Implementation of high-order variational models made easy for image processing

Wenqi Lu\textsuperscript{a}, Jinming Duan\textsuperscript{b}, Zhaowen Qiu\textsuperscript{c,†}, Zhenkuan Pan\textsuperscript{a}, Ryan Wen Liu\textsuperscript{d} and Li Bai\textsuperscript{b}

Communicated by S. Wise

High-order variational models are powerful methods for image processing and analysis, but they can lead to complicated high-order nonlinear partial differential equations that are difficult to discretise to solve computationally. In this paper, we present some representative high-order variational models and provide detailed discretisation of these models and numerical implementation of the split Bregman algorithm for solving these models using the fast Fourier transform. We demonstrate the advantages and disadvantages of these high-order models in the context of image denoising through extensive experiments. The methods and techniques can also be used for other applications, such as image decomposition, inpainting and segmentation. Copyright © 2016 John Wiley & Sons, Ltd.

Keywords: image processing; total variation; bounded Hessian; total generalised variation; total curvature; split Bregman algorithm

1. Introduction

Variational methods have played an important role in many disciplines including quantum mechanics, engineering, statistics and economics. The basis of variational methods is the variational principle used in the calculus of variations, a field of mathematical analysis that deals with maximizing or minimizing functionals. These functionals are often expressed as integrals involving functions and their derivatives, and thus, variational methods involve solving partial differential equations (PDEs). In the context of image processing, a functional normally has a data term (or similarity term) and a regulariser term. The former allows the similarity between the restored image and the original image, whilst the latter imposes some kind of smoothness on the restored image involving first or/and second derivative.

In recent two decades, numerous variational models [1–4], have been applied to image processing. One of the most popular variational models for image processing is the total variation (TV) model proposed by Rudin Osher and Fatemi [5]. The energy functional of the model is as follows:

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\nabla u|, \quad (1.1)
\]

where \( f : \Omega \rightarrow \mathbb{R} \) is a given noisy image defined on a rectangle space \( \Omega \) and penalty parameter \( \alpha \) imposed on the TV regulariser term controls the level of smoothness of resulting clean image.

This first-order TV model is effective in preserving object edges while removing noise. A desirable property of image-denosing algorithms is to restore images without losing important features (i.e. edges, corners, contrast etc.). However, this model has several undesirable side effects, one of which is the staircase effect, that is, the restored image appears jagged. High-order variational models have been proposed to eliminate the staircase effect.

In this paper, detailed implementation of several high-order variational models for image processing are given. These high-order models are briefly introduced later.
The simplest high-order models involve only second-order derivatives. For instance, a second-order model \[6\], named the TL (Total Laplace) model in this paper, used the second-order Laplacian operator:

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\Delta u|,
\]

where the laplace is defined as \(\Delta u = \partial_x \partial_x u + \partial_y \partial_y u\).

A second-order model uses the Bounded Hessian (BH) regulariser, named BH model in this paper, and was proposed by Lysaker et al. \[7\], Scherzer et al. \[8, 9\], Lai et al. \[10\], and Bergounioux et al. \[11\]. The model minimises the following energy functional

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\nabla u|,
\]

where the Hessian matrix of \(u\) (i.e. \(\nabla^2 u\)) in the image domain \(\Omega\) is given as follows:

\[
\nabla^2 u = \begin{pmatrix}
\partial_x \partial_x u & \partial_x \partial_y u \\
\partial_y \partial_x u & \partial_y \partial_y u
\end{pmatrix}
\]

A model, which combines TV regulariser with Laplacian regulariser, named the CEP2-L^2 model in this paper, was proposed in \[12\]:

\[
E(u_1, u_2) = \frac{1}{2} \int_{\Omega} (u_1 - u_2)^2 + \alpha \int_{\Omega} |\nabla u_1| + \beta \int_{\Omega} |\Delta u_2|,
\]

where positive penalty parameters \(\alpha\) and \(\beta\) balance the first and second derivatives.

A model, which combines TV and Laplacian regularisers, named TV and Laplacian model (TVL) in this paper, was proposed in \[13–15\]:

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\nabla u| + \beta \int_{\Omega} |\Delta u|.
\]

A model, which combines TV and Bounded Hessian regularisers, named the TVBH in this paper, was proposed in \[16, 17\]:

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\nabla u| + \beta \int_{\Omega} |\nabla^2 u|.
\]

A model, which uses infimal-convolution of the first-order TV regulariser and BH regulariser, named INFCON in this paper, was proposed in \[18\]. The TV regulariser preserves the edges and the BH keeps the image smooth. The INFCON model is given as

\[
E(u_1, u_2) = \frac{1}{2} \int_{\Omega} (u_1 - u_2)^2 + \alpha \int_{\Omega} |\nabla u_1| + \beta \int_{\Omega} |\nabla^2 u_2|.
\]

More recently, the total generalized variation (TGV) model was proposed in \[19\] for image denoising. The main feature of TGV is that it can deal with different image characteristics (piecewise constant, piecewise affine, piecewise quadratic etc.). The TGV model is of the following form:

\[
E(u, p) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |\nabla u - p| + \beta \int_{\Omega} |\varepsilon(p)|,
\]

where \(\varepsilon(p)\) is the symmetrised derivative and defined as

\[
\varepsilon(p) = \begin{pmatrix}
\partial_x p_1 & \frac{\partial_x p_1 + \partial_y p_2}{2} \\
\frac{\partial_y p_1 + \partial_x p_2}{2} & \partial_y p_2
\end{pmatrix}.
\]

The minimum of (1.9) is taken over all gradients of the deformation field \(p = (p_1, p_2)\) on image space \(\Omega\).

Finally, high-order variational model using the total curvature (TC) regulariser was proposed in \[20–23\] for image denoising, which uses the mean curvature of the image surface as a regulariser. In fact, the curvature-based models were first introduced by Nitzberg and Mumford \[24\] in early 1990s and then widely extended for image inpainting \[25–27\], segmentation with depth \[28\], construction of illusory shapes \[29\] etc. The TC model is given as

\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} \left| \nabla u \right| \left( \frac{\nabla u}{|\nabla u|} \right).
\]

Despite of the effectiveness of these high-order models in removing the staircase effect, it is often a challenging problem to minimise the corresponding functionals. Technically, the Euler equations of the regulariser are fourth order nonlinear partial differential equations (PDEs), which are very difficult to discretise to solve computationally. In this paper, split Bregman algorithm \[30–32\] is applied to transform the energy minimization problem of given model into several subproblems. These subproblems are then efficiently
solved by fast Fourier transform (FFT) [22, 33–36], analytical soft thresholding equation [37] and projection formula [22, 36] without any iteration.

The rest of the paper is organised as follows: Section 2 introduces detailed discretisation of first-order, second-order and fourth-order differential operators using finite difference scheme. Based on Section 2, the fast split Bregman algorithm is designed for all the mentioned models in Section 3. In section 4, we present extensive comparative experiments. Section 5 draws a conclusion about the advantages and disadvantages of these models.

2. Discretisation

In order to solve previous variational models using efficient split Bregman method [30, 38, 39], the discretisation of first-order, second-order and fourth-order derivative operators are required. They are respectively \( \partial_x^{-1}, \partial_x^+, \partial_x^-, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+, \partial_x^{+1}, \partial_x^{-1}, \partial_x^+ \). The gradient, Hessian matrix (1.4) and symmetrised derivative (1.10) can be discretised as follows, respectively:

\[
\nabla u = \begin{pmatrix} \partial_x^+ u \\ \partial_y^+ u \end{pmatrix} \tag{2.1}
\]

\[
\nabla^2 u = \begin{pmatrix} \partial_x^{-1} \partial_x^+ u & \partial_y^{-1} \partial_x^+ u \\ \partial_x^{-1} \partial_y^+ u & \partial_y^{-1} \partial_y^+ u \end{pmatrix} \tag{2.2}
\]

\[
\varepsilon(p) = \left( \frac{\partial_x^{+1} p_1 + \partial_x^{-1} p_2}{2} \frac{\partial_y^{+1} p_1 + \partial_y^{-1} p_2}{2} \right) \tag{2.3}
\]

Theoretically, \( \partial_x \partial_y u \) should be equal to \( \partial_y \partial_x u \) in Hessian matrix (1.4), so it is necessary to define its discrete form such that \( \partial_x^+ \partial_y^+ u = \partial_y^+ \partial_x^+ u \) in (2.2). The issue of boundary conditions is also needed addressing, as it is very important in numerical optimisation. Unlike the Neumann boundary conditions used in [40–42], we apply periodic boundary conditions. By choosing such conditions, FFT can be conveniently utilised for the split Bregman algorithm to improve computational speed.

The first-order forward and backward difference schemes are first given. Let \( \Omega \rightarrow \mathbb{R}^{M \times N} \) denote the two-dimensional grey scale image space with size \( MN \). The coordinates \( x \) and \( y \) are oriented along columns and rows, respectively. So the first-order forward differences of \( u \) at point \( (i, j) \) along \( x \) and \( y \) directions are respectively

\[
\partial_x^+ u_{ij} = \begin{cases} u_{i,j+1} - u_{ij} & \text{if } 1 \leq i \leq M, 1 \leq j < N \\ u_{i+1,j} - u_{ij} & \text{if } 1 \leq i < M, j = N \end{cases} \tag{2.4}
\]

\[
\partial_y^+ u_{ij} = \begin{cases} u_{i+1,j} - u_{ij} & \text{if } 1 \leq i < M, 1 \leq j \leq N \\ u_{i,j} - u_{ij} & \text{if } i = M, 1 \leq j \leq N \end{cases} \tag{2.5}
\]

The first order backward differences are respectively

\[
\partial_x^- u_{ij} = \begin{cases} u_{ij} - u_{i,j-1} & \text{if } 1 \leq i \leq M, 1 \leq j < N \\ u_{ij} - u_{i,j} & \text{if } 1 \leq i < M, j = 1 \end{cases} \tag{2.6}
\]

\[
\partial_y^- u_{ij} = \begin{cases} u_{ij} - u_{i-1,j} & \text{if } 1 \leq i < M, 1 \leq j \leq N \\ u_{i-1,j} - u_{ij} & \text{if } i = 1, 1 \leq j \leq N \end{cases} \tag{2.7}
\]

For every \( p = (p_1, p_2) \in \mathbb{R}^{M \times N} \) and \( u \in \mathbb{R}^{M \times N} \), the discrete version of the first-order adjoint divergence operator satisfies

\[
\sum_{1 \leq j \leq N} \sum_{1 \leq i \leq M} -\text{div}(p) \cdot u = \sum_{1 \leq j \leq N} \sum_{1 \leq i \leq M} p \cdot \nabla u.
\]

Therefore, according to the definition of discrete gradient operator, it is easy to check the discrete divergence as

\[
\text{div}(p_{ij}) = \partial_x^- p_{ij} + \partial_y^- p_{ij}.
\]

The discrete second-order derivatives \( \partial_x^+ \partial_x^- u, \partial_x^+ \partial_y^- u, \partial_y^+ \partial_y^- u, \partial_y^+ \partial_x^- u, \partial_y^+ \partial_x^+ u, \partial_y^+ \partial_y^+ u, \partial_x^+ \partial_x^- u, \partial_x^+ \partial_x^+ u, \partial_x^+ \partial_y^- u, \partial_x^+ \partial_y^+ u, \partial_y^- \partial_x^- u, \partial_y^- \partial_x^+ u, \partial_y^- \partial_y^- u, \partial_y^- \partial_y^+ u \) and \( \partial_y^- \partial_y^- u \) at point \( (i, j) \) can be written down as follows, respectively. These operators are just the corresponding compositions of the discrete first-order derivative.

\[
\partial_x^+ \partial_x^- u_{ij} = \partial_x^+ \partial_x^+ u_{ij} = \begin{cases} u_{i,j} - 2u_{ij} + u_{i,j+1} & \text{if } 1 \leq i \leq M, j = 1 \\ u_{i,j} - 2u_{ij} + u_{i+1,j} & \text{if } 1 \leq i < M, 1 \leq j < N \\ u_{i,j} - 2u_{ij} + u_{i,j+1} & \text{if } 1 \leq i < M, j = N \end{cases}
\]

\[
\partial_y^+ \partial_y^- u_{ij} = \partial_y^+ \partial_y^+ u_{ij} = \begin{cases} u_{i,j} - 2u_{ij} + u_{i+1,j} & \text{if } 1 \leq i < M, 1 \leq j < N \\ u_{i,j} - 2u_{ij} + u_{i,j+1} & \text{if } 1 \leq i \leq M, j = 1 \\ u_{i,j} - 2u_{ij} + u_{i+1,j} & \text{if } 1 \leq i \leq M, 1 \leq j < N \end{cases}
\]
Note that the constraint \( \partial_x \partial_y u = \partial_y \partial_x u \) in Hessian matrix (1.4) is satisfied as now the discrete form \( \partial_x^+ \partial_y^- u_{ij} = \partial_y^- \partial_x^+ u_{ij} \) in (2.11).

Based on (2.9) and (2.10), the definition of discrete Laplace operator is given as

\[
\Delta u_{ij} = \text{div}(\nabla u_{ij}) = \partial_x^+ \partial_y^- u_{ij} + \partial_y^- \partial_x^+ u_{ij}.
\]  

(2.13)

In addition, we need the discrete second-order divergence operator that also has adjointness property: for every \( q = \left( \begin{array}{c} q_1 \\ q_2 \\ q_3 \\ q_4 \end{array} \right) \in \mathbb{R}^{M \times N} \) and \( u \in \mathbb{R}^{M \times N} \), we have \( \sum_{1 \leq i \leq \text{M}} \sum_{1 \leq j \leq \text{N}} \text{div}^2(q) \cdot u = \sum_{1 \leq i \leq \text{M}} \sum_{1 \leq j \leq \text{N}} q \cdot \nabla^2 u \). Then the discrete second-order divergence reads

\[
\text{div}^2(q_{ij}) = \partial_x^+ \partial_y^- q_{ij} + \partial_y^- \partial_x^+ q_{ij} + \partial_y^- \partial_x^- q_{ij} + \partial_y^+ \partial_x^+ q_{ij}.
\]  

(2.14)

In order to implement these high-order models, we give another two definitions of the discrete second-order derivatives

\[
\partial_x^+ \partial_y^- u_{ij} = \begin{cases} 
 u_{i+1,j} - u_{ij} - u_{ij+1} + u_{ij-1} & \text{if } 1 \leq j < \text{N} \\
 u_{i,j} - u_{ij-1} + u_{ij+1} & \text{if } 1 \leq i < \text{M}
\end{cases}
\]  

(2.15)

\[
\partial_y^+ \partial_x^- u_{ij} = \begin{cases} 
 u_{i+1,j} - u_{ij} - u_{ij+1} + u_{ij-1} & \text{if } 1 \leq j < \text{N} \\
 u_{i,j} - u_{ij-1} + u_{ij+1} & \text{if } 1 \leq i < \text{M}
\end{cases}
\]  

(2.16)

and the discrete fourth order derivatives

\[
\partial_x^+ \partial_y^- \partial_x^+ \partial_y^- u_{ij} = \partial_x^+ \partial_y^- \partial_x^+ \partial_y^- u_{ij} = 6u_{ij} + u_{ij+2} + u_{ij-2} - 4u_{ij+1} - 4u_{ij-1}.
\]  

(2.17)

\[
\partial_y^+ \partial_y^- \partial_y^+ \partial_y^- u_{ij} = 6u_{ij} + u_{ij+1} + u_{ij+2} - 4u_{ij+1} - 4u_{ij-1}.
\]  

(2.18)

\[
\partial_x^+ \partial_y^- \partial_y^+ \partial_x^+ u_{ij} = \partial_y^+ \partial_y^- \partial_y^+ \partial_x^+ u_{ij} = \partial_y^+ \partial_y^- \partial_y^+ \partial_x^+ u_{ij} = \partial_y^+ \partial_y^- \partial_y^+ \partial_x^+ u_{ij} = -2 \left( u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} \right) \text{.}
\]  

(2.19)

For simplicity, the finite differences on the boundaries for the discrete fourth-order derivatives are not listed. One can easily compose the already-defined second-order derivatives in (2.9)–(2.12) for the boundary conditions of the fourth-order derivatives in (2.17)–(2.19).

Based on (2.17)–(2.19), we can obtain following two discrete fourth-order differential operators:

\[
\text{div}^2 (\nabla^2 u_{ij}) = \partial_x^+ \partial_y^- \partial_x^+ \partial_y^- u_{ij} + \partial_y^+ \partial_y^- \partial_y^+ \partial_y^- u_{ij} + \partial_x^+ \partial_y^- \partial_x^+ \partial_x^+ u_{ij} + \partial_y^+ \partial_y^- \partial_y^+ \partial_y^- u_{ij} + \partial_x^+ \partial_y^- \partial_x^+ \partial_y^- u_{ij}.
\]  

(2.20)

\[
\Delta (\Delta u_{ij}) = \partial_x^+ \partial_y^- \partial_x^+ \partial_y^- u_{ij} + \partial_y^+ \partial_y^- \partial_y^+ \partial_y^- u_{ij} + \partial_x^+ \partial_y^- \partial_x^+ \partial_x^+ u_{ij} + \partial_y^+ \partial_y^- \partial_y^+ \partial_y^- u_{ij}.
\]  

(2.21)

Note that \( \text{div}^2 (\nabla^2 u_{ij}) = \Delta (\Delta u_{ij}) \). Figure 1 describes the behaviour of all the previous discrete differential operators. Having defined all necessary discrete quantities, the numerical optimisation of the variational models in section 1 can be implemented in the next section.
Figure 1. Illustration of the discrete first-order, second-order and fourth-order derivative approximations.

3. Fast numerical algorithm based on Split Bregman Method

3.1. Split Bregman for total variation model (1.1)

First, we introduce an auxiliary splitting vector variable $w = (w_1, w_2)$, a Bregman iterative parameter $b = (b_1, b_2)$ and a positive penalty parameter $\theta$, transforming the functional (1.1) into following form

$$E(u, w; b) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega |w| + \frac{\theta}{2} \int_\Omega (w - \nabla u - b)^2. \quad (3.1)$$

In order to find the minimiser of functional (1.1), an alternating optimization method should be used to search all the saddle points of functional (3.1). To do so, we first fix variable $w$, deriving the Euler-equation with respect to $u$

$$u - \theta \text{div} (\nabla u) = f - \theta \text{div} (w - b). \quad (3.2)$$

According to the definition of discrete gradient (2.1), divergence (2.8) and Laplace (2.13) in section 2, the discretisation form of (3.2) can be written as

$$u_{ij} - \theta \left( \partial_x^- u_{ij} + \partial_y^+ u_{ij} \right) = G_{ij} \quad (3.3)$$

with $G_{ij} = f_{ij} - \theta \left( \partial_x^- (w_{ij} - b_{ij}) + \partial_y^- (w_{2ij} - b_{2ij}) \right)$. As periodic boundary conditions have been imposed on the discrete derivatives in section 2, the discrete Fourier transform can be directly applied to the both sides of Eqn (3.3)

$$\mathcal{F} \left( u_{ij} - \theta \left( \partial_x^- u_{ij} + \partial_y^+ u_{ij} \right) \right) = \mathcal{F} (G_{ij}),$$

where \( \mathcal{F} \) denotes the discrete Fourier transform. For discrete frequencies \( r \) and \( s \), we have

\[
\left( 1 - 2\theta \left( \cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 2 \right) \right) \mathcal{F}(u_{ij}) = \mathcal{F}(G_{ij}).
\]

(3.4)

\( i \in [1,M] \) and \( j \in [1,N] \) are the indexes in discrete time domain. \( r \in [0,M] \) and \( s \in [0,N] \) are the frequencies in the discrete frequency domain. (3.4) provides us with a closed-form solution of \( u_{ij} \) as

\[
u_{ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{\mathcal{F}(G_{ij})}{\xi} \right) \right),
\]

(3.5)

where \( \mathcal{F}^{-1} \) denotes the discrete inverse Fourier transform. \( \Re \) is the real part of a complex number. “—” stands for pointwise division of matrices.

Fixing variables \( u \) for vector \( w \) next, we can obtain its Euler-equation as follows

\[
\alpha \frac{w}{|w|} + \theta \left( w - \nabla u - b \right) = 0,
\]

which can be solved component wisely through following discrete 2D analytical generalised soft thresholding equation with the convention that 0/0 = 0

\[
\w_{ij} = \max \left( |\nabla u_{ij} + b_{ij}| - \frac{\alpha}{\theta}, 0 \right) \frac{\nabla u_{ij} + b_{ij}}{|\nabla u_{ij} + b_{ij}|}.
\]

(3.6)

Finally, updating the Bregman iterative parameter \( b \), which can be seen from step 6 in the Algorithm 3.1. The split Bregman algorithm for the TV denoising problem (1.1) reads as follows:

**Algorithm 3.1** : Split Bregman algorithm for TV model (1.1)

1. **function** TVDenoising(f)
2. Initialization: Set \( (w; b) = 0, (\alpha, \theta) > 0 \)
3. **repeat**
4. Compute \( u \) according to (3.5)
5. Compute \( w \) according to (3.6)
6. Update Bregman iterative parameter \( b \leftarrow b + \nabla u - w \)
7. **until** convergence of \( u \)
8. **return** \( u \)
9. **end function**

### 3.2. Split Bregman for total Laplace (TL) model (1.2)

In order to minimise TL model (1.2), an auxiliary splitting scalar variable \( w \), a Bregman iterative parameter \( b \) and a penalty parameter \( \theta \) are introduced, transforming (1.2) into

\[
E(u, w; b) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |w| + \frac{\theta}{2} \int_{\Omega} (w - \Delta u - b)^2.
\]

Applying alternating optimization technique, we first fix \( w \) for \( u \) and obtain following Euler equation

\[
u + \theta \Delta (\Delta u) = f + \theta \Delta (w - b).
\]

(3.7)

Referring to definition of (2.13) and (2.21) in section 2, (3.7) is discretised as

\[
u_{ij} + \theta \left( \bar{a}_x^{-} \bar{a}_x^{+} \bar{a}_x^{-} u_{ij} + \bar{a}_x^{-} \bar{a}_x^{+} \bar{a}_x^{-} u_{ij} + \bar{a}_y^{-} \bar{a}_y^{+} \bar{a}_y^{-} u_{ij} + \bar{a}_y^{-} \bar{a}_y^{+} \bar{a}_y^{-} u_{ij} \right) = G_{ij}
\]

(3.8)

with \( G_{ij} = f_{ij} + \theta \left( \bar{a}_x^{-} \bar{a}_x^{+} (w_{ij} - b_{ij}) + \bar{a}_y^{-} \bar{a}_y^{+} (w_{ij} - b_{ij}) \right) \). Applying discrete Fourier transform to the both sides of Eqn (3.8) gives

\[
\mathcal{F} \left( u_{ij} + \theta \left( \bar{a}_x^{-} \bar{a}_x^{+} \bar{a}_x^{-} u_{ij} + \bar{a}_x^{-} \bar{a}_x^{+} \bar{a}_x^{-} u_{ij} + \bar{a}_y^{-} \bar{a}_y^{+} \bar{a}_y^{-} u_{ij} + \bar{a}_y^{-} \bar{a}_y^{+} \bar{a}_y^{-} u_{ij} \right) \right) = \mathcal{F} (G_{ij})
\]

(3.9)
For discrete frequencies $r$ and $s$, we have the following equivalent of Eqn (3.9)

\[
\left(1 + 4\theta \left(\cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 2\right)^2\right) \mathcal{F}(u_{ij}) = \mathcal{F}(G_{ij}).
\]  

(3.10)

(3.11) provides us with a closed-form solution of $u$ as

\[
u_{ij} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{ij})}{\xi}\right)\right).
\]  

(3.11)

Next fixing variable $u$ to derive following Euler equation with respect to scalar variable $w$

\[
\frac{w}{|w|} + \theta (w - \Delta u - b) = 0,
\]

which can be solved component wisely through the following discrete 1D analytical generalised soft thresholding equation with the convention that $0/0 = 0$

\[
w_{ij} = \max\left(|\Delta u_{ij} + b_{ij}| - \frac{\alpha}{\theta} 0\right) \text{sign}(\Delta u_{ij} + b_{ij}).
\]  

(3.12)

Lastly, we give the overall implementation of TL model (1.2), where Bregman iterative parameter $b$ is updated.

**Algorithm 3.2 : Split Bregman algorithm for TL model (1.2)**

1: function $\text{TLDenoising}(f)$
2: Initialization: $\text{Set} (w; b) = 0, (\alpha, \theta) > 0$
3: repeat
4: Compute $u$ according to (3.11)
5: Compute $w$ according to (3.12)
6: Update Bregman iterative parameter $b \leftarrow b + \Delta u - w$
7: until convergence of $u$
8: return $u$
9: end function

3.3. **Split Bregman for bounded Hessian model (1.3)**

To solve BH model (1.3) using split Bregman method, we transform it into following multivariable energy functional

\[
E(u, w; b) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |w| + \frac{\theta}{2} \int_{\Omega} (w - \nabla^2 u - b)^2,
\]  

(3.13)

where $w = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}$ is a matrix-valued function related to the Hessian of the function $u$. $|w| = \sqrt{\sum_{1 \leq n \leq 4} (w_n)^2}$ represents the Frobenius norm of matrix $w$. By introducing Bregman iteration parameter $b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$ and applying Bregman distance technique, constraint $w = \nabla^2 u$ can be effectively enforced when functional (3.13) reaches convergent.

Using the same manner to optimise (3.13), we first fix the 4D vector variable $w$ for $u$

\[
u + \theta \text{div}^2 (\nabla^2 u) = f + \theta \text{div}^2 (w - b)
\]  

(3.14)

The discretisation form of (3.14) is given as follows according to the definition (2.14) and (2.20) in section 2:

\[
u_{ij} + \theta \left(\partial_x^+ \partial_x^- \partial_x^+ u_{ij} + \partial_y^- \partial_x^- \partial_x^+ \partial_x^+ u_{ij} + \partial_x^- \partial_x^- \partial_x^+ \partial_x^+ u_{ij} + \partial_x^+ \partial_x^- \partial_x^+ \partial_x^+ u_{ij}\right) = G_{ij}
\]  

(3.15)

with $G = f + \theta \left(\partial_x^+ \partial_x^- (w_1 - b_1) + \partial_y^- \partial_x^- (w_2 - b_2) + \partial_x^- \partial_x^- (w_3 - b_3) + \partial_y^+ \partial_x^- (w_4 - b_4)\right)$. Applying discrete Fourier transform to the both sides of equation (3.15) gives

\[
\mathcal{F}(\nu_{ij} + \theta \left(\partial_x^+ \partial_x^- \partial_x^+ u_{ij} + \partial_y^- \partial_x^- \partial_x^+ \partial_x^+ u_{ij} + \partial_x^- \partial_x^- \partial_x^+ \partial_x^+ u_{ij} + \partial_x^+ \partial_x^- \partial_x^+ \partial_x^+ u_{ij}\right)) = \mathcal{F}(G_{ij}).
\]  

(3.16)
The equivalent of (3.16) is

\[
\left(1 + 4\theta \left( \cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 2 \right)^2 \right) \mathcal{F}(u_{ij}) = \mathcal{F}(G_{ij}).
\]

(3.17)

Note that the left-hand side of (3.17) is the same as that of (3.11). (3.17) gives following closed-form

\[
u_{ij} = \Im \left( \frac{\mathcal{F}^{-1} \left( \mathcal{F} \left( G_{ij} \right) \right)}{\xi} \right).
\]

(3.18)

Fixing variable \(u\) to derive the following Euler equation with respect to 4D vector \(w\)

\[
\alpha \frac{w}{|w|} + \theta \left( w - \nabla^2 u - b \right) = 0,
\]

which can be analytically solved by following discrete 4D generalised soft thresholding equation with the convention that \(0/0 = 0\)

\[
w_{ij} = \max \left( |\nabla^2 u_{ij} + b_{ij}| - \frac{\alpha}{\theta} \right) \frac{\nabla^2 u_{ij} + b_{ij}}{|\nabla^2 u_{ij} + b_{ij}|}.
\]

(3.19)

**Algorithm 3.3 : Split Bregman algorithm for BH model (1.3)**

1: function \(BH\)Denoising(\(f\))
2: Initialization: Set \((w; b) = 0\) and \((\alpha, \theta) > 0\)
3: repeat
4: Compute \(u\) according to (3.18)
5: Compute \(w\) according to (3.19)
6: Update Bregman iterative parameter \(b \leftarrow b + \nabla^2 u - w\)
7: until convergence of \(u\)
8: return \(u\)
9: end function

3.4. Split Bregman for CEP2-L2 model (1.5)

As CEP2-L2 model (1.5) includes both first-order and second-order derivatives, we should introduce two splitting variables \((w, v)\), two Bregman iterative parameters \((b, d)\) and two penalty parameters \((\theta_1, \theta_2)\), transforming the original functional into

\[
E(u_1, u_2, w, v; b, d) = \frac{1}{2} \int_{\Omega} (f - u_1 - u_2)^2
+ \alpha \int_{\Omega} |w| + \frac{\theta_1}{2} \int_{\Omega} (w - \nabla u_1 - b)^2.
+ \beta \int_{\Omega} |v| + \frac{\theta_2}{2} \int_{\Omega} (v - \Delta u_2 - d)^2
\]

To do optimisation, we first fix variables \((u_2, w, v; b, d)\) for \(u_1\) and obtain following Euler equation

\[
u_1 - \theta_1 \text{div} (\nabla u_1) = (f - u_2) + \theta_1 \text{div} (b - w).
\]

(3.20)

This equation is second-order linear PDE, which has similar form to (3.2). We can directly write the solution of \(u_1\) as

\[
u_{1ij} = \Im \left( \frac{\mathcal{F}^{-1} \left( \mathcal{F} \left( G_{1ij} \right) \right)}{\xi_1} \right),
\]

(3.21)

where \(G_1\) is the right-hand side of (3.20) and

\[
\xi_1 = 1 - 2\theta_1 \left( \cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 2 \right).
\]

(3.22)

Fixing variables \((u_1, w, v; b, d)\) for \(u_2\), we have

\[
u_2 + \theta_2 \Delta (\Delta u_2) = (f - u_1) + \theta_2 \Delta (v - d).
\]

(3.23)
Note that variable \( u_1 \) has been calculated from (3.21) and should be used for (3.23). (3.23) has the similar form to (3.7), and its closed-form solution can be derived as

\[
u_{2j} = \Re \left( F^{-1} \left( \frac{F(G_{2j})}{\xi_2} \right) \right),
\]

where \( G_2 \) is the right-hand side of (3.23) and

\[
\xi_2 = 1 + 4\theta_2 \left( \cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 1 \right)^2.
\]

Fixing variables \((u_1, u_2, v, b, d)\) for vector \( w \), we have

\[
\alpha \frac{w}{|w|} + \theta_1 (w - \nabla u_1 - b) = 0.
\]

Its analytical solution is as follows:

\[
w_{ij} = \max \left( \frac{1}{|\nabla u_{ij} + b_{ij}|} \left( \frac{\alpha}{\theta_1} \frac{\nabla u_{ij} + b_{ij}}{|\nabla u_{ij} + b_{ij}|} \right) \right)
\]

After \( w \) is updated, the following formulation with respect to scalar variable \( v \) is derived

\[
\beta \frac{v}{|v|} + \theta_2 (v - \Delta u_2 - d) = 0.
\]

The solution is similar to (3.12), which is a 1D analytical generalised soft thresholding equation

\[
v_{ij} = \max \left( \frac{1}{|\Delta u_{2ij} + d_{ij}|} \left( \frac{\beta}{\theta_2} \frac{\Delta u_{2ij} + d_{ij}}{|\Delta u_{2ij} + d_{ij}|} \right) \right).
\]

Finally, the update of two Bregman iterative parameters are shown from step 08 and 09 in the overall Algorithm 3.4

**Algorithm 3.4**: Split Bregman algorithm for CEP2-L^2 model (1.5)

01: function CEP2-L\(^2\)\(_{\text{denoising}}\) (\(f\))
02: Initialization: Set \( u_2 = f, (w, v, b, d) = 0 \), \((\alpha, \beta, \theta_1, \theta_2) > 0\)
03: repeat
04: Compute \( u_1 \) according to (3.21)
05: Compute \( u_2 \) according to (3.24)
06: Compute \( w \) according to (3.26)
07: Compute \( v \) according to (3.27)
08: Update Bregman iterative parameter \( b \leftarrow b + \nabla u_1 - w \)
09: Update Bregman iterative parameter \( d \leftarrow d + \Delta u_2 - v \)
10: until convergence of \( u = u_1 + u_2 \)
11: return \( u = u_1 + u_2 \)
12: end function

3.5. Split Bregman for TV and Laplace model (1.6)

Similarly to the optimisation produces of CEP2-L\(^2\) functional, we first introduce same numbers of auxiliary variables and obtain the following energy functional

\[
E(u, w, v; b, d) = \frac{1}{2} \int_{\Omega} (f - u)^2 + \alpha \int_{\Omega} |w| + \frac{\theta_1}{2} \int_{\Omega} (w - \nabla u - b)^2 + \beta \int_{\Omega} |v| + \frac{\theta_2}{2} \int_{\Omega} (v - \Delta u - d)^2
\]

The Euler equation with respect to \( u \) is

\[
u = \theta_1 \text{div} (\nabla u) + \theta_2 \Delta (\Delta u) = f - \theta_1 \text{div} (w - b) + \theta_2 (v - d).
\]
This formulation is a combination of (3.2) and (3.7), which can be also solved by discrete Fourier transform. Its solution is given as

$$u_{ij} = \Re \left( F^{-1} \left( \frac{F(G_i)}{\xi} \right) \right),$$

(3.29)

where \( G \) is the right-hand side of (3.28), \( \xi = \xi_1 + \xi_2 - 1 \) where \( \xi_1 \) and \( \xi_2 \) are (3.22) and (3.25), respectively.

The discretisation problems of \( w \) and \( v \) are given by following two equations, respectively:

$$w_{ij} = \max \left( |\nabla u_{ij} + b_{ij}| - \frac{\alpha}{\theta_1} 0 \right) \frac{\nabla u_{ij} + b_{ij}}{|\nabla u_{ij} + b_{ij}|},$$

(3.30)

$$v_{ij} = \max \left( |\Delta u_{ij} + d_{ij}| - \frac{\beta}{\theta_2} 0 \right) \text{sign} (\Delta u_{ij} + d_{ij}).$$

(3.31)

**Algorithm 3.5**: Split Bregman algorithm for TVL model (1.6)

```
01: function TVLDensifying(f)
02: Initialization: Set \( (w, v, b, d) = 0 \), \( (\alpha, \beta, \theta_1, \theta_2) > 0 \)
03: repeat
04: Compute \( u \) according to (3.29)
05: Compute \( w \) according to (3.30)
06: Compute \( v \) according to (3.31)
07: Update Bregman iterative parameter \( b \leftarrow b + \nabla u - w \)
08: Update Bregman iterative parameter \( d \leftarrow d + \Delta u - v \)
09: until convergence of \( u \)
10: return \( u \)
11: end function
```

3.6. **Split Bregman for infimal-convolution model (1.8)**

The INFCON model (1.8) can be written as the following multivariable functional:

$$E(u_1, u_2, w, v; b, d) = \frac{1}{2} \int_{\Omega} (f - u_1 - u_2)^2$$

$$+ \alpha \int_{\Omega} |w| + \frac{\theta_1}{2} \int_{\Omega} (w - \nabla u_1 - b)^2,$$

$$+ \beta \int_{\Omega} |v| + \frac{\theta_2}{2} \int_{\Omega} (v - \nabla^2 u_2 - d)^2$$

where variables \( w \) and \( b \) are 2D vector, while \( v \) and \( d \) are 4D vector. To minimise the functional, we first fix \((u_2, w, v; b, d)\) for \( u_1 \). The solution form of \( u_1 \) is exactly as same as formulation (3.21).

After \( u_1 \) is solved, we fix \((u_1, w, v; b, d)\) and derive the Euler equation with respect to \( u_2 \)

$$u_2 + \theta_2 \nabla^2 (\nabla^2 u_2) = (f - u_1) + \theta_2 \nabla^2 (v - d)$$

(3.32)

This formulation is similar to (3.14) and can be solved via following FFT under the assumption of circulant boundary condition

$$u_{2ij} = \Re \left( F^{-1} \left( \frac{F(G_{ij})}{\xi_2} \right) \right),$$

(3.33)

where \( G_2 \) is the right-hand side of Eqn (3.32) and \( \xi_2 \) is defined in (3.25).

\((u_1, u_2, v; b, d)\) are fixed next to obtain the solution of the variable \( w \), which is same as (3.26). Fixing \((u_1, u_2, w; b, d)\) for vector \( v \), we have following Euler equation:

$$\beta \frac{v}{|w|} + \theta_2 (v - \nabla^2 u_2 - d) = 0.$$  

Its solution is the following 4D analytical equation with the convention that \(0/0 = 0\)

$$v_{ij} = \max \left( |\nabla^2 u_{2ij} + d_{ij}| - \frac{\beta}{\theta_2} 0 \right) \frac{\nabla^2 u_{2ij} + d_{ij}}{|\nabla^2 u_{2ij} + d_{ij}|}$$

(3.34)
The overall Algorithm 3.6 for INFCON model is as follows:

**Algorithm 3.6**: Split Bregman algorithm for INFCON model (1.8)

```
01: function INFCONDenoising(f)
02: Initialization: Set $u_2 = f$, $(w, v; b, d) = 0$, $(\alpha, \beta, \theta_1, \theta_2) > 0$
03: repeat
04: Compute $u_1$ according to (3.21)
05: Compute $u_2$ according to (3.33)
06: Compute $w$ according to (3.26)
07: Compute $v$ according to (3.34)
08: Update Bregman iterative parameter $b \leftarrow b + \nabla u_1 - w$
09: Update Bregman iterative parameter $d \leftarrow d + \nabla^2 u_2 - v$
10: until convergence of $u$
11: return $u = u_1 + u_2$
12: end function
```

3.7. **Split Bregman for combined TV and BH (TVBH) model (1.7)**

Using the same manner as before, we first transform original (1.7) model into following multivariable formulation

$$
E(u, w, v; b, d) = \frac{1}{2} \int_{\Omega} (f - u)^2 + \alpha \int_{\Omega} |w| + \frac{\theta_1}{2} \int_{\Omega} (w - \nabla u - b)^2 + \beta \int_{\Omega} |v| + \frac{\theta_2}{2} \int_{\Omega} (v - \nabla^2 u - d)^2
$$

To minimise the functional, we first fix $(w, v; b, d)$ for $u$ and obtain its Euler equation

$$
u - \theta_1 \text{div} (\nabla u) + \theta_2 \text{div}^2 (\nabla^2 u) = f - \theta_1 \nabla (w - b) + \theta_2 \text{div}^2 (v - d). \quad (3.35)
$$

This formulation is a combination of (3.2) and (3.14), which can be solved via discrete fast Fourier transform. Its solution is given as

$$
u_{i,j} = \Re \left( F^{-1} \left( \frac{F(G_{i,j})}{\xi} \right) \right), \quad (3.36)
$$

where $G$ is the right-hand side of (3.35). $\xi = \xi_1 + \xi_2 - 1$ where $\xi_1$ and $\xi_2$ are (3.22) and (3.25), respectively.

The solution of variable $w$ is same as (3.30), and vector $v$ can be updated by

$$
v_{i,j} = \max \left( |\nabla^2 u_{i,j} + d_{i,j}| - \frac{\beta}{\theta_2}, 0 \right) \frac{\nabla^2 u_{i,j} + d_{i,j}}{|\nabla^2 u_{i,j} + d_{i,j}|}. \quad (3.37)
$$

**Algorithm 3.7**: Split Bregman algorithm for TVBH model (1.7)

```
01: function TVBH_Denoising(f)
02: Initialization: Set $(w, v; b, d) = 0$, $(\alpha, \beta, \theta_1, \theta_2) > 0$
03: repeat
04: Compute $u$ according to (3.36)
05: Compute $w$ according to (3.30)
06: Compute $v$ according to (3.37)
07: Update Bregman iterative parameter $b \leftarrow b + \nabla u - w$
08: Update Bregman iterative parameter $d \leftarrow d + \nabla^2 u - v$
09: until convergence of $u$
10: return $u$
11: end function
```

3.8. **Split Bregman for total generalised variation model (1.9)**

In order to solve TGV model (1.9) using split Bregman method, we introduce auxiliary variables $(w, v; b, d)$ and positive penalty parameters $(\theta_1, \theta_2)$, transforming it into following form:
The solution of this linear PDE is given as

\begin{align}
E(u, p, w; b, d) &= \frac{1}{2} \int_{\Omega} (u - f)^2 \\
&\quad + \alpha \int_{\Omega} |w| + \frac{\theta_1}{2} \int_{\Omega} (w - \nabla u + p - b)^2, \\
&\quad + \beta \int_{\Omega} |v| + \frac{\theta_2}{2} \int_{\Omega} (v - \epsilon(p) - d)^2.
\end{align}

(3.38)

The meaning of the primal and dual variables in (3.38) is listed as follows:

- \( u \) denotes the denoised image we need to find;
- \( p = (p_1, p_2) \) is the symmetrised gradient of the deformation field;
- \( w = (w_1, w_2) \) is a 2D vector valued function related to vector field \( \nabla u - p \);
- \( v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} \) is a 4D vector valued function related to symmetrised derivative \( \epsilon(p) \);
- \( b = (b_1, b_2) \) denotes the Bregman iterative parameter for enforcing \( w = \nabla u - p \);
- \( d = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \) denotes the Bregman iterative parameter for enforcing \( v = \epsilon(p) \).

Applying alternating optimisation method, we first fix variables \((p, w, v; b, d)\) to obtain the following Euler equation with respect to \( u \)

\[ u - \theta_1 \text{div} (\nabla u) = f - \theta_1 \text{div} (w + p - b). \]

(3.39)

The solution of this linear PDE is given as

\[ u_{ij} = \mathcal{H} \left( \mathcal{F}^{-1} \left( \mathcal{F}(p_{ij}) / \xi_1 \right) \right), \]

(3.40)

where \( \mathcal{G} \) is right-hand side of (3.39) and \( \xi_1 \) is defined as (3.22).

Fixing variables \((u, w, v; b, d)\) next, the Euler equation with respect to \( p = (p_1, p_2) \) can be derived. For each \( p_1 \) and \( p_2 \), we have the following two discrete formulations

\begin{align}
\left( \theta_1 - \theta_2 \partial_x^+ \partial^{-} - \frac{\theta_2}{2} \partial_x^+ \partial_y^- \right) p_{1i,j} - \frac{\theta_2}{2} \partial_x^+ \partial_y^- p_{2i,j} &= h_{1i,j} \\
\left( \theta_1 - \theta_2 \partial_x^+ \partial^{-} - \theta_2 \partial_y^+ \partial^{-} \right) p_{2i,j} - \frac{\theta_2}{2} \partial_x^+ \partial_y^- p_{1i,j} &= h_{2i,j},
\end{align}

(3.41, 3.42)

where the differential operators in (3.41) and (3.42) can be found in section 2 and

\begin{align}
h_1 &= \theta_1 \left( \partial_x^+ u + b_1 - w_1 \right) - \theta_2 \partial_x^+ (v_1, -d_1) - \theta_2 \partial_y^+ (v_2, -d_2) \\
h_2 &= \theta_1 \left( \partial_y^+ u + b_2 - w_2 \right) - \theta_2 \partial_x^+ (v_2, -d_2) - \theta_2 \partial_y^+ (v_2, -d_2).
\end{align}

By applying the discrete Fourier transform to both sides of (3.41) and (3.42), we have the following system of linear equations

\[ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \mathcal{F}(p_{1i,j}) \\ \mathcal{F}(p_{2i,j}) \end{pmatrix} = \begin{pmatrix} \mathcal{F}(h_{1i,j}) \\ \mathcal{F}(h_{2i,j}) \end{pmatrix}, \]

where the coefficients are

\begin{align}
a_{11} &= \theta_1 - \theta_2 \left( 2 \cos \frac{2\pi s}{N} - 2 \right) - \theta_2 \left( 2 \cos \frac{2\pi r}{M} - 2 \right) \\
a_{12} &= -\frac{\theta_2}{2} \left( -1 + \cos \frac{2\pi s}{N} + \sqrt{-1} \sin \frac{2\pi r}{M} \right) \left( 1 - \cos \frac{2\pi s}{N} + \sqrt{-1} \sin \frac{2\pi r}{M} \right) \\
a_{21} &= -\frac{\theta_2}{2} \left( -1 + \cos \frac{2\pi s}{N} + \sqrt{-1} \sin \frac{2\pi r}{M} \right) \left( 1 - \cos \frac{2\pi s}{N} + \sqrt{-1} \sin \frac{2\pi r}{M} \right) \\
a_{22} &= \theta_1 - \theta_2 \left( 2 \cos \frac{2\pi s}{N} - 2 \right) - \theta_2 \left( 2 \cos \frac{2\pi r}{M} - 2 \right).
\end{align}
The coefficient matrix \( \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \) is a \( M \times N \) numbers of \( 2 \times 2 \), whose determinant is
\[
D = \left( \theta_1 - 2 \theta_2 \left( \cos \frac{2 \pi s}{N} + \cos \frac{2 \pi r}{M} - 2 \right) \right) \left( \theta_1 - \theta_2 \left( \cos \frac{2 \pi s}{N} + \cos \frac{2 \pi r}{M} - 2 \right) \right),
\]
which is always positive for all discrete frequencies if \((\theta_1, \theta_2) > 0\). After the systems of linear equations are solved for each frequency \( r \) and \( s \) over the discrete frequency domain, we use the discrete inverse Fourier transform to obtain the analytical forms of \( p_1 \) and \( p_2 \)
\[
p_{1ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{a_{22} \mathcal{F}(h_{1ij}) - a_{12} \mathcal{F}(h_{2ij})}{D} \right) \right), \quad (3.43)
\]
\[
p_{2ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{a_{11} \mathcal{F}(h_{2ij}) - a_{21} \mathcal{F}(h_{1ij})}{D} \right) \right). \quad (3.44)
\]
Fixing variables \((u, p, v, b, d)\) for 2D vector \( w \), we have
\[
\frac{\alpha}{|w|} \frac{\partial u}{\partial i} + \theta_1 (w - \nabla u + p - b) = 0,
\]
which can be solved by following discrete 2D analytical generalised soft thresholding equation with convention that \(0/0 = 0\)
\[
w_{ij} = \max \left( |\nabla u_{ij} - p_{ij} + b_{ij}| - \frac{\alpha}{\theta_1}, 0 \right) \frac{\nabla u_{ij} - p_{ij} + b_{ij}}{|\nabla u_{ij} - p_{ij} + b_{ij}|}. \quad (3.45)
\]
After \( w \) is solved, the Euler equation with respect to \( v \) is given as
\[
\beta \frac{\partial v}{\partial i} + \theta_2 (v - \varepsilon (p) - d) = 0
\]
whose solution is following 4D analytical generalised soft thresholding equation with convention that \(0/0 = 0\)
\[
v_{ij} = \max \left( \varepsilon (p_{ij}) + d_{ij}, \frac{\varepsilon}{\theta_2} (p_{ij}) + d_{ij} \right). \quad (3.46)
\]
Finally, the overall Algorithm 3.8 is presented as

**Algorithm 3.8**: Split Bregman algorithm for TGV model (1.9)

01: function \( \text{TGV}\text{Denosing}(\ell) \)
02: Initialization: Set \((p, w, v; b, d)\) = 0, \((\alpha, \beta, \theta_1, \theta_2) > 0\)
03: repeat
04: Compute \( u \) according to (3.40)
05: Compute \( p \) according to (3.43) and (3.44)
06: Compute \( w \) according to (3.45)
07: Compute \( v \) according to (3.46)
08: Update Bregman iterative parameter \( b \leftarrow b + \nabla u - p - w \)
09: Update Bregman iterative parameter \( d \leftarrow d + \varepsilon (p) - v \)
10: until convergence of \( u \)
11: return \( u \)
12: end function

3.9. **Split Bregman for total curvature (TC) model (1.11)**

In order to solve TC model (1.11), we first consider the following splitting
\[
E(u) = \frac{1}{2} \int_{\Omega} (u - f)^2 dx + \alpha \int_{\Omega} |q| dx \quad (3.47)
\]
\[
s.t. q = \text{div}(n), p = \nabla u, n = p/|p|.
\]
We note that constraint \( n = p/|p| \) is equivalent to following two constraints according to the well-known Hölder inequality
\[
|n| \leq 1, \ |p| = n \cdot p. \quad (3.48)
\]
A new auxiliary vector $m$ is then introduced to replace variable $n$ in (3.48), which can be deemed as a relaxation of $n$ and FFT can be thus applied. After that, we have following five constraints

\[ q = \text{div} (n), \quad p = \nabla u, \quad |m| \leq 1, \quad |p| = m \cdot p, \quad m = n. \]  
(3.49)

For $|m| \leq 1$ in image domain, we have $\Omega, |p| \geq m \cdot p$. By introducing auxiliary variables and penalty parameters, the split Bregman method can be employed to transform (3.47) with constraints in (3.49) into following constrained energy functional

\[
E(u, q, p, n, m; b_1, b_2, b_3, b_4) = \frac{1}{2} \int_{\Omega} (u - f)^2 + \alpha \int_{\Omega} |q| \\
+ \int_{\Omega} (\theta_1 + b_1)(|p| - p \cdot m) + \theta_3 \int_{\Omega} (p - \nabla u - b_2)^2 \\
+ \theta_3 \int_{\Omega} (q - \text{div}(n) - b_3)^2 + \theta_4 \int_{\Omega} (n - m - b_4)^2 \\
\text{s.t.} \quad |m| \leq 1.
\]  
(3.50)

Note that for the constraint $|p| = m \cdot p$, we use augmented Lagrangian method with L1-norm for the penalisation in functional (3.50) as it is true that $|p| - m \cdot p > 0$ when $|m| \leq 1$. The meaning of the primal and dual variables in (3.50) is listed as in the following:

- $u$ denotes the denoised image we need to find;
- $q$ is a scalar valued function related to divergence of vector function $n$;
- $p = (p_1, p_2)$ is a vector valued function related to gradient of the function $u$;
- $n = (n_1, n_2)$ is a vector valued function related to the unit vectors of the level curves of $u$;
- $m = (m_1, m_2)$ is a vector valued function used to relax variable $n$ and thus FFT can be used;
- $b_1$ denotes a scalar Lagrangian multiplier for constraint $|p| = m \cdot p$;
- $b_2 = (b_{21}, b_{22})$ denotes a vector Bregman iterative parameter for constraint $p = \nabla u$;
- $b_3$ denotes a scalar Bregman iterative parameter for constraint $q = \text{div}(n)$;
- $b_4 = (b_{41}, b_{42})$ denotes a vector Bregman iterative parameter for constraint $m = n$.

Using alternating optimisation technique and first fixing variables $(q, p, n, m; b_1, b_2, b_3, b_4)$, we obtain the Euler equation with respect to $u$ as follows:

\[ u - \theta_2 \text{div} (\nabla u) = f - \theta_2 \text{div} (p - b_2). \]  
(3.51)

The solution of this second order linear PDE is given as

\[ u_{ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{\mathcal{F}(G_{ij})}{\xi_1} \right) \right), \]  
(3.52)

where $G$ is right-hand side of (3.51) and $\xi_1$ is defined as (3.22).

Variables $(u, p, n, m; b_1, b_2, b_3, b_4)$ are fixed next, and we obtain the following equation

\[ \alpha \frac{q}{|q|} + \theta_3 (q - \text{div} (n) - b_3) = 0, \]

Its solution reads as

\[ q_{ij} = \max \left( |\text{div}(n)_{ij} + b_{3ij} - \frac{\alpha}{\theta_3}, 0 \right) \text{sign} (\text{div}(n_{ij}) + b_{3ij}) \]  
(3.53)

Fixing variables $(u, q, n, m; b_1, b_2, b_3, b_4)$, we obtain the Euler equation with respect to $p$

\[ (\theta_1 + b_1) \frac{p}{|p|} + \theta_2 (p - \nabla u - b_2) - (\theta_1 + b_1)m = 0, \]

which can be solved by the analytical soft thresholding equation as

\[ p_{ij} = \max \left( \frac{|S_{ij}| - \frac{(\theta_1 + b_{1ij})}{\theta_2}, 0} \right) \frac{S_{ij}}{|S_{ij}|}, \]  
(3.54)

where $S_{ij} = \nabla u_{ij} + b_{2ij} + \frac{(\theta_1 + b_{1ij})m_{ij}}{\theta_2}$. If the $|S_{ij}|$ is 0, we have $0 \cdot (0/0) = 0$.

Fixing $(u, q, p, m; b_1, b_2, b_3, b_4)$, the following two Euler equations for vector variable $n = (n_1, n_2)$ are derived, respectively

\[ (\theta_4 - \theta_3 \frac{\partial^+}{\partial^+} \frac{\partial^-}{\partial^-}) n_1 - \theta_3 \frac{\partial^+}{\partial^+} \frac{\partial^-}{\partial^-} n_2 = h_{1ij} \]  
(3.55)
Algorithm 3.9. The split Bregman algorithm for the TC denoising problem (1.11) is as follows:

\[
\left( \theta_4 - \theta_3 \hat{\delta}_j \right) n_2 - \theta_3 \delta_j n_1 = h_{2ij},
\]

where

\[
h_1 = \theta_4 (m_1 + b_{41}) - \theta_3 \delta_j (q - b_3)
\]

\[
h_2 = \theta_4 (m_2 + b_{42}) - \theta_3 \delta_j (q - b_3)
\]

By applying the discrete Fourier transform to both sides of (3.55) and (3.56), we have the following linear system

\[
\begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{pmatrix}
\begin{pmatrix}
\mathcal{F}(n_{1ij}) \\
\mathcal{F}(n_{2ij})
\end{pmatrix}
= \begin{pmatrix}
\mathcal{F}(h_{1ij}) \\
\mathcal{F}(h_{2ij})
\end{pmatrix},
\]

where the coefficients are

\[
a_{11} = \theta_4 - 2 \theta_3 \left( \cos \frac{2 \pi s}{N} - 1 \right)
\]

\[
a_{12} = -\theta_3 \left( 1 - \cos \frac{2 \pi r}{M} + \sqrt{-1} \sin \frac{2 \pi r}{M} \right) \left( -1 + \cos \frac{2 \pi s}{N} + \sqrt{-1} \sin \frac{2 \pi s}{N} \right)
\]

\[
a_{21} = -\theta_3 \left( 1 - \cos \frac{2 \pi s}{N} + \sqrt{-1} \sin \frac{2 \pi s}{N} \right) \left( -1 + \cos \frac{2 \pi r}{M} + \sqrt{-1} \sin \frac{2 \pi r}{M} \right)
\]

\[
a_{22} = \theta_4 - 2 \theta_3 \left( \cos \frac{2 \pi r}{M} - 1 \right)
\]

The coefficient matrix \( \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \) is also a \( M \times N \) numbers of \( 2 \times 2 \) system, whose determinant is

\[
D = \theta_4^2 - 2 \theta_3 \theta_4 \left( \cos \frac{2 \pi s}{N} + \cos \frac{2 \pi r}{M} - 2 \right),
\]

which is always positive for all discrete frequencies if \( (\theta_3, \theta_4) > 0 \). After the systems of linear equations are solved for each frequency \( r \) and \( s \) over the discrete frequency domain, we use the discrete inverse Fourier transform to obtain the analytical forms of \( n_1 \) and \( n_2 \)

\[
n_{1ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{a_{22} \mathcal{F}(h_{1ij}) - a_{12} \mathcal{F}(h_{2ij})}{D} \right) \right)
\]

\[
n_{2ij} = \Re \left( \mathcal{F}^{-1} \left( \frac{a_{11} \mathcal{F}(h_{2ij}) - a_{21} \mathcal{F}(h_{1ij})}{D} \right) \right).
\]

Fixing \( (u, q, p, n; b_1, b_2, b_3, b_4) \), variable \( m \) can be derived from the following closed form:

\[
m_{ij} = \frac{(\theta_i + b_i) p_{ij}}{\theta_4} + n_{ij} - b_{4ij}.
\]

In order to satisfy constraint \( |m| \leq 1 \), the following projection formulation is then imposed on the \( \tilde{m}_{ij} \)

\[
m_{ij} = \frac{\tilde{m}_{ij}}{\max(|\tilde{m}_{ij}|, 1)}
\]

Finally, updating the auxiliary variables \( (b_1, b_2, b_3, b_4) \), which are listed from step 09 to 12 in the Algorithm 3.9. The split Bregman algorithm for the TC denoising problem (1.11) is as follows:

4. Numerical experiments

In this section, the performance of different methods is compared quantitatively and qualitatively. We apply peak signal-to-noise ratio, signal-to-noise ratio, root mean square error and structure similarity index map to perform quantitative evaluation for different methods. The models and their energy functionals are summarised in Table I in order to easily assess the difference between them. All experiments are performed using Matlab 2014b on a Windows 7 platform with an Intel Xeon CPU E5-1620 at 3.70GHz and 32GB memory.
Algorithm 3.9: Split Bregman algorithm for TC model (1.11)

01: function $\text{TC}_{\text{Denoising}}(f)$
02: Initialization: Set $(q, p, n, m; b_1, b_2, b_3, b_4) = 0, (\alpha, \beta_1, \beta_2, \beta_3, \beta_4) > 0$
03: repeat
04: Compute $u$ according to (3.52)
05: Compute $q$ according to (3.53)
06: Compute $p$ according to (3.54)
07: Compute $a$ according to (3.57) and (3.58)
08: Compute $m$ according to (3.59)
09: Update Lagrangian multiplier $b_1 \leftarrow b_1 + \beta_1(|p| - m \cdot p)$
10: Update Bregman iterative parameter $b_2 \leftarrow b_2 + \nabla u - p$
11: Update Bregman iterative parameter $b_3 \leftarrow b_3 + \text{div}(a) - q$
12: Update Bregman iterative parameter $b_4 \leftarrow b_4 + m - n$
13: until convergence of $u$
14: return $u$
15: end function;

<table>
<thead>
<tr>
<th>No.</th>
<th>Model</th>
<th>Energy functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TV</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>2</td>
<td>TL</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>3</td>
<td>BH</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>4</td>
<td>CEP2-L²</td>
<td>$E(u_1, u_2) = \frac{1}{2} \int_\Omega (f - u_1 - u_2)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>5</td>
<td>TVL</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>6</td>
<td>INFCON</td>
<td>$E(u_1, u_2) = \frac{1}{2} \int_\Omega (f - u_1 - u_2)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>7</td>
<td>TVBH</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>8</td>
<td>TGV</td>
<td>$E(u, p) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
<tr>
<td>9</td>
<td>TC</td>
<td>$E(u) = \frac{1}{2} \int_\Omega (u - f)^2 + \alpha \int_\Omega</td>
</tr>
</tbody>
</table>

4.1. Comparison 1

In Figure 2, we test the edge preservation ability of different models on a piecewise constant image (a) through visual inspection. From (c) to (k), it is clear that the first-order TV and second-order TC models preserve the edges of the strips perfectly. Their denoised results look almost same as the original image (a). TGV also gives very good edge-preservation result. However, it produces additional lines on the denoised stripes because it approximates image (a) with piecewise affine function. Similar visual effect can be observed on results by INFCON and CEP2-L² models. Image (e) illustrates directly applying BH regulariser can blur the edges of object, whilst image (b) shows TL model using Laplace regulariser is the worst method for edge preservation. Its resulting denoised image takes on some undesirable spikes. By incorporating first-order TV regulariser into pure second-order BH and TL models, TVL and TVBH models can improve the denoised quality as can be seen from images (g) and (i) qualitatively and Table II quantitatively. However, the object edges are still smeared by these two models.

In Figure 3, we present the associated residual images $f - u$ for all the models, where $f$ is noisy image Figure 2(b) and $u$ denotes each corresponding denoised one in Figure 2. Image (j) by TC model contains almost all noise, whilst the other eight images contains more or less structural information of Figure 2(a), which means some contrast information has been lost in the denoised images handled by the eight models. Figure 3 illustrates TC model is the best in preserving image contrast.

In Figure 4, the middle slices of the examples in Figure 2 are presented. We can see that the denoised slice curve (red) in (j) almost overlaps with the original noise free slice curve (blue), which demonstrates the preservation of image contrast as well as corners of TC model. Because of the contrast loss, the red curve generated by TV model (b) slightly deviates from original one along vertical directions. Compared with TV and TC models, the other seven high-order models impose too much smoothness on the curve edges, leading to less pleasant fitting results.

4.2. Comparison 2

In Figure 5, we test the smoothness preservation ability of different models on a piecewise smooth image (a). To do so, the original image (a) is corrupted with Gaussian noise of variance 0.005 and 0.015, respectively. The noise has been produced with MATLAB’s built in function imnoise.
Figure 2. Edge preservation ability test. All the methods in Table I are performed on noisy image (b), and their corresponding results are demonstrated from (c) to (k), respectively.

Table II. Comparison of PSNR, SNR, RMSE and SSIM using different methods on images of 2 different noise variances.

<table>
<thead>
<tr>
<th></th>
<th>Figure 6: noise variance 0.005</th>
<th>Figure 8: noise variance 0.015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SNR</td>
</tr>
<tr>
<td>Noisy image</td>
<td>23.6978</td>
<td>19.1788</td>
</tr>
<tr>
<td>TV</td>
<td>32.6621</td>
<td>26.1032</td>
</tr>
<tr>
<td>TL</td>
<td>27.8122</td>
<td>23.2933</td>
</tr>
<tr>
<td>BH</td>
<td>30.0447</td>
<td>25.5257</td>
</tr>
<tr>
<td>CEP2-L²</td>
<td>32.2822</td>
<td>27.7633</td>
</tr>
<tr>
<td>TVL</td>
<td>30.6670</td>
<td>26.1480</td>
</tr>
<tr>
<td>INFCOV</td>
<td>32.9872</td>
<td>28.4682</td>
</tr>
<tr>
<td>TVBH</td>
<td>32.3548</td>
<td>27.8358</td>
</tr>
<tr>
<td>TGV</td>
<td>34.4020</td>
<td>29.8830</td>
</tr>
<tr>
<td>TC</td>
<td>34.2254</td>
<td>29.7064</td>
</tr>
</tbody>
</table>

In Figure 6, the denoised results on Figure 5(b) by different methods are presented. One can easily find that image (a) by TV model are composed of jagged appearance which is staircase effect. In addition, the cleaned results shown from (d)-(g) illustrate these high order models that involve TV regulariser also lead to staircase. Even though TC model cannot completely remove the side effect, which can be observed from image (i). The pure second order TL and BH do not produce unfavourable staircase artifact. However, their results are not very visually satisfactory. For TL model, there exists some spikes at the edges of the geometrical shapes, whilst BH slightly blurs the object edges. The visual effects show TGV totally eliminates staircase effect without smearing edges in the image. Thus it gives best smoothness preservation result (i.e. highest PSNR, SNR, SSIM and lowest RMSE shown in Table II).

In Figure 7, we show the middle slices of the examples in Figure 6. It is obvious to see TL, BH and TGV models produce smoother contour than TV, CEP2-L², TVL, INFCOV, TVBH and TC models.

In Figure 8, the denoised results on Figure 6(c) are demonstrated. As can be seen, as noise increases, the differences between each methods are much clearer. TC and BH models give more blurred results while the staircase effect produced by TV, CEP2-L², TVL, INFCOV,
Figure 3. Residual images of the denoised examples in Figure 2 by different models.

Figure 4. Plots of middle slices of the denoised images in Figure 2 by different models.

Figure 5. Smoothness preservation ability test. The images in second row are the middle slices of these in the first row.
TVBH and TC models become stronger. The TGV again presents a visually more pleasant smooth result. The quantitative metrics in Table II also validate the better smoothness preservation capability of TGV model over the other 8 variational models. Moreover, in Figure 9, the middle slices of examples in Figure 8 can provide further more detailed comparison between different methods.
Figure 8. Performance comparison of different methods on Figure 5(c).

Figure 9. Plots of middle slices of the denoised images in Figure 8 by different models.
Figure 10. Performance comparison of different methods on real noisy image Figure 10(b) (size 480 × 320).

4.3. Comparison 3

In the previous comparison, the real clean image Figure 10(a) is used to further evaluate different methods. In order to make the denoising evaluation more objective and convincing, the clean image is corrupted by additive Gaussian noise with zero-mean and different variance. Six variance values from 0.005 to 0.03 at 0.005 intervals are used.

In Figure 10, we show the cleaned results of (b) which contains Gaussian noise with 0.015 variance. For each denoised image, the corresponding close-up region is displayed on the bottom right corner for detailed comparison. The visual inspection illustrates TV, CEP2-L2, TVL, INFCON, TVBH and TC models yields staircase effect, whilst TL and BH models smear the object edges in the resulting images. TGV model again gives a visually more pleasant result, total eliminating staircase while well preserving sharp edges. Hence, TGV used as a penalty functional prefers images which appear more natural and leads to a more faithful denoising for real images.

Table III contains the four evaluation metrics for quantitative comparison of different methods on the clean image Figure 10(a) with 6 different noise variance. In order to easily assess these metrics, Figure 11 provide, for each of the 6 experiments, plots of PSNR, SNR, RMSE and SSIM against increasing levels of noise. According to Figure 11, these methods can be quantitatively ranked in a decreasing order as TGV > TC > TVBH > INFCON > TVL > CEP2-L2 > TV > BH > TL.

Table IV demonstrates computational efficiency of different methods. For all the methods, we set stopping criteria as $|E_k - E_{k-1}| / E_k \leq \epsilon$, where $E$ is the value of energy functional of each model, and $\epsilon$ is a small tolerance used to stop iteration. $\epsilon = 10^{-5}$ in all cases. To clearly compare the data in Table IV, we draw the bar charts in Figure 12, where each bar corresponds to a average value of the
Table III. Comparison of PSNR, SNR, RMSE and SSIM using different methods on image Figure 10(a) with 6 different noise variances.

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>PSNR test</th>
<th>SNR test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise variance</td>
<td>0.005</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>RMSE test</th>
<th>SSIM test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise variance</td>
<td>0.005</td>
<td>0.01</td>
</tr>
<tr>
<td>Noisy image</td>
<td>8.4469</td>
<td>9.2784</td>
</tr>
<tr>
<td>TV</td>
<td>5.1521</td>
<td>5.5854</td>
</tr>
<tr>
<td>TL</td>
<td>5.4106</td>
<td>5.7604</td>
</tr>
<tr>
<td>BH</td>
<td>5.2352</td>
<td>5.7011</td>
</tr>
<tr>
<td>CEP2-L²</td>
<td>5.1115</td>
<td>5.5586</td>
</tr>
<tr>
<td>TVL</td>
<td>5.0821</td>
<td>5.5556</td>
</tr>
<tr>
<td>INFCON</td>
<td>5.0549</td>
<td>5.4648</td>
</tr>
<tr>
<td>TVBH</td>
<td>5.0036</td>
<td>5.4678</td>
</tr>
<tr>
<td>TVG</td>
<td>4.9287</td>
<td>5.3898</td>
</tr>
<tr>
<td>TC</td>
<td>4.9403</td>
<td>5.4081</td>
</tr>
</tbody>
</table>

Figure 11. Performance plots of different methods applied to real data with different noise variance. (a) PSNR index; (b) SNR index; (c) RMSE index; (d) SSIM index.
Table IV. Comparison of time per iteration, total CPU time and total iterations of different models on image Figure 10(a) with 6 different noise variances.

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>TV</th>
<th>TL</th>
<th>BH</th>
<th>CEP2-L^2</th>
<th>TVL</th>
<th>INFCON</th>
<th>TVBH</th>
<th>TGV</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>0.0471</td>
<td>0.0390</td>
<td>0.0551</td>
<td>0.0739</td>
<td>0.0581</td>
<td>0.0752</td>
<td>0.0741</td>
<td>0.1003</td>
<td>0.1278</td>
</tr>
<tr>
<td></td>
<td>2.6864</td>
<td>3.6391</td>
<td>3.4713</td>
<td>27.5693</td>
<td>5.1863</td>
<td>17.8236</td>
<td>3.6246</td>
<td>13.5646</td>
<td>7.6698</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>100</td>
<td>63</td>
<td>415</td>
<td>96</td>
<td>235</td>
<td>57</td>
<td>147</td>
<td>70</td>
</tr>
<tr>
<td>0.01</td>
<td>0.0463</td>
<td>0.0340</td>
<td>0.0628</td>
<td>0.0781</td>
<td>0.0515</td>
<td>0.0837</td>
<td>0.0747</td>
<td>0.0956</td>
<td>0.1070</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>121</td>
<td>57</td>
<td>653</td>
<td>90</td>
<td>239</td>
<td>61</td>
<td>218</td>
<td>215</td>
</tr>
<tr>
<td>0.015</td>
<td>0.0457</td>
<td>0.0386</td>
<td>0.0499</td>
<td>0.0659</td>
<td>0.0566</td>
<td>0.0738</td>
<td>0.0735</td>
<td>0.1034</td>
<td>0.1175</td>
</tr>
<tr>
<td></td>
<td>2.6152</td>
<td>5.5070</td>
<td>2.8969</td>
<td>42.4323</td>
<td>4.7385</td>
<td>15.1736</td>
<td>4.3204</td>
<td>18.7648</td>
<td>12.6513</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>156</td>
<td>55</td>
<td>654</td>
<td>91</td>
<td>202</td>
<td>67</td>
<td>204</td>
<td>117</td>
</tr>
<tr>
<td>0.02</td>
<td>0.0447</td>
<td>0.0368</td>
<td>0.0536</td>
<td>0.0702</td>
<td>0.0576</td>
<td>0.0765</td>
<td>0.0734</td>
<td>0.1113</td>
<td>0.1241</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>177</td>
<td>65</td>
<td>702</td>
<td>87</td>
<td>176</td>
<td>86</td>
<td>212</td>
<td>77</td>
</tr>
<tr>
<td>0.025</td>
<td>0.0463</td>
<td>0.0379</td>
<td>0.0528</td>
<td>0.0815</td>
<td>0.0577</td>
<td>0.0816</td>
<td>0.0760</td>
<td>0.1083</td>
<td>0.1303</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>203</td>
<td>69</td>
<td>697</td>
<td>90</td>
<td>170</td>
<td>83</td>
<td>335</td>
<td>130</td>
</tr>
<tr>
<td>0.03</td>
<td>0.0456</td>
<td>0.0368</td>
<td>0.0520</td>
<td>0.0754</td>
<td>0.0552</td>
<td>0.0779</td>
<td>0.0793</td>
<td>0.1102</td>
<td>0.1205</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>185</td>
<td>74</td>
<td>655</td>
<td>91</td>
<td>189</td>
<td>92</td>
<td>262</td>
<td>96</td>
</tr>
</tbody>
</table>

Figure 12. Bar expression of the experimental data in Table IV (i.e. average denoising results of 6 noise variance cases). (a) Time per iteration of different methods; (b) Total CPU time of different methods; (c) Total iterations of different methods.

denoising results produced by its corresponding method on different noise level. Figure 12 illustrates CEP2-L^2 is the slowest method, while TV is the fastest method. Furthermore, TL, INFCON, TGV and TC models are less efficient than BH, TVL and TVBH models.

4.4. Comparison 4

In addition to the proposed split Bregman algorithm, there exist quite a few of approaches, such as the augmented Lagrangian al-
gorithm [10, 15, 22, 23, 35, 36, 43], the primal-dual algorithm [19, 44–47] and etc., to address the difficulties of minimising the high order energy functionals. For example, the authors in [15, 22, 23, 35, 43] used the augmented Lagrangian algorithm to solve the TV, BH, TVL and TC models, while the primal dual has been developed for the TV and TGV models in [44, 46, 47]. Note that both the augmented Lagrangian and split Bregman algorithms employ the alternating direction method of multipliers (ADMM) to optimise their objective functionals and the convergence of ADMM is always guaranteed. In [43], the authors has proven that the two algorithms are in fact equivalent. Thus, in the experiment, we only compare the proposed split Bregman algorithm with the popular primal dual method in terms of both the quality of the result and computational speed. To do so, we choose two representative models from the Table I, i.e. the first order TV model and the second order TGV model, and then apply the split Bregman and primal dual algorithms for each of the two models.

Figure 13 shows the denoised results of Figure 10(b) using the two algorithms. (a) and (b) are the TV denoised results, where one can observe that neither the split bregman nor primal dual can eliminate the staircase artifact. However, the artifact can be totally suppressed by the TGV model, as shown in (c) and (d). Note that the quality of the denoised results by the same model looks
Figure 13. Performance comparison of the split Bregman and primal dual algorithms on Figure 10(b). (a) and (b) are the TV denoised results using the split Bregman and primal dual, respectively; (c) and (d) are the TGV denoised results using the split Bregman and primal dual, respectively.

Figure 14. Comparison of the convergence rate of the split Bregman and primal dual algorithms for the examples in Figure 13. (a) and (b) compare the convergence speed of the two algorithms for the TV model; (c) and (d) compare the convergence speed of the two algorithms for the TGV model.

almost identical. Therefore, it can be concluded that the different numerical algorithms do not significantly influence the final image restoration performance. In contrast, the obvious differences in image quality can be easily found from different models.

In Figure 14, we test the convergence rate of the split Bregman and primal dual algorithms. In (a) and (c), the energy values against CPU runtimes are plotted to compare the energy decreasing rate of the two algorithms. The energy values in (a) and (c) are calculated from the TV (1.1) and TGV (1.9) functionals, and the horizontal axis in (a) and (c) represent the total CPU time (number of iterations times CPU time per iteration). As can be seen from (a) and (c), the numerical energy by the split Bregman use much less time than that by the primal dual algorithm in reaching to a steady state. Moreover, in (b) and (d), the SSIM values are plotted to examine the quality of the denoised images against the number of iterations. Compared with the proposed split Bregman, the primal dual can obtain nearly
same SSIM value but with much more iterations. This experiment validates the split Bregman is faster than the primal dual, and the final results by the two algorithms are quantitatively identical.

5. Conclusion

In this paper, several high-order variational models are presented, and their application for image denoising is demonstrated. We present the detailed discretisation process based on discrete finite different scheme and numerical implementation of fast Fourier transform (FFT)-based split Bregman algorithm for solving the equations. We draw following conclusions about the advantages and disadvantages of these models by extensive comparative experiments in last section.

Among all the models, the first-order TV and second-order TC models are the best in preserving edges of object in the piecewise constant image. However, TV causes image contrast loss while TC not only maintains contrast but also object corners. TGV model yields best denoising results for the piecewise smooth image. It removes the staircase effect associated with TV regulariser. However, TGV is more computational expensive than other models. In addition, we notice that those high order models that involve TV regulariser (i.e. CEP2-L2, TVL, INFCON and TVBH) produce the staircase artifact. Directly applying Laplacian regulariser (i.e. TL model) or bounded Hessian regulariser (i.e. BH model) cannot achieve desirable denoising result. However, combining them with TV regulariser (i.e. CEP2-L2, TVL, INFCON, and TVBH) can improve the quality of resulting image. Moreover, high-order models that use bounded Hessian regulariser (i.e. BH, INFCON and TVBH) outperform their counterparts that use Laplacian regulariser (i.e. TL, CEP2-L2 and TVL), and the former are more efficient than the latter.

Lastly, the proposed split Bregman algorithm is more efficient than the primal dual algorithm, and applying these two algorithms to different image restoration models do not significantly influence the quality of the result.

Acknowledgements

This project is supported by National Natural Science Foundation of China (No.61305045, No.61303079 and No.61170106), Qingdao Science and Technology Development Project (No. 13-1-4-190-jch).

References

W. LU ET AL.


