Non-Blind Image Deblurring Method Using Shear High Order Total Variation Norm

0 Introduction

Image deblurring is a classical problem in low level vision. A blurred image can be considered as a convolution between the latent image and the point spread function (PSF) of the image capture device. The blurring process can be modeled as

\[ f = Ku + n \]

where \( u \in \mathbb{R}^{m \times n} \) is an original image without any form of degradation, \( K \in \mathbb{R}^{m \times n} \) is a degradation matrix, \( f \) is an observed image, and \( n \) is the additive noise.

The goal of image deblurring is to estimate the original image \( u \) from an observed image \( f \) degraded by blur kernel and noise\(^{[1]}\). In this paper, we mainly perform image restoration processing on blur kernel and Gaussian noise. The function of Gaussian noise can be expressed as

\[ n(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

where \( \sigma \) and \( \mu \) represent the standard derivative and average value of the noise distribution \( n \), respectively.

Image deblurring is an ill-posed inverse problem. In other words, its solution neither exists nor is unique, and small disturbances in the data may cause large errors in reconstruction. Hence, in order to overcome this ill-posedness, regularization techniques need to be considered to produce reasonable approximate solutions from noisy data. Rudin-Osher-Fatemi (ROF) model\(^{[2]}\) is one of the most successful regularization methods due to its ability to preserve sharp edges\(^{[3]}\). The corresponding minimization task is as follows:
\[
\min_u \left\{ \frac{\lambda}{2} \| f - Ku \|_2^2 + \| u \|_{TV} \right\}
\]  

(3)

where \( \| \cdot \|_2 \) denotes the Euclidean norm, \( \| \cdot \|_{TV} \) is the discrete TV regularization term. The first term of (3) is called the fidelity term, and the second term is called the regularization term (or penalty) which describes prior knowledge of the image. \( \lambda > 0 \) is the regularization parameter which controls the balance between fidelity term and regularization term. During the last few decades, a series of methods based on total variation and its variants have been developed.

The well-known first-order total variation methods such as FTVd (fast total variation deconvolution)\(^4\) and ADMM (alternating direction method of multipliers) TV\(^5,6\) have also achieved satisfactory results in the image restoration. FTVd is a total variation model based on splitting techniques. In Ref. [7], Chan et al proposed a frame-based deblurring problem using the ADMM algorithm. In Ref. [5], Jiao et al used the ADMM iterative method to solve the TV deblurring problem (ADMM TV). Compared with FTVd, the ADMM TV method can achieve comparable experimental performance without the need for parameter selection.

However, it is well known that TV-based restoration methods suffer from the staircase artifact in flat regions. There are many image restoration methods that have been developed to overcome these shortcomings. A well-known method is to replace the first-order total variation in the regularization term with a second-order (or higher) total variation.

In 2000, Chan et al\(^8\) proposed the high-order TV (HOTV) model to improve the restoration quality. Contrary to TV regularization that penalizes gradient magnitude, regularizations using second-order derivatives do not penalize ramp regions whose intensity varies linearly, and therefore do not force the intensity of the region to remain constant. In other words, these regularizations have ability to potentially recover a broader class of images that consist of more than just piecewise-constant regions\(^9\). In Ref. [10], a second-order Laplace model for image deblurring was proposed by You and Kaveh. In 2003, a second-order derivative regularization\(^11\) (Hessian regularization) model was proposed, called the Lysaker, Lundervold and Tai (LLT) model, which can effectively suppress the staircase artifacts and maintain the smoothness of flat areas of the image. In 2012, Hu et al\(^12\) improved the model for image recovery and designed the maximum-minimum (MM) algorithm to solve the model. Their numerical experiments show that second-order TV has an advantage over classical TV regularization in avoiding staircase artifacts. In Ref. [13], the authors pointed out that high-order TV’s ability to preserve edges became weaker compared with first-order TV regularization and tended to smoothen edges and other small details. Thus, hybrid models that combine higher-order TV with other regularization were proposed in many literatures\(^13-15\).

Among the hybrid models, by replacing the first-order TV with a second-order TV, Zhang et al\(^16\) obtained better results than the model with strongly convex terms presented in Ref. [17]. Furthermore, in Refs. [18] and [19], a combination of first-order and second-order TV priors is used to recover blurred images and the minimization model is solved by the ADMM algorithm. To further improve the restoration performance, Liu et al\(^20\) investigated a spatially adaptive regularization parameter updating scheme. As an adaptive balancing scheme between first-order and second-order TV, Wang et al\(^21\) developed a Poisson denoising framework based on an iterative weighted total generalized variational (TGV) model. In Ref. [22], Liu et al pointed out that TGV can maintain sharp edges and perform well in areas of gradual intensity slopes.

TV or HOTV norm only contains finite difference in horizontal and vertical directions, and this leads to losing some detail information in other directions. In order to incorporate more direction information in gradient domain, in this paper we propose a novel shear high-order TV (SHOTV) norm which contains finite difference in different directions. Compared with higher order TV regularization, the proposed SHOTV regularization method provides a more meaningful description of the gradient information as it contains information about the different directions of the image gradients.

### 1 Preliminary

#### 1.1 Shear Operator

Let \( S_{\theta} \) be a shear operator with the shear angle \( \theta \in [0, \frac{\pi}{4}] \) (see Ref. [23]), and \( d \in \{t', t, b', b\} \):

\[
S_{\theta} : \mathbb{R}^{m \times n} \leftrightarrow \mathbb{R}^{m \times n} : X \mapsto X^{\text{shear}}
\]

(4)

where the \((\hat{i}, \hat{j})\)-th entry of \( X^{\text{shear}} \) is given, for \( \hat{i} = 1, \cdots, m \) and \( \hat{j} = 1, \cdots, n \), by \( X_{i,j}^{\text{shear}} = X_{\tau, \tau} \) with
with different angles

are commutative when 't or b'

parameter

image in different directions at an angle of \( \frac{\pi}{4} \)

\[
\begin{bmatrix}
(i-1 + \left\lfloor \frac{4\pi}{\theta}j \right\rfloor \mod n) + 1, & \text{if } d = 't' \\
(i-1 + \left\lfloor \frac{4\pi}{\theta}(n-j) \right\rfloor \mod n) + 1, & \text{if } d = 'b'
\end{bmatrix}
\]

\[
\begin{bmatrix}
(j-1 + \left\lfloor \frac{4\pi}{\theta}i \right\rfloor \mod n) + 1, & \text{if } d = 't' \\
(j-1 + \left\lfloor \frac{4\pi}{\theta}(n-i) \right\rfloor \mod n) + 1, & \text{if } d = 'b'
\end{bmatrix}
\]

where \( \lfloor \cdot \rfloor \) represents the procedure rounding the real number to the nearest integer in the zero direction. The parameter \( d \) indicates the axis of shear (top 't', bottom 'b', left 'l', right 'r'), and \( \theta_d \) determines the degree of shear. For example, \( S_{\theta_d} X \) means that the image \( X \) is sheared with top axis and \( \theta_d \). In particular, \( \theta_d = 0 \) means \( S_{\theta_d} \) is the identity transformation, i.e., no shear. Figure 1 shows the results of shear operator in different directions at an angle of \( \frac{\pi}{4} \).

\[\text{(a)} \quad \text{(b)} \quad \text{(c)}\]

**Fig. 1** Effect of shear operator \( S_{\theta_d} \) with different angles and directions

(a) Original image \( X \); (b) \( S_{\theta_d} X \) with \( \theta_d = \frac{\pi}{4}, d = 't'; \) (c) \( S_{\theta_d} X \) with \( \theta_d = \frac{\pi}{4}, d = 'r'. \)

1.2 Shear Gradient and Shear HOTV Norm

TV regularization was introduced by Rudin et al\(^2\), for image denoising and further extended to other image restoration tasks. For the gray-scale image \( u \in \mathbb{R}^{m \times n} \), its discrete total variation norm is

\[
TV(u) := \sum_{i=1}^{n} \left[ \left| (Du)_{i,j} \right| \right] = \sum_{i=1}^{n} \left[ \left| (D_x u)_{i,j} \right|^2 + \left| (D_y u)_{i,j} \right|^2 \right]
\]

where \( (Du)_{i,j} = (D_x u)_{i,j} + i, (Du)_{i,j} \) is defined as follows

\[
(D_x u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j}, & \text{if } j < n \\ u_{i,j} - u_{i,j+1}, & \text{if } j = n \end{cases}
\]

TV norm only contains finite difference in horizontal and vertical directions, and this leads to losing some detailed information in other directions.

TV regularization preserves fine features and sharp edges, but it usually produces staircase artifact that is not presented in real images\(^2\). In many applications\(^{26,27}\), some authors have proposed a higher-order TV to reduce staircase artifact. HOTV of \( u \) is defined as

\[
\text{HTV}(u) = \left[ (D^2 u)_{i,j} \right] = \sqrt{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}
\]

where \( D^2 u = (D_x u, D_y u, D_x u, D_y u) \) denotes the second-order difference of the \( ((j-1)n+i)\)-th entry of the vector \( u \). By combining the higher-order finite difference operator and the shear operator, we propose the shear high-order gradient (SHOG) operator by the following way:

\[
D_{S} \triangleq (D_{S_{\theta_l}}, D_{S_{\theta_r}}, D_{S_{\theta_t}}, D_{S_{\theta_b}})
\]

where the superscript T means conjugate transpose. \( S_{\theta_{l}} \) and \( S_{\theta_{r}} \) denote shear along the 'top'(or 'bottom') and 'left'(or 'right') direction with different angles, respectively. With different types of texture images, we choose different directions based on our experience. Then for the gray-scale image \( u \), its discrete shear higher-order total variation (SHOTV) norm is

\[
\text{SHOTV}(u) = \sum_{i=1}^{n} \left\| D_{S} u \right\|_1
\]

The shear gradient has following properties:

**Remark 1** (Matrix form of \( S_{\theta_d} \)) The operator \( S_{\theta_d} \) is linear, so it can be represented as matrix, which is denoted by \( S_{\theta_d} \). Moreover, since \( S_{\theta_d} \) is permutation matrix, it is orthonormal.

**Theorem 1** The shear operator \( S_{\theta_d} \) and the horizontal gradient operator \( D_{\theta} \) are commutative when \( d = 't' \) or \( 'r' \); Similarly, \( S_{\theta_d} \) and the vertical operator \( D_{\theta} \) are commutative when \( d = 't' \) or \( 'b' \), i.e.,

\[
S_{\theta_d} D_{\theta}, S_{\theta_d} = D_{\theta} \text{ when } d = 't' \text{ or } 'r' \quad (11)
\]

\[
S_{\theta_d} D_{\theta}, S_{\theta_d} = D_{\theta} \text{ when } d = 't' \text{ or } 'b' \quad (12)
\]
Proof Firstly, we show that \( S_{\theta} D_{\theta} = D_{\theta} S_{\theta} \), i.e., \( S_{\theta} D_{\theta} S_{\theta} = D_{\theta} \). In fact, for \( u \in \mathbb{R}^{n \times n} \), by the definition of gradient operator and shear operator, we have

\[
(S_{\theta} D_{\theta} u)_{i,j} = (D_{\theta} u)_{i,j}
\]

\[
= \begin{cases} 
  u_{i+1,j} - u_{i,j}, & \text{if } i < n \\
  0, & \text{if } i = n
\end{cases}
\]

where \( \overline{j} \) is defined as in Eq. (7).

\[
(D_{\theta} S_{\theta} u)_{i,j} = \begin{cases} 
  (S_{\theta} u)_{i+1,j} - (S_{\theta} u)_{i,j}, & \text{if } i < n \\
  0, & \text{if } i = n
\end{cases}
\]

where \( j \) is defined as in Eq. (7).

By Eq. (13) and Eq. (14), we obtain \( S_{\theta} D_{\theta} = D_{\theta} S_{\theta} \), i.e., \( S_{\theta}^{T} D_{\theta} S_{\theta} = D_{\theta} \).

Lastly, by the orthogonality of matrix \( S_{\theta} \), we can obtain

\[
S_{\theta}^{T} D_{\theta} S_{\theta} = D_{\theta} S_{\theta} S_{\theta}^{T} D_{\theta} S_{\theta} = D_{\theta} = D_{\theta}
\]

By Theorem 1, the shear gradient operator is essentially a generalization of gradient operator.

The comparison between HOG operator and SHOG operator applied on Barbara image are shown in Fig. 2. It can be observed that more direction information is captured and more detailed edge information can be retained by the SHOG operator. From the local comparison graph, it can be seen that the SHOG operator preserves more information in the texture part of the image.

![Fig. 2 Comparison between 2nd-gradient operator and shear 2nd-gradient operator applied on Barbara image](image)

### 1.3 BCCB Matrices

Under the periodic boundary conditions, both the blurring matrix and gradient matrix are Block-Circulant-with-Circulant-Blocks (BCCB), and thus are diagonalizable by the 2D discrete Fourier transform. In this subsection, we will show that the SHOG matrices are BCCB when the shear angle is \( \frac{\pi}{4} \).

Let \( E_{\theta} \) denote a matrix whose \((i,j)\)-entry is 1, zero elsewhere. Then \{\( E_{\theta} \mid i,j = 1,2,3 \}\) is a set of orthonormal basis of linear space \( \mathbb{R}^{3 \times 3} \). Let \( S_{\theta} \) denote the shear matrix of shear operator under the basis \{\( E_{\theta} \mid i,j = 1,2,3 \}\}, i.e.

\[
S_{\theta} \left( E_{11}, E_{12}, E_{13}, E_{21}, E_{22}, E_{23}, E_{31}, E_{32}, E_{33} \right) = \left( E_{11}, E_{21}, E_{31}, E_{12}, E_{22}, E_{32}, E_{13}, E_{23}, E_{33} \right) S_{\theta}
\]

where \( S_{\theta} \) indicates shearing at angle of \( \frac{\pi}{4} \) along the 'l' direction. \( S_{\theta} \) is a permutation matrix of the following form

\[
S_{\theta} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

Since gradient matrices are BCCB matrices under periodic boundary condition, it can be verified that the \( D_{\theta} = (i=1,2,3,4) \) in formula (10) are also BCCB matrices when shear angle is \( \frac{\pi}{4} \). The shear gradient matrices
\[ D^*_s \rightarrow S^*_s D^*_s S^*_s \quad \text{and} \quad D^*_s = S^*_s D^*_s S^*_s \quad \text{with} \quad \theta_i = \frac{\pi}{4} \]

have the following form:

\[
D^1_s =
\begin{bmatrix}
-2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & -2 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & -2 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & -2 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & -2 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & -2 \\
\end{bmatrix}
\]

\[
D^2_s =
\begin{bmatrix}
1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 \\
\end{bmatrix}
\]

The shear gradient matrices \( D^3_s = S^*_s D^*_s S^*_s \) and

\[ D^4_s = S^*_s D^*_s S^*_s \quad \text{with shear angel} \quad \theta_i = \frac{\pi}{4} \]

have the following form:

\[
D^3_s =
\begin{bmatrix}
1 & 1 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & -1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 \\
-2 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
\end{bmatrix}
\]

\[
D^4_s =
\begin{bmatrix}
0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 \\
0 & -2 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -2 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & -2 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & -2 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & -2 \\
\end{bmatrix}
\]

But unfortunately, when the shear angle \( \theta_i \in (0, \frac{\pi}{4}) \), \( D^*_s (i = 1, 2, 3, 4) \) do not always have BCCB structure. When the shear angle is \( \theta_i = 40^\circ \), the shear matrix \( S^*_s \) has the following form:

\[
D^1_s =
\begin{bmatrix}
-2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & -2 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -2 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & -2 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & -2 \\
\end{bmatrix}
\]

It can be noticed that the above matrix is not a BCCB matrix.

## 2 Proposed Model

Considering the advantages of high-order TV and shear operator, we propose the following optimization model

\[
\min_{u} \frac{\lambda}{2} \| Ku - f \|_2^2 + \omega \| D_s u \|_1 + I_{\Omega}(u) \tag{16}
\]

where the term \( \| \cdot \|_1 \) represents convex \( \ell_1 \) norm regularization. The variables \( \lambda, \omega > 0 \) are regularization parameters that control the data fidelity term and the shear high-order regularization. \( I_\Omega(u) \) is an indication function to impose hard constraints on the objective function defined as

\[
I_\Omega(x) = \begin{cases} 
0, & \text{if} \quad x \in \Omega \\
\infty, & \text{if} \quad x \notin \Omega
\end{cases}
\]

where \( \Omega = [0, 255] \). To minimize problem (16) robustly, we hereby adopt framework of alternating direction method of multipliers (ADMM)\(^{[28]}\). Equation (16) can be converted into a constrained optimization problem:

\[
\min_{u} \frac{\lambda}{2} \| Ku - f \|_2^2 + \omega \| D_s u \|_1 + I_\Omega(u)
\]

s.t. \quad \omega = D_s u \tag{17}

\[
\text{z} = u
\]

The corresponding augmented Lagrangian function of (17) can be written as

\[
L_{\lambda, \omega, \mu}(u, w, z; \lambda, \omega, \mu) = \frac{\lambda}{2} \| Ku - f \|_2^2 + \omega \| D_s u \|_1 + I_\Omega(u)
- \mu^T (w - D_s u) + \frac{\omega}{2} \| w - D_s u \|_2^2
- \mu^T (z - u) + \frac{\lambda}{2} \| z - w \|_2^2
\]

(18)
where $\rho_1, \rho_2$ are penalty parameters. Variables $\mu$, and $\mu_z$ are the Lagrange multipliers associated with the constraints $w = D_1u$ and $z = u$, respectively. ADMM minimizes the Lagrangian function on two sets of variables $u$, $w$ and $z$. Since $w$ and $z$ are decoupled from each other, they can be solved separately.

ADMM is an algorithm using proximal splitting techniques for solving the following convex optimization problem:

$$\min \theta_1(x_1) + \theta_2(x_2)$$

s.t. $A_1x_1 + A_2x_2 = c, x_i \in \mathcal{X}_i, i = 1, 2$ (19)

where $\theta_1: \mathcal{X}_1 \rightarrow \mathbb{R}$ are closed convex functions, $A_1$ are linear transforms, $\mathcal{X}_i$ are nonempty closed convex sets, and $c$ is a given vector.

Part of the appeal of ADMM is the fact that the algorithm lends themselves to parallel implementations. The algorithm is given in Algorithm 1.

**Algorithm 1 ADMM**

1. **Initialization**: Starting point $(x_1^0, x_2^0, \rho^0)$.
2. **Iteration**:
   1. $x_{1i}^{k+1} = \arg \min_{x_1} \theta_1(x_1) + \frac{\delta}{2} \|A_1x_1 + A_2x_2 - c + \rho^k||^2_2$
   2. $x_{2i}^{k+1} = \arg \min_{x_2} \theta_2(x_2) + \frac{\delta}{2} \|A_1x_1 + A_2x_2 - c + \rho^k||^2_2$
   3. $\rho^{k+1} = \rho^k + \delta (A_1x_1^{k+1} + A_2x_2^{k+1} - c)$
   4. $k = k + 1$.

Until a stopping criterion is satisfied.

In this section, we discuss the optimization strategies for solving each sub-problem of (18) one by one.

1) **u Sub-Problem**

The sub-problem of $u$ is a least square problem with the following form

$$u^{k+1} = \arg \min_u \frac{1}{2} \|Ku - f||^2_2 - \mu_1^k \|w - D_1u||^2_2$$

$$+ \frac{\mu_2^k}{2} \|z - D_2u||^2_2$$

$$- \mu_3^k (z^k - u^k) + \frac{\mu_4^k}{2} \|z^k - u^k||^2_2$$

(20)

From the optimality conditions, we have

$$(\lambda K^T K + \rho_1(D_1^T D_1 + \rho_2 I))u^k$$

$$= \lambda K^T f + \rho_1(D_1^T D_1^T (w^k - \mu_3^k) - \mu_2 - \mu_3^k z^k$$

(21)

Considering what was discussed in subsection 1.3, $(D_1^T D_1 + \rho_2 I)$ are BCCB matrices only when the shear angle is $\frac{\pi}{4}$. While the shear angle is not $\frac{\pi}{4}$, computing the inverse of the left hand of (21) in each iteration is costly.

In this paper, we set the shear angle $\theta_s = \frac{\pi}{4}$. $K^T K$ is also BCCB matrix under the periodic boundary condition. As a consequence, Eq. (21) can be efficiently solved by one FFT operation and one inverse FFT operation as

$$u^{k+1} = F^{-1} \left( \frac{\lambda K^T f + \rho_1(D_1^T D_1^T (w^k - \mu_3^k) - \mu_2 - \mu_3^k z^k)}{\lambda F(K^T K) + \rho_1 F((D_1^T D_1^T D_2) + \rho_2)} \right)$$

(22)

where $F$ represents the two-dimensional discrete Fourier transform and $F^{-1}$ is the inverse transform.

2) **w Sub-Problem**

The $w$ sub-problem is a convex second-order de-blurring problem

$$w^{k+1} = \arg \min_w \frac{1}{2} \|w - D_1 u^{k+1}||^2_2$$

$$- \mu_4^k (w^k - D_1 u^{k+1} + \omega)||^2_2$$

$$= \arg \min_w \frac{1}{2} \|w - (D_1 u^{k+1} + \frac{\mu_4^k}{\rho_1})||^2_2 + \omega||^2_2$$

(23)

We write $t = D_1 u^{k+1} + \frac{\mu_4^k}{\rho_1}$ for simplicity, and solve problem (23) using a two-dimensional shrinkage operator. i.e.,

$$w^{k+1} = \text{shrink} \left( \frac{t}{\rho_1}, \frac{\omega}{\rho_1} \right)$$

$$= \max \left\{ \frac{|t| - \frac{\omega}{\rho_1}, 0} \right\} \cdot \text{sign}(t)$$

(24)

3) **z Sub-Problem**

$$z^{k+1} = \arg \min_z \frac{1}{2} \|z - u^{k+1}||^2_2 - \mu_3^k (z^k - u^{k+1}) + I_{ob}(z^k)$$

$$= \arg \min_z \frac{1}{2} \|z - (u^{k+1} + \frac{\mu_3^k}{\rho_2})||^2_2 + I_{ob}(z^k)$$

(25)

The $z$ sub-problem is a projection problem. For an 8-bit image, the range of pixel values can be kept at [0, 255], and the closed-form solution can be obtained by using the projection operator

$$z^{k+1} = \text{Proj}_{ob} \left( u^{k+1} + \frac{\mu_3^k}{\rho_2} \right)$$

$$= \max \left( 255, \text{max}(u^{k+1} + \frac{\mu_3^k}{\rho_2}, 0) \right)$$

(26)

The Lagrangian multipliers are updated as follows

$$\mu_1^{k+1} = \mu_1^k + (D_1 u^{k+1} - w^k)$$

$$\mu_2^{k+1} = \mu_2^k + (u^{k+1} - z^{k+1})$$

(27)

The proposed algorithm is named SHOTV and
shown in Algorithm 2.

Algorithm 2 SHOTV

1 Initialization: Starting point \((u^0, w^0, z^0, \mu^0, \lambda^0), \rho_0, \rho_1 > 0, \lambda_0 > 0\).

2 Iteration:
   1) Compute \(u^{k+1}\) according to (21).
   2) Compute \(w^{k+1}\) according to (24).
   3) Compute \(z^{k+1}\) according to (26).
   4) Update \(\mu_{u}^{k+1}\) according to (27).
   5) Update \(\mu_{z}^{k+1}\) according to (28).
   6) Check the stopping criterion \(\frac{\|u^k - u^{k+1}\|}{\|u^k\|} < \delta\).
   7) \(k = k + 1\).
   8) end

3 Numerical Experiments

In this section, we present various numerical results to illustrate the performance of the proposed algorithm for non-blind image deblurring. All experiments are carried out on Windows 10 64-bit and Matlab 2019b running on a desktop equipped with an Intel Core i5-10300 H CPU 2.5 GHz and 16 GB of RAM. The source code for all competing methods was obtained from the original authors and we use the default parameter settings.

3.1 Experiment Setting

The test images are shown in Fig. 3, and the pixel values of the images are normalized to \([0, 1]\) for simplicity. Both Peak Signal to Noise Ratio (PSNR) and Structural Similarity metrics (SSIM) are used to evaluate the quality of the reconstructed images. They are defined as follows:

\[
\text{PSNR} (\text{dB}) = 10 \log_{10} \left( \frac{(\text{Max}_u)^2}{\|f - u\|^2} \right)
\]

\[
\text{SSIM} = \frac{(2\hat{f} \hat{u})(2\sigma_{fu} + c_2)}{\left( \hat{f}^2 + \hat{u}^2 + c_1 \right) \left( \sigma_f^2 + \sigma_u^2 + c_2 \right)}
\]

where \(u\) is the clean image, \(f\) is the recovered image, \(\text{Max}_u\) is the largest possible value of the image \(u\). \(\hat{u}\) and \(\hat{f}\) are the mean values of images \(u\) and \(f\), \(\sigma_u\) and \(\sigma_f\) are the standard variance of images \(u\) and \(f\), respectively, and \(\sigma_{uf}\) is the covariance of \(u\) and \(f\), \(c_1, c_2 \geq 0\) are constants. In general, high PSNR and SSIM values imply better image quality.

3.2 Parameter Selection Discussion

In this subsection, we focus on the choice of parameters \(\rho_1, \rho_2\) and \(\lambda\). For simplicity, we set the parameter \(\rho_1 = \rho_2\). Then we investigate the sensitivity of the parameter \(\rho_1\) and \(\lambda\). We test the selection of \(\rho_1\) on both House and Parrot images. They are corrupted by motion blur (fspecial(‘motion’, 35, 50)) and Gaussian blur (fspecial(‘gaussian’, [9,9], 9)). Figure 4 shows the change of PSNR values versus the parameter \(\rho_1\) under three different noise levels. We empirically set the parameters \(\rho_1 \in [5, 30]\) in the subsequent experiments.

Then, we discuss the selection of the parameter \(\lambda\). Our experiments are tested on Pirate and Barbara images with three different levels of noise and two different blur kernels. For simplicity, we set the parameter \(\rho_1 = \rho_2 = 10\). Figure 5 shows the curves of PSNR value with respect to parameter \(\lambda\) under three different noise levels. For the parameter \(\lambda\) in the subsequent experiments, the empirical value is taken to be in the range \([800, 2000]\). As a matter of fact, the above empirical parameter settings

Fig. 3 The test images
settings are also generally valid for other test images.

3.3 Image Deblurring

In this subsection, we report the performance of the proposed SHOTV method for image deblurring and compare it with some leading deblurring methods, including FTVd\cite{9}, ADMM frame\cite{3}, ADMM TV\cite{5} and HOTV\cite{18}.

First, we test the $9\times9$ Gaussian blur with a standard deviation of 9. The blurred image is generated using the MATLAB function “imfilter” with periodic boundary conditions. Then, the blurred image is corrupted with zero-mean additive white Gaussian noise.
\( \sigma^2 = 0.001 \). The PSNR and SSIM values for the deblurring experiments are reported in Table 1 and Table 2. The numbers in bold indicate the best performance. One can observe that the proposed method outperforms the other competing methods. We choose Fingerprint, Cameraman, Hill and Shirt images for Gaussian deblurring and denoising comparison, and the recovered images are shown in Fig. 6. From the comparison, we can see that the restored images by our method retain more texture and have fewer artifacts. As can be seen from the Shirt image in Fig. 6, our proposed algorithm retains more texture information through the shear operator, resulting in a significant improvement in image deblurring.

Similarly, we test the motion blur with a len size 35 in the blur kernel and a rotation angle 50°. The blurred image is generated using the MATLAB function “imfilter” with periodic boundary conditions, and the blurred image is corrupted by zero-mean additive white Gaussian noise with \( \sigma^2 = 0.001 \). The PSNR and SSIM results for all competing deblurring methods are reported in Table 2. One can observe that the proposed SHOTV method can obtain better result than other competing methods. The blurred images and the deblurred images are shown in Fig. 7. Compared with other methods, our algorithm can preserve more texture structure and the deblurred image contains fewer artifacts compared with first-order TV.

### 3.4 Convergence Analysis

In order to verify the convergence of the algorithm numerically, we apply Algorithm 2 to four different images corrupted by different blur kernels and additive white Gaussian noise with deviation \( \sigma^2 = 0.001 \), where Gaussian blur is applied to Fingerprint and Shirt images, and Gaussian blur is applied to Barbara and Zebra images. We use the equation \( \frac{\| u^{k+1} - u^k \|_1}{\| u^k \|_1} \) to calculate the relative error curve for each iteration of the recovery image of Algorithm 2. As shown in Fig. 8, the relative error decreases as the number of iterations increases, which numerically illustrates the convergence of the algorithm.

<table>
<thead>
<tr>
<th>Image</th>
<th>Noisy PSNR/dB</th>
<th>SSIM</th>
<th>FTVD PSNR/dB</th>
<th>SSIM</th>
<th>ADMM-frame PSNR/dB</th>
<th>SSIM</th>
<th>ADMM-TV PSNR/dB</th>
<th>SSIM</th>
<th>HOTV PSNR/dB</th>
<th>SSIM</th>
<th>SHOTV PSNR/dB</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint</td>
<td>18.362 0.410</td>
<td>29.240 0.937</td>
<td>28.698 0.930</td>
<td>28.864 0.931</td>
<td>29.495 0.938</td>
<td>30.100 0.948</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameraman</td>
<td>20.865 0.623</td>
<td>29.694 0.895</td>
<td>29.823 0.811</td>
<td>30.559 0.855</td>
<td>29.825 0.896</td>
<td>30.455 0.902</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill</td>
<td>25.147 0.580</td>
<td>31.792 0.688</td>
<td>31.421 0.815</td>
<td>32.060 0.845</td>
<td>32.586 0.868</td>
<td>33.096 0.879</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shirt</td>
<td>11.845 0.156</td>
<td>20.076 0.903</td>
<td>21.580 0.910</td>
<td>20.790 0.872</td>
<td>20.315 0.904</td>
<td>21.248 0.932</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brodatz</td>
<td>15.206 0.161</td>
<td>22.178 0.823</td>
<td>22.897 0.842</td>
<td>22.413 0.835</td>
<td>21.252 0.772</td>
<td>23.095 0.851</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monarch</td>
<td>19.378 0.619</td>
<td>29.935 0.923</td>
<td>28.050 0.850</td>
<td>30.577 0.930</td>
<td>29.875 0.930</td>
<td>30.763 0.935</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Numbers in bold indicate the best performance.

<table>
<thead>
<tr>
<th>Image</th>
<th>Noisy PSNR/dB</th>
<th>SSIM</th>
<th>FTVD PSNR/dB</th>
<th>SSIM</th>
<th>ADMM-frame PSNR/dB</th>
<th>SSIM</th>
<th>ADMM-TV PSNR/dB</th>
<th>SSIM</th>
<th>HOTV PSNR/dB</th>
<th>SSIM</th>
<th>SHOTV PSNR/dB</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lion</td>
<td>19.251 0.299</td>
<td>27.818 0.851</td>
<td>26.530 0.838</td>
<td>27.047 0.850</td>
<td>27.251 0.860</td>
<td>27.672 0.875</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parrot</td>
<td>16.242 0.529</td>
<td>32.923 0.903</td>
<td>27.534 0.838</td>
<td>31.381 0.917</td>
<td>31.402 0.896</td>
<td>31.990 0.920</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Starfish</td>
<td>18.068 0.441</td>
<td>32.664 0.919</td>
<td>29.172 0.863</td>
<td>31.131 0.912</td>
<td>31.975 0.917</td>
<td>32.384 0.931</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brodatz</td>
<td>14.590 0.171</td>
<td>27.383 0.923</td>
<td>24.718 0.892</td>
<td>25.841 0.897</td>
<td>26.304 0.901</td>
<td>26.611 0.911</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zebra</td>
<td>16.468 0.339</td>
<td>29.294 0.866</td>
<td>25.671 0.802</td>
<td>29.329 0.872</td>
<td>27.634 0.856</td>
<td>28.647 0.884</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Straw</td>
<td>18.529 0.264</td>
<td>28.400 0.932</td>
<td>27.741 0.925</td>
<td>27.542 0.908</td>
<td>27.340 0.921</td>
<td>28.836 0.948</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barbara</td>
<td>20.064 0.491</td>
<td>30.528 0.881</td>
<td>28.570 0.823</td>
<td>28.721 0.821</td>
<td>29.663 0.848</td>
<td>30.651 0.892</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 6 Restoration of four degraded images with Gaussian blur and comparison with other algorithms
From left to right: the noisy images, FTVd, ADMM-Frame, ADMM-TV, HOTV, and SHOTV; each value in parentheses represents the corresponding SSIM value of the restored image.

Fig. 7 Restoration of four degraded images with Motion blur and comparison with other algorithms
From left to right: the noisy images, FTVd, ADMM-Frame, ADMM-TV, HOTV, and SHOTV; each value in parentheses represents the corresponding SSIM value of the restored image.

4 Conclusion

In this paper, a new SHOG operator is proposed by combining the high-order gradient operator and the shear operator. By both theoretical analysis and experiments, we show that the proposed SHOG operator incorporates more directionality and can detect more abundant edge...
information. We extend the HOTV norm to the SHOTV norm based on the SHOG operator, and then employ SHOTV norm as the regularization term to establish an image deblurring model. We also study some properties of the SHOG operator, and show that the SHOG matrices are Block-Circulant-with-Circulant-Blocks (BCCB) when the shear angle is $\frac{\pi}{4}$. ADMM scheme is exploited to solve the proposed model efficiently. Extensive numerical experiments demonstrate the superiority of the proposed method in terms of visual quality and quantitative metrics.

References


