


## Article

# Image Restoration Using a Fixed-Point Method for a TVL2 Regularization Problem

Kyoum Sun Kim <sup>1</sup> and Jae Heon Yun <sup>2,\*</sup> <sup>1</sup> Department of Mathematics, Chungbuk National University, Cheongju 28644, Korea; giunsun@naver.com<sup>2</sup> Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 28644, Korea

\* Correspondence: gmjae@chungbuk.ac.kr; Tel.: +82-10-8843-2250

Received: 16 November 2019; Accepted: 16 December 2019; Published: 18 December 2019



**Abstract:** In this paper, we first propose a new TVL2 regularization model for image restoration, and then we propose two iterative methods, which are fixed-point and fixed-point-like methods, using CGLS (Conjugate Gradient Least Squares method) for solving the new proposed TVL2 problem. We also provide convergence analysis for the fixed-point method. Lastly, numerical experiments for several test problems are provided to evaluate the effectiveness of the proposed two iterative methods. Numerical results show that the new proposed TVL2 model is preferred over an existing TVL2 model and the proposed fixed-point-like method is well suited for the new TVL2 model.

**Keywords:** image restoration; total variation; fixed-point method; non-expansive operator; proximity operator

## 1. Introduction

Image restoration is the fundamental problem in image processing that recovers a true image from a blurry and noisy image. The problem of image restoration usually reduces to find the optimal solution  $u \in \mathbb{R}^m$  based on the following model:

$$f = Au + \varepsilon, \quad (1)$$

where  $A \in \mathbb{R}^{m \times m}$  is a blurring operator,  $\varepsilon \in \mathbb{R}^m$  is an unknown white Gaussian noise with variance  $\sigma$ , and  $f$  and  $u$  denote the observed degraded image and the original image, respectively. Our purpose is to restore the original image  $u$  from blurred and noisy image  $f$  as well as possible.

Over the past few decades, optimization techniques and various variation models [1–3] have been widely studied and applied in many image processing fields. The well-known ROF (Rudin–Osher–Fatemi) total variation model [3] produces the deblurred image given by the following minimization problem:

$$\min_u \left\{ \frac{1}{2} \|Au - f\|_2^2 + \beta TV(u) \right\}, \quad (2)$$

where  $TV(u)$  is the total variation (TV) of  $u$  and  $\beta > 0$  is a regularization parameter.

Many computational methods for solving problem (2) have been proposed in recent years. For example, the time-marching PDE method, the subgradient descent method, the Newton-like method, the second-order cone programming method, the lagged diffusivity fixed-point method, and the split Bregman method have been proposed by many researchers (see [1,3–13] for details).

Recently, Chen et al. [14] proposed a fixed-point method for solving the following constrained TVL2 deblurring problem:

$$\min_{u \in C} \left\{ \frac{1}{2} \|Au - f\|_2^2 + \frac{\alpha}{2} \|u\|_2^2 + \beta TV(u) \right\}, \quad (3)$$

where  $\alpha$  and  $\beta$  are positive constants, and  $C$  is a closed convex subset of  $\mathbb{R}^{n^2}$ . In this paper, we only consider the case for  $C = \mathbb{R}^{n^2}$ . That is, we only consider the unconstrained TVL2 problem (3) with  $C = \mathbb{R}^{n^2}$ . This approach motivates us to propose a new TVL2 deblurring model

$$\min_{u \in \mathbb{R}^{n^2}} \left\{ \frac{1}{2} \|Au - f\|_2^2 + \alpha \|u\|_2 + \beta TV(u) \right\}, \quad (4)$$

where  $A \in \mathbb{R}^{n^2 \times n^2}$  is a blurring matrix,  $u \in \mathbb{R}^{n^2}$  is an original image,  $f \in \mathbb{R}^{n^2}$  is a degraded image,  $\alpha$  and  $\beta$  are positive regularization parameters, and  $TV(u)$  stands for the isotropic TV of  $u$ . Note that the new TVL2 model (4) uses a non-smooth term  $\|u\|_2$  instead of using a smooth term  $\|u\|_2^2$  for the purpose of better preserving the edges and corners in the restored images. The isotropic TV of  $u$  is defined by

$$TV(u) = \sum_{i=1}^{n^2} \sqrt{|(\nabla u)_i^x|^2 + |(\nabla u)_i^y|^2} = \sum_{i=1}^{n^2} \left\| \begin{pmatrix} (\nabla u)_i^x \\ (\nabla u)_i^y \end{pmatrix} \right\|_2,$$

where the discrete gradient operator  $\nabla : \mathbb{R}^{n^2} \rightarrow \mathbb{R}^{2n^2}$  is defined by

$$(\nabla u)_i = ((\nabla u)_i^x, (\nabla u)_i^y), \quad i = 1, 2, \dots, n^2$$

with

$$(\nabla u)_i^x = \begin{cases} 0 & \text{if } i \bmod n = 1, \\ u_i - u_{i-1} & \text{if } i \bmod n \neq 1, \end{cases} \quad \text{and} \quad (\nabla u)_i^y = \begin{cases} 0 & \text{if } i \leq n, \\ u_i - u_{i-n} & \text{if } i > n. \end{cases}$$

Notice that TVL2 problems (3) have a unique solution since its objective function is strictly convex, while the new TVL2 problem (4) may not have a unique solution since its objective function is just convex, not strictly convex.

The purpose of this paper is to propose two iterative methods, which are fixed-point and fixed-point-like methods, using CGLS (Conjugate Gradient Least Squares method [15]) for solving the new proposed TVL2 problem (4). This paper is organized as follows. In Section 2, we introduce some definitions and properties that are used in this paper. In Section 3, we first propose a fixed-point method using CGLS for solving TVL2 problem (4), and then we provide convergence analysis for the fixed-point method. In Section 4, we propose a fixed-point-like method using CGLS for solving TVL2 problem (4). In Section 5, we just provide the split Bregman methods for solving TVL2 problems (3) and (4) to see how efficiently the fixed-point and fixed-point-like methods perform. In Section 6, we describe how to carry out numerical experiments for several test problems in order to evaluate the effectiveness of the proposed two iterative methods. This can be done by comparing their performances for TVL2 problem (4) with those of the fixed-point method proposed in [14] for TVL2 problem (3) and the split Bregman methods for TVL2 problems (3) and (4). In Section 7, we provide numerical results for all test problems. Lastly, some conclusions are drawn.

## 2. Preliminaries

In this section, we briefly refer to some definitions and important properties which will be the foundation for the development of algorithms proposed in this paper.

**Definition 1.** Let  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  be a proper, convex, and lower semi-continuous function (see [16]). The proximity operator of  $\varphi$  at  $x \in \mathbb{R}^n$  is defined by

$$\text{prox}_{\varphi}(x) = \operatorname{argmin} \left\{ \frac{1}{2} \|u - x\|_2^2 + \varphi(u) : u \in \mathbb{R}^n \right\}. \quad (5)$$

**Definition 2.** Let  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  be a proper, convex, and lower semi-continuous function. The subdifferential of  $\varphi$  at  $x \in \mathbb{R}^n$  is defined by

$$\partial\varphi(x) = \{y \in \mathbb{R}^n : \varphi(z) \geq \varphi(x) + \langle y, z - x \rangle, \forall z \in \mathbb{R}^n\}. \quad (6)$$

**Definition 3.** A nonlinear operator  $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is called non-expansive if for any  $x, y \in \mathbb{R}^n$ ,

$$\|H(x) - H(y)\|_2 \leq \|x - y\|_2. \quad (7)$$

**Definition 4.** A nonlinear operator  $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is called firmly non-expansive if for any  $x, y \in \mathbb{R}^n$ ,

$$\|H(x) - H(y)\|_2^2 \leq \langle x - y, H(x) - H(y) \rangle. \quad (8)$$

It is easy to show that a firmly non-expansive operator is non-expansive. The following proposition shows a relationship between the proximity operator and the subdifferential of a convex function.

**Proposition 1.** If  $\varphi$  is a convex function defined on  $\mathbb{R}^n$  and  $x, y \in \mathbb{R}^n$ , then (see [16–18])

$$y \in \partial\varphi(x) \Leftrightarrow x = \text{prox}_{\varphi}(x + y). \quad (9)$$

Let  $\varphi : \mathbb{R}^{2n^2} \rightarrow \mathbb{R}$  be a convex function defined by

$$\varphi(d) = \sum_{i=1}^{n^2} \left\| \begin{pmatrix} d_i \\ d_{n^2+i} \end{pmatrix} \right\|_2 \quad \text{for each } d = (d_i) \in \mathbb{R}^{2n^2}, \quad (10)$$

and let  $B$  be a  $2n^2 \times n^2$  matrix that represents a discrete gradient operator  $\nabla$ , which is

$$B = \begin{pmatrix} I_n \otimes D \\ D \otimes I_n \end{pmatrix}, \quad (11)$$

where  $I_n$  is the  $n \times n$  identity matrix,  $\otimes$  denotes the Kronecker product, and  $D$  is the first order finite difference matrix of order  $n$

$$D = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}.$$

Then, the isotropic TV of  $u \in \mathbb{R}^{n^2}$  can be expressed as

$$\text{TV}(u) = (\varphi \circ B)(u). \quad (12)$$

We now provide the fixed-point method, called Algorithm 1, for solving TVL2 problem (3), which was proposed by Chen et al. [14].

**Algorithm 1** Fixed-point method for TVL2 problem (3)

---

```

1: Given : observed image  $f$ , positive parameters  $\alpha, \beta, \lambda$  and  $\kappa \in (0, 1)$ 
2: Initialization :  $v^0 = 0$  and  $u^0 = f$ 
3: for  $k = 0$  to  $maxit$  do
4:   Solve  $(A^T A + \alpha I)u^{k+1} = A^T f - \lambda B^T v^k$  for  $u^{k+1}$ 
5:    $v^{k+1} = \kappa v^k + (1 - \kappa)(I - \text{prox}_{\frac{\beta}{\lambda} \varphi})(v^k + Bu^{k+1})$ 
6:   if  $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$  then
7:     Stop
8:   end if
9: end for

```

---

Notice that line 4 of Algorithm 1 is solved using CGLS instead of using CG (Conjugate Gradient method [19–21]) since the linear system in line 4 is equivalent to solving the following least squares problem

$$\min_u \left\| \begin{pmatrix} f \\ -\frac{\lambda}{\sqrt{\alpha}} B^T v^k \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\alpha} I \end{pmatrix} u \right\|_2^2.$$

For all algorithms presented in this paper, *maxit* denotes the maximum number of iterations, and *tol* denotes the tolerance value of the stopping criterion.

**3. Fixed-Point Method for TVL2 Problem (4)**

In this section, we propose a fixed-point method using CGLS for solving the new TVL2 regularization problem (4). From relation (12), TVL2 problem (4) can be expressed as

$$\min_{u \in \mathbb{R}^{n^2}} \left\{ \frac{1}{2} \|Au - f\|_2^2 + \alpha \|u\|_2 + \beta(\varphi \circ B)(u) \right\}. \quad (13)$$

Using Proposition 1, we can obtain the following property for a solution of TVL2 problem (13).

**Theorem 1.** If  $\varphi$  is a real valued convex function on  $\mathbb{R}^{2n^2}$ ,  $B$  is a  $2n^2 \times n^2$  matrix,  $A$  is an  $n^2 \times n^2$  matrix,  $u$  is a solution to model (13), then, for any  $\gamma, \lambda > 0$  there exist vectors  $a \in \mathbb{R}^{n^2}$  and  $b \in \mathbb{R}^{2n^2}$  such that

$$a = (I - \text{prox}_{\frac{1}{\gamma} \|\cdot\|_2})(u + a), \quad (14)$$

$$b = (I - \text{prox}_{\frac{\beta}{\lambda} \varphi})(Bu + b), \quad (15)$$

$$(A^T A)u = A^T f - \alpha \gamma a - \lambda B^T b. \quad (16)$$

Conversely, if there exists  $\gamma, \lambda > 0$ ,  $a \in \mathbb{R}^{n^2}$ ,  $b \in \mathbb{R}^{2n^2}$  and  $u \in \mathbb{R}^{n^2}$  satisfying Equations (14)–(16), then  $u$  is a solution to model (13).

**Proof.** Assume that  $u \in \mathbb{R}^{n^2}$  is a solution to the model (13). By Fermat's rule in convex analysis for model (13), we obtain the following equivalent relation for the solution  $u$

$$\begin{aligned} 0 &\in A^T(Au - f) + \alpha \cdot \partial(\|\cdot\|_2)(u) + \beta B^T \circ (\partial\varphi) \circ (Bu) \\ &= A^T(Au - f) + \alpha \cdot \partial(\|\cdot\|_2)(u) + B^T \partial(\beta\varphi)(Bu). \end{aligned} \quad (17)$$

For any  $\gamma, \lambda > 0$ , we can choose two vectors  $a \in \partial(\frac{1}{\gamma} \|\cdot\|_2)(u)$  and  $b \in \partial(\frac{\beta}{\lambda} \varphi)(Bu)$  satisfying

$$A^T(Au - f) + \alpha \gamma a + \lambda B^T b = 0. \quad (18)$$

Using Proposition 1, one obtains

$$u = \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}(u + a) \quad \text{and} \quad Bu = \text{prox}_{\frac{\beta}{\lambda}\varphi}(Bu + b). \quad (19)$$

From Equation (19), we obtain Equations (14) and (15). In addition, from Equation (18), we obtain Equation (16).

Conversely, suppose that there exist  $\gamma, \lambda > 0, a, b$  and  $u \in \mathbb{R}^{n^2}$  satisfying Equations (14)–(16). From Equation (16), we obtain Equation (18). By Proposition 1, Equations (14) and (15) ensure  $a \in \partial(\frac{1}{\gamma}\|\cdot\|_2)(u)$  and  $b \in \partial(\frac{\beta}{\lambda}\varphi)(Bu)$ , respectively. Using these relations and Equation (18), one obtains

$$\begin{aligned} 0 &= A^T(Au - f) + \alpha\gamma a + \lambda B^T b \\ &\in A^T(Au - f) + \alpha\partial(\|\cdot\|_2)(u) + \beta B^T \circ (\partial\varphi) \circ (Bu). \end{aligned}$$

Consequently, Equation (17) holds. Thus,  $u$  is a solution to model (13).  $\square$

From Equations (14)–(16) in Theorem 1, we can develop a fixed-point algorithm which converges to a solution to TVL2 problem (4). We now describe how to develop the fixed-point algorithm. Let  $u$  be an approximate solution to the ill-conditioned linear system (16) in Theorem 1. Then,  $u$  can be expressed as

$$u = M(A^T f - \alpha\gamma a - \lambda B^T b), \quad (20)$$

where  $M$  is a symmetric positive semi-definite matrix approximating an inverse of the linear system matrix in (16). For example, we can choose  $M = (A^T A)_r^\dagger$ , which is a truncated pseudoinverse of  $A^T A$  using the  $r$  largest positive singular values of  $A^T A$ . Substituting Equation (20) into Equations (14) and (15), one obtains

$$a = (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}) \left( (I_{n^2} - \alpha\gamma M)a - \lambda M B^T b + M A^T f \right), \quad (21)$$

$$b = (I - \text{prox}_{\frac{\beta}{\lambda}\varphi}) \left( -\alpha\gamma B M a + (I_{2n^2} - \lambda B M B^T)b + B M A^T f \right). \quad (22)$$

Let us define some operators. For the given convex functions  $\frac{1}{\gamma}\|\cdot\|_2$  on  $\mathbb{R}^{n^2}$  and  $\frac{\beta}{\lambda}\varphi$  on  $\mathbb{R}^{2n^2}$ , we define an operator  $T : \mathbb{R}^{3n^2} \rightarrow \mathbb{R}^{3n^2}$  at a vector  $\begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{3n^2}$  with  $u \in \mathbb{R}^{n^2}$  and  $v \in \mathbb{R}^{2n^2}$  as follows:

$$T \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(u) \\ (I - \text{prox}_{\frac{\beta}{\lambda}\varphi})(v) \end{pmatrix}. \quad (23)$$

We also introduce an affine transformation  $L : \mathbb{R}^{3n^2} \rightarrow \mathbb{R}^{3n^2}$  defined, for all  $\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{3n^2}$  with  $a \in \mathbb{R}^{n^2}$  and  $b \in \mathbb{R}^{2n^2}$ , by

$$L \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} I_{n^2} - \alpha\gamma M & -\lambda M B^T \\ -\alpha\gamma B M & I_{2n^2} - \lambda B M B^T \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} M A^T f \\ B M A^T f \end{pmatrix}, \quad (24)$$

and an operator  $G : \mathbb{R}^{3n^2} \rightarrow \mathbb{R}^{3n^2}$  defined by

$$G = T \circ L. \quad (25)$$

Then, Equations (21) and (22) can be expressed as

$$w = Gw, \quad (26)$$

where  $w = \begin{pmatrix} a \\ b \end{pmatrix}$ .

**Proposition 2.** The operator  $G$  defined by (25) has a fixed point.

**Proof.** Since a solution of TVL2 problem (13) exists, from Equation (26) and the first part of the proof of Theorem 1,  $G$  has a fixed point.  $\square$

**Lemma 1.** If the operator  $T$  is defined by Equation (23), then the operator  $T$  is non-expansive.

**Proof.** Note that  $I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}$  and  $I - \text{prox}_{\frac{\beta}{\lambda}\varphi}$  are firmly non-expansive and thus non-expansive [7]. For any vectors  $s = \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{3n^2}$  and  $t = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{3n^2}$  with  $u, a \in \mathbb{R}^{n^2}$  and  $v, b \in \mathbb{R}^{2n^2}$

$$\begin{aligned} \|T(s) - T(t)\|_2^2 &= \left\| \begin{pmatrix} (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(u) - (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(a) \\ (I - \text{prox}_{\frac{\beta}{\lambda}\varphi})(v) - (I - \text{prox}_{\frac{\beta}{\lambda}\varphi})(b) \end{pmatrix} \right\|_2^2 \\ &= \left\| (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(u) - (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(a) \right\|_2^2 \\ &\quad + \left\| (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(v) - (I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(b) \right\|_2^2 \\ &\leq \|u - a\|_2^2 + \|v - b\|_2^2 \\ &= \left\| \begin{pmatrix} u - a \\ v - b \end{pmatrix} \right\|_2^2 = \|s - t\|_2^2. \end{aligned}$$

Hence, one obtains  $\|T(s) - T(t)\|_2 \leq \|s - t\|_2$ , which means that the operator  $T$  is non-expansive.  $\square$

Let

$$c := \begin{pmatrix} MA^T f \\ BMA^T f \end{pmatrix} \in \mathbb{R}^{3n^2}, \quad P := \begin{pmatrix} I_{n^2} \\ B \end{pmatrix}, \quad R := \begin{pmatrix} \alpha\gamma I_{n^2} & 0 \\ 0 & \lambda I_{2n^2} \end{pmatrix}.$$

Then, Label (24) can be expressed as

$$Lw = (I_{3n^2} - PMP^T R)w + c. \quad (27)$$

**Proposition 3.** If  $\varphi$  is a convex function on  $\mathbb{R}^{2n^2}$ ,  $B$  is a  $2n^2 \times n^2$  matrix,  $A$  is an  $n^2 \times n^2$  matrix and  $\alpha, \gamma, \lambda$  are positive constants such that  $\|I_{3n^2} - PMP^T R\|_2 \leq 1$ , then  $G$  is non-expansive.

**Proof.** Since the operator  $T$  is non-expansive by Lemma 1, for all  $w_1, w_2 \in \mathbb{R}^{3n^2}$ , we have

$$\begin{aligned} \|G(w_1) - G(w_2)\|_2 &= \|T(Lw_1) - T(Lw_2)\|_2 \\ &\leq \|Lw_1 - Lw_2\|_2. \end{aligned}$$

By the assumption  $\|I_{3n^2} - PMP^T R\|_2 \leq 1$  and Equation (27), we obtain

$$\begin{aligned} \|Lw_1 - Lw_2\|_2 &= \|(I_{3n^2} - PMP^T R)(w_1 - w_2)\|_2 \\ &\leq \|I_{3n^2} - PMP^T R\|_2 \|w_1 - w_2\|_2 \\ &\leq \|w_1 - w_2\|_2. \end{aligned}$$

Hence,  $G$  is non-expansive.  $\square$

Let  $S : \mathbb{R}^N \rightarrow \mathbb{R}^N$  be an operator. Then, the Picard iteration of the operator  $S$  is defined by

$$x_{i+1} = Sx_i \quad (i = 0, 1, 2, \dots)$$

for a given vector  $x_0 \in \mathbb{R}^N$ . For  $\kappa \in (0, 1)$ , the  $\kappa$ -averaged operator  $S_\kappa$  of  $S$  is defined by

$$S_\kappa = \kappa I + (1 - \kappa)S.$$

**Proposition 4** (Optial  $\kappa$ -averaged Theorem). *Let  $C$  be a closed convex set in  $\mathbb{R}^{3n^2}$  and let  $S : C \rightarrow C$  be a non-expansive mapping with at least one fixed point. Then, for any  $w^0 \in C$  and  $\kappa \in (0, 1)$ , the Picard iteration of  $S_\kappa$  converges to a fixed point of  $S$  (see [22]).*

**Theorem 2.** *If  $\varphi$  is a convex function,  $A$  is an  $n^2 \times n^2$  matrix,  $B$  is a  $2n^2 \times n^2$  matrix and  $\alpha, \gamma, \lambda > 0$  are positive constants such that  $\|I_{3n^2} - PMP^T R\|_2 \leq 1$ , then for any  $\kappa \in (0, 1)$  the Picard iteration of  $G_\kappa$  converges to a fixed point of  $G$ .*

**Proof.** From Propositions 2 and 3, we know that the operator  $G$  has a fixed point and is non-expansive. Hence, for any  $w^0 \in \mathbb{R}^{3n^2}$  and  $\kappa \in (0, 1)$ , the Picard iteration of  $G_\kappa$  converges to a fixed point of  $G$  by Proposition 4.  $\square$

For a square matrix  $K$ , let  $\rho(K)$  denote the spectral radius of  $K$ . Then, the following lemma can be obtained.

**Lemma 2.** *Let  $\varphi$  and  $B$  be defined by Equations (10) and (11) respectively, and let  $A$  be a given  $n^2 \times n^2$  blurring matrix. If we choose  $\alpha$ ,  $\gamma$  and  $\lambda$  such that*

$$0 < \lambda = \alpha\gamma < \frac{2}{\rho(PMP^T)},$$

*then  $\|I_{3n^2} - PMP^T R\|_2 \leq 1$  and thus the operator  $G$  is non-expansive.*

**Proof.** Since  $\lambda = \alpha\gamma$  and thus  $R = \lambda I_{3n^2}$ , one obtains

$$I_{3n^2} - PMP^T R = I_{3n^2} - \lambda PMP^T.$$

Note that  $PMP^T$  is a symmetric positive semi-definite matrix. Hence,

$$\begin{aligned} \|I_{3n^2} - PMP^T R\|_2 &= \rho(I_{3n^2} - \lambda PMP^T) \\ &= \max\{|1 - \lambda \cdot \lambda_{\min}(PMP^T)|, |1 - \lambda \cdot \lambda_{\max}(PMP^T)|\}, \end{aligned}$$

where  $\lambda_{\min}(PMP^T)$  and  $\lambda_{\max}(PMP^T)$  denote the minimum and maximum eigenvalues of  $PMP^T$ , respectively. Since  $0 < \lambda = \alpha\gamma < \frac{2}{\rho(PMP^T)}$ ,  $0 \leq \lambda \cdot \lambda_{\min}(PMP^T) < 2$  and  $0 < \lambda \cdot \lambda_{\max}(PMP^T) < 2$ . Hence, one obtains

$$\|I_{3n^2} - PMP^T R\|_2 \leq 1.$$

Therefore, the operator  $G$  is non-expansive by Proposition 3.  $\square$

**Theorem 3.** *If the assumptions of Lemma 2 hold and  $\kappa \in (0, 1)$ , then the Picard iteration of  $G_\kappa$  converges to a fixed point of  $G$ .*

**Proof.** The proof follows from Lemma 2 and Theorem 2.  $\square$

From Theorem 2 and the Picard iteration of the  $\kappa$ -averaged operator  $G_\kappa = \kappa I + (1 - \kappa)G$ , we can obtain a fixed-point method, called Algorithm 2, which converges to a solution to TVL2 problem (4).

**Algorithm 2** Fixed-point method for TVL2 problem (4)

---

```

1: Given : observed image  $f$ , positive parameters  $\alpha, \beta, \gamma, \lambda$  and  $\kappa \in (0, 1)$ 
2: Initialization :  $a^0 = 0, b^0 = 0$  and  $u^0 = f$ 
3: for  $k = 0$  to  $maxit$  do
4:    $\hat{a}^{k+1} = \left( I - \text{prox}_{\frac{1}{\gamma} \|\cdot\|_2} \right) (u^k + a^k)$ 
5:    $a^{k+1} = \kappa a^k + (1 - \kappa) \hat{a}^{k+1}$ 
6:    $\hat{b}^{k+1} = \left( I - \text{prox}_{\frac{\beta}{\lambda} \varphi} \right) (Bu^k + b^k)$ 
7:    $b^{k+1} = \kappa b^k + (1 - \kappa) \hat{b}^{k+1}$ 
8:   Solve  $A^T A u^{k+1} = A^T f - \alpha \gamma a^{k+1} - \lambda B^T b^{k+1}$  for  $u^{k+1}$ 
9:   if  $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$  then
10:     Stop
11:   end if
12: end for

```

---

The linear system in line 8 of Algorithm 2 is ill-conditioned, so we need to consider how to find an approximate solution to the ill-conditioned linear system. A typical method for finding an approximate solution to the linear system is

$$u^{k+1} = M (A^T f - \alpha \gamma a^{k+1} - \lambda B^T b^{k+1}),$$

where  $M = (A^T A)_r^\dagger$ . However, computation of  $(A^T A)_r^\dagger$  is very time-consuming when  $A$  is large. Thus, we want to propose a different approach for finding an approximate solution to the linear system in line 8 of Algorithm 2. We first split the coefficient matrix  $A^T A$  into

$$A^T A = (A^T A + \delta D) - (\delta D), \quad (28)$$

where  $\delta$  is a positive constant and  $D = \text{diag}(A^T A)$  is a diagonal part of  $A^T A$ . Then, the ill-conditioned linear system can be solved using the following iterative method:

**Inner Solver**

```

Choose  $y_1 = u^k$ 
for  $\ell = 1$  to  $maxl$ 
  Solve  $(A^T A + \delta D)y_{\ell+1} = \delta D y_\ell + A^T f - \alpha \gamma a^{k+1} - \lambda B^T b^{k+1}$  for  $y_{\ell+1}$ 
end for
 $u^{k+1} = y_{\ell+1},$ 

```

where  $u^k$  refers to the restored image computed at the previous  $k$ th step, and the optimal parameter  $\delta > 0$  is chosen by numerical tries. Semi-convergence analysis for Inner Solver has been studied by Han and Yun [23]. Algorithm 2 is an iterative method that converges to a solution to TVL2 problem (4) as  $k \rightarrow \infty$ , so we do not have to get an accurate solution to the linear system in line 8 of Algorithm 2. For this reason, we have used  $maxl = 1$  for Inner Solver.

Notice that the linear system in Inner Solver is equivalent to solving the following least squares problem:

$$\min_u \left\| \left( \frac{1}{\sqrt{\delta}} D^{-\frac{1}{2}} (\delta D y_\ell - \alpha \gamma a^{k+1} - \lambda B^T b^{k+1}) \right) - \left( \frac{A}{\sqrt{\delta} D^{\frac{1}{2}}} \right) u \right\|_2^2. \quad (29)$$

Hence, the linear system in Inner Solver is solved by applying the CGLS to (29).

**4. Fixed-Point-like Method for TVL2 Problem (4)**

In this section, we propose a fixed-point-like method using CGLS for solving the new TVL2 problem (4) that can be obtained by modifying Algorithm 2. Notice that Algorithm 2



computes  $\hat{a}^{k+1}$  and  $a^{k+1}$  before the solution step of finding  $u^{k+1}$  (see lines 4 and 5 of Algorithm 2). However, the fixed-point-like method to be proposed in this section computes  $\hat{a}^{k+1}$  and  $a^{k+1}$  after the solution step of finding  $u^{k+1}$ . Below, we describe how to develop the fixed-point-like method in detail. We first split line 4 of Algorithm 2 into

$$\hat{a}^{k+1} = u^k + a^{k+\frac{1}{2}}, \quad (30)$$

where  $a^{k+\frac{1}{2}} = a^k - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}(u^k + a^k)$ . Replacing the *old* value  $u^k$  of Equation (30) with the *new* value  $u^{k+1}$ , one obtains the following equation:

$$\hat{a}^{k+1} = u^{k+1} + a^{k+\frac{1}{2}}. \quad (31)$$

Then, the solution step (i.e., line 8 of Algorithm 2) is changed to

$$A^T A u^{k+1} = A^T f - \alpha \gamma \hat{a}^{k+1} - \lambda B^T b^{k+1}, \quad (32)$$

where  $\hat{a}^{k+1}$  is computed using Equation (31) instead of using Equation (30). Substituting Equation (31) into Equation (32), one obtains

$$(A^T A + \alpha \gamma I) u^{k+1} = A^T f - \alpha \gamma a^{k+\frac{1}{2}} - \lambda B^T b^{k+1}. \quad (33)$$

After finding  $u^{k+1}$  from Equation (33), we compute  $\hat{a}^{k+1}$  using Equation (31) and  $a^{k+1} = \kappa a^k + (1 - \kappa) \hat{a}^{k+1}$ . By incorporating the above ideas into Algorithm 2, we can obtain a fixed-point-like method, called Algorithm 4, for solving TVL2 problem (4).

In addition, notice that the linear system in line 7 of Algorithm 3 is equivalent to solving the following least squares problem:

$$\min_u \left\| \begin{pmatrix} f \\ -\frac{1}{\sqrt{\alpha\gamma}}(\alpha\gamma a^{k+\frac{1}{2}} + \lambda B^T b^{k+1}) \end{pmatrix} - \begin{pmatrix} A \\ \sqrt{\alpha\gamma} I \end{pmatrix} u \right\|_2^2. \quad (34)$$

Hence, the linear system in line 7 of Algorithm 3 is solved using the CGLS instead of using the CG.

---

**Algorithm 3** Fixed-point-like method for TVL2 problem (4)

---

```

1: Given : observed image  $f$ , positive parameters  $\alpha, \beta, \gamma, \lambda$  and  $\kappa \in (0, 1)$ 
2: Initialization :  $a^0 = 0, b^0 = 0$  and  $u^0 = f$ 
3: for  $k=0$  to  $maxit$  do
4:    $a^{k+\frac{1}{2}} = a^k - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}(u^k + a^k)$ 
5:    $\hat{b}^{k+1} = \left( I - \text{prox}_{\frac{\beta}{\lambda}\varphi} \right) (Bu^k + b^k)$ 
6:    $b^{k+1} = \kappa b^k + (1 - \kappa) \hat{b}^{k+1}$ 
7:   Solve  $(A^T A + \alpha \gamma I) u^{k+1} = A^T f - \alpha \gamma a^{k+\frac{1}{2}} - \lambda B^T b^{k+1}$  for  $u^{k+1}$ 
8:    $\hat{a}^{k+1} = u^{k+1} + a^{k+\frac{1}{2}}$ 
9:    $a^{k+1} = \kappa a^k + (1 - \kappa) \hat{a}^{k+1}$ 
10:  if  $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$  then
11:    Stop
12:  end if
13: end for
```

---

## 5. Split Bregman Methods for TVL2 Problems (3) and (4)

In this section, we just provide the alternating split Bregman methods for solving the TVL2 problems (3) and (4) in order to evaluate the performance of Algorithms 2 and 3. For more details about the alternating split Bregman method as well as its convergence analysis, we refer to [5,8].

The following Algorithms 4 and 5 are the alternating split Bregman methods corresponding to the TVL2 problems (3) and (4), respectively.

As was done in line 7 of Algorithm 3, the linear systems in line 4 of Algorithms 4 and 5 are solved using the CGLS instead of using the CG.

---

**Algorithm 4** Split Bregman method for TVL2 problem (3)

---

```

1: Given : observed image  $f$ , positive parameters  $\alpha, \beta, \lambda$ 
2: Initialization :  $a^0 = 0, b^0 = 0$  and  $u^0 = f$ 
3: for  $k = 0$  to  $maxit$  do
4:   Solve  $(A^T A + \lambda B^T B + \alpha I)u^{k+1} = A^T f + \lambda B^T (a^k - b^k)$  for  $u^{k+1}$ 
5:    $a^{k+1} = \text{prox}_{\frac{\beta}{\lambda} \varphi}(Bu^{k+1} + b^k)$ 
6:    $b^{k+1} = b^k + Bu^{k+1} - a^{k+1}$ 
7:   if  $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$  then
8:     Stop
9:   end if
10: end for

```

---



---

**Algorithm 5** Split Bregman method for TVL2 problem (4)

---

```

1: Given : observed image  $f$ , positive parameters  $\alpha, \beta, \lambda, \gamma$ 
2: Initialization :  $a^0 = b^0 = 0, c^0 = d^0 = 0$  and  $u^0 = f$ 
3: for  $k = 0$  to  $maxit$  do
4:   Solve  $(A^T A + \lambda B^T B + \gamma I)u^{k+1} = A^T f + \lambda B^T (a^k - c^k) + \gamma (a^k - b^k)$  for  $u^{k+1}$ 
5:    $a^{k+1} = \text{prox}_{\frac{\alpha}{\gamma} \|\cdot\|_2}(u^{k+1} + b^k)$ 
6:    $d^{k+1} = \text{prox}_{\frac{\beta}{\lambda} \varphi}(Bu^{k+1} + c^k)$ 
7:    $b^{k+1} = b^k + u^{k+1} - a^{k+1}$ 
8:    $c^{k+1} = c^k + Bu^{k+1} - d^{k+1}$ 
9:   if  $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$  then
10:    Stop
11:   end if
12: end for

```

---

## 6. Numerical Experiments

In this section, we describe how to carry out numerical experiments for several test problems to evaluate the efficiency of two iterative methods, called Algorithms 2 and 3, using CGLS for solving the new proposed TVL2 problem (4). Performance of Algorithms 2 and 3 is evaluated by comparing their numerical results with those of the existing fixed-point method called Algorithm 1 and the split Bregman methods called Algorithms 4 and 5.

All numerical tests have been performed using Matlab R2016a on a personal computer equipped with Intel Core i5-3337 1.8 GHz CPU and 8 GB RAM. For numerical experiments, we have used three types of PSFs (point spread functions) which are Gaussian blur with standard deviation 9 and Average blur and Motion blur of size  $9 \times 9$ . PSF arrays  $P$  for Gaussian blur with standard deviation 9 and Average blur and Motion blur of size  $9 \times 9$  are generated by the built-in Matlab function `fspecial('Gaussian', [9,9], 9)`, `fspecial('average', 9)` and

$$P = \text{zeros}(9); P(4:6, :) = \text{fspecial}('motion', 9, 1),$$

respectively. The blurred and noisy image  $f$  is generated by

$$f = A \cdot \text{vec}(X) + \text{vec}(E),$$

where  $A$  stands for the blurring matrix that can be generated by the PSF array  $P$  according to the reflexive boundary condition, and  $E$  is the Gaussian white noise with mean 0 and standard deviation

3 that can be generated using Matlab function  $E = 3 \times \text{randn}(m, n)$ , where  $(m, n)$  denotes the size of true image.

In order to illustrate efficiency of the proposed algorithms, we have used four test images with an intensity range of  $[0, 255]$  such as Cameraman, Lena, House, and Boat with pixel size  $256 \times 256$ . To evaluate the quality of the restored images, we have used the peak signal-to-noise ratio (PSNR) between the original image and restored image, which is defined by

$$\text{PSNR} = 10 \log_{10} \left( \frac{\max_{i,j} |u_{ij}|^2 \cdot m \cdot n}{\|u - \tilde{u}\|_F^2} \right),$$

where  $\|\cdot\|_F$  represents the Frobenius norm,  $\tilde{u}$  denotes the restored image of the original image  $u$  with size  $m \times n$ , and  $u_{ij}$  stands for the value of the original image  $u$  at the pixel point  $(i, j)$ . In general, the larger PSNR stands for the better quality of the restored image.

For all numerical experiments, an initial image was set to the blurred and noisy image  $f$ ,  $\kappa = 1 \times 10^{-6}$ ,  $\text{maxit} = 150$ , and  $\text{tol}$  is set to  $5 \times 10^{-4}$  (for Algorithms 1 and 5),  $2 \times 10^{-4}$  (for Algorithm 4) or  $1 \times 10^{-3}$  (for Algorithms 2 and 3). For the CGLS method that is used to solve a linear system every iteration of Algorithms 1–5, the tolerance for stopping criterion is set to  $5 \times 10^{-2}$  (for Algorithms 1, 4 and 5) or  $1 \times 10^{-3}$  (for Algorithms 2 and 3), and the maximum number of iterations is set to 60.

## 7. Numerical Results

In this section, we provide numerical results for four test images that are listed in Tables 1–4 and Figure 1. In Tables 1–4, “Alg” represents the algorithm number to be used, “P<sub>0</sub>” represents the PSNR values for the blurred and noisy image  $f$ , “PSNR” represents the PSNR values for the restored image, “Iter” denotes the number of iterations required for Algorithms 1–5, the values in parentheses under the “Iter” column refer to the average number of iterations for CGLS, and “ $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\lambda$ ” and “ $\delta$ ” denote parameters that are chosen by numerical tries. Notice that, according to Theorem 1, the parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\lambda$  should be chosen appropriately for good performance of the fixed-point methods.

As can be seen in Tables 1–4, Algorithm 3 restores the true image better than Algorithms 1 and 2. This means that the fixed-point-like method for TVL2 problem (4) restores the true image better than the fixed-point methods for TVL2 problems (3) and (4). The linear system in Algorithm 3 that is obtained by computing  $a^{k+1}$  after the solution step of finding  $u^{k+1}$  is well-conditioned, while the linear system in Algorithm 2 is ill-conditioned. This is the reason why Algorithm 3 restores the true image significantly better than Algorithm 2. Since PSNR values of Algorithm 2 are about 0.3 to 1.0 smaller than those of Algorithm 3 for all test images, numerical results of Algorithm 2 are not provided for House and Boat images in Tables 3 and 4. Figure 1 shows the restored images by the fixed-point method, called Algorithm 1, for TVL2 problem (3) and the fixed-point-like method, called Algorithm 3, for TVL2 problem (4).

The fixed-point method (Algorithm 1) for the TVL2 model (3) performs worse than the corresponding split Bregman method (Algorithm 4), while the fixed-point-like method (Algorithm 3) for the TVL2 model (3) performs almost as well as the corresponding split Bregman method (Algorithm 5). Note that both the split Bregman methods for the TVL2 models (3) and (4) perform the same for almost all cases. When considering the number of iterations required for the CGLS, the total number of iterations for Algorithm 3 is less than that of Algorithm 1. Each iteration of Algorithms 1–3 requires one linear system solver CGLS and two matrix-times-vector operations that are the main time-consuming kernels, and each iteration of CGLS requires two matrix-times-vector operations. In addition, there are some additional vector-update operations that can be negligible as compared with matrix-times-vector operation. This means that the execution time of Algorithm 3 for the new TVL2 problem (4) is less than that of Algorithm 1 for TVL2 problem (3). For example, for the Cameraman image with Gaussian blur, the CPU times for Algorithms 1 and 3 are about 68 and 42 seconds, respectively.

**Table 1.** Numerical results for Cameraman image

Blur	P <sub>0</sub>	Alg	$\alpha$	$\beta$	$\gamma$	$\lambda$	$\delta$	tol	PSNR	Iter
Gaussian	20.85	1	0.0016	0.12		0.00041		$5 \times 10^{-4}$	25.17	67(47)
		2	1.15	0.135	0.05	0.01	4.2	$1 \times 10^{-3}$	24.60	34(9)
		3	1.15	0.135	0.05	0.01		$1 \times 10^{-3}$	25.51	132(8)
		4	0.00001	0.135		0.02		$2 \times 10^{-4}$	25.53	56(10)
		5	0.15	0.14	0.005	0.008		$5 \times 10^{-4}$	25.52	28(13)
Average	20.76	1	0.0016	0.13		0.00041		$5 \times 10^{-4}$	25.24	58(48)
		2	0.75	0.14	0.095	0.0095	4.2	$1 \times 10^{-3}$	24.64	33(9)
		3	0.75	0.14	0.095	0.0095		$1 \times 10^{-3}$	25.59	120(8)
		4	0.00001	0.145		0.02		$2 \times 10^{-4}$	25.60	42(14)
		5	0.01	0.14	0.002	0.009		$5 \times 10^{-4}$	25.61	27(15)
Motion	21.85	1	0.0017	0.22		0.00049		$5 \times 10^{-4}$	28.05	51(36)
		2	0.85	0.23	0.085	0.002	1.3	$1 \times 10^{-3}$	27.52	41(7)
		3	0.85	0.23	0.085	0.002		$1 \times 10^{-3}$	28.57	67(12)
		4	0.00001	0.24		0.03		$2 \times 10^{-4}$	28.51	36(7)
		5	0.01	0.24	0.03	0.003		$5 \times 10^{-4}$	28.59	37(10)

**Table 2.** Numerical results for Lena image

Blur	P <sub>0</sub>	Alg	$\alpha$	$\beta$	$\gamma$	$\lambda$	$\delta$	tol	PSNR	Iter
Gaussian	22.55	1	0.0016	0.17		0.0004		$5 \times 10^{-4}$	26.17	69(48)
		2	1.9	0.17	0.085	0.01	6.2	$1 \times 10^{-3}$	25.89	35(8)
		3	1.9	0.17	0.085	0.01		$1 \times 10^{-3}$	26.22	92(5)
		4	0.00001	0.17		0.09		$2 \times 10^{-4}$	26.29	66(5)
		5	0.01	0.175	0.00009	0.04		$5 \times 10^{-4}$	26.29	28(7)
Average	22.44	1	0.0016	0.17		0.0004		$5 \times 10^{-4}$	26.20	69(48)
		2	1.9	0.18	0.075	0.0095	6.2	$1 \times 10^{-3}$	25.89	36(8)
		3	1.9	0.18	0.075	0.0095		$1 \times 10^{-3}$	26.27	107(5)
		4	0.00001	0.20		0.09		$2 \times 10^{-4}$	26.33	63(5)
		5	0.01	0.180	0.00010	0.04		$5 \times 10^{-4}$	26.33	28(8)
Motion	23.06	1	0.0017	0.24		0.0005		$5 \times 10^{-4}$	28.28	57(37)
		2	1.1	0.26	0.075	0.0075	1.3	$1 \times 10^{-3}$	28.02	32(7)
		3	1.1	0.26	0.075	0.0075		$1 \times 10^{-3}$	28.45	46(11)
		4	0.00001	0.26		0.07		$2 \times 10^{-4}$	28.46	43(5)
		5	0.01	0.265	0.00010	0.025		$5 \times 10^{-4}$	28.46	18(8)

**Table 3.** Numerical results for House image

Blur	P <sub>0</sub>	Alg	$\alpha$	$\beta$	$\gamma$	$\lambda$	tol	PSNR	Iter
Gaussian	24.19	1	0.0020	0.20		0.00051	$5 \times 10^{-4}$	30.09	71(41)
		3	3.2	0.20	0.075	0.02	$1 \times 10^{-3}$	30.30	82(4)
		4	0.00001	0.20		0.13	$2 \times 10^{-4}$	30.52	56(5)
		5	0.01	0.210	0.0001	0.055	$5 \times 10^{-4}$	30.52	24(7)
Average	24.05	1	0.0020	0.20		0.00051	$5 \times 10^{-4}$	30.02	58(41)
		3	3.1	0.19	0.07	0.0095	$1 \times 10^{-3}$	30.24	117(4)
		4	0.00001	0.20		0.13	$2 \times 10^{-4}$	30.50	56(5)
		5	0.01	0.215	0.0001	0.0055	$5 \times 10^{-4}$	30.50	24(7)
Motion	27.01	1	0.0017	0.41		0.00051	$5 \times 10^{-4}$	32.87	68(31)
		3	1.5	0.55	0.045	0.01	$1 \times 10^{-3}$	33.32	82(12)
		4	0.00001	0.50		0.60	$2 \times 10^{-4}$	33.42	53(6)
		5	0.01	0.520	0.0020	0.100	$5 \times 10^{-4}$	33.52	18(6)

**Table 4.** Numerical results for Boat image

Blur	$P_0$	Alg	$\alpha$	$\beta$	$\gamma$	$\lambda$	$tol$	PSNR	Iter
Gaussian	21.28	1	0.0016	0.12		0.00041	$5 \times 10^{-4}$	25.13	60(49)
		3	2.5	0.13	0.07	0.009	$1 \times 10^{-3}$	25.17	58(6)
		4	0.00001	0.14		0.11	$2 \times 10^{-4}$	25.29	74(5)
		5	0.10	0.12	0.0003	0.075	$5 \times 10^{-4}$	25.31	36(6)
Average	21.19	1	0.0016	0.12		0.00041	$5 \times 10^{-4}$	25.19	54(50)
		3	2.4	0.135	0.075	0.008	$1 \times 10^{-3}$	25.24	62(5)
		4	0.00001	0.13		0.18	$2 \times 10^{-4}$	25.38	85(5)
		5	0.01	0.130	0.0001	0.070	$5 \times 10^{-4}$	25.38	35(7)
Motion	23.32	1	0.0018	0.24		0.00052	$5 \times 10^{-4}$	27.70	54(36)
		3	0.6	0.265	0.09	0.02	$1 \times 10^{-3}$	27.87	34(14)
		4	0.00001	0.26		0.12	$2 \times 10^{-4}$	27.91	52(5)
		5	0.01	0.255	0.0001	0.050	$5 \times 10^{-4}$	27.91	22(4)

**Figure 1.** Restored images by Algorithms 1 and 3 (The first row images are Cameramen images for Gaussian blur, the second row images are Lena images for Motion blur, the third row images are House images for Motion blur, and the fourth row images are Boat images for Average blur).

## 8. Conclusions

In this paper, we first proposed a new TVL2 regularization model (4) for image restoration, and then we proposed the fixed-point method (Algorithm 2) and the fixed-point-like method (Algorithm 3) for solving TVL2 problem (4). According to numerical experiments, the fixed-point-like method for the new TVL2 problem (4) restores true image better than the fixed-point method for TVL2 problem (4). The reason for this is that the linear system in line 7 of Algorithm 3 is well-conditioned and  $a^{k+1}$  is computed after the solution step of finding  $u^{k+1}$ , i.e.,  $a^{k+1}$  is computed using the new value of  $u^{k+1}$  instead of using the old value of  $u^k$ .

Both of the split Bregman methods for TVL2 problems (3) and (4) perform the same for almost all cases, while the fixed-point-like method for TVL2 problem (4) performs better than the fixed-point method for TVL2 problem (3). It can be also seen that the execution time of the fixed-point-like method for TVL2 problem (4) is less than that of the fixed-point method for TVL2 problem (3). Hence, it can be concluded that the new proposed TVL2 model (4) for image restoration is preferred over the TVL2 model (3), and the proposed fixed-point-like method (Algorithm 3) is well suited for the new TVL2 model (4).

The fixed-point-like method and TVL2 model (4) proposed in this paper can be applied to the image inpainting problem or image restoration problem with Poisson noise. Future work will study these kinds of problems.

**Author Contributions:** Conceptualization, J.H.Y.; Formal analysis, J.H.Y.; Methodology, K.S.K. and J.H.Y.; Software, K.S.K. and J.H.Y.; Writing—original draft, K.S.K. and J.H.Y.; Writing—review and editing, K.S.K. and J.H.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1A6A3A01012976), and the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. 2019R1F1A1060718).

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which greatly improved the quality of the paper.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Chambolle, A.; Lions, P.L. Image recovery via total variation minimization and related problems. *Numer. Math.* **1997**, *76*, 167–188. [\[CrossRef\]](#)
2. Chan, T.F.; Shen, J. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005.
3. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D* **1992**, *60*, 259–268. [\[CrossRef\]](#)
4. Beck, A.; Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **2009**, *18*, 2419–2434. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Cai, J.F.; Osher, S.; Shen, Z. Split Bregman methods and frame based image restoration. *Multiscale Model. Simul.* **2009**, *8*, 337–369. [\[CrossRef\]](#)
6. Chan, T.F.; Mulet, P. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM J. Numer. Anal.* **1999**, *36*, 354–367. [\[CrossRef\]](#)
7. Combettes, P.L.; Wajs, V.R. Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Model. Simul.* **2005**, *4*, 1168–1200. [\[CrossRef\]](#)
8. Goldstein, T.; Osher, S. The Split Bregman Method for L1 Regularized Problems. *SIAM J. Imaging Sci.* **2008**, *2*, 323–343. [\[CrossRef\]](#)
9. Goldfarb, D.; Yin, W. Second-order cone programming methods for total variation-based image restoration. *SIAM J. Sci. Comput.* **2005**, *27*, 622–645. [\[CrossRef\]](#)
10. Ng, M.K.; Qi, L.; Yang, Y.F.; Huang, Y.M. On Semismooth Newton's Methods for Total Variation Minimization. *J. Math. Imaging Vis.* **2007**, *27*, 265–276. [\[CrossRef\]](#)



11. Vogel, C.R.; Oman, M.E. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.* **1996**, *17*, 227–238. [[CrossRef](#)]
12. Vogel, C.R.; Oman, M.E. First, robust total variation-based reconstruction of noisy, blurring images. *IEEE Trans. Image Process.* **1998**, *7*, 813–824. [[CrossRef](#)] [[PubMed](#)]
13. Yun, J.H. Performance of relaxed iterative methods for image deblurring problems. *J. Algorithms Comput. Technol.* **2019**, *13*, 16. [[CrossRef](#)]
14. Chen, D.Q.; Zhang, H.; Cheng, L.Z. A fast fixed point algorithm for total variation deblurring and segmentation. *J. Math. Imaging Vis.* **2012**, *43*, 167–179. [[CrossRef](#)]
15. Bjorck, A. *Numerical Methods for Least Squares Problems*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1996.
16. Moreau, J.J. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France* **1965**, *93*, 273–299. [[CrossRef](#)]
17. Micchelli, C.A.; Shen, L.; Xu, Y. Proximity algorithms for image models: denoising. *Inverse Probl.* **2011**, *27*, 30. [[CrossRef](#)]
18. Lu, J.; Qiao, K.; Shen, L.; Zou, Y. Fixed-point algorithms for a TVL1 image restoration model. *Int. J. Comput. Math.* **2018**, *95*, 1829–1844. [[CrossRef](#)]
19. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **1952**, *49*, 409–436. [[CrossRef](#)]
20. Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2003.
21. Axelsson, O. *Iterative Solution Methods*; Cambridge University Press: Cambridge, UK, 1994.
22. Opial, Z. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull. Amer. Math. Soc.* **1967**, *73*, 591–597. [[CrossRef](#)]
23. Han, Y.D.; Yun, J.H. Performance of the Restarted Homotopy Perturbation Method and Split Bregman Method for Multiplicative Noise Removal. *Math. Probl. Eng.* **2018**, *2018*, 21. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).