Liu *et al. EURASIP Journal on Image and Video Processing* (2016) 2016:18
DOI 10.1186/s13640-016-0118-5

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

New inexact explicit thresholding/ shrinkage formulas for inverse problems with overlapping group sparsity

Gang Liu[1,2], Ting-Zhu Huang[2*], Xiao-Guang Lv[3] and Jun Liu[2]

Abstract

The least-square regression problems or inverse problems have been widely studied in many fields such as compressive sensing, signal processing, and image processing. To solve this kind of ill-posed problems, a regularization term (i.e., regularizer) should be introduced, under the assumption that the solutions have some specific properties, such as sparsity and group sparsity. Widely used regularizers include the ℓ_1 norm, total variation (TV) semi-norm, and so on. Recently, a new regularization term with overlapping group sparsity has been considered. Majorization minimization iteration method or variable duplication methods are often applied to solve them. However, there have been no direct methods for solving the relevant problems due to the difficulty of overlapping. In this paper, we proposed new inexact explicit shrinkage formulas for one class of these relevant problems, whose regularization terms have translation invariant overlapping groups. Moreover, we apply our results to TV deblurring and denoising problems with overlapping group sparsity. We use alternating direction method of multipliers to iteratively solve them. Numerical results verify the validity and effectiveness of our new inexact explicit shrinkage formulas.

Keywords: Overlapping group sparsity, Total variation, ADMM, Image deblurring, Regularization, Explicit shrinkage formula

1 Introduction

The least-square regression problems or inverse problems have been widely studied in many fields such as compressive sensing, signal processing, image processing, statistics, and machine learning. Regularization terms with sparse representations (for instance, the ℓ_1 norm regularizer) have been developed into an important tool in these applications, and a list of methods have been proposed [1–4]. These methods are based on the assumption that signals or images have a sparse representation, that is, only containing a few nonzero entries. The corresponding task is to solve the following problem

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a given vector, $\mathbf{z} \in \mathbb{R}^n$, $\|\mathbf{z}\|_p = \left(\sum_{i=1}^n |z_i|^p\right)^{\frac{1}{p}}$ with $(p = 1, 2)$ represents the ℓ_p norm of vector z. The first term of (1) is called the regularization term, the second term is called the fidelity term, and $\beta > 0$ is the regularization parameter.

To further improve the solutions, recent studies suggested to go beyond sparsity and took into account additional information about the underlying structure of the solutions [1, 2, 5]. In particular, a wide class of solutions which have specific "group sparsity" structure are considered. In this case, a group sparse vector can be divided into groups of components satisfying (a) only a few of groups contain nonzero values and (b) these groups are not needed to be sparse. If we insert such group vectors into a matrix as row vectors of the matrix, this matrix will only have few nonzero rows and these rows may be not sparse. This property is called "group sparsity" or "joint sparsity," and many literature have considered these new sparse problems [1–6]. These problems can be formulated as the following expression.

*Correspondence: tingzhuhuang@126.com
[2] Research Center for Image and Vision Computing/School of Mathematical Sciences, University of Electronic Science and Technology of China, Xiyuan Ave, 611731 Chengdu, China
Full list of author information is available at the end of the article

$$\min_{\mathbf{z}} \quad \sum_{i=1}^{r} \|\mathbf{z}[\,i]\,\|_2 + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2, \tag{2}$$

where $|z_i|$ is the absolute value of z_i, and $\mathbf{z}[\,i]$ is the ith group of z with $\mathbf{z}[\,i] \cap \mathbf{z}[\,j] = \emptyset$ and $\bigcup_{i=1}^{r} \mathbf{z}[\,i] = \mathbf{z}$. "Group sparsity" solutions from (2) have better representation and have been widely studied both for convex and nonconvex cases [1, 3, 6–9].

More recently, overlapping group sparsity (OGS) have been considered [1, 3, 10–19]. These methods are based on the assumption that signals or images have a special sparse representation with OGS. The task is to solve the following problem

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_{2,1} + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2, \tag{3}$$

where $\|\mathbf{z}\|_{2,1} = \sum_{i=1}^{n} \|(z_i)_g\|_2$ is the generalized $\ell_{2,1}$ norm. Here, each $(z_i)_g$ is a group vector containing s (called group size) elements that surrounding the ith entry of $\mathbf{z}$. For example, $(z_i)_g = (z_{i-1}, z_i, z_{i+1})$ with $s = 3$. In this case, $(z_i)_g$, $(z_{i+1})_g$, and $(z_{i+2})_g$ contain the $(i+1)$th entry of $\mathbf{z}$, z_{i+1}, which means "overlapping" different from the form of group sparsity in (2). In particular, if $s = 1$, the generalized $\ell_{2,1}$ norm degenerates into the original ℓ_1 norm, and the relevant regularization problem (3) degenerates to (1).

To be more general, researchers considered the weighted generalized $\ell_{2,1}$ norm $\|z\|_{w,2,1} = \sum_{i=1}^{n} \|w_g \circ (z_i)_g\|_2$ (we only consider that each group has the same weight, which means translation invariant) instead of the former generalized $\ell_{2,1}$ norm. The task can be extended to

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_{w,2,1} + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2, \tag{4}$$

where w_g is a nonnegative real vector with the same size as $(z_i)_g$ and "$\circ$" is the point-wise product or Hadamard product. For instance, $w_g \circ (z_i)_g = ((w_g)_1 z_i, (w_g)_2 z_{i+1}, (w_g)_3 z_{i+2})$ with $s = 3$ as the former example. In particular, the weighted generalized $\ell_{2,1}$ norm degenerates into the generalized $\ell_{2,1}$ norm if each entry of w_g equals to 1.

The problems (3) and (4) have been considered in [1, 14–19]. They solve the relevant problems by using variable duplication methods (variable splitting, latent/auxilliary variables, etc.). For example, Deng et al. in [1] introduced a diagonal matrix G in their method. However, matrix G is not easy to find and would break the structure of the coefficient matrix, which makes the difficulty of solving solutions under high-dimensional vector cases. Moreover, it is difficult to extend this method to the matrix case.

Considering the matrix case of the problem (4), we can get

$$\min_{A} \quad \|A\|_{W,2,1} + \frac{\beta}{2}\|A-X\|_F^2, \tag{5}$$

where $X, A \in \mathbb{R}^{m\times n}$, $\|A\|_{W,2,1} = \sum_{i=1}^{m}\sum_{j=1}^{n} \|W_g \circ (A_{i,j})_g\|_F$. Here, each $(A_{i,j})_g$ is a group matrix containing $K_1 \times K_2$ (called group size) elements that surround the (i,j)th entry of A. It is defined by

$$W_g \circ (A_{i,j})_g = \begin{bmatrix} (W_g)_{1,1}A_{i-l_1,j-l_2} & (W_g)_{1,2}A_{i-l_1,j-l_2+1} & \cdots & (W_g)_{1,K_2}A_{i-l_1,j+r_2} \\ (W_g)_{2,1}A_{i-l_1+1,j-l_2} & (W_g)_{2,2}A_{i-l_1+1,j-l_2+1} & \cdots & (W_g)_{2,K_2}A_{i-l_1+1,j+r_2} \\ \vdots & \vdots & \ddots & \vdots \\ (W_g)_{K_1,1}A_{i+r_1,j-l_2} & (W_g)_{K_1,2}A_{i+r_1,j-l_1+1} & \cdots & (W_g)_{K_1,K_2}A_{i+r_1,j+r_2} \end{bmatrix} \in \mathbb{R}^{K_1\times K_2},$$

where $l_1 = \lfloor \frac{K_1-1}{2} \rfloor$, $l_2 = \lfloor \frac{K_2-1}{2} \rfloor$, $r_1 = \lfloor \frac{K_1}{2} \rfloor$, $r_2 = \lfloor \frac{K_2}{2} \rfloor$ (with $l_1+r_1+1 = K_1$ and $l_2+r_2+1 = K_2$) and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. In particular, if $K_2 = 1$, this problem degenerates to the former vector case (4). If $K_1 = K_2 = 1$, this problem degenerates to the original ℓ_1 regularization problem (1) for the matrix case.

Much more recently, Chen et al. [10] considered the problem (5) when $(W_g)_{i,j} \equiv 1$ for $i = 1, \cdots, K_1$, $j = 1, \cdots, K_2$. They used an iterative algorithm to solve this problem based on the principle of majorization minimization (MM). Their experiments showed that their method on solving the problem (3) is more efficient than other variable duplication methods (variable splitting, latent/auxilliary variables, etc.) [1, 14–19]. Although their method is efficient for computing the solution, it may cost more time since the solution is obtained after many iterations rather than a direct step computation.

To our knowledge, there have been no direct methods for solving the relevant problems (3), (4), and (5). In this paper, we propose new inexact explicit shrinkage formulas for solving them directly, which are direct methods, and faster or more convenient than other methods, for instance, the most recently MM iteration method [10]. This new direct method can save more time than the MM method in [10] while getting very similar solutions. Numerical results are given to show the effectiveness of our new explicit shrinkage formulas. Moreover, the new method can be used in application for solving the subproblem in many other OGS problems, such as the wavelet regularizer with OGS in compressive sensing and the total variation (TV) regularizer with OGS in image restoration [11, 12]. For example, we will apply our results to image restoration using TV with OGS in this work similarly as [11, 12]. Moveover, we expend the works, only considering ATV in [11, 12] to both ATV and ITV (more detail can be referred to Section 4). Numerical results will show that our method can save more time with getting very similar results.

The outline of the paper is as follows. In Section 2, we deduce our explicit shrinkage formulas in detail for the OGS problems (3), (4), and (5). In Section 3, we propose some extensions for these shrinkage formulas. In Section 4, we apply our results to image deblurring and denoising problems with OGS TV. Numerical results are given in Section 5. Finally, we conclude this paper in Section 6.

2 OGS shrinkage

2.1 Classic shrinkage

In this subsection, we will review the original shrinkage formulas and their principles, since our new explicit OGS shrinkage formulas are based on them. Firstly, we give the following definition.

Definition 1. Define shrinkage mappings Sh_1 and Sh_2 from $\mathbb{R}^N \times R^+$ to $\mathbb{R}^N$ by

$$\mathrm{Sh}_1(\mathbf{x},\beta)_i = \mathrm{sgn}(x_i)\max\left\{|x_i| - \frac{1}{\beta}, 0\right\}, \tag{6}$$

$$\mathrm{Sh}_2(\mathbf{x},\beta) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}\max\left\{\|\mathbf{x}\|_2 - \frac{1}{\beta}, 0\right\}, \tag{7}$$

where the expression is taken to be zero when the second factor is zero, and "sgn" represents the signum function indicating the sign of a number. More precisely, $\mathrm{sgn}(x) = 0$ if $x = 0$, $\mathrm{sgn}(x) = -1$ if $x < 0$, and $\mathrm{sgn}(x) = 1$ if $x > 0$.

The shrinkage (6) is known as soft thresholding and occurs in many algorithms related to sparsity since it is the proximal mapping for the ℓ_1 norm. The shrinkage (7) is known as high-dimensional shrinkage formula.

Now, we consider solving the following problems

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_p + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2, \ p = 1, 2. \tag{8}$$

The minimizer of (8) with $p = 1$ is the following equation.

$$\arg\min_{\mathbf{z}} \quad \|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 = \mathrm{Sh}_1(\mathbf{x},\beta). \tag{9}$$

Due to the additivity and separability of both the ℓ_1 norm and the square of the ℓ_2 norm, Eq. (9) can be deduced easily by the following formula:

$$\min_{\mathbf{z}} \quad \|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 = \sum_{i=1}^{n} \min_{z_i} |z_i| + \frac{\beta}{2}|z_i - x_i|^2. \tag{10}$$

The minimizer of (8) with $p = 2$ is the following equation.

$$\arg\min_{\mathbf{z}} \quad \|\mathbf{z}\|_2 + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 = \mathrm{Sh}_2(\mathbf{x},\beta). \tag{11}$$

This formula is deduced by the Euler equation of (8) with $p = 2$. In particular, without considering $\mathbf{x} = \mathbf{0}$ or $\mathbf{z} = \mathbf{0}$, we obtain the following Euler equation.

$$\beta\,(\mathbf{z}-\mathbf{x}) + \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \ni \mathbf{0}, \tag{12}$$

$$\left(1 + \frac{1}{\beta}\frac{1}{\|\mathbf{z}\|_2}\right)\mathbf{z} - \mathbf{x} \ni \mathbf{0}. \tag{13}$$

We can easily get that the necessary condition is that the vector $\mathbf{z}$ is parallel to the vector $\mathbf{x}$. That is, $\frac{\mathbf{z}}{\|\mathbf{z}\|_2} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. Substituting into (13) and considering $\mathbf{x} = \mathbf{0}$ or $\mathbf{z} = \mathbf{0}$, we obtain the formula (7). More details can be referred to [20, 21].

Our new explicit OGS shrinkage formulas are based on these observations, especially the properties of additivity, separability, and parallelity.

Remarks 1. The problem (2) is easy to be solved by a simple shrinkage formula, which is not used in this work. More details can be referred to [1, 22, 23].

2.2 The OGS shrinkage formulas

Now, we focus on the problem (3) firstly. The difficulty of this problem is "overlapping." Therefore, we must take some special techniques to avoid "overlapping." That is the point of our new explicit OGS shrinkage formulas.

It is obvious that the first term of the problem (3) is additive and separable. So if we find some relative rules such that the second term of the problem (3) has the same properties with the same variable as the first term, one approximate solution of (3) can be easily found similar as classic shrinkage.

Assuming periodic boundary condition (PBC) (boundary condition (BC) is necessary in OGS functionals) is used here, we observe that each entry z_i of the vector $\mathbf{z}$ would appear exactly s times in the first term. Therefore, to hold on the uniformity of vectors $\mathbf{z}$ and $\mathbf{x}$, we need to multiply the second term by s. To maintain the invariability of the problem (3), after some manipulations, we have

$$\begin{aligned} f_m(\mathbf{z}) &= \min_{\mathbf{z}} \|\mathbf{z}\|_{2,1} + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \|(z_i)_g\|_2 + \frac{\beta}{2s} s\|\mathbf{z}-\mathbf{x}\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \|(z_i)_g\|_2 + \frac{\beta}{2s}\sum\nolimits_{i=1}^{n} \|(z_i)_g - (x_i)_g\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \left(\|(z_i)_g\|_2 + \frac{\beta}{2s}\|(z_i)_g - (x_i)_g\|_2^2\right), \end{aligned} \tag{14}$$

where $(x_i)_g$ is similarly defined as $(z_i)_g$ in Section 1.

For example, we set $s = 3$ and define $(z_i)_g = (z_i, z_{i+1}, z_{i+2})$. The generalized $\ell_{2,1}$ norm $\|\mathbf{z}\|_{2,1}$ can be treated as the generalized ℓ_1 norm of generalized points, whose entry $(z_i)_g$ is also a vector, and the absolute value of each entry is treated as the ℓ_2 norm of $(z_i)_g$. See Fig. 1a intuitively, where the top line is the vector $\mathbf{z}$, the rectangles with dashed line are original $(z_i)_g$, and the rectangles with solid line are the generalized points. In the case of PBC, we know that each line of Fig. 1a is translated equal. We operate the vector $\mathbf{x}$ similarly as the vector $\mathbf{z}$. Putting these generalized points (rectangles with solid line in the figure) as the columns of a matrix, we can regard $s\|\mathbf{z}-\mathbf{x}\|_2^2$ as the matrix Frobenius norm $\| \left[(z_i)_g\right] - \left[(x_i)_g\right] \|_F^2$, where $\left[(z_i)_g\right]$ is a matrix as in Fig. 1a with every line being its row ($\left[(x_i)_g\right]$ is similar). This is why the second equality in (14) holds.

Therefore, generally, for each i of the last line of (14), from the Eqs. (7) and (11), we can obtain

$$\begin{aligned} &\arg\min_{(z_i)_g} \|(z_i)_g\|_2 + \frac{\beta}{2s}\|(z_i)_g - (x_i)_g\|_2^2 = \mathrm{Sh}_2\left((x_i)_g, \frac{\beta}{s}\right), \\ &(z_i)_g = \max\left\{\|(x_i)_g\|_2 - \frac{s}{\beta}, 0\right\}\frac{(x_i)_g}{\|(x_i)_g\|_2}, \\ &(z_i)_g = \left(\left[(z_i)_g\right]_1, \left[(z_i)_g\right]_2, \ldots, \left[(z_i)_g\right]_s\right). \end{aligned} \tag{15}$$

Similarly as Fig. 1a, for each i, the ith entry x_i (or z_i) of the vector $\mathbf{x}$ (or $\mathbf{z}$) may appear s times, so we need to compute each z_i for s times in s different groups.

However, the results from (15) are not able to be satisfied simultaneously, because the results z_i in s different groups are different from (15). Moreover, for each i of the last line in (14), the result (15) is given by that the vector $(z_i)_g$ is parallel to the vector $(x_i)_g$ due to Section 2.1.

Notice this point and ignore that $(z_i)_g = \mathbf{0}$ or $(x_i)_g = \mathbf{0}$, particularly for $s = 4$ and $(z_i)_g = (z_{i-1}, z_i, z_{i+1}, z_{i+2})$, the vector $\mathbf{z}$ can be split as follows,

$$\begin{array}{rl} \mathbf{z} = & (z_1,\ z_2,\ z_3,\ z_4,\ z_5,\ \cdots,\ z_{n-2},\ z_{n-1},\ z_n\ ) \\ = +\frac{1}{4} & (z_1,\ z_2,\ z_3,\ 0,\ 0,\ \cdots,\ 0,\ 0,\ z_n\ ) \\ +\frac{1}{4} & (z_1,\ z_2,\ z_3,\ z_4,\ 0,\ \cdots,\ 0,\ 0,\ 0\ ) \\ +\frac{1}{4} & (0,\ z_2,\ z_3,\ z_4,\ z_5,\ \cdots,\ 0,\ 0,\ 0\ ) \\ + & \cdots \\ +\frac{1}{4} & (z_1,\ 0,\ 0,\ 0,\ 0,\ \cdots,\ z_{n-2},\ z_{n-1},\ z_n\ ) \\ +\frac{1}{4} & (z_1,\ z_2,\ 0,\ 0,\ 0,\ \cdots,\ 0,\ z_{n-1},\ z_n\ ). \end{array} \tag{16}$$

Let $(z_i)_g' = (0, \cdots, 0, z_{i-1}, z_i, z_{i+1}, z_{i+2}, 0, \cdots, 0)$ be the expansion of $(z_i)_g$, with $(z_1)_g' = (z_1, z_2, z_3, 0, \cdots, 0, z_n)$, $(z_{n-1})_g' = (z_1, 0, 0, \cdots, 0, z_{n-2}, z_{n-1}, z_n)$, $(z_n)_g' = (z_1, z_2, 0, \cdots, 0, z_{n-1}, z_n)$. Let $(x_i)_g' = (0, \cdots, 0, x_{i-1}, x_i, x_{i+1}, x_{i+2}, 0, \cdots, 0)$ be the expansion of $(x_i)_g$ similarly as $(z_i)_g'$. Then, we have $\mathbf{z} = \frac{1}{4}\sum_{i=1}^n (z_i)_g'$, and $\mathbf{x} = \frac{1}{4}\sum_{i=1}^n (x_i)_g'$. Moreover, we can easily obtain that $\|(z_i)_g'\|_2 = \|(z_i)_g\|_2$ and $\|(x_i)_g'\|_2 = \|(x_i)_g\|_2$ for every i.

On one hand, the Euler equation of $f_m(\mathbf{z})$ (with $s = 4$) is given by

$$\beta\left(\mathbf{z} - \mathbf{x}\right) + \frac{(z_1)_g'}{\|(z_1)_g'\|_2} + \cdots + \frac{(z_n')_g}{\|(z_n')_g\|_2} \ni \mathbf{0}, \tag{17}$$

$$\begin{aligned} \frac{\beta}{4}\sum_{i=1}^n \left((z_i)_g' - (x_i)_g'\right) + \frac{(z_1)_g'}{\|(z_1)_g'\|_2} + \cdots + \frac{(z_i)_g'}{\|(z_i)_g'\|_2} + \cdots \\ + \frac{(z_n)_g'}{\|(z_n)_g'\|_2} \ni \mathbf{0}. \end{aligned} \tag{18}$$

From the deduction of the two-dimensional shrinkage formula (7) in Section 2.1, we know that the necessary condition of minimizing the ith term of the last line in (14)

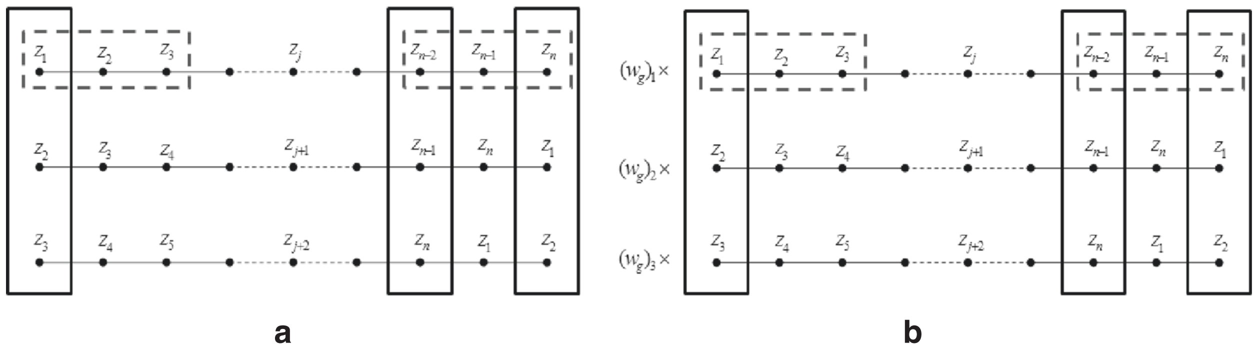

Fig. 1 Vector case. The *top line* is the vector **z**, the *rectangles with dashed line* are original group vector $(z_i)_g$, and the *rectangles with solid line* are the generalized points, which changes the overlapping to non-overlapping. **a** Vector. **b** Weighted vector

is that $(z_i)_g$ is parallel to $(x_i)_g$. That is, $(z_i)'_g$ is parallel to $(x_i)'_g$ for every i. For example,

$$(z_2)'_g = (z_1, z_2, z_3, z_4, 0, \cdots, 0, 0) // (x_2)'_g = (x_1, x_2, x_3, x_4, 0, \cdots, 0, 0). \tag{19}$$

Then, we obtain $\frac{(z_i)'_g}{\|(z_i)'_g\|_2} = \frac{(x_i)'_g}{\|(x_i)'_g\|_2}$. Therefore, (18) can be changed to

$$\frac{\beta}{4}\sum_{i=1}^{n}\left((z_i)'_g - (x_i)'_g\right) + \frac{(x_1)'_g}{\|(x_1)'_g\|_2} + \cdots + \frac{(x_i)'_g}{\|(x_i)'_g\|_2} + \cdots + \frac{(x_n)'_g}{\|(x_n)'_g\|_2} \ni \mathbf{0}, \tag{20}$$

$$\beta\,(\mathbf{z} - \mathbf{x}) + \frac{(x_1)'_g}{\|(x_1)'_g\|_2} + \cdots + \frac{(x_i)'_g}{\|(x_i)'_g\|_2} + \cdots + \frac{(x_n)'_g}{\|(x_n)'_g\|_2} \ni \mathbf{0}, \tag{21}$$

$$\mathbf{z} \ni \mathbf{x} - \frac{1}{\beta}\left(\frac{(x_1)'_g}{\|(x_1)'_g\|_2} + \cdots + \frac{(x_i)'_g}{\|(x_i)'_g\|_2} + \cdots + \frac{(x_n)'_g}{\|(x_n)'_g\|_2}\right), \tag{22}$$

for each component, we obtain

$$z_i \ni x_i - \frac{1}{\beta}\left(\frac{x_i}{\|(x_{i-2})'_g\|_2} + \frac{x_i}{\|(x_{i-1})'_g\|_2} + \frac{x_i}{\|(x_i)'_g\|_2} + \frac{x_i}{\|(x_{i+1})'_g\|_2}\right). \tag{23}$$

Therefore, when $(z_i)_g \neq \mathbf{0}$ and $(x_i)_g \neq \mathbf{0}$, we find a minimizer of (3) on the direction that all the vectors $(z_i)_g$ are parallel to the vectors $(x_i)_g$. In addition, when $\frac{\beta}{4}\left((z_i)'_g - (x_i)'_g\right) + \frac{(z_i)'_g}{\|(z_i)'_g\|_2} = 0$, (18) holds, then $\left(\frac{\beta}{4} + \frac{1}{\|(z_i)'_g\|_2}\right)(z_i)'_g = (x_i)'_g$; therefore, $\frac{(z_i)'_g}{\|(z_i)'_g\|_2} = \frac{(x_i)'_g}{\|(x_i)'_g\|_2}$ holds. Moreover, because of the strict convexity of $f_m(\mathbf{z})$, we know that the minimizer is unique.

On the other hand, when $(z_i)_g = \mathbf{0}$ or $(x_i)_g = \mathbf{0}$, our method may not obtain good results. When $(x_i)_g = \mathbf{0}$, we know that the minimizer of the subproblem $\min\limits_{(z_i)_g} \|(z_i)_g\|_2 + \frac{\beta}{2s}\|(z_i)_g - (x_i)_g\|_2^2$ is exactly that $(z_i)_g = \mathbf{0}$. When $(x_i)_g \neq \mathbf{0}$ and the minimizer of the subproblem $\min\limits_{(z_i)_g} \|(z_i)_g\|_2 + \frac{\beta}{2s}\|(z_i)_g - (x_i)_g\|_2^2$ is that $(z_i)_g = \mathbf{0}$ (since the parameter β/s is two small), our method is not able to obtain good results. For example, that $(z_i)_g = \mathbf{0}$ while $(z_{i+1})_g \neq \mathbf{0}$ makes the element z_i in $\mathbf{z}$ take different values in different subproblems. However, we can obtain an approximate minimizer in this case, which is that the element z_i is a simple summation of corresponding subproblems containing z_i. We will show that in the experiments of Section 5, the approximate minimizer is also good. Moreover, when we take this problem as a subproblem of the image processing problem, we can set the parameter β to be large enough to avoid this drawback.

In addition, from (23), we can know that the element z_i of the minimizer can be treated as in s subproblems independently and then we combine them. In conclusion, after some manipulations, we can get the following two general formulas.

(I)

$$\arg\min_{\mathbf{z}} \quad \|\mathbf{z}\|_{2,1} + \frac{\beta}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 = \mathrm{Sh}_{\mathrm{OGS}}(\mathbf{x}, \beta), \tag{24}$$

with

$$\mathrm{Sh}_{\mathrm{OGS}}(\mathbf{x}, \beta)_i = z_i = \max\left\{1 - \frac{1}{\beta}F(x_i), 0\right\} x_i. \tag{25}$$

Here, for instance, when group size $s = 4$, $(z_i)_g = (z_{i-1}, z_i, z_{i+1}, z_{i+2})$ in $\|\mathbf{z}\|_{2,1}$ and $(x_i)_g$ is defined similarly as $(z_i)_g$, we have $F(x_i) = \left(\frac{1}{\|(x_{i-2})_g\|_2} + \frac{1}{\|(x_{i-1})_g\|_2} + \frac{1}{\|(x_i)_g\|_2} + \frac{1}{\|(x_{i+1})_g\|_2}\right)$. The $\|(x_j)_g\|_2$ is contained in $F(x_i)$ if and only if $\|(x_j)_g\|_2$ has the component x_i, and we follow the convention $(1/0) = 1$ in $F(x_i)$ because $\|(x_i)_g\|_2 = 0$ implies $x_i = 0$ and the value of $F(x_i)$ is insignificant in (25).

(II)

$$\arg\min_{\mathbf{z}} \quad \|\mathbf{z}\|_{2,1} + \frac{\beta}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 = \mathrm{Sh}_{\mathrm{OGS}}(\mathbf{x}, \beta), \tag{26}$$

with

$$\mathrm{Sh}_{\mathrm{OGS}}(\mathbf{x}, \beta)_i = z_i = G(x_i) \cdot x_i. \tag{27}$$

Here, symbols are the same as (I), and $G(x_i) = \max\left(\frac{1}{s} - \frac{1}{\beta\|(x_{i-2})_g\|_2}, 0\right) + \max\left(\frac{1}{s} - \frac{1}{\beta\|(x_{i-1})_g\|_2}, 0\right) + \max\left(\frac{1}{s} - \frac{1}{\beta\|(x_i)_g\|_2}, 0\right) + \max\left(\frac{1}{s} - \frac{1}{\beta\|(x_{i+1})_g\|_2}, 0\right)$.

We call them Formula (I) and Formula (II) respectively throughout this paper. When β is sufficiently large or sufficiently small, the former two formulas are the same. For the other values of β, from the experiments, we find that Formula (II) is better approximate to the MM iteration method than Formula (I), so we list the algorithm for Formula (II) as follows for finding the minimizer of (3). If we would use Formula (I), we can only change the final three steps, which does not change the computation cost.

We can see that Algorithm 1 only needs 2 times of convolution computations with time complexity $n*s$, which is just the same time complexity as one-step iteration in the MM method of [10]. Therefore, our method is much more efficient than the MM method or other variable duplication methods. In Section 5, we will give the numerical experiments for comparison between our method and the MM method. Moreover, if $s = 1$, our Algorithm 1 degenerates to the classic soft thresholding as our formula (27) degenerates to (6). Moreover, when β is sufficiently large or sufficiently small, the results of our new explicit algorithm are almost the same as the MM method (after 20 or more iteration steps) in [10].

Algorithm 1 Direct shrinkage algorithm for the minimization problem (3)

```
Input:
    Given vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, group size $s$,
    parameter $\beta$.
Compute:
    1. Definition of $(x_i)_g$, for example $(x_i)_g = \left(x_{i-s_l}, \cdots, x_i, \cdots, x_{i+s_r}\right)$, with $s_l = [\frac{s-1}{2}]$ and $s_r = [\frac{s}{2}]$ ($s_l + s_r + 1 = s$).
       $\mathbf{w}$ = ones(1,s) = [1,1,$\cdots$,1].
       If $s$ is even, then $\mathbf{w} = [\mathbf{w}, 0]$ and $s = s + 1$.
       Let $\mathbf{w}_r$ = fliplr($\mathbf{w}$) be fliping $\mathbf{w}$ over 180 degrees.
    2. Compute $X_n = \left(\|(x_1)_g\|_2, \cdots, \|(x_i)_g\|_2, \cdots, \|(x_n)_g\|_2\right)$,
       by convolution of $\mathbf{w}$ and $\mathbf{x}$.
    3. Compute $X'_n = \max\left(\frac{1}{s} - 1./(X_n \cdot \beta), 0\right)$ point-wise.
    4. Compute $G(x_i)$ by correlation of $\mathbf{w}$ and $X'_n$,
       or by convolution of $\mathbf{w}_r$ and $X'_n$.
    5. Compute $\mathbf{z}$ by multiplying $\mathbf{x}$ with $G(x_i)$ point-wise.
```

Algorithm 2 Direct shrinkage algorithm for the minimization problem (4)

```
Input:
    Given matrix $A$, group size $s = K_1 \times K_2$, parameter $\beta$,
    weight vector $W_g$.
Compute:
    1. Pad $W_g$ to be an odd-by-odd matrix.
       If $K_1$ is even, then $W_g$ =[ zeros(1, $K_2$); $W_g$] and
       $K_1 = K_1 + 1$.
       If $K_2$ is even, then $W_g$ =[ zeros($K_1$, 1), $W_g$] and
       $K_2 = K_2 + 1$.
       Let $W_g^r$ = fliplr($W_g$) be fliping $W_g$ over 180 degrees.
    2. Compute
       $A_{temp} = \left(\|(A_{1,1})_g\|_2, \cdots, \|(A_{1,K_2})_g\|_2; \cdots; \|(A_{K_1,1})_g\|_2, \cdots, \|(A_{K_1,K_2})_g\|_2\right)$, by correlation of $W_g.^2$ (pointwise square) and $A$, or by convolution of $W_g^r.^2$ and $A$.
    3. Compute $A'_{temp} = \max\left(\frac{1}{\|W_g\|_F^2} - 1./(A_{temp} \cdot \beta), 0\right)$
       pointwise.
    4. Compute $G(A_{i,j})$ by convolution of $W_g.^2$ and $A'_{temp}$,
       or by correlation of $W_g^r.^2$ and $A'_{temp}$.
    5. Compute $X$ by multiplying $A$ with $G(A_{i,j})$ point-wise.
```

Remarks 2. Our new explicit algorithm can be treated as an average estimation algorithm for solving all the overlapping group subproblem independently. In Section 5, our numerical experiments show that the results of our two formulas are almost the same as the MM method in [10] when β is sufficiently large or sufficiently small and is approximate to the MM method for other β. Moreover, this point illuminates that the OGS regularizers coincide to maintain the property of the classical sparse regularizers and smoothen the local regions. For instance, TV with OGS can preserve edges and simultaneously overcome staircases in our experiments and [11, 12].

For the problems (4) (See Fig. 1b intuitively) and (5), we can get similar algorithms. We omit the evolution details here and give them as in Appendix. Here, we directly give the algorithm for solving the problem (5) as follows under Formula (II).

We can also see that Algorithm 2 only needs 2 times of convolution computations with time complexity $n * s$, which is just the same time complexity as one-step iteration in the MM method of [10]. Therefore, our method is much more efficient than the MM method.

3 Several extensions

3.1 Other boundary conditions

In Section 2, we gave the explicit shrinkage formulas for one class of OGS problems (3), (4), and (5). In order to achieve a simply deduction, we assume that PBC is used. One may confuse that whether PBC is always good for regularization problems such as signal processing or image processing, since natural signals or images are often asymmetric. However, in these problems, assuming a kind of boundary condition is necessary for simplifying the problem and making the computation possible [24]. There are kinds of BCs, such as zero BC (ZBC) and reflective BC. PBC is often used in optimization because it can be computed fast as above or other methods, for example, computation of matrix of block circulant with circulant blocks (BCCB) by fast Fourier transforms [20, 21, 24].

In this section, we consider other BC such as ZBC and reflective BC. For simplification, we only consider the vector case, while the results can be easily expanded to the matrix case as that in Section 2. When ZBC is used, we can expand the original vectors (or signal) $\mathbf{z}$ $(= (z_i)_{i=1}^n)$ and $\mathbf{x}$ by two s-length vectors on the both hands of the original vectors, which are $\tilde{\mathbf{z}}$ $(= \left[\tilde{z}_{-s}, \cdots, \tilde{z}_{-1}, (\tilde{z}_i)_{i=1}^n, \tilde{z}_{n+1}, \cdots, \tilde{z}_{n+s}\right] = \left[\mathbf{0}_s, (z_i)_{i=1}^n, \mathbf{0}_s\right])$ and $\tilde{\mathbf{x}}$, respectively. Then, the results and algorithms in Section 2 are similar as the case of PBC on $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{x}}$. See Fig. 2a intuitively.

Moreover, our numerical results will show that the results from a different BC are almost the same in practice (see Section 5.1); and according to the definition of weighted generalized $\ell_{2,1}$ norm, ZBC seems better than PBC. Therefore, we will choose the ZBC to solve the problems (3), (4), and (5) in the following sections.

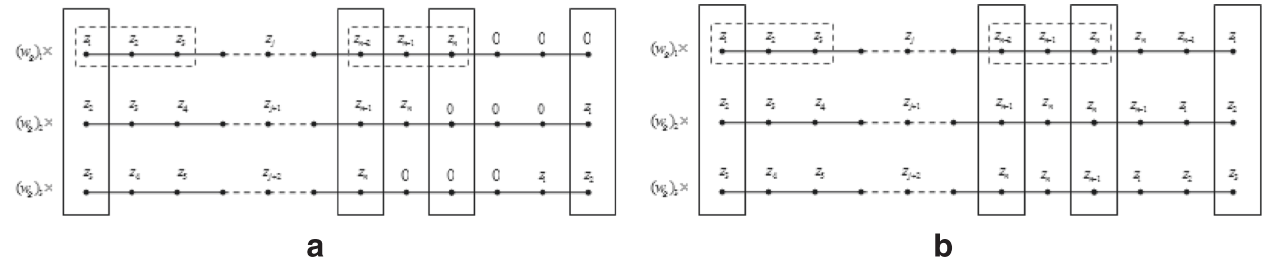

Fig. 2 The case of expanding vectors with other BCs to use the formulas under PBC. **a** ZBC. **b** Reflective BC

When reflective BC is assumed, the results are also the same. We only need to extend the original vector $\mathbf{x}$ to $\hat{\mathbf{x}}$ with reflective BC. See Fig. 2b intuitively.

3.2 Nonpositive weights and different weights in groups

In this section, we show that the weight vector w_g and matrix W_g in the former sections can contain arbitrary s entries with arbitrary real numbers. On one hand, the zero value can be the arbitrary entries of the weight vector or matrix. For example, the original ℓ_1 regularization problem (1) can be seen as a special form of weighted generalized $\ell_{2,1}$ norm regularization problems (4) with $s = 1$ or $s = 3$ and $w_g = (1, 0, 0)$ for the example of Fig. 1b.

On the other hand, the norm is with the property of positive homogeneity, $\|(-1) \cdot w_g\|_{any} = \|w_g\|_{any}$, where "$\| \cdot \|_{any}$" can be arbitrary norm such as $\| \cdot \|_1$ and $\| \cdot \|_2$. Therefore, for the regularization problems (4) (or (5)), the weight vector w_g (or matrix W_g) is the same as $|w_g|$ (or $|W_g|$), where $| \cdot |$ the point-wise absolute value. Therefore, in general, our results are true, whatever the number of entries included in the weight vector w_g and matrix W_g and whatever the real number value of these entries.

However, if the weight w_g is dependent on the group index i, for example, $w_{i,g} \circ (z_i)_g = \big((w_{i,g})_1 z_i, (w_{i,g})_2 z_{i+1}, (w_{i,g})_3 z_{i+2}\big)$, we cannot solve the relevant problems easily since our method fails. As we mentioned above, we only focus on that the weight w_g is independent on the group index i, that is, $w_{i,g} = w_g$ for all i, which means that it is with translation invariant overlapping groups.

4 Applications in TV regularization problems with OGS

The TV regularizer was firstly introduced by Rudin et al. [25] (ROF), which is widely used in many fields, i.e., denoising and deblurring problems [20, 21, 26–31]. Several fast algorithms such as Chambelle [26] and fast TV deconvolution (FTVd) [20, 21] have been proposed. Its corresponding minimization task is

$$\min_f \|f\|_{\mathrm{TV}} + \frac{\mu}{2}\|Hf - g\|_2^2, \tag{28}$$

where $\|f\|_{\mathrm{TV}} := \sum_{1\le i,j\le n} \|(\nabla f)_{i,j}\|_2 = \sum_{1\le i,j\le n} \sqrt{|(\nabla_x f)_{i,j}|^2 + |(\nabla_y f)_{i,j}|^2}$ (called isotropic TV, ITV), or $\|f\|_{TV} := \sum_{1\le i,j\le n} \|(\nabla f)_{i,j}\|_1 = \sum_{1\le i,j\le n} |(\nabla_x f)_{i,j}| + |(\nabla_y f)_{i,j}|$ (called anisotropic TV, ATV), H denotes the blur matrix, and g denotes the given observed image with blur and noise. Operator $\nabla : \mathbb{R}^{n^2} \to \mathbb{R}^{2\times n^2}$ denotes the discrete gradient operator (under PBC) which is defined by $(\nabla f)_{i,j} = ((\nabla_x f)_{i,j}, (\nabla_y f)_{i,j})$, with

$$(\nabla_x f)_{i,j} = \begin{cases} f_{i+1,j} - f_{i,j} & \text{if } i < n, \\ f_{1,j} - f_{n,j} & \text{if } i = n, \end{cases} \quad (\nabla_y f)_{i,j} = \begin{cases} f_{i,j+1} - f_{i,j} & \text{if } j < n, \\ f_{i,1} - f_{i,n} & \text{if } j = n, \end{cases}$$

for $i, j = 1, 2, \cdots, n$, where $f_{i,j}$ refers to the $((j-1)n + i)$th entry of the vector f (it is the (i,j)th pixel location of the $n \times n$ image, and this notation remains valid throughout the paper unless otherwise specified). Notice that H is a matrix of BCCB structure when PBC are applied [24].

Recently, Selesnick et al. [13] proposed an OGS TV regularizer to one-dimensional signal denoising. They applied the MM method to solve their model. Their numerical experiments showed that their method can overcome staircase effects effectively and get better results. However, their method has the disadvantages of the low speed of computation and the difficulty to be directly extended to the two-dimensional image case. More recently, Liu et al. [11] proposed an OGS TV regularizer for two-dimensional image denoising and deblurring under Gaussian noise, and Liu et al. [12] proposed an OGS TV regularizer for image deblurring under impulse noise. Both of them used a variable substitution method and the ADMM framelet with an inner MM iteration for solving the subproblems similar as (3). Therefore, their methods may spend more time than our methods. Moreover, when the MM method is used in the inner iterations in [11, 12], they can only solve the ATV case but not the ITV case with OGS while our methods can solve both ATV and ITV cases.

Firstly, we define the ATV case with OGS under Gaussian noise and impulse noise respectively which is similar as [11, 12]. For the Gaussian noise case,

$$\min_f \|(\nabla_x f)\|_{W,2,1} + \|(\nabla_y f)\|_{W,2,1} + \frac{\mu}{2}\|Hf - g\|_2^2. \quad (29)$$

For the impulse case,

$$\min_f \|(\nabla_x f)\|_{W,2,1} + \|(\nabla_y f)\|_{W,2,1} + \mu\|Hf - g\|_1. \quad (30)$$

We call the former model as TV OGS L_2 model and the latter model as TV OGS L_1 model, respectively.

Then, we defined the ITV case with OGS. For Gaussian noise and impulse noise, we only change the former two terms of (29) and (30) respectively by $\|A\|_{W,2,1}$. Here, A is a high-dimensional matrix (or tensor) with each entry $A_{i,j} = ((\nabla_x f)_{i,j}; (\nabla_y f)_{i,j})$.

Remarks 3. In particular in this work, A can be treated as $((\nabla_x f); (\nabla_y f))$ for simplicity in vector and matrix computation, although the computation of $\|A\|_{W,2,1}$ is should be treated as pointwise with $A_{i,j} = ((\nabla_x f)_{i,j}; (\nabla_y f)_{i,j})$ since the computation on A is almost all pointwise.

Moreover, we consider constrained model as listing in [11, 12, 32]. For any true digital image, its pixel value can attain only a finite number of values. Hence, it is natural to require all pixel values of the restored image to lie in a certain interval $[a, b]$, see [32] for more details. In general, with the easy computation and the certified results in [32], we only consider all the images located on the standard range $[0, 1]$. Therefore,we define a projection operator $\mathcal{P}_\Omega$ on the set $\Omega = \{f \in \mathbb{R}^{n\times n} | 0 \le f \le 1\}$,

$$\mathcal{P}_\Omega(f)_{i,j} = \begin{cases} 0, & f_{i,j} < 0, \\ f_{i,j}, & f_{i,j} \in [0, 1], \\ 1, & f_{i,j} > 1. \end{cases} \quad (31)$$

4.1 Constrained TV OGS L_2 model

For the constrained model (the ATV case) (called CATVOGSL2), we have

$$\min_{u\in\Omega, v_x, v_y, f} \left\{ \|(v_x)\|_{W,2,1} + \|(v_y)\|_{W,2,1} + \frac{\mu}{2}\|Hf - g\|_2^2 : \right.$$
$$\left. v_x = \nabla_x f, v_y = \nabla_y f, u = f \right\}. \quad (32)$$

The augmented Lagrangian function [33] of (32) is

$$\begin{aligned}\mathcal{L}(v_x, v_y, u, f; \lambda_1, \lambda_2, \lambda_3) = {} & \|v_x\|_{W,2,1} - \lambda_1^T(v_x - \nabla_x f) + \frac{\beta_1}{2}\|v_x - \nabla_x f\|_2^2 \\ & + \|v_y\|_{W,2,1} - \lambda_2^T(v_y - \nabla_y f) + \frac{\beta_1}{2}\|v_y - \nabla_y f\|_2^2 \\ & - \lambda_3^T(u - f) + \frac{\beta_2}{2}\|u - f\|_2^2 + \frac{\mu}{2}\|Hf - g\|_2^2,\end{aligned} \quad (33)$$

where $\beta_1, \beta_2 > 0$ are penalty parameters and $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}^{n^2}$ are the Lagrange multipliers.

The solving algorithm is according to the scheme of ADMM in Gabay [34], and we refer to some applications in image processing which can be solved by ADMM, e.g., [35–43]. For a given $\left(v_x^k, v_y^k, u^k, f^k; \lambda_1^k, \lambda_2^k, \lambda_3^k\right)$, the next iteration $(v_x^{k+1}, v_y^{k+1}, u^{k+1}, f^{k+1}; \lambda_1^{k+1}, \lambda_2^{k+1}, \lambda_3^{k+1})$ is generated as follows:

1. Fix $f = f^k$, $\lambda_1 = \lambda_1^k, \lambda_2 = \lambda_2^k, \lambda_3 = \lambda_3^k$ and minimize (33) with respect to v_x, v_y, and u. With respect to v_x and v_y,

$$\begin{aligned} v_x^{k+1} &= \arg\min \|v_x\|_{W,2,1} - {\lambda_1^k}^T(v_x - \nabla_x f^k) + \tfrac{\beta_1}{2}\|v_x - \nabla_x f^k\|_2^2 \\ &= \arg\min \|v_x\|_{W,2,1} + \tfrac{\beta_1}{2}\|v_x - \nabla_x f^k - \tfrac{\lambda_1^k}{\beta_1}\|_2^2, \end{aligned} \quad (34)$$

$$\begin{aligned} v_y^{k+1} &= \arg\min \|v_y\|_{W,2,1} - {\lambda_2^k}^T(v_y - \nabla_y f^k) + \tfrac{\beta_1}{2}\|v_y - \nabla_y f^k\|_2^2 \\ &= \arg\min \|v_y\|_{W,2,1} + \tfrac{\beta_1}{2}\|v_y - \nabla_y f^k - \tfrac{\lambda_2^k}{\beta_1}\|_2^2. \end{aligned} \quad (35)$$

It is obvious that problems (34) and (35) match the framework of the problem (5); thus, the minimizers of (34) and (35) can be obtained by using the formulas in Section 2.2.

With respect to u,

$$\begin{aligned} u^{k+1} &= \arg\min -{\lambda_3^k}^T(u - f^k) + \tfrac{\beta_2}{2}\|u - f^k\|_2^2 \\ &= \arg\min \tfrac{\beta_2}{2}\|u - f^k - \tfrac{\lambda_3^k}{\beta_2}\|_2^2. \end{aligned}$$

The minimizer is given explicitly by

$$u^{k+1} = \mathcal{P}_\Omega\left[f^k + \frac{\lambda_3^k}{\beta_2}\right]. \quad (36)$$

2. Compute f^{k+1} by solving the normal equation

$$\begin{aligned} & (\beta_1(\nabla_x^*\nabla_x + \nabla_y^*\nabla_y) + \mu H^*H + \beta_2 I)f^{k+1} \\ & = \nabla_x^*(\beta_1 v_x^{k+1} - \lambda_1^k) + \nabla_y{}^*(\beta_1 v_y^{k+1} - \lambda_2^k) + \mu H^*g \\ & \quad + \beta_2(u^{k+1} - \tfrac{\lambda_3^k}{\beta_2}), \end{aligned} \quad (37)$$

where "$*$" denotes the conjugate transpose, see [44] for more details. Since all the parameters are positive, the coefficient matrix in (37) are always invertible and symmetric positive-definite. In addition, H, ∇_x, ∇_y and their conjugate transpose have BCCB structure under PBC. We know that the computations with BCCB matrix can be very efficient by using fast Fourier transforms.

3. Update the multipliers via

$$\begin{cases} \lambda_1^{k+1} = \lambda_1^k - \gamma\beta_1(v_x^{k+1} - \nabla_x f^{k+1}), \\ \lambda_2^{k+1} = \lambda_2^k - \gamma\beta_1(v_x^{k+1} - \nabla_x f^{k+1}), \\ \lambda_3^{k+1} = \lambda_3^k - \gamma\beta_2(u^{k+1} - f^{k+1}). \end{cases} \quad (38)$$

Based on the discussions above, we present the ADMM algorithm for solving the convex CATVOGSL2 model (32), which is shown as Algorithm 3.

Algorithm 3 CATVOGSL2 for the minimization problem (32)

```
initialization:
  Starting point $f^0 = g$, $k = 0$, $\beta_1$, $\beta_2$, $\gamma$, $\mu$, group size
  $K_1 \times K_2$, weighted matrix $W_g$, $\lambda_i^0 = 0, i = 1, 2, 3$.
iteration:
  1. Compute $v_x^{k+1}$ and $v_y^{k+1}$ according to (34) and (35).
  2. Compute $u^{k+1}$ according to (36).
  3. Compute $f^{k+1}$ by solving (37).
  4. update $\lambda_i^0 = 0, i = 1, 2, 3$ according to (38).
  5. $k = k + 1$.
until a stopping criterion is satisfied.
```

Algorithm 3 is an application of ADMM for the case with two blocks of variables (v_x, v_y, u) and f. Thus, if the step (1) of Algorithm 3 can be solved exactly, its convergence is guaranteed by the theory of ADMM [34, 35, 37, 43]. Although step (1) of Algorithm 3 cannot be solved exactly, we can find a convergent series to ensure the convergence as [35]. Besides, our numerical experiments verify the convergence of Algorithm 3.

Remarks 4. For the image f, we use PBC for the fast computation. However, for v_x, v_y, we use ZBC. Because v_x and v_y denote the gradient of the image, ZBC seems better for the definition of the generalized norm $\ell_{2,1}$ on v_x and v_y as mentioned in Section 3.1. Therefore, the two kinds of BCs for the image and its gradient are different and independent. These remain valid throughout the paper unless otherwise specified.

For the ITV case, the constrained model (CITVOGSL2) is

$$\min_{u\in\Omega,A,f} \left\{\|A\|_{W,2,1}+\frac{\mu}{2}\|Hf-g\|_2^2 : \ A=(\nabla_x f;\nabla_y f), u=f\right\}. \tag{39}$$

The detail will be presented in the next section for the constrained TV OGS L_1 model. We call this relevant algorithm CITVOGSL2.

4.2 Constrained TV OGS L_1 model

For the constrained model (the ITV case) (called CITVOGSL1), we have

$$\min_{u\in\Omega,A,f} \left\{\|A\|_{W,2,1}+\mu\|z\|_1 : z=Hf-g, A=(\nabla_x f;\nabla_y f), u=f\right\}. \tag{40}$$

The augmented Lagrangian function of (40) is

$$\begin{aligned}\mathcal{L}\left(v_x,v_y,z,w,f;\lambda_1,\lambda_2,\lambda_3,\lambda_4\right) = \|A\|_{W,2,1} - \left(\lambda_1^T,\lambda_2^T\right)\left(A-\begin{bmatrix}\nabla_x f\\ \nabla_y f\end{bmatrix}\right)\\ +\frac{\beta_1}{2}\left\|A-\begin{bmatrix}\nabla_x f\\ \nabla_y f\end{bmatrix}\right\|_2^2\\ +\mu\|z\|_1-\lambda_3^T\left(z-(Hf-g)\right)\\ +\frac{\beta_2}{2}\|z-(Hf-g)\|_2^2\\ -\lambda_4^T(w-f)+\frac{\beta_3}{2}\|w-f\|_2^2,\end{aligned} \tag{41}$$

where $\beta_1, \beta_2, \beta_3 > 0$ are penalty parameters and $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}^{n^2}$ are the Lagrange multipliers.

For a given $(A^k, z^k, u^k, f^k;\ \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k)$, the next iteration $(A^{k+1}, z^{k+1}, u^{k+1}, f^{k+1};\ \lambda_1^{k+1}, \lambda_2^{k+1}, \lambda_3^{k+1}, \lambda_4^{k+1})$ is generated as follows:

1. Fix $f = f^k$, $\lambda_1 = \lambda_1^k, \lambda_2 = \lambda_2^k, \lambda_3 = \lambda_3^k, \lambda_4 = \lambda_4^k$ and minimize (41) with respect to A, z, and u. With respect to A,

$$\begin{aligned}A^{k+1} &= \arg\min \|A\|_{W,2,1} - (\lambda_1^T,\lambda_2^T)(A-\begin{bmatrix}\nabla_x f^k\\ \nabla_y f^k\end{bmatrix}) + \tfrac{\beta_1}{2}\left\|A-\begin{bmatrix}\nabla_x f^k\\ \nabla_y f^k\end{bmatrix}\right\|_2^2,\\ \begin{bmatrix}A_1^{k+1}\\ A_2^{k+1}\end{bmatrix} &= \arg\min \|A\|_{W,2,1} + \tfrac{\beta_1}{2}\left\|\begin{bmatrix}A_1\\ A_2\end{bmatrix}-\begin{bmatrix}\nabla_x f^k\\ \nabla_y f^k\end{bmatrix}-\begin{bmatrix}\lambda_1^k/\beta_1\\ \lambda_2^k/\beta_1\end{bmatrix}\right\|_2^2.\end{aligned} \tag{42}$$

It is obvious that problem (42) match the framework of the problem (5), thus the solution of (42) can be obtained by using the formulas in Section 2.2.

With respect to z,

$$\begin{aligned}z^{k+1} &= \arg\min \mu\|z\|_1 - {\lambda_3^k}^T\left(z-(Hf^k-g)\right) + \tfrac{\beta_2}{2}\|z-(Hf^k-g)\|_2^2\\ &= \arg\min \mu\|z\|_1 + \tfrac{\beta_2}{2}\|z-(Hf^k-g)-\tfrac{\lambda_3^k}{\beta_2}\|_2^2.\end{aligned}$$

The minimization with respect to z can be given by (6) and (9) explicitly, that is,

$$z^{k+1}=\operatorname{sgn}\left\{Hf^k-g+\frac{\lambda_3^k}{\beta_2}\right\}\circ\max\left\{|Hf^k-g+\frac{\lambda_3^k}{\beta_2}|-\frac{\mu}{\beta_2},0\right\}. \tag{43}$$

With respect to u,

$$\begin{aligned}u^{k+1} &= \arg\min -{\lambda_4^k}^T\left(u-f^k\right)+\tfrac{\beta_3}{2}\|u-f^k\|_2^2\\ &= \arg\min \tfrac{\beta_3}{2}\|u-f^k-\tfrac{\lambda_4^k}{\beta_3}\|_2^2.\end{aligned}$$

The minimizer is given explicitly by

$$u^{k+1} = \mathcal{P}_\Omega\left[f^k+\frac{\lambda_4^k}{\beta_3}\right]. \tag{44}$$

2. Compute f^{k+1} by solving the following normal equation similarly as the last section.

$$\begin{aligned}&(\beta_1(\nabla_x^*\nabla_x + \nabla_y^*\nabla_y) + \beta_2 H^*H + \beta_3 I)f^{k+1}\\ &= \nabla_x^*(\beta_1 A_1^{k+1} - \lambda_1^k) + \nabla_y{}^*(\beta_1 A_2^{k+1} - \lambda_2^k) +\\ &H^*(\beta_2 z^{k+1} - \lambda_3^k) + \beta_2 H^* g + \beta_3(u^{k+1} - \tfrac{\lambda_4^k}{\beta_3}).\end{aligned} \tag{45}$$

3. Update the multipliers via

$$\begin{cases}\begin{bmatrix}\lambda_1^{k+1}\\ \lambda_2^{k+1}\end{bmatrix} = \begin{bmatrix}\lambda_1^k\\ \lambda_2^k\end{bmatrix} - \gamma\beta_1\left(\begin{bmatrix}A_1^{k+1}\\ A_2^{k+1}\end{bmatrix} - \begin{bmatrix}\nabla_x f^{k+1}\\ \nabla_y f^{k+1}\end{bmatrix}\right),\\ \lambda_3^{k+1} \quad = \lambda_3^k - \gamma\beta_2(z^{k+1} - (Hf^{k+1} - g)),\\ \lambda_4^{k+1} \quad = \lambda_4^k - \gamma\beta_3(u^{k+1} - f^{k+1}).\end{cases} \tag{46}$$

Based on the discussions above, we present the ADMM algorithm for solving the convex CITVOGSL1 model (40), which is shown as Algorithm 4.

Algorithm 4 CITVOGSL1 for the minimization problem (40)

```
***initialization***:
  Starting point $f^0 = g$, $k = 0$, $\beta_1, \beta_2, \beta_3, \gamma, \mu$, group size
  $K_1 \times K_2$, weighted matrix $W_g$, $\lambda_i^0 = 0, i = 1,2,3,4$.
***iteration***:
  1. Compute $A^{k+1}$ according to (42),
     and compute $z^{k+1}$ according to (43).
  2. Compute $u^{k+1}$ according to (44).
  3. Compute $f^{k+1}$ by solving (45).
  4. update $\lambda_i^0 = 0, i = 1,2,3,4$ according to (46).
  5. $k = k + 1$.
***until a stopping criterion is satisfied.***
```

Algorithm 4 is an application of ADMM for the case with two blocks of variables (A, z, u) and f. Thus, if step (1) of Algorithm 4 can be solved exactly, its convergence is guaranteed by the theory of ADMM [34, 35, 37, 43]. Although step (1) of Algorithm 4 cannot be solved exactly, we can find a convergent series to ensure the convergence as [35]. Besides, our numerical experiments verify the convergence of Algorithm 4.

For the ATV case, the constrained model (CATVOGSL1) is

$$\min_{u\in\Omega, v_x, v_y, f}\left\{\begin{array}{l}\|(v_x)\|_{W,2,1} + \|(v_y)\|_{W,2,1} + \mu\|z\|_1 :\\ z = Hf - g, v_x = \nabla_x f, v_y = \nabla_y f, u = f\end{array}\right\}. \tag{47}$$

The detail has been presented in the last section for the constrained TV OGS L_2 model. We call this relevant algorithm CATVOGSL1.

5 Numerical results

In this section, we present several numerical results to illustrate the performance of the proposed method. All experiments are carried out on a desktop computer using MATLAB 2010a. Our computer is equipped with an Intel Core i3-2130 CPU (3.4 GHz), 3.4 GB of RAM, and a 32-bit Windows 7 operation system.

5.1 Comparison with the MM method for one-dimensional signal denoising

As an illustrative example, we apply our direct formulas in one-dimensional signal denoising. We only compare the results of our explicit shrinkage formulas with the most recent MM iteration method proposed in [10] as a simple example. The one-dimensional group sparse signal $\mathbf{z}$ is shown in the top left of Fig. 3. The noisy signal $\mathbf{x}$ in the top right of Fig. 3 is obtained by adding independent white Gaussian noise with standard deviation $\sigma = 0.5$ same as [10]. The denoising model is as follows

$$\arg\min_{\mathbf{z}} \quad \mathrm{Fun}(\mathbf{z}) = \|\mathbf{z}\|_{2,1} + \frac{\beta}{2}\|\mathbf{z} - \mathbf{x}\|_2^2. \tag{48}$$

We test several different parameters β (any integer from 1 to 50) for the comparison. The results of our formulas are almost the same as the MM method for almost these βs. Due to the limit space, we only illustrate parts of the results ($\beta = 3, 10, 15$) in Fig. 3. We take both our Formulas (I) and (II) for the comparison with the MM method. From the figure, we can visually see that our results by our two kinds of formulas are almost the same as the MM method for different β. Moreover, for the practical problems, $\beta = 10$, can be better than others from the figure. This shows that our formulas are feasible, useful, and effective, because our method can only need the same computation cost as one-step iteration in the MM method while the MM method needs 25 iterations as [10]. That means our method is 25 times faster than the MM method. From the bottom line of Fig. 3, we can see that the MM method may take 5 iterations to be convergent, which is the reason why Liu et al. in [11] and Liu et al. in [12] only choose 5 inner iterations. However, in this case, our method is still 5 times faster than the MM method.

Remarks 5. Although this model is not the best model for signal denoising, it shows the superiority of our method. Moreover, the authors in [10] also take similar experiments.

5.2 Comparison with the MM method on problem (5)

In this section, we give another example by solving the matrix case problem (5) for comparison with the MM method. Without loss of generality, we let $A = \mathrm{rand}(100, 100)$ and $A(45 : 55, 45 : 55) = 0$ in the

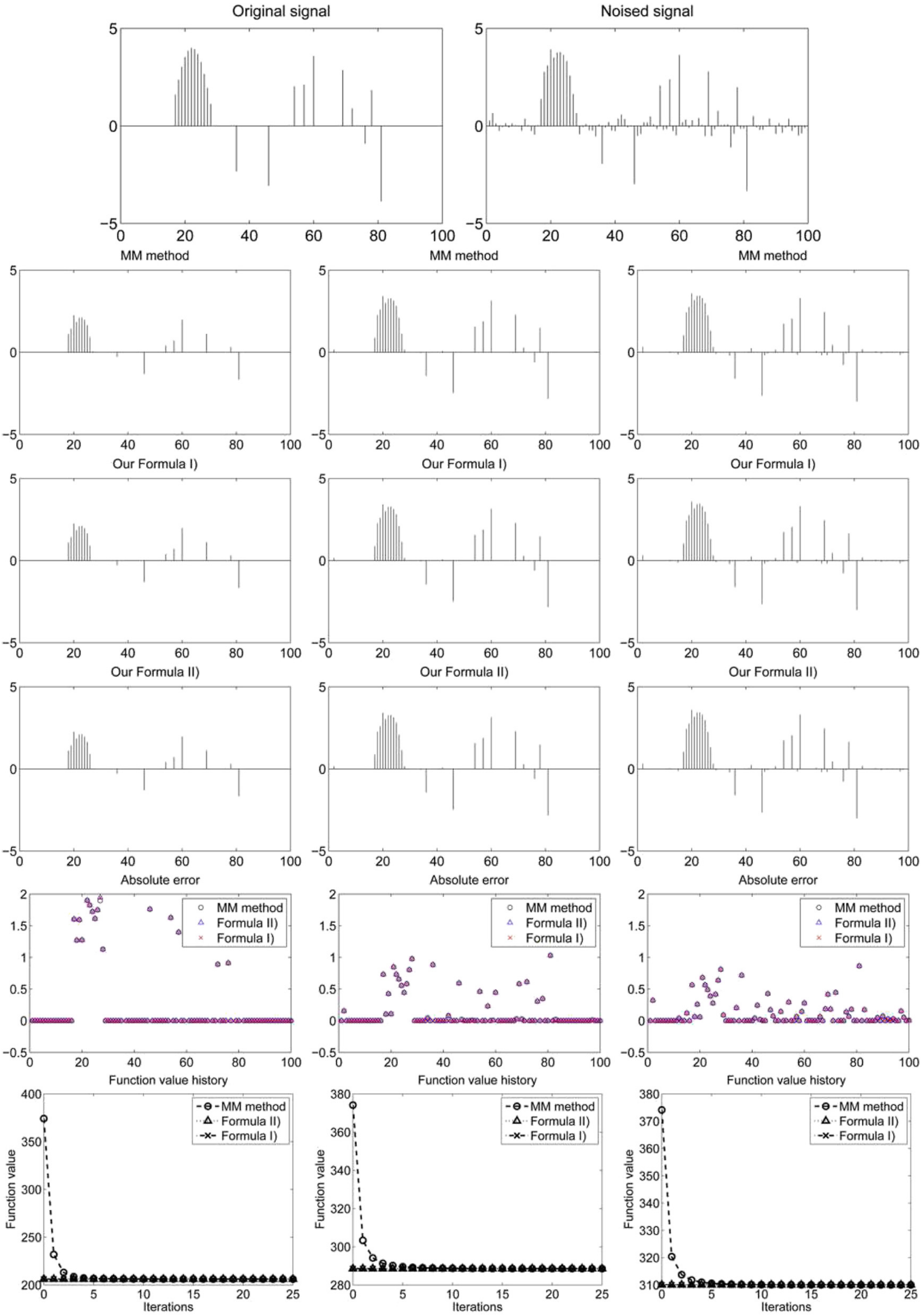

Fig. 3 Comparison between our method and the MM method for one-dimensional signal denoising. *Second row* to *bottom row*, from *top* to *bottom*, results of the MM method, our Formula (I), our Formula (II) respectively, absolute error, and function value history. From *left* to *right*, $\beta = 3, 10, 15$ respectively

example. In particular, we set weighted matrix $W_g \in \mathbb{R}^{3\times3}((W_g)_{i,j} \equiv 1)$ and list (5) again

$$f_A(X) = \min_X \text{Fun}(X) = \|X\|_{2,1} + \frac{\beta}{2}\|X - A\|_F^2. \quad (49)$$

We use different parameters β and use the number of MM iteration by 20 steps. For comparison, we expand our result by explicit shrinkage formulas to length 20 (the same as the MM iteration steps). The values of the object function Fun(X) against iterations are illustrated in the top line in Fig. 4. The cross-sectional elements of the minimizer A are shown in the bottom line in Fig. 4. We choose both ZBC and PBC for our method and the MM method. It can be seen that there is only a little difference between the two kinds of BCs.

It is obvious that our method is much more efficient than the MM method since our time complexity is just the same as one-step iteration in the MM method due to the explicit shrinkage formula. From the top line in Fig. 4, we observe that the MM method is also fast because it only needs less than 20 steps (sometimes 5) to converge. The related error of the function value of two methods is much less than 1 % for different parameters β. From the bottom line in Fig. 4, we can see that when $\beta \leq 1$ which is sufficiently small, and when $\beta \geq 30$ which is sufficiently large, our result is almost the same as the final results by the MM iteration method. This shows that the results by our method are almost the same as the MM method. When $1 < \beta < 30$, the minimizer computed by our method is approximate to the MM method both on the error of the function value and on the minimizer X.

In Table 1, we show the numerical comparison of our method and the MM method on three parts, the related error of function value (ReE$_{f_A}$), related error of minimizer X (ReE$_X$), and the mean absolute error of X (MAE$_X$). The three terms are defined by ReE$_{f_A}$ $= \frac{|f_{A\text{MM}} - f_{A\text{ours}}|}{f_{A\text{ours}}}$, ReE$_X$ $= \frac{\|X_{\text{MM}} - X_{\text{ours}}\|_F}{\|X_{\text{ours}}\|_F}$, and MAE$_X$ $=$ mean absolute error of $(X_{\text{MM}} - X_{\text{ours}})$, where X_{MM} and X_{ours} are the final solution X of the MM method and our method, respectively. $f_{A\text{MM}}$ and $f_{A\text{ours}}$ are the final function value of MM and our method, respectively.

From the table and the figure, we can see that our formula can almost get the same results as the MM method when β is sufficiently large and approximate results similar as the MM method for other β. This is another proof for the feasibility of our formula.

Similarly as weighted matrix $W_g \in \mathbb{R}^{3\times3}((W_g)_{i,j} \equiv 1)$, we also tested other weighted matrix for more than 1000 times. For example, when $A = \text{rand}(100, 100)$ and $A(A >= 0.5) = 0$ (or $A(A <= 0.5) = 1$) (every element is in $[0, 1]$), we find that, generally, $\beta \leq \frac{\|w_g\|_2}{\sqrt{s}}$ is sufficiently small and $\beta \geq 30 \cdot \frac{\|W_g\|_2}{\sqrt{s}}$ is sufficiently large. These results illustrate that the former when $W_g \in \mathbb{R}^{3\times3}((W_g)_{i,j} \equiv 1)$ $\beta \geq 30$ which is sufficiently large again. However, in practice, if β is too large, then $A = X$, and the minimization problem (49) is meaningless. In our experiments after more than 1000 tests, we find that, when every element of A is in $[0, 1]$, $30 \cdot \frac{\|W_g\|_2}{\sqrt{s}} \leq \beta \leq 300 \cdot \frac{\|W_g\|_2}{\sqrt{s}}$ is sufficiently large but not too large to make the minimization problem meaningless. In this case, that means we may tune the parameter to be better in practice. On the other hand, when some elements of A are not in $[0, 1]$, we can first project or stretch A into the region $[0, 1]$ and then choose the better parameter β. Moreover, although the parameter is not chosen to be in this region, the result can also be treated as an approximate solution with small error. That means our formulas are feasible, useful, and effective.

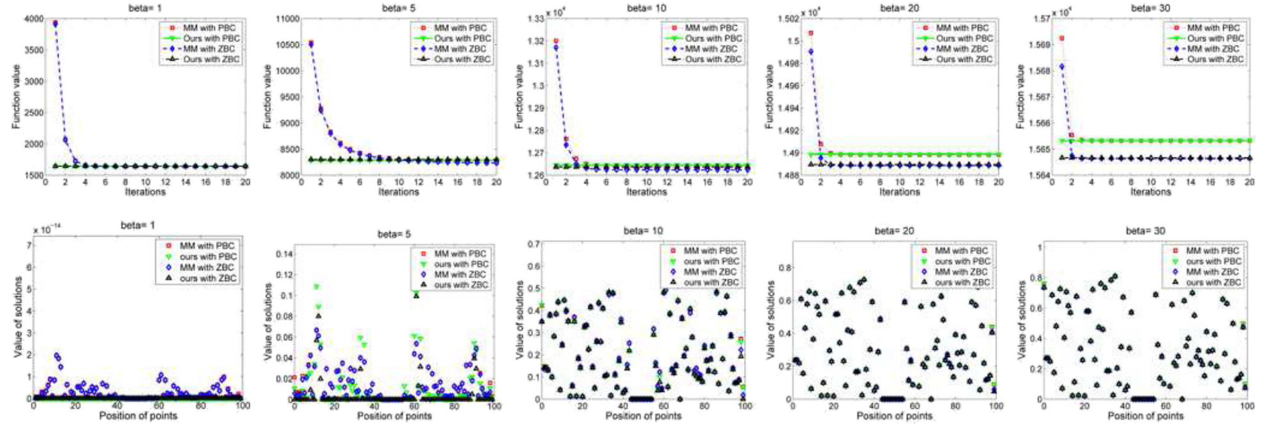

Fig. 4 Comparison between our method and the MM method on two kinds of BCs, ZPC, and PBC. *Top row*, the function value history. *Bottom row*, the cross line of the final minimizer—matrix *X*. From *left* to *right*, $\beta = 1, 5, 10, 20, 30$

Table 1 Numerical comparison of our method and the MM method for two kinds of BCs, ZBC, and PBC

BC	$\beta =$	1	5	7	10	15	20	30	50
ZBC	ReE$_{f_A}$	5.9e-14	0.0055	0.0028	6.5e-4	1.4e-4	5.4e-5	1.4e-5	2.6e-6
	ReE$_X$	—	—	0.0207	0.0017	1.9e-4	4.7e-5	7.5e-6	7.8e-7
	MAE$_X$	3.4e-15	0.0094	0.0130	0.0067	0.0029	0.0016	6.9e-4	2.4e-4
PBC	ReE$_{f_A}$	6.5e-14	0.0053	0.0025	4.1e-4	5.5e-5	1.1e-5	6.3e-7	1.3e-6
	ReE$_X$	—	—	0.0193	0.0014	1.3e-4	2.9e-5	4.3e-6	4.5e-7
	MAE$_X$	3.8e-15	0.0097	0.0130	0.0063	0.0026	0.0014	5.9e-4	2.1e-4

ReE of $f_A(X)$ denotes the related error of final function value fun(X). ReE of X denotes related error of minimizer X. MAE of X denotes the mean absolute error of X

5.3 Comparison with TV methods and TV with OGS with inner iteration MM methods for image deblurring and denoising

In this section, we compare all our algorithms with other methods. All the test images are shown in Fig. 5, one 1024×1024 image as (a) Man and three 512×512 images as (b) Car, (c) Parlor, and (d) Housepole.

The quality of the restoration results is measured quantitatively by using the peak signal-to-noise ratio (PSNR) in decibel (dB) and the relative error (ReE):

$$\text{PSNR} = 10 \log_{10} \frac{n^2 \text{Max}_I^2}{\|f - \bar{f}\|_2^2}, \quad \text{ReE} = \frac{\|f - \bar{f}\|_2}{\|\bar{f}\|_2},$$

where $\bar{f}$ and f denote the original and restored images, respectively, and Max$_I$ represents the maximum possible pixel value of the image. In our experiments, Max$_I = 1$.

The stopping criterion used in our work is set to be as other methods

$$\frac{|\mathcal{F}^{k+1} - \mathcal{F}^k|}{|\mathcal{F}^k|} < 10^{-5}, \tag{50}$$

where $\mathcal{F}^k$ is the objective function value of the respective model in the kth iteration.

We compare our methods with some other methods, such as Chan et al.’s TV method proposed in [32], Liu et al.’s method proposed in [11], and Liu et al.’s method proposed in [12]. Both the latter two methods [11, 12] are with inner iteration MM methods for OGS TV problems, where the number of the inner iterations is set 5 by them. In particular, for a fair comparison, we set weighted matrix $W_g \in \mathbb{R}^{3\times3}((W_g)_{i,j} \equiv 1)$ in all the experiments of our methods as in [11, 12].

5.3.1 Experiments for the constrained TV OGS L_2 model

In this section, we compare our methods (CATVOGSL2 and CITVOGSL2) with some other methods, such as Chan et al.’s method proposed in [32] (Algorithm 1 in [32] for the constrained TV-L2 model) and Liu et al.’s method proposed in [11].

We set the penalty parameters $\beta_1 = 35$, $\beta_2 = 20$, for the ATV case, $\beta_1 = 100$, $\beta_2 = 20$ for the ITV case and relax parameter $\gamma = 1.618$. The blur kernels are generated by MATLAB built-in function (i)`fspecial('average',9)` for 9×9 average blur. We generate all blurring effects using the MATLAB built-in function `imfilter(I,psf, 'circular','conv')` under PBC with "`I`" as the original image and "`psf`" as the blur kernel. We first generated the blurred images operating on images (a)–(c) by the former Gaussian blurs

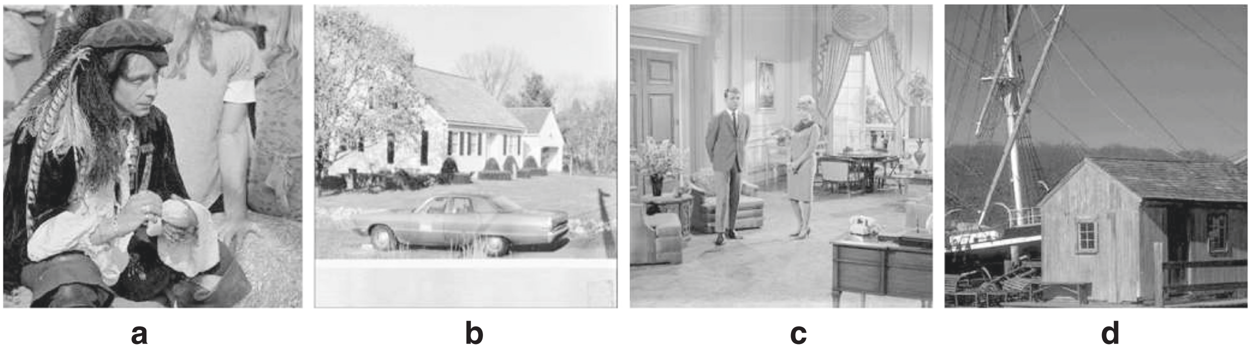

Fig. 5 Original images. The versions of the images in this paper are special formats which are converted by Photoshop from the sources downloaded from `http://decsai.ugr.es/cvg/dbimagenes/`. **a** Man. **b** Car. **c** Parlor. **d** Housepole

Table 2 Numerical comparison of Chan [32], Liu [11], CATVOGSL2, and CITVOGSL2 for images (a)–(c) in Fig. 5

Images		(a) Man				(b) Car				(c) Parlor			
Method	μ	Itr	PSNR	Time	ReE	Itr	PSNR	Time	ReE	Itr	PSNR	Time	ReE
Chan [32]	0.5	15	30.34	6.02	0.0730	11	31.13	2.00	0.0426	12	31.70	2.01	0.0511
Liu [11]	1	12	30.60	9.06	0.0708	12	31.68	3.93	0.0386	13	32.46	4.21	0.0464
CAOL2	1	7	30.62	*3.37*	0.0711	9	31.68	*1.76*	0.0399	7	32.40	*1.34*	0.0472
CIOL2	1	8	30.59	*3.70*	0.0716	11	31.60	*2.04*	0.0403	8	32.03	*1.61*	0.0492

PSNR dB, *Time* s, *Itr* iterations, μ regularization parameter ($\times 10^5$), *CAOL2* CATVOGSL2, *CIOL2* CITVOGSL2

and further corrupted by zero mean Gaussian noise with BSNR = 40. The BSNR is given by

$$\mathrm{BSNR} = 10\log_{10}\frac{\text{variance of} g}{\text{variance of } \eta}.$$

where g and η are the observed image and the noise, respectively.

The numerical results of the three methods are shown in Table 2. We have tuned the parameters for all the methods as in Table 2. From Table 2, we can see that PSNR and ReE of our methods (both ATV and ITV cases) are almost same as [11], which applied MM inner iterations to solve the subproblems (34) and (35) (only for the ATV case). However, each outer iteration of our methods is nearly twice faster than [11] from the experiments. The time of each outer iteration of our methods is almost the same as the traditional TV method in [32]. In Fig. 6, we display the restored "Parlor" images from different algorithms. We can see that OGS TV regularizers can get clearer edges on the "desk" of the image than TV regularizer.

Now, we compute the complexity of each step of our methods and Liu et al.'s method [11]. Firstly, we know that

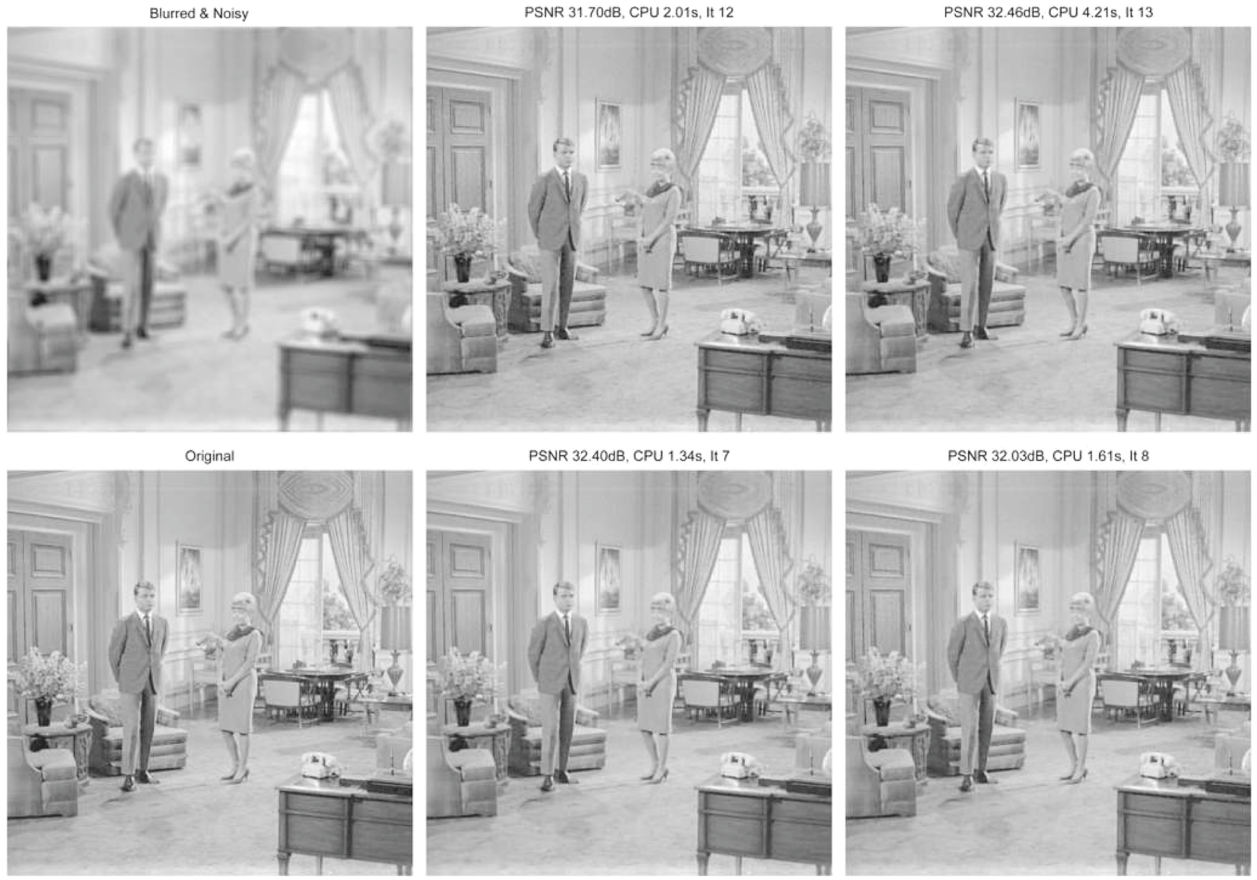
Blurred & Noisy
PSNR 31.70dB, CPU 2.01s, It 12
PSNR 32.46dB, CPU 4.21s, It 13
Original
PSNR 32.40dB, CPU 1.34s, It 7
PSNR 32.03dB, CPU 1.61s, It 8

Fig. 6 *Top row*: blurred and noisy image (*left*), restoration images of Chan [32] (*middle*), Liu [11] (*right*). *Bottom row*: original image (*left*), restoration images of CATVOGSL2 (*middle*), CITVOGSL2 (*right*)

the complexity of all the methods is 512 × 512 × 18 × 4 (4 times $n \log_2 n$) except the OGS subproblems. Then, for the OGS subproblems, the MM method in [11] with five-step inner iteration, the complexity is 512 × 512 × 90 × 2 (2 subproblems). The complexity of our methods is 512 × 512 × 18 × 2 in CATVOGSL2 and 512 × 512 × 18 in CITVOGSL2. Therefore, the total complexity of [11] is $512^2 \times 252$, the total complexity of CATVOGSL2 is $512^2 \times 108$, and the total complexity of CITVOGSL2 is $512^2 \times 90$. That means each step of our methods is more than double faster than the inner iteration method in [11]. In the next section, the common computation parts of our methods and the inner iteration method are much more, and then our methods are only nearly double faster.

Remarks 6. We do not list the results of image (d) because they are almost the same as images (b) and (c). Moreover, when $\beta_1 < 30$ which is not sufficiently large, the numerical results are also good while we did not list them. This shows that the approximate part of our shrinkage formula is also good, and that when the inner step in [11, 12] is chosen to be 5, the numerical experiments are convergent although they did not find a convergence control sequence. In addition, in our experiments, we find that the results of the ATV case are better than the ITV case, which is a little different from the classic TV methods. Moreover, we find that if the parameters of [11] are chosen to be the same as ours, the solutions of our method and their method are always the same on PSNR and visual presentation while our method may save more time (it remains valid for [12]).

5.3.2 *Experiments for the constrained TV OGS L_1 model*

In this section, we compare our methods (CATVOGSL1 and CITVOGSL1) with some other methods, such as Chan et al.'s method proposed in [32] (Algorithm 2 in [32] for the constrained TV-L1 model) and Liu et al.'s method proposed in [12].

Similarly as the last section, we set the penalty parameters $\beta_1 = 80$, $\beta_2 = 2000$, $\beta_3 = 1$, for the ATV case, $\beta_1 = 80$, $\beta_2 = 2000$, $\beta_3 = 1$, for the ITV case and relax parameter $\gamma = 1.618$. The blur kernel is generated by MATLAB built-in function `fspecial('gaussian',7,5)` for 7 × 7 Gaussian blur with standard deviation 5. We first generated the blurred images operating on images (b)–(d) by the former Gaussian blur and further corrupted them by salt-and-pepper noise from 30 to 40 %. We generate all noise effects by MATLAB built-in function `imnoise(Bl,'salt & pepper',level)` with "`Bl`" the blurred image and fix the same random matrix for different methods.

The numerical results by the three methods are shown in Table 3. We have tuned the parameters manually to give the best PSNR improvement for Chan [32] as in Table 3 for

Table 3 Numerical comparison of Chan [32], Liu [12], CATVOGSL1, and CITVOGSL1 for images (a)–(d) in Fig. 5

Is	N	Chan [32]				Liu [12]			
	L	μ/Itr	PSNR	Time	ReE	Itr	PSNR	Time	ReE
(a)	30	25/130	29.59	61.57	0.0796	35	30.92	30.23	0.0683
	40	18/ 99	28.85	46.41	0.0866	37	30.11	31.54	0.0749
(b)	30	25/128	29.71	22.90	0.0501	35	31.81	12.93	0.0393
	40	20/ 98	28.59	17.80	0.0570	40	30.70	14.99	0.0447
(c)	30	26/138	30.21	25.65	0.0607	35	32.15	12.84	0.0485
	40	22/105	29.10	19.27	0.0689	37	31.04	13.67	0.0551
(d)	30	26/127	30.41	23.17	0.0975	36	32.47	13.15	0.0769
	40	20/ 95	29.44	17.43	0.1091	39	31.43	14.27	0.0867
		CATVOGSL1				CITVOGSL1			
(a)	30	32	31.17	*17.97*	0.0663	41	31.34	*21.56*	0.0651
	40	32	30.10	*18.03*	0.0751	37	30.06	*19.95*	0.0754
(b)	30	26	31.77	*5.97*	0.0395	29	31.75	*6.26*	0.0396
	40	26	30.47	*6.12*	0.0459	29	30.22	*6.38*	0.0472
(c)	30	25	32.30	*5.82*	0.0477	30	32.08	*6.75*	0.0489
	40	26	31.01	*5.76*	0.0553	27	30.50	*6.19*	0.0587
(d)	30	25	32.54	*5.68*	0.0764	29	32.51	*6.75*	0.0766
	40	28	31.35	*6.40*	0.0875	32	31.06	*7.33*	0.0905

PSNR dB, *Time* s, *Is* images, *NL* noise level (%)

different images. For Liu [12], we choose the given parameters μ default as 100, 80 for 30 and 40 %, respectively. For our method CATVOGSL1, we set μ as 180, 140 for 30 and 40 %, respectively. For our method CITVOGSL1, we set μ as 140, 100 for 30 and 40 %, respectively. In the experiments, we find that the parameters of our methods are robust and have wide range to choose. Therefore, we set the same μ for different images.

From Table 3, we can also see that PSNR and ReE of our methods (both ATV and ITV cases) are almost the same as Liu [12], which applied MM inner iterations to solve the subproblems (34) and (35) (only for the ATV case). However, each outer iteration of our methods is nearly twice faster than Liu [12] from the experiments. The time of each outer iteration of our methods is almost the same as the traditional TV method in Chan [32]. Moreover, we can also see that sometimes ATV is better than ITV and sometimes on the contrary for OGS TV. Finally, in Fig. 7, we display the degraded image, the original image, and the restored images for 30 % level of noise on image (d) by the four methods. From the figure, we can see that both our methods and Liu [12] can get better edges (handrail and window) than Chan [32].

6 Conclusions

In this paper, we propose the explicit shrinkage formulas for one class of OGS regularization problems with translation invariant overlapping groups. These formulas can be extended to several other regularization OGS problems as a subproblem in many fields. In this work, we apply our results in OGS TV regularization problems—deblurring and denoising problems. Furthermore, we also extend the image deblurring problems with OGS ATV in [11, 12] to both ATV and ITV cases. Since the formulas are very simple, these results can be easily extended to many other applications such as multichannel deconvolution and compress sensing, which we will consider in the future. In addition, in the experiments, we only choose all the entries of the weight matrix W_g equal to 1. We will test for other weights in the future on more experiments in order to choose the better or best weights for some other applications.

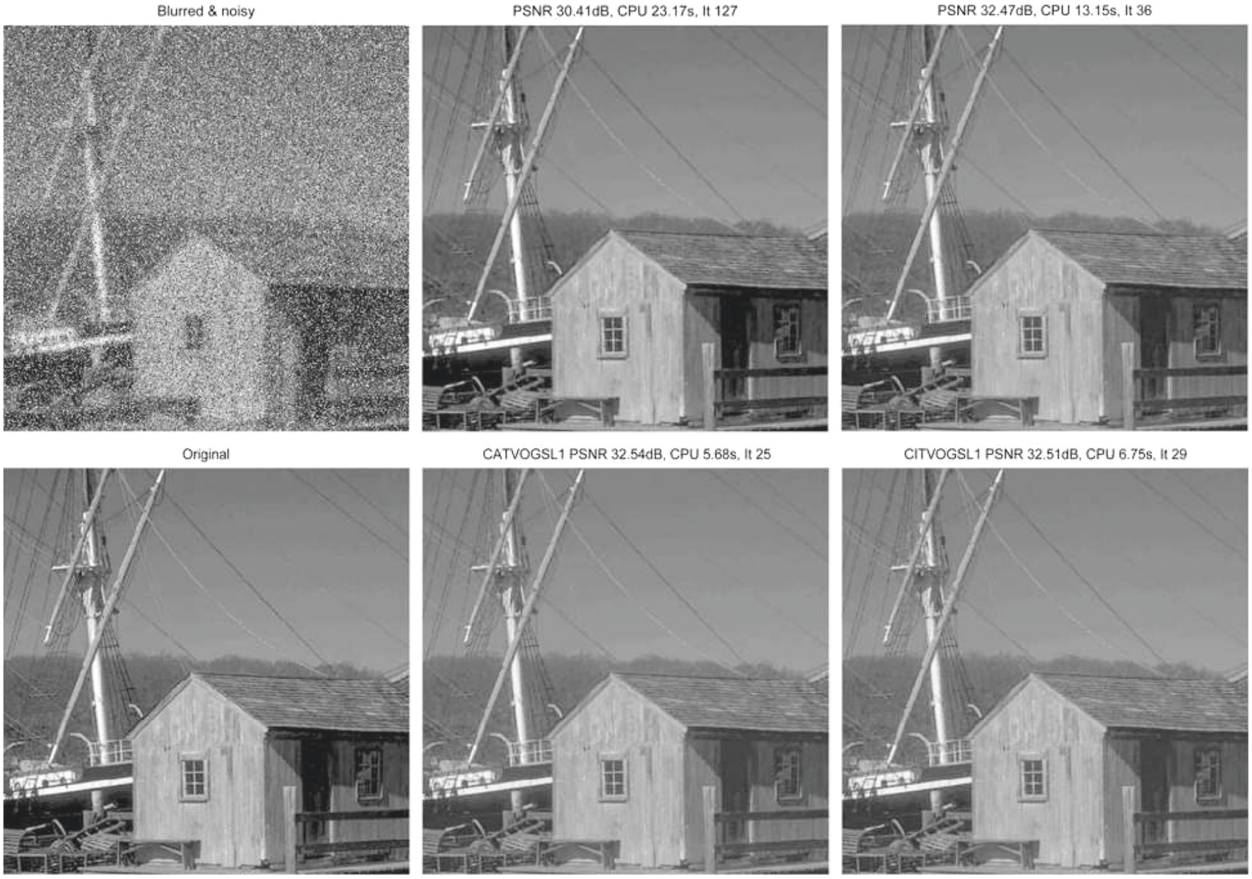
Blurred & noisy
PSNR 30.41dB, CPU 23.17s, It 127
PSNR 32.47dB, CPU 13.15s, It 36
Original
CATVOGSL1 PSNR 32.54dB, CPU 5.68s, It 25
CITVOGSL1 PSNR 32.51dB, CPU 6.75s, It 29

Fig. 7 *Top row*: blurred and noisy image (*left*), restoration images of Chan [32] (*middle*), Liu [12] (*right*). *Bottom row*: original image (*left*), restoration images of CATVOGSL1 (*middle*), CITVOGSL1 (*right*)

Appendix

For problem (4), similar to (14), we can get

$$\begin{aligned} f_w(\mathbf{z}) &= \min_{\mathbf{z}} \|\mathbf{z}\|_{w,2,1} + \frac{\beta}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \|w_g \circ (z_i)_g\|_2 + \frac{\beta}{2\sum_{k=1}^{s}(w_g)_k^2} \sum\nolimits_{k=1}^{s} (w_g)_k^2 \|\mathbf{z}-\mathbf{x}\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \|w_g \circ (z_i)_g\|_2 + \frac{\beta}{2\|w_g\|_2^2} \sum\nolimits_{i=1}^{n} \|w_g \circ \left((z_i)_g - (x_i)_g\right)\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \|w_g \circ (z_i)_g\|_2 + \frac{\beta}{2\|w_g\|_2^2} \sum\nolimits_{i=1}^{n} \|w_g \circ (z_i)_g - w_g \circ (x_i)_g\|_2^2 \\ &= \min_{\mathbf{z}} \sum\nolimits_{i=1}^{n} \left(\|w_g \circ (z_i)_g\|_2 + \frac{\beta}{2\|w_g\|_2^2} \|w_g \circ (z_i)_g - w_g \circ (x_i)_g\|_2^2 \right). \end{aligned} \tag{51}$$

All the symbols are the same as before. Similarly as before, we know that the necessary condition of minimizing the ith term of the last line in (51) is that $(w_g \circ (z_i)_g)$ is parallel to $(w_g \circ (x_i)_g)$. That is, $\frac{w_g \circ (z_i)_g}{\|w_g \circ (z_i)_g\|_2} = \frac{w_g \circ (x_i)_g}{\|w_g \circ (x_i)_g\|_2}$ for every i. On the other hand, Let $W = \mathrm{diag}(w_g)$ be a diagonal matrix with diagonal being the vector w_g, then $w_g \circ (x_i)_g = W(x_i)_g = W^T(x_i)_g$. If the vector $\mathbf{x}'$ (the same size as $(x_i)_g$) is parallel to the vector $\mathbf{z}'$, we have $\mathbf{x}' = \alpha\mathbf{z}'$. Then, $W\mathbf{x}' = W\alpha\mathbf{z}' = \alpha W\mathbf{z}'$. We obtain that the vector $W\mathbf{x}'$ is also parallel to the vector $W\mathbf{z}'$. Therefore, $\frac{W(z_i)_g}{\|W(z_i)_g\|_2} = \frac{W(x_i)_g}{\|W(x_i)_g\|_2}$, and each of them is a unit vector. Then, $\frac{W^TW(z_i)_g}{\|W(z_i)_g\|_2} = \frac{W^TW(x_i)_g}{\|W(x_i)_g\|_2}$. That is,

$$\frac{w_g \circ w_g \circ (z_i)_g}{\|w_g \circ (z_i)_g\|_2} = \frac{w_g \circ w_g \circ (x_i)_g}{\|w_g \circ (x_i)_g\|_2}.$$

In particular, we first consider that $s = 4$, $(z_i)_g = (z_{i-1}, z_i, z_{i+1}, z_{i+2})$ and $w_g = (w_1, w_2, w_3, w_4)$. We mark $(w_g \circ (z_i)_g)'$ as the expansion of $(w_g \circ (z_i)_g)$ similarly as $(z_i)_g'$, and we can get $\mathbf{x} = \frac{1}{\sum_{i=1}^{4} w_i} \sum_{i=1}^{n} (w_g \circ (z_i)_g)'$. Then, the Euler equation of $f_w(\mathbf{z})$ is given by

$$\beta\,(\mathbf{z}-\mathbf{x}) + \frac{(w_g \circ w_g \circ (z_1)_g)'}{\|(w_g \circ (z_1)_g)'\|_2} + \cdots + \frac{(w_g \circ w_g \circ (z_n)_g)'}{\|(w_g \circ (z_n)_g)'\|_2} \ni \mathbf{0}, \tag{52}$$

$$\begin{aligned} &\frac{\beta}{\sum_{i=1}^{4} w_i^2} \sum_{i=1}^{n} \left(w_g \circ (z_i)_g - w_g \circ (x_i)_g \right) + \frac{\left(w_g \circ w_g \circ (z_1)_g\right)'}{\|\left(w_g \circ (z_1)_g\right)'\|_2} \\ &\qquad + \cdots + \frac{\left(w_g \circ w_g \circ (z_n)_g\right)'}{\|\left(w_g \circ (z_n)_g\right)'\|_2} \ni \mathbf{0}, \end{aligned} \tag{53}$$

$$\beta\,(\mathbf{z}-\mathbf{x}) + \frac{(w_g \circ w_g \circ (x_1)_g)'}{\|(w_g \circ (x_1)_g)'\|_2} + \cdots + \frac{(w_g \circ w_g \circ (x_n)_g)'}{\|(w_g \circ (x_n)_g)'\|_2} \ni \mathbf{0}, \tag{54}$$

$$\mathbf{z} \ni \mathbf{x} - \frac{1}{\beta} \left(\frac{(w_g \circ w_g \circ (x_1)_g)'}{\|(w_g \circ (x_1)_g)'\|_2} + \cdots + \frac{(w_g \circ w_g \circ (x_n)_g)'}{\|(w_g \circ (x_n)_g)'\|_2} \right). \tag{55}$$

For each component, we obtained

$$\begin{aligned} z_i \ni x_i - \frac{1}{\beta} \Bigg(&\frac{w_4^2 x_i}{\|w_g \circ (x_{i-2})_g\|_2} + \frac{w_3^2 x_i}{\|w_g \circ (x_{i-1})_g\|_2} \\ &+ \frac{w_2^2 x_i}{\|w_g \circ (x_i)_g\|_2} + \frac{w_1^2 x_i}{\|w_g \circ (x_{i+1})_g\|_2} \Bigg). \end{aligned} \tag{56}$$

Another expression is as follows.

$$\begin{aligned} z_i \ni &\left(\frac{w_4^2 x_i}{\|w_g\|_2^2} - \frac{1}{\beta} \frac{w_4^2 x_i}{\|w_g \circ (x_{i-2})_g\|_2} \right) + \left(\frac{w_3^2 x_i}{\|w_g\|_2^2} - \frac{1}{\beta} \frac{w_3^2 x_i}{\|w_g \circ (x_{i-1})_g\|_2} \right) \\ &+ \cdots + \left(\frac{w_1^2 x_i}{\|w_g\|_2^2} - \frac{1}{\beta} \frac{w_1^2 x_i}{\|w_g \circ (x_{i+1})_g\|_2} \right). \end{aligned} \tag{57}$$

Similarly as Algorithm 1, we obtain the following algorithm for finding the minimizer of (4).

Algorithm 5 Direct shrinkage algorithm for the minimization problem (4)

```
Input:
  Given vector $\mathbf{x}$, group size $s$, parameter $\beta$, weight vector
  $w_g = (w_1, w_2, \cdots, w_s)$.
Compute:
Definition of $(x_i)_g$, for example $(x_i)_g = \left(x_{i-s_l}, \cdots, x_i, \cdots, x_{i+s_r}\right)$, with $s_l = [\frac{s-1}{2}]$ and $s_r = [\frac{s}{2}]$.
If $s$ is even, then $w_g = [0, w_g]$ and $s = s + 1$.
Let $w_g^r$ = fliplr($w_g$) be fliping $w_g$ over 180 degrees.
Compute $X_n = \left(\|(x_1)_g\|_2, \cdots, \|(x_i)_g\|_2, \cdots, \|(x_n)_g\|_2\right)$,
    by correlation of $w_g.^2$ (pointwise square)
    and $\mathbf{x}$, or by convolution of $w_g^r.^2$ and $\mathbf{x}$.
Compute $X_n' = \max\left(\frac{1}{\|w_g\|_2^2} - 1./(X_n \cdot \beta), 0\right)$ pointwise.
Compute $F(x_i)$ by convolution of $w_g.^2$ and $X_n'$, or by
    correlation of $w_g^r.^2$ and $X_n'$.
```

We can also see that Algorithm 2 only needs 2 times of convolution computations with time complexity $n * s$, which is just the same time complexity as one-step iteration in the MM method in [10]. Therefore, our method is much more efficient than the MM method.

Here, due to the properties of inequalities, without loss of generality, let $\mathbf{x} \in [0,1]^n$, then we obtain that if $\beta \le \frac{\|w_g\|_2}{\sqrt{s}} \le \frac{\|w_g\|_2}{\|x_g\|_2} = \frac{\|w_g\|_2^2}{\|w_g\|\|x_g\|_2} \le \frac{\|w_g\|_2^2}{\|w_g \cdot x_g\|_2}$, β is sufficiently small. However, we do not directly know why β is sufficiently large. In Section 5, after more than 1000 tests, we find that $\beta \ge 30 \cdot \frac{\|w_g\|_2}{\sqrt{s}}$ is sufficiently large generally.

For problem (5), similar to (51), we can obtain

$$\begin{aligned}
f_W(A) &= \min_A \|A\|_{W,2,1} + \tfrac{\beta}{2}\|A - X\|_F^2 \\
&= \min_A \textstyle\sum_{i=1}^m \sum_{j=1}^n \|W_g \circ (A_{i,j})_g\|_F + \\
&\quad \tfrac{\beta}{2\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}(W_g)_{k_1,k_2}^2} \textstyle\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}(W_g)_{k_1,k_2}^2 \|A - X\|_F^2 \\
&= \min_A \textstyle\sum_{i=1}^m \sum_{j=1}^n \|W_g \circ (A_{i,j})_g\|_F + \\
&\quad \tfrac{\beta}{2\|W_g\|_F^2} \textstyle\sum_{i=1}^m \sum_{j=1}^n \|W_g \circ \left((A_{i,j})_g - (X_{i,j})_g\right)\|_F^2 \\
&= \min_A \textstyle\sum_{i=1}^m \sum_{j=1}^n \|W_g \circ (A_{i,j})_g\|_F + \\
&\quad \tfrac{\beta}{2\|W_g\|_F^2} \textstyle\sum_{i=1}^m \sum_{j=1}^n \|W_g \circ (A_{i,j})_g - W_g \circ (X_{i,j})_g\|_F^2 \\
&= \min_A \textstyle\sum_{i=1}^m \sum_{j=1}^n \left(\|W_g \circ (A_{i,j})_g\|_F + \tfrac{\beta}{2\|W_g\|_F^2}\|W_g \circ (A_{i,j})_g - W_g \circ (X_{i,j})_g\|_F^2\right).
\end{aligned} \tag{58}$$

For example, we set $K_1 = 2, K_2 = 2$ and define $(A_{i,j})_g = (A_{i,j}, A_{i,j+1}; A_{i+1,j}, A_{i+1,j+1})$ and $W_g = (W_{1,1}, W_{1,2}; W_{2,1}, W_{2,2})$. Similar as the vector case, each $(A_{i,j})_g$ is a matrix with $\|(A_{i,j})_g\|_F = \sqrt{\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}((A_{i,j})_g)_{k_1,k_2}^2}$. Notice that the Frobenius norm of a matrix is equal to the ℓ_2 norm of a vector reshaped by the matrix. Then, the Euler equation of $f_w(\mathbf{z})$ is given by

$$\begin{aligned}
\beta(A - F) &+ \frac{(W_g \circ W_g \circ (A_{1,1})_g)'}{\|(W_g \circ (A_{1,1})_g)'\|_2} + \cdots \\
&+ \frac{(W_g \circ W_g \circ (A_{n,n})_g)'}{\|(W_g \circ (A_{n,n})_g)'\|_2} \ni \mathbf{0},
\end{aligned} \tag{59}$$

where $((A_{i,j})_g)'$ is defined similarly as $((z_i)_g)'$, which is an expansion of $(A_{i,j})_g$. These symbols remain consistent as default through this paper.

$$\begin{aligned}
\beta(A - X) &+ \frac{(W_g \circ W_g \circ (X_{1,1})_g)'}{\|(W_g \circ (X_{1,1})_g)'\|_2} + \cdots \\
&+ \frac{(W_g \circ W_g \circ (X_{n,n})_g)'}{\|(W_g \circ (X_{n,n})_g)'\|_2} \ni \mathbf{0},
\end{aligned} \tag{60}$$

$$\begin{aligned}
A \ni X - \frac{1}{\beta}\Bigg(&\frac{(W_g \circ W_g \circ (X_{1,1})_g)'}{\|(W_g \circ (X_{1,1})_g)'\|_2} + \cdots \\
&+ \frac{(W_g \circ W_g \circ (X_{n,n})_g)'}{\|(W_g \circ (X_{n,n})_g)'\|_2}\Bigg).
\end{aligned} \tag{61}$$

For each component, we obtained

$$\begin{aligned}
A_{i,j} \ni X_{i,j} - \tfrac{1}{\beta}\Bigg(&\frac{W_{2,2}^2 X_{i,j}}{\|W_g \circ (X_{i-1,j-1})_g\|_2} + \frac{W_{2,1}^2 X_{i,j}}{\|W_g \circ (X_{i-1,j})_g\|_2} + \\
&\frac{W_{1,2}^2 X_{i,j}}{\|W_g \circ (X_{i,j-1})_g\|_2} + \frac{W_{1,1}^2 X_{i,j}}{\|W_g \circ (X_{i,j})_g\|_2}\Bigg).
\end{aligned} \tag{62}$$

Therefore, we can obtain a similar algorithm on the former formula (62) for finding the minimizer of (5).

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

GL carried out the main part of this manuscript. TH participated in the discussion and carried out the numerical results. XL and JL participated in the discussion and corrected the English errors. All authors read and approved the final manuscript.

Acknowledgements

The work of Gang Liu, Ting-Zhu Huang, and Jun Liu was supported by the 973 Program (2013CB329404), NSFC (61370147), and Fundamental Research Funds for the Central Universities (ZYGX2013Z005). The work of Xiao-Guang Lv is supported by NSFC (61401172).

Author details

[1]College of Applied Mathematics, Chengdu University of Information Technology, Xuefu Road, 610225 Chengdu, China. [2]Research Center for Image and Vision Computing/School of Mathematical Sciences, University of Electronic Science and Technology of China, Xiyuan Ave, 611731 Chengdu, China. [3]School of Science, Huaihai Institute of Technology, Cangwu Road, 222005 Lianyungang, China.

Received: 13 July 2015 Accepted: 12 April 2016
Published online: 29 April 2016

References

1. W Deng, W Yin, Y Zhang, in *Proc. SPIE*. Group sparse optimization by alternating direction method (SPIE, San Diego, US, 2013), pp. 88580R–15
2. J Huang, T Zhang, The benefit of group sparsity. Ann. Stat. **38**(4), 1978–2004 (2010). doi:10.1214/09-AOS778
3. R Chartrand, B Wohlberg, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. A nonconvex ADMM algorithm for group sparsity with sparse groups (IEEE, Vancouver, BC, 2013), pp. 6009–6013
4. M Stojnic, F Parvaresh, B Hassibi, On the reconstruction of block-sparse signals with an optimal number of measurements. IEEE Trans. Signal Process. **57**(8), 3075–3085 (2009)
5. L Jacob, G Obozinski, J-P Vert, in *Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09*. Group lasso with overlap and graph lasso (ACM, New York, 2009), pp. 433–440
6. A Majumdar, RK Ward, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Classification via group sparsity promoting regularization (IEEE, Taipei, 2009), pp. 861–864
7. E Elhamifar, R Vidal, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Robust Classification using structured sparse representation (IEEE, Providence, 2011), pp. 1873–1879
8. J Gao, Q Shi, TS Caetano, Dimensionality reduction via compressive sensing. Pattern Recognit. Lett. **33**(9), 1163–1170 (2012)
9. YN Liu, F Wu, YT Zhuang, Group sparse representation for image categorization and semantic video retrieval. Sci. China Inform. Sci. **54**(10), 2051–2063 (2011)
10. P-Y Chen, IW Selesnick, Translation-invariant shrinkage/thresholding of group sparse signals. Signal Process. **94**(0), 476–489 (2014)
11. J Liu, T-Z Huang, IW Selesnick, X-G Lv, P-Y Chen, Image restoration using total variation with overlapping group sparsity. Inform. Sci. **295**(20), 232–246 (2015)
12. G Liu, T-Z Huang, J Liu, X-G Lv, Total variation with overlapping group sparsity for image deblurring under impulse noise. Plos One. **10**(4), 0122562 (2015)
13. IW Selesnick, P-Y Chen, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Total variation denoising with overlapping group sparsity (IEEE, Vancouver, BC, 2013), pp. 5696–5700
14. M Figueiredo, J Bioucas-Dias, in *Signal Processing with Adaptive Sparse Structured Representations - SPARS11*. An alternating direction algorithm for (overlapping) group regularization, (Edinburgh, Scotland, UK, 2011)
15. R Jenatton, J-Y Audibert, F Bach, Structured variable selection with sparsity-inducing norms. J. Mach. Learn. Res. **12**, 2777–2824 (2011)
16. M Kowalski, Sparse regression using mixed norms. Appl. Comput. Harmonic Anal. **27**(3), 303–324 (2009)
17. G Peyré, JM Fadili, in *EUSIPCO*. Group sparsity with overlapping partition functions, (Barcelona, Spain, 2011), pp. 303–307
18. F Bach, R Jenatton, J Mairal, G Obozinski, Structured sparsity through convex optimization. Stat. Sci. **27**(4), 450–468 (2012)

19. I Bayram, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mixed norms with overlapping groups as signal priors (IEEE, Prague, 2011), pp. 4036–4039
20. Y Wang, J Yang, W Yin, Y Zhang, A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**(3), 248–272 (2008)
21. J Yang, W Yin, Y Zhang, Y Wang, A fast algorithm for edge-preserving variational multichannel image restoration. SIAM J. Imaging Sci. **2**(2), 569–592 (2009). doi:10.1137/080730421
22. B Wohlberg, R Chartrand, J Theiler, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Local principal component pursuit for nonlinear datasets (IEEE, Kyoto, 2012), pp. 3925–3928
23. P Sprechmann, I Ramirez, G Sapiro, YC Eldar, C-HiLasso: a collaborative hierarchical sparse modeling framework. IEEE Trans. Signal Process. **59**(9), 4183–4198 (2011)
24. PC Hansen, JG Nagy, DP O'Leary, *Deblurring images: matrices, spectra, and filtering*, 1st edn. (Society for Industrial and Applied Mathematics, Philadelphia, United States, 2006)
25. LI Rudin, S Osher, E Fatemi, Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena. **60**(1–4), 259–268 (1992)
26. A Chambolle, An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004)
27. G Liu, T-Z Huang, J Liu, High-order TVL1-based images restoration and spatially adapted regularization parameter selection. Comput. Math. Appl. **67**(10), 2015–2026 (2014)
28. X-L Zhao, W Wang, T-Y Zeng, T-Z Huang, MK Ng, Total variation structured total least squares method for image restoration. SIAM J. Sci. Comput. **35**(6), 1304–1320 (2013)
29. X-L Zhao, F Wang, MK Ng, A new convex optimization model for multiplicative noise and blur removal. SIAM J. Imaging Sci. **7**(1), 456–475 (2014)
30. C Chen, MK Ng, X-L Zhao, Alternating direction method of multipliers for nonlinear image restoration problems. IEEE Trans. Image Process. **24**(1), 33–43 (2015)
31. L-J Deng, H Guo, T-Z Huang, A fast image recovery algorithm based on splitting deblurring and denoising. J. Comput. Appl. Math. **287**, 88–97 (2015)
32. R Chan, M Tao, X Yuan, Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers. SIAM J. Imaging Sci. **6**(1), 680–697 (2013)
33. J Nocedal, SJ Wright, *Numerical optimization*. 1431–8598. (Springer, New York, 2006), p. 664
34. D Gabay, B Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976)
35. J Eckstein, D Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Programm. **55**(1–3), 293–318 (1992). doi:10.1007/BF01581204
36. R Glowinski, *Numerical methods for nonlinear variational problems*. 1434–8322. (Springer, Berlin Heidelberg, 1984), p. 493
37. B He, H Yang, Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. Oper. Res. Lett. **23**(3–5), 151–161 (1998)
38. E Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman. Los Angeles: UCLA CAM report, 9–31 (2009). ftp://arachne.math.ucla.edu/pub/camreport/cam09-31.pdf
39. M Ng, P Weiss, X Yuan, Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM J. Sci. Comput. **32**(5), 2710–2736 (2010)
40. X Zhang, M Burger, X Bresson, S Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. SIAM J. Imaging Sci. **3**(3), 253–276 (2010)
41. J Yang, Y Zhang, W Yin, A fast alternating direction method for TVL1-L2 signal reconstruction from partial fourier data. IEEE J. Selected Topics Signal Process. **4**(2), 288–297 (2010)
42. X Zhang, M Burger, S Osher, A unified primal-dual algorithm framework based on Bregman iteration. J. Sci. Comput. **46**(1), 20–46 (2011)
43. S Boyd, N Parikh, E Chu, B Peleato, J Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations Trends Mach. Learn. **3**(1), 1–122 (2011)
44. C Wu, X Tai, Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. SIAM J. Imaging Sci. **3**(3), 300–339 (2010)