Low-Complexity Total Variation-based Signal Reconstruction with Adaptive Gradient Descent for Compressive sensing

Pei-Cheng Yeh*, Chieh-Li Wang† and Yuan-Hao Huang*‡
*Department of Electrical Engineering, Natioinal Tsing Hua University, Taiwan, R.O.C.
†Graduate Institute of Electronics Engineering, Natioinal Taiwan University, Taiwan, R.O.C.
‡Institute of Communications Engineering, Natioinal Tsing Hua University, Taiwan, R.O.C.
E-mail: yhhuang@ee.nthu.edu.tw

***Abstract*—Compressive sensing (CS) has become a powerful signal processing technique, enabling signal reconstruction from a limited number of measurements by exploiting signal sparsity. This paper proposes a low-complexity, high-quality total variation (TV)-based reconstruction algorithm, termed 2D TVAL3 AdGD (Two-Dimensional Total Variation Augmented Lagrangian Alternating-Direction with Adaptive Gradient Descent). The proposed algorithm builds upon the well-known TVAL3 framework, which addresses TV regularization problems within the CS context. By adopting separable sensing operators under a two-dimensional CS model and incorporating an adaptive gradient descent method, 2D TVAL3 AdGD significantly reduces computational complexity and memory usage. This approach accelerates the reconstruction process without degrading reconstruction quality. Simulation results demonstrate that the proposed algorithm achieves superior reconstruction quality over TVAL3 and significantly outperforms other conventional CS algorithms in terms of the Structural Similarity Index (SSIM). Furthermore, it is approximately 21 times faster than TVAL3. On representative test patterns, 2D TVAL3 AdGD achieves an average SSIM of 0.9656 with a runtime of 0.016 s over 101 iterations.**

I. Introduction

Compressive sensing (CS) is a transformative signal processing paradigm that reconstructs signals from significantly fewer measurements than traditional methods by exploiting signal sparsity [1]. This enables sub-Nyquist sampling and highly efficient data acquisition, particularly for high-dimensional signals like natural images, which are sparse in various transform domains. Within this framework, image reconstruction is a critical application, recovering missing or corrupted image content from incomplete CS measurements. This fundamental task is vital across diverse fields like medical imaging [2], remote sensing [3], and astronomical imaging [4], enabling high-quality reconstructions from subsampled observations to reduce acquisition time and hardware constraints.

Early CS image reconstruction algorithms, such as Orthogonal Matching Pursuit (OMP) [5] and Approximate Message Passing (AMP) [6], assume image sparsity in a predefined transform basis. OMP is a greedy algorithm, while AMP performs iterative denoising. Their primary limitation is performance degradation when this predefined basis does not precisely capture the image's true structure. To overcome fixed-basis sparsity constraints, total variation (TV)-based methods promote sparsity in the image gradient domain, preserving edges and suppressing noise effectively [7]. Unlike basis-dependent methods, TV regularization offers more robust reconstruction quality. A notable example, the total variation augmented Lagrangian alternating-direction algorithm (TVAL3) [8], provides high-performance TV minimization. However, TVAL3's reliance on 1D vectorization and backtracking line search increases computational burden and memory usage, limiting its scalability for high-resolution images.

To address TVAL3's limitations in high-resolution image reconstruction, this paper proposes a novel two-dimensional low-complexity TV-based algorithm: the two-dimensional total variation augmented Lagrangian alternating-direction algorithm with adaptive gradient descent (2D TVAL3 AdGD). This method extends TVAL3 to operate directly under a 2D CS model with separable sampling operators. It incorporates an adaptive gradient descent (AdGD) strategy to ensure convergence without a computationally intensive backtracking line search. 2D TVAL3 AdGD significantly reduces computational complexity and memory usage while maintaining high reconstruction quality.

The remainder of this paper is organized as follows. Section II reviews the principles of total variation regularization. Section III presents the proposed 2D TVAL3 AdGD algorithm in detail. Section IV reports simulation results and performance analysis. Finally, Section V concludes the paper.

II. Total Variation Regularization

Compressive sensing aims to reconstruct a high-dimensional signal from a limited number of linear measurements. For a 2D image $\mathbf{X} \in \mathbb{R}^{n \times n}$, the compressive sensing model is typically represented as

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{x} = \mathrm{vec}(\mathbf{X}) \in \mathbb{R}^N$ is the vectorized image, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the sampling matrix with $M \ll N$, and $\mathbf{y} \in \mathbb{R}^M$ is the observation vector. Since this system is underdetermined, additional constraints are required to ensure a unique and stable

979-8-3315-7206-8/25/$31.00

solution. Regularization techniques are therefore employed to incorporate prior knowledge about the signal structure. An effective regularization strategy for image reconstruction is total variation (TV), which exploits sparsity in the image's gradient domain. This approach is particularly suitable for natural images, often piecewise smooth, consisting of homogeneous regions separated by sharp transitions, making TV regularization a natural choice. TV regularization can efficiently reduce noise while preserving edges, making it effective for image and signal processing. TV is the sum of the absolute difference between the adjacent elements throughout the signal. For a 2D signal $\mathbf{X}$, total variation (TV) is defined as

$$\mathrm{TV}(\mathbf{X}) \triangleq \sum_{i,j} \|\nabla \mathbf{X}_{i,j}\|_p, \tag{2}$$

where $p = 1$ or 2 denotes the anisotropic and isotropic TV cases, respectively, and $\nabla \mathbf{X}_{i,j}$ is the discrete gradient at pixel (i, j). The discrete gradient at pixel (i, j) can be expressed as

$$\nabla \mathbf{X}_{i,j} \triangleq \begin{bmatrix} \mathbf{X}_{i+1,j} - \mathbf{X}_{i,j} \\ \mathbf{X}_{i,j+1} - \mathbf{X}_{i,j} \end{bmatrix} = \mathbf{D}_{i,j}\mathbf{x}, \tag{3}$$

where $\mathbf{D}_{i,j} \in \mathbb{R}^{2\times n^2}$ is the discrete gradient operator at pixel (i, j). Since the ℓ_1-norm computation is generally more efficient than that of the ℓ_2-norm, this work adopts the ℓ_1-norm to evaluate the total variation throughout the remaining discussion. For the anisotropic TV case, it can be reformulated as

$$\mathrm{TV}(\mathbf{X}) = \|\mathrm{vec}(\mathbf{X}\mathbf{D}^T)\|_1 + \|\mathrm{vec}(\mathbf{D}\mathbf{X})\|_1, \tag{4}$$

where $\mathbf{D}$ is the discrete gradient matrix defined as

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 1 & & & -1 \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{5}$$

By applying TV regularization to equation (1), the overall optimization problem can be formulated as

$$\underset{\mathbf{x}}{\mathrm{argmin}}\ \mathrm{TV}(\mathrm{mtx}(\mathbf{x})) \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{6}$$

where the operator $\mathrm{mtx}(\cdot)$ denotes reshaping a vector into its 2D matrix form. To solve the constrained optimization problem in (6), TVAL3 adopts the augmented Lagrangian method (ALM), where the augmented Lagrangian function is formulated as

$$\mathcal{L}_A(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i,j} \|\mathbf{D}_{i,j}\mathbf{x}\|_1 - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) + \frac{\mu}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2. \tag{7}$$

This formulation incorporates the Lagrange multiplier $\boldsymbol{\lambda}$ and a penalty term μ. TVAL3 iteratively updates the variable $\mathbf{x}$ and the multiplier $\boldsymbol{\lambda}$ to approach the optimal solution. To overcome the non-differentiability of the ℓ_1-norm in (7), TVAL3 introduces an auxiliary variable $\mathbf{w}_{i,j}$ and redefines the optimization problem as

$$\arg\min_{\mathbf{x}} \sum_{i,j} \|\mathbf{w}_{i,j}\|_1, \quad \text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x},\ \mathbf{w}_{i,j} = \mathbf{D}_{i,j}\mathbf{x}. \tag{8}$$

The resulting constrained problem in (8) is then tackled using the augmented Lagrangian method, which introduces a penalty term to enforce the constraint $\mathbf{w}_{i,j} = \mathbf{D}_{i,j}\mathbf{x}$ while maintaining data fidelity concerning the measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$, leading to the following augmented Lagrangian function as follows:

$$\begin{aligned} \mathcal{L}_A(\mathbf{x}, \mathbf{w}, \boldsymbol{\lambda}, \mathbf{v}) = \sum_{i,j} \Big(\|\mathbf{w}_{i,j}\|_1 - \mathbf{v}_{i,j}^T(\mathbf{D}_{i,j}\mathbf{x} - \mathbf{w}_{i,j}) \\ + \frac{\beta}{2}\|\mathbf{D}_{i,j}\mathbf{x} - \mathbf{w}_{i,j}\|^2 \Big) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{y}) + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2, \end{aligned} \tag{9}$$

where $\boldsymbol{\nu}_{i,j} \in \mathbb{R}^{2\times 1}$ is another Lagrangian multiplier, and β is another penalty coefficient. **Algorithm 1** shows the pseudo algorithm of TVAL3.

To efficiently minimize the augmented Lagrangian function, the alternating direction method of multipliers (ADMM) [9] is employed. ADMM decomposes the problem into two subproblems that are solved iteratively. TVAL3 applies ADMM to decompose the problem in (9) into the following two subproblems:

$$\arg\min_{\mathbf{x}} \mathcal{L}_A(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, \boldsymbol{\lambda}^{(k)}, \boldsymbol{\nu}^{(k)}), \tag{10}$$

$$\arg\min_{\mathbf{w}} \mathcal{L}_A(\mathbf{x}^{(k+1)}, \mathbf{w}^{(k)}, \boldsymbol{\lambda}^{(k)}, \boldsymbol{\nu}^{(k)}). \tag{11}$$

To address these subproblems, the authors of TVAL3 propose the Nonmonotone Alternating Direction Algorithm (NADA), as shown in **Algorithm 2**.

Algorithm 1 TVAL3

Input: Measurement $\boldsymbol{y}$, sampling matrix $\mathbf{A}$
Output: Reconstructed result $\boldsymbol{x}^{(q)}$
Initialize: $\boldsymbol{x}^{(0)}$, $\mathbf{w}^{(0)}$; $\mu^{(0)}, \beta^{(0)} \geq 0$; $\boldsymbol{\lambda}^{(0)} = 0$, $\boldsymbol{\nu}_{i,j}^{(0)} = 0$; $\gamma \geq 1$; $q = 0$
repeat
  $(\boldsymbol{x}^{(q+1)}, \mathbf{w}^{(q+1)}) = \arg\min_{\boldsymbol{x},\mathbf{w}} \mathcal{L}_A(\boldsymbol{x}^{(q)}, \mathbf{w}^{(q)}, \boldsymbol{\lambda}^{(q)}, \boldsymbol{\nu}^{(q)})$ by NADA
  $\boldsymbol{\lambda}^{(q+1)} = \boldsymbol{\lambda}^{(q)} - \mu^{(q)}(\mathbf{A}\boldsymbol{x}^{(q+1)} - \boldsymbol{y})$
  $\boldsymbol{\nu}_{i,j}^{(q+1)} = \boldsymbol{\nu}_{i,j}^{(q)} - \beta^{(q)}(\mathbf{D}_{i,j}\boldsymbol{x}^{(q+1)} - \mathbf{w}_{i,j}^{(q+1)})$
  $\mu^{(q+1)} = \gamma\mu^{(q)}$
  $\beta^{(q+1)} = \gamma\beta^{(q)}$
  $q = q + 1$
until $\|\boldsymbol{x}^{(q)} - \boldsymbol{x}^{(q-1)}\| \leq$ tolerance

The $\mathbf{x}$-subproblem is solved via gradient descent, as $\mathcal{L}_A$ is differentiable with respect to $\mathbf{x}$ after variable splitting. The gradient is computed, and the step size α is estimated using the Barzilai–Borwein (BB) method [10]. However, since BB method may lead to divergence in certain cases, TVAL3 introduces a backtracking line search based on the nonmonotone Armijo condition [11] to ensure convergence. As shown in algorithm 2, this requires repeated evaluations of the augmented Lagrangian function at each iteration. On the other hand, the $\mathbf{w}$-subproblem involves a non-differentiable term due to the ℓ_1-norm. NADA addresses this challenge by applying the

Algorithm 2 Nonmonotone Alternating Direction Algorithm

Input: Measurement $\boldsymbol{y}$, sampling matrix $\mathbf{A}$, Lagrange multipliers $\boldsymbol{\lambda}, \boldsymbol{\nu}_{i,j}$, penalty coefficients β, μ, and initial guess $\boldsymbol{x}^{(0)}, \mathbf{w}_{i,j}^{(0)}$

Output: $\boldsymbol{x}^{(k)}$, $\mathbf{w}_{i,j}^{(k)}$

Initialize $0 \leq \delta, \rho, \eta \leq 1$; $C^{(0)} = \mathcal{L}_A(\boldsymbol{x}^{(0)}, \mathbf{w}_{i,j}^{(0)})$; $Q^{(0)} = 1$; $k = 0$

repeat

  $\boldsymbol{d}^{(k)} = \nabla_{\boldsymbol{x}} \mathcal{L}_A(\boldsymbol{x}^{(k)}, \mathbf{w}_{i,j}^{(k)})$

  $\alpha = \frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|^2}{(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)})^T (\boldsymbol{d}^{(k)} - \boldsymbol{d}^{(k-1)})}$

  repeat

    $\alpha = \rho\alpha$

  until $\mathcal{L}_A(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{d}^{(k)}) \leq C^{(k)} - \alpha\delta \|\boldsymbol{d}^{(k)}\|^2$

  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \boldsymbol{d}^{(k)}$

  $\mathbf{w}_{i,j}^{(k+1)} = \arg\min_{\mathbf{w}_i} \mathcal{L}_A(\boldsymbol{x}^{(k+1)}, \mathbf{w}_{i,j}^{(k)})$

  $Q^{(k+1)} = \eta Q^{(k)} + 1$

  $C^{(k+1)} = \frac{(\eta Q^{(k)} C^{(k)} + \mathcal{L}_A(\boldsymbol{x}^{(k+1)}, \mathbf{w}_{i,j}^{(k+1)}))}{Q^{(k+1)}}$

  $k = k + 1$

until $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| / \|\boldsymbol{x}^{(k-1)}\| \leq$ tolerance

subdifferential of the ℓ_1-norm to derive a closed-form solution. This update, known as the shrinkage formula, is expressed as follows

$$\begin{aligned} \mathbf{w}_{i,j} = &\max\left\{ \left| \mathbf{D}_{i,j}\mathbf{x} - \frac{\boldsymbol{\nu}_{i,j}}{\beta} \right| - \frac{1}{\beta},\ 0 \right\} \\ &\circ \operatorname{sgn}\left(\mathbf{D}_{i,j}\mathbf{x} - \frac{\boldsymbol{\nu}_{i,j}}{\beta} \right). \end{aligned} \tag{12}$$

This formulation, known as TVAL3, enables efficient and accurate total variation-based reconstruction within the CS framework. Despite its effectiveness, TVAL3 exhibits two main limitations. First, CS flattens a 2D image to a 1D signal, which substantially increases the size of the sampling matrix and consequently leads to higher computational complexity and memory consumption. Second, the incorporation of a backtracking line search to satisfy the nonmonotone Armijo condition necessitates repeated evaluations of the augmented Lagrangian function, thereby introducing significant computational overhead. These drawbacks motivate the development of a more efficient alternative, which will be presented in the next section.

III. Proposed 2D TV-based Reconstruction Algorithm

A. 2D Compressive Sensing

Compressive sensing (CS) models typically operate on vectorized signals, requiring the original 2D image to be flattened into a 1D vector prior to sampling. While effective in theory, this transformation leads to a large, dense sampling matrix whose dimensionality scales with the square of the image size, imposing high computational and memory demands. To address these limitations, a two-dimensional compressive sensing (2D CS) framework has been proposed, which maintains the structural integrity of the image and allows sampling to be performed separably along rows and columns. In the 2D CS paradigm, the sampling matrix $\mathbf{A} \in \mathbb{R}^{m^2 \times n^2}$ is decomposed into the Kronecker product of two smaller matrices $\mathbf{A}_1 \in \mathbb{R}^{m \times n}$ and $\mathbf{A}_2 \in \mathbb{R}^{m \times n}$:

$$\mathbf{y} = \mathbf{A}\mathbf{x} = (\mathbf{A}_2 \otimes \mathbf{A}_1)\mathbf{x} = \operatorname{vec}(\mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T). \tag{13}$$

This factorization enables the measurement model (1) to be reformulated as a compact 2D system

$$\mathbf{Y} = \mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T, \tag{14}$$

where $\mathbf{X} \in \mathbb{R}^{n \times n}$ denotes the original 2D image and $\mathbf{Y} \in \mathbb{R}^{m \times m}$ is the 2D measurement. This representation not only preserves the inherent 2D structure of the signal but also significantly reduces both computational complexity and memory usage. Specifically, this approach reduces the size of the sampling matrix from $m^2 n^2$ to $2mn$, and lowers the complexity of the matrix multiplication $\mathbf{A}\mathbf{x}$ from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$.

B. 2D Augmented Lagrangian function for TV Minimization

In the 2D compressive sensing (2D CS) framework, total variation (TV) regularization is extended by evaluating the image gradient separately along the vertical and horizontal directions. Two auxiliary variables, $\mathbf{W}_1$ and $\mathbf{W}_2$, are introduced to represent the vertical and horizontal gradients, computed as $\mathbf{D}\mathbf{X}$ and $\mathbf{X}\mathbf{D}^T$, with $\mathbf{D}$ denoting the discrete gradient matrix.

Moreover, the use of 2D CS enables the data fidelity term to be expressed as $\mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T$ instead of the conventional vectorized $\mathbf{A}\mathbf{x}$. This change significantly reduces computational complexity and memory consumption. The overall optimization problem is then addressed using an augmented Lagrangian method. The resulting 2D augmented Lagrangian function is summarized in (15).

$$\begin{aligned} &\mathcal{L}_A(\mathbf{W}_1, \mathbf{W}_2, \mathbf{X}, \mathbf{N}_1, \mathbf{N}_2, \boldsymbol{\Lambda}) = \\ &\|\mathbf{W}_1\|_1 - \langle \mathbf{N}_1, \mathbf{D}\mathbf{X} - \mathbf{W}_1 \rangle + \frac{\beta}{2} \|\mathbf{D}\mathbf{X} - \mathbf{W}_1\|^2 \\ &+ \|\mathbf{W}_2\|_1 - \langle \mathbf{N}_2, \mathbf{X}\mathbf{D}^T - \mathbf{W}_2 \rangle + \frac{\beta}{2} \|\mathbf{X}\mathbf{D}^T - \mathbf{W}_2\|^2 \\ &- \langle \boldsymbol{\Lambda}, \mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T - \mathbf{Y} \rangle + \frac{\mu}{2} \|\mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T - \mathbf{Y}\|^2 \end{aligned} \tag{15}$$

The variables $\mathbf{N}_1$, $\mathbf{N}_2$, and $\boldsymbol{\Lambda}$ serve as Lagrangian multipliers, and β, μ are the corresponding penalty coefficients.

C. 2D TVAL3 AdGD

Building upon the 2D augmented Lagrangian formulation in (15), this work proposes an efficient reconstruction algorithm, termed 2D TVAL3 AdGD, which adopts the alternating direction method of multipliers (ADMM) to decouple the optimization into two subproblems: the $\mathbf{X}$-subproblem and the $\mathbf{W}$-subproblem. To mitigate the computational overhead required in the $\mathbf{x}$-subproblem of TVAL3, this work replaces the Barzilai–Borwein (BB) method with an adaptive gradient descent strategy, referred to as adaptive gradient descent without descent (AdGD) [12][13]. AdGD dynamically adjusts the step size α_k based on local curvature and ensures convergence

without requiring line search, thereby avoiding repeated evaluations of the augmented Lagrangian function. The convergence of AdGD is theoretically guaranteed under the assumption that the objective function is locally Lipschitz smooth. In other words, its gradient is Lipschitz continuous in a neighborhood. In this case, the augmented Lagrangian function $\mathcal{L}_A(\mathbf{X})$ is differentiable and consists of linear and quadratic terms in $\mathbf{X}$. As a result, its gradient is Lipschitz continuous and the smoothness requirement of AdGD is satisfied. The AdGD algorithm can be extended to handle 2D matrix variables, such as images, by applying it directly in matrix form with Frobenius norms in (16).

Algorithm 3 Adaptive gradient descent without descent

Input: $\mathbf{x}^0 \in \mathbb{R}^d$, $\theta_0 = 0$, $\alpha_0 > 0$
$\mathbf{x}^1 = \mathbf{x}^0 - \alpha_0 \nabla f(\mathbf{x}^0)$
for $k = 1, 2, \ldots$ **do**
  $L_k = \frac{\|\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1})\|}{\|\mathbf{x}^k - \mathbf{x}^{k-1}\|}$
  $\alpha_k = \min\left\{\sqrt{1+\theta_{k-1}}\,\alpha_{k-1},\ \frac{1}{\sqrt{2}L_k}\right\}$
  $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)$
  $\theta_k = \frac{\alpha_k}{\alpha_{k-1}}$
end for

$$\begin{aligned} \alpha_k &= \min\left\{\sqrt{1+\theta_{k-1}}\,\alpha_{k-1},\ \frac{1}{\sqrt{2}L_k}\right\} \\ L_k &= \frac{\|\nabla f(\mathbf{X}^k) - \nabla f(\mathbf{X}^{k-1})\|_F}{\|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F}, \quad \theta_k = \frac{\alpha_k}{\alpha_{k-1}} \end{aligned} \tag{16}$$

Therefore, the $\mathbf{X}$-subproblem can be solved with adaptive gradient descent method directly in its matrix form.

$$\mathbf{X}^{k+1} = \mathbf{X}^k - \alpha_k \nabla f(\mathbf{X}^k) \tag{17}$$

$$\begin{aligned} \nabla f(\mathbf{X}^k) = \mathbf{d}^{(k)} &= \beta \mathbf{D}^T(\mathbf{D}\mathbf{X}^{(k)} - \mathbf{W}_1^{(k)}) - \mathbf{D}^T\mathbf{N}_1^{(k)} \\ &\quad + \beta(\mathbf{X}^{(k)}\mathbf{D}^T - \mathbf{W}_2^{(k)})\mathbf{D} - \mathbf{N}_2^{(k)}\mathbf{D} \\ &\quad + \mathbf{A}_1^T\left(\mu(\mathbf{A}_1\mathbf{X}^{(k)}\mathbf{A}_2^T - \mathbf{Y}) - \mathbf{\Lambda}\right)\mathbf{A}_2 \end{aligned} \tag{18}$$

In the proposed 2D formulation, the $\mathbf{W}$-subproblem retains its closed-form shrinkage update, as shown in (19).

$$\begin{aligned} \mathbf{W}_1 &= \max\left\{\left|\mathbf{D}\mathbf{X} - \frac{\mathbf{N}_1}{\beta}\right| - \frac{1}{\beta},\ 0\right\} \\ &\quad \circ \operatorname{sgn}\left(\mathbf{D}\mathbf{X} - \frac{\mathbf{N}_1}{\beta}\right), \\ \mathbf{W}_2 &= \max\left\{\left|\mathbf{X}\mathbf{D}^T - \frac{\mathbf{N}_2}{\beta}\right| - \frac{1}{\beta},\ 0\right\} \\ &\quad \circ \operatorname{sgn}\left(\mathbf{X}\mathbf{D}^T - \frac{\mathbf{N}_2}{\beta}\right) \end{aligned} \tag{19}$$

The overall algorithm for the proposed 2D TV-based reconstruction method is summarized in Algorithm 4. The method iteratively updates $\mathbf{X}$-subproblem and $\mathbf{W}$-subproblem through ADMM. The $\mathbf{X}$-subproblem is solved using AdGD, which guarantees convergence without the need for costly backtracking line search based on the nonmonotone Armijo condition, as required in TVAL3. The $\mathbf{W}$-subproblem retains its closed-form shrinkage solution, efficiently enforcing sparsity in the gradient domain. By leveraging the separable structure of the sensing matrix via Kronecker products and the efficient AdGD solver, 2D TVAL3 AdGD achieves substantial improvements in computational complexity and memory consumption compared to TVAL3.

Algorithm 4 2D TVAL3 AdGD

Input: Measurement $\boldsymbol{Y}$, sampling matrix $\mathbf{A_1}$, $\mathbf{A_2}$
Output: Reconstructed result $\boldsymbol{X}^{(q)}$
Initialize $\boldsymbol{X}^{(0)}$, $\mathbf{W}^{(0)}$; $\mu^{(0)}$, $\beta^{(0)} \geq 0$; $\boldsymbol{N}_i = 0$; $\gamma \geq 1$; $q, k = 0$
repeat
  repeat
    $\boldsymbol{d}^{(k)} = \nabla_{\boldsymbol{X}} \mathcal{L}_A(\boldsymbol{X}^{(k)}, \mathbf{W}^{(k)})$
    Set α_j through AdGD
    $\boldsymbol{X}^{(k+1)} = \boldsymbol{X}^{(k)} - \alpha \boldsymbol{d}^{(k)}$
    $\mathbf{W}_i^{(k+1)} = \arg\min_{\mathbf{W}_i} \mathcal{L}_A(\boldsymbol{X}^{(k+1)}, \mathbf{W}_i^{(k)})$
    $k = k + 1$
  until $\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k-1)}\| \leq$ tolerance
  $\boldsymbol{X}^{(q+1)} = \boldsymbol{X}^{(k)}$, $\boldsymbol{W}^{(q+1)} = \boldsymbol{W}^{(k)}$
  $\boldsymbol{N}_i^{(q+1)} = \boldsymbol{N}_i^{(q)} - \beta^{(q)}(\mathbf{D}\boldsymbol{X}^{(q+1)} - \mathbf{W}_i^{(q+1)})$
  $\mu^{(q+1)} = \gamma \mu^{(q)}$
  $\beta^{(q+1)} = \gamma \beta^{(q)}$
  $q = q + 1$
until $\|\boldsymbol{X}^{(q)} - \boldsymbol{X}^{(q-1)}\| \leq$ tolerance

IV. Simulation Results

Three test patterns, the original images in Fig. 1, were used to simulate the performances of the proposed algorithm. The image resolution is 64×64 pixels, and the compression ratio is 40%, i.e., the number of measurements is $M = 0.4 \times N$, where N is the total number of pixels. In the 2D CS setting, this corresponds to a measurement size of 40×40 pixels. For TV-based algorithms, TVAL3 samples the image after vectorization using a large DCT matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. In contrast, the proposed 2D TVAL3 AdGD follows a 2D CS framework, applying separable DCT matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{m \times n}$ to directly sample the image in matrix form. The 2D CS measurement process is given by $\mathbf{Y} = \mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T$. For conventional CS algorithms, a random Bernoulli matrix is defined as the sampling matrix, with the DCT matrix as the sparsifying basis. All simulations were conducted on a desktop computer with a 2.5 GHz Intel Core i5-12400F CPU, 32 GB of DDR4 memory, and MATLAB R2022a (64-bit). Figure 1 shows the original and reconstructed images using different reconstruction algorithms. Each column shows the original image, followed by the results obtained using 2D TVAL3 AdGD, TVAL3, AMP, and OMP, respectively. It can be observed that 2D TVAL3 AdGD yields visually sharper and less noisy reconstruction results compared to other methods.

Figure 2 shows the variation of the structural similarity index (SSIM) over iterations. The proposed algorithm

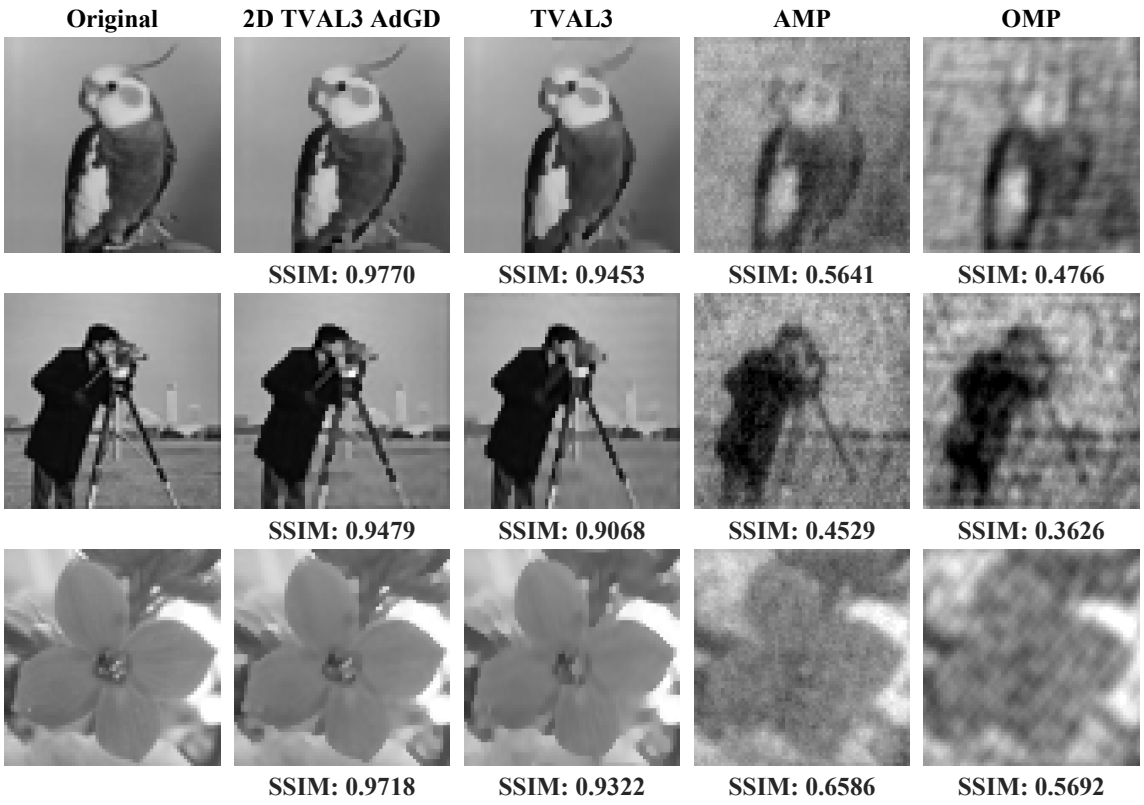

Fig. 1: Image reconstruction results of different algorithms.

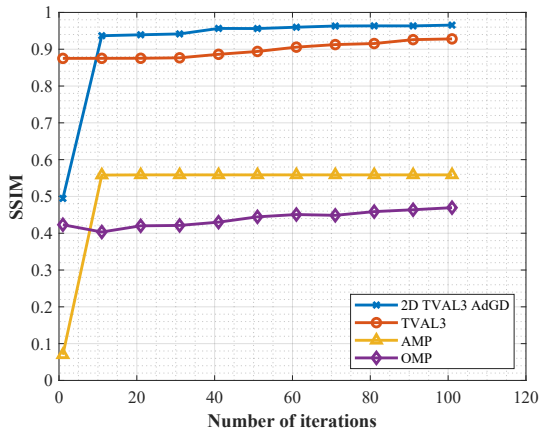

Fig. 2: Comparison of SSIM vs iteration.

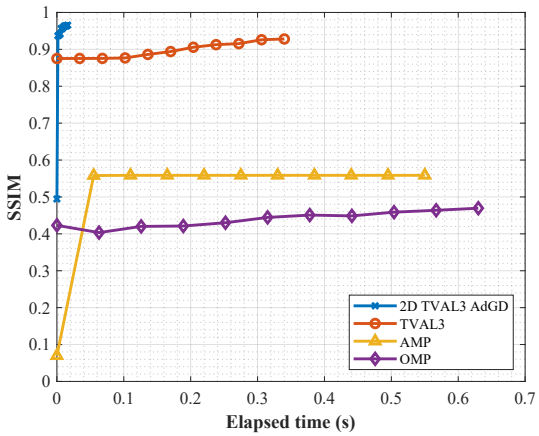

Fig. 3: Comparison of SSIM vs elapsed time.

converges faster and achieves slightly higher SSIM than the original TVAL3, demonstrating the effectiveness of the 2D CS framework in preserving spatial structures during sampling and reconstruction. In contrast, conventional CS algorithms exhibit significantly lower reconstruction quality than TV-based methods. Figure 3 presents SSIM performance over the elapsed time. Benefiting from separable 2D operators and an adaptive gradient descent strategy that avoids the costly non-monotone line search, the proposed algorithm converges much faster in the elapsed time, reaching high SSIM values almost instantaneously. TVAL3, while effective, requires a longer runtime to attain similar reconstruction quality. These results demonstrate the efficiency of the proposed algorithm in terms of convergence speed. In summary, the average SSIM and elapsed time of 2D TVAL3 AdGD are 0.9656 and 0.016 s, respectively, with 101 iterations averaged over the test patterns.

Table I summarizes the total number of multiply-accumulate (MAC) operations required per iteration for TVAL3 and 2D TVAL3 AdGD algorithm. We can observe that TVAL3 incurs a significantly higher computational load due to the use of a large-scale sensing matrix $\mathbf{A} \in \mathbb{R}^{m^2 \times n^2}$ and repeated evaluations of the augmented Lagrangian function required by the backtracking line search. In contrast, 2D TVAL3 AdGD reformulates the reconstruction problem under a 2D CS frame-

TABLE I: MAC operation comparison

Algorithm	MAC Operations per Iteration
TVAL3	$6m^2n^2 + 46n^2 + 11m^2$
2D TVAL3 AdGD	$4mn^2 + 4m^2n + 26n^2 + 3m^2$

TABLE II: Memory usage comparison

Algorithm	Memory Usage
TVAL3	$m^2n^2 + 48n^2 + 9m^2$
2D TVAL3 AdGD	$2mn + 32n^2 + 2m^2$

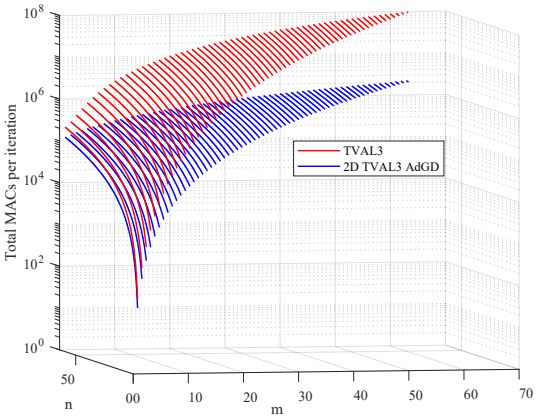

Fig. 4: Comparison of MAC operations per iteration.

work, reducing the dimension of matrix multiplications. Furthermore, by employing AdGD, the algorithm avoids repeated evaluations of the augmented Lagrangian function. As shown in Table II, 2D TVAL3 AdGD additionally offers considerable memory savings. The memory savings primarily stem from the 2D CS framework, which reduces the size of the sensing matrix from $O(m^2n^2)$ to $O(mn)$. Additionally, the simplified update rules in AdGD reduce the need to store intermediate terms involved in the augmented Lagrangian evaluation. Figure 4 further illustrates the computational savings by comparing the number of MAC operations per iteration under varying problem sizes. The proposed method consistently achieves at least one order of magnitude reduction in MAC complexity, especially when the image size n increases, demonstrating its scalability and computational efficiency.

V. Conclusion

This paper presents a total variation-based image reconstruction algorithm that leverages a 2D compressive sensing (CS) model and an adaptive gradient descent strategy. The proposed 2D TVAL3 AdGD significantly reduces computational complexity and memory usage. Compared with existing methods, experimental results confirm that the proposed 2D TVAL3 AdGD algorithm achieves better reconstruction quality than TVAL3, outperforms other conventional compressive sensing algorithms in terms of SSIM, and delivers approximately 21× faster convergence speed than TVAL3, validating its effectiveness and efficiency in practice. In summary, 2D TVAL3 AdGD offers a low-complexity, high-accuracy solution for image reconstruction, making it well-suited for large-scale and embedded imaging applications.

References

[1] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] Y. Nan, Z. Yi, and C. Bingxia, "Review of compressed sensing for biomedical imaging," in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, 2015, pp. 225–228.

[3] X. X. Zhu and R. Bamler, "Exploiting sparsity in remote sensing and earth observation: Theory, applications and future trends," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 2840–2843.

[4] R. Yao and Y. Zhang, "Compressive sensing for small moving space object detection in astronomical images," *Journal of Systems Engineering and Electronics*, vol. 23, no. 3, pp. 378–384, 2012.

[5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[6] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[7] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992, ISSN: 0167-2789.

[8] C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," *Computational Optimization and Applications*, vol. 56, pp. 507–530, 2013.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[10] J. M. B. J. Barzilai, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, Jan. 1988.

[11] H. Zhang and W. W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[12] Y. Malitsky and K. Mishchenko, *Adaptive gradient descent without descent*, 2020. arXiv: 1910 . 09529 [math.OC].

[13] Y. Malitsky and K. Mishchenko, *Adaptive proximal gradient method for convex optimization*, 2024. arXiv: 2308.02261 [math.OC].