An Efficient ADMM Algorithm for Multidimensional Anisotropic Total Variation Regularization Problems

Sen Yang\textsuperscript{1,2}, Jie Wang\textsuperscript{1,2}, Wei Fan\textsuperscript{3}, Xiatian Zhang\textsuperscript{1}, Peter Wonka\textsuperscript{2}, Jieping Ye\textsuperscript{1,2}
\textsuperscript{1}Center for Evolutionary Medicine and Informatics, The Biodesign Institute, Arizona State University, Tempe, USA
\textsuperscript{2}Computer Science and Engineering, Arizona State University, Tempe, USA
\textsuperscript{3}Huawei Noah’s Ark Lab, Hong Kong, China
{senyang, jie.wang.2, peter.wonka, jieping.ye}@asu.edu
{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 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 $\epsilon$-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:

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

(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 $\lambda$ provides a tradeoff between the noise sensitivity and closeness to the observed image. There are two popular choices for the discrete TV norm: $\ell_2$-based isotropic TV defined by

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

and the $\ell_1$-based anisotropic TV defined by

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

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

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

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
\[ \nabla_2 x_{i,j} = \begin{cases} x_{i,j} - x_{i,j+1} & \text{if } 1 \leq j < n \\ 0 & \text{if } i = m. \end{cases} \]

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 a fast but simple ADMM method to solve the 1D TV problem in Section 3. We report empirical results in Section 4, and conclude this paper in Section 5.

### 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 \( X \in \mathbb{R}^{i_1 \times i_2 \times \cdots \times i_d} \). Its entries are denoted as \( x_{j_1,...,j_d} \), 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. \( x_{j_1,...,j_d} \) denotes the mode-i fiber at \( \{j_1,...,j_{i-1},j_{i+1},...,j_d\} \), 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 \( \|X\|_F = \left( \sum_{j_1,...,j_d} x_{j_1,...,j_d}^2 \right)^{\frac{1}{2}} \). The inner product in the tensor space is defined as \( \langle X, Y \rangle = \sum_{j_1,...,j_d} x_{j_1,...,j_d} y_{j_1,...,j_d} \). For simplicity of notation, we use \( \{j_i\} \) to represent the index set excluding \( j_i \), i.e., \( \{j_1,...,j_{i-1},j_{i+1},...,j_d\} \). For instance, \( \sum_{j_1,...,j_d} x_{j_1,...,j_d} \) can be simply written as \( \sum_{\{j_i\}} \). In addition, we use a nonnegative superscript number to denote the iteration index, e.g., \( X^{k} \) denotes the value of \( X \) 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 \(\hat{\mathfrak{g}}(X)\) as the fused operator along the \(i\)-th mode of \(X\) taking the form of

\[
\hat{\mathfrak{g}}(X) = \sum_{j=1}^{d} |x_{j1,\ldots,j_{d-1},j_{d}} - x_{j1,\ldots,j_{d-1},(j_{d}+1)}|.
\]

In the case of matrix, \(\hat{\mathfrak{g}}(X)\) only involves the rows or columns of \(X\). For example, \(\hat{\mathfrak{g}}(X) = \sum_{j=1}^{m} \sum_{i=1}^{n} |x_{i,j} - x_{i,j+1}|, X \in \mathbb{R}^{m \times n}\). It is clear that the \(\ell_1\)-based anisotropic TV norm for matrices can be rewritten as \(\sum_{i=1}^{d} \hat{\mathfrak{g}}(X)\). 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:

\[
\|X\|_{TV} = \sum_{i=1}^{d} \hat{\mathfrak{g}}(X).
\]

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:

\[
\min_{X} \frac{1}{2} \|Y - X\|_F^2 + \lambda \sum_{i=1}^{d} \hat{\mathfrak{g}}(X),
\]  

(2)

where \(Y \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}\) is the observed data represented as a tensor, \(X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}\) is the unknown tensor to be estimated, \(\sum_{i=1}^{d} \hat{\mathfrak{g}}(X)\) is the tensor TV norm, and \(\lambda\) is a nonnegative regularization parameter. The tensor TV regularization encourages \(X\) 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 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:

\[
\min_{x,z} f(x) + g(z)
\]

\[
s.t. \ Ax + Bz = c,
\]

(3)

where \(x, z\) are unknown variables to be estimated.

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

\[
\mathcal{L}_\rho(x, z, \mu) = f(x) + g(z) + \mu^T (Ax + Bz - c) + \rho \|Ax + Bz - c\|^2
\]

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

\[
x^{k+1} = \arg \min_{x} \mathcal{L}_\rho(x, z^k, \mu^k)
\]

\[
z^{k+1} = \arg \min_{z} \mathcal{L}_\rho(x^{k+1}, z, \mu^k)
\]

\[
\mu^{k+1} = \mu^k + \rho (Ax^{k+1} + Bz^{k+1} - c).
\]

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

\[
\min_{X} \frac{1}{2} \|Y - X\|_F^2 + \lambda \sum_{i=1}^{d} \hat{\mathfrak{g}}(Z_i)
\]

\[
s.t. \ X = Z_i, \text{ for } 1 \leq i \leq d,
\]

(4)

where \(Z_i, 1 \leq i \leq d\) are slack variables. The optimization problem in (4) can be solved by ADMM. The augmented Lagrangian of (4) is given by

\[
\mathcal{L}(X, Z_i, U_i) = \frac{1}{2} \|Y - X\|_F^2 + \lambda \sum_{i=1}^{d} \hat{\mathfrak{g}}(Z_i) +
\]

\[
\sum_{i=1}^{d} \langle U_i, Z_i - X \rangle + \frac{\rho}{2} \sum_{i=1}^{d} \|Z_i - X\|_F^2.
\]

(5)

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

**Step 1** Update \(X^{k+1}\) with \(Z_i^k\) and \(U_i^k\) fixed:

\[
X^{k+1} = \arg \min_{X} \frac{1}{2} \|Y - X\|_F^2 - \sum_{i=1}^{d} \langle U_i^k, X \rangle
\]

\[
+ \frac{\rho}{2} \sum_{i=1}^{d} \|Z_i^k - X\|_F^2.
\]

(6)

The optimal solution is given by

\[
X^{k+1} = \frac{Y + \sum_{i=1}^{d} (U_i^k + \rho Z_i^k)}{1 + d \rho}.
\]

(7)

**Step 2** Compute \(Z_i^{k+1}, i = 1, \ldots, d\) with \(X^{k+1}\), and \(U_i^k, i = 1, \ldots, d\) fixed:

\[
\{Z_i^{k+1}\} = \arg \min_{Z_i} \frac{\rho}{2} \|Z_i - X^{k+1}\|_F^2 + \sum_{i=1}^{d} \langle U_i^k, Z_i \rangle
\]

\[
+ \lambda \sum_{i=1}^{d} \hat{\mathfrak{g}}(Z_i),
\]

(8)

where \(\{Z_i\}\) denotes the set \(\{Z_1, \ldots, Z_d\}\). This problem is decomposable, i.e., we can solve \(Z_i^{k+1}, 1 \leq i \leq d\) separately,

\[
Z_i^{k+1} = \arg \min_{Z_i} \frac{\rho}{2} \|Z_i - X^{k+1}\|_F^2 + \langle U_i^k, Z_i \rangle + \lambda \hat{\mathfrak{g}}(Z_i),
\]

(9)

which can be equivalently written as

\[
Z_i^{k+1} = \arg \min_{Z_i} \frac{1}{2} \|Z_i - T_i\|_F^2 + \lambda \hat{\mathfrak{g}}(Z_i)
\]

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

\[
v^{k+1} = \arg \min_{v} \frac{1}{2} \|v - t\|^2 + \lambda \sum_{i=1}^{I_i-1} |v_i - v_{i+1}|,
\]

(10)

where \(t\) is the corresponding mode-i fiber of \(T_i\). (10) is the formulation of 1D total variation regularization problem, which can be solved exactly and very efficiently [8, 16].
The problem of computing \( Z_{k+1} \), \( 1 \leq i \leq d \) 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 \( U_{k+1}^i, \quad i = 1, \ldots, d: \)

\[
U_{k+1}^i = U_k^i + \rho (Z_{k+1}^i - X_{k+1}^i).
\]  

(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**: \( Y, \lambda, \rho \)

**Output**: \( X \)

**Initialization**: \( Z_0 = X, U_0 = 0 \)

**do**

- Compute \( X_{k+1}^i \) according to Eq. (7).
- Compute \( Z_{k+1}^i, \quad i = 1, \ldots, d \) according to Eq. (9).
- Compute \( U_{k+1}^i, \quad i = 1, \ldots, d \) according to Eq. (11).

**Until** Convergence

**return** \( X \).

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 \( \epsilon_{abs} \) and relative tolerance \( \epsilon_{rel} \) (see Boyd et al. [4] for more details). The penalty parameter \( \rho \) affects the primal and dual residuals, hence affects the termination of the algorithm.

**Algorithm 2**, \( \sum_i \), tends to produce small primal residuals, but increases the dual residuals [4]. A fixed \( \rho \) (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 \( \ell_1 \) regularization in the formulation of multidimensional TV problems for a sparse solution. The subproblem with \( \ell_1 \) 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 \( x \) as the vectorization of \( X \), i.e., \( x = \text{vec}(X) \in \mathbb{R}^{d \times 1} \), \( y = \text{vec}(Y) \in \mathbb{R}^{d \times 1} \), \( z = \text{vec}(Z)^T \) \( \ldots, \text{vec}(Z_d)^T \in \mathbb{R}^{d \times 1} \), \( f(x) = \frac{1}{2} \|y - x\|^2 \), and \( g(z) = \lambda \sum_{i=1}^d \delta_i(Z_i) \). Then the MTV problem in (4) can be rewritten as

\[
\min_{x, z} f(x) + g(z),
\]

s.t. \( Ax - z = 0 \),

(12)

where \( A = \left[ I, \ldots, I \right]^T \in \mathbb{R}^{d \times 1} \times \mathbb{R}^{d \times 1} \), and \( I \) is the identity matrix of size \( \prod I_i \times \prod I_i \). The first and second steps of Algorithm 1 are exactly the steps of updating \( x \) and \( z \) 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 \( X_{k+1}^i, 1 \leq i \leq d \). Computing \( X_{k+1}^i \) 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 \( O(d \prod I_j) \). The time complexity of the second and third steps are \( O(d \prod I_j) \). Hence, the complexity of each iteration is \( O(d \prod I_j) \). The number of iterations in Algorithm 1 to obtain an \( \epsilon \)-optimal solution is \( O(1/\epsilon) \) [13]. Thus, the total complexity of Algorithm 1 is \( O(d \prod I_j/\epsilon) \) for achieving an \( \epsilon \)-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 10^2 to 10^6 with \( \lambda = 1 \). Right: \( \lambda \) varies from 0.15 to 1.45 with dimension 10^4. The data is sampled from standard normal distribution.]
in which \( y, x \in \mathbb{R}^n \), \( G \in \mathbb{R}^{(n-1) \times n} \) encodes the structure of the 1D TV norm. We have
\[
g_{i,j} = \begin{cases} 
1 & \text{if } j = i + 1 \\
-1 & \text{if } j = i \\
0 & \text{otherwise.}
\end{cases}
\]

\[ (14) \]

### 3.1 The Dual Problem

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

**Definition 1. (Coercivity)** [9] A function \( \phi : \mathbb{R}^n \to \mathbb{R} \) is said to be coercive over a set \( S \subseteq \mathbb{R}^n \) if for every sequence \( \{x_k\} \subseteq S \)
\[
\lim_{k \to \infty} \phi(x_k) = +\infty \text{ whenever } \|x_k\| \to +\infty.
\]

For \( S = \mathbb{R}^n \), \( \phi \) is simply called coercive.

Denote the objective function in problem (13) as:
\[
f(x) = \frac{1}{2}\|y - x\|_2^2 + \lambda \|Gx\|_1.
\]

It is easy to see that \( f(x) \) is coercive. For each \( \alpha \in \mathbb{R} \), we define the \( \alpha \) sublevel set of \( f(x) \) as \( S_\alpha = \{x : f(x) \leq \alpha\} \). Then we have the following lemma.

**Lemma 1.** For any \( \alpha \in \mathbb{R} \), the sublevel set \( S_\alpha \) is the closure of \( \{x : f(x) \leq \alpha\} \) is bounded.

**Proof.** We prove the lemma by contradiction. Suppose there exists an \( \alpha \) such that \( S_\alpha \) is unbounded. Then we can find a sequence \( \{x_k\} \subseteq S_\alpha \) such that \( \lim_{k \to \infty} \|x_k\| = \infty \). Because \( f(x) \) is coercive, we can conclude that \( \lim_{k \to \infty} f(x_k) = +\infty \). However, since \( \{x_k\} \subseteq S_\alpha \), we know \( f(x_k) \leq \alpha \) for all \( k \), which leads to a contradiction. Therefore, the proof is complete. \( \square \)

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

\[
\inf_{x} \frac{1}{2}\|y - x\|_2^2 + \lambda \|Gx\|_1 = \inf_{x \in S'} \frac{1}{2}\|y - x\|_2^2 + \lambda \|Gx\|_1
\]
\[
= \inf_{x \in S'} \frac{1}{2}\|y - x\|_2^2 + \lambda \sup_{\|s\|_\infty \leq 1} (s, Gx) \quad (16)
\]
\[
= \inf_{x \in S'} \sup_{\|s\|_\infty \leq 1} \frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx).
\]

By Lemma 1, we know that \( S' \) is compact. Moreover, the function
\[
\frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx)
\]
is convex and concave with respect to \( x \) and \( s \) respectively. Thus, by the Sion’s Minimax Theorem [21], we have
\[
\inf_{x} \frac{1}{2}\|y - x\|_2^2 + \lambda \|Gx\|_1 \quad (17)
\]
\[
= \inf_{x \in S'} \sup_{\|s\|_\infty \leq 1} \frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx)
\]
\[
= \sup_{\|s\|_\infty \leq 1} \inf_{x \in S'} \frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx).
\]

We can see that
\[
x^*(s) = y - \lambda G^T s = \arg \min_{x \in S'} \frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx) \quad (18)
\]
and thus
\[
\inf_{x \in S'} \frac{1}{2}\|y - x\|_2^2 + \lambda (s, Gx) = \frac{\lambda^2}{2}\|G^T s\|_2^2 + \lambda (s, Gy)
\]
\[
= \frac{\lambda^2}{2}\|y\|_2^2 - \frac{\lambda^2}{2}\|G - G^T s\|_2^2.
\]

Therefore the primal problem (16) is transformed to its dual problem:
\[
\sup_{\|s\|_\infty \leq 1} \frac{1}{2}\|y\|_2^2 - \frac{\lambda^2}{2}\|G - G^T s\|_2^2, \quad (19)
\]
which is equivalent to
\[
\min_s \frac{\|y\|_2^2}{\lambda} - \frac{\lambda^2}{2}\|G - G^T s\|_2^2, \quad (20)
\]
s.t. \( \|s\|_\infty \leq 1 \).

### 3.2 Computing the Maximal Value for \( \lambda \)

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

Let \( P = I - \alpha e e^T \), and \( P^* \) denote the projection operator into \( V_{G^T} \) and \( V_e \) respectively. Therefore the equation
\[
Py = G^T s \quad (21)
\]
must have a unique solution for each \( y \).

Let \( Py = \tilde{y} \), then it follows that
\[
s_i = -\sum_{j=1}^i \tilde{y}_j, \quad \forall i = 1, \ldots, n - 1
\]
and clearly \( s_{n-1} = -\sum_{j=1}^{n-1} \tilde{y}_j = \tilde{y}_n \) since \( \langle e, \tilde{y} \rangle = 0 \). Denote
\[
\lambda_{\max} = \|s\|_\infty = \max_i \{\sum_{j=1}^i \tilde{y}_j : i = 1, \ldots, n - 1\}. \quad (22)
\]

From the above analysis, it is easy to see that when \( \lambda \geq \lambda_{\max} \), there is an \( s^* \) such that
\[
\frac{Py}{\lambda} = G^T s^* \quad \text{and} \quad \|s^*\|_\infty \leq 1.
\]

According to (18), we have
\[
x^* = (P + P^*) y - \lambda G^T s^* = P^* y = \frac{(e, y)}{(e, e)} e = \frac{(e, y)}{n} e. \quad (23)
\]

The maximal value for \( \lambda \) 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 \( \lambda \)

We rewrite the dual problem (19) as:
\[
\min_{\|s\|_\infty \leq 1} \frac{1}{2}\|y - \lambda G^T s\|_2^2 \quad (24)
\]
s.t. \( \|s\|_\infty \leq 1 \).

Denote \( g(s) = \frac{1}{2}\|y - \lambda G^T s\|_2^2 \). The gradient of \( g(s) \) can be found as: \( g'(s) = -\lambda G(y - \lambda G^T s) \).
Let $B_\infty$ denote the unit $\infty$ norm ball. We know that $s^*$ is the unique optimal solution to the problem (24) if and only if
\[-g'(s^*) \in N_{B_\infty}(s^*),\]
where $N_{B_\infty}(s^*)$ is the normal cone at $s^*$ with respect to $B_\infty$.

Let $T^+(s) = \{ i : s_i = 1 \}$, $T^-(s) = \{ i : s_i = -1 \}$, and $T^0(s) = \{ i : s_i \in (-1, 1) \}$. Assume $d \in N_{B_\infty}(s)$, then $d$ can be found as:
\[
d_i \in \begin{cases}
0, & \text{if } i \in T^+(s) \\
\lambda, & \text{if } i \in T^-(s) \\
0, & \text{if } i \in T^0(s)
\end{cases}
\]
Therefore the optimality condition can be expressed as:
\[
ts^* = \arg \min_s \frac{1}{2} \| y - \lambda G^T s \|^2
\]
if and only if
\[
\lambda G(y - \lambda G^T s^*) \in N_{B_\infty}(s^*).\]
Because $\lambda > 0$, $\lambda G(y - \lambda G^T s) \in N_{B_\infty}(s^*)$ is equivalent to
\[
G(y - \lambda G^T s^*) \in N_{B_\infty}(s^*). \tag{25}
\]
According to (18), we have
\[
x_i^* = y_i + \lambda s_i^* \\
x_i^* = y_i - \lambda (s_i^* - s_i^*), \text{ for } 1 < i < n \\
x_n^* = y_n - \lambda s_n^*,
\]
and
\[
G(y - \lambda G^T s^*) = \begin{pmatrix}
x_1^* - x_1^* \\
\vdots \\
x_n^* - x_{n-1}^*
\end{pmatrix}. \tag{27}
\]
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
\[
\lambda_{\min} = \min \left\{ \frac{|y_{i+1} - y_i|}{3}, i \in \{1, n-1\}; \frac{|y_{i+1} - y_i|}{4}, i \in \{2, \ldots, n-2\} \right\}. \tag{29}
\]
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;](image-url)

4.1.1 2D case

We generate synthetic images $Y \in \mathbb{R}^{N \times N}$ of different $N$. The value of each pixel is 1 or 0. A Gaussian noise $\epsilon = \sqrt{N(0, 0.2^2)}$ 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 $\lambda$, we first run MTV until a certain precision level specified by $\epsilon_{abs}$ and $\epsilon_{rel}$ 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 $\rho$ for MTV and ADAL are fixed to 10. We vary the size of image ($N \times N$) from 50 $\times$ 50 to 2000 $\times$ 2000 with $\lambda = 0.35$, and vary the regularization parameter $\lambda$ 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 faster 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
Split Bregman method which perform soft-thresholding.

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 $\lambda$ increases, i.e., the portion of the nonsmooth part increases. However, MTV and Dykstra are more stable when $\lambda$ 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 Dykstra, unlike ADAL and the Split Bregman method which perform soft-thresholding.

### 3D case

The synthetic 3D images are generated in a similar manner to the 2D case. Gaussian noise $\epsilon \sim N(0, 0.2^2)$ 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 $\lambda$ 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 $\epsilon_{abs} = \epsilon_{rel} = 1e^{-4}$.

### 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 $2500 \times 2500$ to $11000 \times 11000$ with 12 processors, and the number of processors from 1 to 16 with a fixed image size. The regularization parameter $\lambda$ 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.

### Applications

#### 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
undersampled measurements \((FSCA\text{-Dual})[14]\) and \(FCSA\) with \(MTV\) \((FSCA\text{-MTV})\). We solve the proximal operator to achieve a speedup gain. The same framework as \(FCSA\), but apply the proposed \(MTV\) to

in \([2]\) is used to solve the proximal operator. We follow the

is a special case of \(MTV\). In \([14]\), the dual method proposed

\(FCSA\) is the proximal operator of the 2D TV norm, which

introduced by Huang et al. \([14]\). One of the key steps in

formulation of image reconstruction is given by

\[
\hat{X} = \arg\min_X \frac{1}{2} \|\mathcal{R}(X) - b\|^2 + \lambda_1 \|\mathcal{W}(X)\|_1 + \lambda_2 \|X\|_{TV}
\]  

(31)

where \(b\) is the undersampled measurements of \(K\)-space data, \(\mathcal{R}\) is partial Fourier transformation and \(\mathcal{W}\) is wavelet transform. We try to reconstruct the image \(X \in \mathbb{R}^{m \times n}\) from the undersampled measurements \(b\). 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 \((FSCA\text{-Dual})[14]\) and \(FCSA\) with \(MTV\) \((FSCA\text{-MTV})\). We apply these two methods on four 2D MR images\(^7\): cardiac, brain, chest, and artery. We follow the same sampling strategy as in \([14]\). The sample ratio is set to about 25%. A Gauss-

\(^7\)ranger.uta.edu/~huang/R_CSMRI.htm

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>Cardiac</th>
<th>Brain</th>
<th>Chest</th>
<th>Artery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual</td>
<td>0.6762</td>
<td>0.5855</td>
<td>0.5813</td>
<td>0.7588</td>
</tr>
<tr>
<td>(MTV)</td>
<td>0.0066</td>
<td>0.0061</td>
<td>0.0056</td>
<td>0.0078</td>
</tr>
<tr>
<td>Speedup</td>
<td>102.45</td>
<td>95.98</td>
<td>103.80</td>
<td>97.28</td>
</tr>
</tbody>
</table>

Figure 7: MRI reconstruction. Columns: original (left), \(FCSA\text{-Dual}\) and \(FCSA\text{-MTV}\)(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 16.082; (c) Chest: SNR are 16.082; (d) Artery: the SNR are 23.769;

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

\[
\hat{X} = \arg\min_X \frac{1}{2} \|\mathcal{B}(X) - Y\|^2 + \lambda \|X\|_{TV},
\]  

(32)

where \(Y \in \mathbb{R}^{m \times n}\) is the observed blurred and noisy image, \(\mathcal{B} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}\) is a linear transformation encoding the blurring operator, and \(X \in \mathbb{R}^{m \times n}\) 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 \(\lambda\) 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.

4.2.2 Image deblurring

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.
downloaded from the website of the Cardiac Atlas. The 2D MR images are in the format of avi, which includes 32 frames. We applied the proposed method and the Dykstra’s method to denoise all the MR images as a 3-mode tensor of size $257 \times 209 \times 32$. 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 broadband) data and 3G network data, both are big data problems.

6. REFERENCES


---

atlas.scmr.org/download.html