



**HAL**  
open science

# An Accelerated Tensorial Double Proximal Gradient Method for Total Variation Regularization Problem

Oumaima Benchettou, Abdeslem Hafid Bentbib, Abderrahman Bouhamidi

► **To cite this version:**

Oumaima Benchettou, Abdeslem Hafid Bentbib, Abderrahman Bouhamidi. An Accelerated Tensorial Double Proximal Gradient Method for Total Variation Regularization Problem. *Journal of Optimization Theory and Applications*, 2023, 198 (1), pp.111-134. 10.1007/s10957-023-02234-z. hal-04411759

**HAL Id: hal-04411759**

**<https://hal.science/hal-04411759v1>**

Submitted on 23 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# AN ACCELERATED TENSORIAL DOUBLE PROXIMAL GRADIENT METHOD FOR TOTAL VARIATION REGULARIZATION PROBLEM

O. Benchettou · A. H. Bentbib ·

A. Bouhamidi

Received: date / Accepted: date

**Abstract** We consider the constrained tensorial total variation minimization problem for regularizing ill-posed multidimensional problems arising in many fields, such as image and video processing, multidimensional data completion, etc. The non-linearity and the non-differentiability of the total variation minimization problem make the resolution directly more complex. The aim of the

---

O. Benchettou

Laboratory LAMAI, University Cadi Ayyad, Marrakech, Morocco.

Laboratory L.M.P.A, University Littoral, Côte d'Opale, Calais, France.

E-mail: oumaima.benchettou@univ-littoral.fr, oumaima.benchettou@ced.uca.ma

A. H. Bentbib

Laboratory LAMAI, University Cadi Ayyad, Marrakech, Morocco.

E-mail: a.bentbib@uca.ac.ma

A. Bouhamidi

Laboratory L.M.P.A, University Littoral, Côte d'Opale, Calais, France.

E-mail: abderrahman.bouhamidi@univ-littoral.fr

present paper is to bring together the resolution of this problem using an iterative tensorial double proximal gradient algorithm and the acceleration of the convergence rate by updating some efficient extrapolation techniques in the tensor form. The general structure of the proposed method expands its fields of application. We will restrict our numerical application to the multidimensional data completion which illustrates the effectiveness of the proposed algorithm.

**Keywords** Tensorial total variation regularization, proximal gradient method, polynomial extrapolation, t-product, video completion and inpainting.

## 1 Introduction

We consider solving a class of tensor convex optimization problem of the form:

$$\min_{\mathcal{X} \in \Omega} \left( \frac{1}{2} \|\mathcal{H}(\mathcal{X}) - \mathcal{B}\|_F^2 + \mu \|\|\nabla \mathcal{X}\|_1 \right), \quad (1)$$

where the solution  $\mathcal{X}$  and the observation  $\mathcal{B}$  are  $N$ th-order tensors,  $\mathcal{H}$  is a given linear tensor operator and the set  $\Omega$  is assumed to be a convex constraint over  $\mathcal{X}$ . The regularization term consists of the tensorial total variation regularization operator  $\|\|\nabla \mathcal{X}\|_1$  and the positive regularization parameter  $\mu$ . The norms  $\|\cdot\|_F$  and  $\|\|\cdot\|_1$  will be defined in the next section.

The convex optimization problem (1) has the form of the well-known total variation regularization method. This regularization technique was first introduced in [42] with the explicit goal of preserving sharp discontinuities

(edges) in a two-dimensional image while removing noise and other unwanted fine-scale detail. Over the years, this model has been extended to many other applications as image processing tasks [14, 18] including inpainting [31], blind-deconvolution and data completion [46]. It has also been modified in a variety of ways to improve its performance [35, 39, 30, 4].

The proposed problem (1) represents a constrained multidimensional total variation regularization problem that will cover a wide range of application fields, such as color image and video processing. Our contribution to this work is threefold. Firstly, a gradient descent-like algorithm is developed to minimize the non-differentiable and non-linear total variation problem over a convex set by computing first the proximal mapping of the total variation term and projecting after the problem using Tseng's Splitting Algorithm [2]. Secondly, since the gradient algorithms have a slow convergence rate, we will accelerate our proposed algorithm using some extrapolation techniques. Finally, such methods are represented in the tensorial representation which may extend the range of application of our model and developed algorithm. To the best of our knowledge the proposed techniques are new.

The paper is organized as follows. In Section 2, we review some standard definitions. In Section 3, we establish a multidimensional double proximal gradient algorithm applied with the tensorial form of the objective function in the minimization problem (1). We propose, in Section 4, to accelerate the algorithm by updating some efficient extrapolation techniques. Section 5 is devoted to apply the proposed algorithm to the low-rank tensor completion.

The performance of the proposed algorithm is tested on color images, video completion, and inpainting in Section 6. Finally, we state the conclusions in Section 7.

## 2 Notation and Background

Tensor algebra is developed in direct response to higher dimensional representation and analysis. Keeping the data or the operation in its natural multidimensional form increases the ability of systems to collect and store vast volumes of multifaceted data and also preserve the modeling accuracy.

First, let us recall some preliminaries and notation on tensor algebra (more details about tensor algebra can be found in [11, 26, 28]). In the remainder of our paper, we adopt the notation used in [26]. We use low-case letters for vectors, e.g.  $a$ , upper-case letters for matrices, e.g.  $A$ , and calligraphic letters for tensors, e.g.  $\mathcal{A}$ . Let us denote by  $\mathbb{X}$  the space  $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and by  $\mathbb{Y}$  the space  $\mathbb{X}^N := \mathbb{X} \times \mathbb{X} \times \dots \times \mathbb{X}$ . Let  $\mathcal{X} \in \mathbb{X}$  be an  $N$ th-order tensor of size  $I_1 \times \dots \times I_N$ . Its entries are denoted by  $\mathcal{X}_{i_1, i_2, \dots, i_N}$  or  $\mathcal{X}(i_1, i_2, \dots, i_N)$ . Let us denote by  $\mathcal{O}$  the tensor having all its entries equal to zero. The *mode- $n$  matricization* (also known as *mode- $n$  unfolding*) of a tensor  $\mathcal{X}$  is denoted by the matrix  $\mathbf{X}_{(n)}$ , and the frontal slices are denoted by the matrices  $X_n$ . Let  $k \geq 1$  be an integer. The norm of the tensor  $\mathcal{X}$  is defined by

$$\|\mathcal{X}\|_k = \left( \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} |\mathcal{X}(i_1, i_2, \dots, i_N)|^k \right)^{1/k}.$$

We also will need to introduce the norm  $\|\cdot\|_\infty$  given by

$$\|\mathcal{X}\|_\infty = \max_{\substack{1 \leq i_j \leq I_j \\ 1 \leq j \leq N}} |\mathcal{X}(i_1, i_2, \dots, i_N)|.$$

The inner product of two same sized tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is defined by

$$\langle \mathcal{X} | \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} \mathcal{X}(i_1, i_2, \dots, i_N) \mathcal{Y}(i_1, i_2, \dots, i_N).$$

It follows immediately that  $\|\mathcal{X}\|_2 = \sqrt{\langle \mathcal{X} | \mathcal{X} \rangle}$ . For the case  $k = 2$ , we use the standard notation of the Frobenius norm  $\|\cdot\|_F$  to denote the tensor norm  $\|\cdot\|_2$ .

Let  $f : \mathbb{X} \rightarrow \mathbb{R} \cup \{\infty\}$  be a closed proper convex function [40]. We recall the proximal operator of  $f$  in the following definition.

**Definition 2.1** [40] *The proximal operator (also called the proximal mapping) of  $f$  is the operator given by*

$$\text{prox}_f(\mathcal{U}) = \underset{\mathcal{X}}{\operatorname{argmin}} \left( f(\mathcal{X}) + \frac{1}{2} \|\mathcal{X} - \mathcal{U}\|^2 \right) \quad \text{for any } \mathcal{U} \text{ in } \mathbb{X}. \quad (2)$$

Since the cost function of the minimization problem defined above is strongly convex and not everywhere infinite, then, there exist a unique minimizer for every  $\mathcal{U} \in \mathbb{X}$ , see [2, 40] for more details. We will often encounter the proximal operator of the scaled function  $\lambda f$  with  $\lambda > 0$ , which can be expressed as

$$\text{prox}_{\lambda f}(\mathcal{U}) = \underset{\mathcal{X}}{\operatorname{argmin}} \left( f(\mathcal{X}) + \frac{1}{2\lambda} \|\mathcal{X} - \mathcal{U}\|^2 \right). \quad (3)$$

The operator  $\text{prox}_{\lambda f}$  is also called the proximal operator of  $f$  with the parameter  $\lambda$ .

**Definition 2.2** *The convex conjugate of  $f$ , denoted  $f^*$ , is given by*

$$f^*(\mathcal{Y}) = \sup_{\mathcal{X}} (\langle \mathcal{X} | \mathcal{Y} \rangle - f(\mathcal{X})), \quad \forall \mathcal{Y} \in \mathbb{X}, \quad (4)$$

### 3 Tensorial Double Proximal Gradient Method for Total Variation Problem

The main goal of this work is the resolution of the constrained tensorial total variation minimization problem (1). It will be useful to consider the following closed proper convex functions

$$\begin{aligned} \mathcal{F} : \mathbb{X} &\longrightarrow \mathbb{R}_+ \\ \mathcal{X} &\longrightarrow \mathcal{F}(\mathcal{X}) = \frac{1}{2} \|\mathcal{H}(\mathcal{X}) - \mathcal{B}\|_F^2, \\ \mathcal{G} : \mathbb{X} &\longrightarrow \mathbb{R}_+ \\ \mathcal{X} &\longrightarrow \mathcal{G}(\mathcal{X}) = \mu \|\|\nabla \mathcal{X}\|_1. \end{aligned}$$

Note that the gradient operator of an  $N$ th-order tensor  $\mathcal{X} \in \mathbb{X}$  is defined as a column block tensor  $\nabla \mathcal{X}$  in  $\mathbb{Y}$  consisting of the partial derivatives  $(\nabla_{(n)} \mathcal{X})_n$ , i.e.  $\nabla \mathcal{X} = \left( \nabla_{(1)} \mathcal{X}, \dots, \nabla_{(N)} \mathcal{X} \right)$ . For  $n = 1, \dots, N$ , the block tensor  $\nabla_{(n)} \mathcal{X}$  is given by

$$(\nabla_{(n)} \mathcal{X})_{i_1, \dots, i_N} = \begin{cases} \mathcal{X}_{i_1, \dots, i_n+1, \dots, i_N} - \mathcal{X}_{i_1, \dots, i_n, \dots, i_N} & \text{if } i_n < I_n \\ 0 & \text{if } i_n = I_n. \end{cases}$$

The convex constrained minimization problem (1) can be written as:

$$\min_{\mathcal{X} \in \Omega} (\mathcal{F}(\mathcal{X}) + \mathcal{G}(\mathcal{X})), \quad (5)$$

where  $\Omega$  is a convex nonempty bounded set. As  $\mathcal{F}$  and  $\mathcal{G}$  are proper lower semicontinuous convex functions,  $\mathcal{F}$  is Gateau differentiable and uniformly

convex on  $\mathbb{X}$  and if it is further assumed that  $\Omega$  is closed, then, there exists a unique solution of the minimization problem (5), see [2,41,13,21,15] for a deeper discussion.

As in the classical vectorial case, the function  $\mathcal{F}$  is differentiable, and its gradient on  $\mathcal{X}$  is given by

$$\nabla \mathcal{F}(\mathcal{X}) = \mathcal{H}^*(\mathcal{H}(\mathcal{X}) - \mathcal{B}), \quad (6)$$

and the function  $\mathcal{G}$  is not differentiable due to the non differentiability of the  $l^1$  norm, which make the resolution of this minimization problem more complex.

### 3.1 Tensorial Double Proximal Gradient Algorithm

In this section, we will introduce an interesting extension of gradient descent method to handle this tensorial convex minimization problem. In the literature, the gradient descent technique is developed in a variety of ways to handle different minimization problems, such as nonlinear minimization problems [13], fractional optimization problems [5] and others. The proximal gradient method represents a generalized form of the gradient descent method in the presence of non differentiability in the cost function [1,2,40,41]. First, we consider the unconstrained minimization problem

$$\min_{\mathcal{X} \in \mathbb{X}} (\mathcal{F}(\mathcal{X}) + \mathcal{G}(\mathcal{X})). \quad (7)$$

Suppose that, at the step  $k$ , we have constructed an iterate tensor  $\mathcal{X}_k$  that approximate the solution of the constrained minimization problem (5). The



quadratic approximation of  $\mathcal{F}$  based at the iterate tensor  $\mathcal{X}_k$ , for  $\lambda_k > 0$ , is given by

$$\Phi_k(\mathcal{X}) = \mathcal{F}(\mathcal{X}_k) + \langle \mathcal{X} - \mathcal{X}_k | \nabla \mathcal{F}(\mathcal{X}_k) \rangle + \frac{1}{2\lambda_k} \|\mathcal{X} - \mathcal{X}_k\|_F^2. \quad (8)$$

Then, it is immediate to see that the problem (7) is approached, at each step  $k$ , by the following minimization problem

$$\min_{\mathcal{X}} \left( \mathcal{G}(\mathcal{X}) + \frac{1}{2\lambda_k} \|\mathcal{X} - \mathcal{X}_k + \lambda_k \nabla \mathcal{F}(\mathcal{X}_k)\|_F^2 \right), \quad (9)$$

which admits a unique minimizer  $\mathcal{Z}_k$  given by

$$\mathcal{Z}_k = \text{prox}_{\lambda_k \mathcal{G}}(\mathcal{X}_k - \lambda_k \nabla \mathcal{F}(\mathcal{X}_k)), \quad \forall k \in \mathbb{N}, \quad (10)$$

where the operator  $\text{prox}_{\lambda_k \mathcal{G}}$  denotes the proximal mapping of  $\mathcal{G}$  with the parameter  $\lambda_k$ .

In general, the algorithm proposed for computing  $\mathcal{Z}_k$  by (10) required two essential elements. The first one is an optimal selection of the step size sequence  $(\lambda_k)_k$  that depend on the Lipschitz constant of  $\nabla \mathcal{F}$  (to be discussed later), and the second one is the computation of the proximal operator of  $\lambda_k \mathcal{G}$  which is given in the following proposition.

**Proposition 3.1** *For all  $\mathcal{Y} \in \mathbb{X}$  and  $\lambda > 0$ , the proximal operator of  $\lambda \mathcal{G}$  is given by*

$$\mathcal{Z} = \text{prox}_{\lambda \mathcal{G}}(\mathcal{Y}) = \mathcal{Y} + \lambda \nabla^T(\mathcal{P}^*), \quad (11)$$

where  $\mathcal{P}^*$  is an optimal solution of

$$\min_{\mathcal{P}} \left( \mathcal{G}_1^*(-\mathcal{P}) + \left\langle \frac{\lambda}{2} \nabla(\nabla^T \mathcal{P}) + \nabla \mathcal{Y} | \mathcal{P} \right\rangle \right), \quad (12)$$

with the function  $\mathcal{G}_1^*$  is the conjugate function of  $\mathcal{G}_1 := \mu \|\cdot\|_1$ .

*Proof* By Definition 2.1, for all  $\mathcal{Y}$  and  $\lambda > 0$ ,  $\text{prox}_{\lambda\mathcal{G}}(\mathcal{Y})$  is the the unique optimal solution of following unconstrained minimization problem

$$\min_{\mathcal{U}} \left( \mathcal{G}(\mathcal{U}) + \frac{1}{2\lambda} \|\mathcal{U} - \mathcal{Y}\|_F^2 \right). \quad (13)$$

If we assume that  $\mathcal{G}(\mathcal{U}) = \mathcal{G}_1(\nabla\mathcal{U})$ , with  $\mathcal{G}_1(\mathcal{V}) = \mu\|\mathcal{V}\|_1$ , the minimization problem can be transformed to a constrained minimization problem as follow

$$\begin{cases} \min_{\mathcal{U}, \mathcal{V}} \left( \mathcal{G}_1(\mathcal{V}) + \frac{1}{2\lambda} \|\mathcal{U} - \mathcal{Y}\|_F^2 \right) \\ \text{s.t.} \quad \mathcal{V} = \nabla\mathcal{U}. \end{cases} \quad (14)$$

The Lagrangian function associated with this problem is defined as

$$\mathcal{L}(\mathcal{U}, \mathcal{V}, \mathcal{P}) = \mathcal{G}_1(\mathcal{V}) + \frac{1}{2\lambda} \|\mathcal{U} - \mathcal{Y}\|_F^2 + \langle \mathcal{P} | \mathcal{V} - \nabla\mathcal{U} \rangle. \quad (15)$$

As a consequence, the solution of (14) is exactly the saddle point of  $\mathcal{L}$  (see [20]), which is the solution of the Lagrangian primal problem

$$\min_{\mathcal{U}, \mathcal{V}} \max_{\mathcal{P}} \mathcal{L}(\mathcal{U}, \mathcal{V}, \mathcal{P}). \quad (16)$$

Since the Lagrangian is separable with respect to  $\mathcal{U}$  and  $\mathcal{V}$ , then we may switch the min-max order based on the min-max theorem [15, 41]. As a consequence, the Lagrangian dual problem can be written as

$$\max_{\mathcal{P}} \left[ \min_{\mathcal{U}} \left( \frac{1}{2\lambda} \|\mathcal{U} - \mathcal{Y}\|_F^2 - \langle \mathcal{P} | \nabla\mathcal{U} \rangle \right) + \min_{\mathcal{V}} (\mathcal{G}_1(\mathcal{V}) + \langle \mathcal{P} | \mathcal{V} \rangle) \right]. \quad (17)$$

On the one hand, it is clear that the minimizer of the problem in  $\mathcal{U}$  is given by  $\mathcal{U}^* = \mathcal{Y} + \lambda\nabla^T\mathcal{P}$  with a corresponding optimal value equal to:

$$\begin{aligned} \min_{\mathcal{U}} \left( \frac{1}{2\lambda} \|\mathcal{U} - \mathcal{Y}\|_F^2 - \langle \mathcal{P} | \nabla\mathcal{U} \rangle \right) &= \frac{1}{2\lambda} \|\mathcal{U}^* - \mathcal{Y}\|_F^2 - \langle \mathcal{P} | \nabla\mathcal{U}^* \rangle \\ &= -\langle \frac{\lambda}{2} \nabla(\nabla^T\mathcal{P}) + \nabla\mathcal{Y} | \mathcal{P} \rangle. \end{aligned} \quad (18)$$

On the other hand, the second minimization problem verifies

$$\min_{\mathcal{V}} (\mathcal{G}_1(\mathcal{V}) + \langle \mathcal{P} | \mathcal{V} \rangle) = - \max_{\mathcal{V}} (\langle -\mathcal{P} | \mathcal{V} \rangle - \mathcal{G}_1(\mathcal{V})) = -\mathcal{G}_1^*(-\mathcal{P}), \quad (19)$$

where we recall that  $\mathcal{G}_1^*$  is the convex conjugate function of  $\mathcal{G}_1$ . As a result, we obtain the following dual problem

$$\max_{\mathcal{P}} \left[ -\mathcal{G}_1^*(-\mathcal{P}) - \left\langle \frac{\lambda}{2} \nabla(\nabla^T \mathcal{P}) + \nabla \mathcal{Y} | \mathcal{P} \right\rangle \right], \quad (20)$$

that can be rewritten as the minimization problem (12).  $\square$

So far, we have shown that  $\text{prox}_{\lambda \mathcal{G}}(\mathcal{Y}) = \mathcal{Y} + \lambda \nabla^T \mathcal{P}^*$ , where  $\mathcal{P}^*$  is an optimal solution of the minimization problem (12). In other words, the calculation of the proximal mapping of the function  $\mathcal{G}$  required, at each iteration  $k$ , the resolution of the minimization problem (12). For  $\mathcal{Y} = \mathcal{Y}_k := \mathcal{X}_k - \lambda_k \nabla \mathcal{F}(\mathcal{X}_k)$ , and  $\lambda = \lambda_k$ , at the step  $k$ , we have to solve the problem

$$\min_{\mathcal{P}} \left( \mathcal{G}_1^*(-\mathcal{P}) + \left\langle \frac{\lambda_k}{2} \nabla(\nabla^T \mathcal{P}) + \nabla \mathcal{Y}_k | \mathcal{P} \right\rangle \right). \quad (21)$$

For all  $k \in \mathbb{N}$ , let us consider the operators  $\mathcal{K}$  and  $(\mathcal{D}_k)_k$  defined as

$$\begin{aligned} \mathcal{K} : \mathbb{Y} &\longrightarrow \mathbb{R}_+ \\ \mathcal{P} &\longrightarrow \mathcal{K}(\mathcal{P}) = \mathcal{G}_1^*(-\mathcal{P}), \\ \mathcal{D}_k : \mathbb{Y} &\longrightarrow \mathbb{R} \\ \mathcal{P} &\longrightarrow \mathcal{D}_k(\mathcal{P}) = \left\langle \frac{\lambda_k}{2} \nabla(\nabla^T \mathcal{P}) + \nabla \mathcal{Y}_k | \mathcal{P} \right\rangle. \end{aligned}$$

Since the objective functional of the minimization problem (21) is a sum of closed proper convex function  $\mathcal{K}$ , and closed proper convex differentiable function  $\mathcal{D}_k$ , then, we can use again the proximal gradient approaches to solve (21).

Hence, the solution can be approximated by the sequence  $(\mathcal{P}_l)_l$  defined as

$$\forall l \in \mathbb{N}, \quad \mathcal{P}_{l+1} = \text{prox}_{\alpha_l \mathcal{K}}(\mathcal{P}_l - \alpha_l \nabla \mathcal{D}_k(\mathcal{P}_l)), \quad (22)$$

with  $\alpha_l > 0$  is a step size parameter. Notice that the expression of the proximal gradient method (22) required two important ingredients; the gradient of the differentiable function  $\mathcal{D}_k$  and the proximal mapping of the non differentiable function  $\mathcal{K}$ . It is immediate to see that the gradient of  $\mathcal{D}_k$  is given by:

$$\forall k, \quad \nabla \mathcal{D}_k(\mathcal{P}) = \lambda_k \nabla(\nabla^T \mathcal{P}) + \nabla \mathcal{Y}_k. \quad (23)$$

In the other hand, the proximal mapping of  $\text{prox}_{\alpha_l \mathcal{K}}$  is discussed in the following proposition.

**Proposition 3.2** *For all  $\mathcal{P} \in \mathbb{Y}$ , the proximal mapping of  $\alpha_l \mathcal{K}$  is given by*

$$\text{prox}_{\alpha_l \mathcal{K}}(\mathcal{P}) = \mathcal{P} + \text{prox}_{\alpha_l \mathcal{G}_1}(-\mathcal{P}), \quad \forall l \in \mathbb{N}, \quad (24)$$

where  $\mathcal{G}_1 := \mu \|\cdot\|_1$ .

*Proof* For any  $\mathcal{P} \in \mathbb{Y}$ , we have

$$\begin{aligned} \text{prox}_{\alpha_l \mathcal{K}}(\mathcal{P}) &= \underset{\mathcal{W}}{\text{argmin}} \left( \mathcal{K}(\mathcal{W}) + \frac{1}{2\alpha_l} \|\mathcal{W} - \mathcal{P}\|_F^2 \right), \\ &= \underset{\mathcal{W}}{\text{argmin}} \left( \mathcal{G}_1^*(-\mathcal{W}) + \frac{1}{2\alpha_l} \|\mathcal{W} - \mathcal{P}\|_F^2 \right), \\ &= - \underset{\mathcal{V}}{\text{argmin}} \left( \mathcal{G}_1^*(\mathcal{V}) + \frac{1}{2\alpha_l} \|\mathcal{V} + \mathcal{P}\|_F^2 \right), \\ &= -\text{prox}_{\alpha_l \mathcal{G}_1^*}(-\mathcal{P}). \end{aligned}$$

According to Moreau decomposition [37], we have the following property that relate the *prox* operator of any proper closed convex function  $f$  by their conjugates

$$\text{prox}_f(x) + \text{prox}_{f^*}(x) = x, \quad \forall x. \quad (25)$$

Using the relation (25), we obtain the desired conclusion

$$\text{prox}_{\alpha_l \mathcal{G}_1}(\mathcal{P}) = \mathcal{P} + \text{prox}_{\alpha_l \mathcal{G}_1}(-\mathcal{P}). \quad (26)$$

Furthermore, the tensorial proximal mapping  $\text{prox}_{\alpha_l \mathcal{G}_1} := \text{prox}_{\alpha_l \mu \|\cdot\|_1}$  of the function  $\mathcal{G}_1$  is a direct result of the proximal operator of the  $\|\cdot\|_1$  norm, also known as the soft thresholding operator in the vector case [40]. Then, by using the Moreau's formula (25) and the fact that the  $\|\cdot\|_\infty$  norm is the dual norm of the  $\|\cdot\|_1$  norm, thus, the proximal operator  $\text{prox}_{\eta \|\cdot\|_1}$ , with any  $\eta > 0$ , can be computed based on the orthogonal projection on the  $\|\cdot\|_\infty$ -unit ball [40], which is the unit box. This leads to

$$(\text{prox}_{\eta \|\cdot\|_1}(\mathcal{P}))_{i_1, \dots, i_N} = \begin{cases} \mathcal{P}_{i_1, \dots, i_N} - \eta, & \mathcal{P}_{i_1, \dots, i_N} \geq \eta, \\ 0, & |\mathcal{P}_{i_1, \dots, i_N}| < \eta, \\ \mathcal{P}_{i_1, \dots, i_N} + \eta, & \mathcal{P}_{i_1, \dots, i_N} \leq -\eta, \end{cases} \quad (27)$$

which establishes the formula and ends the proof.  $\square$

As a result, the algorithm (10) computing the sequence  $(\mathcal{X}_k)_k$  can be summarized as the following double iterative algorithm

$$\forall k \in \mathbb{N}, \begin{cases} \forall l = 1, \dots, l_k, \begin{cases} \mathcal{Q}_l = \mathcal{P}_l - \alpha_l \nabla \mathcal{D}_k(\mathcal{P}_l), \\ \mathcal{P}_{l+1} = \mathcal{Q}_l + \text{prox}_{\alpha_l \mathcal{G}_1}(-\mathcal{Q}_l), \end{cases} \\ \mathcal{Y}_k = \mathcal{X}_k - \lambda_k \nabla \mathcal{F}(\mathcal{X}_k), \\ \mathcal{Z}_k = \mathcal{Y}_k + \lambda_k \nabla^T(\mathcal{P}_{l_{k+1}}), \end{cases} \quad (28)$$

where the step size scalar sequences  $(\lambda_k)_k$  and  $(\alpha_l)_l$  depend on the Lipschitz constants  $L(\nabla \mathcal{F})$  and  $L(\nabla \mathcal{D}_k)$ , respectively, if they exist.

In the following subsection, we will compute the approximated solution  $\mathcal{X}_{k+1}$  of the constrained minimization problem (5) at the step  $k + 1$  by projecting the iterate  $\mathcal{Z}_k$  in the convex set  $\Omega$  using the Tseng's splitting approach.

### 3.2 Tseng's Splitting Algorithm

Tseng's splitting algorithm proposed in [2] considers as a modified version of the proximal gradient algorithm (10) used to handle the constrained convex nonsmooth optimization problem. Under a nonempty closed and convex constraint  $\Omega$ , the algorithm may given as follows:

$$\forall k \in \mathbb{N}, \quad \begin{cases} \mathcal{Y}_k = \mathcal{X}_k - \lambda_k \nabla \mathcal{F}(\mathcal{X}_k), \\ \mathcal{Z}_k = \text{prox}_{\lambda_k \mathcal{G}}(\mathcal{Y}_k), \\ \mathcal{R}_k = \mathcal{Z}_k - \lambda_k \nabla \mathcal{F}(\mathcal{Z}_k), \\ \mathcal{X}_{k+1} = \Pi_{\Omega}(\mathcal{X}_k - \mathcal{Y}_k + \mathcal{R}_k). \end{cases} \quad (29)$$

where  $\Pi_{\Omega}$  denotes the orthogonal projection on the convex set  $\Omega$ .

**Theorem 3.1** *Let  $\mathcal{X}^*$  denotes the unique solution of the problem (5). Suppose that  $\Omega$  is furthermore a closed set in  $\mathbb{X}$ . Then, the sequence  $(\mathcal{X}_k)_k$  generated by Algorithm 29 satisfies  $\|\mathcal{X}_k - \mathcal{X}^*\|_F \rightarrow 0$  as  $k \rightarrow \infty$ .*

*Proof* As the functional  $\mathcal{F}$  and  $\mathcal{G}$  are proper lower semicontinuous convex functions,  $\mathcal{F}$  is Gateau differentiable and uniformly convex on  $\mathbb{X}$  and the set  $\Omega$  is a closed convex nonempty subset of  $\mathbb{X}$ , then, the strong convergence of the sequence  $(\mathcal{X}_k)_k$  is an immediate consequence of the general result in [2] (see Pr.27.13 pg. 407).  $\square$

### 3.3 The step size parameters selection

The choice of the step size parameters is considered as a typical condition that ensures the convergence of the sequence  $(\mathcal{X}_k)_k$  to the minimizer of the problem (5). It is required that the values of the step size parameters  $\lambda_k$  and  $\alpha_l$  be in the intervals  $(0, \frac{1}{L(\nabla\mathcal{F})})$  and  $(0, \frac{1}{L(\nabla\mathcal{D}_k)})$ , respectively, where  $L(\nabla\mathcal{F})$  and  $L(\nabla\mathcal{D}_k)$  denote the Lipschitz constants of the operators  $\nabla\mathcal{F}$  and  $\nabla\mathcal{D}_k$ , respectively (see [2, 13] for further details).

For all couple  $(\mathcal{X}, \mathcal{Y})$  in  $\mathbb{X} \times \mathbb{X}$ , we have

$$\begin{aligned} \|\nabla\mathcal{F}(\mathcal{X}) - \nabla\mathcal{F}(\mathcal{Y})\|_F &= 2\|\mathcal{H}^T(\mathcal{H}(\mathcal{X})) - \mathcal{H}^T(\mathcal{H}(\mathcal{Y}))\|_F, \\ &= 2\|\mathcal{H}^T(\mathcal{H}(\mathcal{X} - \mathcal{Y}))\|_F, \\ &\leq 2\|\|\mathcal{H}^T \circ \mathcal{H}\|\|\cdot\|\|\mathcal{X} - \mathcal{Y}\|_F, \end{aligned}$$

where  $\circ$  stands for the composition operation. Then, we may choose as a Lipschitz constant of the operator  $\nabla\mathcal{F}$  the constant

$$L(\nabla\mathcal{F}) = 2\|\|\mathcal{H}^T \circ \mathcal{H}\|\|. \quad (30)$$

As consequence, the step size  $(\lambda_k)_k$  can be chosen as a fixed step size value  $\lambda_k := \lambda \in (0, \frac{1}{L(\nabla\mathcal{F})})$ .

In the case of the operator  $\nabla\mathcal{D}_k$ , for a fixed step  $k$ , a Lipschitz constant  $L(\nabla\mathcal{D}_k)$  is not known, then, the step sizes  $(\alpha_l)$  can be found by a line search method [40], which mean that we apply the proximal gradient method with an easy backtracking stepsize rule as follows

$$\forall l \in \mathbb{N}, \quad \alpha_l = \tau\alpha_{l-1}, \quad (31)$$

where the scalar  $\tau > 0$  is a line search parameter.

#### 4 Accelerated Tensorial Double Proximal Gradient Algorithms for Total Variation Problem

It is well known that the proximal gradient algorithm suffers from a slow convergence rate. We will present in this section an accelerated version of the proximal gradient algorithm consists in adding an extrapolation step in the algorithm, in order to compute the solution in less steps than the basic proximal gradient. A large amount of research has been conducted to different extrapolation algorithms applied to a variety of general problems [7, 24, 43, 9, 33], and others developed of the proximal gradient method [3, 38].

**Definition 4.1** *Let  $(\mathcal{X}_k)_k$  and  $(\mathcal{T}_k)_k$  two convergent sequences to the same limit  $\mathcal{X}^*$ , we say that  $(\mathcal{T}_k)$  converges faster than the sequence  $(\mathcal{X}_k)$  if*

$$\lim_{k \rightarrow \infty} \frac{\|\mathcal{T}_k - \mathcal{X}^*\|}{\|\mathcal{X}_k - \mathcal{X}^*\|} = 0. \quad (32)$$

The goal of the extrapolation is to find a sequence  $(\mathcal{T}_k)_k$  from the sequence  $(\mathcal{X}_k)_k$  so that  $(\mathcal{T}_k)_k$  converges faster to the same limit as  $(\mathcal{X}_k)_k$ . There are many extrapolation methods in the literature, but we will only be interested to apply the Nesterov's algorithm approach and the polynomial extrapolation methods to our tensorial nonlinear minimization problem.



#### 4.1 The tensorial Nesterov acceleration techniques

One simple and widely studied strategy is to perform extrapolation in the spirit of Nesterov's extrapolation techniques [29,38]. The basic idea of this technique is to make use of historical information at each iteration in order to reduce the convergence rate from  $O(1/k)$  to  $O(1/k^2)$ . Thus, using the position of the current iteration tensor and the previous iteration tensor, the tensorial double proximal gradient method is accelerated by adding an extrapolation step given by

$$\mathcal{T}_{k+1} = \mathcal{X}_k + \left( \frac{t_k - 1}{t_{k+1}} \right) (\mathcal{X}_k - \mathcal{X}_{k-1}), \quad (33)$$

where the scalars  $(t_k)$  is given, at each step  $k$ , by  $t_{k+1} = \frac{\sqrt{4t_k^2 + 1} + 1}{2}$ .

The convergence of the sequence  $(\mathcal{T}_k)_{k \in \mathbb{N}}$  may be investigated by following the approaches given in several papers, e.g [13,12]. The complete algorithm summarizes the accelerated tensorial double proximal gradient with Nesterov's extrapolation is given in Algorithm 1.

#### 4.2 The global tensorial polynomial extrapolation methods

The polynomial extrapolation methods are among the best known extrapolation methods thanks to their theoretical clarity and numerical efficiency, especially when applied to solving nonlinear problems such as the case of our main problem (1). The polynomial extrapolation methods was introduced in [43] for the vectorial extrapolation case that developed after in [6,23,24] using efficient implementation techniques. Those methods was also developed

**Algorithm 1:** *TDPG*-Nesterov algorithm

- 
- 1: **Inputs:** Initial guess  $\mathcal{X}_1 = \mathcal{O}$ ,  $\mathcal{P}_1 = \mathcal{O}$ ,  $\mathcal{T}_1 = \mathcal{X}_0$ ,  $\nabla \mathcal{F}$ ,  $t_1 = 1$ ,  $\tau$ ,  $\mu$ ,  $\lambda$ ,  $\alpha_0$ ,  $tol$ .
  - 2: **for**  $k = 1, \dots$  until convergence, **do**
  - 3:    $\mathcal{Y}_k = \mathcal{T}_k - \lambda \nabla \mathcal{F}(\mathcal{T}_k)$ ,
  - 4:   Compute the operator  $\nabla \mathcal{G}_k$  using the formula (23),
  - 5:   **for**  $l = 1, \dots, l_k$  **do**
  - 6:     Update the line search parameter  $\alpha_l$  using (31).
  - 7:      $\mathcal{Q}_l = \mathcal{P}_l - \alpha_l \nabla \mathcal{G}_k(\mathcal{P}_l)$ ,
  - 8:      $\mathcal{P}_{l+1} = \mathcal{Q}_l + \text{prox}_{\alpha_l \mu \|\cdot\|_{\mathbb{H}_1}}(-\mathcal{Q}_l)$ , with  $\text{prox}_{\alpha_l \mu \|\cdot\|_{\mathbb{H}_1}}$  is given in (27).
  - 9:   **end for**
  - 10:    $\mathcal{Z}_k = \mathcal{Y}_k + \lambda \nabla^T(\mathcal{P}_{l_k+1})$ ,
  - 11:    $\mathcal{R}_k = \mathcal{Z}_k - \lambda \nabla \mathcal{F}(\mathcal{Z}_k)$ ,
  - 12:    $\mathcal{X}_k = \Pi_{\Omega}(\mathcal{T}_k - \mathcal{Y}_k + \mathcal{R}_k)$ .
  - 13:    $t_{k+1} = \frac{\sqrt{4t_k^2 + 1} + 1}{2}$ ,
  - 14:   Compute  $\mathcal{T}_{k+1} = \mathcal{X}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathcal{X}_k - \mathcal{X}_{k-1})$ .
  - 15:   End the iteration if  $\|\mathcal{T}_{k+1} - \mathcal{T}_k\|_F / \|\mathcal{T}_k\|_F < tol$ .
  - 16: **end for**
  - 17: **return**  $\mathcal{T}_{k+1}$ .
- 

in a matrix global form in [22], and recently was generalized for the tensor sequences in [16] using tensor product. In the spirit of [24], we define the transformation in the following form

$$\mathcal{T}_{k,q} = \sum_{j=0}^q \gamma_j \mathcal{X}_{k+j}, \quad (34)$$

where  $k$  defines the first term of the sequence, the integer  $q$  stands for the number of terms of the sequence, and the scalars  $(\gamma_j)$  verify the following two conditions

$$\begin{cases} \sum_{j=0}^q \gamma_j = 1, \\ \sum_{j=0}^q \gamma_j \langle \mathcal{V}_i | \Delta \mathcal{X}_{k+j} \rangle = 0, \quad i = 0, 1, \dots, q-1, \end{cases} \quad (35)$$

where  $\Delta \mathcal{X}_j = \mathcal{X}_{j+1} - \mathcal{X}_j$  and  $\mathcal{V}_i$  are given tensors.

As a consequence, the conditions (35) lead to the following linear system

$$\begin{cases} \gamma_0 + \gamma_1 + \dots + \gamma_q = 1 \\ \gamma_0 \langle \mathcal{V}_0 | \Delta \mathcal{X}_k \rangle + \gamma_1 \langle \mathcal{V}_0 | \Delta \mathcal{X}_{k+1} \rangle + \dots + \gamma_q \langle \mathcal{V}_0 | \Delta \mathcal{X}_{k+q} \rangle = 0 \\ \gamma_0 \langle \mathcal{V}_1 | \Delta \mathcal{X}_k \rangle + \gamma_1 \langle \mathcal{V}_1 | \Delta \mathcal{X}_{k+1} \rangle + \dots + \gamma_q \langle \mathcal{V}_1 | \Delta \mathcal{X}_{k+q} \rangle = 0 \\ \vdots \\ \gamma_0 \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_k \rangle + \gamma_1 \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_{k+1} \rangle + \dots + \gamma_q \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_{k+q} \rangle = 0 \end{cases} \quad (36)$$

where the vector  $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \dots, \gamma_q]^T$  is the solution of the matrix equation:

$$\underbrace{\begin{pmatrix} 1 & 1 & \dots & 1 \\ \langle \mathcal{V}_0 | \Delta \mathcal{X}_k \rangle & \langle \mathcal{V}_0 | \Delta \mathcal{X}_{k+1} \rangle & \dots & \langle \mathcal{V}_0 | \Delta \mathcal{X}_{k+q} \rangle \\ \langle \mathcal{V}_1 | \Delta \mathcal{X}_k \rangle & \langle \mathcal{V}_1 | \Delta \mathcal{X}_{k+1} \rangle & \dots & \langle \mathcal{V}_1 | \Delta \mathcal{X}_{k+q} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_k \rangle & \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_{k+1} \rangle & \dots & \langle \mathcal{V}_{q-1} | \Delta \mathcal{X}_{k+q} \rangle \end{pmatrix}}_M \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (37)$$

It is clear that  $\mathcal{T}_{k,q}$  exists and is unique if and only if the square matrix  $M$  is nonsingular. In this work, we consider two polynomial extrapolation methods, the global tensor minimal polynomial extrapolation (*GT-MPE*), where the sequence  $\mathcal{V}_i$  is defined as

$$\mathcal{V}_i = \Delta \mathcal{X}_{i+k}, \quad (38)$$

and the global tensor reduced rank extrapolation method (*GT-RRE*) with

$$\mathcal{V}_i = \Delta^2 \mathcal{X}_{i+k} = \Delta \mathcal{X}_{i+k+1} - \Delta \mathcal{X}_{i+k}. \quad (39)$$

Note that, there is an integer  $q_0$ , such that  $\{\Delta \mathcal{X}_0, \Delta \mathcal{X}_1, \dots, \Delta \mathcal{X}_{q_0-1}\}$  is a linearly independent set, but  $\{\Delta \mathcal{X}_0, \Delta \mathcal{X}_1, \dots, \Delta \mathcal{X}_{q_0-1}, \Delta \mathcal{X}_{q_0}\}$  is not. As a consequence, under the condition  $q < q_0$ , both *GT-MPE* and *GT-RRE* produce approximations  $\mathcal{T}_{k,q}$  of the solution  $\mathcal{X}^*$  in the form (34). For more details, we refer the reader to [43,24].

The process of polynomial extrapolation using *GT-MPE* or *GT-RRE* is summarized in Algorithm 2.

---

**Algorithm 2:** The *GT-MPE/ GT-RRE* algorithm

---

- 1: **Inputs:** The sequence  $\{\mathcal{X}_k, \dots, \mathcal{X}_{k+q+1}\}$ .
  - 2: Compute  $M$ :
  - 3: For *GT-MPE*:  $\mathcal{V}_i = \Delta \mathcal{X}_{i+k}$ .
  - 4: For *GT-RRE*:  $\mathcal{V}_i = \Delta^2 \mathcal{X}_{i+k}$ .
  - 5: Solve the matrix equation  $M\boldsymbol{\gamma} = \mathbf{e}_1$ .
  - 6: Compute the approximation  $\mathcal{T}_{k,q}$  using the expression (34).
  - 7: **Output:**  $\mathcal{T}_{k,q}$ .
- 

For the *GT-MPE* and *GT-RRE* methods, the number of calculations required increases quadratically with the number of iterations  $q$  and the storage cost increases linearly. A good method to keep the cost of storage and the cost of the lowest possible calculations is to restart these algorithms periodically. The following algorithm describes the restarted version of those methods called also a cycling mode.

---

**Algorithm 3:** Cycling mode.
 

---

- 1: Fix the integer  $q$ .
  - 2: Form the sequence  $\{\mathcal{X}_1, \dots, \mathcal{X}_{q+1}\}$ .
  - 3: Calculate the approximation  $\mathcal{T}_{k,q}$  using the algorithm 2.
  - 4: If  $\mathcal{T}_{k,q}$  is satisfactory, stop. Otherwise, set  $\mathcal{X}_k = \mathcal{T}_{k,q}$  as a new initialization and  $k = k + 1$ , and go to the second step.
- 

**Remark 4.1** *In the polynomial vector extrapolation case, the author in [43, 24] proposed a numerically stable and computationally economical algorithm for computing the  $(\gamma_i)_i$  via the QR factorization. The same concept was developed for the tensor case in [16] by defining a new QR factorization of the tensor  $\mathcal{U}_k$  contents the sequence  $(\Delta\mathcal{X}_i)$  as frontal slices. However, in our situation, we have only used a direct method for solving the matrix equation (37).*

The accelerated version of the tensorial double proximal gradient algorithm using the polynomial method is summarized in Algorithm 4.

## 5 Application in Low Rank Tensor Completion

In this section, we will apply the proposed algorithm to the tensor completion problems. Completion is a technique of filling missing elements of incomplete data using the values of available elements and the structural assumptions of data. Let us consider the  $N$ th-order tensor  $\mathcal{G}$  with observed entries indexed by the set  $C$  i.e.  $C = \{(i_1, i_2, \dots, i_N) : \mathcal{X}_{i_1, i_2, \dots, i_N} \text{ is observed}\}$ . Following the same definition of Candes and Tao in [8], we define, in the tensor form, the projection  $P_C(\mathcal{X})$  to be the  $N$ th-order tensor with the observed elements of

**Algorithm 4:** *TDPG*-Polynomial extrapolation algorithm

- 
- 1: Initial guess  $\mathcal{X}_0 = \mathcal{O}$ ,  $\mathcal{P}_0 = \mathcal{O}$   $\mathcal{T}_1 = \mathcal{X}_0$ ,  $\nabla \mathcal{F}$ ,  $\mu$ ,  $\lambda$ ,  $\tau$ ,  $\alpha_0$ ,  $q$ ,  $tol$ .
  - 2: **for**  $k = 1, \dots$  until convergence, **do**
  - 3:    $\mathcal{Y}_k = \mathcal{T}_k - \lambda_k \nabla \mathcal{F}(\mathcal{T}_k)$ ,
  - 4:   Compute the operator  $\nabla \mathcal{G}_k$  using the formula (23),
  - 5:   **for**  $l = 1, \dots, l_k$  **do**
  - 6:     Update the line search parameter  $\alpha_l$  using (31).
  - 7:      $\mathcal{Q}_l = \mathcal{P}_l - \alpha_l \nabla \mathcal{G}_k(\mathcal{P}_l)$ ,
  - 8:      $\mathcal{P}_{l+1} = \mathcal{Q}_l + \text{prox}_{\alpha_l \mu \|\cdot\|_{\mathbb{1}_1}}(-\mathcal{Q}_l)$ , with  $\text{prox}_{\alpha_l \mu \|\cdot\|_{\mathbb{1}_1}}$  is given in (27).
  - 9:   **end for**
  - 10:    $\mathcal{Z}_k = \mathcal{Y}_k + \lambda \nabla^T(\mathcal{P}_{l_{k+1}})$ ,
  - 11:    $\mathcal{R}_k = \mathcal{Z}_k - \lambda \nabla \mathcal{F}(\mathcal{Z}_k)$ ,
  - 12:    $\mathcal{X}_k = \Pi_{\Omega}(\mathcal{T}_k - \mathcal{Y}_k + \mathcal{R}_k)$ .
  - 13:   Compute the iterate  $\mathcal{T}_{k+1}$  using Algorithm 3.
  - 14:   End the iteration if  $\|\mathcal{T}_{k+1} - \mathcal{T}_k\|_F / \|\mathcal{T}_k\|_F < tol$
  - 15: **end for**
  - 16: **return**  $\mathcal{T}_{k+1}$ .
- 

$\mathcal{X}$  preserved and the missing entries replaced with 0, namely,

$$P_C(\mathcal{X}) = \begin{cases} \mathcal{X}_{i_1, i_2, \dots, i_N} & \text{if } (i_1, i_2, \dots, i_N) \in C, \\ 0 & \text{otherwise.} \end{cases} \quad (40)$$

One of the variants of the data completion problem is to find the lowest rank which matches the observed data. This leads to an NP-hard rank minimization problem due to the non-convexity of the rank function [45]. For that purpose, we need to replace the rank function with something similar that provides the same results. Therefore, the nuclear norm minimization method [32, 17, 47] is widely used in this case to replace the rank minimization problem by

the following one

$$\begin{cases} \min_{\mathcal{X}} \|\mathcal{X}\|_* \\ \text{subject to } P_C(\mathcal{X}) = P_C(\mathcal{G}). \end{cases} \quad (41)$$

where  $\|\cdot\|_*$  stands for the tensor nuclear norm defined as sum of the singular values of the  $n$ -mode matricization  $\mathbf{X}_{(\mathbf{n})}$  of the tensor  $\mathcal{X}$ , i.e.  $\|\mathcal{X}\|_* = \sum_n \|\mathbf{X}_{(\mathbf{n})}\|_*$ . Tensor completion via total variation minimization was proposed in [46,31] as an efficient technique to regularize the minimization problem (41).

The tensor total variation completion problem is given in the following form:

$$\min_{\mathcal{X}} \|P_C(\mathcal{X}) - P_C(\mathcal{G})\|_F^2 + \|\mathcal{X}\|_* + \mu \|\nabla \mathcal{X}\|_1. \quad (42)$$

The problem (42) can be formulated to a constrained minimization problem as

$$\min_{\mathcal{X} \in \Omega} \|P_C(\mathcal{X}) - P_C(\mathcal{G})\|_F^2 + \mu \|\nabla \mathcal{X}\|_1, \quad (43)$$

where  $\Omega = \{\mathcal{X}, \|\mathcal{X}\|_* \leq \epsilon\}$ .

By setting  $\mathcal{H} = P_C$  and  $\mathcal{B} = P_C(\mathcal{G})$ , the problem (43) leads to the main problem (1). Then, to solve the minimization problem (43), we can use our proposed accelerated tensorial double proximal gradient algorithm. To apply the algorithm, we need to introduce some notation, definitions and properties. The first notion we have to introduce is the t-product. We give the definitions only for the third-order tensor, the other cases are defined recursively [25].

**Definition 5.1** [25] *The t-product of two tensors  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$  and  $\mathcal{B} \in \mathbb{R}^{I_2 \times J \times I_3}$ , is a tensor  $\mathcal{A} *_t \mathcal{B} \in \mathbb{R}^{I_1 \times J \times I_3}$  defined as*

$$\mathcal{A} *_t \mathcal{B} = \mathbf{ibvec}(\mathbf{bcirc}(\mathcal{A}) \cdot \mathbf{bvec}(\mathcal{B})) \quad (44)$$

where the dot  $\cdot$  stands for the usual matrix product, the matrix  $\mathbf{bcirc}(\mathcal{A})$  is a block circulant matrix defined by using the frontal slices of  $\mathcal{A}$ ,  $\mathbf{bvec}(\mathcal{B})$  defined as a block vector contain the frontal slices of  $\mathcal{B}$  and  $\mathbf{ibvec}$  stands for the inverse operation of  $\mathbf{bvec}$ .

$$\mathbf{bcirc}(\mathcal{A}) = \begin{pmatrix} A_1 & A_{I_3} & \cdots & A_2 \\ A_2 & A_1 & \cdots & A_3 \\ \vdots & \ddots & & \vdots \\ A_{I_3} & A_{I_3-1} & \cdots & A_1 \end{pmatrix} \quad \text{and} \quad \mathbf{bvec}(\mathcal{B}) = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_{I_3} \end{pmatrix}.$$

**Definition 5.2** [25] The conjugate transpose of a tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$  is the  $\mathcal{A}^* \in \mathbb{R}^{I_2 \times I_1 \times I_3}$  obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices 2 through  $I_3$ .

**Definition 5.3 (Tensor Singular Value Decomposition: t-SVD)** [26] For  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ , the t-SVD of the third tensor  $\mathcal{A}$  is given by

$$\mathcal{A} = \mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^*, \quad (45)$$

where  $\mathcal{U}$  and  $\mathcal{V}$  are orthogonal tensors of size  $I_1 \times I_1 \times I_3$  and  $I_2 \times I_2 \times I_3$  respectively. The tensor  $\mathcal{S}$  is a rectangular f-diagonal tensor of size  $I_1 \times I_2 \times I_3$ , and the entries in  $\mathcal{S}$  are called the singular values of  $\mathcal{A}$ .

**Definition 5.4** [26] The nuclear norm of  $\mathcal{X}$  is defined using t-SVD decomposition  $\mathcal{X} = \mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^*$  as follow:

$$\|\mathcal{X}\|_* = \sum_{i=1}^r \mathcal{S}(i, i, 1), \quad (46)$$

where  $r$  denotes the tubal rank of  $\mathcal{X}$ .



Let  $\Pi_\Omega$  denotes the projection on the convex set  $\Omega$ . It is immediate to proof that the expression of the projection  $\Pi_\Omega$  on  $\Omega$  reduces to the proximal mapping of the nuclear norm. Namely, for all  $\mathcal{Z}$  in the tensor space  $\mathbb{X}$

$$\Pi_\Omega(\mathcal{Z}) = \text{prox}_{\sigma\|\cdot\|_*}(\mathcal{Z}), \quad (47)$$

where the proximal mapping of the nuclear norm is given in Proposition 5.1.

**Proposition 5.1** [34] *Let  $\mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^*$  be the  $t$ -SVD decomposition of the tensor  $\mathcal{X}$ . The proximal mapping of the nuclear norm is given by*

$$\text{prox}_{\sigma\|\cdot\|_*}(\mathcal{X}) = \mathcal{U} *_t \mathcal{S}_\sigma *_t \mathcal{V}^*, \quad \text{for any } \sigma > 0, \quad (48)$$

where  $\mathcal{S}_\sigma$  is the result of the inverse discrete Fourier transform (IDFT) on  $\max(\bar{\mathcal{S}} - \sigma, 0)$  along the third dimension, which means performing the IDFT on all the tubes.  $\bar{\mathcal{S}}$  is the result of DFT on  $\mathcal{S}$  along the third dimension.

## 6 Numerical Results

This section presents some results illustrate the performance of the developed algorithm. The described model is successfully applied on multidimensional data. Let  $\mathcal{X}_{true}$  denotes the original image or video, and the tensor  $\mathcal{B}$  represents the incomplete data. To evaluate the effectiveness of our algorithm, we use the Peak Signal to Noise Ratio (*PSNR*) and the relative error (*RE*) defined as

$$PSNR = 10 \log_{10} \left( \frac{d^2 I_1 \cdots I_N}{\|\mathcal{X}_{true} - \tilde{\mathcal{X}}\|_F^2} \right) \quad \text{and} \quad RE = \frac{\|\mathcal{X}_{true} - \tilde{\mathcal{X}}\|_F}{\|\mathcal{X}_{true}\|_F}, \quad (49)$$

where  $I_1 \times \cdots \times I_N$  is the size of the approximate solution  $\tilde{\mathcal{X}}$  and  $d$  is the maximal variation in the input data. On the other hand, we adopt the stopping

criterion of the algorithm as follows

$$e_k = \frac{\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F}{\|\mathcal{X}_k\|_F} \leq tol,$$

where the tolerance  $tol$  is chosen in the interval  $[10^{-4}, 10^{-1}]$ . All computations were carried out using the MATLAB R2018a environment on an Intel(R) Core(TM) i7-8550 CPU @ 1,80 GHz 1,99 GHz computer with 16 GB of RAM.

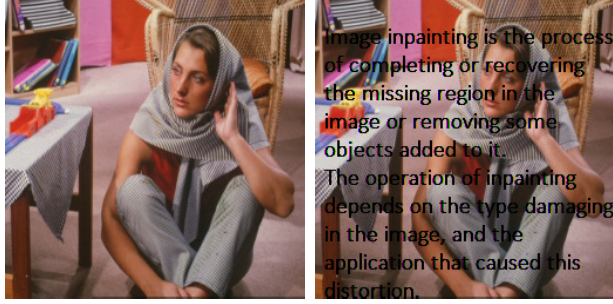
In the following subsections, we will be interested in two essential parts of tensor completion: Color image and video inpainting (text removal) and grayscale video completion. We first illustrate the performance of the proposed algorithms 1 and 4 by comparing three acceleration methods in inpainting different color images and grayscale videos. The completion of the grayscale video is reported after to show the efficiency of our algorithm in case of uniformly random missing pixels. Finally, we end with some comparisons with state-of-art algorithms.

### 6.1 Text removal

As an interesting application of completion problems, the text removal is a process of data inpainting that based on the completion techniques to recover the missing region in the tensor data or removing some objects added to it. The operation of inpainting depends on the type damaging in the image, and the application that caused this distortion. For example, in the text removal process, we talk about removing the text that can be found in an image or a video [44]. In the literature, many techniques have been developed to solve this

problem [27, 19, 36]. The total variation was among the most efficient method to solve such a problem.

In this example, we took the original Barbara color image of size  $256 \times 256 \times 3$  and we added a text to this image like shown in Figure 1. The crite-



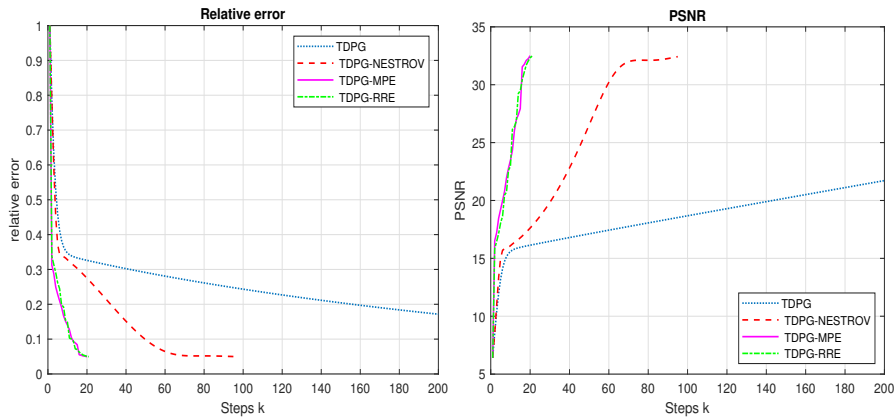
**Fig. 1** Original image (left), the observed image ( $PSNR = 15.5$ ) (right).

riterion for stopping the proposed algorithms consists of the tolerance  $tol = 10^{-2}$ , the maximum number of iterations  $k_{max} = 200$ . We hand turned all the parameters  $\lambda$ ,  $\mu$ ,  $\alpha$  and  $\sigma$  by choosing each one in its appropriate interval. We hand turned the value of  $\lambda$  in the interval  $(0, 1/L(\nabla P_\Omega)) = (0, 1/2)$  by choosing  $\lambda = 2.5 \times 10^{-1}$ . On the other hand the step size sequence  $(\alpha_k)$  computed using the line search iterative method starting by  $\alpha_0 = 1.1$  with a line search parameter  $\tau = 0.5$ . We set  $\sigma = 6.5 \times 10^{-2}$  and finally the regularization parameter was chosen to be  $\mu = 1.2e^{-2}$ . The corresponding results are shown in Figure 2. Clearly, the accelerated version of the tensorial double proximal gradient method provide clearer images by removing all the added text, either using Algorithm 1 based on Nesterov acceleration approach or Algorithm 4 using the polynomial extrapolation techniques. Moreover, the relative error and

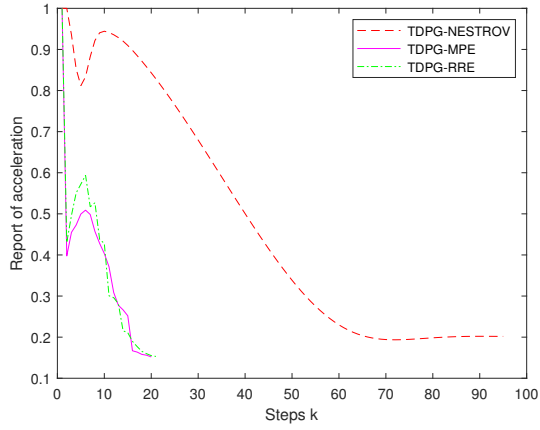


**Fig. 2** Inpainted image without acceleration ( $PSNR = 21.71$ ), inpainted image with TDPG-Nesterov ( $PSNR = 32.45$ ), inpainted image with TDPG-MPE ( $PSNR = 32.5$ ), inpainted image with TDPG-RRE ( $PSNR = 32.47$ ).

the  $PSNR$  curves represented in Figure 3 show that the results produced by the  $TDPG$  algorithm accelerated by the polynomial extrapolation techniques  $RRE$  or  $MPE$  converge faster than those produced by the  $TDPG$  accelerated by Nesterov's technique. The speed of the convergence of the polynomial extrapolation method in comparison with the Nesterov's approach are clearly illustrated in the report of acceleration in Figure 4, that show the fast convergence of  $TDPG-MPE$  and  $TDPG-RRE$  in less iterations than  $TDPG-Nesterov$ .



**Fig. 3** The PSNR and relative error curves.



**Fig. 4** The report of acceleration  $\frac{\|\mathcal{T}_k - \mathcal{X}_*\|}{\|\mathcal{X}_k - \mathcal{X}_*\|}$ .

**Table 1** Comparison between the proposed acceleration techniques for tensorial total variation proximal gradient method.

Data size	Algorithm	$PSNR(\mathcal{X}^*)$	RE	Step	time (s)
$256 \times 256 \times 3$ <i>barbara.bmp</i>	<i>TDPG</i>	21.71	$1.72e-1$	200	27.41
	<i>TDPG-Nesterov</i>	32.45	$5e-2$	97	13.37
	<i>TDPG-MPE</i> ( $q = 5$ )	32.50	$5e-2$	20	19.72
	<i>TDPG-RRE</i> ( $q = 5$ )	32.47	$5e-2$	21	19.88
$120 \times 160 \times 30$ <i>xylophone.mpg</i>	<i>TDPG</i>	28.62	$4.80e-2$	18	6.71
	<i>TDPG-Nesterov</i>	29.65	$3.98e-2$	12	4.10
	<i>TDPG-MPE</i> ( $q = 5$ )	32.27	$2.99e-2$	3	5.38
	<i>TDPG-RRE</i> ( $q = 5$ )	29.53	$4.03e-2$	3	5.66

In Table 1, we have reported the  $PSNR$  of the completed tensor, the relative error, as well as the number of iterations and the CPU-time results for “barbara.bmp” color image and “xylophone.mpg” grayscale video of size

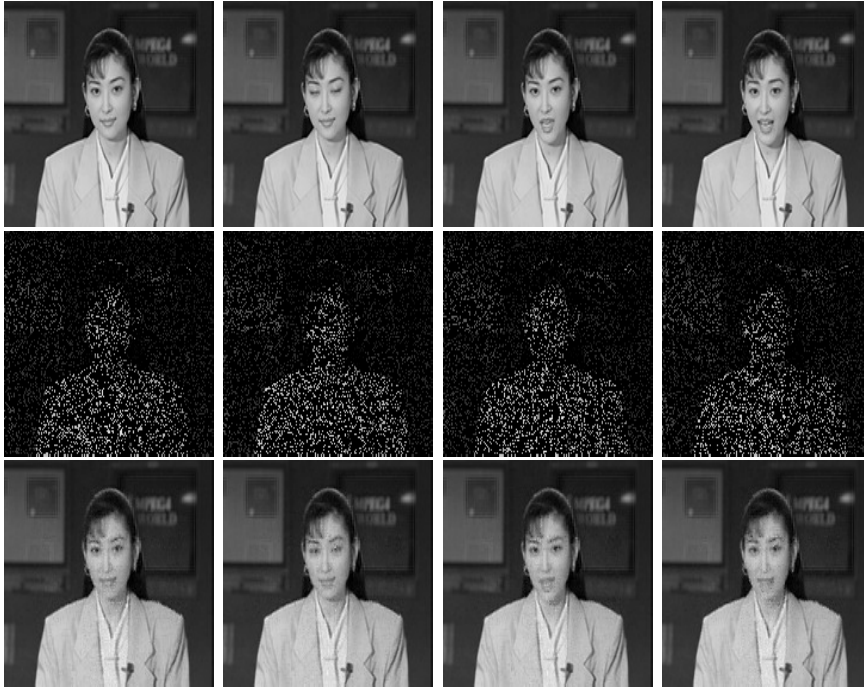
$120 \times 160 \times 30$ . Based on the tests reported in Table 1 and many more unreported tests, we remark that our proposed algorithm works very effectively for image and video inpainting problems, in terms of the *PSNR* as well as in terms of the relative error.

## 6.2 Grayscale video completion

In order to have more quantitative evaluations on the proposed approach, we used the “news.mpg” grayscale video of size  $144 \times 176 \times 10$  as original data and we randomly mask off about 80% of entries that we regard them as missing values, as shown in the second line of Figure 5. The completed frames in Figure 5 with  $PSNR = 33.21$  are obtained by using Algorithm 4 with the *GT-RRE* polynomial extrapolation technique. The criterion for stopping the algorithm consists of the tolerance  $tol = 10^{-2}$  and the maximum number of iterations  $k_{max} = 200$ . We set the step size parameters  $\lambda = 2 \times 10^{-1}$ ,  $\alpha_0 = 1.1$ ,  $\sigma = 6 \times 10^{-1}$  and the regularization parameter  $\mu = 2 \times 10^{-2}$ . We can see that the results are visually pleasant using the accelerated version of the tensorial double proximal gradient method that achieves a *PSNR* value equal to 33.21.

## 6.3 Comparison with state-of-the-art algorithms

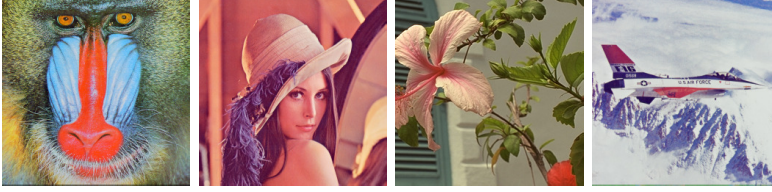
In this subsection, we compare the performance of our proposed method with the following state-of-the-art tensor completion algorithms: *LRTC*, *TNN*, *FBCP* and *RTC*. The *LRTC* algorithms are based on minimizing the sum of nuclear norms of the unfolded matrices of a given tensor. The *LRTC* has two



**Fig. 5** Original frames (top), incompleted frames with  $PSNR = 7.94$  (center), recovered frames with  $PSNR = 33.21$  (bottom).

versions, *HaLRTC* and *FaLRTC* [32]. The first one stands for a fast low-rank tensor completion algorithm and the second stands for a high accuracy low-rank tensor completion algorithm. We also found the *TNN* method [48] which is a tensor nuclear norm-based method developed using the tensor-Singular Value Decomposition (*t-SVD*) [25]. The concept of automatic tensor rank determination was introduced in [49] which is based on a Bayesian *CP* Factorization (*FBCP*) in order to recover incomplete tensor. For the same goal of completing tensors, recently, the *RTC* algorithm [10] was developed as an auto-weighted approach using this time the well-known tensor trains decomposition [26]. Four benchmark color images, of size  $256 \times 256 \times 3$ , are used in

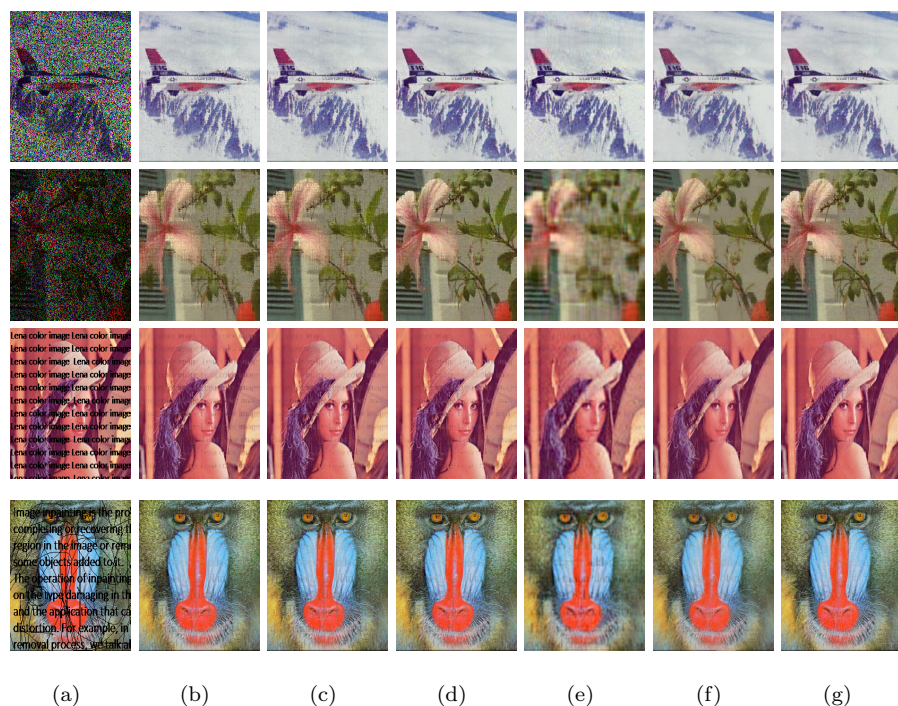
the comparisons, Baboon, Lena, Flower, and Airplane (see Figure 6). To show



**Fig. 6** Four benchmark color images: Baboon, Lena, Flower and Airplane, respectively.

the efficiency of the proposed algorithm for different types of tensor completion, we generate different incomplete images either using uniformly random missing pixels or non-random missing pixels. In the first case, 80% and 60% of missing pixels are uniformly distributed in Flower and Airplane color images, respectively. Non-random missing pixels, such as text and scrabble, are used to corrupt the color images of Lena and Baboon. The corrupted images are shown in the first column of Figure 7. In order to provide a fair and unified framework for comparison, all six algorithms are endowed with the same convergence criterion, i.e. the iterations for all algorithms were terminated when the relative error between two successive iterates of approximated primal variable is less than the tolerance  $tol = 10^{-4}$  or when a maximum of 200 iterations has been performed. In addition, the parameters of the six algorithms are refined in relation to the best  $PSNR$ ,  $RE$ , and  $CPU$  times scores on the images. Table 2 reports the  $PSNR$ , the relative error  $RE$ , as well as the  $CPU$  time in seconds for all the six algorithms. While the recovered images are shown in Figure 7. For the uniformly random missing pixels examples, the proposed  $TGPG$  algorithm is comparable with the TNN algorithm. Both approaches reach the best





**Fig. 7** Image completion comparisons of Airplane, Flower, Lena and Baboon by six algorithms. (a) the column of the observed (incomplete) images, (b) the completed images with *FaLRTV* algorithm, (c) the completed images with *HaLRTV* algorithm, (d) the completed images with *TNN* algorithm, (e) the completed images with *FBCP* algorithm, (f) the completed images with *RTC* algorithm and (g) the completed images with the proposed *TDPG* algorithm.

results in terms of *PSNR* and *RE*. However, by increasing the missing pixels in the Flower color image, the proposed algorithm converges faster than *TNN*. On the other hand, in the non-random missing pixels example, the proposed algorithm achieves the best *PSNR* and *RE* among all other methods. Meanwhile, in terms of computation time, algorithms *RTC* and *FBCP* are faster than ours.

**Table 2** Comparison of the results of six methods applied to four different images.

Data	Methods	<i>FaLRTV</i>	<i>HaLRTV</i>	<i>TNN</i>	<i>FBCP</i>	<i>RTC</i>	<i>TDPG</i>
Airplane	<i>PSNR</i>	25.39	26.14	29.37	26.70	26.83	29.03
	<i>RE</i>	$6.69e-2$	$6.13e-2$	$4.23e-2$	$5.75e-2$	$5.66e-2$	$4.40e-2$
	Time (s)	89.44	58.76	26.30	38.43	15.64	36.44
Flower	<i>PSNR</i>	22.04	22.40	24.94	24.29	23.84	24.97
	<i>RE</i>	$1.64e-1$	$1.58e-1$	$1.18e-1$	$1.27e-1$	$1.34e-1$	$1.17e-1$
	Time (s)	214.71	117.12	43.81	83.66	43.70	28.05
Lena	<i>PSNR</i>	28.13	28.94	28.91	26.39	28.57	30.29
	<i>RE</i>	$7.07e-2$	$6.44e-2$	$6.46e-2$	$8.64e-2$	$6.73e-2$	$5.51e-2$
	Time (s)	226.40	148.06	42.67	27.19	8.81	44.91
Baboon	<i>PSNR</i>	25.69	25.45	25.41	21.13	25.10	26.84
	<i>RE</i>	$9.63e-2$	$9.89e-2$	$9.94e-2$	$1.63e-1$	$1.03e-1$	$8.44e-2$
	Time (s)	133.15	72.92	28.91	23.38	8.30	25.54

## 7 Conclusion

The accelerated tensorial double proximal gradient algorithms described in this paper can be applied to other fields like image and video restoration. The applications illustrated in this paper are only a small portion of what these algorithms can tackle. The tensorial structure of the main problem offers the possibility of treating multidimensional data, in addition, the use of extrapolation techniques has improved significantly the convergence speed of the proposed algorithm.

**Acknowledgments** We would like to thank the anonymous referees for their recommendations and helpful remarks, which improved the quality of this

paper. The second author would like to thank the Moroccan Ministry of Higher Education, Scientific Research and Innovation and the OCP Foundation who support him through the APRD research program.

## References

1. H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz. *Fixed-point algorithms for inverse problems in science and engineering*, volume 49. Springer Science & Business Media, 2011.
2. H. H. Bauschke, P. L. Combettes, et al. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, volume 408. Springer, 2011.
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. O. Benchettou, A. H. Bentbib, and A. Bouhamidi. Tensorial total variation-based image and video restoration with optimized projection methods. *Optimization Methods and Software*, pages 1–32, 2022.
5. A. Bouhamidi, M. Bellalij, R. Enkhbat, K. Jbilou, and M. Raydan. Conditional gradient method for double-convex fractional programming matrix problems. *Journal of Optimization Theory and Applications*, 176(1):163–177, 2018.
6. A. Bouhamidi, K. Jbilou, L. Reichel, and H. Sadok. An extrapolated TSVD method for linear discrete ill-posed problems with Kronecker structure. *Linear Algebra and its Applications*, 434(7):1677–1688, 2011.
7. C. Brezinski and M. R. Zaglia. *Extrapolation methods: theory and practice*. Elsevier, 2013.
8. E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
9. J.-P. Chehab and M. Raydan. Geometrical inverse matrix approximation for least-squares problems and acceleration strategies. *Numerical Algorithms*, pages 1–19, 2020.

10. C. Chen, Z.-B. Wu, Z.-T. Chen, Z.-B. Zheng, and X.-J. Zhang. Auto-weighted robust low-rank tensor completion via tensor-train. *Information Sciences*, 567:100–115, 2021.
11. A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley & Sons, 2009.
12. P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.
13. P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.
14. J. Duran, B. Coll, and C. Sbert. Chambolle’s projection algorithm for total variation denoising. *Image processing on Line*, 3:311–331, 2013.
15. I. Ekeland and R. Temam. *Convex analysis and variational problems (studies in mathematics and its applications, vol. 1 north-holland, amsterdam, 1976)*.
16. A. El Ichi, K. Jbilou, and R. Sadaka. Tensor global extrapolation methods using the n-mode and the Einstein products. *Mathematics*, 8(8):1298, 2020.
17. S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.
18. P. Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *Image Processing On Line*, 2:74–95, 2012.
19. P. Getreuer. Total variation inpainting using split Bregman. *Image Processing On Line*, 2:147–157, 2012.
20. R. Glowinski. *Numerical methods for nonlinear variational problems*. Tata Institute of Fundamental Research, 1980.
21. A. Haddad. Stability in a class of variational methods. *Applied and Computational Harmonic Analysis*, 23(1):57–73, 2007.
22. K. Jbilou and A. Messaoudi. Block extrapolation methods with applications. *Applied Numerical Mathematics*, 106:154–164, 2016.
23. K. Jbilou, L. Reichel, and H. Sadok. Vector extrapolation enhanced TSVD for linear discrete ill-posed problems. *Numerical Algorithms*, 51(2):195–208, 2009.

24. K. Jbilou and H. Sadok. Vector extrapolation methods. applications and numerical comparison. *Journal of Computational and Applied Mathematics*, 122(1-2):149–165, 2000.
25. M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.
26. T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
27. C. W. Lee, K. Jung, and H. J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24(15):2607–2623, 2003.
28. N. Lee and A. Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, 29(3):921–960, 2018.
29. H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, pages 379–387, 2015.
30. M. Li and B. Li. A double fittings based total variation model for image denoising. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pages 467–471. IEEE, 2021.
31. X. Li, Y. Ye, and X. Xu. Low-rank tensor completion with total variation for visual data inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
32. J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2012.
33. W. López and M. Raydan. An acceleration scheme for Dykstra’s algorithm. *Computational Optimization and Applications*, 63(1):29–44, 2016.
34. C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2019.
35. M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on Image Processing*, 12(12):1579–1590, 2003.

36. U. Modha and P. Dave. Image inpainting-automatic detection and removal of text from images. *International Journal of Engineering Research and Applications (IJERA)*, ISSN, pages 2248–9622, 2014.
37. J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.
38. Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.
39. Z.-F. Pang, Y.-M. Zhou, T. Wu, and D.-J. Li. Image denoising via a new anisotropic total-variation-based model. *Signal Processing: Image Communication*, 74:140–152, 2019.
40. N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.
41. R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 2015.
42. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
43. A. Sidi. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *Journal of Computational and Applied Mathematics*, 36(3):305–337, 1991.
44. G. Sridevi and S. S. Kumar. Image inpainting based on fractional-order nonlinear diffusion for image reconstruction. *Circuits, Systems, and Signal Processing*, 38(8):3802–3817, 2019.
45. J. Van Leeuwen and J. Leeuwen. *Handbook of Theoretical Computer Science: Algorithms and Complexity*, volume 1. Elsevier, 1990.
46. T. Yokota, Q. Zhao, and A. Cichocki. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.
47. Z. Zhang and S. Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.
48. Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849, 2014.

- 
49. Q. Zhao, L. Zhang, and A. Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.