J. Appl. Math. & Informatics Vol. **38***(2020), No. 5 - 6, pp. 519 - 532*
https://doi.org/10.14317/jami.2020.519

FIXED-POINT-LIKE METHOD FOR A NEW TOTAL VARIATION-BASED IMAGE RESTORATION MODEL[†]

YU JIN WON AND JAE HEON YUN[*]

Abstract. In this paper, we first propose a new total variation-based regularization model for image restoration. We next propose a fixed-point-like method for solving the new image restoration model, and then we provide convergence analysis for the fixed-point-like method. To evaluate the feasibility and efficiency of the fixed-point-like method for the new proposed total variation-based regularization model, we provide numerical experiments for several test problems.

AMS Mathematics Subject Classification : 47H10, 54H25, 68U10, 94A08.
Key word and phrases : Image restoration, Total variation, Fixed-point method, Split Bregman method, Proximity operator.

1. Introduction

Image restoration is the essential problem in image processing which recovers a true image from a blurred and noisy image. The problem of image restoration usually reduces to find the optimal solution $u \in \mathbb{R}^{n^2}$ based on the following model:

$$f = Au + \epsilon, \tag{1}$$

where $A \in \mathbb{R}^{n^2 \times n^2}$ is a blurring operator, $\epsilon \in \mathbb{R}^{n^2}$ is an unknown white Gaussian noise with variance σ, f denotes an observed degraded image and u denotes the original image. 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 have been widely studied and applied in many image processing fields.

Received May 26, 2020. Revised July 23, 2020. Accepted July 30, 2020. [*]Corresponding author

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

The well-known ROF (Rudin-Osher-Fatemi) total variation model [13] produces the deblurred image given by the following minimization problem:

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

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

Many computational methods for solving the problem (2) have been proposed in recent years. For instance, the subgradient descent method, the split Bregman method, the fixed-point method, and the fixed-point-like method have been proposed by many researchers [1, 4, 5, 6, 7, 9, 10, 11, 14, 15, 16].

Recently, Chen et al. [6] proposed a fixed-point method for solving the following regularization model for image restoration, called TVL2I2,

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

where α and β are positive regularization parameters. This approach motivates us to propose a new regularization model for image restoration, called TVL2D2,

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

where α and β are positive regularization parameters, $D = -\Delta$ and Δ denotes a discrete Laplacian operator, and $TV(u)$ represents the isotropic TV of u. The isotropic TV of u is defined by

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

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

$$(\nabla u)_k = ((\nabla u)_k^x, (\nabla u)_{n^2+k}^y)^T, \quad k = 1, 2, \cdots, n^2$$

with

$$(\nabla u)_k^x = \begin{cases} 0 & \text{if } k \text{ mod n} = 1 \\ u_k - u_{k-1} & \text{if } k \text{ mod n} \neq 1 \end{cases} \quad \text{and} \quad (\nabla u)_{n^2+k}^y = \begin{cases} 0 & \text{if } k \leq \text{n} \\ u_k - u_{k-n} & \text{if } k > \text{n} \end{cases}$$

Notice that the TVL2I2 problem (3) has a unique solution since its objective function is strictly convex, while the new TVL2D2 problem (4) may not have a unique solution since its objective function is not strictly convex, but just convex.

The purpose of this paper is to propose a fixed-point-like method for solving the new proposed TVL2D2 problem (4). This paper is organized as follows. In the second section, we introduce some definitions and properties that are used in this paper. In the third section, we review the fixed-point method for solving the existing TVL2I2 problem (3). In the fourth section, we first propose a fixed-point-like method for solving the new TVL2D2 problem (4), and then we provide a convergence analysis for the fixed-point-like method. In the fifth section, we just provide the split Bregman method for solving the TVL2D2 problem (4) to see how efficiently the fixed-point-like method performs. In the sixth section, we provide numerical experiments for several test problems in order to evaluate the effectiveness of the fixed-point-like method for solving the new proposed TVL2D2 problem (4). Lastly, some conclusions are drawn.

2. Preliminaries

In this section, we provide some definitions and important properties which we refer to later.

Definition 2.1. Let $\psi : \mathbb{R}^n \to (-\infty, \infty]$ be a proper function, and let $dom(\psi)$ denote the *domain* of ψ, that is, $dom(\psi) = \{x \in \mathbb{R}^n : \psi(x) < \infty\}$. For an $x \in dom(\psi)$, the *subdifferential* of ψ at $x \in \mathbb{R}^n$ is defined by

$$\partial\psi(x) = \{y \in \mathbb{R}^n : \psi(z) \geq \psi(x) + \langle y, z - x\rangle, \ {}^{\forall}z \in \mathbb{R}^n\}. \tag{5}$$

Theorem 2.2 ([2])**.** Let $\psi : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function, and assume that $x \in \text{int}(dom(\psi))$. Then $\partial\psi(x)$ is nonempty and bounded.

Definition 2.3. Let $\psi : \mathbb{R}^n \to \mathbb{R}^n$ be a non-linear operator. ψ is *non-expansive* if for any $x, y \in \mathbb{R}^n$,

$$\|\psi(x) - \psi(y)\|_2 \leq \|x - y\|_2. \tag{6}$$

Definition 2.4. Let $\psi : \mathbb{R}^n \to \mathbb{R}^n$ be a non-linear operator. ψ is *firmly non-expansive* if for any $x, y \in \mathbb{R}^n$,

$$\|\psi(x) - \psi(y)\|_2^2 \leq \langle x - y, \psi(x) - \psi(y)\rangle \,. \tag{7}$$

It is easy to show that if a non-linear operator $\psi : \mathbb{R}^n \to \mathbb{R}^n$ is firmly non-expansive, then ψ is non-expansive.

Definition 2.5. Let $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and lower semi-continuous function. The *proximity operator* of ψ at $x \in \mathbb{R}^n$ is defined by

$$prox_{\psi}(x) = \arg\min_{u} \left\{ \psi(u) + \frac{1}{2}\|u - x\|_2^2 : u \in \mathbb{R}^n \right\}. \tag{8}$$

Proposition 2.6 ([6, 9])**.** Let $\psi : \mathbb{R}^n \to \mathbb{R}^n$ be a convex function. For $x, y \in \mathbb{R}^n$,

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

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

$$\varphi(d) = \sum_{i=1}^{n^2} \left\| \begin{pmatrix} d_i \\ d_{n^2+i} \end{pmatrix} \right\|_2 \text{ for each } d = (d_i) \in \mathbb{R}^{2\mathrm{n}^2}, \tag{9}$$

and let B be a $2n^2 \times n^2$ matrix that represents a discrete gradient operator ∇. Then the matrix B can be expressed as

$$B = \begin{pmatrix} I_n \otimes D_1 \\ D_1 \otimes I_n \end{pmatrix} \in \mathbb{R}^{2n^2 \times n^2},$$

where I_n is the identity matrix of order n, $\otimes$ denotes the Kronecker product, and

$$D_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & 0 & \ddots & & \vdots \\ 0 & -1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix},$$

which is a matrix of order n. Hence, $TV(u)$ can be represented by

$$TV(u) = (\varphi \circ B)(u). \tag{10}$$

Since D is a discrete Laplacian operator, D can be expressed as

$$D = (I_n \otimes D_2 + D_2 \otimes I_n) \in \mathbb{R}^{n^2 \times n^2},$$

where

$$D_2 = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix},$$

which is a matrix of order n.

3. Fixed-point Method for the TVL2I2 Problem

In this section, we briefly review the fixed-point method proposed in [6] for solving the TVL2I2 problem (3) for the purpose of comparison with a fixed-point-like method to be proposed in this paper. The fixed-point method, called Algorithm 1, for the TVL2I2 problem (3) is described below (see [6] for the detailed description of this algorithm).

In this paper, the linear system in line 6 of Algorithm 1 is solved using the CGLS (Conjugate gradient least squares method) [3] instead of using the CG (Conjugate gradient method) [8].

4. Fixed-point-like method for the TVL2D2 Problem

In this section, we first propose a fixed-point-like method for solving the TVL2D2 regularization problem (4), and then we provide a convergence analysis

Algorithm 1 Fixed-point method for the TVL2I2 problem (3)

```
Given : observed image $f$, positive parameters $\alpha, \beta, \gamma$, and $\kappa \in (0,1)$
Initialization : $b_0 = 0$ and $u_0 = f$
for $k = 0$ to $maxit$ do
    $\hat{b}_{k+1} = \left(I - \text{prox}_{\frac{\beta}{\gamma}\varphi}\right)\left(Bu_k + b_k\right)$
    $b_{k+1} = \kappa b_k + (1-\kappa)\,\hat{b}_{k+1}$
    Solve $\left(A^T A + \alpha I\right) u_{k+1} = A^T f - \gamma B^T b_{k+1}$ for $u_{k+1}$
    if $\frac{\|u_{k+1}-u_k\|_2}{\|u_{k+1}\|_2} < tol$ then
        Stop
    end if
end for
```

for the fixed-point-like method. Using (10), the TVL2D2 problem (4) can be expressed as

$$\min_{u\in\mathbb{R}^{n^2}} \left\{ \frac{1}{2}\|Au-f\|_2^2 + \frac{\alpha}{2}\|Du\|_2^2 + \beta(\varphi\circ B)(u) \right\}. \tag{11}$$

Theorem 4.1. *If u is a solution of the TVL2D2 problem* (11)*, then for any $\gamma > 0$ there exists a vector $b \in \mathbb{R}^{2n^2}$ such that*

$$b = (I - \text{prox}_{\frac{\beta}{\gamma}\varphi})(\text{Bu} + \text{b}), \tag{12}$$

$$(A^T A + \alpha D^T D)u = A^T f - \gamma B^T b. \tag{13}$$

Conversely, if there exist $\gamma > 0$, $b \in \mathbb{R}^{2n^2}$ and $u \in \mathbb{R}^{n^2}$ satisfying (12) *and* (13)*, then u is a solution of the TVL2D2 problem* (11)*.*

Proof. Assume that u is a solution of (11). Using the Fermet's rule for the equation (11), we can obtain

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

This relation is equivalent to

$$0 \in (A^T A + \alpha D^T D)u - A^T f + \beta B^T \circ (\partial\varphi)\circ(Bu). \tag{14}$$

From (14), for any $\gamma > 0$ we can choose $b \in \partial(\frac{\beta}{\gamma}\varphi)(Bu)$ satisfying

$$A^T(Au-f) + \alpha D^T Du + \gamma B^T b = 0 \tag{15}$$

By Proposition 2.6, the relation $b \in \partial(\frac{\beta}{\gamma}\varphi)(Bu)$ implies

$$Bu = \text{prox}_{\frac{\beta}{\gamma}\varphi}(Bu + b) \tag{16}$$

From (15) and (16), we obtain (12) and (13).

Conversely, suppose that there exist $\gamma > 0$, $b \in \mathbb{R}^{2n^2}$ and $u \in \mathbb{R}^{n^2}$ satisfying (12) and (13). From (13), we obtain the equation (15). By Proposition 2.6, (12) ensures that $b \in \partial(\frac{\beta}{\gamma}\varphi)(Bu)$. Therefore, we obtain

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

Consequently, (14) holds. Thus, u is a solution of (11). □

From (12) and (13), we can develop a fixed-point algorithm which converges to a solution to (4). We now describe how to develop the fixed-point algorithm. Let u be an approximate solution to the ill-conditioned linear system (12). Then, u can be expressed as

$$u = M_\alpha(A^T f - \gamma D^T D) \tag{17}$$

where M_α is a symmetric positive semi-definite matrix approximating a solution of (13). For example, we can choose $M_\alpha = (A^T A + \alpha D^T D)_r^\dagger$, which is the truncated pseudoinverse of $A^T A + \alpha D^T D$ using the r largest positive singular values.

Substituting (17) into (12), one obtains

$$b = (I - \text{prox}_{\frac{\beta}{\gamma}\varphi})((I - \gamma B M_\alpha B^T)b + B M_\alpha A^T f) \tag{18}$$

We introduce the affine transformation $L : \mathbb{R}^{2n^2} \to \mathbb{R}^{2n^2}$ defined by

$$L(b) = (I - \gamma B M_\alpha B^T)b + B M_\alpha A^T f \text{ at a vector } b \in \mathbb{R}^{2n^2} \tag{19}$$

and the operator $G : \mathbb{R}^{2n^2} \to \mathbb{R}^{2n^2}$ given by

$$G = (I - \text{prox}_{\frac{\beta}{\gamma}\varphi}) \circ L. \tag{20}$$

Then, (18) can be expressed as

$$b = Gb. \tag{21}$$

In other words, G has a fixed point.

Since $\text{prox}_{\frac{\beta}{\gamma}\varphi}$ and $\left(I - \text{prox}_{\frac{\beta}{\gamma}\varphi}\right)$ are firmly non-expansive, they are non-expansive. Using these properties, we can obtain the following theorem.

Theorem 4.2. *If α and γ are positive parameters such that $\|I - \gamma B M_\alpha B^T\|_2 \leq 1$, then the operator G defined by* (20) *is non-expansive.*

Proof. Since $I - \text{prox}_{\frac{\beta}{\gamma}\varphi}$ is non-expansive and $\|I - \gamma B M_\alpha B^T\|_2 \leq 1$, for all $b_1, b_2 \in \mathbb{R}^{2n^2}$

$$\begin{aligned} \|G(b_1) - G(b_2)\|_2 &\leq \|(I - \text{prox}_{\frac{\beta}{\gamma}\varphi})(L(b_1)) - (I - \text{prox}_{\frac{\beta}{\gamma}\varphi})(L(b_2))\|_2 \\ &\leq \|L(b_1) - L(b_2)\|_2 \\ &= \|(I - \gamma B M_\alpha B^T)(b_1 - b_2)\|_2 \\ &\leq \|I - \gamma B M_\alpha B^T\|_2 \cdot \|b_1 - b_2\|_2 \\ &\leq \|b_1 - b_2\|_2. \end{aligned}$$

Thus, G is non-expansive. □

Definition 4.3. Let $S : \mathbb{R}^n \to \mathbb{R}^n$ be an operator. Then, the *Picard iteration* of S is defined by

$$x_{k+1} = Sx_k, \text{ for } k = 0, 1, 2, \cdots$$

for a given vector $x_0 \in \mathbb{R}^n$.

Definition 4.4. Let $S : \mathbb{R}^n \to \mathbb{R}^n$ be an operator and $\kappa \in (0, 1)$. Then, the *κ-averaged operator* S_κ of S is defined by

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

Proposition 4.5 ([6, 12])**.** *Let $C \subset \mathbb{R}^{2n^2}$ 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 $\kappa \in (0, 1)$, the Picard iteration of S_κ converges to a fixed point of S.*

Theorem 4.6. *Let G be the operator defined by* (20)*. If α and γ are positive parameters such that $\|I - \gamma BM_\alpha B^T\|_2 \leq 1$, then for any $\kappa \in (0, 1)$ the Picard iteration of G_κ converges to a fixed point of G.*

Proof. Since G has a fixed point from (21) and G is non-expansive by Theorem 4.2, for any $x_0 \in \mathbb{R}^{2n^2}$ and $\kappa \in (0, 1)$ the Picard iteration of G_κ converges to a fixed point of G by Proposition 4.5 □

From Theorem 4.6 and the Picard iteration of the κ-averaged operator $G_\kappa = \kappa I + (1 - \kappa) G$ of G, we can obtain a fixed-point method, called Algorithm 2, which converges to a solution to the TVL2D2 problem (4).

Algorithm 2 Fixed-point method for the TVL2D2 problem (4)

```
1:  Given : observed image $f$, positive parameters $\alpha, \beta, \gamma$, and $\kappa \in (0,1)$
2:  Initialization : $b_0 = 0$ and $u_0 = f$
3:  for $k = 0$ to $maxit$ do
4:      $\hat{b}_{k+1} = \left(I - \mathrm{prox}_{\frac{\beta}{\gamma}\varphi}\right)\left(Bu_k + b_k\right)$
5:      $b_{k+1} = \kappa b_k + (1 - \kappa)\,\hat{b}_{k+1}$
6:      Solve $\left(A^T A + \alpha D^T D\right) u_{k+1} = A^T f - \gamma B^T b_{k+1}$ for $u_{k+1}$
7:      if $\frac{\|u_{k+1} - u_k\|_2}{\|u_{k+1}\|_2} < tol$ then
8:          Stop
9:      end if
10: end for
```

The linear system in line 6 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

$$u_{k+1} = M_\alpha(A^T f - \gamma B^T b_{k+1}),$$

where $M_\alpha = \left(A^TA + \alpha D^TD\right)_r^\dagger$. However, when A is large, the computation of $\left(A^TA + \alpha D^TD\right)_r^\dagger$ is very time-consuming. So we need to develop more efficient method than Algorithm 2. To this end, we propose a fixed-poin-like method that can be obtained by modifying Algorithm 2. Below we describe how to develop the fixed-point-like method in detail. Since

$$\begin{aligned}\hat{b}_{k+1} &= (I - \operatorname{prox}_{\frac{\beta}{\gamma}\varphi})(Bu_k + b_k) \\ &= Bu_k + (b_k - \operatorname{prox}_{\frac{\beta}{\gamma}\varphi}(Bu_k + b_k)),\end{aligned}$$

we can split line 4 of Algorithm 2 into

$$\hat{b}_{k+1} = Bu_k + b_{k+\frac{1}{2}}, \tag{22}$$

where $b_{k+\frac{1}{2}} = b_k - \operatorname{prox}_{\frac{\beta}{\gamma}\varphi}(Bu_k + b_k)$. Replacing u_k of (22) with the new updated value of u_{k+1}, one can obtain

$$\hat{b}_{k+1} = Bu_{k+1} + b_{k+\frac{1}{2}}. \tag{23}$$

Then, the solution step in line 6 of Algorithm 2 is changed to

$$\left(A^TA + \alpha D^TD\right) u_{k+1} = A^Tf - \gamma B^T\hat{b}_{k+1}, \tag{24}$$

where $\hat{b}_{k+1}$ is computed using (23) instead of (22). Substituting (23) into (24), one obtains

$$\left(A^TA + \alpha D^TD + \gamma B^TB\right) u_{k+1} = A^Tf - \gamma B^Tb_{k+\frac{1}{2}}. \tag{25}$$

After finding u_{k+1} from the linear system (25), we compute $\hat{b}_{k+1}$ using (23) and $b_{k+1} = \kappa b_k + (1-\kappa)\,\hat{b}_{k+1}$. By applying the above ideas to Algorithm 2, we can obtain a fixed-point-like method, called Algorithm 3, for solving the TVL2D2 problem (4).

Algorithm 3 Fixed-point-like method for the TVL2D2 problem (4)

```
Given : observed image $f$, positive parameters $\alpha, \beta, \gamma$, and $\kappa \in (0,1)$
Initialization : $b_0 = 0$ and $u_0 = f$
for $k = 0$ to $maxit$ do
    $b_{k+\frac{1}{2}} = b_k - \operatorname{prox}_{\frac{\beta}{\gamma}\varphi}(Bu_k + b_k)$
    Solve $\left(A^TA + \alpha D^TD + \gamma B^TB\right) u_{k+1} = A^Tf - \gamma B^Tb_{k+\frac{1}{2}}$ for $u_{k+1}$
    $\hat{b}_{k+1} = Bu_{k+1} + b_{k+\frac{1}{2}}$
    $b_{k+1} = \kappa b_k + (1-\kappa)\,\hat{b}_{k+1}$
    if $\frac{\|u_{k+1} - u_k\|_2}{\|u_{k+1}\|_2} < tol$ then
        Stop
    end if
end for
```

Notice that the linear system in line 5 of Algorithm 3 is equivalent to solving the following least squares problem

$$\min_u \left\| \begin{pmatrix} A \\ \sqrt{\alpha}D \\ \sqrt{\gamma}B \end{pmatrix} u - \begin{pmatrix} f \\ 0 \\ -\sqrt{\gamma}b_{k+\frac{1}{2}} \end{pmatrix} \right\|_2^2 . \tag{26}$$

Hence, the linear system in line 5 of Algorithm 3 is solved by applying the CGLS to the problem (26).

5. Split Bregman Method for the TVL2D2 problem

In this section, we just provide the alternating split Bregman method, called Algorithm 4, for solving the TVL2 problem (4) in order to evaluate the performance of Algorithm 3. For more details about the alternating split Bregman method as well as its convergence analysis, we refer to [4, 7].

Algorithm 4 Split Bregman method for the TVL2D2 problem (4)

```
1: Given : observed image $f$, positive parameters $\alpha, \beta$ and $\gamma$
2: Initialization : $c_0 = 0, d_0 = 0$ and $u_0 = f$
3: for $k = 0$ to $maxit$ do
4:     Solve $\left(A^T A + \alpha D^T D + \gamma B^T B\right) u_{k+1} = A^T f - \gamma B^T \left(d_k - c_k\right)$ for $u_{k+1}$
5:     for $i = 1, 2, \cdots, n^2$ do
6:         $\begin{pmatrix} (d_{k+1})_i \\ (d_{k+1})_{n^2+i} \end{pmatrix} = \max\left\{ \left\| \begin{pmatrix} (Bu_k + c_k)_i \\ (Bu_k + c_k)_{n^2+i} \end{pmatrix} \right\|_2 - \frac{\beta}{\gamma}, 0 \right\} \cdot \frac{\begin{pmatrix} (Bu_k + c_k)_i \\ (Bu_k + c_k)_{n^2+i} \end{pmatrix}}{\left\| \begin{pmatrix} (Bu_k + c_k)_i \\ (Bu_k + c_k)_{n^2+i} \end{pmatrix} \right\|_2}$
7:     end for
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
```

As was done in line 5 of Algorithm 3, the linear system in line 4 of Algorithm 4 is also solved using the CGLS.

6. Numerical Experiments

In this section, we provide numerical experiments for several test problems to evaluate the feasibility and efficiency of the fixed-point-like method, called Algorithm 3, for the TVL2D2 problem. Performances of Algorithm 3 are evaluated by comparing its numerical results with those of Algorithms 1 and 4.

All numerical tests have been performed using Matlab R2016a on a personal computer equipped with Intel(R) Core(TM) i7-7700HQ CPU and 8.00GB RAM. For numerical 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 of size 9×9 are generated by the built-in Matlab functions fspecial('gaussian',[9,9],9), fspecial('average',9) and

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

respectively. The blurred and noisy image f is generated by

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

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

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

$$\frac{\|u_{k+1} - u_k\|_2}{\|u_{k+1}\|_2} < tol,$$

where tol represents a prescribed tolerance value. We set $tol = 5 \times 10^{-4}$ for Algorithms 1 and 3 and $tol = 2 \times 10^{-4}$ for Algorithm 4. For all test problems, an initial image was set to the observed blurred and noisy image f, we set $\kappa = 1 \times 10^{-6}$ and $maxit = 150$. For the CGLS, the tolerance number is set to 5×10^{-2} and the maximum number of iterations is set to 60. In Table 1, "Alg." represents the algorithm number to be used, Cam denotes the Cameraman image, PSNR_0 represents the PSNR values for the blurred and noisy image, "PSNR" represents the PSNR values for the restored image, "Iter" denotes the number of iterations required for Algorithms 1, 3 and 4, "α, β, γ" denote parameters which are chosen by numerical tries, and "Time" represents the elapsed CPU time in seconds.

In order to illustrate the efficiency of the proposed fixed-point-like method, we have used 5 test images such as the Cameraman, Lena, House, Boat and Caribou with pixel size 256×256. 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

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

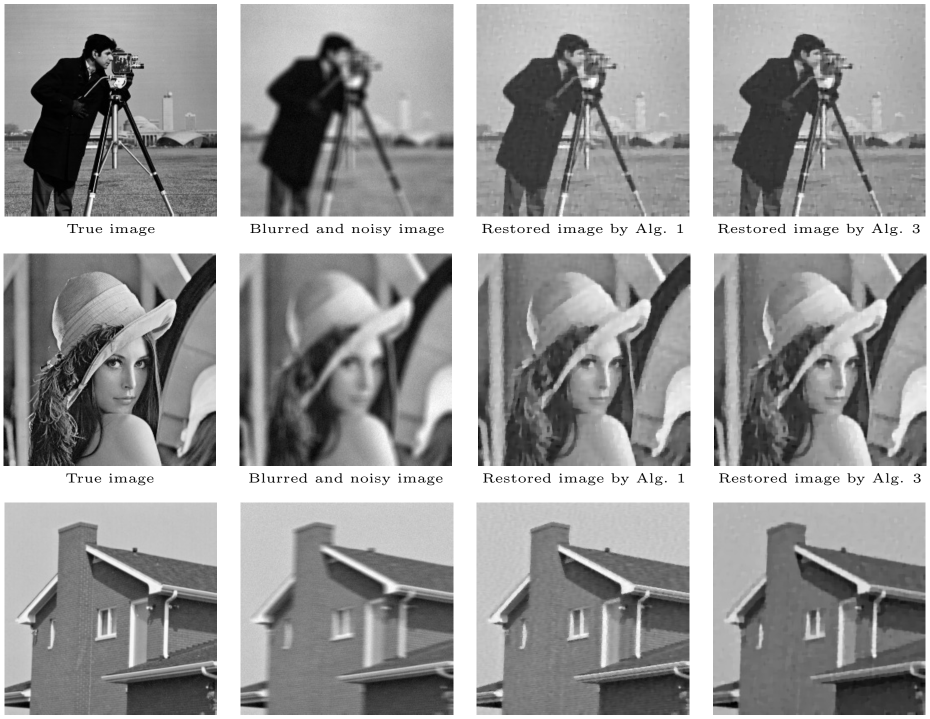
where $\|\cdot\|_F$ represents the Frobenius norm, $\tilde{U}$ denotes the restored image of size $m \times n$, and u_{ij} stands for the value of the original image U at a pixel point (i, j) for $1 \leq i \leq m$ and $1 \leq j \leq n$.

Table 1. Numerical results for Algorithms 1, 3 and 4

Image	PSF	PSNR$_0$	Alg.	α	β	γ	PSNR	Iter	Time
Cam	Gaussian	20.85	1	0.00160	0.120	0.00041	25.17	67	42.75
			3	0.00010	0.127	0.0063	25.53	18	6.47
			4	0.00008	0.130	0.0080	25.54	29	8.95
	Average	20.76	1	0.00160	0.130	0.00041	25.24	58	37.46
			3	0.00008	0.129	0.0071	25.61	19	6.47
			4	0.00010	0.126	0.0084	25.61	28	8.67
	Motion	21.85	1	0.00170	0.220	0.00049	28.05	51	11.26
			3	0.00001	0.229	0.0066	28.56	16	2.68
			4	0.00002	0.230	0.0160	28.53	25	2.77
Lena	Gaussian	22.55	1	0.00160	0.170	0.00040	26.17	69	45.16
			3	0.00070	0.139	0.0030	26.35	19	9.45
			4	0.00110	0.130	0.0080	26.31	19	6.50
	Average	22.44	1	0.00160	0.170	0.00040	26.20	69	46.72
			3	0.00050	0.159	0.0020	26.40	26	15.24
			4	0.00070	0.149	0.0090	26.38	20	6.62
	Motion	23.06	1	0.00170	0.240	0.00050	28.28	57	12.96
			3	0.00041	0.239	0.0142	28.48	12	1.53
			4	0.00020	0.260	0.0200	28.48	20	2.11
House	Gaussian	24.19	1	0.00200	0.200	0.00051	30.09	71	42.13
			3	0.00060	0.190	0.0180	30.57	14	3.31
			4	0.00080	0.190	0.0190	30.57	19	4.36
	Average	24.05	1	0.00200	0.200	0.00051	30.02	58	34.94
			3	0.00070	0.190	0.0173	30.55	13	3.10
			4	0.00070	0.190	0.0130	30.56	21	5.64
	Motion	27.01	1	0.00170	0.410	0.00051	32.87	68	14.02
			3	0.00230	0.400	0.0260	33.69	10	1.05
			4	0.00230	0.390	0.0320	33.67	15	1.39

All numerical results are provided in Table 1, and Figure 1 shows the restored images by Algorithms 1 and 3. As can be seen in Table 1, Algorithm 3 restores the true image better than Algorithm 1, and Algorithm 3 takes less CPU time than Algorithm 1. This means that the fixed-point-like method (i.e., Algorithm 3) for the TVL2D2 problem proposed in this paper performs better in both PSNR value and CPU time than the fixed-point method (i.e., Algorithm 1) for TVL2I2 problem. Also notice that the fixed-point-like method (i.e., Algorithm 3) for the TVL2D2 problem restores the true image as well as the split Bregman method (i.e., Algorithm 4) for the TVL2D2 problem, but Algorithm 3 takes less CPU time than Algorithm 4.

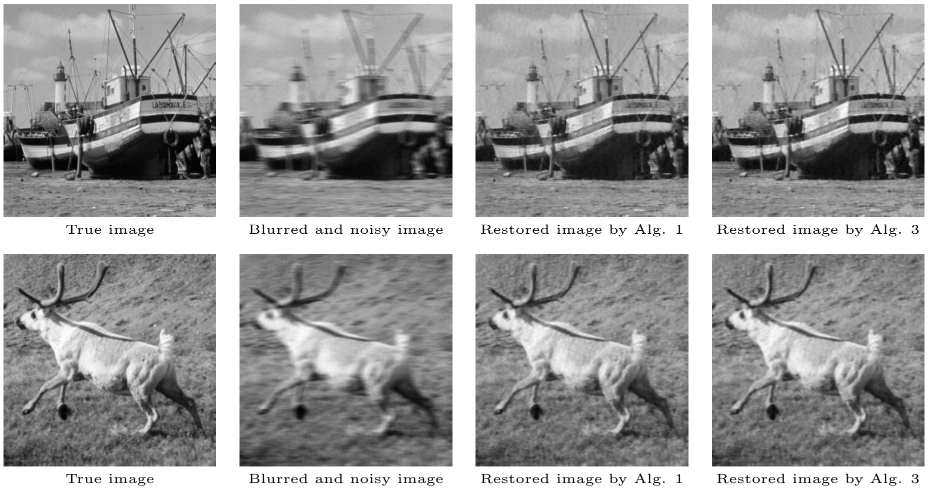
Image	PSF	PSNR$_0$	Alg.	α	β	γ	PSNR	Iter	Time
Boat	Gaussian	21.28	1	0.00160	0.120	0.00041	25.13	60	40.62
			3	0.00080	0.100	0.0069	25.34	11	3.90
			4	0.00090	0.100	0.0080	25.33	16	5.28
	Average	21.19	1	0.00160	0.120	0.00041	25.19	54	36.67
			3	0.00080	0.110	0.0100	25.41	11	3.38
			4	0.00090	0.100	0.0080	25.40	15	4.62
	Motion	23.32	1	0.00180	0.240	0.00052	27.70	54	12.07
			3	0.00100	0.200	0.0100	28.00	10	1.46
			4	0.00100	0.200	0.0130	27.99	15	1.93
Caribou	Gaussian	23.69	1	0.00170	0.145	0.00009	27.10	100	61.59
			3	0.00330	0.070	0.0063	27.40	7	2.57
			4	0.00300	0.090	0.0050	27.40	13	4.84
	Average	23.57	1	0.00200	0.145	0.00009	27.09	97	56.03
			3	0.00420	0.070	0.0080	27.40	7	2.14
			4	0.00200	0.100	0.0100	27.38	12	3.49
	Motion	25.64	1	0.00190	0.290	0.00011	29.79	90	17.32
			3	0.00480	0.115	0.0020	30.25	10	1.75
			4	0.00700	0.104	0.0007	30.20	23	4.18



True image Blurred and noisy image Restored image by Alg. 1 Restored image by Alg. 3

True image Blurred and noisy image Restored image by Alg. 1 Restored image by Alg. 3

True image Blurred and noisy image Restored image by Alg. 1 Restored image by Alg. 3



True image

Blurred and noisy image

Restored image by Alg. 1

Restored image by Alg. 3

True image

Blurred and noisy image

Restored image by Alg. 1

Restored image by Alg. 3

Fig. 1. Image restoration by Algorithms 1 and 3 (The first row contains Cameraman images for Gaussian blur, the second row contains Lena images for Average blur, the third row contains House images for Motion blur, the fourth row contains Boat images for Motion blur and the last row contains Caribou images for Motion blur).

7. Conclusion

In this paper, we have proposed the TVL2D2 regularization problem (4) for image restoration, and we developed the fixed-point-like method (i.e., Algorithm 3) for solving the TVL2D2 problem. According to numerical experiments for several test problems, the fixed-point-like method for the new proposed TVL2D2 problem performs better in both PSNR value and CPU time than the fixed-point method (i.e., Algorithm 1) for the existing TVL2I2 problem (3). As compared with the split Bregman method (i.e., Algorithm 4) for the TVL2D2 problem, Algorithm 3 takes less CPU time than Algorithm 4, and Algorithm 3 restores the true image as well as Algorithm 4. Hence, it can be concluded that the fixed-point-like method (i.e., Algorithm 3) for the TVL2D2 problem is preferred over the fixed-point method (i.e., Algorithm 1) for the TVL2I2 problem and the split Bregman method (i.e., Algorithm 4) for the TVL2D2 problem.

References

1. A. Beck and M. Teboulle, *Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems*, IEEE Transactions on Image Processing **18** (2009), 2419-2433.
2. A. Beck, *First-Order Methods in Optimization*, SIAM, Philadelphia, PA, 2017.
3. A. Bjorck, *Numerical methods for least squares problems*, SIAM, Philadelphia, PA, 1996.
4. J.F. Cai, S. Osher and Z. Shen, *Split Bregman Methods and Frame Based Image Restoration*, Multiscale Model. Simul. **8** (2009), 337-369.

5. R. Campagna, S. Crisci, S. Cuomo, L. Marcellino and G. Toraldo, *Modification of TV-ROF denoising model based on Split Bregman iterations*, Applied Mathematics and Computation **315** (2017), 453-467.
6. D.Q. Chen, H. Zhang and L.Z. Cheng, *A Fast Fixed Point Algorithm for Total Variation Deblurring and Segmentation*, Journal of Mathematical Imaging and Vision **43** (2012), 167-179.
7. T. Goldstein and S. Osher, *The Split Bregman Method for L1 Regularized Problems*, SIAM Journal on Imaging Sciences **2** (2009), 323-343.
8. C.T. Kelly, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, USA, 1995.
9. K.S. Kim and J.H. Yun, *Image Restoration Using a Fixed-Point Method for a TVL2 Regularization Problem*, Algorithms **13** (2020), 1-15.
10. Q. Li, C.A. Micchelli, L. Shen and Y. Xu, *A proximity algorithm accelerated by Gauss-Seidel iterations for L1/TV denoising models*, Inverse Problems **28** (2012).
11. X. Liu and L. Huang, *Split Bregman iteration algorithm for total bounded variation regularization based image deblurring*, Journal of Mathematical Analysis and Applications **372** (2010), 486-495.
12. J. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Am. Math. Soc. **73** (1967), 591-597.
13. L.I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D. **60** (1992), 259-268.
14. J.H. Yun, *Image Deblurring Using Relaxation Iterative Methods*, Journal of Algorithms and Computational Technology **13** (2019), 16 pages, Article ID 861732.
15. J.H. Yun, *Image denoising methods for new TVL1 models with impulse noise*, Int. J. Eng. Res. Tech. **13** (2020), 686-698.
16. J.H. Yun and H.J. Lim, *Image Restoration Using Fixed-Point-Like Methods for New TVL1 Variational Problems*, Electronics **9** (2020), 17 pages, Article ID 735.

Yu Jin Won received M.Sc. from Chungbuk National University. His research interests are numerical optimization and image processing.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 28644, Korea.
`e-mail: copysonic@chungbuk.ac.kr`

Jae Heon Yun received M.Sc. from Kyungpook National University, and Ph.D. from Iowa State University. He is currently a professor at Chungbuk National University since 1991. His research interests are scientific computing, iterative method and image processing.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 28644, Korea.
`e-mail: gmjae@chungbuk.ac.kr`