Hindawi
Journal of Applied Mathematics
Volume 2023, Article ID 6689311, 15 pages
https://doi.org/10.1155/2023/6689311

Research Article

Comparative Analysis of the Prox Penalty and Bregman Algorithms for Image Denoising

Soulef Bougueroua and Nourreddine Daili

Department of Mathematics, Faculty of Sciences, University F. ABBAS Setif 1, Setif 19000, Algeria

Correspondence should be addressed to Soulef Bougueroua; soulef.bougueroua@univ-setif.dz

Received 22 December 2022; Revised 14 April 2023; Accepted 26 April 2023; Published 20 October 2023

Academic Editor: Wei-Chiang Hong

Image restoration is an interesting ill-posed problem. It plays a critical role in the concept of image processing. We are looking for an image that is as near to the original as possible among images that have been skewed by Gaussian and additive noise. Image deconstruction is a technique for restoring a noisy image after it has been captured. The numerical results achieved by the prox-penalty method and the split Bregman algorithm for anisotropic and isotropic TV denoising problems in terms of image quality, convergence, and signal noise rate (SNR) are compared in this paper. It should be mentioned that isotropic TV denoising is faster than anisotropic. Experimental results indicate that the prox algorithm produces the best high-quality output (clean, not smooth, and textures are preserved). In particular, we obtained (21.4, 21) the SNR of the denoising image by the prox for sigma 0.08 and 0.501, such as we obtained (10.0884, 10.1155) the SNR of the denoising image by the anisotropic TV and the isotropic TV for sigma 0.08 and (-1.4635, -1.4733) for sigma 0.501.

1. Introduction

Image processing is a subset of signal processing that focuses on images and videos. All procedures done on an image in order to increase readability and facilitate interpretation are referred to as image processing. In the industry, image restoration is an important topic. This is a fundamental problem in a variety of applied sciences, including medical imaging [1] [2], microscopy and astronomy [3], film restoration, and image and video coding [4] [5]. Image restoration is an interesting subject in image processing since it occurs at the very beginning of the acquisition chain and involves recovering a clean original image from a degraded image [3] [2].

By applying a proximal algorithm to solve a minimization problem, we propose a unique technique for image restoration. It is believed that an original image has been deteriorated by additive noise.

We are attempting to rebuild u from the image we have seen Im (which is therefore a degraded version of the original image u). The maximum likelihood technique leads us to seek u as a solution to the following optimization problems, assuming that the additive noise is Gaussian:

$$(\mathscr{P}) \quad \alpha \coloneqq \arg \min_{u \in U^{ad}} \left\{ \frac{1}{2} \|\mathrm{Im} - u\|_2^2 \right\}, \tag{1}$$

where U^{ad} the admissible set is determined by

$$U^{ad} \coloneqq \{u \in BV(\Omega), J(u) \leq 0\}, \tag{2}$$

and $J(u)$ represents the total variation in u as specified by

$$J(u) \coloneqq \sup \left\{ \int_\Omega u(x) \ \mathrm{div}\ (\varphi(x)) dx : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_\infty \leq 1 \right\}. \tag{3}$$

Total variation regularization (TV) is a regularization term $J(u) = |D(u)|$ that allows for discontinuous solutions.

Because of its capacity to include "jumps" in the solution, it has become particularly popular for image rendering. The space of bounded variation function is known as BV. It is characterized by

$$\mathrm{BV}(\Omega) \coloneqq \left\{u \in L^1(\Omega), J(u) < +\infty\right\}. \tag{4}$$

To tackle this problem, we will apply the proximal penalty approach (Aujol [6]; Micchelliy, Shenz, and Xux [7]). To accomplish this, we state the method's idea. The fact that the problem $(\mathscr{P})$ is equivalent to the problem $(\mathscr{P}_{\alpha_e})$ is the principle behind it.

$$\begin{aligned}&(\mathscr{P}) \quad \alpha \coloneqq \arg\min_{u\in U^{ad}}\left\{\frac{1}{2}\|\mathrm{Im}-u\|_2^2\right\}\\ &\Leftrightarrow (\mathscr{P}_{\alpha_e}) \quad \alpha_e \coloneqq \arg\min_{u\in BV}\left\{\frac{1}{2}\|\mathrm{Im}-u\|_2^2+\Psi_{U^{ad}}(u)\right\},\end{aligned} \tag{5}$$

where $\Psi_{U^{ad}}$ indicates the U^{ad} indicator function defined by

$$\Psi_{U^{ad}}(u) \coloneqq \begin{cases} 0 & \text{if } u \in U^{ad},\\ +\infty & \text{otherwise.}\end{cases} \tag{6}$$

Definition 1. [7] We denote by $\mathbb{R}^d$ the usual d-dimensional Euclidean space.

Let ψ be a real-valued convex function on $\mathbb{R}^d$. For all $x \in \mathbb{R}^d$, the proximal operator of ψ is defined by

$$\mathrm{prox}(x) \coloneqq \mathrm{argmin}\left\{\frac{1}{2}\|u-x\|_2^2+\psi(u)\colon u\in\mathbb{R}^d\right\}. \tag{7}$$

We give the following three examples.

Example 1. If $\lambda > 0$ and $x \in \mathbb{R}$, then

$$\mathrm{prox}_{1/\lambda|.|}(x) = \max\left(|x|-\frac{1}{\lambda},0\right)\mathrm{sing}(x). \tag{8}$$

Example 2. If $\lambda > 0$ and $x \in \mathbb{R}^m$, then

$$\mathrm{prox}_{1/\lambda\|.\|_1}(x) = \left[prox_{1/\lambda|.|}(x_1), \mathrm{prox}_{1/\lambda|.|}(x_2), \cdots, \mathrm{prox}_{1/\lambda|.|}(x_m)\right]^t, \tag{9}$$

with t is the transpose of a line vector.

Example 3. If $\lambda > 0$ and $x \in \mathbb{R}^m$, then

$$\begin{aligned}\mathrm{prox}_{1/\lambda\|.\|_2}(x) &= \max\left(\|x\|_2-\frac{1}{\lambda},0\right)\frac{x}{\|x\|_2}\\ &= \mathrm{prox}_{1/\lambda|.|}(\|x\|_2)\frac{x}{\|x\|_2}.\end{aligned} \tag{10}$$

2. General Principle of the Proximal-Penalty Methods

According to [8–12], the following is the general principle of the proximal-penalty methods:

(1) Replace the problem $(\mathscr{P})$ with the problem $(\mathscr{P}_r)$ that has no constraints

$$(\mathscr{P}_r) \quad \alpha_r \coloneqq \arg\min_{u\in BV}\left\{\varphi(u,r)=\frac{1}{2}\|\mathrm{Im}-u\|_2^2+r.\max\left(0,\Pi_K(u)\right)\right\}. \tag{11}$$

The penalty coefficient is r whith $r > 0$. The external penalty function is known as h where $h(u) = \max(0, \Pi_K(u))$, and $\Pi_K(u)$ is the projection of u in K where

$$K \coloneqq \left\{\mathrm{div}\left(\varphi(x)\colon \varphi\in C_c^1\left(\Omega,\mathbb{R}^2\right), \|\varphi\|_\infty \le 1\right\}. \tag{12}$$

When $(r \longrightarrow +\infty)$, the obtained solution $u(r)$ is a solution of $(\mathscr{P})$.

(2) We begin by selecting a penalty coefficient r_1 and then solve the problem without constraints

$$\left(\mathscr{P}_{r_1}\right) \quad \alpha_{r_1} \coloneqq \arg\min_{u\in BV}\left\{\varphi(u,r_1)=\frac{1}{2}\|\mathrm{Im}-u\|_2^2+r_1.\max\left(0,\Pi_K(u)\right)\right\}, \tag{13}$$

let $u(r_1)$ be the obtained point.

(3) The stop test: $u(r_1)$ is a good approximation of the optimum if the amount $r_1 h(u(r_1))$ is sufficiently small; otherwise, a penalty coefficient $r_2 > r_1$ will be determined and the following new problem will be solved without constraint $(\mathscr{P}_{r_2})$

The problem is linked to

$$\begin{aligned}(\mathscr{P}_r)_w \quad \alpha_r(w) \coloneqq \arg\min_{(u,w)\in BV^2}\Big\{&\frac{1}{2}\|\mathrm{Im}-u\|_2^2\\ &+r.\max\left(0,\Pi_K(u)\right)+\frac{1}{2}\|u-w\|_2^2\Big\}.\end{aligned} \tag{14}$$

The relaxation algorithm transforms and generates a sequence $\{u^k, w^k\}_k$ such that u^{k+1} is a solution to the problem when we applied to this problem.

$$\begin{aligned}(\mathscr{P}_r)_{w^k} \quad \alpha_r\left(w^k\right) \coloneqq \arg\min_{u\in BV}\Big\{&\frac{1}{2}\|\mathrm{Im}-u\|_2^2\\ &+r.\max\left(0,\Pi_K(u)\right)+\frac{1}{2}\left\|u-w^k\right\|_2^2\Big\},\end{aligned} \tag{15}$$

Step 0: $(k=0)$ Let $u^0 \in \mathbb{R}^n$, $\varepsilon > 0$ be a precision.
Step 1: we use the minimization approach to find a solution u^1 to the following problem using a penalty coefficient r^0 and a precision $\delta > 0$.

$$\arg \min_{u\in\mathbb{R}^n}\{1/2\|\mathrm{Im}-u\|_2^2 + r_0.\max(0, \Pi_K(u)) + 1/2\|u-u^0\|_2^2\},$$

Step 2: the solution obtained is $u^1(r_0) = u^1$.
If $\|u^1 - u^0\|_2^2 < \varepsilon$ and if $r_0 h(u^1(r_0)) < \delta$, then u^1 is an excellent approximation of the optimal solution, and the calculations stop at iteration $k+1$.
Otherwise, we use $r_1 > r_0$ as a penalty coefficient. We put $r_0 = r_1$, $u^0 = u^1$, and $k = k+1$ and return to step 1.

Algorithm 1: [11, 12] Algorithm of proximal penalty.

and the solution of the problem is w^{k+1}

$$(\mathscr{P}_r)_{u^{k+1}} \quad \alpha_r(w) \coloneqq \arg\min_{u\in BV}\left\{\frac{1}{2}\|\mathrm{Im}-u\|_2^2 + r.\max\left(0, \Pi_K\left(u^{k+1}\right)\right) + \frac{1}{2}\left\|u^{k+1}-w\right\|_2^2\right\}. \tag{16}$$

As a result, a simple iteration u^{k+1} solves the following problem:

$$(\mathscr{P}_r)_{u^k} \quad \alpha_r \coloneqq \arg\min_{u\in BV}\left\{\frac{1}{2}\|\mathrm{Im}-u\|_2^2 + r.\max(0, \Pi_K(u)) + \frac{1}{2}\left\|u-u^k\right\|_2^2\right\}. \tag{17}$$

3. Bregman Algorithms and Imaging

3.1. Bregman Projection

Definition 2 (Bregman distance). Let X be a Banach space, $g : X \longrightarrow]-\infty,+\infty[$ be a lower semicontinuous proper convex function, and let $C \subset \mathrm{int}\,(\mathrm{dom}\,(g))$ be a nonempty closed convex set. Suppose g is Gâteaux differentiable in $\mathrm{int}\,(\mathrm{dom}\,(g))$ with its Gâteaux derivative denoted by ∇g. The Bregman distance D_g associated with g is a function defined as follows:

$$\begin{aligned} &D_g : X \times \mathrm{int}\,(\mathrm{dom}\,(g)) \longrightarrow [0,+\infty], \\ &(y,x) \longrightarrow D_g(y,x) = g(y) - g(x) - \langle \nabla g(x), y-x\rangle, \end{aligned} \tag{18}$$

with $\mathrm{int}\,(\mathrm{dom}\,(g))$ is the interior of the domain g.

Let

$$C_1 \coloneqq \mathbb{R}_+^n = \{x \in \mathbb{R}^n, x_i \geq 0, i = 1 \cdots n\}, g_1(x) \coloneqq \sum_{i=1}^n x_i \log x_i - x_i. \tag{19}$$

Then, there is a K-L divergence (Kullback-Leibler divergence).

$$D_{g_1}(y,x) \coloneqq \sum_{i=1}^n y_i \log \frac{y_i}{x_i} + x_i - y_i \tag{20}$$

is a Bregman distance.

Remark 1. The following are some historical notes:

(i) Bregman was the first to adopt this distance measurement in 1967 [13]

(ii) Censor and Lent created and developed the concept [14]

(iii) In 1976, Bregman devised a simple and effective method for using the D_g function in the design and analysis of feasibility and optimization algorithms. This has spawned a burgeoning field of research in which Bregman's technique is used to create and analyze iterative algorithms for nonlinear applications, not only to address feasibility and optimization problems but also to solve variational inequalities and calculate fixed points. More information can be found in the sources [15] [16] [17].

Definition 3 (Bregman projection). Let $g : X \longrightarrow]-\infty,+\infty[$ be a lower semicontinuous smooth convex function. Let C be a closed convex set in X with $C \cap \mathrm{int}\,(\mathrm{dom}\,(g)) \neq \varnothing$. The approximation problem $\underset{y\in C}{\mathrm{argmin}} D_g(y,x)$ thus permits a single solution $\mathrm{proj}_C^g(x) \in \mathrm{int}\,(\mathrm{dom}\,(g))$, known as the Bregman projection of x over C defined by

$$\mathrm{proj}_C^g(x) \coloneqq \underset{y\in C}{\mathrm{argmin}} D_g(y,x). \tag{21}$$

3.2. Bregman Algorithm. The iterative technique of Bregman was first introduced and studied in the field of image processing by Osher et al. Osher et al. proposed the iterative Bregman algorithm as an effective algorithm for solving optimization problems in [18]. Their main idea was to first transform a constraint optimization problem into a constraint-free problem by using the Bregman distance. This problem-solving algorithm is as follows:

$$\min_u\{z(u) + H(u,f)\}, \tag{22}$$

Initialization: $k = 0$, $u_0 = 0$ and $p_0 = 0$.
While "u^k not converge," do,
$u^{k+1} \longleftarrow \operatorname{argmin} D_z^{p^k}(u, u^k) + H(u)$.
$p^{k+1} \longleftarrow p^k - \nabla H(u^{k+1}) \in \partial z(u^{k+1})$.
$k \longleftarrow k + 1$.
End while.

Algorithm 2: [18] Algorithm iteratives of Bregman.

such that $z : X \longrightarrow \mathbb{R}$ and $H : X \longrightarrow \mathbb{R}$ are nonnegative convex functions of $u \in X$, and $H(u, f)$ is a smooth nonnegative convex function in relation to u for a given f, and X is a closed convex set.

The Bregman iterative algorithm is defined as follows by Osher et al. in [18].

3.3. The Convergence Theorem. In [18], the variant of Bregman was presented for TV-based image rendering. Other features of this iterative Bregman scheme, as well as the convergence analysis, have been proven in detail in [18–20]. $u^1 = \min_u(z(u) + H(u))$ is the first iteration of this method. The residual term must be minimal to solve the initial problem; once the residual term converges, the Bregman iterative algorithm continues. Because of its excellent convergence features, the Bregman iterative algorithm has been applied to a variety of problems, including badly posed problems and image dissection. The following are some of these qualities: with noisy data, we can achieve convergence to the original image we are seeking to recover, as well as convergence in terms of Bregman distance to the original image and a monotonous decline in the residual term. We have $z(u) = \|u\|_{BV}$ where $\|\nabla u\|_1$ and $H(u) = 1/2\|u - f\|_2^2$ learned a lot about image redaction during our research.

The technique generates a series that reduces H in a monotonous manner.

Proposition 1. *[19] There exists a monotonous decrease H*

$$H\left(u^{k+1}\right) \le H\left(u^{k+1}\right) + D_z^{p^k}\left(u^{k+1}, u^k\right) \le H\left(u^k\right). \tag{23}$$

Proposition 2. *[19] The remaining residual terms* $\{H(u^k)\}$ *converge to the smallest value of H.*

If $H : X \longrightarrow \mathbb{R}$ *and* $z(\tilde{u}) < \infty$ *are minimized by ũ, then*

$$H\left(u^k\right) \le H(\tilde{u}) + \frac{z(\tilde{u})}{k}. \tag{24}$$

3.4. The Split Bregman Algorithm. Goldstein and Osher first proposed the split Bregman algorithm in [21] to handle more general form optimization problems:

$$\min_{u \in X} \left\{H(u) + \|\Phi(u)\|_1\right\}, \tag{25}$$

where X is a closed convex set and $\Phi : X \longrightarrow \mathbb{R}$ and $H : X \longrightarrow \mathbb{R}$ are the convex functions. This problem is the same

Initialization: $k = 0$, $u^0 = 0$, $b^0 = 0$.
While $\|u^k - u^{k-1}\| > \text{tol}$ do,
$u^{k+1} = \min_u H(u) + (\lambda/2)\|d^k - \Phi(u) - b^k\|_2^2$.
$d^{k+1} = \min_d |d| + (\lambda/2)\|d - \Phi(u^{k+1}) - b^k\|_2^2$.
$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1})$.
$k = k + 1$.
End while.

Algorithm 3: [19] The split Bregman algorithm.

as the stress minimization problem as follows:

$$\min_{u \in X, d \in \mathbb{R}} \left\{H(u) + \|d\|_1\right\}, \text{ such that } d = \Phi(u). \tag{26}$$

Goldstein and Osher introduced the split Bregman algorithm, which was written as follows:

The split Bregman algorithm is used to solve some of the most common form optimization problems:

$$\min_{u \in X} \left\{z(u) + \frac{1}{2}\|u - f\|_2^2\right\}. \tag{27}$$

Anisotropic and isotropic TV denoising problems are solved using the split Bregman method.

3.4.1. Anisotropic TV Denoising Problem. The problem of anisotropic TV denoising is considered in [19].

$$(\mathscr{P}_1) \quad \min_u \left\{\left\|\frac{\partial u}{\partial x}\right\|_1 + \left\|\frac{\partial u}{\partial y}\right\|_1 + \frac{\mu}{2}\|u - f\|_2^2\right\}, \tag{28}$$

where f is the noisy image, $\partial u/\partial x$ and $\partial u/\partial y$ will be noted by u_x and u_y, respectively. The problem is solved using a constraint equivalent to a problem $(\mathscr{P}_1)$.

We answer the problem $(\mathscr{P}_2)$ as follows:

$$(\mathscr{P}_2) \quad \begin{cases} \min_u \|d_x\|_1 + \|d_y\|_1 + \dfrac{\mu}{2}\|u - f\|_2^2 \\ \text{subject to} \quad d_x = u_x, d_y = u_y. \end{cases} \tag{29}$$

The split Bregman algorithm can be used to tackle this last problem:

$$(\mathscr{P}_3) \quad \min_{u, d_x, d_y} \left\{\|d_x\|_1 + \|d_y\|_1 + \frac{\mu}{2}\|u - f\|_2^2 + \frac{\lambda}{2}\|d_x - u_x\|_2^2 + \frac{\lambda}{2}\|d_y - u_y\|_2^2\right\}. \tag{30}$$

We use

$$\text{shrink}(x, a) = \begin{cases} x - a & \text{if } x > a, \\ x + a & \text{if } x < -a, \\ 0 & \text{else.} \end{cases} \tag{31}$$

Initialization: $k = 0$, $u^0 = 0$, $b^0 = 0$.
While $\|u^k - u^{k-1}\| > \text{tol}$ do,
$u^{k+1} = G^k$, where G is the Gauss-Seidel function.
$d_x^{k+1} = shrink\ (\nabla_x u^{k+1} + b_x^k, (1/\lambda))$.
$d_y^{k+1} = shrink\ (\nabla_y u^{k+1} + b_y^k, (1/\lambda))$.
$b_x^{k+1} = b_x^k(\nabla_x u^{k+1} - d_x^{k+1})$.
$b_y^{k+1} = b_y^k(\nabla_y u^{k+1} - d_y^{k+1})$.
$k = k + 1$.
End while.

Algorithm 4: [19] The split Bregman algorithm of anisotropic TV denoising.

The Gauss-Seidel function is also useful.

$$G_{i,j}^k = \frac{\lambda}{\mu + 4\lambda}\left(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k + d_{x,i-1,j}^k + d_{x,i,j}^k + d_{y,i,j-1}^k + d_{y,i,j}^k + b_{x,i-1,j}^k + b_{x,i,j}^k + b_{y,i,j-1}^k + b_{y,i,j}^k\right) + \frac{\mu}{\mu + 4\lambda} f_{i,j}. \tag{32}$$

3.4.2. Isotropic TV Denoising Problem. The problem of isotropic TV denoising is considered in [19].

$$\left(\mathscr{P}_1'\right) \quad \min_u \left\{\|\nabla u\|_2 + \frac{\mu}{2}\|u - f\|_2^2\right\}. \tag{33}$$

The problem $(\mathscr{P}_1')$ is solved using a constraint equivalent problem $(\mathscr{P}_2')$:

$$\left(\mathscr{P}_2'\right) \quad \begin{cases} \min_u \left\|\left(d_x, d_y\right)\right\|_2 + \dfrac{\mu}{2}\|u - f\|_2^2 \\ \text{subject to} \quad d_x = u_x, d_y = u_y. \end{cases} \tag{34}$$

To solve the problem $(\mathscr{P}_2')$, we solve the following problem without constraint:

$$\left(\mathscr{P}_3'\right) \quad \min_{u,d_x,d_y}\left\{\left\|\left(d_x, d_y\right)\right\|_2 + \frac{\mu}{2}\|u - f\|_2^2 + \frac{\lambda}{2}\|d_x - u_x\|_2^2 + \frac{\lambda}{2}\left\|d_y - u_y\right\|_2^2\right\}. \tag{35}$$

The split Bregman algorithm can be used to tackle this last difficulty.

We give the following definitions:

$$s^k = \sqrt{\left|u_x^k - b_x^k\right|^2 + \left|u_y^k - b_y^k\right|^2}. \tag{36}$$

3.4.3. Combining Anisotropic and Isotropic TV Denoising Problems. We propose an image-denoising method by combining the anisotropic and isotropic TV denoising problem.

Initialization: $k = 0$, $u^0 = 0$, $b^0 = 0$.
While $\|u^k - u^{k+1}\| > \text{tol}$ do,
$u^{k+1} = G^k$, where G is the Gauss-Seidel function.
$d_x^{k+1} = s^k\lambda(u_x^k + b_x^k)/s^k\lambda + 1$.
$d_y^{k+1} = s^k\lambda(u_y^k + b_y^k)/s^k\lambda + 1$.
$b_x^{k+1} = b_x^k + (u_x^{k+1} - d_x^{k+1})$.
$b_y^{k+1} = b_y^k + (u_y^{k+1} - b_y^{k+1})$.
$k = k + 1$.
End while.

Algorithm 5: [19] The split Bregman algorithm of isotropic TV denoising.

$$(\mathscr{P}_{A+I}) \quad \min_u \left\{\left\|\frac{\partial u}{\partial x}\right\|_1 + \left\|\frac{\partial u}{\partial y}\right\|_1 + \|\nabla u\|_2 + \frac{\mu}{2}\|u - f\|_2^2\right\}. \tag{37}$$

The problem $(\mathscr{P}_{A+I})$ is solved using a constraint equivalent to a problem $(\mathscr{P}_{A+I}')$:

$$\left(\mathscr{P}_{A+I}'\right) \quad \begin{cases} \min_u \|d_x\|_1 + \left\|d_y\right\|_1 + \left\|\left(d_x, d_y\right)\right\|_2 + \dfrac{\mu}{2}\|u - f\|_2^2 \\ \text{subject to} \quad d_x = u_x, d_y = u_y. \end{cases} \tag{38}$$

To solve the problem $(\mathscr{P}_{A+I}')$, we solve the following problem without constraint:

$$\left(\mathscr{P}_{A+I}'{}'\right) \quad \min_{u,d_x,d_y}\left\{\|d_x\|_1 + \left\|d_y\right\|_1 + \left\|\left(d_x, d_y\right)\right\|_2 + \frac{\mu}{2}\|u - f\|_2^2 + \frac{\lambda}{2}\|d_x - u_x\|_2^2 + \frac{\lambda}{2}\left\|d_y - u_y\right\|_2^2\right\}. \tag{39}$$

In our next studies, we would like to program the method combining anisotropic and isotropic TV denoising in MATLAB and we compare denoising methods.

3.5. The Convergence Theorem

3.5.1. Anisotropic TV Denoising Algorithm. The following relationship is defined based on the various algorithms of split Bregman anisotropic [7, 19].

$$d^{k+1} = \text{prox}_{\mu/\lambda\|\cdot\|_1}\left(Bx^{k+1} + b^k\right) = \begin{cases} d_x^{k+1} = \text{shrink}\left(u_x^{k+1} + b_x^k, \dfrac{1}{\lambda}\right), \\ d_y^{k+1} = \text{shrink}\left(u_y^{k+1} + b_y^k, \dfrac{1}{\lambda}\right), \end{cases}$$

$$b^{k+1} = b^k + \left(Bx^{k+1} - d^{k+1}\right) = \begin{cases} b_x^{k+1} = b_x^k + u_x^{k+1} - d_x^{k+1} \\ b_y^{k+1} = b_y^k + u_y^{k+1} - d_y^{k+1} \end{cases}, \tag{40}$$

with

$$Bx^{k+1} = \nabla u^{k+1}. \tag{41}$$

$2N^2 \times N$ is the size of the matrix B which is defined by

$$B \coloneqq \begin{bmatrix} I_N \otimes D \\ D \otimes I_N, \end{bmatrix}. \tag{42}$$

with I_N matrix identity $N \times N$, $P \otimes Q$ the Kronecker product of matrices P and Q, and D is a matrix $N \times N$ defined by

$$D \coloneqq \begin{bmatrix} 0 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}. \tag{43}$$

Proposition 3. *If $\lambda, \mu > 0$ and $c, x^0 \in \mathbb{R}^m$ as a result, the iteration scheme*

$$x^{k+1} = \left(I - prox_{\mu/\lambda\|\cdot\|_1}\right)\left(x^k + c\right), \quad k = 0, 1, \cdots \tag{44}$$

converges towards its limit in a finite number of steps, for $i = 1, 2, \cdots, m$,

$$\lim_{k \longrightarrow \infty}\left(x^k\right)_i = \begin{cases} \dfrac{\mu \text{ sign } (c_i)}{\lambda}, & c_i \neq 0, \\ \left(I - prox_{\mu/\lambda|\cdot|}\right)\left(x^0\right)_i, & c_i = 0, \end{cases} \tag{45}$$

Proof. In the case $m = 1$, this is the result. The evidence is divided into three categories: $c > 0$, $c < 0$, and $c = 0$.

In the case where $c > 0$, we have

$$\frac{\mu}{\lambda} = \left(I - \text{prox}_{\mu/\lambda|\cdot|}\right)\left(\frac{\mu}{\lambda} + c\right). \tag{46}$$

To put it another way, the number μ/λ represents a fixed point in the iterative system. If $x^0 + c > \mu/\lambda$ is true, then $x^1 = \mu/\lambda$ for all $k \geq 1$. In other words, one iteration is sufficient to approach the iterative scheme's limit. If $|x^0 + c| \leq \mu/\lambda$, we have:

$$x^k = \begin{cases} x^0 + kc \text{ if } 0 \leq k < \left[\dfrac{\mu}{\lambda c} - \dfrac{x^0}{c}\right], \\ x^k = \dfrac{\mu}{\lambda} \text{ if } k \geq \left[\dfrac{\mu}{\lambda c} - \dfrac{x^0}{c}\right], \end{cases} \tag{47}$$

The lowest integer that surpasses v is represented by $[v]$.

In step $[\mu/\mu - x^0/c]$, the iterative system finds its limit. If $x^0 + c < -\mu/\lambda$ is true, then $x^1 = -\mu/\lambda$.

In step $[2\mu/\lambda c] + 1$, the scheme reaches its limit. □

Case $c < 0$ is comparable to that of case $c > 0$. The iterative pattern, in particular, converges in a finite number of steps to the limit $-\mu/\lambda$.

Finally, we will look at example $c = 0$. The iterative pattern becomes $x^{k+1} = (I - \text{prox}_{\mu/\lambda|\cdot|})(x^k)$ in this example.

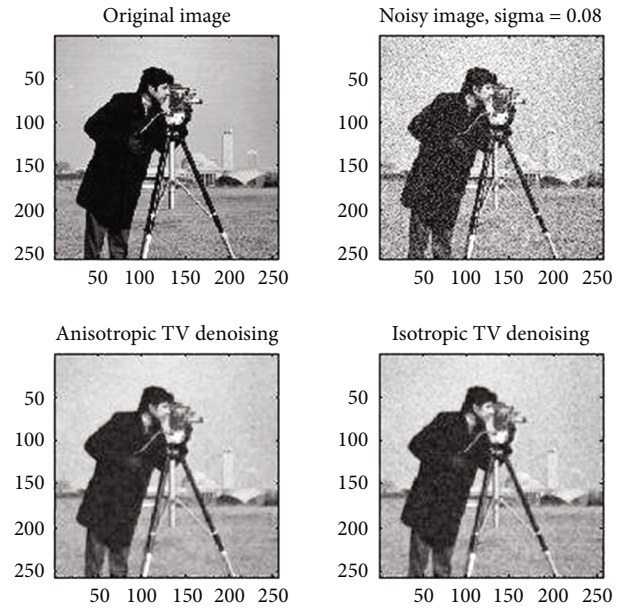

Figure 1: Split Bregman results using cameraman image.

Table 1: Results for the anisotropic TV denoising algorithm, sigma = 0.08.

Images	Size $n \times m$	Number of iterations	Relative error	Time (s)
Cameraman	256×256	49	0.16604	34.527229
Barbara	510×510	41	0.173054	85.609672
Flower	256×256	51	0.211968	24.340052
Girl	216×233	47	0.149544	17.045064
Iline	1961×3553	164	0.143347	10438.736396
University	480×640	141	0.143296	300.869216

Table 2: Results for the isotropic TV denoising algorithm, sigma = 0.08.

Images	Size $n \times m$	Number of iterations	Relative error	Time (s)
Cameraman	256×256	143	0.167415	63.170180
Barbara	510×510	137	0.17464	182.507988
Flower	256×256	22	0.210941	41.570893
Girl	216×233	24	0.151122	18.938243
Iline	1961×3553	15	0.143349	771.958677
University	480×640	24	0.143312	120.298105

Table 3: Performance metrics for the anisotropic TV denoising algorithm, sigma = 0.08.

Images	MSE	SNR	PSNR	IQI
NK	AD	SC	MD	NAE
Cameraman	$1.1758e+04$	12.4348	7.4276	$-7.3644e-05$
1.3689	-73.9475	0.4184	228	0.7373
Barbara	$1.4142e+04$	11.8896	6.6257	$-3.5635e-06$
1.3828	-77.6333	0.3766	234	0.8772
Flower	$1.6863e+04$	10.0637	5.8614	$5.5853e-06$
1.5642	-88.7960	0.2694	255	1.1090
Girl	$1.3374e+04$	12.9140	6.8682	$8.4450e-08$
1.3312	-75.0687	0.4254	221	0.7501
Iline	232.5184	12.9318	24.4662	0.0052
1.0129	-5.3346	0.9706	255	0.0268
University	$1.4088e+04$	13.1774	6.6423	$2.3093e-04$
1.1405	-61.5217	0.4907	255	0.7909

For every $k \geq 1$, we can see that $x^k = \mu/\lambda$ if $x^0 \geq \mu/\lambda$; $x^0 \leq -\mu/\lambda$ if $x^0 = \mu/\lambda$; $x^k = x^0$ if $|x^0| \leq \mu/\lambda$. As a result, the iteration to an end in a single step.

The convergence of the split Bregman anisotropic TV denoising algorithm was proved in [22]. As can be seen, we must solve a linear system according to the Goldstein-Osher split Bregman denoising.

The split Bregman denoising method is reduced to the Jia-Zhao denoising algorithm [23] as a result of this adjustment, which can be called the FP2O-ATV algorithm for $k = 0$.

In addition, both the Goldstein-Osher split Bregman algorithm and the Jia-Zhao denoising method make substantial use of the Bregman distance features and show signs of convergence.

In particular, the parameter λ that guarantees convergence of the Jia-Zhao denoising algorithm must be less than 1/8, whereas for the algorithm FP2O-ATV to converge, it is relaxed to a number less than $1/4 \sin^{-2(N-1)\pi/2N}$, which is somewhat higher than 1/4.

3.5.2. Isotropic TV Denoising Algorithm. The following relationship is defined based on the various algorithms of the split Bregman isotropic [7, 19].

Table 4: Performance metrics for the isotropic TV denoising algorithm, sigma = 0.08.

Images	MSE	SNR	PSNR	IQI
NK	AD	SC	MD	NAE
Cameraman	$1.1626e+04$	12.4178	7.4764	$-7.4216e-05$
1.3804	-76.0561	0.4157	228	0.7384
Barbara	$1.3937e+04$	11.8882	6.6890	$-3.3960e-06$
1.3831	-77.6695	0.3784	242	0.8666
Flower	$1.6711e+04$	10.1155	5.9007	$5.1306e-06$
1.5626	-88.6839	0.2706	255	1.0941
Girl	$1.3120e+04$	12.9056	6.9516	$4.4023e-07$
1.3339	-75.3454	0.4267	221	0.7382
Iline	210.5850	12.9318	24.8965	0.0052
1.0122	-5.0294	0.9724	255	0.0250
University	$1.3858e+04$	13.1735	6.7138	$2.3140e-04$
1.1553	-63.5094	0.4866	255	0.7864

$$d^{k+1} = \text{prox}_{1/\lambda\varphi}\left(Bx^{k+1} + b^k\right) = \begin{cases} d_x^{k+1} = \dfrac{s^k\lambda\left(u_x^k + b_x^k\right)}{s^k\lambda + 1}, \\ d_y^{k+1} = \dfrac{s^k\lambda\left(u_y^k + b_y^k\right)}{s^k\lambda + 1}, \end{cases}$$
$$b^{k+1} = b^k + \left(Bx^{k+1} - d^{k+1}\right) = \begin{cases} b_x^{k+1} = b_x^k + u_x^{k+1} - d_x^{k+1} \\ b_y^{k+1} = b_y^k + u_y^{k+1} - d_y^{k+1} \end{cases}. \tag{48}$$

Proposition 4. *If $\lambda, \mu > 0$ and $c, x^0 \in \mathbb{R}^m$ as a result, the iterative scheme*

$$x^{k+1} = \left(I - prox_{\mu/\lambda\|\cdot\|_2}\right)\left(x^k + c\right), \quad k = 0, 1, \cdots \tag{49}$$

converges towards its limit in a finite number of steps, for $i = 1, 2$

$$\lim_{k\longrightarrow\infty} x^k = \begin{cases} x^0 - \max\left(\left\|x^0\right\|_2 - \dfrac{\mu}{\lambda}, 0\right)\dfrac{x^0}{\|x^0\|_2}, & c = 0, \\ \dfrac{\mu c}{\lambda\|c\|_2}, & c \neq 0. \end{cases} \tag{50}$$

Proof. We will look at the scenario where $c = 0$. In this situation, the equation is reduced to $x^{k+1} = (I - \text{prox}_{\mu/\lambda\|\cdot\|_2})(x^k)$. If $\|x^0\|_2 \geq \mu/\lambda$, we have $x^k = \mu x^0/\lambda\|x^0\|_2$ for all $k \geq 1$ and $x^k = x^0$ for all $k \geq 0$ if $\|x^0\|_2 < \mu/\lambda$, as a result

$$\lim_{k\longrightarrow\infty} x^k = x^0 - \max\left(\left\|x^0\right\|_2 - \frac{\mu}{\lambda}, 0\right)\frac{x^0}{\|x^0\|_2}. \tag{51}$$

Table 5: Results for image flower the anisotropic TV denoising algorithm.

Sigma	SNR_TV_AS	Number of iterations	Relative error	Time (s)
0.08	10.0884	51	0.210941	73.156270
0.15	7.1100	47	0.393911	57.259335
0.25	3.6790	46	0.651452	78.888763
0.35	1.1272	68	0.905142	73.037620
0.501	-1.4576	182	1.24135	166.640882

Table 6: Results for image flower the isotropic TV denoising algorithm.

Sigma	SNR_TV_IS	Number of iterations	Relative error	Time (s)
0.08	10.1155	22	0.210019	41.570893
0.15	7.0932	21	0.394318	22.865811
0.25	3.6670	20	0.652357	37.383759
0.35	1.1227	19	0.905396	15.270150
0.501	-1.4507	18	1.24031	14.801319

Table 7: The different values of SNR of a denoising image by prox, the anisotropic TV, and the isotropic TV.

Sigma	SNR_Prox	SNR_TV_AS	SNR_TV_IS
0.08	21.4	10.0884	10.1155
0.15	25.1	7.1100	7.0932
0.25	20.4	3.6790	3.6670
0.35	21.9	1.1272	1.1227
0.501	21	-1.4635	-1.4733

In the case of $c \neq 0$, we have $x^{\infty} = (I - \text{prox}_{\mu/\lambda\|.\|_2})(x^{\infty} + c)$, i.e., $c = \text{prox}_{\mu/\lambda\|.\|_2}(x^{\infty} + c)$, where x^{∞} is the limit of the Picard iterations. If $\Psi = \mu/\lambda\|.\|_2$ is a convex function on $\mathbb{R}^d$ with $x \in \mathbb{R}^d$ then,

$$c \in \partial_{\mu/\lambda\|.\|_2}(x) \Leftrightarrow x = \text{prox}_{\mu/\lambda\|.\|_2}(x + c); \tag{52}$$

and by Example 3 in the proximal operator, resulting in $x^{\infty} = \mu c/\lambda\|c\|_2$.

The convergence of the split Bregman isotropic TV denoising algorithm has been proved in [22, 24]. The linear system

$$\left(I + \lambda B^t B\right)x^{k+1} = x - \lambda B^t\left(b^k - d^k\right) \tag{53}$$

must be resolved at each iteration of the split Bregman isotropic TV denoising algorithm, just as it must be resolved at each iteration of the split Bregman anisotropic TV denoising method. To reach an acceptable approximation of x^{k+1} [21], another Gauss-Seidel iteration step was used. The examination of the convergence of the resulting iterative scheme does not apply if the linear problem is solved using Gauss-Seidel iteration steps. □

4. An Overview of Relevant Recent Works and Methods in Image Processing

This work is an introduction to image restoration, which has an interesting ill-posed problem. It is important to improve the quality of the images. As noise damages images and reduces the accuracy and performance of processing tasks, there are many modern ways to remove noise. In this section, we will mention the relevant modern methods based on the ROF model to remove Gaussian noise. Below are some recommended works.

4.1. Rudin-Osher-Fatemi Model. A denoising model based on the first-order total variation was proposed by Rudin et al. The ROF model is the popular name of the model. The ROF model does a great work of removing Gaussian noise, but it does not preserve image structures well. Otherwise, it creates artifacts.

Let $u_0(x)$, $u(x)$, and $v(x) \in \mathbb{R}$ be an original image, a restored image, and a noisy image, respectively, where $x(i, j) \in \Omega \subset \mathbb{R}^2$ is a pixel location, $\Omega = \{1, \cdots, m\} \times \{1, \cdots, n\}$ is an image domain, and m and n are the number of pixels by the image height and the image width. Rudin et al. proposed the ROF model to remove Gaussian noise as follows:

$$u^* = \arg\min_{u \in \Omega}\left(\frac{\lambda}{2}\|u - v\|_2^2 + \|\nabla u\|_2^2\right). \tag{54}$$

A parameter for regularization is λ. Data fidelity is the first term, while smoothness measured by total variance is the second term. The ROF model is used for solving the denoising problem based on total variation regularization (TV).

4.2. Method of Overlapping Group Sparsity and Second-Order Total Variation Regularization. In [25], we propose an image denoising method named OGS-SOTV by combining

(a) Overlapping group sparsity total variation regularization OGS-TV

$$u^* = \arg\min_{u \in \Omega}\left(\frac{\lambda}{2}\|u - v\|_2^2 + \varphi(\nabla u)\right), \tag{55}$$

where $\varphi(.)$ is an overlapping group sparsity functional with a group size of K and it is defined in [25] as

$$\varphi(u) = \sum_{i,j=1}^{K^2}\left|u_{ij,K}\right|_2. \tag{56}$$

(b) The second-order total variation regularization SOTV or TV2

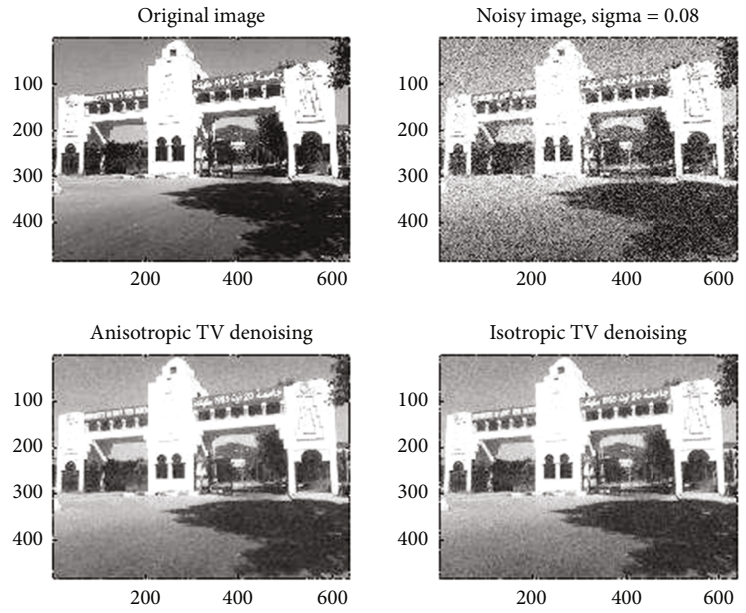

Figure 2: Split Bregman results using the university image.

$$u^* = \arg \min_{u\in\Omega} \left(\frac{\lambda}{2} \|u - v\|_2^2 + \left\|\nabla^2 u\right\|_2^2 \right). \quad (57)$$

A combined model with $\delta_1 > 0$ and $\delta_2 > 0$ can be considered as follows:

$$u^* = \arg \min_{u\in\Omega} \left(\frac{\lambda}{2} \|u - v\|_2^2 + \delta_1 \varphi(\nabla u) + \delta_2 \left\|\nabla^2 u\right\|_2^2 \right). \quad (58)$$

Assume $\delta_2 = \gamma.\delta_1$ and $\lambda = 2/\delta_1$. So, we obtain the adaptive denoising model based on OGS-TV and SO-TV which are defined as follows:

$$u^* = \arg \min_{u\in\Omega} \left(\frac{\lambda}{2} \|u - v\|_2^2 + \varphi(\nabla u) + \gamma \left\|\nabla^2 u\right\|_2^2 \right), \quad (59)$$

where γ is a balancing parameter of the noise removal term and artifact elimination term and λ is a regularization parameter. The numerical solution is obtained with ADMM (alternative direction method of multipliers) or the split Bregman method.

4.2.1. Purpose and Results of the Method. The purpose and results of the method are as follows:

(i) performance of noise removal of OGS-TV

(ii) performance of artifacts removed from TV-OS

(iii) regularization estimation parameter is also proposed to implement the method automatically

(iv) OGS-SOTV can remove noise effectively as well as eliminate artifacts

(v) the OGS-TV, TV-OS, and OGS-SOTV removed noise and artifacts better than the ROF model.

4.3. Medical Image Denoising Methods. In [26], we consider the denoising problem with medical images produced by X-ray/CT imaging techniques. The images are corrupted by Poisson noise. Since the Poisson noise is dependent on the signal, we cannot control the intensity of the noise, so we use Gaussian noise.

We proposed a medical image denoising method, by combining

(i) the total variation in the regularization of TV

(ii) the Anscombe transformation

4.3.1. Implementation Method

(1) Based on the ROF model, Le et al. proposed the Poisson denoising problem; the model is well known as the modified ROF model (mROF)

$$u^* = \arg \min_{u\in\Omega} \left(\frac{\lambda}{2} \|u - v \ln (u)\|_2^2 + \beta \|\nabla u\|_2^2 \right), \quad (60)$$

where $\beta = 1/\lambda u$ is a regularization parameter; it depends on the restored image of every iteration step. This matter reduces the accuracy and performance of the evaluation process.

(2) Anscombe transform is a mathematics tool for converting a Poisson data ϕ to standard Gaussian data v, it has the following form: $v = 2\sqrt{\phi + 3/8}$

(3) Instead of solving the mROF problem, we can solve the ROF problem by obtaining a solution u^*

(4) Apply the inverse Anscombe transform u^* to acquire the final denoised image $u_p = (u^*/2)^2 - 3/8$

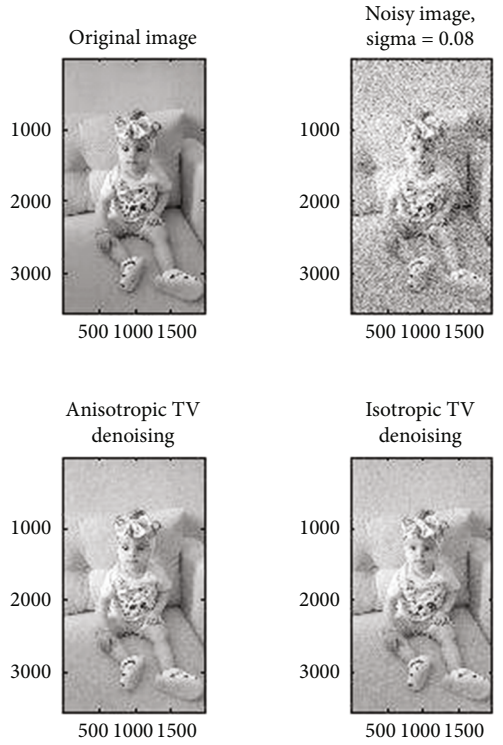

Figure 3: Split Bregman results using the Iline image.

4.3.2. Purpose and Results. The purpose and results of the method are as follows:

(i) the Anscombe transform is used to convert Gaussian noise in medical images into Poisson noise

(ii) apply the ROF model that is based on TV to remove Gaussian noise

(iii) the proposed method also gives a better denoising result than ROF or mROF by both visual result and restoration quality assessment metrics such as PSNR and SSIM

4.4. Adaptive Method for Image Restoration Based on TV1 and TV2. In [27], we propose an adaptive method for image restoration based on a combination of

(i) the first-order total variation regularization (TV1) also known as the ROF model

(ii) the second-order total variation regularization SO-TV or TV2 with

$$u^* = \arg\min_{u\in\Omega}\left(\frac{\lambda}{2}\|u-v\|_2^2 + \left\|\nabla^2 u\right\|_2^2\right). \quad (61)$$

A combined model with $\alpha > 0$ and $\beta > 0$ can be considered as follows:

$$u^* = \arg\min_{u\in\Omega}\left(\frac{\lambda}{2}\|u-v\|_2^2 + \alpha\|\nabla u\|_2^2 + \beta\left\|\nabla^2 u\right\|_2^2\right). \quad (62)$$

The model is well known as the TV-bounded Hessian model (TV-BH).

Suppose $\alpha = \beta/k$ and $\lambda = k/\beta, k > 0$. So, we obtain the adaptive image restoration model based on TV1 and TV2 as follows:

$$u^* = \arg\min_{u\in\Omega}\left(\frac{\lambda(v)}{2}\|u-v\|_2^2 + \|\nabla u\|_2^2 + k\left\|\nabla^2 u\right\|_2^2\right), \quad (63)$$

where k is a balancing parameter between the (TV1) and (TV2) and λ is a regularization parameter. We choose the regularization parameter λ as based on the inverse gradient:

$$\lambda(v) = \frac{\mu}{1+\tau\max_{\rho}\left|G_\rho * \nabla v\right|_2^2}, \quad (64)$$

where G_ρ is a $2D$ Gaussian kernel.

The numerical solution is obtained with ADMM (alternative direction method of multipliers) or the split Bregman.

4.4.1. Purpose and Results. The purpose and results of the method are as follows:

(i) uses the advantages of noise removal and edge preservation of the ROF model that is based on TV1

(ii) artifacts elimination of the second-order total variation TV2

(iii) adaptive multiscale parameter estimation. If $k > 1$, artifacts are eliminated; if $0 < k < 1$ remove noise;

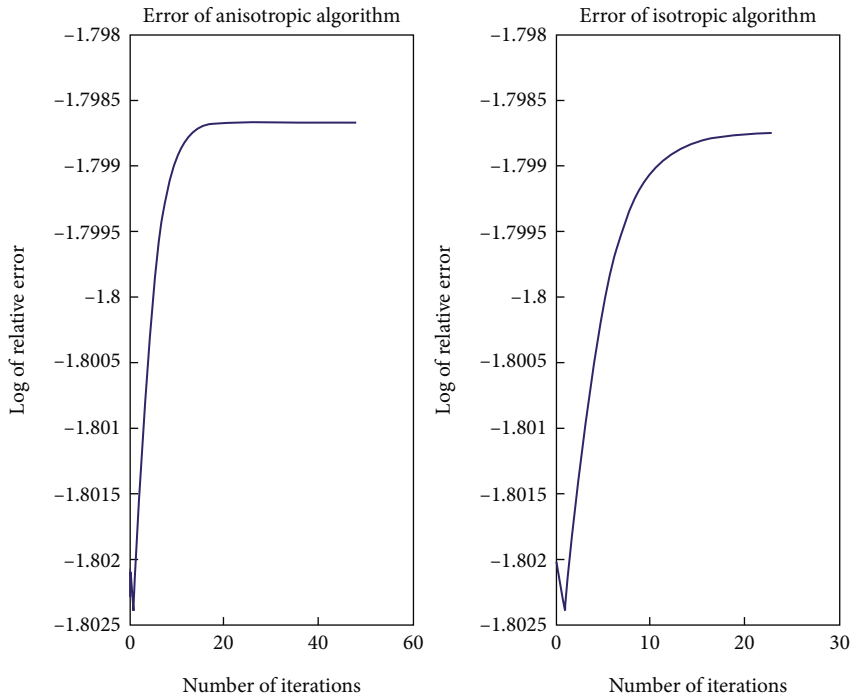

Figure 4: Split Bregman error results using the cameraman image.

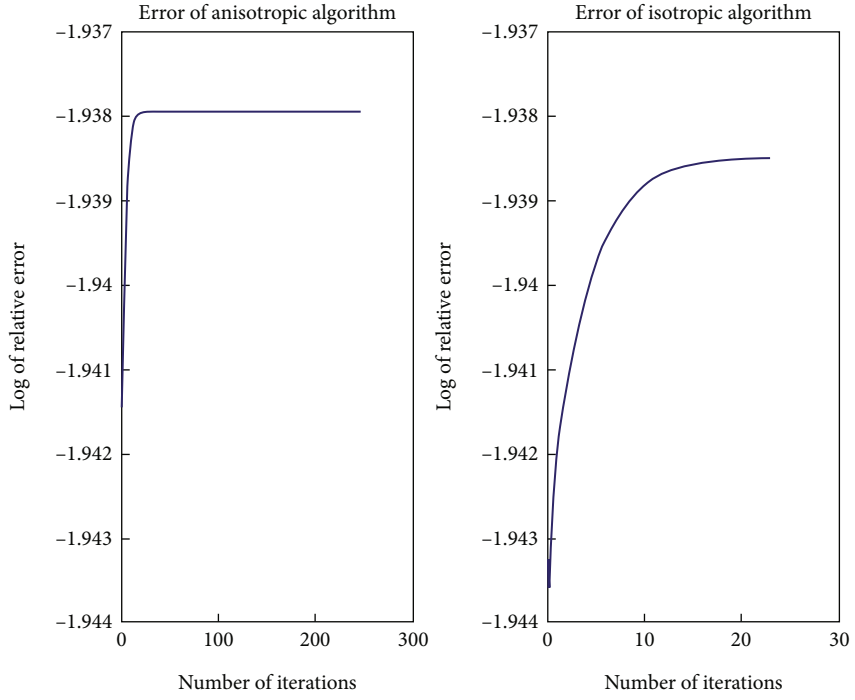

Figure 5: Split Bregman error results using the university image.

and if $k = 1$ to balance the performance of noise removal and artifact elimination

(iv) experimental results indicate that the proposed method obtains better restorations in terms of visual quality as well as quantitatively by PSNR and SSIM

5. Numerical Results

We use several images; the introduced additive noise is Gaussian, and we attempt to recover the original image to test the split Bregman methods in the problem of anisotropic and isotropic TV denoising. Let X be the matrices that depict an image of size $m \times n$. We then used

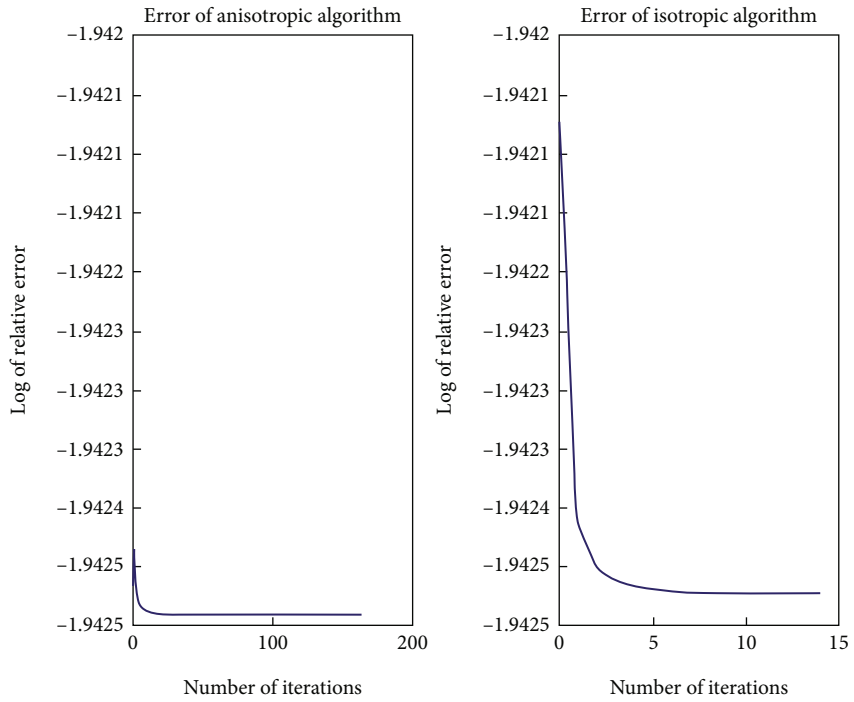

Figure 6: Split Bregman error results using the Iline image.

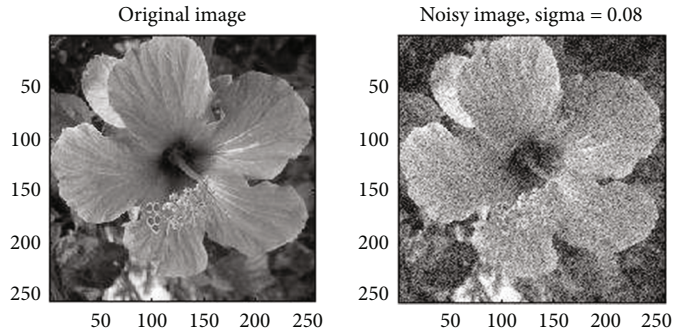

Figure 7: The original image and noisy image for sigma = 0.08.

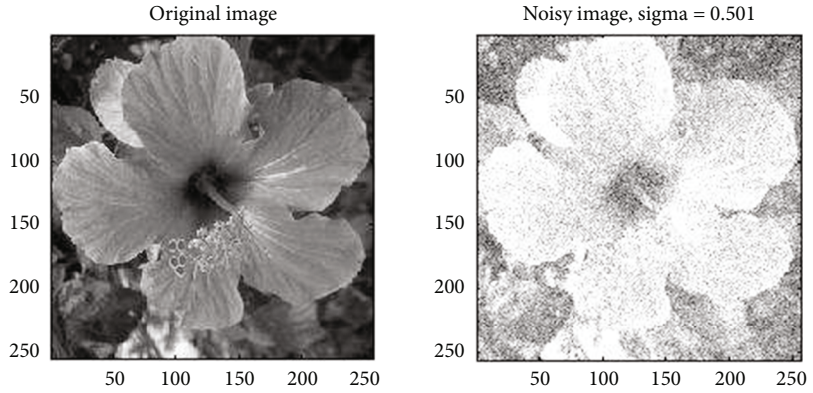

Figure 8: The original image and noisy image for sigma = 0.501.

MATLAB $f = \text{imnoise}(X, 'gaussian', \text{sigma})$ command to define our noise image f, where sigma is a version of the Gaussian noise level.

Clarification with the sources of the image used in Figure 1, I photographed my daughter Iline, with my mobile phone, and then I used MATLAB for converted

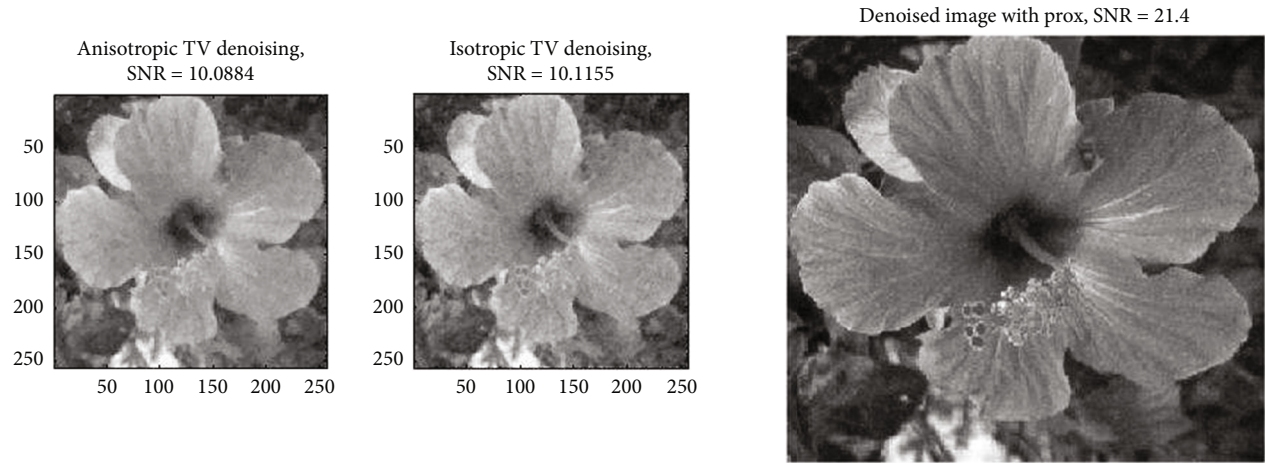

Figure 9: The SNR of denoising image by the anisotropic TV, the isotropic TV, and prox for sigma = 0.08.

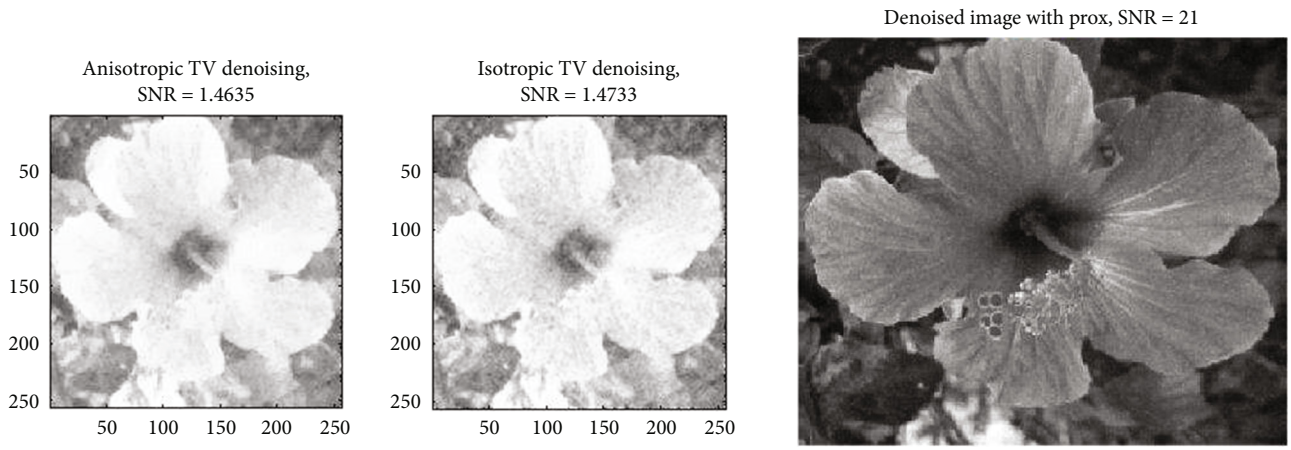

Figure 10: The SNR of denoising image by the anisotropic TV, the isotropic TV, and prox for sigma = 0.501.

the color image to grayscale $Y = \text{imread}('\text{Iline.png}')$; $X = \text{rgb2gray}(Y)$; then conduct studies on the image Iline.

We used the values $\mu = 0.1$ and $\lambda = 0.2$ and the tolerance $\text{Tol} = 10^{-5}$ in our studies.

The results of the anisotropic and isotropic TV denoising algorithms for various images are shown in Tables 1 and 2. The $\|u - X\|_2^2$ is used to calculate relative errors.

Tables 3 and 4 show the performance metrics for the anisotropic and isotropic TV denoising algorithms, with sigma = 0.08.

Tables 5 and 6 prove the *SNR* for the anisotropic and isotropic TV denoising algorithms for a single "flower" image, as well as different sigma values.

Table 7 shows the SNR values for the "flower" image that has been detached using the prox-penality, anisotropic TV denoising, and isotropic TV denoising algorithms.

5.1. Comments on Experimental Results

(i) As seen in Tables 1 and 2, isotropic TV denoising is faster and more accurate than the anisotropic version of the denoising method. In our experiments, we discovered that the sequence of residues for the image "Iline" converged monotonously to a lower value than the original noisy image. However, due to the large size of this image, this takes a long time. The errors in the cameraman and university images increased monotonously and converged to a relative error point higher than the initial noisy image's relative error. This can be seen in the images and visuals that have been presented

(ii) In Tables 3 and 4, we evaluate the quality of images restored by the image restoration models using square error (MSE), signal noise rate (SNR), peak signal-to-noise ratio (PSNR), image quality index (IQI), normalized cross-correlation (NK), average difference (AD), structural content (SC), maximum difference (MD), and normalized absolute error (NAE)

(iii) For the "flower" image of the isotropic and anisotropic TV denoising algorithms and the same segregation value of 0.08, Tables 5 and 6 illustrate the varied SNR results, number of iterations, relative error, and time

(iv) The relative error of split Bregman, an iterative technique with anisotropic and isotropic filters,

does not necessarily converge monotonously, as we have shown

(v) The quality of images produced with the prox method, on the other hand, remains constant, like the prox Lagrange value increases, maintaining the texture's performance after release

(vi) Figures 1–3 illustrate the outcomes of restoration methods such as the TV anisotropic and TV isotropic images: cameraman, university, Iline, and for sigma equal to 0.08

(vii) The results of errors applied to the TV anisotropic and TV isotropic renderings of the images are: cameraman, university, Iline, for a value of sigma equal to 0.08 are shown in Figures 4–6

(viii) Figures 7 and 8 illustrate the original image and noisy image of the flower for sigma equal to 0.08 and 0.501

(ix) Figures 9 and 10 also show that the restoration algorithms, such as the TV anisotropic and TV isotropic denoising, are not reliable and have problems during the restoration processing. In other words, they diverge as white noise invariance (sigma) increases. However, this is not the case when using the proximal algorithms. The latter appears to be very old and reliable. It provides a high sigma variance for the deforested image in Table 7

(x) The SNR of the prox algorithm-restored images is nearly constant

6. Conclusions

In this paper, we have compared the proximal penalty algorithms to solve a class of nondifferentiable optimization problems with the anisotropic TV and isotropic TV denoising algorithms for solving optimization problems. Based on the comparison of the restoration results from different related models, we can confirm that the prox algorithm suitable for image restoration produces the best high-quality results (clear, not smooth, and textures are kept), and the convergence method is guaranteed regardless of the SNR values if we compare it with other methods. Based on previous findings, we can conclude that the anisotropic TV and isotropic TV denoising algorithms work in a direct correlation relationship. In other words, regardless of how little the sigma value is, we get better and more old image quality results. The approach converges monotonously towards equal tolerance 10^{-5} despite the vast size of the image; it takes a long time to compute them, and the isotropic TV denoising algorithms are faster than the anisotropic TV denoising algorithms. In our tests, we discovered that the restored image is sharper and more accurate. The prox algorithm also gives better denoising results than TV that is anisotropic or isotropic, both in terms of the visual results and the restoration quality assessment metrics such as PSNR and SNR.

In our next studies, we would like to combine anisotropic and isotropic TV for image denoising to program the method in MATLAB, and we compared denoising methods.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Ngwa and E. Agyingi, "Effect of an external medium on tumor growth-induced stress," *IAENG International Journal of Applied Mathematics (JAM)*, vol. 42, no. 2, pp. 4-5, 2012.

[2] E. Y. Sidky, J. H. Jørgensen, and X. Pan, "Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm," *Physics in Medicine and Biology*, vol. 57, no. 10, pp. 3065–3091, 2012.

[3] C. Brune, A. Sawatzky, and M. Burger, "Primal and dual Bregman methods with application to optical nanoscopy," *International Journal of Computer Vision*, vol. 92, no. 2, pp. 211–229, 2011.

[4] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1964–1977, 1999.

[5] Y. Chen, J. Wu, and G. Yu, "Adaptive proximal point algorithms for total variation, image restoration," *Statistics, Optimization & Information Computing*, vol. 3, no. 1, pp. 15–29, 2015.

[6] J.-F. Aujol, "*Traitement d'image par approches variationnelles et équations aux dérivées partielles, Semestre d'enseignement UNESCO sur le traitement des images numériques, TUNIS, ENIT*," 2005.

[7] C. A. Micchelliy, L. Shenz, and Y. Xux, "Proximity algorithms for image models: denoising," *Inverse Problems*, vol. 27, no. 4, article 045009, 2011.

[8] N. Daili and K. Saadi, "Nondifferentiable augmented Lagrangian, ε-proximal penalty methods and applications," *Malaya Journal of Matematik*, vol. 4, no. 4, pp. 534–555, 2016.

[9] N. Daili, "Some augmented Lagrangian algorithms applied to convex nondifferentiable optimization problems," *Journal of information & Optimization Sciences (JIOS)*, vol. 33, no. 4-5, pp. 487–526, 2012.

[10] N. Daili and K. Saadi, "Epsilon-proximal point algorithms for nondifferentiable convex optimization problems and applications," *AMO-Advanced Modeling and Optimization*, vol. 14, no. 1, pp. 175–195, 2012.

[11] S. Gheraibia, A. Guesmia, and N. Daili, "The robustness of proximal penalty algorithms in restoration of noisy image, Hacettepe University," *Journal of Mathematics and Statistics*, vol. 46, no. 6, pp. 1043–1052, 2017.

[12] S. Gheraibia and N. Daili, "Restoration of the noised images by the proximal penalty algorithms," *Pacific Journal of Applied Mathematics*, vol. 7, no. 3, pp. 149–161, 2015.

[13] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution

of problems in convex programming," *Zhurnal Vychislitel'noy Matematiki i Matematicheskoy Fiziki*, vol. 7, no. 3, pp. 200–217, 1967.

[14] Y. Censor and A. Lent, "An iterative row-action method for interval convex programming," *Journal of Optimization Theory and Applications*, vol. 34, no. 3, pp. 321–353, 1981.

[15] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, "Bregman monotone optimization algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 2, pp. 596–636, 2003.

[16] D. Butnariu, Y. Censor, and S. Reich, "Iterative averaging of entropic projections for solving stochastic convex feasibility problems," *Computational Optimization and Applications*, vol. 8, no. 1, pp. 21–39, 1997.

[17] D. Butnariu and A. N. Iusem, *Totally Convex Functions for Fixed Points Computation and Infinite Dimensional Optimization*, Kluwer Academic Publishers, Dordrecht, 2000.

[18] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[19] J. Bush, *Bregman Algorithms*, Senior Thesis, University of California, Santa Barbara, 2011.

[20] F. Schopfer, A. K. Louis, and T. Schuster, "Nonlinear iterative methods for linear ill-posed problems in Banach spaces," *Inverse Problems*, vol. 22, no. 1, pp. 311–329, 2006.

[21] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[22] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, vol. 8, no. 2, pp. 337–369, 2010.

[23] R.-Q. Jia and H. Zhao, "A fast algorithm for the total variation model of image denoising," *Advances in Computational Mathematics*, vol. 33, no. 2, pp. 231–241, 2010.

[24] R.-Q. Jia, H. Zhao, and W. Zhao, "Convergence analysis of the Bregman method for the variational model of image denoising," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 367–379, 2009.

[25] N. M. Hue, D. N. H. Thanh, L. T. Thanh, N. N. Hien, and V. B. S. Prasath, "Image denoising with overlapping group sparsity and second order total variation regularization," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 370–374, Hanoi, Vietnam, 2019.

[26] L. T. Thanh and D. N. H. Thanh, "Medical images denoising method based on total variation regularization and Anscombe transform," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 26–30, Ho Chi Minh City, Vietna, September 2019.

[27] N. H. Dang, V. B. Thanh, S. Prasath, L. M. Hieu, and S. Dvoenko, "An adaptive method for image restoration based on high-order total variation and inverse gradient," Springer-Verlag, London Ltd., part of Springer Nature, 2020.