An Efficient ADMM Algorithm for Multidimensional Anisotropic Total Variation Regularization Problems

Sen Yang[1,2], Jie Wang[1,2], Wei Fan[3], Xiatian Zhang[3], Peter Wonka[2], Jieping Ye[1,2]
[1]Center for Evolutionary Medicine and Informatics, The Biodesign Institute,
[2]Computer Science and Engineering,
Arizona State University, Tempe, USA
{senyang, jie.wang.2, peter.wonka, jieping.ye}@asu.edu
[3]Huawei Noah's Ark Lab, Hong Kong, China
{david.fanwei, zhangxiatian}@huawei.com

ABSTRACT

Total variation (TV) regularization has important applications in signal processing including image denoising, image deblurring, and image reconstruction. A significant challenge in the practical use of TV regularization lies in the non-differentiable convex optimization, which is difficult to solve especially for large-scale problems. In this paper, we propose an efficient alternating augmented Lagrangian method (ADMM) to solve total variation regularization problems. The proposed algorithm is applicable for tensors, thus it can solve multidimensional total variation regularization problems. One appealing feature of the proposed algorithm is that it does not need to solve a linear system of equations, which is often the most expensive part in previous ADMM-based methods. In addition, each step of the proposed algorithm involves a set of independent and smaller problems, which can be solved in parallel. Thus, the proposed algorithm scales to large size problems. Furthermore, the global convergence of the proposed algorithm is guaranteed, and the time complexity of the proposed algorithm is $O(dN/\epsilon)$ on a d-mode tensor with N entries for achieving an ϵ-optimal solution. Extensive experimental results demonstrate the superior performance of the proposed algorithm in comparison with current state-of-the-art methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

Multidimensional total variation, ADMM, parallel computing, large scale

1. INTRODUCTION

The presence of noise in signals is unavoidable. To recover original signals, many noise reduction techniques have been developed to reduce or remove the noise. Noisy signals usually have high total variation (TV). Several total variation regularization approaches have been developed to exploit the special properties of noisy signals and they have been widely used in noise reduction in signal processing. The total variation model was first introduced by Rudin, Osher and Fatemi in [19] as a regularization approach to remove noise and handle proper edges in images. More recently, the total variation models have been applied successfully for image reconstruction, e.g. Magnetic Resonance (MR) image reconstruction [14, 17]. The wide range of applications including image restoration, image denoising and deblurring [1, 2, 14, 15, 22, 24], underscore its success in signal/image processing. The discrete penalized version of the TV-based image denoising model solves an unconstrained convex minimization problem of the following form:

$$\min_X \frac{1}{2}\|X - Y\|_F^2 + \lambda\|X\|_{TV}, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm defined as $\|X\|_F = \sqrt{\sum_{i,j} x_{i,j}^2}$, Y is the observed image, X is the desired unknown image to be recovered, and $\|\cdot\|_{TV}$ is the discrete TV norm defined below. The nonnegative regularization parameter λ provides a tradeoff between the noise sensitivity and closeness to the observed image. There are two popular choices for the discrete TV norm: ℓ_2-based *isotropic* TV defined by

$$\|X\|_{TV} = \sum_{i=1}^{m}\sum_{j=1}^{n} \|\nabla x_{i,j}\|_2, \; X \in \Re^{m\times n},$$

and the ℓ_1-based *anisotropic* TV defined by

$$\|X\|_{TV} = \sum_{i=1}^{m}\sum_{j=1}^{n} \|\nabla x_{i,j}\|_1, \; X \in \Re^{m\times n},$$

where ∇ denotes the forward finite difference operators on the vertical and horizonal directions, i.e., $\nabla x_{i,j} = (\nabla_1 x_{i,j}, \nabla_2 x_{i,j})^T$:

$$\nabla_1 x_{i,j} = \begin{cases} x_{i,j} - x_{i+1,j} & \text{if } 1 \le i < m \\ 0 & \text{if } j = n \end{cases}$$

KDD'13 August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

$$\nabla_2 x_{i,j} = \begin{cases} x_{i,j} - x_{i,j+1} & \text{if } 1 \leq j < n \\ 0 & \text{if } i = m. \end{cases}$$

Despite the simple form of the TV norm, it is a challenge to solve TV-based regularization problems efficiently. One of the key difficulties in the TV-based image denoising problem is the nonsmoothness of the TV norm. Continued research efforts have been made to build fast and scalable numerical methods in the last few years. Existing methods aim to balance the tradeoff between the convergence rate and the simplicity of each iterative step. For example, computing the exact optimal solution at each iteration leads to a better convergence rate [20]. However, this usually requires heavy computations, for instance, a large linear system of equations. Simple methods with less computation efforts at each iteration are more suitable for large-scale problems, but usually they have a slow convergence rate. To this end, we propose a fast but simple ADMM algorithm to solve TV-based problems. The key idea of the proposed method is to decompose the large problem into a set of smaller and independent problems, which can be solved efficiently and exactly. Moreover, these small problems are decoupled, thus they can be solved in parallel. Therefore, the proposed method scales to large-size problems.

Although the TV problems have been extensively studied for matrices (e.g. two-dimensional images), there is not much work on tensors, a higher-dimensional extension of matrices. Tensor data is common in real world applications, for instance, functional magnetic resonance imaging (fMRI) is a 3-mode tensor and a color video is a 4-mode tensor. Another contribution of this paper is that the proposed ADMM algorithm is designed to solve TV problems for tensors, e.g., multidimensional TV problems. The 2D TV problem can be solved efficiently by a special case of the proposed algorithm (for matrices). Our experiments show that the proposed method is more efficient than state-of-the-art approaches for solving 2D TV problems. We further demonstrate the efficiency of the proposed method for multidimensional TV problems in image reconstruction, video denoising and image deblurring.

1.1 Related Work

Due to the nonsmoothness of the TV norm, solving large-scale TV problems efficiently continues to be a challenging issue despite its simple form. In the past, considerable efforts have been devoted to develop an efficient and scalable algorithm for TV problems. The 1D total variation, also known as the fused signal approximator, has been widely used in signal noise reduction. Liu et al. [16] propose an efficient method to solve the fused signal approximator using a warm start technique. It has been shown to be very efficient in practice, though the convergence rate has not been established. Barabero and Sra [1] introduce a fast Newton-type method for 1D total variation regularization, and solve the 2D total variation problem using the Dyktra's method [7]. Wahlberg et al. [23] propose an ADMM method to solve the 1D total variation problem. A linear system of equations has to be solved at each iteration. Recently, a very fast direct, noniterative, algorithm for 1D total variation problem has been proposed in [8]. A dual-based approach to solve the 2D total variation problems is introduced in [6]. Beck and Teboulle [2] propose a fast gradient-based method by combining the dual-based approach with the acceleration technique in Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [3]. One potential drawback of the dual-based approaches is that it may not scale well. Goldstein and Osher introduce the split Bregman method to solve the 2D total variation problem, which is an application of split Bregman method solving ℓ_1 based problems. The total variation has also been widely used in Magnetic Resonance (MR) image reconstruction [14, 17]. Ma et al.[17] introduce an operator-splitting algorithm (TVCMRI) to solve the MR image reconstruction problem. By combining the composite splitting algorithm [7] and the acceleration technique in FISTA, Huang et al. [14] propose an efficient MR image reconstruction algorithm called FCSA. We show that our proposed method is much more efficient than these methods for solving 2D TV problems.

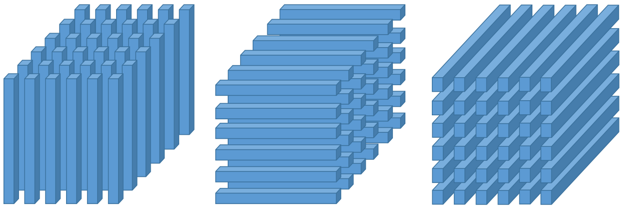


Figure 1: Fibers of a 3-mode tensor: mode-1 fibers $\mathbf{x}_{:,j_2,j_3}$, mode-2 fibers $\mathbf{x}_{j_1,:,j_3}$, and mode-3 fibers $\mathbf{x}_{j_1,j_2,:}$(left to right).

1.2 Notation

We use upper case letters for matrices, e.g. X, lower case letters for the entries, e.g. $x_{i,j}$, and bold lower case letters for vectors, e.g. $\mathbf{x}$. The inner product in the matrix space is defined as $\langle X, Y\rangle = \sum_{i,j} x_{i,j} y_{i,j}$. A d-mode tensor (or d-order tensor) is defined as $\mathcal{X} \in \Re^{I_1 \times I_2 \times \cdots \times I_d}$. Its entries are denoted as $x_{j_1,\ldots,j_d}$, where $1 \leq j_k \leq I_k,\ 1 \leq k \leq d$. For example, 1-mode tensor is a vector, and 2-mode tensor is a matrix. $\mathbf{x}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$ denotes the mode-i fiber at $\{j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_d\}$, which is the higher order analogue of matrix rows and columns (see Figure 1 for an illustration). The Frobenius norm of a tensor is defined as $\|\mathcal{X}\|_F = (\sum_{j_1,j_2,\ldots,j_d} x^2_{j_1,j_2,\ldots,j_d})^{\frac{1}{2}}$.The inner product in the tensor space is defined as $\langle \mathcal{X}, \mathcal{Y}\rangle = \sum_{j_1,j_2,\ldots,j_d} x_{j_1,j_2,\ldots,j_d} y_{j_1,j_2,\ldots,j_d}$. For simplicity of notation, we use $/\{j_i\}$ to represent the index set excluding j_i, i.e., $\{j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_d\}$. For instance, $\sum_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_d}$ can be simply written as $\sum_{/\{j_i\}}$. In addition, we use a nonnegative superscript number to denote the iteration index, e.g., $\mathcal{X}^k$ denotes the value of $\mathcal{X}$ at the k-th iteration.

1.3 Organization

We present the multidimensional total variation regularization problems and the proposed ADMM method in Section 2. One of the key steps in the proposed algorithm involves the solution of a 1D TV problem; we show how to estimate the active regularization parameter range for 1D TV problem in Section 3. We report empirical results in Section 4, and conclude this paper in Section 5.

2. THE PROPOSED ALGORITHM FOR MULTIDIMENSIONAL TV PROBLEMS

We first introduce the multidimensional total variation regularization problems in Section 2.1. In Section 2.2, we present the details of the proposed algorithm. The global convergence is established in Section 2.3. Section 2.4 presents the time complexity of the proposed algorithm.

2.1 The Multidimensional TV Problem

Denote $\mathfrak{F}_i(\mathcal{X})$ as the fused operator along the i-th mode of $\mathcal{X}$ taking the form of

$$\mathfrak{F}_i(\mathcal{X}) = \sum_{/\{j_i\}} \sum_{j_i=1}^{I_i-1} |x_{j_1,\ldots,j_i,\ldots,j_d} - x_{j_1,\ldots,(j_i+1),\ldots,j_d}|.$$

In the case of matrix, $\mathfrak{F}_i(X)$ only involves the rows or columns of X. For example, $\mathfrak{F}_1(X) = \sum_{j=1}^{n} \sum_{i=1}^{m-1} |x_{i,j} - x_{i+1,j}|, X \in \Re^{m\times n}$. It is clear that the ℓ_1-based *anisotropic* TV norm for matrices can be rewritten as $\sum_{i=1}^{2} \mathfrak{F}_i(X)$. The tensor is the generalization of the matrix concept. We generalize the TV norm for the matrix case to higher-order tensors by the following tensor TV norm:

$$\|\mathcal{X}\|_{TV} = \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X}).$$

Based on the definition above, the TV-based denoising problem for the matrix case can be generalized to tensors by solving the following optimization problem:

$$\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X}), \tag{2}$$

where $\mathcal{Y} \in \Re^{I_1\times I_2\times\ldots,\times I_d}$ is the observed data represented as a tensor, $\mathcal{X} \in \Re^{I_1\times I_2\times\ldots,\times I_d}$ is the unknown tensor to be estimated, $\sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X})$ is the tensor TV norm, and λ is a nonnegative regularization parameter. The tensor TV regularization encourages $\mathcal{X}$ to be smooth along all dimensions.

2.2 The Proposed Algorithm

We propose to solve the multidimensional TV problem (MTV) using ADMM [4]. ADMM decomposes a large global problem into a series of smaller local subproblems, and coordinates the local solutions to compute the globally optimal solution. ADMM attempts to combine the benefits of augmented Lagrangian methods and dual decomposition for constrained optimization problems [4]. The problem solved by ADMM takes the following form:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ &s.t. \quad A\mathbf{x} + B\mathbf{z} = \mathbf{c}, \end{aligned} \tag{3}$$

where x, z are unknown variables to be estimated.

ADMM reformulates the problem using a variant of the augmented Lagrangian method as follows:

$$\mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}, \mu) = f(\mathbf{x}) + g(\mathbf{z}) + \mu^T(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|A\mathbf{x} + B\mathbf{z} - \mathbf{c}\|^2$$

with μ being the augmented Lagrangian multiplier, and ρ being the nonnegative penalty parameter (or dual update length). ADMM solves the original constrained problem by iteratively minimizing $\mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}, \mu)$ over $\mathbf{x}$, $\mathbf{z}$, and updating μ according to the following update rule:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}^k, \mu^k) \\ \mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \mathfrak{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mu^k) \\ \mu^{k+1} &= \mu^k + \rho(A\mathbf{x}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c}). \end{aligned}$$

Consider the unconstrained optimization problem in (2), which can be reformulated as the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathcal{X},\mathcal{Z}_i} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i) \\ &s.t. \quad \mathcal{X} = \mathcal{Z}_i, \text{for } 1 \le i \le d, \end{aligned} \tag{4}$$

where $\mathcal{Z}_i, 1 \le i \le d$ are slack variables. The optimization problem in (4) can be solved by ADMM. The augmented Lagrangian of (4) is given by

$$\begin{aligned} \mathcal{L}(\mathcal{X}, \mathcal{Z}_i, \mathcal{U}_i) =& \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i) + \\ & \sum_{i=1}^{d} \langle \mathcal{U}_i, \mathcal{Z}_i - \mathcal{X} \rangle + \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i - \mathcal{X}\|_F^2. \end{aligned} \tag{5}$$

Applying ADMM, we carry out the following steps at each iteration:

Step 1 Update $\mathcal{X}^{k+1}$ with $\mathcal{Z}_i^k$ and $\mathcal{U}_i^k$ fixed:

$$\begin{aligned} \mathcal{X}^{k+1} =& \arg\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 - \sum_{i=1}^{d} \langle \mathcal{U}_i^k, \mathcal{X} \rangle \\ & + \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i^k - \mathcal{X}\|_F^2. \end{aligned} \tag{6}$$

The optimal solution is given by

$$\mathcal{X}^{k+1} = \frac{\mathcal{Y} + \sum_{i=1}^{d} (\mathcal{U}_i^k + \rho \mathcal{Z}_i^k)}{1 + d\rho}. \tag{7}$$

Step 2 Compute $\mathcal{Z}_i^{k+1}, i = 1, \cdots, d$ with $\mathcal{X}^{k+1}$, and $\mathcal{U}_i^k, i = 1, \cdots, d$ fixed:

$$\begin{aligned} \{\mathcal{Z}_i^{k+1}\} =& \arg\min_{\{\mathcal{Z}_i\}} \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i - \mathcal{X}^{k+1}\|_F^2 + \sum_{i=1}^{d} \langle \mathcal{U}_i^k, \mathcal{Z}_i \rangle \\ & + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i), \end{aligned} \tag{8}$$

where $\{\mathcal{Z}_i\}$ denotes the set $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_d\}$. This problem is decomposable, i.e., we can solve $\mathcal{Z}_i^{k+1}, 1 \le i \le d$ separately,

$$\mathcal{Z}_i^{k+1} = \arg\min_{\mathcal{Z}_i} \frac{\rho}{2}\|\mathcal{Z}_i - \mathcal{X}^{k+1}\|_F^2 + \langle \mathcal{U}_i^k, \mathcal{Z}_i \rangle + \lambda \mathfrak{F}_i(\mathcal{Z}_i),$$

which can be equivalently written as

$$\mathcal{Z}_i^{k+1} = \arg\min_{\mathcal{Z}_i} \frac{1}{2}\|\mathcal{Z}_i - \mathcal{T}_i\|_F^2 + \frac{\lambda}{\rho} \mathfrak{F}_i(\mathcal{Z}_i) \tag{9}$$

with $\mathcal{T}_i = -\frac{1}{\rho}\mathcal{U}_i^k + \mathcal{X}^{k+1}$. The problem in (9) is decomposable for different mode-i fibers. Denote $\mathbf{z}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$ as a mode-i fiber to be estimated, which is a vector of I_i length. For simplicity, we use $\mathbf{v}$ to represent the vector $\mathbf{z}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$. Then, (9) can be decomposed into a set of independent and much smaller problems:

$$\begin{aligned} \mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \frac{1}{2}\|\mathbf{v} - \mathbf{t}\|^2 + \frac{\lambda}{\rho} \sum_{i=1}^{I_i-1} |v_i - v_{i+1}|, \\ \forall j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_d, \end{aligned} \tag{10}$$

where $\mathbf{t}$ is the corresponding mode-i fiber of $\mathcal{T}_i$. (10) is the formulation of 1D total variation regularization problem, which can be solved exactly and very efficiently [8, 16].

The problem of computing $\mathcal{Z}_i^{k+1}, 1 \leq i \leq d$ in (8) is therefore decomposed into a set of much smaller problems of computing fibers. Each fiber problem is independent, enabling that the whole set of problems can be computed in parallel. **Step 3** Update $\mathcal{U}_i^{k+1}, i = 1, \ldots, d$:

$$\mathcal{U}_i^{k+1} = \mathcal{U}_i^k + \rho(\mathcal{Z}_i^{k+1} - \mathcal{X}^{k+1}). \tag{11}$$

A summary of the proposed method is shown in Algorithm 1 below.

Algorithm 1: The proposed ADMM algorithm for multi-dimensional total variation

```
Input: $\mathcal{Y}, \lambda, \rho$
Output: $\mathcal{X}$
Initialization: $\mathcal{Z}_i^0 = \mathcal{X}^0 \leftarrow \mathcal{Y}, \mathcal{U}_i^0 \leftarrow 0$;
do
    Compute $\mathcal{X}^{k+1}$ according to Eq. (7).
    Compute $\mathcal{Z}_i^{k+1}, i = 1, \ldots, d$ according to Eq. (9).
    Compute $\mathcal{U}_i^{k+1}, i = 1, \ldots, d$ according to Eq. (11).
Until Convergence;
return $\mathcal{X}$;
```

The algorithm stops when the primal and dual residuals [4] satisfy a certain stopping criterion. The stopping criterion can be specified by two thresholds: absolute tolerance ϵ_{abs} and relative tolerance ϵ_{rel} (see Boyd et al. [4] for more details). The penalty parameter ρ affects the primal and dual residuals, hence affects the termination of the algorithm. A large ρ tends to produce small primal residuals, but increases the dual residuals [4]. A fixed ρ (say 10) is commonly used. But there are some schemes of varying the penalty parameter to achieve better convergence. We refer interested readers to Boyd et al. [4] for more details.

Remark 1. *We can add the ℓ_1 regularization in the formulation of multidimensional TV problems for a sparse solution. The subproblem with ℓ_1 regularization is called the fused signal approximator. The optimal solution can be obtained by first solving 1D total variation problem, then applying soft-thresholding [11, 16].*

2.3 Convergence Analysis

The convergence of ADMM to solve the standard form (3) has been extensively studied [4, 10, 13]. We establish the convergence of Algorithm 1 by transforming the MTV problem in (4) into a standard form (3), and show that the transformed optimization problem satisfies the condition needed to establish the convergence.

Denote $\mathbf{x}$ as the vectorization of $\mathcal{X}$, i.e., $\mathbf{x} = vec(\mathcal{X}) \in \Re^{\prod_i I_i \times 1}$, $\mathbf{y} = vec(\mathcal{Y}) \in \Re^{\prod_i I_i \times 1}$, $\mathbf{z} = [vec(\mathcal{Z}_1)^T, \ldots, vec(\mathcal{Z}_d)^T]^T \in \Re^{d \prod_i I_i \times 1}$, $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$, and $g(\mathbf{z}) = \lambda \sum_{i=1}^d \mathfrak{F}_i(\mathcal{Z}_i)$. Then the MTV problem in (4) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}), \\ &s.t. \quad A\mathbf{x} - \mathbf{z} = \mathbf{0}, \end{aligned} \tag{12}$$

where $A = [I, \ldots, I]^T \in \Re^{d \prod_i I_i \times \prod_i I_i}$, and I is the identity matrix of size $\prod_i I_i \times \prod_i I_i$. The first and second steps of Algorithm 1 are exactly the steps of updating $\mathbf{x}$ and $\mathbf{z}$ in the standard form. Since f, g are proper, closed, and convex, and A is of column full rank, the convergence of Algorithm 1 directly follows from the results in [4, 10, 13]. Moreover, an $O(1/k)$ convergence rate of Algorithm 1 can be established following the conclusion in [13].

2.4 Time Complexity Analysis

The first step of Algorithm 1 involves computations of $\mathcal{X}_i^{k+1}, i = 1, \ldots, d$. Computing $\mathcal{X}_i^{k+1}$ needs to compute $\prod_{j \neq i} I_j$ mode-i fibers of I_i length by the 1D total variation algorithm. The complexity of solving the 1D total variation is $O(I_i)$, but $O(I_i^2)$ in the worst case [8]. However, we observe that the empirical complexity is $O(I_i)$ in our experiments (see Figure 2). Thus, the complexity of the first step is $O(d \prod_j I_j)$. The time complexity of the second and third steps are $O(\prod_j I_j)$. Hence, the complexity of each iteration is $O(d \prod_j I_j)$. The number of iterations in Algorithm 1 to obtain an ϵ-optimal solution is $O(1/\epsilon)$ [13]. Thus, the total complexity of Algorithm 1 is $O(d \prod_j I_j / \epsilon)$ for achieving an ϵ-optimal solution.

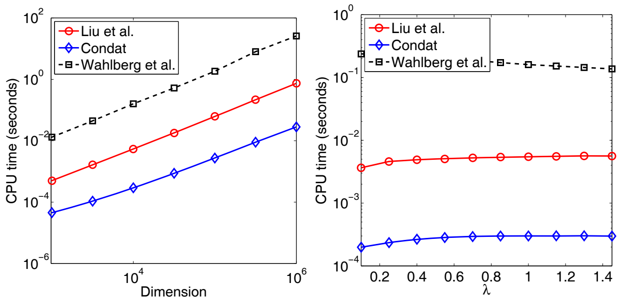


Figure 2: Computational time (seconds) of three efficient 1D total variation algorithms: Liu et al. [16], Condat [8], and Wahlberg et al. [23]. Left: dimension varies from 10^3 to 10^6 with $\lambda = 1$. Right: λ varies from 0.15 to 1.45 with dimension 10^4. The data is sampled from standard normal distribution.

3. ACTIVE REGULARIZATION RANGE FOR 1D TOTAL VARIATION

The most time-consuming part of the proposed ADMM algorithm is the first step, which involves the computation of $\mathcal{X}_i^{k+1}, 1 \leq i \leq d$. We decompose the problem of computing $\mathcal{X}_i^{k+1}, 1 \leq i \leq d$ into a set of small 1D total variation problems. Thus, the computation of the proposed method highly depends on that of 1D total variation. In this section, we show how to estimate the active regularization range for 1D total variation, which only relies on the regularization parameter and the observed vector, to directly compute the optimal solution. More specifically, we compute λ_{min} and λ_{max} based on the observed vector; if $\lambda \notin (\lambda_{min}, \lambda_{max})$, the optimal solution can be computed in a closed form, thus significantly improving the efficiency.

Consider the formulation of 1D total variation, i.e.,

$$\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{i=1}^{n-1} |x_i - x_{i+1}|,$$

which can be rewritten as

$$\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|G\mathbf{x}\|_1 \tag{13}$$

in which $\mathbf{y}, \mathbf{x} \in \Re^n$. $G \in \Re^{(n-1)\times n}$ encodes the structure of the 1D TV norm. We have

$$g_{i,j} = \begin{cases} 1 & \text{if } j = i+1 \\ -1 & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

3.1 The Dual Problem

Before we derive the dual formualtion of problem in (13) [5, 9], we first introduce some useful definitions and lemmas.

Definition 1. **(Coercivity).***[9] A function $\phi : \Re^n \to \bar{\Re}$ is said to be coercive over a set $\mathcal{S} \subset \Re^n$ if for every sequence $\{x_k\} \subset \mathcal{S}$*

$$\lim_{k\to\infty} \phi(x_k) = +\infty \text{ whenever } \|x_k\| \to +\infty.$$

For $\mathcal{S} = \Re^n$, ϕ is simply called coercive.

Denote the objective function in problem (13) as:

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|G\mathbf{x}\|_1. \tag{15}$$

It is easy to see that $f(\mathbf{x})$ is coercive. For each $\alpha \in \Re$, we define the α sublevel set of $f(\mathbf{x})$ as $\mathcal{S}_\alpha = \{\mathbf{x} : f(\mathbf{x}) \le \alpha\}$. Then we have the following lemma.

Lemma 1. *For any $\alpha \in \Re$, the sublevel set $\mathcal{S}_\alpha = \{\mathbf{x} : f(\mathbf{x}) \le \alpha\}$ is bounded.*

Proof. We prove the lemma by contradiction. Suppose there exists an α such that $\mathcal{S}_\alpha$ is unbounded. Then we can find a sequence $\{\mathbf{x}_k\} \subset \mathcal{S}_\alpha$ such that $\lim_{k\to\infty}\|\mathbf{x}_k\| = \infty$. Because $f(\mathbf{x})$ is coercive, we can conclude that $\lim_{k\to\infty} f(\mathbf{x}_k) = +\infty$. However, since $\{\mathbf{x}_k\} \subset \mathcal{S}_\alpha$, we know $f(\mathbf{x}_k) \le \alpha$ for all k, which leads to a contradiction. Therefore, the proof is complete. □

We derive the dual formulation of problem (13) via the Sion's Minimax Theorem [9, 21]. Let $\mathcal{B} = \{\mathbf{x} : \mathbf{y} - \lambda G^T\mathbf{s}, \|\mathbf{s}\|_\infty \le 1\}$ and $\mathbf{x}' = \arg\max_{\mathbf{x}\in\mathcal{B}} f(\mathbf{x})$. Because $\mathcal{B}$ is compact, $\mathbf{x}'$ must exist. Denote $\alpha' = f(\mathbf{x}')$ and $\mathcal{S}' = \mathcal{S}_{\alpha'}$.

$$\begin{aligned} \inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|G\mathbf{x}\|_1 &= \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|G\mathbf{x}\|_1 \\ &= \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sup_{\|\mathbf{s}\|_\infty \le 1} \langle \mathbf{s}, G\mathbf{x}\rangle \\ &= \inf_{\mathbf{x}\in\mathcal{S}'} \sup_{\|\mathbf{s}\|_\infty \le 1} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle. \end{aligned} \tag{16}$$

By Lemma 1, we know that $\mathcal{S}'$ is compact. Moreover, the function

$$\frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle$$

is convex and concave with respect to $\mathbf{x}$ and $\mathbf{s}$ respectively. Thus, by the Sion's Minimax Theorem [21], we have

$$\begin{aligned} &\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|G\mathbf{x}\|_1 \\ =& \inf_{\mathbf{x}\in\mathcal{S}'} \sup_{\|\mathbf{s}\|_\infty \le 1} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle \\ =& \sup_{\|\mathbf{s}\|_\infty \le 1} \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle. \end{aligned} \tag{17}$$

We can see that

$$\mathbf{x}^*(\mathbf{s}) = \mathbf{y} - \lambda G^T\mathbf{s} = \arg\min_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle \tag{18}$$

and thus

$$\begin{aligned} \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, G\mathbf{x}\rangle &= -\frac{\lambda^2}{2}\|G^T\mathbf{s}\|^2 + \lambda\langle \mathbf{s}, G\mathbf{y}\rangle \\ &= \frac{1}{2}\|\mathbf{y}\|^2 - \frac{\lambda^2}{2}\|\frac{\mathbf{y}}{\lambda} - G^T\mathbf{s}\|^2. \end{aligned}$$

Therefore the primal problem (16) is transformed to its dual problem:

$$\sup_{\|\mathbf{s}\|_\infty \le 1} \frac{1}{2}\|\mathbf{y}\|^2 - \frac{\lambda^2}{2}\|\frac{\mathbf{y}}{\lambda} - G^T\mathbf{s}\|^2, \tag{19}$$

which is equivalent to

$$\begin{aligned} &\min_{\mathbf{s}} \|\frac{\mathbf{y}}{\lambda} - G^T\mathbf{s}\|^2 \\ &\text{s.t. } \|\mathbf{s}\|_\infty \le 1. \end{aligned} \tag{20}$$

3.2 Computing the Maximal Value for λ

Recall the definition of G in (14), it follows that G^T has full column rank. Denote $e = (1, \cdots, 1)^T \in \Re^n$, and the subspace spanned by the rows of G and e as V_{G^T} and V_e. Clearly $\Re^n = V_{G^T} \oplus V_e$ and $V_{G^T} \perp V_e$.

Let $P = I - \frac{ee^T}{\langle e, e\rangle}$ and $P^\perp$ denote the projection operator into V_{G^T} and V_e respectively. Therefore the equation

$$P\mathbf{y} = G^T\mathbf{s} \tag{21}$$

must have a unique solution for each $\mathbf{y}$.

Let $P\mathbf{y} = \widetilde{\mathbf{y}}$, then it follows that

$$s_i = -\sum_{j=1}^{i} \widetilde{y}_j, \quad \forall i = 1, \cdots, n-1$$

and clearly $s_{n-1} = -\sum_{j=1}^{n-1} \widetilde{y}_j = \widetilde{y}_n$ since $\langle e, \widetilde{\mathbf{y}}\rangle = 0$. Denote

$$\lambda_{max} = \|\mathbf{s}\|_\infty = \max\{|\sum_{j=1}^{i} \widetilde{y}_j| : i = 1, \cdots, n-1\}. \tag{22}$$

From the above analysis, it is easy to see that when $\lambda \ge \lambda_{max}$, there is an $\mathbf{s}^*$ such that

$$\frac{P\mathbf{y}}{\lambda} = G^T\mathbf{s}^* \quad \text{and} \quad \|\mathbf{s}^*\|_\infty \le 1.$$

According to (18), we have

$$\mathbf{x}^* = (P + P^\perp)\mathbf{y} - \lambda G^T\mathbf{s}^* = P^\perp\mathbf{y} = \frac{\langle e, \mathbf{y}\rangle}{\langle e, e\rangle} e = \frac{\langle e, \mathbf{y}\rangle}{n} e. \tag{23}$$

The maximal value for λ has been studied in [16]. However, a linear system has to be solved. From (22), it is easy to see that the maximal value can be obtained by a close form solution. Thus, our approach is more efficient.

3.3 Computing the Minimum Value for λ

We rewrite the dual problem (19) as:

$$\begin{aligned} &\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \lambda G^T\mathbf{s}\|^2 \\ &\text{s.t. } \|\mathbf{s}\|_\infty \le 1. \end{aligned} \tag{24}$$

Denote $g(\mathbf{s}) = \frac{1}{2}\|\mathbf{y} - \lambda G^T\mathbf{s}\|^2$. The gradient of $g(\mathbf{s})$ can be found as: $g'(\mathbf{s}) = -\lambda G(\mathbf{y} - \lambda G^T\mathbf{s})$.

Let $\mathcal{B}_\infty$ denote the unit ∞ norm ball. We know that $\mathbf{s}^*$ is the unique optimal solution to the problem (24) if and only if

$$-g'(\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*),$$

where $N_{\mathcal{B}_\infty}(\mathbf{s}^*)$ is the normal cone at $\mathbf{s}^*$ with respect to $\mathcal{B}_\infty$.

Let $\mathcal{I}^+(\mathbf{s}) = \{i : s_i = 1\}$, $\mathcal{I}^-(\mathbf{s}) = \{i : s_i = -1\}$, and $\mathcal{I}^\circ(\mathbf{s}) = \{i : s_i \in (-1,1)\}$. Assume $\mathbf{d} \in N_{\mathcal{B}_\infty}(\mathbf{s})$, then $\mathbf{d}$ can be found as:

$$d_i \in \begin{cases} [0,+\infty), & \text{if } i \in \mathcal{I}^+(\mathbf{s}) \\ (-\infty,0], & \text{if } i \in \mathcal{I}^-(\mathbf{s}) \\ 0, & \text{if } i \in \mathcal{I}^\circ(\mathbf{s}) \end{cases}$$

Therefore the optimality condition can be expressed as:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \lambda G^T\mathbf{s}\|^2$$

if and only if

$$\lambda G(\mathbf{y} - \lambda G^T\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*).$$

Because $\lambda > 0$, $\lambda G(\mathbf{y} - \lambda G^T\mathbf{s}) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*)$ is equivalent to

$$G(\mathbf{y} - \lambda G^T\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*). \tag{25}$$

According to (18), we have

$$\begin{aligned} x_1^* &= y_1 + \lambda s_1^* \\ x_i^* &= y_i - \lambda(s_{i-1}^* - s_i^*), \text{for } 1 < i < n \\ x_n^* &= y_n - \lambda s_n^*, \end{aligned} \tag{26}$$

and

$$G(\mathbf{y} - \lambda G^T\mathbf{s}^*) = \begin{pmatrix} x_2^* - x_1^* \\ \vdots \\ x_n^* - x_{n-1}^* \end{pmatrix}. \tag{27}$$

By (25) and (27), we have the following observations:

B1. If $x_{i+1}^* > x_i^*$, $s_i^* = 1$;

B2. If $x_{i+1}^* = x_i^*$, $s_i^* \in [-1,1]$;

B3. If $x_{i+1}^* < x_i^*$, $s_i^* = -1$.

Notice that, from (26), we can estimate a range for every x_i^*, which is not necessarily the tightest one. In fact, we have

$$x_i^* \in \begin{cases} [y_i - \lambda, y_i + \lambda], & \text{if } i \in \{1,n\} \\ [y_i - 2\lambda, y_i + 2\lambda], & \text{otherwise.} \end{cases} \tag{28}$$

Define

$$\lambda_{min} = \min\left\{\frac{|y_{i+1} - y_i|}{3}, i \in \{1, n-1\}; \right. \tag{29}$$
$$\left. \frac{|y_{i+1} - y_i|}{4}, i \in \{2, \ldots, n-2\}\right\}.$$

It follows that when $\lambda < \lambda_{min}$, the solution to (26) is fixed and can be found as:

$$s_i^* = \text{sign}(y_{i+1} - y_i), \quad i = 1, \ldots, n-1. \tag{30}$$

Then x_i^* can be computed accordingly by (26).

4. RESULTS

In this section, we evaluate the efficiency of the proposed algorithm on synthetic and real-word data, and show several applications of the proposed algorithm.

4.1 Efficiency Comparison

We examine the efficiency of the proposed algorithm using synthetic datasets on 2D and 3D cases. For the 2D case, the competitors include

- SplitBregman written in C[1] [12];
- ADAL written in C faithfully based on the paper [18];
- The dual method in Matlab[2] [2];
- Dykstra written in C [7];

For the 3D case, only the Dykstra's method and the proposed method (MTV) are compared, since the other algorithms are designed specifically for the 2D case.

The experiments are performed on a PC with quad-core Intel 2.67GHz CPU and 9GB memory. The code of MTV is written in C. Since the proposed method and the Dykstra's method can be implemented in parallel, we also compare their parallel versions implemented with OpenMP.

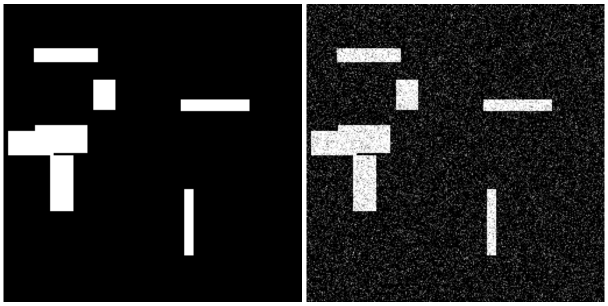


Figure 3: Left: clean image; right: noisy image;

4.1.1 2D case

We generate synthetic images $Y \in \Re^{N \times N}$ of different N. The value of each pixel is 1 or 0. A Gaussian noise $\epsilon = \mathcal{N}(0, 0.2^2)$ is added to each image as $\tilde{Y} = Y + \epsilon$. A synthetic example image is shown in Figure 3. The comparisons are based on the computation time. For a given λ, we first run MTV until a certain precision level specified by ϵ_{abs} and ϵ_{rel} is reached, and then run the others until they achieve an objective function value smaller than or equal to that of MTV. Different precision levels of the solutions are evaluated such that a fair comparison can be made. In addition, we set the maximal iteration number of all methods to be 2000 in order to avoid slow convergence. The penalty parameters ρ for MTV and ADAL are fixed to 10. We vary the size of image ($N \times N$) from 50×50 to 2000×2000 with $\lambda = 0.35$, and vary the regularization parameter λ from 0.15 to 1 with a step size of 0.05 with a fixed $N = 500$. For each setting, we perform 20 trials and report the average computational time (seconds). The results are shown in Figure 4.

From Figure 4, we observe that the proposed method is much more efficient than its competitors. The non-parallel version of MTV is about 70 times faster than the dual method, and 8 times fasters than ADAL when N is 2000 and $\epsilon_{abs} = \epsilon_{rel} = 1e-3$. Although the subproblems of MTV and Dykstra are the same, Dykstra is about 12 times slower than MTV, demonstrating that MTV has faster convergence than Dykstra. Utilizing parallel computing, the parallel version of

[1] `tag7.web.rice.edu/Split_Bregman.html`

[2] `iew3.technion.ac.il/~becka/papers/tv_fista.zip`

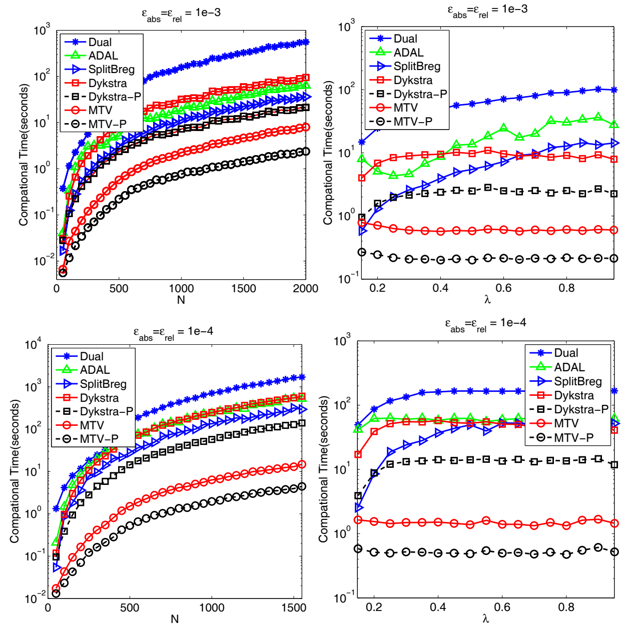


Figure 4: Comparison of SplitBregman [12], ADAL [18], Dual Method [2], Dykstra [7], and our proposed MTV algorithm in terms of computational time (in seconds and in the logarithmic scale). Dykstra-P and MTV-P are the parallel version of Dykstra and MTV. Different precision levels are used for comparison. The size of image is $N \times N$. Left column: $\lambda = 0.35$ with N varying from 50 to 2000; right column: $N = 500$ with λ varying from 0.15 to 0.95.

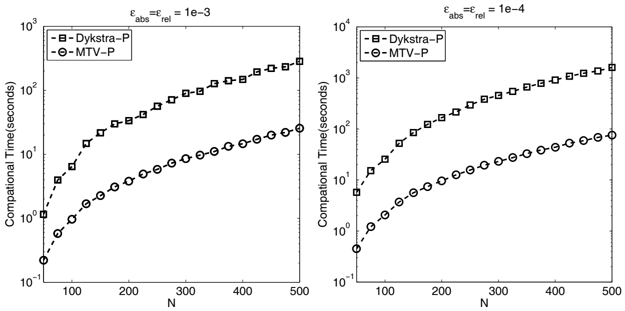


Figure 5: Comparison of Dykstra and the proposed MTV in terms of computational time (in seconds and in the logarithmic scale) in the 3D case. Different precision levels are used for comparison. The size of 3D images is $N \times N \times 50$, and N varies from 50 to 500 with $\lambda = 0.35$.

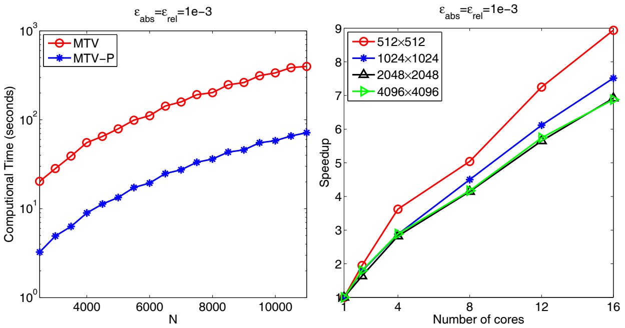


Figure 6: Scalability of the proposed method. The size of image is $N \times N$, and $\lambda = 0.35$. Left: the computational time of MTV and MTV-P with 12 processors and N varying from 2500 to 11000; right: the speedup of MTV-P with respect to the number of processors varying from 1 to 16.

MTV and Dykstra are about 3.5 times more efficient than their non-parallel version in a quad-core PC. We also observe that the Split Bregman method, dual method, and ADAL need more iterations to achieve a similar precision to that of MTV when the regularization parameter λ increases, i.e., the portion of the nonsmooth part increases. However, MTV and Dykstra are more stable when λ varies. The reason is that we directly compute the exact optimal solution of the proximal operator of the fused regularization in the subproblems of MTV and Drystra, unlike ADAL and the Split Bregman method which perform soft-thresholding.

4.1.2 3D case

The synthetic 3D images are generated in a similar manner to the 2D case. Gaussian noise $\epsilon = \mathcal{N}(0, 0.2^2)$ is added to each pixel. We set the size of 3D images to $N \times N \times 50$, and vary N from 50 to 500 with a step size of 25. The regularization parameter λ is set to 0.35. We apply the Dykstra's method and MTV on the noisy 3D images. In this experiment, we compare the computational time of Dykstra and MTV in a similar setting to the 2D case. Figure 5 shows the comparison between the Dykstra's method and MTV. From Figure 5, we can see that MTV is much more efficient than Dykstra, demonstrating the efficiency of MTV. MTV is about 20 times faster than Dykstra when $N = 500$ and $\epsilon_{abs} = \epsilon_{rel} = 1e - 4$.

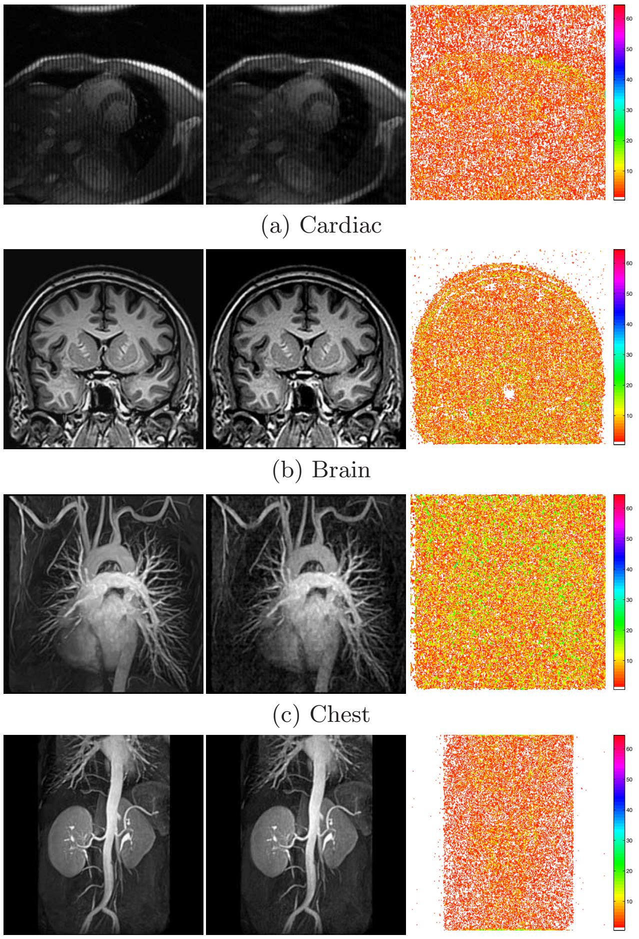
4.1.3 Scalability

We conduct experiments to evaluate the scalability of the proposed method. The experiments are performed on a Linux server with 4 quad-core Intel Xeon 2.93GHz CPUs and 65GB memory. We vary the size of images ($N \times N$) from 2500×2500 to 11000×11000 with 12 processors, and the number of processors from 1 to 16 with a fixed image size. The regularization parameter λ is set to be 0.35. For each setting, the average computational time of 10 trials is reported to demonstrate the efficiency/speedup of MTV-P (Figure 6). As shown in Figure 6, the computational time of MTV-P is less than 100 seconds when $N = 11000$, demonstrating the superiority of the proposed method. We also observe that the speedup increases almost linearly with the number of processors used. The speedup is less than the number of processors used because of the parallel overhead.

4.2 Applications

4.2.1 Image reconstruction

Due to the excellent depiction of soft tissue changes, Magnetic Resonance Imaging (MRI) has been widely used in medical diagnosis. Based on the compressive sensing theory, it is possible to reconstruct perfect signals from a limited number of samples by taking advantage of the sparse nature of the signals in a transform domain. In the case of MRI, an accurate reconstruction of MR images from undersampled K-space data is possible, reducing the cost of scanning. The



(a) Cardiac

(b) Brain

(c) Chest

(d) Artery

Figure 7: MRI reconstruction. Columns: orignal (left), FCSA-Dual and FCSA-MTV(middle), and the difference image between original image and reconstructed image (right); (a) Cardiac: SNR of two methods are 17.615; (b) Brain: SNR are 20.376; (c) Chest: SNR are 16.082; (d) Artery: the SNR are 23.769;

formulation of image reconstruction is given by

$$\hat{X} = \arg\min_{X} \frac{1}{2}\|\mathfrak{R}(X) - \mathbf{b}\|^2 + \lambda_1\|\mathfrak{W}(X)\|_1 + \lambda_2\|X\|_{TV} \quad (31)$$

where $\mathbf{b}$ is the undersampled measurements of K-space data, $\mathfrak{R}$ is partial Fourier transformation and $\mathfrak{W}$ is wavelet transform. We try to reconstruct the image $X \in \Re^{m\times n}$ from the undersampled measurements $\mathbf{b}$. A fast algorithm, FCSA, is introduced by Huang et al. [14]. One of the key steps in FCSA is the proximal operator of the 2D TV norm, which is a special case of MTV. In [14], the dual method proposed in [2] is used to solve the proximal operator. We follow the same framework as FCSA, but apply the proposed MTV to solve the proximal operator to achieve a speedup gain.

We compare two approaches: FCSA with the dual method (FSCA-Dual)[14] and FCSA with MTV (FSCA-MTV). We apply these two methods on four 2D MR images[3]: cardiac, brain, chest, and artery. We follow the same sampling strategy as in [14]. The sample ratio is set to about 25%. A Gaussian noise $\epsilon = \mathcal{N}(0, 0.01^2)$ is added to the observed measurements $\mathbf{b}$. For a fair comparison, we first run FCSA-MTV and keep track of the objective function values of MTV in each iteration, then run FCSA-Dual. In each outer iteration, the dual method stops when its objective function value is equal to or smaller than the corresponding tracked objective function value of MTV. Both FCSA-Dual and FCSA-MTV run 50 iterations. Only the computational time of the proximal operator by dual method and MTV, is recorded. The precision parameters of MTV are set to $\epsilon_{abs} = \epsilon_{rel} = 1e-3$, and the dual update step length ρ is set to 10. Since the objective function of both methods are identical, and the precision of each iteration are about the same, the solutions of both methods are expected to be the same.

Table 1: Comparison of the dual method and MTV in FCSA in terms of average computational time of 50 iterations (seconds).

Methods	Cardiac	Brain	Chest	Artery
Dual	0.6762	0.5855	0.5813	0.7588
MTV	0.0066	0.0061	0.0056	0.0078
Speedup	102.45	95.98	103.80	97.28

The reconstruction results of the MR images are shown in Figure 7. Table 1 shows the average time of dual method and MTV for 50 iterations. Since each iteration of FCSA-MTV and FCSA-Dual are the same, FCSA-MTV and FCSA-Dual have the same SNR. But we can observe from Table 1 that MTV is more efficient than dual method(about 100 times speedup), thus FCSA-MTV is more efficient than FCSA-Dual.

4.2.2 *Image deblurring*

The proposed method can be used to deblur images. The formulation of TV-based image deblurring model is given by

$$\hat{X} = \arg\min_{X} \frac{1}{2}\|\mathfrak{B}(X) - Y\|^2 + \lambda\|X\|_{TV}, \qquad (32)$$

where $Y \in \Re^{m\times n}$ is the observed blurred and noisy image, $\mathfrak{B} : \Re^{m\times n} \rightarrow \Re^{m\times n}$ is a linear transformation encoding the blurring operator, and $X \in \Re^{m\times n}$ is the image to be restored. A popular approach to solve the convex optimization problem in (32) is FISTA [2, 3]. One of the key steps is the proximal operator of TV regularization. Similar to the previous experiment, we use MTV instead of the dual method [2] to solve the proximal operator of TV regularization to achieve a speedup gain. The “lena” image of size 512 × 512 is used in this experiment. The image is rescaled to [0,1], and then blurred by an average filter of size 9 × 9. Furthermore, a Gaussian noise, $\mathcal{N}(0, 0.001^2)$, is added to the blurred image. The parameter setting of MTV is the same as the previous experiment. The regularization parameter λ is set to 0.001. The results are shown in Figure 8. The average computation time of the dual method for 100 iterations is 1.066 seconds, while that of MTV is 0.037 seconds. The proposed MTV method achieves about 29 times speedup.

4.2.3 *Video denoising*

A video is a 3-mode tensor. The proposed method in the 3D case can be used to denoise video. We expect that pixel values should be smooth along all 3 modes. In this experiment, we use a time series of 2D MR images of heart beats

[3] ranger.uta.edu/~huang/R_CSMRI.htm

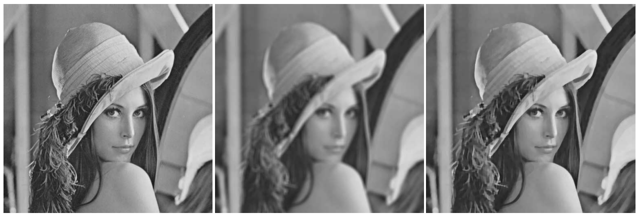


Figure 8: Image deblurring: original image(left), blurred and noisy image (middle), and deblurred image (right). The SNR of the blurred image is 11.01, and the SNR of the deblurred image is 17.23.

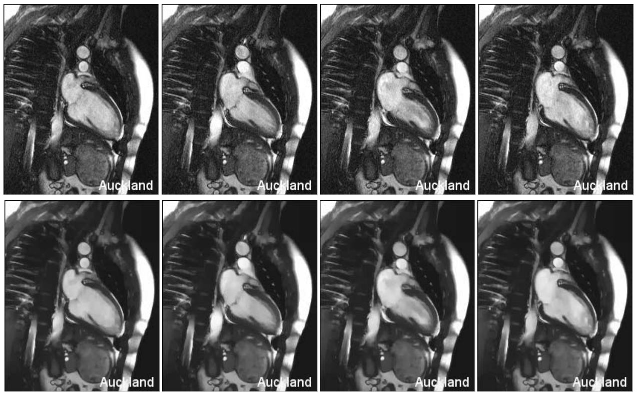


Figure 9: Sample frames of video denoising: original frames (top), and denoised frames (bottom) (best viewed on a screen).

downloaded from the website of the Cardiac Atlas[4]. The 2D MR images are in the format of avi, which includes 32 frames. We applied the proposed method and the Dystra's method to denoise all the MR images as a 3-mode tensor of size 257 × 209 × 32. The computational time of MTV is 4.482 seconds, and the computational time of the Dykstra's method is 43.751 seconds. The speedup is about 10 times. Some sample result frames are shown in Figure 9. This experiment demonstrates the effectiveness of total variation regularization in video denoising.

5. CONCLUSION

In this paper, we propose an efficient optimization of the multidimensional total variation regularization problems. We employ an efficient ADMM algorithm to solve the formulation. The key idea of our algorithm is to decompose the original problem into a set of independent and small problems, which can be solved exactly and efficiently. Furthermore, the set of independent problems can be solved in parallel. Thus, the proposed method can handle large-scale problems efficiently. We also establish the global convergence of the proposed algorithm. The experimental results demonstrate the efficiency of the proposed algorithm. The proposed algorithm opens the possibility of utilizing the power of GPU computing to further improve the efficiency of the proposed algorithm. We will explore the GPU computing in the future work. Moreover, we plan to apply the proposed algorithm to other real-world applications, such as MBB (mobile broad band) data and 3G network data, both are big data problems.

[4] atlas.scmr.org/download.html

6. REFERENCES

[1] A. Barbero and S. Sra. Fast newton-type methods for total variation regularization. In *ICML*, 2011.

[2] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] S. Boyd and L. Vandenberghe. Convex optimation, 2004.

[6] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97, 2004.

[7] P. Combettes and J. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212, 2011.

[8] L. Condat. A direct algorithm for 1d total variation denoising. 2012.

[9] A. Dhara and J. Dutta. *Optimality Conditions in Convex Optimization, A finite-dimensional view.* CRC Press, 2012.

[10] J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

[11] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[12] T. Goldstein and S. Osher. The split Bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[13] B. He and X. Yuan. On the o(1/n) convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[14] J. Huang, S. Zhang, and D. Metaxas. Efficient MR image reconstruction for compressed mr imaging. *Medical Image Analysis*, 15(5):670–679, 2011.

[15] C. Li, W. Yin, and Y. Zhang. Tval3: Tv minimization by augmented lagrangian and alternating direction algorithms, 2009.

[16] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *KDD*, pages 323–332, 2010.

[17] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty. An efficient algorithm for compressed MR imaging using total variation and wavelets. In *CVPR*, pages 1–8, 2008.

[18] Z. Qin, D. Goldfarb, and S. Ma. An alternating direction method for total variation denoising. *arXiv preprint arXiv:1108.1587*, 2011.

[19] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[20] M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS*, 2011.

[21] M. Sion. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958.

[22] C. Vogel and M. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *Image Processing, IEEE Transactions on*, 7(6):813–824, 1998.

[23] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang. An admm algorithm for a class of total variation regularized estimation problems. *IFAC*, 2012.

[24] J. Yang, Y. Zhang, and W. Yin. An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM Journal on Scientific Computing*, 31(4):2842–2865, 2009.