Expert Systems With Applications 232 (2023) 120848

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Using the ℓ_1-norm for Image-based tomographic reconstruction

José J. Calvino [b], Elena Fernández [a], Miguel López-Haro [b], Juan M. Muñoz-Ocaña [a,*], Antonio M. Rodríguez-Chía [a]

[a] *Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Campus Universitario, Puerto Real, (Cádiz), Spain*
[b] *Departamento de Ciencia de los Materiales e Ingeniería Metalúrgica y Química Inorgánica, Universidad de Cádiz, Campus Universitario, Puerto Real, (Cádiz), Spain*

ARTICLE INFO

Keywords:
Image reconstruction
Electron tomography
Linear programming
Lagrangian relaxation

ABSTRACT

This paper introduces an ℓ_1-norm model based on Total Variation Minimization for tomographic reconstruction. The reconstructions produced by the proposed model are more accurate than those obtained with classical reconstruction models based on the ℓ_2-norm. This model can be linearized and solved by linear programming techniques. Furthermore, the complementary slackness conditions can be exploited to reduce the dimension of the resulting formulation by removing unnecessary variables and constraints. Since the efficacy of the reduced formulation strongly depends on the quality of the dual-multipliers used when applying the reduction method, Lagrangian relaxation is used to obtain near-optimal multipliers. This allows solving larger instances in an efficient way.

1. Introduction

Nowadays, the rapid advances in Nanoscience and Nanotechnologies are providing new nanomaterials with a wide range of applications in environmental protection, new green energy sources, photonics and catalysis, among others. The usefulness of these new materials strongly depends on the ability of controlling their structure and morphology at the nanometer (nm) scale, given that many of their physical and chemical properties are highly dependent on the shapes exhibited by their components.

In this regard, Electron Tomography (ET), the subject of study of this paper, has become the primary imaging tool in Materials Science for the reconstruction of 3D materials measured at the *nm* scale, see Araújo et al. (2021), Midgley and Dunin-Borkowski (2009), Nicoletti et al. (2013), Thavavel et al. (2012) for alternatives image processing. ET is a well-known technique widely used for reconstructing 3D structures from tilt series of 2D-projections of a sample. This kind of reconstruction method provides an image of the nanomaterial under study from a set of projections recorded using an electron microscope. This set of projections of the nanomaterial, which are obtained from different tilt angles, is the input data of the reconstruction algorithms and it is known as sinogram. Together with Computerized Axial Tomography and Multidetector Computed Tomography, ET constitutes one of the main techniques used in the field of tomography. These techniques differ from each other in the way in which projections are obtained or the number of images that are taken simultaneously, and usually have different application areas.

Relevant applications of ET arise, for instance, in the study of the morphology of highly efficient 3D hybrid polymer solar cells, see, e.g. Oosterhout et al. (2009) (the analysis of this polymer is particularly relevant since it is used to replace silicon in solar cells because silicon is very expensive and polluting) or the ionomer Nafion studied by López-Haro, Guétaz, et al. (2014), used in the proton exchange membrane fuel cell (this material is very promising as a zero-emission power sources). Therefore, ET reconstructions are of great importance to analyse the structures of nanomaterials which are at *nm* scale, since these particles are no visible to humans. This technique allows us to recover the volumes of the nano-objects aiming at studying the morphology of these structures.

Experiments in ET are tilted around a single axis (or a dual axis) at incremental degrees of rotation, see Midgley et al. 2007 and Supplementary material S1.1 For each tilt, a projection is obtained when a set of parallel electron beams crosses the sample. Fig. 1 shows the process of recording different projections of a nanomaterial where $\theta = -70°$, $\theta = 0°$ and $\theta = 70°$ are different tilt angles. The recorded images are grey-scale images showing bright areas (high intensities), which correspond to the locations where the electron beam crosses the material, and dark areas, which correspond to the background (low intensities).

---

* Corresponding author.
E-mail addresses: jose.calvino@uca.es (J.J. Calvino), elena.fernandez@uca.es (E. Fernández), miguel.lopezharo@uca.es (M. López-Haro), juanmanuel.munoz@uca.es (J.M. Muñoz-Ocaña), antonio.rodriguezchia@uca.es (A.M. Rodríguez-Chía).

https://doi.org/10.1016/j.eswa.2023.120848
Received 18 May 2021; Received in revised form 13 May 2022; Accepted 12 June 2023

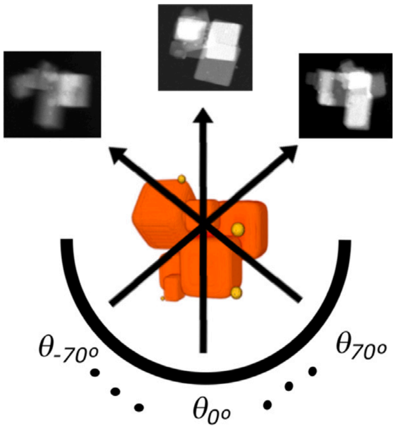

Fig. 1. Projections of a nanomaterial recorded from different tilt angles.

The most commonly reconstruction methods used in ET are based on algebraic approaches. These methods find a suitable solution to a system of equations $Ax = b$, where the unknown x denotes the intensity of each pixel in the reconstructed $n \times n$ image. A is the so-called projection matrix and b is the vector of the 2D-projection, recorded experimentally (also known as sinogram, see Section 2).

Taking into account the above observations, the three main difficulties in the reconstruction process in ET are the following. First, as described, most experiments cover just the [−70°, +70°] tilt range due to technical limitations. This leads to the so-called missing wedge effect, which produces distorted or elongated morphologies in the XZ direction where Y is the rotation axis; for example, a circle is reconstructed as an ellipse. Second, many materials can suffer severe morphological changes during the tilt series acquisition, due to beam damage (materials are exposed to high energy electron beams, e.g. 200 keV, while projections are recorded). To avoid this deleterious effect, the number of projections is often limited during the experiment, which may also result in a reduction of the reconstruction quality. And third, the dimension of matrix A is huge because it has n^2 columns and the number of rows is on the order of n, see Section 2 for further details, and n usually takes values in $\{256, 512, 1024\}$. Therefore, a difficulty stems from the very high dimension of the system $Ax = b$ to be solved. This notably reduces the alternatives as for the solution methods that can be effectively applied, to those able to handle the systems efficiently.

To overcome the above difficulties, the efforts in ET have focused not only at improving the reliability of the information recorded by microscopes, but also at decreasing the total number of projections required to reconstruct the volumes with sufficient accuracy, as well as at looking for effective solution methods for the optimization problem associated with each projection. In this regard, most recent methods are based on so-called Compressed Sensing. In particular, algorithms aiming at Total Variation Minimization (TVM) have proven to be very efficient (Goris et al., 2012; Li et al., 2013; Luo et al., 2018). These methods consider an objective function based on the sum of the ℓ_2-norm of the image gradient and the deviation of the reconstructed sinogram (Ax) with respect to the original sinogram (b), measured with the ℓ_2-norm. This approach leads to solve optimization problems with a quadratic objective function and linear constraints. The most efficient methods used nowadays for solving these problems are based on the augmented Lagrangian multiplier method (Li et al., 2013).

In this paper we consider a reconstruction optimization model based on the TVM paradigm but using the ℓ_1-norm, instead of the classical ℓ_2-norm. In the literature, we can find references to some applications where solution methods based on the use of the ℓ_1-norm outperform classical solution methods considering the ℓ_2-norm. Two examples of such approaches are (Fu et al., 2006) for image restoration problems, and Pi et al. (2021) for convex clustering problems. The proposed model has several advantages in comparison to the ℓ_2-version. As will be seen, the formulation using the ℓ_1-norm provides more accurate definition of the different structures that constitute the object under study and, therefore, they can be differentiated more efficiently in the reconstructions, providing better quality reconstructions, using a smaller number of projections. Furthermore, the use of the ℓ_1-norm allows us to formulate the problem as a linear program (LP), so linear programming tools can be used to obtain efficiently an optimal solution of the proposed model. This is important, taking into account the high dimensions of the instances that arise in practice, since the solution of the resulting linear programme with off-the-shelf solvers can be rather time-consuming for large images. In particular, we develop a solution method, which allows us to reconstruct such images in an efficient way. This procedure is based on the solution of a smaller formulation, obtained after reducing the dimension of the original model by removing unnecessary variables and deriving tighter expressions for some of the constraints. This elimination test exploits the complementary slackness conditions of linear programming, using a set of near-optimal dual variables. Still, in order to take advantage of this test, the dimensions of the original LPs advise for the use of decomposition methods. In our case we resort to Lagrangian relaxation to first obtain the values of the near-optimal dual variables. Lagrangian relaxation is a technique that has often been successfully applied to solve large-scale optimization problems of various types (Balasundaram & Kapil, 2010; Fisher, 1985; Knudsen et al., 2014).

We have applied the ℓ_1-norm in the proposed formulation for different reasons. First, as we can observe in the experiments carried out in this paper, the ℓ_1-norm formulation provides more accurate edges of thin structures and therefore, they can be differentiate more efficiently in the reconstructions. Second, the proposed formulation has shown a better performance achieving higher quality reconstructions using a smaller number of projections Moreover, the ℓ_1-norm has allowed us to formulate the proposed problem as a linear program using linear mathematical programming tools. We have to include one sentence in the manuscript where the advantage of using the ℓ_1-norm is explained (Page 4).

For our computational tests we have used well-known 2D and 3D phantoms with 512×512 and 1024×1024 pixels and an experiment using a real object. Throughout the text, we will refer as phantom to a perfectly known non-experimental image, which can be compared with the one obtained with any reconstruction method. The numerical results obtained in our computational experiments support our proposal, both from a modelling and an algorithmic point of view. In particular, the proposed ℓ_1-norm model provides reconstructions of better quality than those obtained with an ℓ_2-norm model. Algorithmically, the proposed solution algorithm is highly effective in comparison with both well-known algorithms usually applied to classical TVM models, and the solution of the proposed ℓ_1-norm formulation with off-the-shelf solvers. Actually, we show that the effectiveness of the proposed procedure can be enhanced substantially, because slightly less accurate reconstructions can be obtained, significantly reducing computing times. Finally, it should be highlighted that only 8 and 16 projections have been considered.

The main contributions of this paper are: (1) The use of TVM models using ℓ_1-norm, which outperform state-of-the-art reconstruction models, both, in terms of the quality of reconstructed nano-object using ET and with respect to the number of projections needed for the reconstruction (a remarkable advantage of using fewer projections is a reduction of the morphological damage of samples due to the electron beam); and (2) The development of an efficient solution algorithm for the resulting model, suitable for dealing with the very high dimension instances that arise in practice. Such algorithm is based on the formulation of the problem as a linear programming one. This allows us (i) to exploit linear programming properties, in particular,

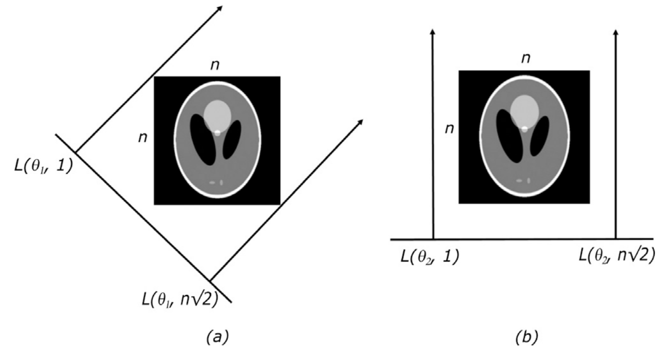

Fig. 2. Projections of $\bar{n}$ electron beams for tilt angles $\theta_1 = 45°$ (a) and $\theta_2 = 0°$ (b).

the complementary slackness conditions to reduce the dimension of the problem; and, (*ii*) to apply efficiently decomposition methods for dealing with large-scale linear programming models.

The paper is structured as follows. To make the paper self-contained, Supplementary Material S1.1 gives a short overview of the principles of electron tomography and Section 2 introduces the relevant notation and overviews the state-of-the-art on TVMs. In Section 3, we introduce the TVM model applying the ℓ_1-norm to the image gradient and to the deviation of the reconstructed sinogram (Ax) with respect to the experimental one (b) and present a linear formulation for it. In this section we also give a formulation for its dual, and state the corresponding primal/dual complementary slackness conditions. Then, in Section 4, we present the elimination test that allows us removing unnecessary variables and constraints leading to a reduced ℓ_1 reconstruction model, as well as a solution procedure to solve it. Section 5 develops the Lagrangian relaxation that we apply to obtain the near-optimal dual values that will be used in our implementation of the elimination test. In this section we also describe how to solve the Lagrangian dual using both subgradient optimization (Bertsekas, 1999; Shor, 1985) and the volume algorithm (Barahona & Anbil, 2000). Section 5.1 introduces some properties derived from the complementary slackness conditions together with other enhancements, which lead to remarkable improvements in computing times and in the quality of the reconstructed images. Section 6 summarizes and analyses the results of the computational experiments carried out. The paper closes in Section 7 providing the main conclusions.

2. State of the art of electron tomography

Since Johan Radon published the first attempt to solve tomography problems (Radon, 1917), a great effort has been made to develop new reconstruction methods. The first one is weighted back projection, which is based on Fourier Central Slice theorem, see Feldkamp et al. (1984) and Tang et al. (2006). The main objective of these reconstruction models is to estimate the intensities of each pixel as accurately as possible. Since reconstruction algorithms based on this type of models do not provide a high quality images, algebraic algorithms were developed.

Algebraic reconstruction methods try to calculate a solution for the system of equations $Ax = b$, where x is the image to be reconstructed, A a known matrix called projection matrix, and b the sinogram, i.e., the input data obtained from an experimental study with the electron microscope. Algebraic reconstruction methods usually consider a squared image with $n \times n$ pixels and $\bar{n} := n\sqrt{2}$ electron beams for each projection. This number of electron beams for each projection, $\bar{n}$, is chosen to allow obtaining information of the whole main diagonal of the image, which corresponds to the 45° tilted projection and represents the widest area to be covered, as shown in Fig. 2. Let p denote the number of considered projections, and $A = (A_1^T, \dots, A_p^T)^T$ be the aforementioned projection matrix, such that the columns represent the pixels of the image and, for a fixed tilt angle $\theta \in \Theta := \{1, \dots, p\}$, the kth row of A_θ, $A_{\theta,k} \in \mathbb{R}^{n^2}$, contains information about the pixels that are went through by the kth parallel electron beam with $k \in K := \{1, \dots, \bar{n}\}$, i.e., $A_\theta \in \mathbb{R}^{\bar{n} \times n^2}$. Since each column of A represents a specific pixel and an electron beam can cross very few pixels of the total image, most components in the projection matrix A take the value 0. Vector $b = (b_1, \dots, b_p)^T$, where $b_\theta \in \mathbb{R}^{\bar{n}}$, i.e., $b_{\theta,k} \in \mathbb{R}$ is the intensity reported by the electron microscope, of the kth electron beam from the projection with tilt angle $\theta \in \Theta$.

Note that the dimension of matrix A, is $(p\bar{n}) \times (n^2)$, which can be huge because usual values for the parameter n are $\{256, 512, 1024\}$. This has motivated the development of *ad hoc* models and solution methods for dealing with the system $Ax = b$, able to produce accurate images even for a small number of projections.

Traditionally, the system $Ax = b$ was solved with iterative reconstruction algorithms such as *Kaczmarz's algorithm*, see Kaczmarz (1937), renamed as the *Algebraic Reconstruction Technique* by Gordon et al. (1970). However, the most popular algebraic reconstruction model in the literature is the *Simultaneous Iterative Reconstruction Technique* which provides accurate reconstructions in low computing times, see Andersen and Kak (1984) for further details. Still, such models have some important disadvantages: they require a high number of projections to produce good reconstructions, and they are highly sensitive to the noise that appears when recording the experimental projected images with the electron microscope.

This has motivated the development of *ad hoc* models and solution methods for dealing with the system $Ax = b$, able to produce accurate images even for a small number of projections, and also able to overcome the above mentioned difficulties. This kind of models are called Compressed Sensing models, see Candes et al. (2006), Donoho (2006), and Lustig et al. (2007) for further information. The most popular Compressed Sensing reconstruction models are based on TVM. These optimization models remove a high level of noise in the reconstruction process by looking for a solution to system $Ax = b$ that minimizes the norm of the image gradient. The TVM models analysed in the literature usually consider the sum of the ℓ_2-norm of the image gradient as well as the deviation of the reconstructed sinogram (Ax) with respect to the original sinogram (b), measured with the ℓ_2-norm.

A TVM model can be expressed as:

$$\begin{array}{lll} \text{(TVM)} & \min_{x \in \mathbb{R}^{n^2}} & \sum_{i \in I} \|D_i x\|, \\ & \text{s.t.} & Ax = b, \end{array}$$

where $I := \{1, \dots, n^2\}$ denotes the index set of the x variables, $\|\cdot\|$ represents the norm used to measure the Total Variation of image x, and $D_i \in \mathbb{R}^{2 \times n^2}$ is used to calculate the image gradient at pixel i where $D_i x \in \mathbb{R}^2$. In order to compute this gradient, we define its horizontal and vertical components as $D_i = (H_i^T, V_i^T)^T$. Let $H \in \mathbb{R}^{n^2 \times n^2}$ be the matrix that determines the horizontal component of the gradient of x, which is defined by

$$H =: \begin{array}{c|cccccc} & 1 & 2 & 3 & \cdots & n^2-1 & n^2 \\ \hline H_1 & 1 & -1 & 0 & \cdots & 0 & 0 \\ H_2 & 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{n^2-1} & 0 & 0 & 0 & \cdots & 1 & -1 \\ H_{n^2} & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}. \tag{1}$$

Similarly, the vertical component of the image gradient is given by matrix $V \in \mathbb{R}^{n^2 \times n^2}$, defined by

$$V = \begin{array}{c} \\ V_1 \\ V_2 \\ \vdots \\ V_{n^2-n} \\ V_{n^2-n+1} \\ \vdots \\ V_{n^2-1} \\ V_{n^2} \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & \cdots & n & n+1 & \cdots & n^2-1 & n^2 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & \cdots & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \end{array}. \qquad (2)$$

The rows of matrices H and V are denoted by $H_i \in \mathbb{R}^{n^2}$ and $V_i \in \mathbb{R}^{n^2}$, respectively, for $i \in I$. Therefore, the image gradient at pixel i can be calculated as

$$D_i x = (H_i x, V_i x) = \begin{cases} (x_i - x_{i+1}, x_i - x_{i+n}), & \text{if } i \leq n^2 - n, \\ (x_i - x_{i+1}, 0), & \text{if } n^2 - n + 1 \leq i \leq n^2 - 1, \\ (0, 0), & \text{otherwise.} \end{cases}$$

Another important TVM model was proposed in Rudin et al. (1992). This model, which works more efficiently with noisy images, has a squared penalty term in the objective function instead of the set of constraints $Ax = b$, and it can be written in the following way:

$$(\text{TVM}_{ROF}) \quad \min_{x \in \mathbb{R}^{n^2}} \sum_{i \in I} \|D_i x\| + \frac{\mu}{2} \|Ax - b\|_2^2,$$

where $\mu > 0$ is a parameter that regulates the trade-off between the gradient of the image and the squared norm of the difference between the projection of the reconstructed image and the sinogram. In addition, the intensities of the images we reconstruct will be between 0 and 1, i.e., $x \in [0,1]^{n^2}$.

One of the best-known and widely used algorithms for solving the TVM and TVM_{ROF} models is the Total Variation minimization by Augmented Lagrangian and ALternating direction ALgorithm (TVAL3), which is based on the augmented Lagrangian multiplier method, see Li et al. (2013) for further details.

3. The ℓ_1-norm reconstruction model

Next we introduce a reconstruction model, which can be seen as a TVM model based on the ℓ_1-norm. The use of the ℓ_1-norm allows us to reformulate the resulting optimization problem as a linear program, so, after a linearization phase, linear programming tools can be used to solve it. As we will show, the resulting model has proven to be very effective. Specifically, the ℓ_1-norm model produces better quality reconstructions than the corresponding ℓ_2 version using the same number of projections. This is a very important aspect because, as mentioned in the Introduction, acquiring a large set of projections may damage the original particle. Therefore, sometimes only a limited number of projections are available from a particle. Moreover, a small number of projections also implies large savings in computing times in the experimental phase.

The ℓ_1-norm reconstruction model that we propose to ET experiments is:

$$(\text{TV}_{\ell_1}) \quad \min_{x \in [0,1]^{n^2}} \sum_{i \in I} \|D_i x\|_1 + \frac{\mu}{2} \|Ax - b\|_1,$$

where μ is a regularization parameter that controls the level of smoothness in the reconstruction. A small value of μ provides smooth reconstructions because the image gradient term, which controls its softness, becomes the most relevant term in the objective function. On the other hand, a large value of μ produces reconstructions with a high level of detail although noise is not removed from the recovered image. As will be explained in the Computational Experiments Section, we consider a wide range of μ values, from 2^0 to 2^{14}, in order to observe the effect made by this parameter.

First, we will linearize TV_{ℓ_1} so we can apply linear programming techniques to handle it. To this end, we introduce new non-negative variables that avoid the non-linearity of the ℓ_1-norm.

Since the differences $A_{\theta,k}x - b_{\theta,k}$ that affect the term $\|Ax - b\|_1$ can take positive or negative values, we will express each of these terms as the difference of two non-negative variables, $z^+_{\theta,k}$ and $z^-_{\theta,k}$, for $\theta \in \Theta$, $k \in K$. Therefore, these differences can be rewritten as $A_{\theta,k}x - b_{\theta,k} = z^+_{\theta,k} - z^-_{\theta,k}$, where $z^+_{\theta,k}, z^-_{\theta,k} \geq 0$ for $\theta \in \Theta$, $k \in K$. Accordingly, any optimal solution of TV_{ℓ_1}, satisfies that $|A_{\theta,k}x - b_{\theta,k}| = z^+_{\theta,k} + z^-_{\theta,k}$, where at most one of the two new non-negative variables will be non-zero because $z^+_{\theta,k} + z^-_{\theta,k}$ is part of the objective function that is minimized. Horizontal and vertical gradients can be decomposed similarly obtaining $x_i - x_{i+1} = h^+_i - h^-_i$ for $i \in I^h := \{1, \dots, n^2 - 1\}$ and $x_i - x_{i+n} = v^+_i - v^-_i$ for $i \in I^v := \{1, \dots, n^2 - n\}$ respectively, where h^+_i, h^-_i, v^+_i, $v^-_i \geq 0$. Applying this decomposition to TV_{ℓ_1}, the following linear TV problem is obtained:

$$(\text{LTV}_{\ell_1}) \quad \min \sum_{i \in I^h} (h^+_i + h^-_i) + \sum_{i \in I^v} (v^+_i + v^-_i) + \frac{\mu}{2} \sum_{\theta \in \Theta} \sum_{k \in K} (z^+_{\theta,k} + z^-_{\theta,k})$$

$$\begin{aligned} \text{s.t.} \quad & A_{\theta,k}x - b_{\theta,k} = z^+_{\theta,k} - z^-_{\theta,k}, && \theta \in \Theta,\ k \in K, && (3a) \\ & x_i - x_{i+1} = h^+_i - h^-_i, && i \in I^h, && (3b) \\ & x_i - x_{i+n} = v^+_i - v^-_i, && i \in I^v, && (3c) \\ & 0 \leq x_i \leq 1, && i \in I, && (3d) \\ & z^+_{\theta,k}, z^-_{\theta,k} \geq 0, && \theta \in \Theta,\ k \in K, && (3e) \\ & h^+_i, h^-_i \geq 0, && i \in I^h, && (3f) \\ & v^+_i, v^-_i \geq 0, && i \in I^v. && (3g) \end{aligned}$$

The main drawback of formulation LTV_{ℓ_1} is the large number of variables and constraints that it involves for usual size images. For instance, for a 1024 × 1024 image with 8 different projections, LTV_{ℓ_1} has 3,206,471 variables and 2,116,607 constraints. These dimensions involve large memory requirements to load the data instances as well as large computing times to solve them, which are unrealistic in practice. This motivates the use of some alternative solution method in which the burden due to the large dimension of the formulation is alleviated.

3.1. Complementary slackness conditions for LTV_{ℓ_1}

Below we analyse the complementary slackness conditions for LTV_{ℓ_1} and its associated dual problem, which will be exploited to develop an elimination technique leading to a (reduced) ℓ_1-reconstruction model of lower dimension.

Consider the dual variables $(\tau_{\theta,k})_{\theta \in \Theta, k \in K}$, $(\alpha_i)_{i \in I^h}$, $(\delta_i)_{i \in I^v}$, and $(\sigma_i)_{i \in I}$ associated with constraints (3a), (3b), (3c) and (3d), respectively. Let also $A_{\theta,k,i} \in \mathbb{R}$ denote the ith component of vector $A_{\theta,k}$. The linear dual of LTV_{ℓ_1} is:

$$(\text{D-LTV}_{\ell_1}) \max \sum_{\theta \in \Theta} \sum_{k \in K} b_{\theta,k} \cdot \tau_{\theta,k} - \sum_{i \in I} \sigma_i \qquad (4a)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{\theta \in \Theta} \sum_{k \in K} \tau_{\theta,k} A_{\theta,k,i} - \alpha_{i-1} + \alpha_i - \delta_{i-n} + \delta_i - \sigma_i \leq 0, \ i \in I, && (4b) \\ & -\tau_{\theta,k} \leq \frac{\mu}{2}, && \theta \in \Theta,\ k \in K, \quad (4c) \\ & \tau_{\theta,k} \leq \frac{\mu}{2}, && \theta \in \Theta,\ k \in K, \quad (4d) \\ & \alpha_i \leq 1, && i \in I^h, \quad (4e) \\ & -\alpha_i \leq 1, && i \in I^h, \quad (4f) \\ & \delta_i \leq 1, && i \in I^v, \quad (4g) \\ & -\delta_i \leq 1, && i \in I^v, \quad (4h) \end{aligned}$$

where slightly abusing notation we assume that variables are set to zero for the indices they are not defined. Observe that constraints (4c)–(4h), define in fact a range of values for the dual variables τ, α and δ. Indeed,

$$|\tau_{\theta,k}| \le \frac{\mu}{2}, \qquad \theta \in \Theta,\ k \in K, \tag{5a}$$

$$|\alpha_i| \le 1, \qquad i \in I^h, \tag{5b}$$

$$|\delta_i| \le 1, \qquad i \in I^v. \tag{5c}$$

Once the dual problem has been formulated, the complementary slackness conditions can be derived. Recall that these conditions characterize optimal solutions to pairs of primal/dual linear problems, see e.g. Vanderbei (2007). When applied to the primal/dual pair LTV$_{\ell_1}$ and D-LTV$_{\ell_1}$ the conditions that relate the slacks of primal constraints with the associated dual variables are:

$$z^+_{\theta,k}\left(\frac{\mu}{2} + \tau_{\theta,k}\right) = 0, \qquad \theta \in \Theta,\ k \in K, \tag{6a}$$

$$z^-_{\theta,k}\left(\frac{\mu}{2} - \tau_{\theta,k}\right) = 0, \qquad \theta \in \Theta,\ k \in K, \tag{6b}$$

$$h^+_i(1 - \alpha_i) = 0, \qquad i \in I^h, \tag{6c}$$

$$h^-_i(1 + \alpha_i) = 0, \qquad i \in I^h, \tag{6d}$$

$$v^+_i(1 - \delta_i) = 0, \qquad i \in I^v, \tag{6e}$$

$$v^-_i(1 + \delta_i) = 0, \qquad i \in I^v. \tag{6f}$$

Likewise, the conditions that relate the slacks of the dual constraints with the corresponding primal variables are:

$$\tau_{\theta,k}(A_{\theta,k}x - z^+_{\theta,k} + z^-_{\theta,k} - b_{\theta,k}) = 0, \qquad \theta \in \Theta,\ k \in K, \tag{7a}$$

$$\alpha_i(x_i - x_{i+1} - h^+_i + h^-_i) = 0, \qquad i \in I^h, \tag{7b}$$

$$\delta_i(x_i - x_{i+n} - v^+_i + v^-_i) = 0, \qquad i \in I^v. \tag{7c}$$

4. A solution algorithm for LTV$_{\ell_1}$

The high dimension of the resulting optimization problem associated with LTV$_{\ell_1}$ for the instances that arise in practice makes it to be rather time-consuming solving with off-the-shelf solvers. For this reason, in this section we develop an ad-hoc efficient solution method for dealing with the obtained large-scale linear programming problem. First, we present an elimination test based on the complementary slackness conditions of Section 3.1, which allows removing unnecessary variables and constraints, leading to a lower dimension reconstruction model.

4.1. The reduced ℓ_1 reconstruction model

The complementary slackness conditions of Section 3.1 can be used to fix the values of some of the primal variables, so tighter expressions of some constraints of the primal problem LTV$_{\ell_1}$ are obtained. We will focus on variables h^+, h^-, v^+ and v^-, which contain information about the image gradient, and the complementary slackness conditions (6c)–(6f) relating these variables with the dual variables α and δ associated with the primal constraints (3b)–(3c) where they appear. In particular, for a given index $i \in I^h$, the corresponding primal constraint (3b) together with the complementary slackness conditions (6c)–(6d) indicate that:

$$x_i - x_{i+1} = h^+_i - h^-_i, \qquad i \in I^h,$$

$$h^+_i(1 - \alpha_i) = 0, \qquad i \in I^h,$$

$$h^-_i(1 + \alpha_i) = 0, \qquad i \in I^h.$$

Therefore, in any optimal solution to the pair LTV$_{\ell_1}$/D-LTV$_{\ell_1}$, when both $|1 - \alpha_i| > 0$ and $|1 + \alpha_i| > 0$, then $h^+_i = -h^-_i = 0$, so the constraint (3b) reduces to $x_i = x_{i+1}$. Similarly, from the primal constraint (3c) and the complementary slackness conditions (6e)–(6f) we may conclude that in any optimal solution to the pair LTV$_{\ell_1}$/D-LTV$_{\ell_1}$, when both $|1 - \delta_i| > 0$ and $|1 + \delta_i| > 0$, then $v^+_i = -v^-_i = 0$, so the constraint (3c) reduces to $x_i = x_{i+n}$.

The above implications can be exploited to derive an elimination test based on any given vector of dual variables, $\overline{\alpha}$ and $\overline{\delta}$, using a threshold value $\varepsilon \in (0, 1)$ for considering as "non-zero" terms only those with an absolute value greater than the considered ϵ. The higher the value of ϵ, the higher precision is obtained since less variables h^+, h^-, v^+ and v^- are removed from LTV$_{\ell_1}$ formulation. Let us define H_ε and V_ε as the index sets of the pixels that verify that $|1 - \overline{\alpha}_i| > \varepsilon$ and $|1 + \overline{\alpha}_i| > \varepsilon$, as well as $|1 - \overline{\delta}_i| > \varepsilon$ and $|1 + \overline{\delta}_i| > \varepsilon$, respectively. That is, $H_\varepsilon = \{i \in I^h : |1 - \overline{\alpha}_i| > \varepsilon \text{ and } |1 + \overline{\alpha}_i| > \varepsilon\}$ and $V_\varepsilon = \{i \in I^v : |1 - \overline{\delta}_i| > \varepsilon \text{ and } |1 + \overline{\delta}_i| > \varepsilon\}$. Then, we fix $h^+_i = h^-_i = 0$ for all $i \in H_\varepsilon$ and update constraints (3b) accordingly. Similarly, we fix $v^+_i = v^-_i = 0$ for all $i \in V_\varepsilon$ and update constraints (3c) accordingly. After the above elimination, we obtain a reduced formulation of LTV$_{\ell_1}$ given by:

$$(\text{R-LTV}_{\ell_1}) \quad \min \quad \frac{\mu}{2}\sum_{\theta\in\Theta}\sum_{k\in K}(z^+_{\theta,k} + z^-_{\theta,k}) + \sum_{i\in I^v}(h^+_i + h^-_i + v^+_i + v^-_i) + \sum_{i\in I^h\setminus I^v}(h^+_i + h^-_i) \tag{8a}$$

$$\text{s.t.} \quad A_{\theta,k}x - b_{\theta,k} = z^+_{\theta,k} - z^-_{\theta,k}, \qquad \theta \in \Theta,\ k \in K, \tag{8b}$$

$$x_i = x_{i+1}, \qquad i \in H_\varepsilon, \tag{8c}$$

$$x_i - x_{i+1} = h^+_i - h^-_i, \qquad i \in I^H \setminus H_\varepsilon, \tag{8d}$$

$$x_i = x_{i+n}, \qquad i \in V_\varepsilon, \tag{8e}$$

$$x_i - x_{i+n} = v^+_i - v^-_i, \qquad i \in I^v \setminus V_\varepsilon, \tag{8f}$$

$$z^+_{\theta,k}, z^-_{\theta,k} \ge 0, \qquad \theta \in \Theta,\ k \in K, \tag{8g}$$

$$0 \le x_i \le 1, \qquad i \in I, \tag{8h}$$

$$h^+_i, h^-_i \ge 0, \qquad i \in I^h \setminus H_\varepsilon, \tag{8i}$$

$$v^+_i, v^-_i \ge 0, \qquad i \in I^v \setminus V_\varepsilon. \tag{8j}$$

4.2. The solution algorithm

In this subsection, we present a solution algorithm for LTV_{ℓ_1}. It is based on the solution of the reduced formulation R-LTV$_{\ell_1}$, which, in turn, is obtained from the application of the elimination test derived from the complementary slackness conditions. The overall solution method is presented in Algorithm 1.

Indeed Algorithm 1 can be highly sensitive to the choice of the dual vector $\overline{\lambda} = (\overline{\tau}, \overline{\alpha}, \overline{\delta}, \overline{\sigma})$. In particular, $\overline{\alpha}$ and $\overline{\delta}$ dictate the outcome of the elimination test, which, in its turn, determines the reduced formulation R-LTV$_{\ell_1}$ that is actually solved in Step 3. On the one hand, using values of $\overline{\alpha}$ and $\overline{\delta}$ that are not *close enough* to the optimal ones can be useless, as it may produce a reduced model R-LTV$_{\ell_1}$ that does not approximate accurately the original formulation LTV$_{\ell_1}$. On the other hand, finding an optimal dual vector can be too much time consuming. Preliminary computational experiments indicate that a very good tradeoff can be achieved by using a near-optimal dual vector obtained from an approximate solution to the Lagrangian dual problem that we develop in the next section.

Algorithm 1: ℓ_1-norm reconstruction algorithm.

```
INPUT: Sinogram $b$ provided by electron microscopes,
    projection matrix $A$.
Step 1: Obtain a suitable dual vector $\overline{\lambda} = (\overline{\tau}, \overline{\alpha}, \overline{\delta}, \overline{\sigma})$.
Step 2: Set an accuracy $\varepsilon \in (0, 1)$ for the elimination test.
for $i = 1 : n^2$ do
    if $|1 - \overline{\alpha}_i| > \varepsilon$ and $|1 + \overline{\alpha}_i| > \varepsilon$ then
        Replace constraint $x_i - x_{i+1} = h_i^+ - h_i^-$ by $x_i = x_{i+1}$.
    if $|1 - \overline{\delta}_i| > \varepsilon$ and $|1 + \overline{\delta}_i| > \varepsilon$ then
        Replace constraint $x_i - x_{i+n} = h_i^+ - h_i^-$ by $x_i = x_{i+n}$.
Step 3: Solve the reduced reconstruction formulation R-LTV$_{\ell_1}$
    obtained in Step 2.
OUTPUT: Reconstruction $x$ of the particle put into the
    microscope.
```

5. Lagrangian relaxation for LTV$_{\ell_1}$

Consider the Lagrangian relaxation of LTV$_{\ell_1}$ resulting from removing constraints (3a)–(3c) and incorporating them to the objective function weighted with suitable multipliers. Note that the constraints that we relax are the ones involving the linearizing variables z^+, z^-, h^+, h^-, v^+ and v^-. Let $\lambda = (\tau, \alpha, \delta) \in \mathbb{R}^{p\bar{n}+|I^h|+|I^v|}$ denote the vector of Lagrangian multipliers associated with the relaxed constraints, where τ, α and δ correspond to constraints (3a), (3b), and (3c), respectively. Observe that these multipliers are not restricted in sign, because they are associated with equality constraints. The Lagrangian function is:

$$
\begin{aligned}
(\text{LTV-RL}_\lambda) \quad Z(\lambda) = \min \quad & L(\lambda, \mathbf{y}) \\
& \mathbf{y} \geq 0, \\
& x \leq 1,
\end{aligned}
$$

where $\mathbf{y} = (x, z^+, z^-, h^+, h^-, v^+, v^-)$ and $L(\lambda, \mathbf{y})$ is given by:

$$
\begin{aligned}
L(\lambda, \mathbf{y}) = & \sum_{\theta \in \Theta} \sum_{k \in K} \Big(\frac{\mu}{2}(z^+_{\theta,k} + z^-_{\theta,k}) + \tau_{\theta,k}(A_{\theta,k}x - b_{\theta,k} - z^+_{\theta,k} + z^-_{\theta,k}) \Big) \\
& + \sum_{i \in I^v} \Big((h_i^+ + h_i^- + v_i^+ + v_i^-) + \alpha_i(x_i - x_{i+1} - h_i^+ + h_i^-) \\
& \quad + \delta_i(x_i - x_{i+n} - v_i^+ + v_i^-) \Big) \\
& + \sum_{i \in I^h \setminus I^v} \Big((h_i^+ + h_i^-) + \alpha_i(x_i - x_{i+1} - h_i^+ + h_i^-) \Big),
\end{aligned}
$$

which, after regrouping terms, can be rewritten as:

$$
\begin{aligned}
L(\lambda, \mathbf{y}) = & \sum_{\theta \in \Theta} \sum_{k \in K} \Big(z^+_{\theta,k}(\frac{\mu}{2} - \tau_{\theta,k}) + z^-_{\theta,k}(\frac{\mu}{2} + \tau_{\theta,k}) \Big) \\
& + \sum_{i \in I^v} \Big(x_i(\alpha_i + \delta_i + \sum_{\theta \in \Theta} \sum_{k \in K} \tau_{\theta,k} A_{\theta,k,i}) \\
& + x_{i+1}(-\alpha_i) + x_{i+n}(-\delta_i) + h_i^+(1 - \alpha_i) + h_i^-(1 + \alpha_i) \\
& + v_i^+(1 - \delta_i) + v_i^-(1 + \delta_i) \Big) \\
& + \sum_{i \in I^h \setminus I^v} \Big(x_i(\alpha_i + \sum_{\theta \in \Theta} \sum_{k \in K} \tau_{\theta,k} A_{\theta,k,i}) + x_{i+1}(-\alpha_i) \\
& + h_i^+(1 - \alpha_i) + h_i^-(1 + \alpha_i) \Big),
\end{aligned}
\tag{9}
$$

where $A_{\theta,k,i}$ corresponds to the proportion of pixel i went through electron beam k in projection θ. The Lagrangian Dual is therefore:

$$
\begin{aligned}
(\text{LTV-DL}) \qquad \max \quad & Z(\lambda) \\
& \lambda \in \mathbb{R}^{p\bar{n}+|I^h|+|I^v|}.
\end{aligned}
$$

Since LTV$_{\ell_1}$ is a linear programming problem, it is well-known (see, for instance, Bertsimas & Tsitsiklis, 1997, for further details) that the optimal value of the Lagrangian dual LTV-DL coincides with the optimal value of the linear programming dual of LTV$_{\ell_1}$, which, in its turn, coincides with the optimal value of LTV$_{\ell_1}$, and any optimal vector of Lagrangian multipliers will be optimal for the linear dual problem D-LTV$_{\ell_1}$ as well.

We consider two alternative methods for solving the Lagrangian dual LTV-DL: an *ad hoc* algorithm based on subgradient optimization (Bertsekas, 1999; Shor, 1985), as well as a volume algorithm (Barahona & Anbil, 2000). For the sake of completeness we first describe a basic subgradient method and then the volume algorithm. Subgradient optimization is an iterative technique widely used for solving Lagrangian duals. The algorithm starts by choosing an initial multipliers vector $\lambda^0 = (\tau^0, \alpha^0, \delta^0) \in \mathbb{R}^{p\bar{n}+|I^h|+|I^v|}$. At iteration s, the Lagrangian function associated with the current multipliers vector is solved. That is, a vector $\mathbf{y}^s = (x^s, z^{s+}, z^{s-}, h^{s+}, h^{s-}, v^{s+}, v^{s-})$ is identified such that $Z(\lambda^s) = L(\lambda^s, \mathbf{y}^s)$. Then, the Lagrangian multipliers are updated according to:

$$
\lambda^{s+1} := \lambda^s + \nu^s \nabla Z(\lambda^s), \tag{10}
$$

where $\nabla Z(\lambda^s)$ is a subgradient of the objective function of problem LTV-RL$_\lambda$, i.e.,

$$
\begin{aligned}
\nabla_{\tau_{\theta,k}} Z(\lambda^s) &= A_{\theta,k}x^s - b_{\theta,k} - {z^s}^+_{\theta,k} + {z^s}^-_{\theta,k}, \quad && \theta \in \Theta,\ k \in K, \\
\nabla_{\alpha_i} Z(\lambda^s) &= {x^s}_i - {x^s}_{i+1} - {h^s}^+_i + {h^s}^-_i, && i \in I^h, \\
\nabla_{\delta_i} Z(\lambda^s) &= {x^s}_i - {x^s}_{i+n} - {v^s}^+_i + {v^s}^-_i, && i \in I^v,
\end{aligned}
$$

and ν^s is the step size along the normalized subgradient direction. A popular choice for ν^s is the formula given in Held et al. (1974):

$$
\nu^s = \frac{\rho^s(Z_{UB} - Z(\lambda^s))}{\|\nabla Z(\lambda^s)\|^2}, \tag{11}
$$

where Z_{UB} is the optimal objective value of the original problem LTV$_{\ell_1}$ or an upper bound of this value, and the parameter ρ^s must take values between 0 and 2 and can be updated according to different rules. The procedure is repeated until a stopping criterion is met, which usually includes a limit for the maximum number of iterations as well as some sort of convergence test, i.e. the variation in the value of the Lagrangian function is beneath a prefixed threshold for a number of consecutive iterations. Algorithm 2 gives a template of a subgradient optimization algorithm applied to D-LTV$_{\ell_1}$.

Algorithm 2: Template of subgradient optimization adapted to D-LTV$_{\ell_1}$.

```
INPUT: $\lambda^1 = (\tau^1, \alpha^1, \delta^1)$, $Z_{UB}$, $\rho^1 \in [0, 2]$,
    Termination-criterion, end:=false, $s := 1$.
while (not end) do
    Solve $Z(\lambda^s)$ and identify $\mathbf{y}^s = (x^s, z^{s+}, z^{s-}, h^{s+}, h^{s-}, v^{s+}, v^{s-})$
        such that $Z(\lambda^s) = L(\lambda^s, \mathbf{y}^s)$.
    Update the multipliers: $\lambda^{s+1} := \lambda^s + \nu^s \nabla Z(\lambda^s)$
    if (Termination-criterion) then
        end:=true
    $s := s + 1$.
Output: Near-optimal multipliers of problem D-LTV$_{\ell_1}$, $\lambda^s$.
```

It is well-known that one of the weaknesses of subgradient optimization can be its slow convergence, which, to a large extent, is due to the fact that it is not guaranteed that the value of the Lagrangian function improves in two consecutive iterations, i.e. it is not guaranteed that $Z(\lambda^{s+1}) > Z(\lambda^s)$. The volume algorithm was proposed in Barahona and Anbil (2000) as an alternative to subgradient optimization that guarantees that such condition holds. This strategy provides a higher speed of convergence in fewer iterations, which is a very attractive characteristic of any iterative method.

In the classical subgradient optimization algorithm applied to D-LTV$_{\ell_1}$, the movement direction at each iteration is the subgradient $\nabla Z(\lambda^s)$, whose components are the slacks of the relaxed constraints

(3a)–(3c), evaluated at the points $\mathbf{y}^s = (x^s, z^{s+}, z^{s-}, h^{s+}, h^{s-}, \upsilon^{s+}, \upsilon^{s-})$ such that $Z(\lambda^s) = L(\lambda^s, \mathbf{y}^s)$; then the step length is computed according to (11). Instead, in the volume algorithm a series of inner iterations are carried out within each major iteration of subgradient optimization in order to identify a move direction $\overline{\nabla} Z$ and associated step length $\bar{\nu}$ such that $Z(\lambda^s + \bar{\nu}\overline{\nabla} Z) > Z(\lambda^s)$. When such a direction and step length are found, then the multipliers vector λ^s is updated to $\lambda^s + \bar{\nu}\overline{\nabla} Z$ and the *regular* iterative procedure continues. Quite similarly to subgradient optimization, the components of $\overline{\nabla} Z$ are the slacks of the relaxed constraints (3a)–(3c), only that evaluated at a tentative point $\bar{\mathbf{y}} = (\bar{x}, \bar{z}^+, \bar{z}^-, \bar{h}^+, \bar{h}^-, \bar{\upsilon}^+, \bar{\upsilon}^-)$ defined as a convex combination of the current point $\mathbf{y}^s$ and the tentative point computed in the previous inner iteration. That is, $\bar{\mathbf{y}} := \pi\mathbf{y}^s + (1-\pi)\bar{\mathbf{y}}$, with $0 \le \pi \le 1$. Therefore, the components of the movement direction $\overline{\nabla} Z$ associated with a given tentative point $\bar{\mathbf{y}}$ are:

$$\overline{\nabla}_{\tau_{\theta,k}} Z = A_{\theta,k}\bar{x} - b_{\theta,k} - \bar{z}^+_{\theta,k} + \bar{z}^-_{\theta,k}, \qquad \theta \in \Theta,\ k \in K,$$

$$\overline{\nabla}_{\alpha_i} Z = \bar{x}_i - \bar{x}_{i+1} - \bar{h}^+_i + \bar{h}^-_i, \qquad i \in I^h,$$

$$\overline{\nabla}_{\delta_i} Z = \bar{x}_i - \bar{x}_{i+n} - \bar{\upsilon}^+_i + \bar{\upsilon}^-_i, \qquad i \in I^\upsilon.$$

Once $\overline{\nabla} Z$ has been calculated, its associated step length $\bar{\nu}$ is computed according to (11) using $\|\overline{\nabla} Z\|^2$ in the denominator. Algorithm 3 gives a template of a volume algorithm applied to D-LTV$_{\ell_1}$. Note that the first iteration of Algorithm 3 is just a subgradient optimization iteration because at that stage no tentative point has yet been defined.

Algorithm 3: Volume algorithm applied to D-LTV$_{\ell_1}$.

```
INPUT: $\lambda^1 = (\tau^1, \alpha^1, \delta^1)$, $Z_{UB}$, $\rho^1 \in [0,2]$,
  Termination-criterion, end:=false, $s := 1$.
while (not end) do
    Solve $Z(\lambda^s)$ and identify $y^s = (x^s, z^{s+}, z^{s-}, h^{s+}, h^{s-}, \upsilon^{s+}, \upsilon^{s-})$
      such that $Z(\lambda^s) = L(\lambda^s, \mathbf{y}^s)$.
    $\bar{\lambda} := \lambda^s + \nu^s \nabla Z(\lambda^s)$
    if ($s = 1$) then
        $\bar{\mathbf{y}} := \mathbf{y}^s$
    else
        while ($Z(\bar{\lambda}) \le Z(\lambda^{s-1})$) do
            $\bar{\mathbf{y}} := \pi\mathbf{y}^s + (1-\pi)\bar{\mathbf{y}}$.
            Compute $\overline{\nabla} Z$.
            Compute $\bar{\nu}$ according to (11) using $\|\overline{\nabla} Z\|^2$ in the
              denominator. $\bar{\lambda} := \lambda^s + \bar{\nu}\overline{\nabla} Z$
    Update the multipliers: $\lambda^{s+1} := \bar{\lambda}$
    if (some termination criterion is met) then
        end:=true
    $s := s + 1$.
Output: Near-optimal multipliers of problem D-LTV$_{\ell_1}$, $\lambda^s$.
```

5.1. Enhancing the subgradient and volume algorithms

As we will see in Section 6 where we summarize the numerical results from our computational experiments, it is possible to improve considerably the performance of the subgradient and volume algorithms by exploiting in several ways properties derived from linear programming and the complementary slackness conditions discussed in Section 3, as we explain below. The importance of these improvements can be appreciated in Tables S1–S3 where we can observe that for medium-size and large instances our procedure is effective only if these improvements are used to assist in finding the right multipliers.

1. An important aspect refers to how to select the optimal solution $\mathbf{y}^s = (h^{s+}, h^{s-}, \upsilon^{s+}, \upsilon^{s-})$ at each iteration of Algorithms 2 and 3. Note that, at a given iteration, $Z(\lambda^s)$ may have alternative optimal solutions, and a suitable choice of the selected solution may affect considerably the performance of the algorithms. In particular, looking at the expression of the Lagrangian function (9) it can be observed that for $i \in I^h$, when the coefficients $1 - \alpha_i = 0$, then h^{s+}_i can take any value. A similar observation applies to variables h^{s-}_i when their coefficients $1 + \alpha_i = 0$, $i \in I^h$, as well as to variables υ^+_i and υ^-_i, $i \in I^\upsilon$, when their respective coefficients $1 - \delta_i$ and $1 + \delta_i$ take a value of zero. In such cases, the complementary slackness conditions (7b) and (7c) can be used again to select suitable values for the involved variables. Specifically, we apply the following rules:

$$\begin{array}{llll}
1 - \alpha_i = 0 \Rightarrow & h^+_i = x_i - x_{i+1}; & h^-_i = 0, & i \in I^h, \\
1 + \alpha_i = 0 \Rightarrow & h^+_i = 0; & h^-_i = -(x_i - x_{i+1}) & i \in I^h, \\
1 - \delta_i = 0 \Rightarrow & \upsilon^+_i = x_i - x_{i+n}; & \upsilon^-_i = 0 & i \in I^\upsilon, \\
1 + \delta_i = 0 \Rightarrow & \upsilon^+_i = 0; & \upsilon^-_i = -(x_i - x_{i+n}) & i \in I^\upsilon.
\end{array}$$

   This modification enhances the feasibility with respect to the original problem of solution $\mathbf{y}^s = (h^{s+}, h^{s-}, \upsilon^{s+}, \upsilon^{s-})$.

2. Another enhancement of Algorithms 2 and 3 refers to setting a constrained domain for the Lagrangian dual multipliers $\lambda \in \mathbb{R}^{p\bar{n}+|I^h|+|I^\upsilon|}$. We can do this by taking into account the dual constraints (5a)–(5c) at each iteration of the volume algorithm, so the values of the multipliers are projected as follows:

$$\tau^{s+1} = \begin{cases} \tau^s + \nu^s \nabla_\tau Z(\lambda^s), & \text{if } |\tau^s + \nu^s \nabla_\tau Z(\lambda^s)| < \frac{\mu}{2}, \\ \frac{\mu}{2}, & \text{otherwise,} \end{cases} \tag{12a}$$

$$\alpha^{s+1} = \begin{cases} \alpha^s + \nu^s \nabla_\alpha Z(\lambda^s), & \text{if } |\alpha^s + \nu^s \nabla_\alpha Z(\lambda^s)| < 1, \\ 1, & \text{otherwise,} \end{cases} \tag{12b}$$

$$\delta^{s+1} = \begin{cases} \delta^s + \nu^s \nabla_\delta Z(\lambda^s), & \text{if } |\delta^s + \nu^s \nabla_\delta Z(\lambda^s)| < 1, \\ 1, & \text{otherwise.} \end{cases} \tag{12c}$$

3. Finally, the empirical performance of Algorithms 2 and 3 can be very sensitive to the initial choice of the Lagrange multipliers. Choosing them equal to zero may require a large number of iterations to obtain a tight lower bound of LTV$_{\ell_1}$. In order to obtain a better choice of the initial multipliers, we first solve problem LTV$_{\ell_1}$ for a smaller-size instance obtained from a courser version of the original image. That is, we use the same original image but coded with a smaller number of pixels. For instance, if the original image size is 1024 × 1024 pixels, then it is resized to an image with 256 × 256 pixels. This reduction in the dimension allows us to optimally solve LTV$_{\ell_1}$ for the resized image, and to obtain optimal values for the dual variables of the small dimension instance in a short computing time. Then, an estimation of a dual vector for the original size instance is calculated by extrapolating the optimal dual solution of the resized instance. This estimation is used as the initial vector of Lagrangian multipliers in both algorithms. Initializing the Lagrangian multipliers according to this procedure allows us to solve D-LTV$_{\ell_1}$ in considerably fewer iterations than with any naive initialization choice.

As we will see, incorporating the above enhancements to Algorithms 2 and 3 improves remarkably their performance and produces a good estimation of the optimal values of the dual variables α and δ of D-LTV$_{\ell_1}$. In fact, in order to obtain a good approximation of the optimal dual values it is not necessary to run these algorithms until their convergence and it is enough to do a fixed number of iterations. Preliminary computational experiments confirm the effectiveness of this alternative, and show that it does not affect noticeably the quality of the results of the overall process.

Table 1
Reconstruction errors without/with the improvements of Section 5.1 for 512 × 512 images.

Phantom	μ	TVM_{ROF}	LTV_{ℓ_1}	R-LTV_{ℓ_1} $(\varepsilon = 0.9375)$			
		Error TVM_{ROF}	Error LTV_{ℓ_1}	Error using Subgradient		Error using Volume	
				Unimproved	Improved	Unimproved	Improved
Catalyst 20°	2^7	2.9506	0.0026	35.0342	1.0452	155.7231	0.6765
	2^{10}	2.4749	0.0759	55.4694	0.1300	40.4247	0.1278
Circle 20°	2^4	10.1022	0.0026	29.6402	1.4609	214.6287	1.5052
	2^8	8.5764	0.0102	25.8016	0.5927	33.0813	0.4105
Shepp–Logan 20°	2^7	38.7823	33.9012	43.3303	34.5340	61.1401	34.0633
	2^{11}	42.0587	33.5665	66.9059	33.6400	38.0184	34.0079
Catalyst 10°	2^1	8.5064	0.0013	156.7208	0.0014	155.4036	0.1159
	2^9	0.2928	0.0875	26.5637	0.7744	23.8076	0.8510
Circle 10°	2^6	1.3590	0.0012	12.4856	0.0012	51.4175	0.0012
	2^{10}	0.2667	0.0065	27.1065	0.1218	4.0835	0.0068
Shepp–Logan 10°	2^3	10.8441	0.0024	25.0873	0.1051	94.7652	0.1044
	2^{10}	8.9705	0.8287	46.3489	0.9723	8.3319	1.6694

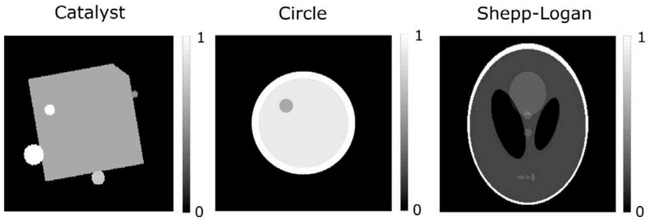

Fig. 3. 2D phantoms used to apply the proposed model.

6. Computational experiments

In this section, we compare the performance of the most usual reconstruction model, TVM_{ROF} with ℓ_2-norm solved using TVAL3 algorithm, with that of the TV_{ℓ_1} model proposed in this paper, solved with the methodology we have developed. 2D images of 512 × 512 and 1024 × 1024 pixels sizes with known features (intensities of the pixels, sizes and shapes of the particles or images with edges) have been used, such as Catalyst (López-Haro et al., 2018), Circle (Durst et al., 2014; López-Haro, Dubau, et al., 2014), and the well-known Shepp–Logan (Shepp & Logan, 1974; Tovey et al., 2019), see Fig. 3. The Shepp–Logan phantom is a standard test image created by Larry Shepp and Benjamin F. Logan, which serves as the model of a human head in the development and testing of image reconstruction algorithms. The Catalyst and Circle phantoms have been used to simulate square and circular shaped particles, respectively, because most of the objects studied in ET have such structures. We also analyse the results for a 3D particle (López-Haro et al., 2018) and for a real experiment. These last two objects are used to test the effectiveness of the models for reconstructing thin layers covering a given object. We show that the images obtained with the reconstructions produced by our ℓ_1 model are of better quality than those obtained from the ℓ_2 model. Moreover, it should be highlighted that only 8 and 16 projections have been considered, tilted from −70° to 70°. Furthermore, we show that the effectiveness of the procedure proposed in Algorithm 1, considering the values for dual variables obtained from Algorithm 2 and 3 with the improvements described in Section 5.1, substantially reduces the computing time of the original TV_{ℓ_1}, at the expense of obtaining only slightly less accurate reconstructions.

For the implementation of the TVAL3 routine, used in the solution of TVM_{ROF}, we have employed the ASTRA tomography MATLAB R2018a toolbox (http://www.astra-toolbox.com/index.html). LTV_{ℓ_1} has also been implemented in MATLAB R2018a thanks to the API that links Cplex and MATLAB, and solved using the Barrier algorithm. In all the implementations of Algorithms 2 and 3, the initial value of parameter ρ was fixed to 2. All the experiments have been performed on an Intel Core I7 8700 workstation, 64 Gb RAM, VGA NVIDIA RTX 2070.

Two projection sets have been chosen for each analysed image, both ranging from −70° to 70°. In the first one a projection is recorded every 20°, whereas in the second one a projection is recorded every 10°. Hence, in the reconstruction algorithms the initial data that constitutes the sinogram consists of a collection of 8 and 16 projections, respectively.

For comparing the different reconstruction methods, we define the *error* associated with each method as the ℓ_2-norm of the difference at each pixel between the original image (P) and the reconstructed one with the considered method (Rec). Furthermore, for our experiments, formulation LTV_{ℓ_1} was first solved and the computing time needed to solve it used as the stopping criterion for TVAL3 applied to TVM_{ROF}. This allows us to compare the quality of the images obtained for both procedures in a fair way (using the same computing time). It is worth mentioning that, in general, TVAL3 provides a solution in less time using the default stopping criteria, so by extending the limit computing time, we give TVAL3 the opportunity of further improving such a solution.

We start our analysis by highlighting the huge influence of the improvements described in Section 5.1 in the performance of Algorithms 2 and 3, which is summarized in Table 1 (see Tables S1–S3 in the Supplementary Material for more details). The first two columns of this table report the name of the reconstructed phantom and the value of parameter μ, respectively. The two values of μ reported for each instance are the ones that produced the best results for TVM_{ROF} and LTV_{ℓ_1}, respectively. The third and fourth columns give the errors of the solutions produced by TVM_{ROF} and LTV_{ℓ_1}, respectively. The remaining columns show the errors of the solutions obtained with R-LTV_{ℓ_1}, when the reduction procedure of Section 4.1 was applied with $\varepsilon = 0.9375$. Columns 5–6 report the results when the near-optimal dual solution was produced by Algorithm 2 and columns 7–8 when Algorithm 3 was used instead. As can be seen, the quality of the images reconstructed with Algorithm 1 using the multipliers provided by Algorithms 2 and 3 without improvements are substantially worse than those obtained using the improvements. In particular, the errors with the improvements are more than one order of magnitude smaller. For this reason, in all the experiments that we report in the following, Algorithms 2 and 3 are always enhanced with the improvements of Section 5.1. Furthermore, the reduced formulations obtained after applying the elimination procedure of Section 4.1 with a fixed value of ε and the near-optimal dual solutions produced by the enhanced Algorithms 2 and 3 will be referred to as, R-LTV_{ℓ_1}-2 and R-LTV_{ℓ_1}-3, respectively.

As recommended in Li et al. (2013), for the reconstruction of the images shown in Fig. 3 we have tested multiple values of parameter μ, ranging from $\mu = 2^0$ to $\mu = 2^{14}$, which allows to appreciate the influence of this parameter in the considered models. The smaller the μ value, the

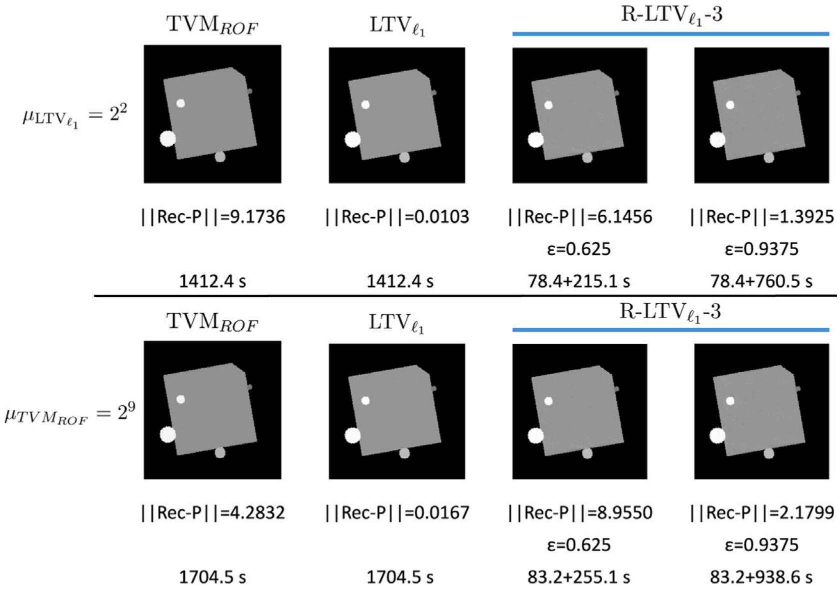

Fig. 4. Catalyst reconstruction with 1024 × 1024 pixels.

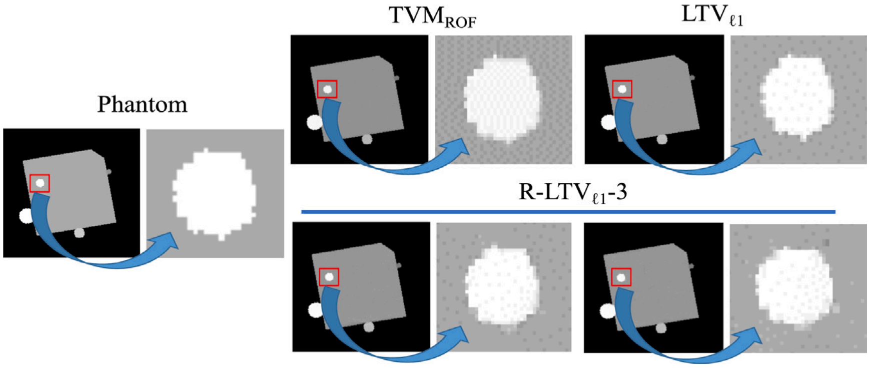

Fig. 5. Enlargements of the areas marked by the red squares.

smoother the reconstruction is provided. For these experiments, after preliminary testing, $\varepsilon = 0.625$ (10/16) and $\varepsilon = 0.9375$ (15/16) were chosen for the elimination test of Section 4.1 with projections every 20°. These values were chosen to illustrate how values of ε close to 1 affect the trade-off between accuracy and computing time.

The first reconstructed phantom simulates a catalyst with a square shape that contains round particles of different sizes and intensities. With this phantom we aim at illustrating how straight shapes and small particles are recovered with the different reconstruction models (López-Haro et al., 2018). Fig. 4 shows the results obtained with the 1024 × 1024 catalyst with projections ranged from −70° to 70° recorded every 20° using the following two values of μ: the one corresponding to the best reconstruction for TVM_{ROF} ($\mu_{\mathrm{TVM}_{ROF}}$); and the one corresponding to the best reconstruction obtained with LTV_{ℓ_1} ($\mu_{\mathrm{LTV}_{\ell_1}}$). Table S6 in the Supplementary material gives the detailed results of TVM_{ROF}, LTV_{ℓ_1}, R-LTV_{ℓ_1}-2 and R-LTV_{ℓ_1}-3 for the different values of μ with this instance. In Fig. 4 the computing time of R-LTV_{ℓ_1}-3 is divided into two terms: the time needed to solve the Lagrangian relaxation and the time to solve R-LTV_{ℓ_1}-3. We can see that the catalyst reconstruction obtained with LTV_{ℓ_1} is of higher quality than that obtained with TVM_{ROF} (more than 99% of improvement in both cases). Moreover, our procedure produces a good reconstruction in considerably less computing time (improvement of about 80% and 40% with $\varepsilon = 0.625$ and $\varepsilon = 0.9375$, respectively). Fig. 5 shows enlargements of a specific area in the reconstructions provided by the different models using $\mu = 2^2$. It can be observed that the solution obtained using TVM_{ROF} contains a lot of pixels with intensities that do not correspond with the phantom. Nevertheless, the pixel intensities of the reconstruction using LTV_{ℓ_1} as well as R-LTV_{ℓ_1}-3 are closer to the pixel intensities of the phantom.

The second phantom tested is a circular object with a thin ring. This instance has been chosen to observe the accuracy to recover the white ring (see Durst et al., 2014; López-Haro, Dubau, et al., 2014, for further information about similar objects). Fig. 6 shows that LTV_{ℓ_1} provides a more accurate white ring as compared to the reconstruction obtained by TVM_{ROF} (more than 86% of improvement in both cases) with a considerable reduction in the running times with our procedure (more than 83% and 70% reduction with $\varepsilon = 0.625$ and $\varepsilon = 0.9375$, respectively). Fig. 7 shows a specific area of these reconstructions to highlight the image quality provided by the different methods in a visual way. The μ value was set to 2^3. We can observe that LTV_{ℓ_1} recovers the white ring around the grey circle more accurately, since most pixels of the ring reconstructed using TVM_{ROF} have grey intensities instead of white intensities. Moreover, the results provided by R-LTV_{ℓ_1}-3 considering $\varepsilon = 0.625$ and $\varepsilon = 0.9375$ also show a good performance for reconstructing the white ring.

The last 2D image in which TVM_{ROF} and LTV_{ℓ_1} have been compared is the Shepp–Logan phantom (Shepp & Logan, 1974; Tovey et al., 2019), see Fig. 8. This phantom is more sophisticated than catalyst and circle due to the different particles contained in the image. Fig. 8 shows that LTV_{ℓ_1} outperforms TVM_{ROF} with around 12% error reduction in both cases. Moreover, this example allows to highlight the good

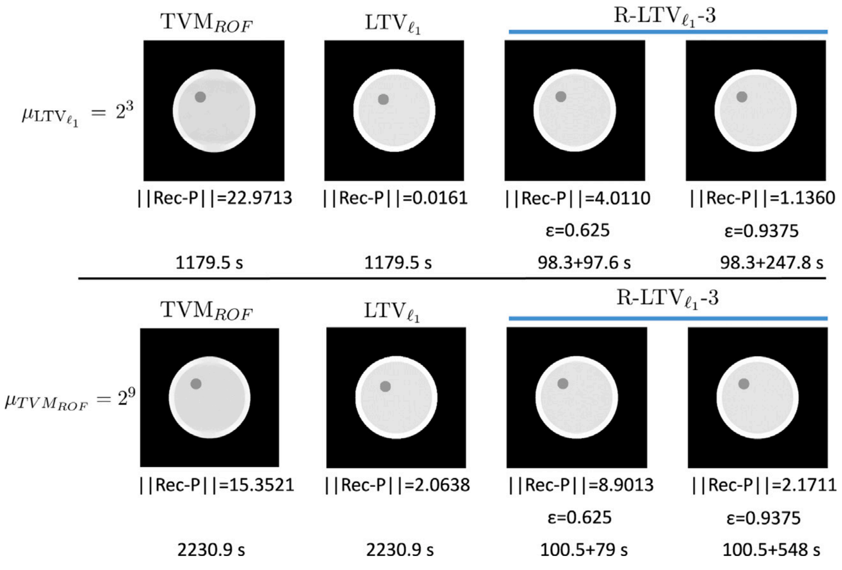

Fig. 6. Reconstruction of the Circle with 1024 × 1024 pixels.

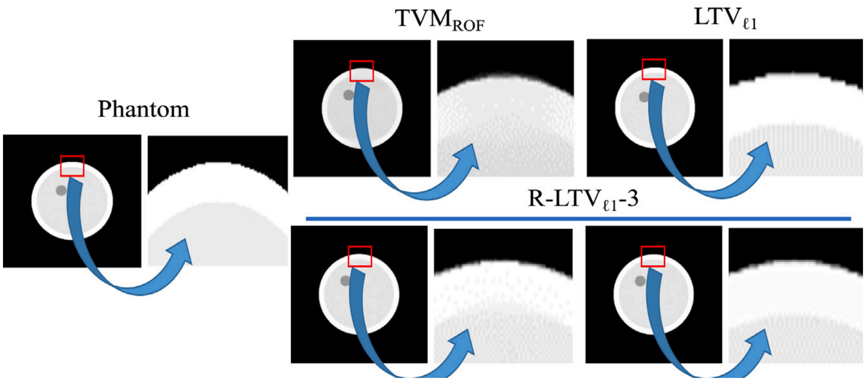

Fig. 7. Enlargements of the areas marked by the red squares.

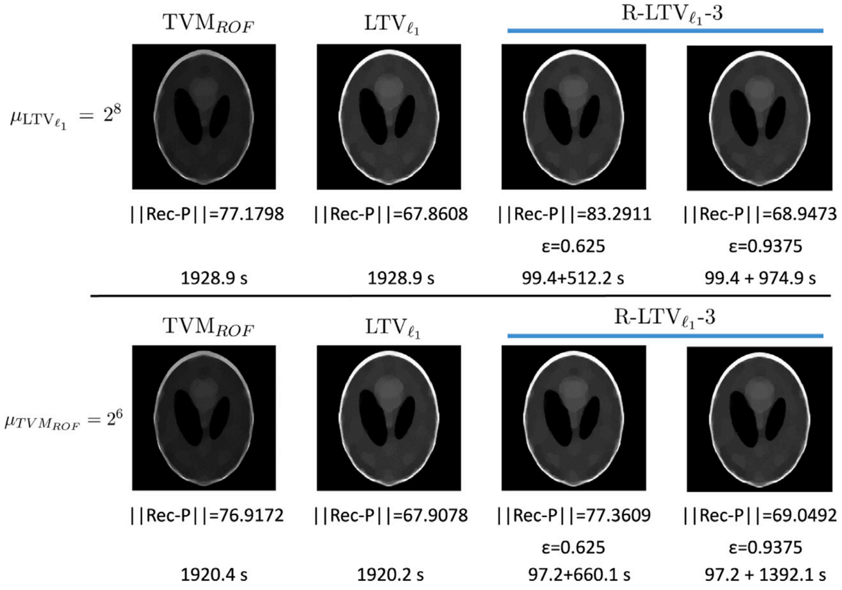

Fig. 8. Reconstruction of the Shepp–Logan phantom with 1024 × 1024 pixels.

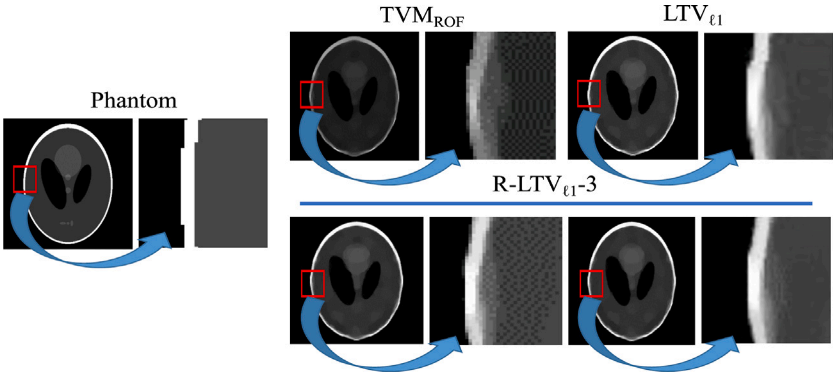

Fig. 9. Enlargements of the areas marked by the red squares.

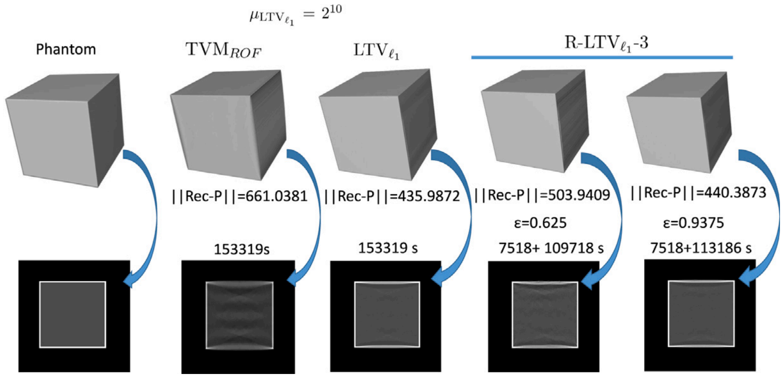

Fig. 10. Reconstruction of a 3D phantom with 512 × 512 × 512 pixels.

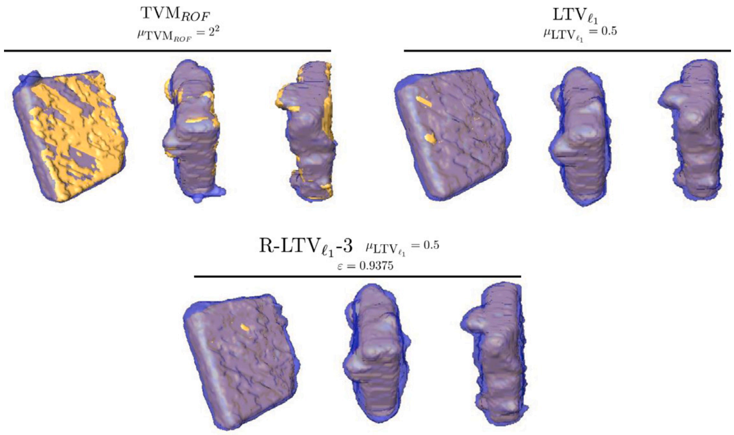

Fig. 11. Reconstruction of a 3D real object with 128 × 128 × 128 pixels.

performance of R-LTV$_{\ell_1}$-3, which provides very good reconstructions in low computing times. In particular, with $\varepsilon = 0.9375$, R-LTV$_{\ell_1}$-3 produced reconstructions of similar quality as those of LTV$_{\ell_1}$ in smaller computing times, with a reduction of 22.44% and 44.30% with respect to those of LTV$_{\ell_1}$ for $\mu = 2^6$ and $\mu = 2^8$, respectively. Fig. 9 highlights an specific area of the white thin cover of the object. These reconstructions have been obtained using $\mu = 2^8$. We can observe that the LTV$_{\ell_1}$ model achieves a more accurate image than the TVM$_{ROF}$ model, since the white structure observed in the enlargement is recovered quite similarly to the phantom. In addition, R-LTV$_{\ell_1}$-3 using $\varepsilon = 0.625$ and $\varepsilon = 0.9375$ has also provided an accurate reconstruction.

The ℓ_1-norm models have also been tested on a 3D phantom (López-Haro et al., 2018). To reconstruct this kind of particles, the whole volume is divided into different 2D slices and every slice is considered independently as a 2D image. In this particular instance, a set of 512 2D reconstructions, with 512 × 512 pixels each, are obtained. Once every slice has been recovered, some specific image processing software is used to generate the reconstructed volume from the 512 2D reconstructions obtained using a standard assembling procedure available at Avizo software (Avizo, 2019). Fig. 10 shows that the results obtained with this 3D instance are similar to those obtained with the considered 2D instances. In the reconstruction produced by TVM$_{ROF}$ the top and bottom parts of the thin coverage are not reproduced. Instead, LTV$_{\ell_1}$ produces a reconstruction of the coverage very similar to the original phantom. As can be seen, in comparison to TVM$_{ROF}$, R-LTV$_{\ell_1}$-3 yields a considerable decrease in the error of 23.76% for $\varepsilon = 0.625$ and 33.37% for $\varepsilon = 0.9375$. This is accompanied with a remarkable reduction in the computing times of 23.53% and 21.27%,

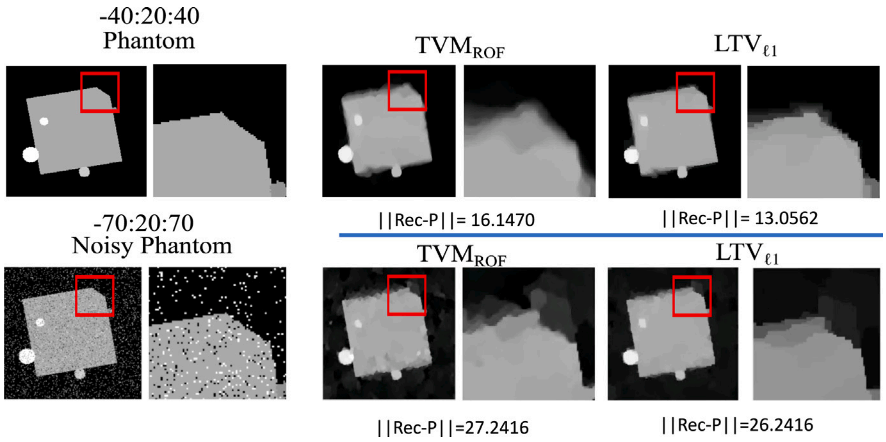

Fig. 12. Performance of the different reconstruction models using a smaller number of projections and including noise in the catalyst phantom.

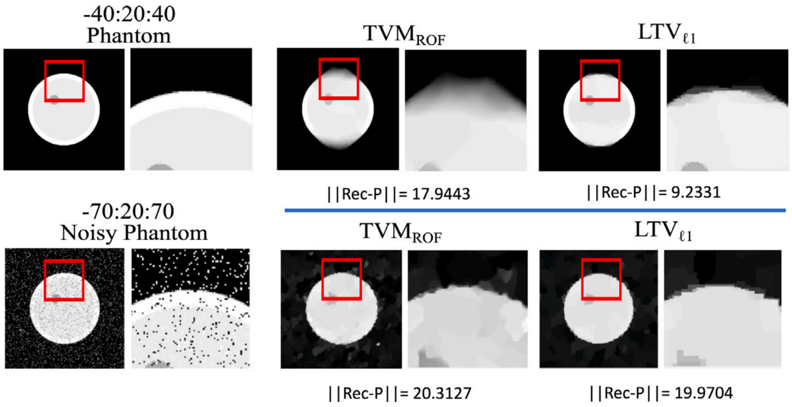

Fig. 13. Performance of the different reconstruction models using a smaller number of projections and including noise in the circle phantom.

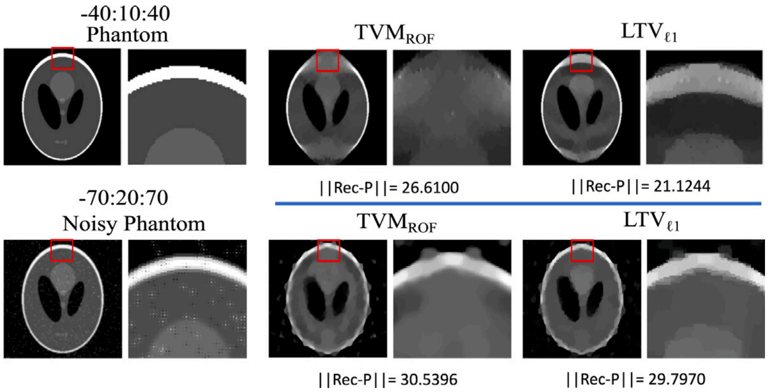

Fig. 14. Performance of the different reconstruction models using a smaller number of projections and including noise in the Shepp–Logan phantom.

for $\varepsilon = 0.625$ and $\varepsilon = 0.9375$, respectively, in each 2D slice, which means a considerable time saving in the complete 3D-reconstruction.

We have also applied the LTV_{ℓ_1} model in an experiment in which the reconstruction has been obtained from a real object instead of using a phantom. This 3D real object is a cube-shaped particle covered with a thin shell of a different material. Experimental STEM HAADF data were collected at the facilities of the DME-UCA node of the Spanish Singular Infrastructure (ICTS) on Electron Microscopy of Materials ELECMI. The difficulty of this reconstruction is mainly due to the thinness of the outer layer of the object. Fig. 11 shows different views (top, front and side) of the reconstructed images obtained with TVM_{ROF}, LTV_{ℓ_1} and R-LTV_{ℓ_1}-3 respectively, where the inner particle is represented in yellow and the coverage in blue. TVM_{ROF} only reconstructs the coverage of some faces of the cube-shaped particle and ignores the top and bottom coverage. Instead, LTV_{ℓ_1} accurately reconstructs the whole coverage.

In the following, we highlight the robustness of the LTV_{ℓ_1} model in electron tomography instances with small number of projections (generating distorted images) and including noise in the original images. Figs. 12–14 show the effect of the number of projections used in the sinogram and the effect of noise in the reconstructions provided by the TVM_{ROF} and the LTV_{ℓ_1} models. We can observe the good performance of the LTV_{ℓ_1} model when reconstructing the image using a tilt range from −40° to 40° with projections recorded every 20° for the catalyst and the circle phantoms, and every 10° for the Shepp–Logan phantom. We observe that the enlarged areas of the images are

Table 2

Best computational results obtained with different values of parameter μ using Algorithms 2 and 3. Projections recorded from −70° to 70° every 20° and 10°.

	Size	μ	TVM_{ROF}		LTV_{ℓ_1}			R-LTV_{ℓ_1}															
								$\varepsilon = 0.625$ (20°) and $\varepsilon = 0.9375$ (10°)								$\varepsilon = 0.9375$ (20°) and $\varepsilon = 0.9937$ (10°)							
								Subgradient (R-LTV_{ℓ_1}-2)				Volume (R-LTV_{ℓ_1}-3)				Subgradient (R-LTV_{ℓ_1}-2)				Volume (R-LTV_{ℓ_1}-3)			
			Error	Time (s)	Error	I(E)(%)	Time (s)	Error	I(E)(%)	Time (s)	I(T)(%)	Error	I(E)(%)	Time (s)	I(T)(%)	Error	I(E)(%)	Time (s)	I(T)(%)	Error	I(E)(%)	Time (s)	I(T)(%)
Catalyst 20°	512 × 512	2^{7}	2.9506	208.4	0.0026	99.91	208.4	4.0988	−28.01	58.0	72.16	3.9048	−24.43	61.6	70.44	1.0452	64.57	88.9	57.34	0.6765	77.07	99.0	52.49
		2^{10}	2.4749	178.3	0.0759	96.93	178.3	5.4212	−54.34	71.0	60.17	4.1199	−39.92	68.9	61.35	0.1300	94.74	118.5	33.53	0.1278	94.83	97.8	45.14
	1024 × 1024	2^{2}	9.1736	1412.4	0.0103	99.88	1412.4	6.4205	30.01	295.0	79.11	6.1456	33.00	293.5	79.21	1.3113	85.70	991.4	29.80	1.3925	84.82	838.9	40.60
		2^{9}	4.2832	1704.5	0.0167	99.61	1704.5	6.6866	−35.94	425.8	75.01	8.9550	−52.16	338.3	80.15	1.4860	65.30	859.6	49.56	2.1799	49.10	1021.8	40.05
Circle 20°	512 × 512	2^{4}	10.1022	138.1	0.0026	99.97	138.1	2.3753	76.48	43.8	68.33	2.4378	75.86	43.0	68.86	1.4609	85.53	61.7	55.32	1.5052	85.10	60.8	55.97
		2^{8}	8.5764	156.9	0.0102	99.88	156.9	2.6110	69.55	44.4	71.70	2.4050	71.95	43.8	72.08	0.5927	93.08	72.3	53.91	0.4105	95.21	62.9	59.91
	1024 × 1024	2^{3}	22.9713	1179.5	0.0161	99.92	1179.5	4.2989	81.28	183.2	84.46	4.0110	82.53	192.1	83.71	0.8763	96.18	350.0	70.32	1.1360	95.05	346.1	70.65
		2^{9}	15.3521	2230.9	2.0638	86.55	2230.9	6.6054	56.97	210.5	90.56	8.9013	42.01	179.5	91.95	2.1441	86.03	660.0	70.41	2.1711	85.85	648.7	70.92
Shepp–Logan 20°	512 × 512	2^{11}	42.0587	305.2	33.5665	20.19	305.2	36.3299	13.62	98.1	67.85	37.1811	11.59	99.0	67.56	33.6400	20.01	201.2	34.07	34.0079	19.14	184.0	39.71
		2^{7}	38.7823	380.6	33.9012	12.58	380.6	36.2255	6.59	79.9	79.0	37.1757	4.14	78.3	79.42	34.5340	10.95	182.1	52.15	34.0633	12.16	156.4	58.90
	1024 × 1024	2^{8}	77.1798	1928.9	67.8608	12.07	1928.9	83.5349	−7.60	694.7	63.98	83.2911	−7.33	620.6	67.82	69.0697	10.50	892.2	53.74	68.9473	10.66	1074.3	44.30
		2^{6}	76.9172	1920.4	67.9078	11.71	1920.4	77.4542	−0.69	573.5	70.12	77.3609	−0.57	757.3	60.56	69.1973	10.03	1249.6	34.93	69.0492	10.22	1490.1	22.40
3D particle 20°	512 × 512	2^{13}	997.3568	50583.8	941.2781	5.62	50583.8	948.5734	4.89	47908.3	5.28	948.3910	4.90	46019.5	9.02	945.8866	5.15	47700.3	5.70	946.2156	5.12	48042.3	5.02
		2^{9}	958.1982	72610.3	946.4871	1.22	72610.3	956.8915	0.13	24683.4	66.00	946.5640	1.21	30570.4	57.89	946.8748	1.18	47159.8	35.05	948.7411	0.98	43261.3	40.41
Catalyst 10°	512 × 512	2^{1}	8.5064	356.7	0.0013	99.98	356.7	0.6142	92.77	274.4	23.07	0.5553	93.47	282.1	20.91	0.0014	99.98	341.6	4.23	0.1159	98.63	315.8	11.46
		2^{9}	0.2928	505.0	0.0875	70.11	505.0	4.0747	−92.81	363.2	28.07	2.4391	−87.99	231.3	54.19	0.7744	−62.19	409.3	18.95	0.8510	−65.59	317.4	37.14
	1024 × 1024	2^{3}	3.9529	3617.2	0.0055	99.86	3617.2	1.5286	61.32	1750.5	51.60	2.0567	47.96	1407.9	61.07	0.8521	78.44	1946.7	46.22	0.0055	99.86	1319.5	63.52
		2^{8}	0.2931	3802.2	0.0918	68.67	3802.2	2.9466	−90.05	1310.4	65.63	3.8415	−92.37	1309.3	65.56	0.5916	−50.45	1557.5	59.03	1.5972	−81.64	1412.5	62.85
Circle 10°	512 × 512	2^{6}	1.3590	668.4	0.0012	99.91	668.4	0.9146	32.70	158.3	76.31	1.5818	−14.08	165.3	75.26	0.0012	99.91	238.5	64.31	0.0012	99.91	294.1	55.99
		2^{10}	0.2667	855.1	0.0065	97.56	855.1	0.4260	−37.39	241.6	71.74	0.4576	−41.71	250.5	70.70	0.1218	54.33	319.6	62.62	0.0068	97.45	311.9	63.52
	1024 × 1024	2^{5}	3.7564	2873.6	0.0052	99.86	2873.6	0.5787	84.59	832.8	71.01	0.1231	96.72	1474.8	48.67	5.5833	−32.72	884.9	69.20	1.5497	58.74	1041.2	63.76
		2^{9}	1.3287	7191.9	0.0063	99.52	7191.9	0.8890	33.09	1416.0	80.31	0.2143	83.87	1931.6	73.14	2.3855	−44.30	1144.0	84.09	1.2719	4.27	1941.6	73.00
Shepp–Logan 10°	512 × 512	2^{3}	10.8441	435.3	0.0024	99.97	435.3	5.1500	52.50	292.1	32.89	5.6144	48.22	260.2	40.22	0.1051	99.03	358.9	17.55	0.1044	99.03	340.1	21.86
		2^{10}	8.9705	463.8	0.8287	90.76	463.8	1.2472	86.09	410.4	11.51	8.4128	6.21	390.6	15.78	0.9723	89.16	450.3	2.91	1.6694	81.39	420.4	3.53
	1024 × 1024	2^{1}	30.3607	3701.3	0.0104	99.96	3701.3	12.3224	59.41	1623.1	56.14	1.8386	93.94	2481.1	32.96	14.2066	53.20	1332.9	63.98	1.7935	94.09	3502.4	5.37
		2^{10}	20.5717	3250.9	6.5423	68.19	3250.9	14.6974	28.55	2952.7	9.17	10.4050	49.42	3185.0	2.02	24.0474	−16.63	2658.1	18.23	10.2799	50.02	2709.4	16.65
3D particle 10°	512 × 512	2^{10}	661.0381	153319.3	435.9872	34.04	153319.3	494.2268	25.23	117382.7	23.43	503.9409	23.76	117236.3	23.53	437.4056	33.83	153418.3	−0.06	440.3873	33.37	120704.6	21.27
		2^{14}	637.3490	123952.7	461.9227	27.52	123952.7	607.1224	4.74	85839.0	30.74	474.6064	25.53	104151.3	15.97	467.6632	26.62	119395.3	3.67	465.1980	27.01	121198.5	2.22

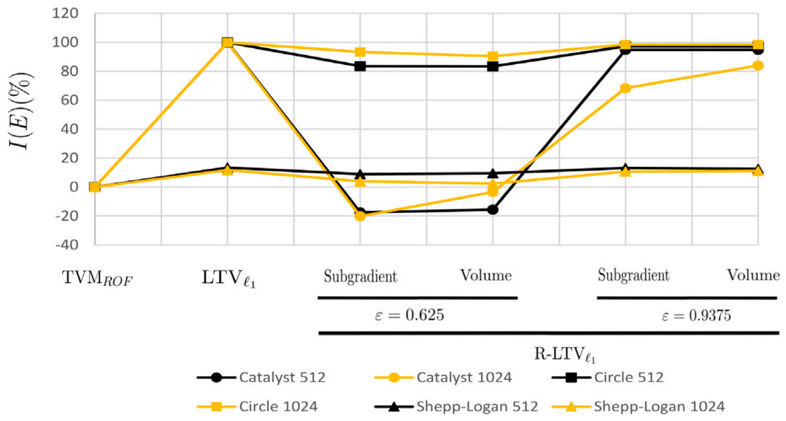

Fig. 15. Percentage improvement of the error, I(E)(%).

accurately reconstructed. In addition, a similar performance can be observed for noisy images.

As we have seen above, Algorithm 1 can be very effective for reconstructing high quality images in small computing times. Still, for the above analysis we have used the values of μ corresponding to the best reconstruction for TVM_{ROF} and to the best reconstruction obtained with formulation LTV_{ℓ_1}. Thus, the reader may wonder how sensitive the different versions of our solution method are to the choice of these parameters, and, in particular, what is the quality of the solutions that they provide and how they compare to TVM_{ROF}. Table 2, together with the more detailed Tables S4–S9 in the Supplementary Material, address this issue. These tables summarize the computational results obtained when applying Algorithm 2 or Algorithm 3 in `Step 1` of Algorithm 1, and compare them with those of TVM_{ROF} for the same parameter values, for projections every 20° and every 10°, respectively with images of 512 × 512 and 1024 × 1024 pixels. Table 2 is structured in four main blocks of columns. The first block contains three columns, which indicate the phantom that is tested, the image sizes, and the value of the parameter μ, respectively. Again, the values of μ correspond to the best reconstruction for TVM_{ROF} and to the best reconstruction obtained with LTV_{ℓ_1}. The second block, with two columns, reports the results obtained with TVM_{ROF}, which are used as the reference for the comparison with the results shown in the remaining blocks. The third block has three columns, corresponding to the results of LTV_{ℓ_1}, whereas the fourth block contains 16 columns corresponding to R-LTV_{ℓ_1}. The first eight columns of the last block correspond to $\varepsilon = 0.625$ and $\varepsilon = 0.9375$ for 8 and 16 projections respectively, and the last eight columns correspond to $\varepsilon = 0.9375$ (8 projections) and $\varepsilon = 0.9937$ (16 projections). In its turn, each of these eight-column blocks is divided in two: one for R-LTV_{ℓ_1}-2, where subgradient optimization was used in `Step 1` of Algorithm 1 and one for R-LTV_{ℓ_1}-3 where the volume algorithm was used instead.

For each reconstruction algorithm we report its error (*Error*), computed as indicated before (the ℓ_2-norm of the difference between the reconstructed and original images), as well as the computing time needed to obtain it (*Time* (s)). All computing times are measured in seconds. In addition, the tables show the percentage improvements, both for quality and time of each procedure relative to TVM_{ROF}, respectively computed as:

$$I(E)(\%) = \frac{Error_{TVM_{ROF}} - Error_{Procedure}}{Error_{TVM_{ROF}}} \times 100,$$

and

$$I(T)(\%) = \frac{Time_{TVM_{ROF}} - Time_{Procedure}}{Time_{TVM_{ROF}}} \times 100.$$

As mentioned, we have fixed the solution time of LTV_{ℓ_1} as stopping criteria of TVAL3.

In Table 2, we can appreciate the improvements of the ℓ_1 based models with respect to TVM_{ROF}, which are more remarkable for the Catalyst and Circle phantoms, with a quality improvement of about 99%. On the other hand, the computing times of R-LTV_{ℓ_1} for 2D instances are particularly important when dealing with 3D volumes, because the 2D reconstruction procedures are applied to every slice that constitutes the whole volume. As explained, two different ε values have been chosen for two different accuracies. The closer to one the value of ε is, the higher are the quality of the image and the computing time. While R-LTV_{ℓ_1} does not provide a good reconstruction when $\varepsilon = 0.625$, a significant improvement in the reconstruction quality is achieved when $\varepsilon = 0.9375$. To analyse the influence of the number of projections on the quality of the reconstructions produced by TVM_{ROF} and LTV_{ℓ_1}, in each case we have computed the overall percentage average improvements for 512 × 512 and 1024 × 1024 images when projections are computed every 10° instead of every 20°, considering in each case the best value of μ-parameter. For the different phantoms these improvements are the following. With Catalyst, 90.66% and 48.30% for TVM_{ROF} and LTV_{ℓ_1}, respectively. With Circle, 94.11% and 60.77% for TVM_{ROF} and LTV_{ℓ_1}, respectively. For the Shepp–Logan phantom, 75.06% and 99.98%, respectively. Finally, for the 3D particle, 33.48% and 53.68% for TVM_{ROF} and LTV_{ℓ_1}, respectively. These results show that LTV_{ℓ_1} is less sensitive to the number of projections for Catalyst and Circle phantoms. Regarding the Shepp–Logan and the 3D particle, we can observe that much better reconstructions are obtained when projections are computed every 10° instead every 20°, both for TVM_{ROF} and LTV_{ℓ_1}. In any case, the reconstructions obtained with LTV_{ℓ_1} are still better than those obtained with TVM_{ROF}, obtaining in the case of projections every 10° even much higher percentages of improvement than every 20°. In addition, we observe that there is not a significant difference between the reconstructions made by R-LTV_{ℓ_1} using the subgradient or the volume algorithms. Actually, the performance of these two options depends on the case under study, although in general we appreciate a slightly better behaviour (both in quality of solutions and in reduction of computing times) in the volume algorithm. These comparative results, reported in Table S4–S9 of Supplementary Material, are illustrated in Figs. 15 and 16 considering as the value of μ, the one that provides the best solution in each reconstruction procedure.

7. Conclusions

In this paper, we have considered an image reconstruction model based on Total Variation Minimization using the ℓ_1-norm, which outperforms state-of-the-art reconstruction models, both, in terms of the

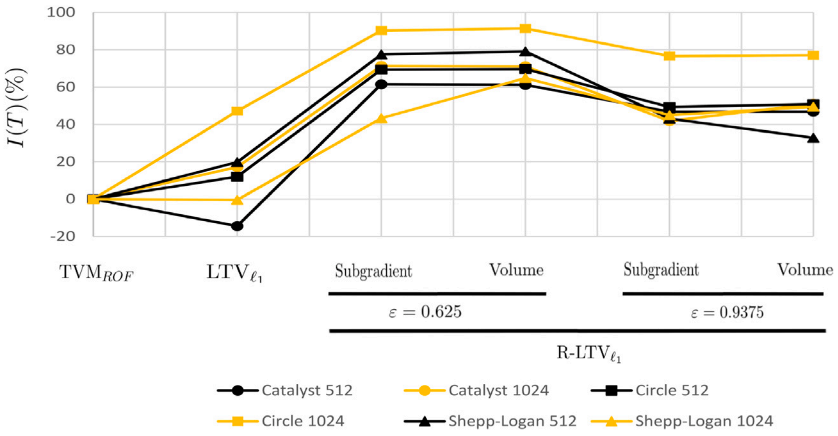

Fig. 16. Percentage improvement of time, I(T)(%).

quality of reconstructed nano-objects using ET and with respect to the number of projections needed for the reconstruction (reducing the morphological damage of samples due to the electron beams). Moreover, the resulting mathematical programming formulation that arises in practice with the proposed model, uses a large number of variables and constraints, which means that it takes a long computing time to obtain optimal solutions with off-the-shelf solvers. Thus, a procedure based on information from dual variables was developed to obtain a reduced problem. In particular, by considering the complementary slackness conditions, some constraints that did not provide useful information for the image reconstruction, were eliminated or modified. It has been possible to verify with different reconstructions that this procedure gives good results in terms of image quality and computing time and produces more accurate reconstructions than TVM_{ROF}.

CRediT authorship contribution statement

José J. Calvino: Conceptualization, Methodology, Software, Writing. **Elena Fernández:** Conceptualization, Methodology, Software, Writing. **Miguel López-Haro:** Conceptualization, Methodology, Software, Writing. **Juan M. Muñoz-Ocaña:** Conceptualization, Methodology, Software, Writing. **Antonio M. Rodríguez-Chía:** Conceptualization, Methodology, Software, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors have been partially supported by Agencia Estatal de Investigación, Spain and ERDF through projects MTM2019-105824GB-I00, PID2020-114594GB-C22, PID2020-113006-RB-I00 and RED2022-134149-T; MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR through project TED2021-130875B-I00; Universidad de Cádiz: PhD grant UCA/REC44VPCT/2021.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.120848.

References

Andersen, A., & Kak, A. (1984). Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging*, *6*(1), 81–94.

Araújo, J. D. L., da Cruz, L. B., Ferreira, J. L., da Silva Neto, O. P., Silva, A. C., de Paiva, A. C., & Gattass, M. (2021). An automatic method for segmentation of liver lesions in computed tomography images using deep neural networks. *Expert Systems with Applications*, *180*, Article 115064.

Avizo (2019). *User's guide avizo software 2019*. Thermo Fisher Scientific.

Balasundaram, S., & Kapil (2010). On Lagrangian support vector regression. *Expert Systems with Applications*, *37*(12), 8784–8792.

Barahona, F., & Anbil, R. (2000). The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, *87*, 385–399.

Bertsekas, D. (1999). *Nonlinear programming*. Athena Scientific.

Bertsimas, D., & Tsitsiklis, J. (1997). *Athena scientific series in optimization and neural computation, Introduction to linear optimization*. Athena Scientific.

Candes, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, *52*(2), 489–509.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306.

Durst, J., López-Haro, M., Dubau, L., Chatenet, L., Soldo-Olivier, Y., Guétaz, L., Bayle-Guillemaud, P., & Maillard, F. (2014). Reversibility of Pt-skin and Pt-skeleton nanostructures in acidic media. *The Journal of Physical Chemistry Letters*, *5*(3), 434–439.

Feldkamp, L. A., Davis, L. C., & Kress, J. W. (1984). Practical cone-beam algorithm. *Journal of the Optical Society of America*, *1*(6), 612–619.

Fisher, M. (1985). An applications oriented guide to Lagrangian relaxation. *Interfaces*, *15*(2), 10–21.

Fu, H., Ng, M., Nikolova, M., & Barlow, J. (2006). Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration. *SIAM Journal on Scientific Computing*, *27*, 1881–1902.

Gordon, R., Bender, R., & Herman, G. T. (1970). Algebraic reconstruction technique for three-dimensional electron microscopy and X-ray photography. *Journal of Theoretical Biology*, *29*(3), 471–476.

Goris, B., Van den Broek, W., Batenburg, K., Heidari Mezerji, H., & Bals, S. (2012). Electron tomography based on a total variation minimization reconstruction technique. *Ultramicroscopy*, *113*, 120–130.

Held, M., Wolfe, P., & Crowder, H. (1974). Validation of subgradient optimization. *Mathematical Programming*, *6*, 62–88.

Kaczmarz, S. (1937). Angenherte Auflsung von Systemen linearer Gleichungen. *Bulletin International de L'Acadmie Polonaise Des Sciences Et Des Lettres*, *35*, 355–357.

Knudsen, B. R., Grossmann, I. E., Foss, B., & Conn, A. R. (2014). Lagrangian relaxation based decomposition for well scheduling in shale-gas systems. *Computers & Chemical Engineering*, *63*, 234–249.

Li, C., Yin, W., Jiang, H., & Zhang, Y. (2013). An efficient augmented Lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, *56*(3), 507–530.

López-Haro, M., Dubau, L., Guétaz, L., Bayle-Guillemaud, P., Chatenet, M., André, J., Caqué, N., Rossinot, E., & Maillard, F. (2014). Atomic-scale structure and composition of Pt3Co/C nanocrystallites during real PEMFC operation: A STEM–EELS study. *Applied Catalysis B: Environmental*, *152–153*(1), 300–308.

López-Haro, M., Guétaz, L., Printemps, T., Morin, A., Escribano, S., Jouneau, P.-H., Bayle-Guillemaud, P., Chandezon, F., & Gebel, G. (2014). Three-dimensional analysis of Nafion layers in fuel cell electrodes. *Nature Communications*, *5*(5229).

López-Haro, M., Tinoco, M., Fernández-Garcia, S., Chen, X., Hungria, A., Cauqui, M., & Calvino, J. (2018). A macroscopically relevant 3D-Metrology approach for nanocatalysis research. *Particle & Particle Systems Characterization*, *35*(3), Article 1700343.

Luo, X., Yu, W., & Wang, C. (2018). An image reconstruction method based on total variation and wavelet tight frame for limited-angle CT. *IEEE Access*, *6*, 1461–1470.

Lustig, M., Donoho, D., & Pauly, J. (2007). Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, *58*(6), 1182–1195.

Midgley, P., & Dunin-Borkowski, R. (2009). Electron tomography and holography in materials science. *Nature Materials*, *8*(4), 271–280.

Midgley, P., Ward, E., Hungría, A., & Thomas, J. (2007). Nanotomography in the chemical, biological and materials sciences. *Chemical Society Reviews*, *36*, 1477–1494.

Nicoletti, O., de La Peña, F., Leary, R., Holland, D., Ducati, C., & Midgley, P. (2013). Three-dimensional imaging of localized surface plasmon resonances of metal nanoparticles. *Nature*, *502*(7469), 80–84.

Oosterhout, S. D., Wienk, M. M., van Bavel, S. S., Thiedmann, L., A., J., Gilot, J., Loos, J., Schmidt, V., & Janssen, R. A. J. (2009). The effect of three-dimensional morphology on the efficiency of hybrid polymer solar cells. *Nature Materials*, *8*(10), 818–824.

Pi, J., Wang, H., & Pardalos, P. M. (2021). A dual reformulation and solution framework for regularized convex clustering problems. *European Journal of Operational Research*, *290*(3), 844–856.

Radon, J. (1917). Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Mathematisch-Physische Klasse*, *69*, 262–277.

Rudin, L., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, *60*(1–4), 259–268.

Shepp, L., & Logan, B. (1974). The Fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, *21*(3), 21–43.

Shor, N. (1985). *Springer series in computational mathematics*, *Minimization methods for non-differentiable functions*. Springer.

Tang, X., Hsieh, J., Nilsen, R. A., Dutta, S., Samsonov, D., & Hagiwara, A. (2006). A three dimensional-weighted cone beam filtered backprojection (CB-FBP) algorithm for image reconstruction in volumetric CT-helical scanning. *Physics in Medicine and Biology*, *51*(4), 855–874.

Thavavel, V., Basha, J. J., Krishna, M., & Murugesan, R. (2012). Heuristic wavelet approach for low-dose EPR tomographic reconstruction: An applicability analysis with phantom and in vivo imaging. *Expert Systems with Applications*, *39*(5), 5717–5726.

Tovey, R., Benning, M., Brune, C., Lagerwerf, M., Collins, S., Leary, R., Midgley, P., & Schönlieb, C. (2019). Directional sinogram inpainting for limited angle tomography. *Inverse Problems*, *35*(2), Article 024004.

Vanderbei, R. (2007). *Linear programming: foundations and extensions*. New York: Springer.