Research Article

A Coordinate Descent Method for Total Variation Minimization

Hong Deng,[1] Dongwei Ren,[2] Gang Xiao,[3] David Zhang,[2,4] and Wangmeng Zuo[2]

[1]*Northeast Agricultural University, Harbin 150001, China*
[2]*Harbin Institute of Technology, Research Group for Computational Photography and Statistical Learning,*
 School of Computer Science and Technology, Harbin 150001, China
[3]*No. 211 Hospital of PLA, Harbin 150001, China*
[4]*Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Correspondence should be addressed to Wangmeng Zuo; cswmzuo@gmail.com

Total variation (TV) is a well-known image model with extensive applications in various images and vision tasks, for example, denoising, deblurring, superresolution, inpainting, and compressed sensing. In this paper, we systematically study the coordinate descent (CoD) method for solving general total variation (TV) minimization problems. Based on multidirectional gradients representation, the proposed CoD method provides a unified solution for both anisotropic and isotropic TV-based denoising (CoDenoise). With sequential sweeping and small random perturbations, CoDenoise is efficient in denoising and empirically converges to optimal solution. Moreover, CoDenoise also delivers new perspective on understanding recursive weighted median filtering. By incorporating with the Augmented Lagrangian Method (ALM), CoD was further extended to TV-based image deblurring (ALMCD). The results on denoising and deblurring validate the efficiency and effectiveness of the CoD-based methods.

1. Introduction

Total variation (TV), also known as the ROF model [1], was introduced by Rudin et al. The TV model is effective in preserving sharp and salient edges while suppressing noise and has been extensively adopted as a regularizer in various image restoration applications, for example, deblurring [2, 3], super-resolution [4, 5], inpainting [6, 7], and compressed sensing [8, 9].

Recently, other image models, such as dictionary-based sparse coding [10–12] and nonlocal similarity [13–17], have been developed. Compared with these models, TV is much more efficient to be solved, making TV-based methods remain active in image and vision studies [17–24]. Moreover, TV may be complementary with the other models, and thus proper combination of them can lead to better performance [25, 26]. Besides, extensions of TV regularizer were also studied. For color images, TV can be extended to a class of vectorial TV (VTV) [27, 28], where interchannel correlation is taken into account to reduce the uneven color effects. While TV only considers first-order gradients, Total Generalized Variation (TGV) [29] was proposed to involve higher-order derivatives. For structure extraction, relative TV [30] was employed to distinguish structure from textures. Considering that the gradient distribution of each pixel is actually spatially variant, nonlocal extension of TV model [17, 31] was presented to leverage the similar patches for adaptive distribution estimation.

A basic TV minimization problem is TV-based image denoising formulated as

$$\min_{\mathbf{x}} \quad \frac{1}{2} \left\| \mathbf{x} - \mathbf{y} \right\|^2 + \lambda \mathrm{TV}\left(\mathbf{x}\right), \tag{1}$$

where $\mathrm{TV}(\mathbf{x})$ is the TV regularizer, λ is the trade-off parameter, and $\mathbf{x}$ and $\mathbf{y}$ are the latent clear image and the noisy observation, respectively. Various methods to solve TV denoising problem had been proposed and can be roughly categorized from three directions, that is, gradient based, Markov Random Fields (MRF) based, and CoD-based methods. First,

gradient descent-based algorithms have been widely adopted in image processing tasks [18, 32–37]. As to TV minimization, gradient projection based PDE methods [1] originally were adopted to solve the associated nonlinear Euler-Lagrange equation. Following this line, a number of methods tried to directly solve primal variables [38–42]. To avoid nonsmoothness trap, the dual formulation of TV minimization was proposed and several variants came forward [43–45]. Recently, a hybrid primal dual scheme that alternatively solves primal and dual variables had been developed [46–48]. Most specially, Chambolle's fixed point algorithm [43] solving dual variable is the most successful, which has been widely adopted in general image restoration methods, for example, TwIST [49], FISTA [50], and SALSA [51]. Second, TV minimization can be mapped to a class of binary MRFs [52–54], such that it can be solved by graph-cut techniques. Third, another entirely different direction is to employ CoD method, decomposing optimization problem with respect to each pixel and updating coordinate variables via some appropriate patterns. For the high efficiency of decomposed scalar optimizations, the CoD-based methods are usually efficient. However, the sole attempt based on CoD to solve TV minimization [55] only considers the anisotropic TV minimization, while isotropic TV minimization is unreachable for CoD-based methods, since it cannot be decomposed with respect to each pixel.

In this paper, we systematically study the CoD-based methods for TV minimization problem. First, we provide a unified formulation of anisotropic and isotropic TV minimization problem based on multidirectional gradients representation, via which the isotropic TV regularizer can also be decomposed into a sequence of scalar convex problems with respect to each pixel. The scalar convex problem can be efficiently solved, and by sequentially updating each pixel, the CoD-based denoising (CoDenoise) algorithm converges fast. Due to the nondifferentiability of TV regularizer, CoDenoise may get stuck at nonstationary points [55–57]; however fortunately it is experimentally verified that CoDenoise can bypass nonstationary points and converge to optimal solution by adding small random perturbations. The CoDenoise algorithm only requires updating the pixels poisoned by noises, due to which the CoDenoise algorithm is more efficient than other methods, especially for low noise levels. Interestingly, the CoDenoise algorithm can be interpreted as the recursive weighted median operations on noisy images. Based on the more recent progress in weighted median filter [58, 59], the CoDenoise algorithm should be much more improved in terms of efficiency. Then, by combining variable splitting strategy and Augmented Lagrangian Method (ALM), we further embed CoDenoise algorithm to solve general image restoration problem, for example, image deblurring, resulting in the ALMCD algorithm. In deblurring problems, the blurry images are usually poisoned by relatively low level noises, and thus the incorporated CoDenoise algorithm for denoising subproblem contributes significantly to efficiency improvement of the ALMCD algorithm. Compared with TwIST, FISTA, and SALSA, ALMCD can obtain satisfactory results but is more efficient.

Our contribution can be summarized from two aspects:

(i) We systematically study the CoD-based methods for TV minimization and develop an extremely simple unified CoD-based solution for both anisotropic and isotropic TV minimization. The resulting CoDenoise algorithm is more efficient than gradient based and MRF based methods and achieves satisfactory denoising results.

(ii) By incorporating with ALM, CoDenoise is extended to image deblurring problem. In the deblurring problems, the blurry images usually suffer from severe blur and relatively low level noises, and thus the proposed ALMCD algorithm with CoDenoise embedded for denoising subproblem is much more efficient and can concurrently provide satisfactory deblurring quality compared with several state-of-the-art methods.

This paper is organized as follows: Section 2 presents some preliminaries, including definition of TV regularizers and multidirectional gradient approximation of TV regularizers. The CoDenoise algorithm together with its convergence proof and computational complexity is proposed in Section 3. In Section 4, we embed CoDenoise to image deblurring. Section 5 demonstrates experimental results, and Section 6 ends this paper with some concluding remarks.

2. Preliminaries

In this section, we first present the definitions of the discrete anisotropic and isotropic TV operators. In previous studies, CoD-based solution is only available for anisotropic TV minimization problem. To address this, we then introduce the multidirectional gradient representation to establish the connection between the anisotropic and isotropic TV models, making it possible to use the unified CoD method for TV minimization.

2.1. The Discrete TV Operators. For an image $\mathbf{x}$ with $N = m \times n$ pixels, the discrete gradient operators $\mathscr{D}$ including both horizontal gradient operator $\mathscr{D}_h$ and vertical gradient operator $\mathscr{D}_v$ are defined as

$$
\begin{aligned}
(\mathscr{D}_h \mathbf{x})_{k,l} = x_{k,l} - x_{k,l-1}, \quad \text{with } x_{k,0} = x_{k,n}, \\
(\mathscr{D}_v \mathbf{x})_{k,l} = x_{k,l} - x_{k-1,l}, \quad \text{with } x_{0,l} = x_{m,l},
\end{aligned}
\tag{2}
$$

where $k = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, n$. The anisotropic TV regularizer [50, 60] is defined as

$$
\text{TV}_a(\mathbf{x}) = \sum_{k=1}^{m} \sum_{l=1}^{n} \left(\left| (\mathscr{D}_h \mathbf{x})_{k,l} \right| + \left| (\mathscr{D}_v \mathbf{x})_{k,l} \right| \right). \tag{3}
$$

With this definition, it is easy to obtain the anisotropic TV regularization with respect to coordinate as

$$
\begin{aligned}
\text{TV}_a(x_{k,l}) = \left| x_{k,l} - x_{k-1,l} \right| + \left| x_{k,l} - x_{k,l-1} \right| \\
+ \left| x_{k,l} - x_{k+1,l} \right| + \left| x_{k,l} - x_{k,l+1} \right|.
\end{aligned}
\tag{4}
$$

Thus, the CoD method can be directly used to solve the anisotropic TV minimization problem. Similarly, the isotropic TV regularizer [50, 60] is defined as

$$TV_i(\mathbf{x}) = \sum_{k=1}^{m} \sum_{l=1}^{n} \sqrt{(\mathscr{D}_h \mathbf{x})_{k,l}^2 + (\mathscr{D}_v \mathbf{x})_{k,l}^2}. \tag{5}$$

Apparently the isotropic TV cannot be decomposed with respect to coordinate (k, l) since the quadratic interactions with horizontal and vertical gradients, making the CoD method unfeasible to solve isotropic TV minimization problem. Therefore, to extend the results of CoD to isotropic TV minimization problem, we tempt to find a connection between TV_a and TV_i.

2.2. Multidirectional Gradients Approximation. The isotropic TV regularizer can be approximated by multidirectional gradients representation, and thus the anisotropic and isotropic TV models can be connected in a unified formulation [61]. For any pair of real numbers a and b, the identity

$$\frac{\int_0^{\pi/2} (|a\cos\theta + b\sin\theta| + |b\cos\theta - a\sin\theta|)\, d\theta}{\int_0^{\pi/2} (\cos\theta + \sin\theta)\, d\theta}$$

$$= \sqrt{a^2 + b^2} \tag{6}$$

always holds, which can be discretized by Riemannian approximation. Now, let $\boldsymbol{\theta}_L = [\theta_1, \theta_2, \ldots, \theta_L]^T$ be a set of L points uniformly distributed in $[0, \pi/2)$. Equation (6) can then be discretized as

$$I(a, b; L)$$

$$= \frac{\sum_{i=1}^{L} (|a\cos\theta_i + b\sin\theta_i| + |b\cos\theta_i - a\sin\theta_i|)}{\sum_{i=1}^{L} (\cos\theta_i + \sin\theta_i)}. \tag{7}$$

Thus we can approximate TV regularizer as

$$TV_L(\mathbf{x}) = \sum_{k=1}^{m} \sum_{l=1}^{n} I\left((\mathbf{d}_h)_{k,l}, (\mathbf{d}_v)_{k,l}; L\right)$$

$$= d_L \sum_{k=1}^{m} \sum_{l=1}^{n} \sum_{i=1}^{L} \left| (\mathbf{d}_h)_{k,l} \cos\theta_i + (\mathbf{d}_v)_{k,l} \sin\theta_i \right| \tag{8}$$

$$+ \left| (\mathbf{d}_v)_{k,l} \cos\theta_i - (\mathbf{d}_h)_{k,l} \sin\theta_i \right|,$$

where $d_L = (\sum_{i=1}^{L} (\cos\theta_i + \sin\theta_i))^{-1}$ and $\mathbf{d}_h = \mathscr{D}_h \mathbf{x}$ and $\mathbf{d}_v = \mathscr{D}_v \mathbf{x}$. Equation (8) provides a unified formulation of anisotropic and isotropic TV models,

$$TV_L(\mathbf{x}) = \begin{cases} TV_a(\mathbf{x}), & L = 1, \\ TV_i(\mathbf{x}), & L \longrightarrow \infty. \end{cases} \tag{9}$$

In later context, we will use TV_L to represent the TV regularizers.

3. The Unified Coordinate Descent Method for TV-Based Denoising

With TV_L regularizer, anisotropic and isotropic TV denoising models are reformulated in the unified form

$$\min_{\mathbf{x}} \quad F(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \lambda TV_L(\mathbf{x}) \tag{10}$$

which is exactly anisotropic TV-based denoising when $L = 1$ and infinitely approximates isotropic TV-based denoising when L increases. We thus can decompose the objective function into a sequence of one-dimensional subproblems, which can be solved efficiently via simple convex optimization. With simple sequential updating pattern, we then obtain the unified CoD denoising algorithm for both anisotropic and isotropic TV minimization.

3.1. The Coordinate Subproblem. Let first present equivalent decomposition of the TV_L image denoising objective function with respect to each pixel $x_{k,l}$,

$$\min_{x_{k,l}} \quad F(\mathbf{x}) = \min_{x_{k,l}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \lambda TV_L(\mathbf{x})$$

$$= \frac{1}{2} (x_{k,l} - y_{k,l})^2$$

$$+ \lambda d_L \sum_{i=1}^{L} (|(c_i + s_i) x_{k,l} - (c_i x_{k,l-1} + s_i x_{k-1,l})| + |(c_i - s_i) x_{k,l} - (c_i x_{k-1,l} - s_i x_{k,l-1})| + |(c_i + s_i) x_{k,l} - (c_i x_{k,l+1} + s_i x_{k+1,l})| + |(c_i - s_i) x_{k,l} - (c_i x_{k+1,l} - s_i x_{k,l+1})|)$$

$$= \frac{1}{2} (x_{k,l} - y_{k,l})^2 + \lambda d_L \sum_{i=1}^{4L} |a_i x_{k,l} - c_i|, \tag{11}$$

where $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. Vectors $\mathbf{a}$ and $\mathbf{c}$ are both of length $4 \times L$, which are the coefficients of $x_{k,l}$ and the combinations of its 4 neighbourhoods, respectively.

3.2. Solving Subproblem. For simplicity, we unify the formulation of subproblems as

$$\min_{x} \quad f(x) = \frac{1}{2}(x-b)^2 + \tau \sum_{i=1}^{d} |a_i x - c_i|. \tag{12}$$

The scalar optimization problem is convex but nonsmooth. We assume $a_i > 0$ (the case $a_i < 0$ can be easily generalized). Let $(l_1, l_2, \ldots, l_d)$ be the permutation of $(1, 2, \ldots, d)$ according to the ascending order of $(c_1/a_1, c_2/a_2, \ldots, c_d/a_d)$. Let $(s_0, s_1, \ldots, s_d)$ be an ascending sequence with $s_0 = -\sum_{j=1}^{d} a_{l_j}$, $s_d = \sum_{j=1}^{d} a_{l_j}$, and $s_i = \sum_{j=1}^{i} a_{l_j} - \sum_{j=i+1}^{d} a_{l_j}$. Let $g_i(x) = |x - c_i/a_i|$. Thus, (12) is transformed to

$$\min_{x} \quad f(x) = \frac{1}{2}(x-b)^2 + \tau \sum_{i=1}^{d} a_{l_i} \left| x - \frac{c_{l_i}}{a_{l_i}} \right|$$
$$= \frac{1}{2}(x-b)^2 + \tau \sum_{i=1}^{d} a_{l_i} g_{l_i}(x). \tag{13}$$

The solution to (13) can be obtained by making its first-order derivative be 0,

$$\partial f(x) = (x-b) + \tau \sum_{i=1}^{d} a_{l_i} \partial g_{l_i}(x) = 0. \tag{14}$$

We then discuss the solution with different cases of x:

(1) When $x < c_{l_1}/a_{l_1}$,

$$\partial f(x) = (x-b) + \tau s_0 = 0. \tag{15}$$

Then $x^* = b - \tau s_0$ is the optimal solution to (13), if $x^* < c_{l_1}/a_{l_1}$.

(2) When $x > c_{l_d}/a_{l_d}$,

$$\partial f(x) = (x-b) + \tau s_d = 0. \tag{16}$$

Then $x^* = b - \tau s_d$ is the optimal solution to (13), if $x^* > c_{l_d}/a_{l_d}$.

(3) When $c_{l_i}/a_{l_i} < x < c_{l_{i+1}}/a_{l_{i+1}}$,

$$\partial f(x) = (x-b) + \tau s_i = 0. \tag{17}$$

Then $x^* = b - \tau s_i$ is the optimal solution to (13), if $c_{l_i}/a_{l_i} < x^* < c_{l_{i+1}}/a_{l_{i+1}}$.

(4) When $x = c_{l_i}/a_{l_i}$,

$$\partial f(x) = (x-b) + \tau \partial g_{l_i}(x) + \tau s_i = 0. \tag{18}$$

Since $\partial g_{l_d}(x) \in [-a_{l_i}, a_{l_i}]$, for $x = c_{l_i}/a_{l_i}$, we thus have

$$0 \in \partial f(x) = (x-b) + \tau [s_{i-1}, s_i] \tag{19}$$

and $x \in [b - \tau s_i, b - \tau s_{i-1}]$. Then $x^* = c_{l_i}/a_{l_i}$ is the optimal solution to (13), if $x^* \in [b - \tau s_i, b - \tau s_{i-1}]$.

As a summary, we notate procedures (1)–(4) as an operator,

$$x^* = \mathscr{P}(\tau, b, \mathbf{a}, \mathbf{c}). \tag{20}$$

Interestingly, solution (20) can be interpreted as finding the median value of vector (28), which is discussed in Section 3.4.3.

```
Input: y, L
Output: x.
(1)  Initialize t = 0, M = zeros(m, n), x^t = y
(2)  while not converged do
(3)      for each coordinate (k, l) do.
(4)          if M(k, l) = 0 then
(5)              x_{k,l}^{(t+1)} = 𝒫(λd_L, y_{k,l}, a, c)
(6)              Set M(k, l) as 1
(7)              if |x_{k,l}^{(t+1)} - x_{k,l}^{(t)}| > ε_0 then
(8)                  Set M(k - 1, l), M(k + 1, l), M(k, l - 1),
         M(k, l + 1) as 0
(9)              end if
(10)         end if
(11)     end for
(12)     t = t + 1
(13) end while
```

ALGORITHM 1: CoDenoise.

3.3. CoDenoise. Therefore, the subproblem with respect to $x_{k,l}$ (11) can be solved by

$$x_{k,l} = \mathscr{P}(\lambda d_L, y_{k,l}, \mathbf{a}, \mathbf{c}). \tag{21}$$

The following question is how to choose coordinate updating pattern. Li and Osher adopted the checkerboard pattern [55], in which the pixels are divided into black and white blocks. The pixels in the same group are not neighbors, and then the pixels in two blocks can be alternatively updated. Another greedy strategy is also popular [62], in which the selected coordinate makes the biggest contribution to the decrease of the energy function. And by the divide and conquer strategy, the corresponding coordinate can be searched with complexity $O(\log N)$ [63].

The proposed CoDenoise algorithm adopted the simple cyclic updating pattern, sequentially sweeping each pixel. If the computed solution at new selected coordinate makes a big progress than that in last iteration (evaluated by a tolerance ε_0), then it will be updated. In our implementation, we use a binary mask matrix $\mathbf{M}$ to indicate whether a pixel will be updated or not. If any four neighbor of pixel (k, l) is updated, $\mathbf{M}(k, l)$ is marked as 1, and the pixel (k, l) will be updated in the next iteration, otherwise 0. For the nondifferentiability of TV norm, the solution generated by CoDenoise may get stuck at nonstationary points, which can be easily bypassed by adding small random perturbations. The perturbations decrease along with the increasing iteration number.

To stop the CoDenoise algorithm, we check whether the relative difference between two iterations is below tolerance ε; that is,

$$\frac{\left\| \mathbf{x}^{(t)} - \mathbf{x}^{(t-1)} \right\|}{\left\| \mathbf{x}^{(t)} \right\|} \le \varepsilon. \tag{22}$$

The CoDenoise algorithm is summarized as Algorithm 1.

3.4. Convergence and Complexity. We first discuss the convergence of the CoDenoise algorithm and then analyze its computational complexity.

3.4.1. Convergence

Theorem 1. *For the optimization problem equation (12), one can obtain its optimal solution using $x^* = \mathcal{P}(\tau, b, \mathbf{a}, \mathbf{c})$; then $f(x) - f(x^*) \geq 1/2(x - x^*)^2$ holds for any x.*

Proof. First, observe that

$$(x - b)^2 - (x^* - b)^2 = (x - x^*)^2 \\ + 2(x^* - b)(x - x^*). \tag{23}$$

Suppose $c_{l_i}/a_{l_i} \leq x^* \leq c_{l_{i+1}}/a_{l_{i+1}}$; then $(x^* - b) + \tau \sum_{i=1}^{d} a_{l_i} \partial g_{l_i}(x^*) = 0$.

(1) If $c_{l_i}/a_{l_i} < x^* < c_{l_{i+1}}/a_{l_{i+1}}$, we have $x^* - b = -\tau s_i$. Therefore,

$$f(x) - f(x^*)$$
$$= \frac{1}{2}(x - x^*)^2 + (x^* - b)(x - x^*)$$
$$+ \tau \sum_{j=1}^{d} a_{l_j} \left| x - \frac{c_{l_j}}{a_{l_j}} \right| - \tau \sum_{j=1}^{d} a_{l_j} \left| x^* - \frac{c_{l_j}}{a_{l_j}} \right|$$
$$= \frac{1}{2}(x - x^*)^2 - \tau s_i (x - x^*) + \tau \sum_{j=1}^{d} a_{l_j} \left| x - \frac{c_{l_j}}{a_{l_j}} \right|$$
$$- \tau \sum_{j=1}^{i} a_{l_j} \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right) + \tau \sum_{j=i+1}^{d} a_{l_j} \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right)$$
$$\geq \frac{1}{2}(x - x^*)^2 - \tau s_i (x - x^*)$$
$$+ \tau \sum_{j=1}^{i} a_{l_j} \left(\left(x - \frac{c_{l_j}}{a_{l_j}} \right) - \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right) \right)$$
$$- \tau \sum_{j=i+1}^{d} a_{l_j} \left(\left(x - \frac{c_{l_j}}{a_{l_j}} \right) - \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right) \right)$$
$$= \frac{1}{2}(x - x^*)^2 - \tau s_i (x - x^*)$$
$$+ \tau \left(\sum_{j=1}^{i} a_{l_j} - \sum_{j=i+1}^{d} a_{l_j} \right) (x - x^*) = \frac{1}{2}(x - x^*)^2. \tag{24}$$

(2) If $x^* = c_{l_i}/a_{l_i}$, then $0 \in (x^* - b) + \tau[s_{i-1}, s_i]$.

So for some $p \in [-a_{l_i}, a_{l_i}]$, we have $x^* - b = \tau(\sum_{j=i+1}^{d} a_{l_j} - \sum_{j=1}^{i-1} a_{l_j} - p)$. Therefore,

$$f(x) - f(x^*)$$
$$= \frac{1}{2}(x - x^*)^2 + (x^* - b)(x - x^*)$$
$$+ \tau \sum_{j=1}^{d} a_{l_j} \left| x - \frac{c_{l_j}}{a_{l_j}} \right| - \tau \sum_{j=1}^{d} a_{l_j} \left| x^* - \frac{c_{l_j}}{a_{l_j}} \right|$$

$$= \frac{1}{2}(x - x^*)^2 + \tau \left(\sum_{j=i+1}^{d} a_{l_j} - \sum_{j=1}^{i-1} a_{l_j} - p \right) (x - x^*)$$
$$+ \tau \sum_{j=1}^{d} a_{l_j} \left| x - \frac{c_{l_j}}{a_{l_j}} \right| - \tau \sum_{j=1}^{i-1} a_{l_j} \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right)$$
$$+ \tau \sum_{j=i+1}^{d} a_{l_j} \left(x^* - \frac{c_{l_j}}{a_{l_j}} \right) - \tau p$$
$$\geq \frac{1}{2}(x - x^*)^2 + \tau \left(\sum_{j=i+1}^{d} a_{l_j} - \sum_{j=1}^{i-1} a_{l_j} - p \right) (x - x^*)$$
$$- \tau \sum_{j=1}^{i-1} a_{l_j} (x - x^*) + \tau \sum_{j=i+1}^{d} a_{l_j} (x - x^*)$$
$$+ \tau a_{l_i} \left| x - \frac{c_{l_i}}{a_{l_i}} \right|$$
$$= \frac{1}{2}(x - x^*)^2 - \tau p (x - x^*) + \tau a_{l_i} |x - x^*|$$
$$\geq \frac{1}{2}(x - x^*)^2. \tag{25}$$

We hence conclude that, for any x, $f(x) - f(x^*) \geq 1/2(x - x^*)^2$ always holds.

$\square$

Theorem 2. *The sequence $\{\mathbf{x}^{(t)}\}$ generated by the CoDenoise algorithm converges.*

Proof. From (21), we have $x_{k,l}^{(t+1)} = \arg\min_{x_{k,l}} F^{(t)}(x_{k,l})$, so $F^{(t)}(x_{k,l}^{(t+1)}) \leq F^{(t)}(x_{k,l}^{(t)})$. It implies that the energy over all pixels decreases, $F(\mathbf{x}^{(t+1)}) \leq F(\mathbf{x}^{(t)})$, and since it has lower bound 0, the sequence $\{F(\mathbf{x}^{(t)})\}$ converges. From Theorem 1, we have that

$$\left\| \mathbf{x}^{(t)} - \mathbf{x}^{(t+1)} \right\|_\infty \leq \left(2 \left(F\left(\mathbf{x}^{(t)}\right) - F\left(\mathbf{x}^{(t+1)}\right) \right) \right)^{1/2} \tag{26}$$

converges.

$\square$

3.4.2. Computational Complexity.

First, we present the analysis of computational complexity of the operator $\mathcal{P}$. The operation with the heaviest computational cost is to sort vector

$$\left[\frac{c_1}{a_1}, \frac{c_2}{a_2}, \ldots, \frac{c_d}{a_d} \right] \tag{27}$$

which can be done by existing sorting algorithms, for example, max-heap sort, and thus the sorting the vectors in (11) can be done with computational complexity $O(L \log L)$. Then, the optimal solution can be searched in $O(L)$ at worst. And thus, the complexity of proximal operator $\mathcal{P}$ at worst is $O(L^2 \log L)$.

Then, CoDenoise requires calling operator $\mathscr{P}$ N' times in each iteration, where N' is the number of nonzero entries of mask matrix $\mathbf{M}$, proportional to the noise level, and thus the computational complexity of CoDenoise is $O(N'L^2 \log L)$.

3.4.3. Discussions. The proposed operator (20) can be interpreted as finding the median [55]

$$x^* = \text{median}\left(\frac{c_{l_1}}{a_{l_1}}, \ldots, \frac{c_{l_d}}{a_{l_d}}, p_0, \ldots, p_d\right), \quad (28)$$

where $p_i = b - \tau s_i$. Since the sequence $\{s_i\}$ is nondescending, we have

$$p_d \leq \cdots \leq p_i \leq \cdots \leq p_0. \quad (29)$$

We discuss the equivalence of (28) and (20) using the following two cases:

(1) Suppose that $c_{l_i}/a_{l_i} < x^* < c_{l_{i+1}}/a_{l_{i+1}}$. The optimal solution is

$$x^* = b - \tau s_i = p_i. \quad (30)$$

Now we have $c_{l_i}/a_{l_i} < x^* = p_i < c_{l_{i+1}}/a_{l_{i+1}}$, so

$$\frac{c_{l_1}}{a_{l_1}} \leq \cdots \leq \frac{c_{l_i}}{a_{l_i}} < x^* = p_i < \frac{c_{l_{i+1}}}{a_{l_{i+1}}} \leq \cdots \leq \frac{c_{l_d}}{a_{l_d}}. \quad (31)$$

In the sequence $\{c_{l_i}/a_{l_i}\}$, there are i elements less than or equal to x^* and $d-i$ elements greater than or equal to x^*, and from (29), in the sequence $\{p_i\}$, there are $d-i$ elements less than or equal to p_i and i elements greater than or equal to x^*. And thus, x^* is the median (28). Specially, when $x^* < c_{l_1}/a_{l_1}$, all the d elements in the sequence $\{c_{l_i}/a_{l_i}\}$ are greater than or equal to x^*, and d elements in the sequence $\{p_i\}$ are less than or equal to x^*, and thus x^* is the median (28). Also when $x^* > c_{l_d}/a_{l_d}$, the same conclusion can be similarly drawn.

(2) Suppose that $x^* = c_{l_i}/a_{l_i}$, and it lies in $[p_i, p_{i-1}]$. Similarly, in the sequence $\{c_{l_i}/a_{l_i}\}$, there are $i-1$ elements less than or equal to x^* and $d-i$ elements greater than or equal to x^*, and from (29), in the sequence $\{p_i\}$, there are $d-i+1$ elements less than or equal to x^* and i elements greater than or equal to x^*. And thus, x^* is the median (28).

With the equivalence of proposed operator (20) and finding median value (28), the CoDenoise algorithm provides an interesting interpretation of the recursive weighted median operations on noisy image. By the recently great progress of studies on median filter or reweighted filter [58, 59], the computational efficiency of the proposed CoDenoise algorithm should be further improved.

Input: $\mathbf{y}, \mathbf{A}, L$
Output: $\mathbf{x}$.
(1) Initialize $t = 0$, $\mathbf{x}^t = y$, *precompute* $\mathscr{F}(\mathbf{A}^T\mathbf{A} + \delta\mathbf{I})$
(2) **while** not converged **do**

(3) $\mathbf{u}^{(t+1)} = F^{-1}\left(\dfrac{F\left(\mathbf{A}^T\mathbf{y} + \delta\left(\mathbf{x}^{(t)} + \mathbf{q}^{(t)}\right)\right)}{F\left(\mathbf{A}^T\mathbf{A} + \delta\mathbf{I}\right)}\right)$

(4) $\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \dfrac{\delta}{2}\left\|\mathbf{x} - \mathbf{u}^{(t+1)} + \mathbf{q}^{(t)}\right\|^2 + \lambda\text{TV}_L(\mathbf{x})$

(5) $\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)} + \mathbf{u}^{(t+1)} - \mathbf{x}^{(t+1)}$
(6)   update $\delta^{(t)}$ to $\delta^{(t+1)}$
(7) **end while**

ALGORITHM 2: ALMCD.

4. CoD for TV-Based Image Deblurring

TV_L based image deblurring problem is formulated as

$$\min_{\mathbf{x}} \quad \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda\text{TV}_L(\mathbf{x}), \quad (32)$$

where $\mathbf{A}$ is convolution matrix, which is an ill-posed problem. By combining variable splitting and ALM, we employ ADMM to solve this problem.

First, by introducing an auxiliary variable $\mathbf{u}$, the TV_L-based image deblurring problem is reformulated as

$$\min_{\mathbf{x}} \quad \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2 + \lambda\text{TV}_L(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{u} = \mathbf{x}. \quad (33)$$

Then the augmented Lagrangian function of (33) is

$$\mathscr{L} = \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2 + \lambda\text{TV}_L(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x} - \mathbf{u} + \mathbf{q}\|^2, \quad (34)$$

where δ is a positive penalty parameter and $\mathbf{q}$ is related to Lagrangian vector. Then, the two variables $\mathbf{u}$ and $\mathbf{x}$ can be updated alternatively until some convergence criterion is satisfied. Given $\mathbf{x}$, the $\mathbf{u}$-subproblem can be efficiently solved in Fourier domain,

$$\mathbf{u} = \mathscr{F}^{-1}\left(\frac{\mathscr{F}\left(\mathbf{A}^T\mathbf{y} + \delta(\mathbf{x} + \mathbf{q})\right)}{\mathscr{F}\left(\mathbf{A}^T\mathbf{A} + \delta\mathbf{I}\right)}\right), \quad (35)$$

where $\mathscr{F}$ and $\mathscr{F}^{-1}$ are Fourier and inverse Fourier transformation, respectively. Given $\mathbf{u}$, the $\mathbf{x}$-subproblem

$$\arg\min_{\mathbf{x}} \quad \frac{\delta}{2}\|\mathbf{x} - \mathbf{u} + \mathbf{q}\|^2 + \lambda\text{TV}_L(\mathbf{x}) \quad (36)$$

can be directly solved by the proposed CoDenoise algorithm. Finally, the parameters $\mathbf{q}$ and δ are updated. The overall algorithm is summarized as Algorithm 2.

5. Experimental Results

In this section, we report the experimental results on image denoising and deblurring to validate the proposed CoD-based methods. First, as to the image denoising, CoDenoise is

(a) Couple

(b) Man

(c) Hill

(d) Boat

Figure 1: Four test images.

compared with three Chambolle's works, that is, fixed points (CFP) algorithm on dual variables [43], first-order primal dual (CPD) algorithm [46], and graph cut- (CGC-) based algorithm [53]. Then, we compare ALMCD with several state-of-the-art deblurring algorithms with the denoising subproblem embedded, including accelerated IST algorithms, that is, TwIST [49] and FISTA [50], and ALM-based algorithm, that is, SALSA [51]. All the experiments ran on a 2.40 GHz Core(TM) i7-4700MQ processor. The CoDenoise algorithm is coded in C/C++, and ALMCD is coded in Matlab. We provide Matlab wrapper of CoDenoise which can be called by ALMCD. And for the parameter settings, the updating tolerances ϵ and ϵ_0 are both set as 10^{-3}. We set $L = 1$ for anisotropic TV minimization and $L = 3$ for isotropic TV minimization.

5.1. Image Denoising. As to the competing denoising algorithms, CFP and CPD are both only designed for isotropic TV minimization, and by modifying the projection step, CFP and CPD are easily applied to anisotropic TV minimization. Since only anisotropic TV model can be mapped to binary MRF, CGC is only feasible to anisotropic TV minimization. The denoising experiments were conducted on four 512×512 images, that is, Couple, Man, Hill, and Boat, shown in Figure 1, which were degraded by different Gaussian noise levels, with standard deviation (std.) as 0.05, 0.10, 0.15, and 0.20. Corresponding to each noise level, the trade-off parameter was chosen as 0.04, 0.09, 0.16, and 0.23, respectively, with best PSNR values.

For the convexity of TV minimization problem, CFD and CPD are proved to converge to global optimal solution. From the PSNR and SSIM comparison of different algorithms, shown in Tables 2 and 3, CoDenoise can also converge to the same solutions with CFD and CPD for each noise level. As to the computational efficiency shown in Table 1, for anisotropic TV denoising CoDenoise is much faster than all the competing algorithms, especially for low level noises, and for isotropic TV denoising CoDenoise has to handle more extra entries, thus being little computational expensive than CFP and comparable to CPD. Figure 2 delivers the visual denoising effect of different algorithms, and CoDenoise can achieve satisfactory denoising results.

TABLE 1: CPU time *(seconds)* comparisons of different noise levels.

Image	Method	Anisotropic TV				Isotropic TV			
		0.05	0.10	0.15	0.20	0.05	0.10	0.15	0.20
Couple	CGC	1.021	1.301	1.678	2.255	—	—	—	—
	CPD	0.701	0.505	0.513	0.519	0.633	0.574	0.578	0.603
	CFP	0.443	0.390	0.369	0.362	0.434	0.424	0.427	0.415
	CoD	0.165	0.164	0.229	0.315	0.612	0.627	0.607	0.620
Man	CGC	1.009	1.312	1.754	2.209	—	—	—	—
	CPD	0.504	0.512	0.502	0.516	0.578	0.567	0.567	0.575
	CFP	0.369	0.368	0.363	0.368	0.419	0.413	0.411	0.411
	CoD	0.156	0.155	0.223	0.300	0.612	0.603	0.602	0.659
Hill	CGC	1.023	1.290	1.757	2.146	—	—	—	—
	CPD	0.517	0.511	0.507	0.514	0.568	0.568	0.571	0.560
	CFP	0.364	0.376	0.366	0.361	0.416	0.412	0.408	0.411
	CoD	0.170	0.166	0.231	0.298	0.610	0.628	0.626	0.624
Boat	CGC	1.039	1.344	1.944	2.470	—	—	—	—
	CPD	0.499	0.545	0.513	0.523	0.576	0.586	0.564	0.560
	CFP	0.374	0.361	0.370	0.360	0.413	0.412	0.411	0.407
	CoD	0.157	0.156	0.247	0.295	0.610	0.610	0.609	0.606
Avg.	CGC	1.023	1.312	1.783	2.270	—	—	—	—
	CPD	0.555	0.518	0.509	0.518	0.589	0.573	0.570	0.575
	CFP	0.388	0.374	0.367	0.363	0.420	0.415	0.414	0.412
	CoD	0.162	0.160	0.232	0.302	0.611	0.617	0.611	0.627

TABLE 2: PSNR comparisons of different noise levels.

Image	Method	Anisotropic TV				Isotropic TV			
		0.05	0.10	0.15	0.20	0.05	0.10	0.15	0.20
Couple	CGC	30.72	27.17	25.25	24.10	—	—	—	—
	CPD	30.69	27.12	25.14	24.07	30.73	27.49	25.57	24.55
	CFP	30.66	27.08	25.11	24.10	30.71	27.43	25.49	24.47
	CoD	30.77	27.22	25.39	24.27	30.68	27.49	25.75	24.73
Man	CGC	31.10	27.84	26.16	25.10	—	—	—	—
	CPD	31.15	27.93	26.21	25.15	31.38	28.35	26.61	25.61
	CFP	31.12	27.90	26.21	25.15	31.35	28.29	26.58	25.49
	CoD	31.22	27.86	26.21	25.09	31.25	28.21	26.62	25.51
Hill	CGC	31.00	27.96	26.42	25.44	—	—	—	—
	CPD	31.07	28.13	26.58	25.72	31.19	28.42	26.87	25.99
	CFP	31.04	28.09	26.58	25.72	31.16	28.36	26.80	25.90
	CoD	31.10	27.97	26.44	25.33	31.14	28.34	26.88	25.84
Boat	CGC	30.85	27.48	25.66	24.43	—	—	—	—
	CPD	30.89	27.48	25.52	24.31	31.03	27.91	26.09	24.88
	CFP	30.85	27.44	25.5	24.35	31	27.84	25.94	24.78
	CoD	30.94	27.5	25.76	24.56	30.96	27.81	26.12	25.00
Avg.	CGC	30.92	27.61	25.87	24.77	—	—	—	—
	CPD	30.95	27.66	25.86	24.81	31.08	28.04	26.28	25.26
	CFP	30.92	27.63	25.85	24.83	31.06	27.98	26.2	25.16
	CoD	31.01	27.64	25.95	24.81	31.01	27.96	26.34	25.27

TABLE 3: SSIM comparisons of different noise levels.

Image	Method	Anisotropic TV				Isotropic TV			
		0.05	0.10	0.15	0.20	0.05	0.10	0.15	0.20
Couple	CGC	0.928	0.838	0.744	0.689	—	—	—	—
	CPD	0.930	0.837	0.746	0.689	0.931	0.852	0.769	0.713
	CFP	0.929	0.837	0.748	0.692	0.93	0.850	0.766	0.710
	CoD	0.931	0.840	0.760	0.699	0.931	0.854	0.781	0.726
Man	CGC	0.927	0.842	0.765	0.725	—	—	—	—
	CPD	0.930	0.845	0.777	0.726	0.933	0.860	0.795	0.744
	CFP	0.929	0.845	0.778	0.723	0.932	0.858	0.792	0.735
	CoD	0.930	0.838	0.771	0.725	0.932	0.856	0.795	0.738
Hill	CGC	0.920	0.827	0.742	0.705	—	—	—	—
	CPD	0.921	0.830	0.76	0.713	0.923	0.845	0.776	0.727
	CFP	0.921	0.831	0.763	0.714	0.923	0.844	0.775	0.723
	CoD	0.924	0.829	0.761	0.704	0.924	0.846	0.783	0.727
Boat	CGC	0.926	0.844	0.764	0.707	—	—	—	—
	CPD	0.931	0.846	0.768	0.708	0.933	0.859	0.789	0.731
	CFP	0.930	0.845	0.767	0.703	0.932	0.856	0.783	0.720
	CoD	0.929	0.838	0.765	0.706	0.932	0.856	0.789	0.728
Avg.	CGC	0.925	0.838	0.764	0.706	—	—	—	—
	CPD	0.928	0.839	0.763	0.709	0.930	0.854	0.782	0.729
	CFP	0.927	0.839	0.764	0.708	0.929	0.852	0.779	0.722
	CoD	0.929	0.836	0.762	0.707	0.930	0.853	0.787	0.730

TABLE 4: Deblurring results comparison for anisotropic TV model. T stands for TwIST, F stands for FISTA, S stands for SALSA, and A stands for ALMCD.

Image	CPU time (seconds)				PSNR				SSIM			
	T	F	S	A	T	F	S	A	T	F	S	A
Couple	21.66	31.22	16.48	1.53	27.97	27.86	28.4	28.73	0.898	0.894	0.910	0.914
Man	39.03	37.42	13.79	1.55	28.46	28.08	28.73	28.87	0.893	0.881	0.899	0.904
Hill	26.48	31.52	17.12	1.57	29.36	29.2	29.58	29.58	0.895	0.888	0.908	0.904
Boat	33.01	34.37	14.59	1.44	28.11	27.6	28.32	28.67	0.907	0.890	0.899	0.911
Average	30.04	33.63	15.50	1.52	28.48	28.18	28.76	28.96	0.899	0.888	0.904	0.908

5.2. Image Deblurring. The proposed ALMCD algorithm is compared with TwIST, FISTA, and SALSA, where CFP is adopted to solve the involved denoising subproblem. In the experiments, the test images were degraded with Gaussian kernel with std. 7 and Gaussian noise with zero mean value and std. 1×10^{-3}. The trade-off parameter λ is set as 5×10^{-5} for all the four algorithms.

Since blurry images usually suffer from severe blur and low level noise, the subproblem involved should be more efficiently solved by CoDenoise. Table 4 presents the deblurring results comparison for anisotropic TV deblurring, and one can see that ALMCD is significantly faster than all the competing algorithms. Even though CoDenoise is slower than CFP for isotropic TV minimization, ALMCD is instead more efficient than the competing algorithms, shown in Table 5. Particularly, SALSA adopted the same variable splitting strategy with ALMCD, generating the same subproblems, so the efficiency superiority of ALMCD over SALSA heavily confirms that CoD-based method contributes more to efficiency

improvement of ALMCD. In terms of deblurring quality, both PSNR and SSIM for anisotropic and isotropic TV models, shown in Tables 4 and 5, achieved by ALMCD are comparable to all the other competing algorithms. Figure 3 presents the visual deblurring results of Boat, from which one can see that the ALMCD can obtain visually plausible deblurring results. As a summary, CoD-based methods can provide comparable solutions compared with competing algorithms, while CoDenoise for anisotropic TV model is much more efficient than all the competing denoising algorithms, and ALMCD with CoDenoise embedded to solve denoising subproblem is much faster than state-of-the-art deblurring algorithms for both anisotropic and isotropic TV models.

6. Conclusion

In this paper, we propose a novel unified solution based on CoD method to solve TV minimization problems. With the unified formulation, both anisotropic and isotropic TV

(a) Noised image (16.48, 0.508)

(b) CPD (26.61, 0.795)

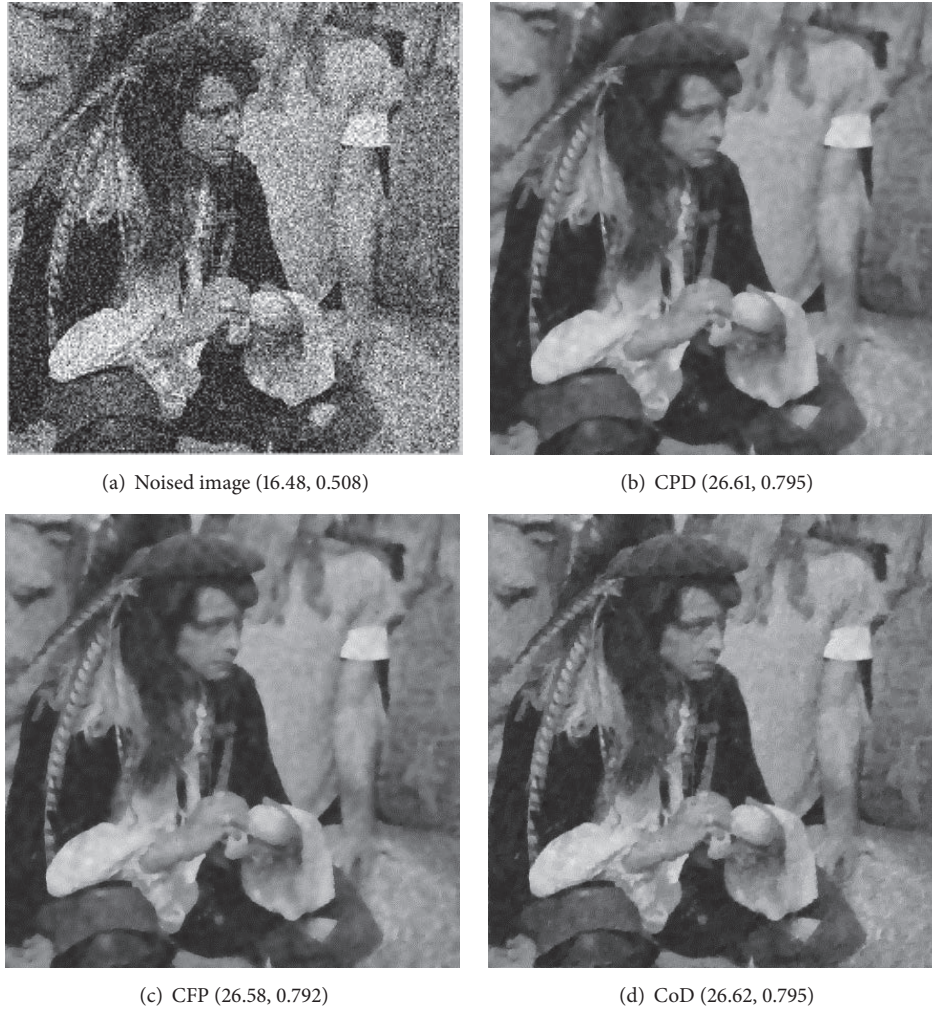(c) CFP (26.58, 0.792)

(d) CoD (26.62, 0.795)

FIGURE 2: Denoising results comparison for isotropic TV model. The Gaussian noise is with std. 0.15, and the two values in each bracket are PSNR and SSIM, respectively.

TABLE 5: Deblurring results comparison for isotropic TV model. T stands for TwIST, F stands for FISTA, S stands for SALSA, and A stands for ALMCD.

Image	CPU time (seconds)				PSNR				SSIM			
	T	F	S	A	T	F	S	A	T	F	S	A
Couple	25.57	33.11	17.77	2.64	28.16	27.95	28.38	28.47	0.905	0.895	0.908	0.914
Man	22.69	34.94	14.28	2.51	28.42	28.22	28.65	28.62	0.892	0.885	0.899	0.899
Hill	17.00	33.57	18.94	2.53	29.21	29.34	29.55	29.53	0.888	0.891	0.890	0.902
Boat	23.66	33.31	15.29	2.57	28.10	27.73	28.30	28.38	0.907	0.893	0.909	0.913
Average	22.23	33.73	16.57	2.56	28.47	28.31	28.72	28.75	0.898	0.891	0.905	0.907

minimization can be decomposed into scalar problems that can be efficiently solved by convex optimization. With simple cyclic updating pattern and random perturbations, CoDenoise can empirically converge to the optimal solution. Also when applied in image deblurring, the CoDenoise algorithm embedded in ALMCD makes significant contributions in terms of efficiency, compared with competing deblurring algorithms. In terms of deblurring quality, ALMCD can provide comparable or superior results, validating the effectiveness of CoD-based methods. Furthermore, with the great improvements in weighted median filter or parallel implementation, the CoDenoise algorithm should be much more efficient. Also, CoD could be extended to other TV variants, for example, nonlocal TV and vectorial TV.

(a) Original image

(b) TwIST (28.11, 0.907)

(c) FISTA (27.60, 0.890)

(d) Blurred image (22.15, 0.601)

(e) SALSA (28.32, 0.899)

(f) ALMCD (28.67, 0.911)

Figure 3: Deblurring results for anisotropic TV model. The two values in each bracket are PSNR and SSIM, respectively.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D. Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[2] J. Wen, J. Zhao, W. Cailing, S. Yan, and W. Wang, "Blind deblurring from single motion image based on adaptive weighted total variation algorithm," *IET Signal Processing*, vol. 10, no. 6, pp. 611–618, 2016.

[3] D. Perrone and P. Favaro, "A Clearer Picture of Total Variation Blind Deconvolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1041–1055, 2016.

[4] K. Dillon, Y. Fainman, and Y.-P. Wang, "Computational estimation of resolution in reconstruction techniques utilizing sparsity, total variation, and nonnegativity," *Journal of Electronic Imaging*, vol. 25, no. 5, Article ID 053016, 2016.

[5] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, Article ID 74585, 2007.

[6] Q. Cheng, H. Shen, L. Zhang, and P. Li, "Inpainting for remotely sensed images with a multichannel nonlocal total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 175–187, 2014.

[7] T. F. Chan, J. Shen, and H.-M. Zhou, "Total variation wavelet inpainting," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 1, pp. 107–125, 2006.

[8] V. M. Patel, R. Maleh, A. C. Gilbert, and R. Chellappa, "Gradient-based image recovery methods from incomplete Fourier measurements," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 94–105, 2012.

[9] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data," *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 288–297, 2010.

[10] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1924–1931, June 2006.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[12] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[13] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 244–252, December 2015.

[14] W. Yang, J. Liu, S. Yang, and Z. Quo, "Image super-resolution via nonlocal similarity and group structured sparse representation," in *Proceedings of the Visual Communications and Image Processing, VCIP 2015*, December 2015.

[15] R. Xiong, H. Liu, X. Zhang et al., "Image denoising via bandwise adaptive modeling and regularization exploiting nonlocal similarity," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5793–5805, 2016.

[16] S. Wang, L. Jiao, and S. Yang, "SAR Images Change Detection Based on Spatial Coding and Nonlocal Similarity Pooling," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 8, pp. 3452–3466, 2016.

[17] H. Liu, R. Xiong, S. Ma, X. Fan, and W. Gao, "Non-local extension of total variation regularization for image restoration," in *Proceedings of the 2014 IEEE International Symposium on Circuits and Systems, ISCAS 2014*, pp. 1102–1105, June 2014.

[18] D. Perrone and P. Favaro, "Total variation blind deconvolution: the devil is in the details," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2909–2916, IEEE, June 2014.

[19] D. Needell and R. Ward, "Stable image reconstruction using total variation minimization," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 1035–1058, 2013.

[20] R. I. Boţ and C. Hendrich, "Convergence analysis for a primal-dual monotone + skew splitting algorithm with applications to total variation minimization," *Journal of Mathematical Imaging and Vision*, vol. 49, no. 3, pp. 551–568, 2014.

[21] L. Ambrosio and S. Di Marino, "Equivalent definitions of BV space and of total variation on metric measure spaces," *Journal of Functional Analysis*, vol. 266, no. 7, pp. 4150–4188, 2014.

[22] H. Wu, Y. Wu, and Z. Wen, "Texture Smoothing Based on Adaptive Total Variation," *Advances in Intelligent Systems and Computing*, vol. 277, pp. 43–54, 2014.

[23] R. W. Liu, L. Shi, W. Huang, J. Xu, S. C. H. Yu, and D. Wang, "Generalized total variation-based MRI Rician denoising model with spatially adaptive regularization parameters," *Magnetic Resonance Imaging*, vol. 32, no. 6, pp. 702–720, 2014.

[24] J. Xu, D. Ren, L. Zhang, and D. Zhang, "Patch Group Based Bayesian Learning for Blind Image Denoising," in *Computer Vision – ACCV 2016 Workshops*, vol. 10116 of *Lecture Notes in Computer Science*, pp. 79–95, Springer International Publishing, Berlin, Germany, 2017.

[25] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1570–1582, 2005.

[26] T. Zeng, *Study on the variational models and dictionary learning [Ph.D. thesis]*, Université Paris-Nord-Paris XIII, France, 2007.

[27] S. Ono and I. Yamada, "Decorrelated vectorial total variation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 4090–4097, June 2014.

[28] J.-F. Aujol and S. H. Kang, "Color image decomposition and restoration," *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 916–928, 2006.

[29] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[30] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics*, vol. 31, no. 6, article 139, 2012.

[31] H. Xu, Q. Sun, N. Luo, G. Cao, and D. Xia, "Iterative nonlocal totalvariation regularization method for image restoration," *PLoS ONE*, vol. 8, no. 6, Article ID e65865, 2013.

[32] L. Antonelli, V. De Simone, and D. di Serafino, "On the application of the spectral projected gradient method in image segmentation," *Journal of Mathematical Imaging and Vision*, vol. 54, no. 1, pp. 106–116, 2016.

[33] R. De Asmundis, D. di Serafino, and G. Landi, "On the regularizing behavior of the SDA and SDC gradient methods in the

solution of linear ill-posed problems," *Journal of Computational and Applied Mathematics*, vol. 302, pp. 81–93, 2016.

[34] S. Bonettini, R. Zanella, and L. Zanni, "A scaled gradient projection method for constrained image deblurring," *Inverse Problems*, vol. 25, no. 1, Article ID 015002, 015002, 23 pages, 2009.

[35] R. Fletcher, "A limited memory steepest descent method," *Mathematical Programming*, vol. 135, no. 1-2, Ser. A, pp. 413–436, 2012.

[36] C. C. Gonzaga and R. M. Schneider, "On the steepest descent algorithm for quadratic functions," *Computational Optimization and Applications. An International Journal*, vol. 63, no. 2, pp. 523–542, 2016.

[37] R. De Asmundis, D. di Serafino, W. W. Hager, G. Toraldo, and H. Zhang, "An efficient gradient method using the Yuan steplength," *Computational Optimization and Applications. An International Journal*, vol. 59, no. 3, pp. 541–563, 2014.

[38] Y. Li and F. Santosa, "A computational algorithm for minimizing total variation in image restoration," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 987–995, 1996.

[39] T. F. Chan, H. M. Zhou, and R. H. Chan, "Continuation method for total variation denoising problems," in *Proceedings of the Advanced Signal Processing Algorithms*, pp. 314–325, San Diego, CA, USA, 1995.

[40] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 813–824, 1998.

[41] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996.

[42] D. Yi, "An iterative scheme for total variation-based image denoising," *Journal of Scientific Computing*, vol. 58, no. 3, pp. 648–671, 2014.

[43] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[44] D. Yi, B. Choi, and E.-Y. Kim, "An effective method for solving nonlinear equations and its application," *Applied Mathematics and Computation*, vol. 220, pp. 568–579, 2013.

[45] T. F. Chan, K. Chen, and J. L. Carter, "Iterative methods for solving the dual formulation arising from image restoration," *Electronic Transactions on Numerical Analysis*, vol. 26, pp. 299–311, 2007.

[46] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

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

[48] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," *UCLA CAM Report*, pp. 8–34, 2008.

[49] J. M. Bioucas-Dias and M. A. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[50] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[51] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.

[52] J. Darbon and M. Sigelle, "A Fast and Exact Algorithm for Total Variation Minimization," in *Pattern Recognition and Image Analysis*, vol. 3522 of *Lecture Notes in Computer Science*, pp. 351–359, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[53] A. Chambolle, "Total variation minimization and a class of binary MRF models," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3757, pp. 136–152, 2005.

[54] D. Goldfarb and W. Yin, "Parametric maximum flow algorithms for fast total variation minimization," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3712–3743, 2009.

[55] Y. Li and S. Osher, "A new median formula with applications to PDE based denoising," *Communications in Mathematical Sciences*, vol. 7, no. 3, pp. 741–753, 2009.

[56] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[57] T. F. Chan and K. Chen, "An optimization-based multilevel algorithm for total variation image denoising," *Multiscale Modeling and Simulation*, vol. 5, no. 2, pp. 615–645, 2006.

[58] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 2830–2837, June 2014.

[59] S. Perreault and P. Hébert, "Median filtering in constant time," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2389–2394, 2007.

[60] W. Zuo and Z. Lin, "A generalized accelerated proximal gradient approach for total-variation-based image restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2748–2759, 2011.

[61] O. V. Michailovich, "An iterative shrinkage approach to total-variation image restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1281–1299, 2011.

[62] Y. Li and S. Osher, "Coordinate descent optimization for l^1 minimization with application to compressed sensing; a greedy algorithm," *Inverse Problems and Imaging*, vol. 3, no. 3, pp. 487–503, 2009.

[63] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.