corrupted entries". A toy example to produce such a "contaminated" Ω with given `RC` is as follows: (1) for a given 10×10 image with 100 pixels totally, there are 62

(NR=62%) corrupted entries, and 38 uncorrupted entries; (2) some first stage algorithm produces an Ω with 36 entries where 2 uncorrupted entries are missed; (3) set RC, e.g., RC= 0.1, we then choose 4 indices randomly from Ω^C, and put them into Ω, such that the resulting Ω have 10% corrupted entries (RC= $0.1 = 4/(36+4)$). The RC can measure the quality of the Ω in a continuous manner, hence we are able to investigate the performance change in a more precise way. Unless specified otherwise, the default setting of RC is 0.1 in our experiments.

We put major concern on the comparison of the quality of restored images, and the "Signal to Noise Ratio" (SNR) is used as a performance index of image quality for comparison, although it does not always coincide with visual feeling. The computing times are also reported in our comparison, while the computing times in the first stage for acquiring Ω (if needed) is not included.

For fairness, all algorithms start from the same initial point: x starts from f (for full model) or $\bar{f}$ (for partial model), other variables begins with zero. All algorithms stop when either the relative error $\texttt{relerr}(x) \triangleq \|x^k - x^{k+1}\|_F / \|x^k\|_F$ is smaller than some prescribed tolerance tol or the iteration number reaches the maximal iteration number maxit=500. We use different settings of *tol* for different algorithms such that their final SNRs are roughly equal.

4.2 Basic test with high noise ratios

In order to have a quick experience of the performance of our algorithm, we first do a basic test with NR=60% and 80% which is relatively high. The experimental results of images cameraman and clock are shown in Table 1 and Figure 1.

Figure 1: From left to right are the results of FTVD, New algorithm, TVAL3, and SNR history plot. First row: $NR = 60\%$; second row: $NR = 80\%$.

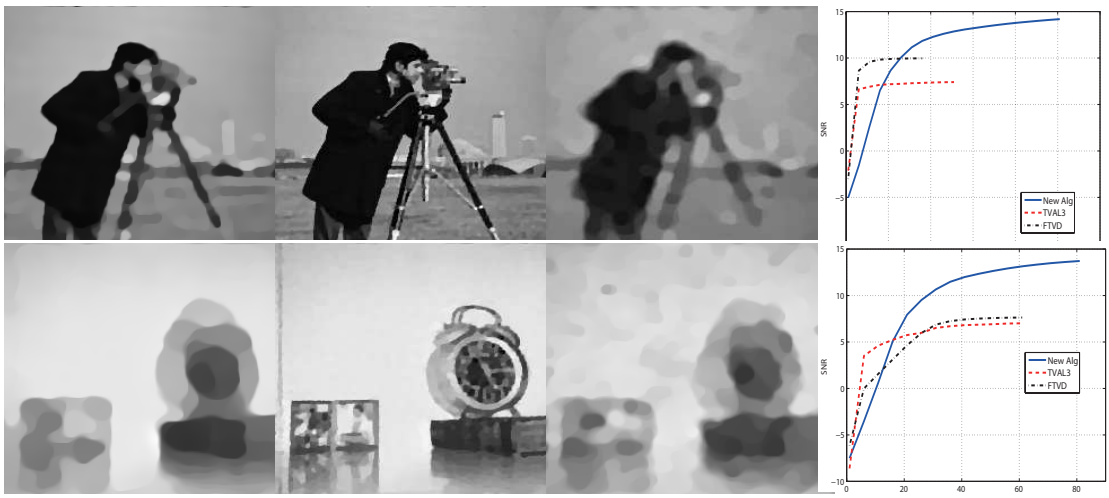


Table 1: Numerical results of basic test.

	FTVD		New algorithm		TVAL3	
Noise Ratio	time	SNR	time	SNR	time	SNR
60%	2.855	9.963	8.798	14.226	5.351	5.507
80%	5.398	7.637	6.973	13.777	6.786	5.121

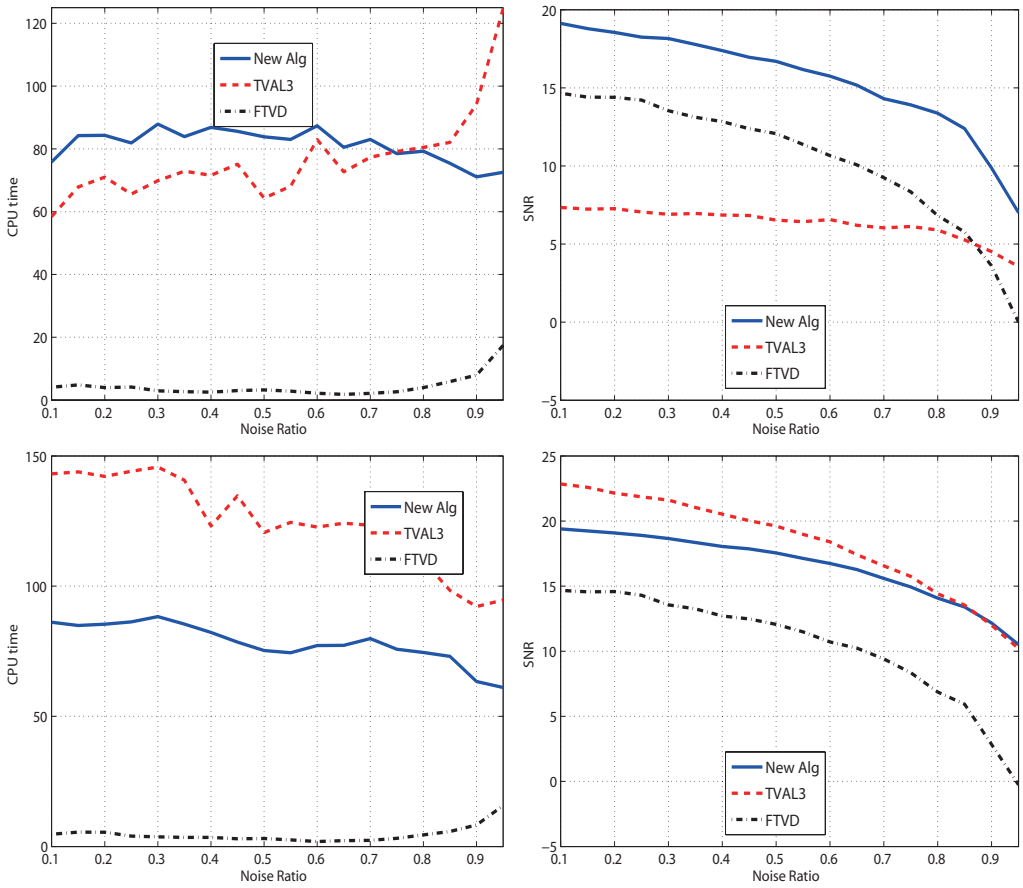
We see from Figure 1 that our algorithm can give a satisfactory result (SNR¿13). In contrary, the FTVD which uses TVL1 model can only produce a blurry result(SNR¡10); the TVAL3 is designed for handling noiseless partial TVL1 model, hence, the deliberately contaminated Ω (note RC=0.1) leads to a result with quite a few unpleasant "patches" as seen in the two subfigures in third column(SNR¡6). This result reveals

the better ability of our algorithm for handling high noise ratio case though it usually takes longer computing time.

4.3 Performance changes as noise ratio varies

The experimental results in subsection 4.1 reveals that our algorithm can better tolerate high noise ratios. To investigate the performance of our algorithm as noise ratio varies in a more precise way, we do comparisons by varying the setting of `NR` between 0.1 and 0.95 with interval 0.05 while other settings are fixed. The results under two settings of `RC` are shown in the first and second row of Figure 2, respectively.

Figure 2: Performance v.s. noise ratio. Left: time, right: SNR. First row: $RC = 0.1$; second row: $RC = 0$



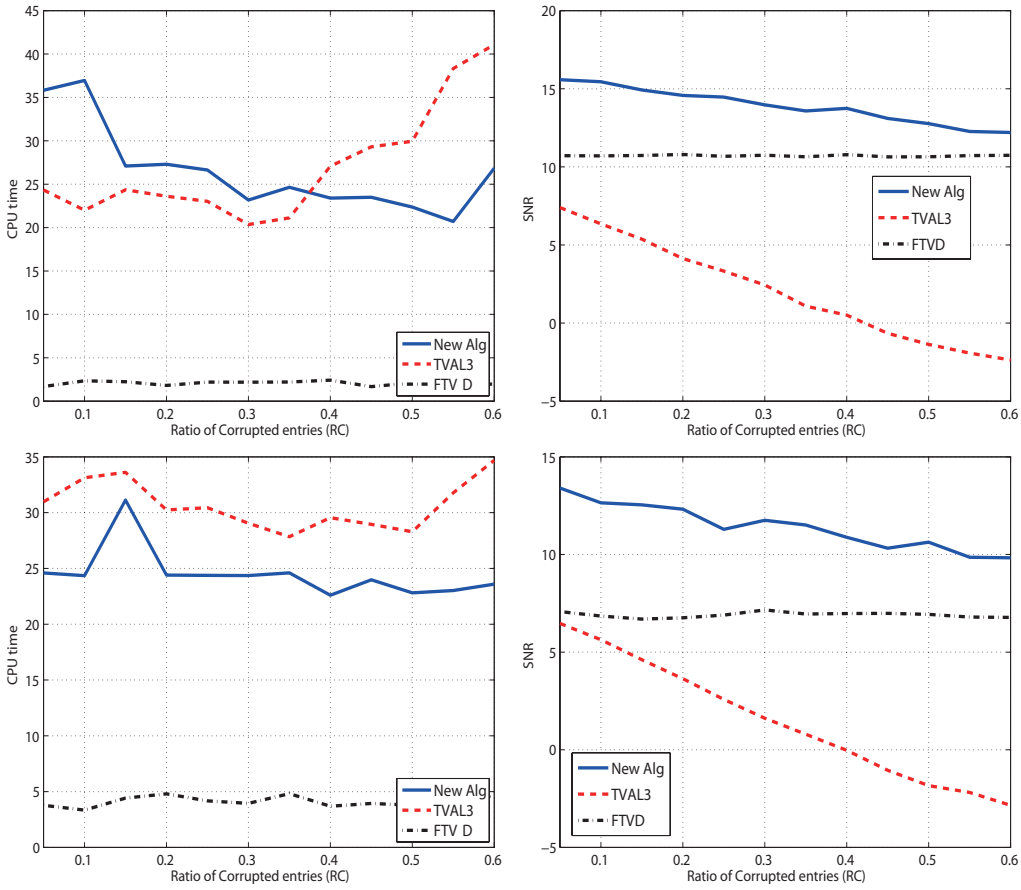
Since larger setting of `NR` complicates the problem, we can observe from Figure 2 that the SNR of the results obtained by all three algorithms decrease as `NR` increases, while their computing times are relatively stable as `NR` increases. In terms of SNR, our algorithm always outperforms FTVD by around 5 and 7 when `RC=0.1` and `RC=0`, respectively; our algorithm outperforms TVAL3 by around 10 when `RC=0.1`; TVAL3 outperforms our algorithm by around 3 when `RC=0`, but this advantage becomes vanishing as `NR` increases. In a word, our algorithm performs stably as `NR` varies.

4.4 Performance change as ratio of corrupted entries varies

A noticeable feature of our algorithm lies in its two-stage scheme, hence its performance can highly rely on the indices matrix Ω obtained by the first stage. This is also true for any algorithm based on partial model such as TVAL3. The corrupted entries can be more difficult to detect in real applications, hence it is meaningful to investigate the performance change as the quality of Ω is deteriorated.

As a consequence, we do experiments by varying the setting of `RC` between 0.05 and 0.6 with interval 0.05. The results for two different settings of `NR` are shown the first and second row of Figure 3, respectively. We need to point out that the maximal setting of `RC`=0.6 might be impractical, but it can reveal the algorithms' ability to tolerate the poor quality of Ω. Note since FTVD uses full model without using Ω, hence we see that the result quality of FTVD is invariant to the change of `RC`, hence we put major concern on the comparison between our algorithm and TVAL3, while the result of FTVD is still shown as reference.

Figure 3: Performance v.s. indices matrix quality. Left: time, right: SNR. First row: $NR = 60\%$; second row: $NR = 80\%$.



We see from Figure 3 that the SNR of both our algorithm and TVAL3 become deteriorated as `RC` increases. On one hand, our algorithm always outperforms the other two algorithms in terms of SNR, on the other hand, our algorithm has satisfactory resistance to the poor quality of indices matrix Ω. Concretely, even under the impractical setting of `RC`=0.6, our algorithm can still produce a reasonably good result (SNR>10). In contrary, under the same setting of `RC`, TVAL3 can not give any meaningful result (SNR<0).

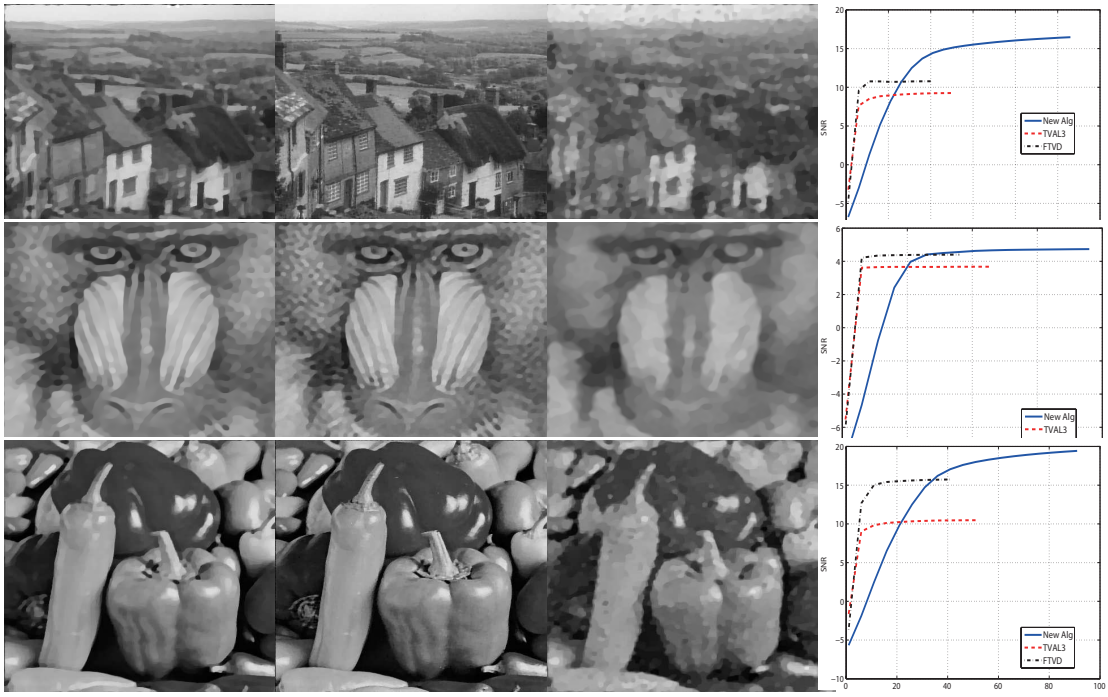
4.5 Miscellaneous tests

As complement, we do some miscellaneous tests in this subsection. We first study the performance under non-circular boundary conditions, we did a test with dirichlet boundary condition, "average" convolution kernel and `NR=0.6`, and the results are shown in the first row of figure 4. We can observe flaws near the boundary of images restored by TVAL3 and FTVD, while our algorithm can still give a satisfactory result.

To investigate the performance stability under different convolution kernels, another test was done with two non-default convolution kernels: `Gaussian` kernel with region size `21` and standard deviation `5`; `Disk` kernel with `radius=9`, and the results are shown in the second and third rows of Figure 4. We see that under

these two types of convolution kernels, our algorithm still gives satisfactory result in terms of image quality. This indicates that the performance of our algorithm is stable under different settings of convolution kernel.

Figure 4: From left to right to below are the results of FTVD, New algorithm, and TVAL3, and SNR history plot. First row: "average" kernel with dirichlet boundary condition; second row: "Gaussian" kernel; third row: "Disk" kernel.



5 Conclusions

In this paper, we consider a partial convolution TVL1 model for image deblurring and denoising, and propose a new linearized ADMM to solve it. From the theoretical aspect, our algorithm has relaxed parameter condition which allows more flexibility on choosing parameters; it also implements an extension step which enables faster computing speed. From the practical aspect, our algorithm can adapt to a wider range of problems: compared with some existing efficient algorithms, our algorithm can tolerate higher noise ratio by handling a partial convolution model; additionally, it can handle the non-circular boundary condition while some existing algorithms such as FTVD can only handle the circular boundary condition. Preliminary experimental results show that our algorithm performs stably and efficiently compared with some state-of-the-art algorithms.

Although the new LADMM shows satisfactory performance, it can still be improved further. For instance, we can use the BB step size and Nesterov's accelerating technique to accelerate the convergence. Furthermore, the new LADMM can be extended to solve some related models such as multichannel TVL1 model. However, to restrict the capacity of the present paper, these are left to the future works.

References

[1] R. Acar and C. R. Vogel. Analysis of total variation penalty methods. *Inverse Probl.*, 10:1217–1229, 1994.

[2] S. Alliney. Digital filters as absolute norm regularizers. *IEEE Trans. Signal Proces.*, 40(6):1548–1562, 1992.

[3] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[5] S. Becker, J. Bobin, and E. Candes. Nesta: A fast and accurate first-order method for sparse recovery. Technical report, California Institute of Technology, 2009.

[6] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.

[7] J.M. Bioucas-Dias and M.A.T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. Image Proces.*, 16(12):2992–3004, Dec. 2007.

[8] S. Boyd, E. Parikh, N.and Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[9] J. Cai, R. H. Chan, and M. Nikolova. Two phase methods for deblurring images corrupted by impulse plus gaussian noise. *Inverse Probl. Imag.*, 2(2):187–204, 2008.

[10] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011.

[11] R. H. Chan, C. W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detector and detail-preserving regularization. *IEEE Trans. Imag. Process.*, 14(10):1479–1485, 2005.

[12] R. H. Chan, Min Tao, and Xiaoming Yuan. Linearized alternating direction method of multipliers for constrained linear least-squares problem. *E. Asian J. Appl. Math.*, 2(4):326–341, 2012.

[13] T. F. Chan and S. Esedoglu. Aspects of total variation regularized l1 function approximation. *SIAM J. Appl. Math.*, 65(5):1817 – 1837, 2005.

[14] T. F. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977, 1999.

[15] C. Chen, R. H. Chan, S. Ma, and J. Yang. Inertial proximal admm for linearly constrained separable convex optimization. *SIAM J. Imaging Sci.*, 8(4):2239–2267, 2015.

[16] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Math. Program. Series A*, 155(1):57–79, January 2016.

[17] C. Chen, Y. Shen, and Y. You. On the convergence analysis of the alternating direction method of multipliers with three blocks. *Abstr. Appl. Anal.*, 2013, Article ID 183961:7 pages, 2013.

[18] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. Huang. Total variation models for variable lighting face recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 28(9):1519–1524, 2006.

[19] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57:1413–1457, 2004.

[20] J. Douglas and H. H. Rachford. On the numerical solution of the heat conduction problem in 2 and 3 space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[21] E. Esser. Applications of lagrangian-based alternating direction methods and connections to split bregman. *Manuscript*, 2009. ftp://ftp.math.ucla.edu/pub/camreport/cam09-31.pdf.

[22] Glowinski R. (Eds.) Fortin, M. *Augmented Lagrangian methods: Applications to the numerical solution of Boundary-Value Problems*. North-Holland, Amsterdam, 1983.

[23] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*. Horth-Hollan, Amsterdam, 1983.

[24] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Comput. Math. Appl.*, 2:17–40, 1976.

[25] R. Glowinski. *Numerical methods for nonlinear variational problems*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo,, 1984.

[26] R. Glowinski and A. Marrocco. Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de dirichlet nonlineaires. *Rev. Francaise dAut. Inf. Rech. Oper., R-2*, pages 41–76, 1975.

[27] D. Han and X. Yuan. A note on the alternating direction method of multipliers. *J. Optim. Theory Appl.*, 155(1):227–238, 2012.

[28] B. He, M. Tao, and X. Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM J. Optim.*, 22(2):313–340, 2012.

[29] B. He, M. Xu, and X. Yuan. Solving large-scale least squares semidefinite programming by alternating direction methods. *SIAM. J. Matrix Anal. & Appl.*, 32(1):136–152, 2011.

[30] B. He and H. Yang. Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. *Oper. Res. Lett.*, 23:151–161, 1998.

[31] B.S. He, L. Liao, D. Han, and H. Yang. A new inexact alternating directions method for monotone variational inequalities. *Math. Program.*, 92(1, Ser. A):103–118, 2002.

[32] M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

[33] M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers. *Arxiv preprint,https://arxiv.org/abs/1208.3922*, 2013.

[34] Ng M. Wen Y. Huang, Y. A fast total variation minimization method for image restoration. *Multiscale Model. Sim.*, 7:774–795, 2008.

[35] C. Li, W. Yin, and Y. Zhang. Tv minimization by augmented lagrangian and alternating direction algorithms. Technical report, Department of CAAM, Rice University, Houston, Texas, 77005, 2009. http://www.caam.rice.edu/ optimization/L1/TVAL3/.

[36] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16:964–979, 1979.

[37] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–684, 1989.

[38] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

[39] M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers. *SIAM J. Numer. Anal.*, 40(3):965 – 994, 2002.

[40] S. Osher, Y. Mao, B. Dong, and W. Yin. Fast linearized bregman iteration for compressive sensing and sparse denoising. *Rice CAAM technical report, Rice_CAAM_TR08-07.PDF*, 2008.

[41] Y. Ouyang, Y. Chen, G. Lan, and E. Jr. Pasiliao. An accelerated linearized alternating direction method of multipliers. *SIAM J Imaging Sci.*, 8(1):644–681, 2015.

[42] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1972.

[43] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[44] L. Rudin and S. Osher. Total variation based image restoration with free local constraints. *Proc. 1st IEEE ICIP*, 1(31–35), 1994.

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

[46] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. Winston, New York, 1977.

[47] C. Vogel and M. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Proces.*, 7(6):813–824, 1998.

[48] X. Wang and X. Yuan. The linearized alternating direction method of multipliers for dantzig selector. *SIAM J. Sci. Comput.*, 34(5):2792–2811, 2012.

[49] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Image Sci.*, 1(3):248–272, 2008.

[50] T. Wu. Variable splitting based method for image restoration with impulse plus gaussian noise. *Math. Probl. Eng.*, 2016(Article ID 3151303):16 pages, 2016.

[51] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM J. Sci. Comput.*, 33(1):250–278, 2011.

[52] J. Yang, Y. Zhang, and W. Yin. An efficient tvl1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM J. Sci. Comput.*, 31(4):2842–2865, 2009.

[53] C. Ye and X. Yuan. A descent method for structured monotone variational inequalities. *Optim. Methods Softw.*, 22(2):329–338, 2007.

[54] W. Yin, D. Goldfarb, and S. Osher. The total variation regularized L^1 model for multiscale decomposition. *Multiscale Model. Sim.*, 6(1):190–211, 2006.