International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

Image Restoration Methods for a New TVL2 Regularization Model*

Hyo Jin Lim[1], Kyoum Sun Kim[1], and Jae Heon Yun†[2]

[1]*Department of Mathematics, Chungbuk National University, Cheongju, Korea 28644.*
[2]*Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju, Korea 28644.*
ORCID: 0000-0001-6841-8137

Abstract
In this paper, we first propose a new TVL2 regularization model for image restoration, and then we propose a fixed-point-like method and a split Bregman method for solving the new image restoration model. We next provide convergence analysis for both the fixed-point-like method and the split Bregman method. Finally, we provide numerical experiments for several test problems in order to evaluate the effectiveness of two iterative methods for the new proposed image restoration model.

Keywords: Total variation, fixed-point-like method, image restoration, proximal operator, split Bregman method.

1991 Mathematics Subject Classification. 94A08, 54E05, 49Q20, 35M85.

1. INTRODUCTION

Image restoration which recovers a true image from a blurry and noisy image is one of the most challenging tasks in image processing. Let us assume that the original image u has an $N \times N$ array. For convenience, the image u is represented by a long vector u of size $m = N^2$. In this paper, we consider the problem of finding the unknown original image $u \in \mathbb{R}^m$ from an observed degraded image $f \in \mathbb{R}^m$ which is defined by

$$f = Au + \eta, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times m}$ is a blurring matrix and $\eta \in \mathbb{R}^m$ is a Gaussian noise. The amplitude of the Gaussian noise distribution can be expressed as

$$\eta(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

where σ and μ represent standard derivation and mean value of the noise distribution η, respectively. More specifically, our goal is to approximate the original image u as well as possible from an observed image f degraded by a Gaussian noise.

The model based on the total variation (TV) has made a lot of impact in image processing despite its weaknesses. In last decades, many TV-based models have been developed in the literature. One of the most popular model is the Rudin-Osher-Fatemi (ROF) model [15]. This model produces a deblurred image given by the following minimization problem:

$$\min_{u\in R^m} \left\{ \frac{1}{2}\|Au - f\|_2^2 + \beta TV(u) \right\}, \tag{1.2}$$

where $u \in \mathbb{R}^m$ and $f \in \mathbb{R}^m$ represent the original and observed images respectively, TV(u) stands for a discrete total variation of u, $\beta > 0$ is a regularization parameter, $A \in \mathbb{R}^{m\times m}$ is a blurring matrix, and $\|\cdot\|_2$ denotes the L_2-norm. The first term of $\|Au - f\|_2^2$ is called the data-fitting term, and the second term ***TV(u)*** is called the regularization (or penalty) term. In the last few decades, many approaches have been proposed to approximate the true image u from the ROF model (1.2). For example, the split Bregman method [3, 8], the lagged diffusivity fixed-point method [4], proximal forward-backward splitting method [6], proximity method [11], Newton-like method [13], and time-marching PDE method [15] have been proposed by many researchers.

In 2012, Chen et al. [5] proposed a fixed-point method for solving the following TVL2 regularization model

$$\min_{u\in R^m} \left\{ \frac{1}{2}\|Au - f\|_2^2 + \frac{\alpha}{2}\|u\|_2^2 + \beta TV(u) \right\}, \tag{1.3}$$

where α and β are positive regularization parameters. Recently, Kim and Yun [10] proposed a fixed-point-like method for solving the following TVL2 regularization model

$$\min_{u\in R^m} \left\{ \frac{1}{2}\|Au - f\|_2^2 + \alpha\|u\|_2 + \beta TV(u) \right\}, \tag{1.4}$$

where α and β are positive regularization parameters. It was shown in [10] that the fixed-point-like method for the TVL2 model (1.4) performs better than the fixed-point method for the TVL2 model (1.3). This observation gives us an idea of proposing a new TVL2 regularization model

$$\min_{u\in R^m} \left\{ \frac{1}{2}\|Au - f\|_2^2 + \alpha\|Du\|_p + \beta TV(u) \right\} \ (p = 1 \text{ or } 2), \tag{1.5}$$

where α and β are positive regularization parameters, $D = -\Delta$ and Δ denotes a discrete Laplacian operator.

*This work was supported by the National Research Foundation of Korea(NRF) funded by the Korea government(MSIT) (No. 2019R1F1A1060718)
†Corresponding author.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

There are two cases of $TV(u)$ (discrete total variation of u) in the literature. One case is the *anisotropic total variation* defined by

$$TV(u) = \sum_{i=1}^{m} |(\nabla u)_i^x)| + |(\nabla u)_i^y)|\,,$$

and the other case is the *isotropic total variation* defined by

$$TV(u) = \sum_{i=1}^{m} \sqrt{|(\bigtriangledown u)_i^x|^2 + |(\bigtriangledown u)_i^y|^2} = \sum_{i=1}^{m} \| \ (\nabla u)_i \ \|_2\,,$$

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

$$(\nabla u)_i = \begin{pmatrix} (\nabla u)_i^x \\ (\nabla u)_i^y \end{pmatrix}, \quad i = 1, 2, ..., m$$

with

$$(\nabla u)_i^x = \begin{cases} 0 & \text{if } i \bmod N = 1, \\ u_i - u_{i-1} & \text{if } i \bmod N \neq 1, \end{cases}$$

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

In this paper, we only consider the isotropic total variation of $u \in \mathbb{R}^m$, and we assume that the reflexive boundary condition is used for blurred images.

The purpose of this paper is to propose two iterative methods which are a fixed-point-like method and a split Bregman method for solving the new proposed TVL2 regularization model (1.5). This paper is organized as follows. In Section 2, we provide some definitions and important properties that are used in this paper. In Section 3, we simply provide the fixed-point-like method and the split Bregman method for solving the TVL2 problem (1.4) proposed by Kim and Yun [10] for the purpose of comparison with those methods to be proposed in this paper. In Section 4, we first propose a fixed-point-like method for solving the new TVL2 problem (1.5), and then we provide convergence analysis for the fixed-point-like method. In Section 5, we first provide a split Bregman method for solving the new TVL2 problem (1.5), and then we study convergence of the split Bregman method. In Section 6, we provide numerical experiments for several test problems in order to evaluate the effectiveness of the proposed two iterative methods for the TVL2 problem (1.5). This can be done by comparing their performances with the corresponding two iterative methods for the TVL2 problem (1.4). Lastly, some conclusion are drawn.

2. PRELIMINARIES

In this section, we will introduce several definitions and notations as well as some useful properties that are used in this paper.

Definition 2.1 ([12])**.** Let ψ: $\mathbb{R}^m \to (-\infty, \infty]$ be a proper, convex and lower semi-continuous function. The *proximal operator* of ψ at $x \in \mathbb{R}^m$ is defined by

$$\text{prox}_{\psi}(x) = \underset{u}{\text{argmin}} \left\{ \frac{1}{2}\|u - x\|_2^2 + \psi(u) : u \in R^m \right\}. \quad (2.1)$$

Definition 2.2. Let ψ: $\mathbb{R}^m \to (-\infty, \infty]$ be a proper function, and let $dom(\psi)$ denote the *domain* of ψ, that is, $dom(\psi) = \{x \in \mathbb{R}^m : \psi(x) < \infty\}$. For an $x \in dom(\psi)$, the *subdifferential* of ψ at $x \in \mathbb{R}^m$ is defined by

$$\partial\psi(x) = \{y \in \mathbb{R}^m : \psi(z) \geq \psi(x) + \langle y, z - x\rangle, \forall z \in \mathbb{R}^m\}. \quad (2.2)$$

For a nonlinear operator $H : \mathbb{R}^m \to \mathbb{R}^m$, H is called *non-expansive* if for any $x,\ y \in \mathbb{R}^m$

$$\| H(x) - H(y) \|_2 \ \leq \ \| x - y \|_2, \quad (2.3)$$

and H is called *firmly non-expansive* if for any $x,\ y \in \mathbb{R}^m$

$$\| H(x) - H(y) \|_2^2 \ \leq \ \langle x - y, H(x) - H(y)\rangle. \quad (2.4)$$

By the application of the Cauchy-Schwarz inequality, it is easy to show that a *firmly non-expansive* operator is also *non-expansive*.

Let $S : \mathbb{R}^m \to \mathbb{R}^m$ be an operator and $\kappa \in (0, 1)$. Then the *Picard iteration* of S is defined by

$$x^{k+1} = Sx^k \quad \text{for} \quad k = 0,\ 1,\ 2, \ldots \quad (2.5)$$

for a given vector $x^0 \in \mathbb{R}^m$, and the κ-averaged operator S_k of S is defined by

$$S_k = \kappa I + (1 - \kappa)S \quad (2.6)$$

where *I* denotes an identity operator.

Proposition 2.3 ([2])**.** *Let $\psi : \mathbb{R}^m \to (-\infty, \infty]$ be a proper convex function. Then*

$$x^* \in \text{argmin}\Big\{\psi(x) : x \in \mathbb{R}^m\Big\} \quad \textit{if and only if} \quad 0 \in \partial\psi(x^*).$$

The following result illustrates the relationship between the proximal operator and the subdifferential of a convex function which is a basic tool of developing iterative algorithms for the regularization models (1.3) to (1.5).

Proposition 2.4 ([11, 15])**.** *Let $\psi : \mathbb{R}^m \to \mathbb{R}$ be a convex function. Then for $x,\ y \in \mathbb{R}^m$, the following holds*

$$y \in \partial\psi(x) \ \Leftrightarrow \ x = \text{prox}_{\psi}(x + y). \quad (2.7)$$

Let $D \in \mathbb{R}^{m\times m}$ be a finite difference matrix corresponding to the negative Laplacian $-(u_{xx} + u_{yy})$ of the image $u \in \mathbb{R}^m$, where u_{xx} and u_{yy} denote the second order partial derivatives in the vertical direction and the horizontal direction, respectively. Then the matrix $D \in \mathbb{R}^{m\times m}$ can be expressed as

$$D = (I_N \otimes D_1 + D_1 \otimes I_N) \in \mathbb{R}^{m\times m},$$

where I_N is the identity matrix of order N, $\otimes$ denotes the Kronecker product, and D_1 is an $m \times m$ singular matrix obtained by finite difference approximations to the second

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

order partial derivatives $-u_{xx}$ or $-u_{yy}$ [9]. That is, D_1 is given by

$$D_1 = \begin{pmatrix} 1 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & \dots & \dots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & 0 \\ 0 & \dots & \dots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}.$$

Let B be a $2m \times m$ matrix that represents a discrete gradient operator ∇ with $m = N^2$. Then, the matrix B can be expressed as

$$B = \begin{pmatrix} I_N \otimes D_2 \\ D_2 \otimes I_N \end{pmatrix} \in \mathbb{R}^{2m \times m}, \tag{2.8}$$

where D_2 is the first order finite difference singular matrix of order N

$$D_2 = \begin{pmatrix} 0 & 0 & 0 & \dots & \dots & 0 \\ -1 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 1 & \dots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & 0 \\ 0 & \dots & \dots & -1 & 1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}.$$

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

$$\varphi(d) = \sum_{i=1}^{m} \|(d_i, d_{m+i})^T\|_2 \text{ for each } d = (d_i) \in \mathbb{R}^{2m}, \tag{2.9}$$

where d_i denotes the ith component of the vector d. Then the isotropic TV of $u \in \mathbb{R}^m$ can be expressed as

$$TV(u) = (\varphi \circ B)(u) = \varphi(Bu). \tag{2.10}$$

3. IMAGE RESTORATION METHODS FOR THE TVL2 PROBLEM (1.4)

In this section, we just provide the fixed-point-like method and split Bregman method proposed in [10] for solving the TVL2 problem (1.4) for the purpose of comparison with those methods for solving the new proposed TVL2 problem (1.5). The fixed-point-like method, called Algorithm 1, and the split Bregman method, called Algorithm 2, for the TVL2 problem (1.4) are described below (see [10] for detailed description of algorithms).

Algorithm 1 Fixed-point-like method for the TVL2 problem (1.4)

1: Given : observed image f, positive parameters α, β, γ, λ 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: $\quad a^{k+\frac{1}{2}} = a^k - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_2}\left(u^k + a^k\right)$
5: $\quad \hat{b}^{k+1} = \left(I - \text{prox}_{\frac{\beta}{\lambda}\varphi}\right)\left(Bu^k + b^k\right)$
6: $\quad b^{k+1} = \kappa b^k + (1-\kappa)\hat{b}^{k+1}$
7: $\quad$ Solve $(A^TA + \alpha\gamma I)u^{k+1} = A^Tf - \alpha\gamma a^{k+\frac{1}{2}} - \lambda B^T b^{k+1}$ for u^{k+1}
8: $\quad \hat{a}^{k+1} = u^{k+1} + a^{k+\frac{1}{2}}$
9: $\quad a^{k+1} = \kappa a^k + (1-\kappa)\hat{a}^{k+1}$
10: $\quad$ **if** $\frac{\|u^{k+1}-u^k\|_2}{\|u^{k+1}\|_2} < tol$ **then**
11: $\quad\quad$ Stop
12: $\quad$ **end if**
13: **end for**

Algorithm 2 Split Bregman method for the TVL2 problem (1.4)

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

In this paper, the linear system in line 7 of Algorithm 1 and the linear system in line 4 of Algorithm 2 are approximately solved using the CGLS (Conjugate Gradient Least Squares method [1]) with a tolerance value instead of using the CG (Conjugate Gradient method [7]).

4. FIXED-POINT-LIKE METHOD FOR THE TVL2 PROBLEM (1.5)

In this section, we first propose a fixed-point-like method for solving the new TVL2 problem (1.5), and then we provide a convergence analysis for the fixed-point-like method. Using

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

(2.10), the TVL2 problem (1.5) can be expressed as

$$\min_{u\in\mathbb{R}^m}\left\{\frac{1}{2}\|Au-f\|_2^2+\alpha\|Du\|_p+\beta(\varphi\circ B)(u)\right\}\ (p=1 \text{ or } 2). \tag{4.1}$$

Using Propositions 2.3 and 2.4, we can obtain the following property for a solution to the TVL2 problem (4.1).

Theorem 4.1. *If $u\in\mathbb{R}^m$ is a solution of the problem* (4.1)*, then for any $\gamma,\lambda>0$, there exist vectors $a\in\mathbb{R}^m$ and $b\in\mathbb{R}^{2m}$ such that*

$$a=(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p})(Du+a), \tag{4.2}$$

$$b=(I-\operatorname{prox}_{\frac{\beta}{\lambda}\varphi})(Bu+b), \tag{4.3}$$

$$(A^TA)u=A^Tf-\alpha\gamma D^Ta-\lambda B^Tb. \tag{4.4}$$

Conversely, if there exist $\gamma,\lambda>0$, $a\in\mathbb{R}^m$, $b\in\mathbb{R}^{2m}$ and $u\in\mathbb{R}^m$ satisfying (4.2)-(4.4)*, then u is a solution of the problem* (4.1)*.*

Proof. Suppose that $u\in\mathbb{R}^m$ is a solution of the problem (4.1). Using Proposition 2.3 and the chain rule of subdifferential, we can obtain

$$\begin{aligned}0&\in A^T(Au-f)+\alpha D^T(\partial\|\cdot\|_p)(Du)+\beta B^T\circ(\partial\varphi)\circ(Bu)\\&=A^T(Au-f)+\alpha D^T(\partial\|\cdot\|_p)(Du)+B^T\partial(\beta\varphi)(Bu).\end{aligned} \tag{4.5}$$

From relation (4.5), for any $\gamma,\lambda>0$ we can choose two vectors $a\in\mathbb{R}^m$ and $b\in\mathbb{R}^{2m}$ such that

$$a\in\partial(\frac{1}{\gamma}\|\cdot\|_p)(Du),\quad b\in\partial(\frac{\beta}{\lambda}\varphi)(Bu) \tag{4.6}$$

and

$$0=A^T(Au-f)+\alpha\gamma D^Ta+\lambda B^Tb. \tag{4.7}$$

Using Proposition 2.4, the inclusions (4.6) reduce to

$$Du=\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p}(Du+a)\ \text{ and }\ Bu=\operatorname{prox}_{\frac{\beta}{\lambda}\varphi}(Bu+b). \tag{4.8}$$

From (4.7) and (4.8), one obtains (4.2), (4.3) and (4.4).

Conversely, assume that (4.2)-(4.4) are satisfied for some $\gamma,\lambda>0$, $a\in\mathbb{R}^m$, $b\in\mathbb{R}^{2m}$ and $u\in\mathbb{R}^m$. From (4.2) and (4.3), we have

$$\begin{aligned}a\in\partial(\frac{1}{\gamma}\|\cdot\|_p)(Du)=\frac{1}{\gamma}(\partial\|\cdot\|_p)(Du)\\ \text{and } b\in\partial(\frac{\beta}{\lambda}\varphi)(Bu)=\frac{\beta}{\lambda}(\partial\varphi)(Bu).\end{aligned} \tag{4.9}$$

Using (4.4) and (4.9), we can obtain

$$0\in A^T(Au-f)+\alpha D^T(\partial\|\cdot\|_p)(Du)+\beta B^T\partial\varphi(Bu).$$

Consequently (4.5) holds. Hence, u is a solution of the problem (4.1). □

From (4.2)-(4.4) in Theorem 4.1, we can develop a fixed-point algorithm which converges to a solution to the TVL2 problem (1.5). We now describe how to develop the fixed-point algorithm. Let u be an approximate solution to the ill-condition linear system (4.4) in Theorem 4.1. Then u can be expressed as

$$u=M(A^Tf-\alpha\gamma D^Ta-\lambda B^Tb), \tag{4.10}$$

where M is a symmetric positive semi-definite matrix approximating a solution of the linear system (4.4). For example, we can choose $M=(A^TA)_r^\dagger$, which is a truncated psedoinverse of A^TA using the r largest positive singular values of A^TA. Substituting (4.10) into (4.2) and (4.3), we can obtain

$$\begin{aligned}a=(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p})((I_m-\alpha\gamma MDD^T)a\\ -\lambda MB^Tb+MDA^Tf),\end{aligned} \tag{4.11}$$

$$\begin{aligned}b=(I-\operatorname{prox}_{\frac{\beta}{\lambda}\varphi})(-\alpha\gamma MD^TBa\\ +(I_{2m}-\lambda BMB^T)b+\lambda BMA^Tf).\end{aligned} \tag{4.12}$$

(4.11) and (4.12) can be rewritten as a unified fixed-point equation

$$\begin{aligned}\begin{pmatrix}a\\b\end{pmatrix}&=\begin{pmatrix}I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p}&0\\0&I-\operatorname{prox}_{\frac{\beta}{\lambda}\varphi}\end{pmatrix}\\&\times\left[\begin{pmatrix}I_m-\alpha\gamma MDD^T&-\lambda DMB^T\\-\alpha\gamma MD^TB&I_{2m}-\lambda BMB^T\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix}\right.\\&\left.+\begin{pmatrix}MDA^Tf\\MBA^Tf\end{pmatrix}\right].\end{aligned} \tag{4.13}$$

We now formulate some nonlinear operators on $\mathbb{R}^{3m}$. Let us define an operator $\mathcal{P}:\mathbb{R}^{3m}\to\mathbb{R}^{3m}$ at a vector $\binom{x}{y}\in\mathbb{R}^{3m}$ with $x\in\mathbb{R}^m$ and $y\in\mathbb{R}^{2m}$ by

$$\begin{aligned}\mathcal{P}\begin{pmatrix}x\\y\end{pmatrix}&=\begin{pmatrix}I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p}&0\\0&I-\operatorname{prox}_{\frac{\beta}{\lambda}\varphi}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}\\&=\begin{pmatrix}(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_p})(x)\\(I-\operatorname{prox}_{\frac{\beta}{\lambda}\varphi})(y)\end{pmatrix},\end{aligned} \tag{4.14}$$

an affine transformation $\mathcal{Q}:\mathbb{R}^{3m}\to\mathbb{R}^{3m}$ at a vector $\binom{a}{b}\in\mathbb{R}^{3m}$ with $a\in\mathbb{R}^m$ and $b\in\mathbb{R}^{2m}$ by

$$\begin{aligned}\mathcal{Q}\begin{pmatrix}a\\b\end{pmatrix}&=\begin{pmatrix}I_m-\alpha\gamma MDD^T&-\lambda DMB^T\\-\alpha\gamma MD^TB&I_{2m}-\lambda BMB^T\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix}\\&+\begin{pmatrix}MDA^Tf\\MBA^Tf\end{pmatrix},\end{aligned} \tag{4.15}$$

and an operator $\mathcal{G}:\mathbb{R}^{3m}\to\mathbb{R}^{3m}$ by

$$\mathcal{G}=\mathcal{P}\circ\mathcal{Q}. \tag{4.16}$$

Let $\omega=(a,b)^T$. Then (4.13) can be expressed as

$$\omega=\mathcal{G}\omega. \tag{4.17}$$

That is, ω is a fixed point of the operator $\mathcal{G}$.
Since the TVL2 problem (4.1) has been transformed to the fixed point problem of (4.17), some useful results in fixed point theory are studied below in order to develop a convergent algorithm for solving the problem (4.1).

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

Proposition 4.2. *The operator $\mathcal{G}$ defined by* (4.16) *has a fixed point.*

Proof. Since a solution u of the TVL2 problem (4.1) exists, from (4.17) and the first part of the proof of Theorem 4.1 $\mathcal{G}$ has a fixed point. □

Let $c = \begin{pmatrix} MDA^T f \\ MBA^T f \end{pmatrix} \in \mathbb{R}^{3m}$, $P = \begin{pmatrix} D \\ B \end{pmatrix}$, and $R = \begin{pmatrix} \alpha\gamma I_m & 0 \\ 0 & \lambda I_{2m} \end{pmatrix}$. Then (4) can be expressed as

$$\mathcal{Q}w = (I_{3m} - PMP^T R)w + c, \tag{4.18}$$

where $\omega = (a, b)^T$

Theorem 4.3. *Let M, P and R be defined as above. If $\|I_{3m} - PMP^T R\|_2 \leq 1$, then the operator $\mathcal{G}$ defined by* (4.16) *is non-expansive.*

Proof. Since $I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_p}$ and $I - \text{prox}_{\frac{\beta}{\lambda}\varphi}$ are firmly non-expansive [6], the operator $\mathcal{P}$ defined by (4) is non-expansive [10]. For all $v_1, v_2 \in \mathbb{R}^{3m}$, we have

$$\begin{aligned} \|\mathcal{G}(v_1) - \mathcal{G}(v_2)\|_2 &= \|\mathcal{P}(\mathcal{Q}(v_1)) - \mathcal{P}(\mathcal{Q}(v_2))\|_2 \\ &\leq \|\mathcal{Q}(v_1) - \mathcal{Q}(v_2)\|_2. \end{aligned}$$

From (4.18) and the assumption $\|I_{3m} - PMP^T R\|_2 \leq 1$, we have

$$\begin{aligned} \|\mathcal{Q}(v_1) - \mathcal{Q}(v_2)\|_2 &= \|(I_{3m} - PMP^T R)(v_1 - v_2)\|_2 \\ &\leq \|(I_{3m} - PMP^T R)\|_2 \cdot \|v_1 - v_2\|_2 \\ &\leq \|v_1 - v_2\|_2. \end{aligned}$$

Hence, the operator $\mathcal{G}$ is non-expansive. □

The following proposition provides a convergence result for the Picard iteration of a κ-averaged operator.

Proposition 4.4 ([14])**.** *Let $C \subset \mathbb{R}^{3m}$ be a closed convex set and $S : C \to C$ be a non-expansive mapping with at least one fixed point. Then for any $\omega^0 \in C$ and any $\kappa \in (0, 1)$, the Picard iteration of S_κ converges to a fixed point of S.*

Theorem 4.5. *Let M, P, R and $\mathcal{G}$ be defined as in Theorem 4.3. If $\|I_{3m} - PMP^T R\|_2 \leq 1$, then for any $\kappa \in (0, 1)$ the Picard iteration of $\mathcal{G}_\kappa$ converges to a fixed point of $\mathcal{G}$.*

Proof. From Proposition 4.2 and Theorem 4.3, the operator $\mathcal{G}$ has a fixed point and $\mathcal{G}$ is non-expansive. Hence, Proposition 4.4 ensures that for any vector $w^0 \in \mathbb{R}^{3m}$ and $\kappa \in (0, 1)$, the Picard iteration of $\mathcal{G}_k$ converges to a fixed point of $\mathcal{G}$. □

Lemma 4.6. *Let M, P, R and $\mathcal{G}$ be defined as in Theorem 4.3. If we choose positive parameters α, γ and λ such that*

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

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

Proof. The proof of this lemma is skipped since it can be done in a similar way to the proof of Lemma 2 in [10]. □

Theorem 4.7. *If the assumptions of Lemma 4.6 hold, then for any $\kappa \in (0, 1)$ the Picard iteration of $\mathcal{G}_\kappa$ converges to a fixed point of $\mathcal{G}$.*

Proof. From Lemma 4.6 and Theorem 4.5, this theorem clearly hold. □

From Theorem 4.5 and the Picard iteration of the κ-averaged operator $\mathcal{G}_\kappa = \kappa I + (1-\kappa)\mathcal{G}$ of $\mathcal{G}$, we can obtain a fixed-point method, called Algorithm 3, which converges to a solution to the TVL2 problem (1.5) for some appropriately chosen positive parameters α, γ, λ.

Algorithm 3 Fixed-point algorithm for the TVL2 problem (1.5)

1: Given : observed image f, positive parameters α, β, γ, λ 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: $\quad \hat{a}^{k+1} = \left(I - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_p}\right)\left(Du^k + a^k\right)$
5: $\quad a^{k+1} = \kappa a^k + (1-k)\hat{a}^{k+1}$
6: $\quad \hat{b}^{k+1} = \left(I - \text{prox}_{\frac{\beta}{\lambda}\varphi}\right)\left(Bu^k + b^k\right)$
7: $\quad b^{k+1} = \kappa b^k + (1-k)\hat{b}^{k+1}$
8: $\quad$ Solve $A^T A u^{k+1} = A^T f - \alpha\gamma D^T a^{k+1} - \lambda B^T b^{k+1}$ for u^{k+1}
9: $\quad$ **if** $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$ **then**
10: $\quad\quad$ Stop
11: $\quad$ **end if**
12: **end for**

The linear system in line 8 of Algorithm 3 is highly 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

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

where $M = (A^T A)_r^\dagger$. However, computation of $(A^T A)_r^\dagger$ is very time-consuming when A is large, and choosing an optimal number of r is also difficult. So we need to develop more efficient method than Algorithm 3. For this reason, we propose a fixed-point-like method for solving the new TVL2L1D problem (1.5) which can be obtained by modifying Algorithm 3. Notice that Algorithm 3 computes $\hat{a}^{k+1}$, a^{k+1}, $\hat{b}^{k+1}$ and b^{k+1} before the solution step of finding u^{k+1}. However, the fixed-point-like method to be proposed in this section computes $\hat{a}^{k+1}$, a^{k+1}, $\hat{b}^{k+1}$ and b^{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 3 into

$$\hat{a}^{k+1} = Du^k + a^{k+\frac{1}{2}}. \tag{4.19}$$

where $a^{k+\frac{1}{2}} = a^k - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_p}(Du^k + a^k)$. Next, split line 6 of Algorithm 3 into

$$\hat{b}^{k+1} = Bu^k + b^{k+\frac{1}{2}}. \tag{4.20}$$

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

where $b^{k+\frac{1}{2}} = b^k - \text{prox}_{\frac{\beta}{\lambda}\varphi}(Bu^k + b^k)$. Replacing the old value u^k of (4.19) and (4.20) with the new updated value u^{k+1}, one can obtain

$$\tilde{a}^{k+1} = Du^{k+1} + a^{k+\frac{1}{2}}, \tag{4.21}$$

and

$$\tilde{b}^{k+1} = Bu^{k+1} + b^{k+\frac{1}{2}}. \tag{4.22}$$

Then the solution step (i.e line 8 of Algorithm 3) is changed to

$$A^T Au^{k+1} = A^T f - \alpha\gamma D^T \tilde{a}^{k+1} - \lambda B^T \tilde{b}^{k+1}. \tag{4.23}$$

Note that $\tilde{a}^{k+1}$ is computed using (4.21) instead of using (4.19), and $\tilde{b}^{k+1}$ is computed using (4.22) instead of using (4.20).

Substituting (4.21) and (4.22) into (4.23), one obtains

$$(A^T A + \alpha\gamma D^T D + \lambda B^T B)u^{k+1} = A^T f - \alpha\gamma D^T a^{k+\frac{1}{2}} - \lambda B^T b^{k+\frac{1}{2}}. \tag{4.24}$$

After finding u^{k+1} form (4.24), we compute $\tilde{a}^{k+1}$ using (4.21) and $\tilde{b}^{k+1}$ using (4.22), and then we compute $a^{k+1} = \kappa a^k + (1-\kappa)\tilde{a}^{k+1}$ and $b^{k+1} = \kappa b^k + (1-\kappa)\tilde{b}^{k+1}$. By incorporating the above ideas into Algorithm 3, we can obtain a fixed-point-like method, called Algorithm 4, for solving the TVL2 problem (1.5).

Algorithm 4 Fixed-point-like method for the TVL2 problem (1.5)

1: Given : observed image f, positive parameters α, β, γ, λ 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: $\quad a^{k+\frac{1}{2}} = a^k - \text{prox}_{\frac{1}{\gamma}\|\cdot\|_p}(Du^k + a^k)$
5: $\quad b^{k+\frac{1}{2}} = b^k - \text{prox}_{\frac{\beta}{\lambda}\varphi}(Bu^k + b^k)$
6: $\quad$ Solve $(A^T A + \alpha\gamma D^T D + \lambda B^T B)u^{k+1} = A^T f - \alpha\gamma D^T a^{k+\frac{1}{2}} - \lambda B^T b^{k+\frac{1}{2}}$ for u^{k+1}
7: $\quad \tilde{a}^{k+1} = Du^{k+1} + a^{k+\frac{1}{2}}$
8: $\quad a^{k+1} = \kappa a^k + (1-k)\tilde{a}^{k+1}$
9: $\quad \tilde{b}^{k+1} = Bu^{k+1} + b^{k+\frac{1}{2}}$
10: $\quad b^{k+1} = \kappa b^k + (1-k)\tilde{b}^{k+1}$
11: $\quad$ **if** $\frac{\|u^{k+1}-u^k\|_2}{\|u^{k+1}\|_2} < tol$ **then**
12: $\quad\quad$ Stop
13: $\quad$ **end if**
14: **end for**

Notice that the linear system in line 6 of Algorithm 4 is equivalent to solving the following least squares problem

$$\min_u \left\| \begin{pmatrix} A \\ \sqrt{\alpha\gamma}D \\ \sqrt{\lambda}B \end{pmatrix} u - \begin{pmatrix} f \\ -\sqrt{\alpha\gamma}a^{k+\frac{1}{2}} \\ -\sqrt{\lambda}b^{k+\frac{1}{2}} \end{pmatrix} \right\|_2^2. \tag{4.25}$$

Hence, the linear system in line 6 of Algorithm 4 is approximately solved by applying the CGLS with a tolerance value to the problem (4.25).

5. SPLIT BREGMAN METHOD FOR THE TVL2 PROBLEM (1.5)

In this section, we study an application of the alternating split Bregman method proposed in [8] to the TVL2 problem (1.5). The problem (1.5), or equivalently (4.1), can be considered as a constrained minimization problem

$$\min_{u,\,d,\,w} \left\{ \frac{1}{2}\|Au - f\|_2^2 + \alpha\|w\|_p + \beta\varphi(d) \right\} \text{ such that } d = Bu \text{ and } \omega = Du, \tag{5.1}$$

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

where $d \in \mathbb{R}^{2m}$, $B \in \mathbb{R}^{2m\times m}$, $w \in \mathbb{R}^m$ and $D \in \mathbb{R}^{m\times m}$. Rather than considering (5.1), we will consider the following unconstrained minimization problem with a penalty parameters $\lambda, \gamma > 0$

$$\min_{u,d,w}\left\{\frac{1}{2}\|Au-f\|_2^2+\alpha\|w\|_p+\beta\varphi(d)+\frac{\lambda}{2}\|w-Du-b\|_2^2+\frac{\gamma}{2}\|d-Bu-c\|_2^2\right\}. \tag{5.2}$$

Minimizing (5.2) alternatingly with respect to *u, w* and *d*, one can obtain the following alternating split Bregman method using auxiliary vectors *b* and *c* for solving the problem (1.5):

Given $u^0 = f$ and $\omega^0 = d^0 = b^0 = c^0 = 0$, the sequence $\{u^{k+1},\, w^{k+1},\, d^{k+1}\}$ is generated by the following iteration step

$$\begin{cases} u^{k+1} =\underset{u}{\operatorname{argmin}}\left\{\frac{1}{2}\|Au-f\|_2^2+\frac{\lambda}{2}\|w^k-Du-b^k\|_2^2+\frac{\gamma}{2}\|d^k-Bu-c^k\|_2^2\right\},\\ w^{k+1} =\underset{w}{\operatorname{argmin}}\left\{\alpha\|w\|_p+\frac{\lambda}{2}\|w-Du^{k+1}-b^k\|_2^2\right\},\\ d^{k+1} =\underset{d}{\operatorname{argmin}}\left\{\beta\varphi(d)+\frac{\gamma}{2}\|d-Bu^{k+1}-c^k\|_2^2\right\},\\ b^{k+1} =b^k+Du^{k+1}-w^{k+1},\\ c^{k+1} =c^k+Bu^{k+1}-d^{k+1}. \end{cases} \tag{5.3}$$

The convergence of the split Bregman method (5.3) is provided in the following theorem.

Theorem 5.1. *Assume that u^* is a solution of the problem* (1.5) *and $\alpha,\ \beta > 0$. Then we have the following property for the split Bregman method* (5.3)

$$\lim_{k\to\infty}\left\{\frac{1}{2}\|Au^k-f\|_2^2+\alpha\|Du^k\|_p+\beta\varphi(Bu^k)\right\}=\frac{1}{2}\|Au^*-f\|_2^2+\alpha\|Du^*\|_p+\beta\varphi(Bu^*).$$

Proof. We skip the proof of this theorem which can be done in a similar way to the convergence proof in [3]. □

The first equation of (5.3) is equivalent to solving the following linear system for $u^{k+1} \in \mathbb{R}^m$

$$(A^TA+\lambda D^TD+\gamma B^TB)u^{k+1}=A^Tf+\lambda D^T(w^k-b^k)+\gamma B^T(d^k-c^k), \tag{5.4}$$

and the second equation of (5.3) is equivalent to the following nonlinear equation for the proximal operator of $\frac{\alpha}{\lambda}\|\cdot\|_p$ at $(Du^{k+1}+b^k)$

$$\omega^{k+1}=\operatorname{prox}_{\frac{\alpha}{\lambda}\|\cdot\|_p}(Du^{k+1}+b^k). \tag{5.5}$$

From the third equation of (5.3), $d^{k+1} \in \mathbb{R}^{2m}$ can be obtained by using the following generalized shrinkage formula [8]:

For each $i = 1, 2, ..., m$, compute

$$\begin{pmatrix}(d^{k+1})_i\\(d^{k+1})_{m+i}\end{pmatrix}=\max\left\{\left\|\begin{pmatrix}(Bu^{k+1}+c^k)_i\\(Bu^{k+1}+c^k)_{m+i}\end{pmatrix}\right\|_2-\frac{\beta}{\gamma},0\right\}\cdot\frac{\begin{pmatrix}(Bu^{k+1}+c^k)_i\\(Bu^{k+1}+c^k)_{m+i}\end{pmatrix}}{\left\|\begin{pmatrix}(Bu^{k+1}+c^k)_i\\(Bu^{k+1}+c^k)_{m+i}\end{pmatrix}\right\|_2}. \tag{5.6}$$

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

Using (5.3), (5.4), (5.5) and (5.6), we can obtain a split Bregman method, called Algorithm 5, for solving the TVL2 problem (1.5).

Algorithm 5 Split Bregman method for the TVL2 problem (1.5)

1: Given : observed image f, positive parameters α, β, γ, λ
2: Initialization : $w^0 = 0$, $b^0 = 0$, $c^0 = 0$, $d^0 = 0$ and $u^0 = f$
3: **for** $k = 0$ to $maxit$ **do**
4: Solve $(A^TA + \lambda D^TD + \gamma B^TB)u^{k+1} = A^Tf + \lambda D^T(w^k - b^k) + \gamma B^T(d^k - c^k)$ for u^{k+1}
5: $w^{k+1} = \text{prox}_{\frac{\alpha}{\lambda}\|\cdot\|_p}(Du^{k+1} + b^k)$
6: **for** $i = 1, 2, \cdots, m$ **do**
7: $$\begin{pmatrix} (d^{k+1})_i \\ (d^{k+1})_{m+i} \end{pmatrix} = \max\left\{ \left\| \begin{pmatrix} (Bu^{k+1} + c^k)_i \\ (Bu^{k+1} + c^k)_{m+i} \end{pmatrix} \right\|_2 - \frac{\beta}{\gamma}, 0 \right\} \cdot \frac{\begin{pmatrix} (Bu^{k+1} + c^k)_i \\ (Bu^{k+1} + c^k)_{m+i} \end{pmatrix}}{\left\| \begin{pmatrix} (Bu^{k+1} + c^k)_i \\ (Bu^{k+1} + c^k)_{m+i} \end{pmatrix} \right\|_2}$$
8: **end for**
9: $b^{k+1} = (b^k + Du^{k+1}) - w^{k+1}$
10: $c^{k+1} = (c^k + Bu^{k+1}) - d^{k+1}$
11: **if** $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$ **then**
12: Stop
13: **end if**
14: **end for**

Note that the liner system in line 4 of Algorithm 5 is approximately solved using the CGLS with a tolerance value as was done in line 6 of Algorithm 4.

6. NUMERICAL EXPERIMENTS

In this section, we provide numerical experiments for several test problems to estimate the efficiency of the fixed-point-like method (Algorithm 4) and split Bregman method (Algorithm 5) for solving the new proposed TVL2 problem (1.5). Performances of these algorithms are evaluated by comparing their numerical results with those of Algorithms 1 and 2 for the TVL2 problem (1.4) proposed in [10].

All the experiments are implemented under Matlab R2016a running on a notebook computer with Intel(R) Core(TM) i5-3337U CPU and 8.00GB RAM Memory. In our experiments, we have used five test images which are the "Cameraman (Cam)", "Lena", "House", "Boat" and "Caribou" images. The pixel size of five test images is 256×256. In these experiments, we have used 3 types of point spread functions (PSFs) which are Gaussian blur, Average blur and Motion blur of size 9×9. The PSF arrays P for Gaussian blur, Average blur and Motion blur are generated by the built-in Matlab functions

$$\textit{fspecial('gaussian'}, [9.9], 9), \quad \textit{fspecial('average'}, 9),$$

and

$$P = \textit{zero}(9); \quad P(4:6,:) = \textit{fspecial('motion'}, 9, 1),$$

respectively. The blurred and noisy image f is generated by

$$f = A \cdot X(:) + \eta(:),$$

where A stands for the blurring matrix which can be generated by the original PSF array P according to the reflexive boundary condition, X represents the true image, and η is the Gaussian noise with mean 0 and standard derivation 3 which can be generated using the Matlab function *randn*. That is, $\eta = 3 \times randn(m, n)$, where (m, n) denotes the size of the true image X.

All algorithms tested in numerical experiments are terminated when the following stopping criterion was satisfied:

$$\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} \leq tol,$$

where *tol* denotes a prescribed tolerance value. We set $tol = 1 \times 10^{-3}$ for Algorithms 1 and 4, $tol = 5 \times 10^{-4}$ for Algorithm 2, and $tol = 2 \times 10^{-4}$ for Algorithm 5. For all test problems, an initial image was set to the blurred and noisy image f, we set $\kappa = 1 \times 10^{-6}$ and *maxit* = 150. For the CGLS, the maximum number of iterations is set to 60, and the tolerance number is set to 1×10^{-3} for Algorithms 1 and 4, and 5×10^{-2} for Algorithms 2 and 5.

To evaluate the quality of the restored image, we use the PSNR (Peak Singal to Noise Ratio) value between the restored image and original image which is defined by

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

where $\|\cdot\|_F$ represents the Frobenius norm, $\tilde{U}$ denotes the restored image of size $L \times N$, and u_{ij} stands for the value of the original image U at a pixel point (i, j) for $1 \leq i \leq L$ and $1 \leq j \leq N$. In Tables 1 and 2, "Alg." represents the algorithm number to be used, 4(k) and 5(k) under the "Alg." column denote Algorithm 4 with $p = k$ and Algorithm 5 with $p = k$ respectively, "PSNR_0" represents the PSNR value for the blurred and noisy images, "Iter" denotes the number of iterations required for Algorithms 1, 2, 4 and 5, "α, β, γ, λ" denote parameters which are chosen as the best one by numerical tries, and "Time" denotes the elapsed CPU time in seconds.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

Table 1: Numerical results for Fixed-point-like methods (i.e., Algorithms 1 and 4)

Image	PSF	$PSNR_0$	Alg.	α	β	γ	λ	PSNR	Iter	Time
Cam	Gaussian	20.85	1	1.1500	0.135	0.050	0.0100	25.51	132	39.1
			4(2)	0.0200	0.133	0.065	0.0050	25.53	19	20.7
			4(1)	0.0060	0.125	0.008	0.0080	25.54	16	16.0
	Average	20.76	1	0.7500	0.140	0.095	0.0095	25.59	120	33.9
			4(2)	0.0200	0.140	0.065	0.0050	25.61	19	22.3
			4(1)	0.0070	0.130	0.008	0.0070	25.62	15	16.1
	Motion	21.85	1	0.8500	0.230	0.085	0.0020	28.57	67	10.4
			4(2)	0.0003	0.230	0.010	0.0055	28.56	12	7.25
			4(1)	0.0120	0.200	0.010	0.0050	28.56	12	5.63
Lena	Gaussian	22.55	1	1.9000	0.170	0.085	0.0100	26.22	92	21.4
			4(2)	0.0900	0.165	0.145	0.0070	26.32	26	16.2
			4(1)	0.0160	0.150	0.100	0.0050	26.39	15	17.1
	Average	22.44	1	1.9000	0.180	0.075	0.0095	26.27	107	25.4
			4(2)	0.0900	0.175	0.150	0.0065	26.36	27	17.6
			4(1)	0.0190	0.150	0.080	0.0050	26.43	15	17.2
	Motion	23.06	1	1.1000	0.260	0.075	0.0075	28.45	46	6.95
			4(2)	0.0480	0.260	0.135	0.0088	28.47	19	6.86
			4(1)	0.0300	0.210	0.010	0.0100	28.59	11	5.08
House	Gaussian	24.19	1	3.2000	0.200	0.075	0.0200	30.30	82	18.1
			4(2)	0.0400	0.210	0.180	0.0250	30.56	15	7.64
			4(1)	0.0230	0.180	0.100	0.0070	30.57	12	12.3
	Average	24.05	1	3.1000	0.190	0.070	0.0095	30.24	117	25.2
			4(2)	0.0500	0.210	0.260	0.0160	30.56	18	9.88
			4(1)	0.0210	0.180	0.400	0.0090	30.57	14	9.79
	Motion	27.01	1	1.5000	0.550	0.045	0.0100	33.32	82	13.1
			4(2)	2.6500	0.450	0.007	0.0300	33.65	13	3.51
			4(1)	0.0950	0.400	0.002	0.0220	33.68	11	3.67
Boat	Gaussian	21.28	1	2.5000	0.130	0.070	0.0090	25.17	58	17.6
			4(2)	0.0160	0.125	0.750	0.0150	25.35	20	13.4
			4(1)	0.0160	0.110	0.500	0.0100	25.37	16	12.6
	Average	21.19	1	2.4000	0.135	0.075	0.0080	25.24	62	15.9
			4(2)	0.0160	0.125	0.850	0.0150	25.42	21	14.0
			4(1)	0.0140	0.110	0.600	0.0100	25.44	17	13.7
	Motion	23.32	1	0.6000	0.265	0.090	0.0200	27.87	34	6.54
			4(2)	0.0180	0.240	0.850	0.0170	27.99	18	5.76
			4(1)	0.0280	0.200	0.400	0.0080	28.03	16	6.39
Caribou	Gaussian	23.69	1	3.5000	0.175	0.050	0.0250	27.00	42	10.8
			4(2)	0.0380	0.120	0.900	0.0250	27.33	17	13.2
			4(1)	0.0310	0.090	0.900	0.0100	27.37	15	12.1
	Average	23.57	1	3.3000	0.170	0.050	0.0230	27.00	42	11.0
			4(2)	0.0530	0.125	0.900	0.0150	27.35	17	15.5
			4(1)	0.0330	0.095	1.100	0.0080	27.38	15	13.2
	Motion	25.64	1	9.3000	0.340	0.009	0.0060	29.62	44	7.19
			4(2)	0.0800	0.200	0.900	0.0200	30.13	16	7.31
			4(1)	0.0650	0.150	0.820	0.0040	30.20	15	7.29

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

Table 2: Numerical results for Split Bregman methods (i.e., Algorithms 2 and 5)

Image	PSF	$PSNR_0$	Alg.	α	β	γ	λ	PSNR	Iter	Time
Cam	Gaussian	20.85	2	0.1500	0.140	0.005000	0.0080	25.52	28	11.7
			5(2)	0.0800	0.132	0.007000	0.0045	25.55	64	25.0
			5(1)	0.0060	0.125	0.000050	0.0080	25.55	39	19.3
	Average	20.76	2	0.0100	0.140	0.002000	0.0090	25.61	27	12.0
			5(2)	0.0200	0.140	0.007000	0.0050	25.63	64	24.4
			5(1)	0.0074	0.124	0.000100	0.0058	25.63	35	20.1
	Motion	21.85	2	0.0100	0.240	0.030000	0.0030	28.59	37	5.84
			5(2)	0.0010	0.235	0.000600	0.0050	28.56	32	8.93
			5(1)	0.0100	0.210	0.001000	0.0080	28.57	26	6.03
Lena	Gaussian	22.55	2	0.0100	0.175	0.000090	0.0400	26.29	28	8.14
			5(2)	0.0100	0.165	0.040000	0.0200	26.33	67	23.9
			5(1)	0.0200	0.140	0.000100	0.0090	26.40	40	19.2
	Average	22.44	2	0.0100	0.180	0.000100	0.0400	26.33	28	8.33
			5(2)	0.0400	0.175	0.040000	0.0200	26.37	68	23.4
			5(1)	0.0200	0.150	0.000200	0.0100	26.44	34	16.3
	Motion	23.06	2	0.0100	0.265	0.000200	0.0250	28.46	18	2.88
			5(2)	0.0100	0.255	0.030000	0.0200	28.48	57	8.76
			5(1)	0.0300	0.210	0.002000	0.0100	28.58	25	5.30
House	Gaussian	24.19	2	0.0100	0.210	0.000100	0.0550	30.52	24	6.76
			5(2)	0.0100	0.200	0.060000	0.0150	30.60	61	24.4
			5(1)	0.0200	0.180	0.010000	0.0150	30.61	36	12.2
	Average	24.05	2	0.0100	0.215	0.000100	0.0550	30.50	24	6.75
			5(2)	0.0100	0.210	0.065000	0.0150	30.59	61	24.8
			5(1)	0.0200	0.190	0.020000	0.0100	30.61	44	15.8
	Motion	27.01	2	0.0100	0.520	0.002000	0.1000	33.52	18	2.29
			5(2)	0.1000	0.450	0.180000	0.0200	33.70	56	11.3
			5(1)	0.3000	0.430	0.000027	0.0370	33.70	61	9.44
Boat	Gaussian	21.28	2	0.1000	0.120	0.000300	0.7500	25.31	36	10.3
			5(2)	0.1000	0.120	0.070000	0.0200	25.37	80	33.8
			5(1)	0.0100	0.110	0.040000	0.0100	25.38	67	27.6
	Average	21.19	2	0.0100	0.130	0.000100	0.0700	25.38	35	10.2
			5(2)	0.1000	0.125	0.075000	0.0150	25.44	81	34.3
			5(1)	0.0090	0.120	0.050000	0.0090	25.45	72	31.3
	Motion	23.32	2	0.0100	0.255	0.000100	0.0500	27.91	22	3.25
			5(2)	0.1000	0.230	0.080000	0.0100	28.02	74	13.6
			5(1)	0.0300	0.210	0.040000	0.0070	28.03	55	9.96
Caribou	Gaussian	23.69	2	0.1000	0.115	0.000100	0.1900	27.28	33	9.02
			5(2)	0.3000	0.100	0.400000	0.0010	27.43	74	49.0
			5(1)	0.0200	0.090	0.220000	0.0009	27.43	61	37.9
	Average	23.57	2	0.1000	0.115	0.000100	0.1900	27.28	33	9.04
			5(2)	0.2000	0.100	0.400000	0.0010	27.44	75	49.9
			5(1)	0.0190	0.090	0.260000	0.0005	27.44	71	46.0
	Motion	25.64	2	0.0500	0.215	0.000100	0.3500	29.94	32	4.20
			5(2)	0.3500	0.200	0.450000	0.0010	30.28	70	19.0
			5(1)	0.0400	0.180	0.300000	0.0008	30.28	61	16.0

Table 1 contains numerical results for Algorithms 1 and 4 which are fixed-point-like methods corresponding to the TVL2 problems (1.4) and (1.5), and Table 2 contains numerical results for Algorithms 2 and 5 which are split Bregman methods corresponding to the TVL2 problems (1.4) and (1.5). Figure 1 shows the true images, and Figure 2 shows the restored images by Algorithms 1, 4 and 5 with $p = 1$.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 10 (2020), pp. 2674-2685
 https://dx.doi.org/10.37624/IJERT/13.10.2020.2674-2685

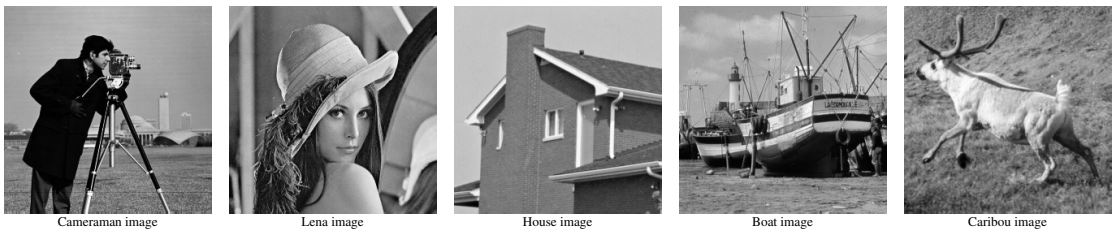

Figure 1: True images

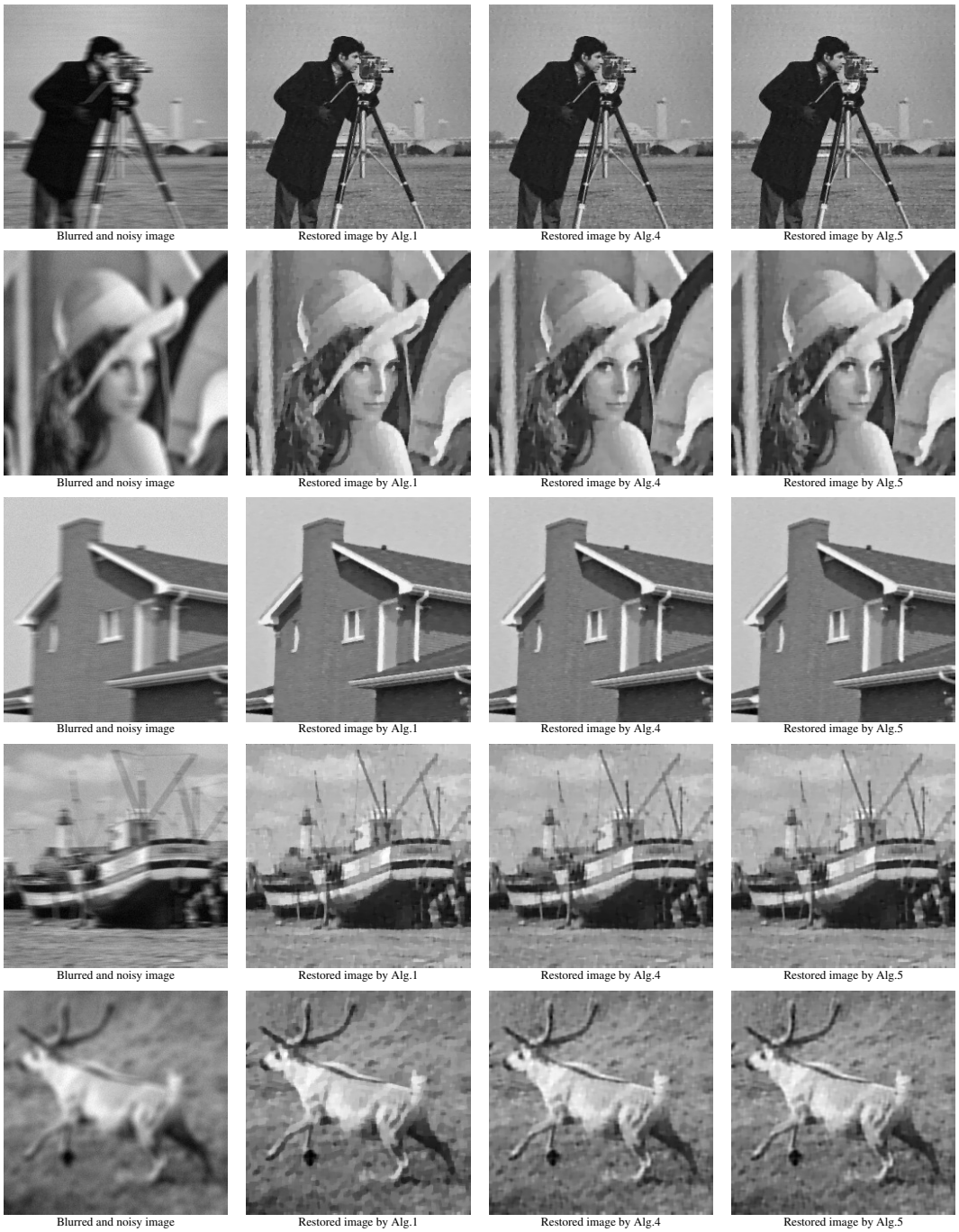

Figure 2: Restored images by Algorithms 1, 4 and 5 with $p = 1$ (The first row images are Cameraman images for Motion blur, the second row images are Lena images for Average blur, the third row images are House images for Motion blur and the fourth row images are Boat images for Average blur and the last row images are Caribou images for Gaussian blur).

By comparing the PSNR values in Tables 1 and 2, it can be seen that Algorithms 4 and 5 restore the true image better than Algorithms 1 and 2 respectively, and Algorithms 4 and 5 with $p = 1$ restore the true image as well as or a little bit better than Algorithms 4 and 5 with $p = 2$ respectively. By comparing the CPU times in Tables 1 and 2, it can be seen that for the fixed-point-like methods Algorithm 4 takes less CPU time than Algorithm 1, while for the split Bregman methods Algorithm 5 takes more CPU time than Algorithm 2. In addition, Algorithms 4 and 5 with $p = 1$ take less CPU time than Algorithms 4 and 5 with $p = 2$ for most cases. Overall, Algorithm 4 for the new proposed TVL2 model (1.5) performs better in both PSNR value and CPU time than Algorithm 1 for the TVL2 model (1.4), and Algorithm 5 for the new TVL2 model (1.5) restores the true image better than Algorithm 2 for the TVL2 model (1.4) at the expense of much increase in CPU time. It can be also seen that Algorithms 4 and 5 with $p = 1$ perform a little bit better than Algorithms 4 and 5 with $p = 2$ respectively.

As can be seen in Tables 1 and 2, for the TVL2 model (1.4) the fixed-point-like method restores the true image significantly worse than the corresponding split Bregman method, but for the TVL2 model (1.5) the fixed-point-like method restores the true image as well as or a little bit worse than the corresponding split Bregman method. In other words, as compared with the split Bregman method, the fixed-point-like method for the TVL2 model (1.5) works better than that for the TVL2 model (1.4).

7. CONCLUSION

In this paper, we have proposed a new TVL2 regularization model (1.5) for image restoration, and then we have developed a fixed-point-like method, called Algorithm 4, and a split Bregman method, called Algorithm 5, for solving the new proposed TVL2 model (1.5).

Numerical experiments for serval test problems showed that Algorithm 4 for the new TVL2 model (1.5) performs better in both PSNR value and CPU time than Algorithm 1 for the TVL2 model (1.4), and Algorithm 5 for the new TVL2 model (1.5) restores the true image better than Algorithm 2 for the TVL2 model (1.4) at the expense of much increase in CPU time. It was also shown that Algorithms 4 and 5 with $p = 1$ perform a little bit better than Algorithms 4 and 5 with $p = 2$, respectively. As compared with the split Bregman methods for the TVL2 models (1.4) and (1.5), the fixed-point-like method for the TVL2 model (1.5) works better than that for the TVL2 model (1.4). Hence, it can be concluded that the new TVL2 model (1.5) is preferred over the TVL2 model (1.4), and the fixed-point-like method (Algorithm 4 with $p = 1$) and split Bregman method (Algorithm 5 with $p = 1$) for the new TVL2 model (1.5) are recommended to use for image restoration.

Future work will study the problem of finding optimal or nearly optimal parameters for the fixed-point-like method and split Bregman method which is very challenging problem.

REFERENCES

[1] A. Bjorck, *Numerical methods for least squares problems*, SIAM, Philadelphia **1996**

[2] A. Beck, *First-order methods in optimization,* SIAM, Philadelphia **2017**

[3] J.F. Cai, S. Osher and Z. Shen, *Split Bregman methods and frame based image restoration,* Multiscale Model. Simul. **2009** (8), 337-369.

[4] T.F. Chan and P. Mulet, *On the convergence of the lagged diffusivity fixed point method in total variation image restoration,* SIAM J. Numer. Anal. **1999** (36), 354-367.

[5] D.Q. Chen, H. Zhang and L.Z. Cheng, *A fast fixed point algorithm for total variation deblurring and segmentation,* Journal of Math. Imaging Vis. **2012** (43), 167-179.

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

[7] W. Gautschi *Numerical analysis,* Springer, Berlin **1997**

[8] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems,* SIAM J. Imaging Sci. **2009** (2), 323-343.

[9] P.C. Hansen, J.G. Nagy and D.P. O'leary *Deblurring images: matrices, spectra, and filtering,* SIAM, Philadelphia **2006**

[10] K.S. Kim and J.H. Yun, *Image Restoration Using a Fixed-Point Method for a TVL2 Regularization Problem,* Algorithms **2020** (13), Article No. 1 (15 pages).

[11] C.A. Micchelli, L. Shen and Y. Xu, *Proximity algorithms for image models: denoising,* Inverse Problems **2011** (27), Article No. 045009 (25 pages).

[12] J.J. Moreau, *Proximité et dualité dans un espace hilbertien,* Bull. Soc. Math. France. **1965** (93), 273-299.

[13] M.K. Ng, L. Qi, Y.F. Yang and Y.M. Huang, *On semismooth Newton's methods for total variation minimization,* J. Math. Imaging Vis. **2007** (27), 265-276.

[14] Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings,* Bull. Amer. Math. Soc. **1967** (73), 591-597.

[15] L.I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms,* Physica D **1992** (60), 259-268.