Geophysical Journal International

Geophys. J. Int. (2022) **231,** 638–649 https://doi.org/10.1093/gji/ggac211
Advance Access publication 2022 June 10
GJI General Geophysical Methods

Tensor-based reconstruction applied to regularized time-lapse data

Jonathan Popa, Susan E. Minkoff and Yifei Lou

Department of Mathematical Sciences, The University of Texas at Dallas, Richardson, TX 75080, USA. E-mail: jmp108020@utdallas.edu

Accepted 2022 June 7. Received 2022 May 19; in original form 2022 January 18

SUMMARY

Repeatedly recording seismic data over a period of months or years is one way to identify trapped oil and gas and to monitor CO_2 injection in underground storage reservoirs and saline aquifers. This process of recording data over time and then differencing the images assumes the recording of the data over a particular subsurface region is repeatable. In other words, the hope is that one can recover changes in the Earth when the survey parameters are held fixed between data collection times. Unfortunately, perfect experimental repeatability almost never occurs. Acquisition inconsistencies such as changes in weather (currents, wind) for marine seismic data are inevitable, resulting in source and receiver location differences between surveys at the very least. Thus, data processing aimed at improving repeatability between baseline and monitor surveys is extremely useful. One such processing tool is regularization (or binning) that aligns multiple surveys with different source or receiver configurations onto a common grid. Data binned onto a regular grid can be stored in a high-dimensional data structure called a tensor with, for example, x and y receiver coordinates and time as indices of the tensor. Such a higher-order data structure describing a subsection of the Earth often exhibits redundancies which one can exploit to fill in gaps caused by sampling the surveys onto the common grid. In fact, since data gaps and noise increase the rank of the tensor, seeking to recover the original data by reducing the rank (low-rank tensor-based completion) successfully fills in gaps caused by binning. The tensor nuclear norm (TNN) is defined by the tensor singular value decomposition (tSVD) which generalizes the matrix SVD. In this work we complete missing time-lapse data caused by binning using the alternating direction method of multipliers (or ADMM) to minimize the TNN. For a synthetic experiment with three parabolic events in which the time-lapse difference involves an amplitude increase in one of these events between baseline and monitor data sets, the binning and reconstruction algorithm (TNN-ADMM) correctly recovers this time-lapse change. We also apply this workflow of binning and TNN-ADMM reconstruction to a real marine survey from offshore Western Australia in which the binning onto a regular grid results in significant data gaps. The data after reconstruction varies continuously without the large gaps caused by the binning process.

Key words: Image processing; Numerical modeling; Numerical solutions; Computational seismology.

1 INTRODUCTION

Time-lapse (4-D) seismic data is used for applications such as reservoir monitoring (Lumley 2001) and monitoring of CO_2 injection wells (Eiken *et al.* 2000). To acquire time-lapse data, multiple surveys are conducted over a region, over a period of months or years. These surveys can reveal changes in the region caused by production or injection. Repeatability is desired for time-lapse surveys in order to obtain accurate differences and allow for precise interpretation. However, acquisition inconsistencies such as cable feathering caused by ocean currents can result in non-repeatable receiver locations, which makes taking a time-lapse difference challenging. Feathering refers to the angle between a streamer and a straight reference line. This angle varies as streamers drift due to wind and other environmental factors. Schonewille (2005) demonstrates how feathering reduces repeatability and how non-repeated shot-lines with azimuth differences cause time-shifts in the data. Pevzner *et al.* (2011) find that the soil saturation affects the signal velocity of land surveys, demonstrating how time-lapse surveys taken during different seasons have less repeatability. Zhou & Lumley (2021) show how non-repeatability can cause artefacts in the velocity models resulting from 4-D full waveform inversion.

To mitigate non-repeatability issues, pre-stack and post-stack processing is used to help make the data sets comparable. Processing often involves cross-equalization workflows consisting of substeps such as regularizing, matched-filtering, amplitude balancing, and warping (Rickett & Lumley 2001; Nguyen *et al.* 2015). Regularizing or binning data aligns multiple surveys with different receiver configurations onto a common grid. Non-repeatability issues cause receiver coordinates to vary between surveys over the same region. By aligning two data sets which contain recording inconsistencies, we inevitably create empty cells that do not have recorded values. Regularizing the two data sets followed by data completion allows the surveys to be compared with one another. Lane (1994) describes a simple binning algorithm to transform the receivers' spatial coordinates to an integer coordinate system for a single 3-D survey. Rickett & Lumley (2001) include a spatial anti-aliasing filter to realign two surveys to a common grid. The filter serves to equalize dip content, as steep events on a fine grid are spatially aliased on a coarse grid. Gierse *et al.* (2010) use a common-reflection-surface method to both regularize and interpolate pre-stack data from two 3-D seismic surveys onto a common grid. Once data are binned to a regular grid, it can be stored in discrete data structures such as matrices or tensors (i.e. multidimensional arrays). Similar to data binning, image registration (Hill *et al.* 2001; Lou *et al.* 2013) and pansharpening (Li *et al.* 2018; Vivone *et al.* 2021) have been used to align data defined on different domains. There are a variety of medical imaging modalities such as computerized tomography (CT), magnetic resonance imaging (MRI) and positron emission tomography (PET) that require data to be regularized onto a common grid to facilitate diagnostics (El-Baz *et al.* 2011a, b). Pansharpening involves interpolating or sharpening a sequence of spectral images with low spatial resolution to match a single high-resolution panchromatic image thereby creating high-resolution hyperspectral data.

After binning, the resulting data structure is not necessarily complete. Gaps between receivers greater than the mesh size result in empty grid cells. We seek to fill in the gaps of the regularized data by applying a data completion method and thus improve the repeatability of the data. Oghenekohwo *et al.* (2017) use a joint-recovery method based on compressive sensing to complete regularized data. Their method groups a pair of jointly sparse measurement vectors from the baseline and monitor surveys into a matrix and solves for the common and unique components of their reconstructed signals. This joint method improves the repeatability of time-lapse data compared to splitting the matrix into a set of vectors and recovering each independently. Our novel approach to this time-lapse completion problem is to apply low rank tensor-based reconstruction.

Tensor-based data completion has grown enormously in the past decade and has been applied in several studies to seismic data (Stanton *et al.* 2012; Kreimer *et al.* 2013; [data set] Chen *et al.* 2016; Sacchi & Cheng 2017; Cheng *et al.* 2019; Carozzi & Sacchi 2019; Popa *et al.* 2021), but not previously to time-lapse data. The motivation for using tensor-based methods is to exploit the multidimensional structure in a manner not possible for matrices or vectors. For instance, a single gather taken as a matrix may be nearly full rank. However, forming a tensor from a group of gathers often leads to a low rank structure due to redundancy across the gathers. The theory behind low rank reconstruction assumes the underlying data has low rank, but missing data and noise present in an observation can increase the data's rank, hence we can recover missing data by minimizing the rank. In practice rank reduction methods for matrices use the singular value decomposition (SVD), as the rank of a matrix is equal to the number of non-zero singular values in its SVD.

There are numerous low rank reconstruction methods applicable to high dimensional data (tensors) such as multichannel singular spectrum analysis (MSSA, Oropeza & Sacchi 2011), parallel matrix factorization (PMF, Xu *et al.* 2015), higher-order SVD (HOSVD) reduction (Kreimer *et al.* 2013) and tensor nuclear norm (TNN) minimization (Ely *et al.* 2015). The MSSA method minimizes the rank of Hankel matrices formed from frequency slices. Oropeza & Sacchi (2011) apply MSSA to 3-D seismic data, demonstrating successful recovery of missing traces and noise reduction. Chen *et al.* (2019) apply MSSA to USArray recordings of 3-D earthquake data for simultaneous reconstruction and denoising. Assuming the data is generated from plane waves, they implement a localized version of the rank reduction method to subsets of the data volume, showing improvement over global rank-reduction.

The PMF method unfolds a tensor into a matrix and utilizes an alternating least squares algorithm that is SVD-free to approximate the matrix as a product of two low rank matrices. Gao *et al.* (2015) apply PMF to seismic data and compare with SVD-based methods. Gao *et al.* (2017) present parallel square matrix factorization (PSMF) as an improvement over PMF by reshaping the unfolded tensor into a nearly square matrix. This reshaping is achieved by partitioning the dimensions into two sets, then reshaping the tensor into a matrix with rows and columns equal to the product of the first and second set of partitioned dimensions respectively. This reshaping reduces the required number of measurements for reconstruction and maintains the low-rank property of the tensor (Mu *et al.* 2014). The HOSVD is a generalization of the matrix SVD for higher orders of data (De Lathauwer *et al.* 2000). This generalization is based on unfolding the tensor into a matrix and computing the matrix SVD. Kreimer *et al.* (2013) develop a HOSVD reduction method and find completion of a 4-D spatial volume (with sources and receivers in two dimensions) to be more accurate than 2-D and 3-D completion.

The TNN is defined by the tensor SVD (tSVD), a generalization of SVD (Kilmer & Martin 2011). Ely *et al.* (2015) apply alternating direction method of multipliers (ADMM, Boyd *et al.* 2010) for TNN minimization to synthetic and field data, demonstrating successful reconstruction and denoising. This combination of model and algorithm is referred to as TNN-ADMM (Liu *et al.* 2020). Popa *et al.* (2019) show that TNN-ADMM results in improved reconstruction as the amount of data in the tensor increases and as the physical space between samples decreases. This result is explained by how data sampled closely together contains more redundancies and results in a tensor of lower rank. Popa *et al.* (2020, 2021) demonstrate how using conjugate symmetry significantly speeds up the computation of the tSVD for tensors of any dimension. Additionally, they analyse the impact of a tensor's orientation on reconstruction. Since the singular values of a tensor are equivalent to the singular values of a block diagonal matrix containing the Fourier transformed slices (Zhang *et al.* 2014), TNN minimization is expected to perform best for the tensor with the most-square orientation since rectangular matrices perform worse for low rank reconstruction methods than do square matrices. Furthermore, Popa *et al.* (2021) provide a physical explanation to motivate the success of the most-square orientation for seismic data: the most-square orientation of a seismic data tensor often has frontal faces formed over the spatial dimensions, and lateral continuity of a region causes data to be low rank with respect to the spatial dimensions. In Popa *et al.* (2021) TNN-ADMM is extensively compared to two well known methods for data completion: projection onto convex sets (POCS) and MSSA. They demonstrate that TNN-ADMM provides more accurate results in less runtime than these other completion methods for a synthetic and real data example.

In this paper our contribution is a data reconstruction workflow for pre-stack time-lapse seismic data, consisting of regularizing data into a tensor and applying TNN-ADMM, a tensor-based completion method. By regularizing the time-lapse data, we create a domain that allows seismic traces to be compared at the same physical locations for each survey. However, there is no guarantee that time-lapse data is recorded at the same physical locations due to issues of non-repeatability. Hence we use data completion to fill in missing data so that these surveys can be compared. We use a variant of the TNN-ADMM algorithm described by Popa *et al.* (2021), optimized for efficiency using conjugate symmetry and accuracy using the most-square tensor orientation. We validate this method on a synthetic example and demonstrate its performance on real data. In the synthetic example we generate baseline and monitor surveys with different acquisition geometries. We regularize the receiver data onto a grid to construct tensors containing the observed data. We then apply TNN-ADMM to the baseline and monitor data tensors independently to reconstruct the data over the entire grid. By completing the data we are able to accurately determine the time-lapse difference. In the real data example we regularize a shot from a baseline and monitor marine survey and apply the same completion method described for the synthetic problem. The reconstructed gathers appear continuous and reasonable for seismic data.

Our paper is organized as follows; in Section 2 we provide an overview of our method. We first give background on tensor linear algebra, then we detail how to bin data to a regular grid and use this binning to form a tensor. Finally, we describe TNN-ADMM, our choice of completion algorithm. In the third and fourth sections we apply our method to synthetic and real data examples, respectively. The discussion in Section 5 covers topics including determining whether low rank reconstruction is appropriate for the data, how the choice of mesh size affects receiver binning, correcting for receiver centre offsets and temporal smoothing.

2 METHOD

This section contains three subsections. First, we review terminology and notation related to tensors. Next, we discuss how we regularize data onto a grid, leading to a tensor with data gaps. In the last subsection, we present details about TNN-ADMM, the completion algorithm we use for reconstruction.

2.1 Terminology and notation

We define necessary terminology and background on tensor algebra in this section. Tensor algebra generalizes linear algebra to higher dimensions. The *order* of a tensor refers to the number of dimensions of the tensor. For example, vectors and matrices can be considered as order-1 and order-2 tensors, respectively. In this paper, we are primarily concerned with order-3 tensors. However the theory holds for higher order tensors as well. An order-3 tensor is a cube of data. We use capital letters to denote matrices and script capital letters to denote tensors of orders three and greater, such as $\mathcal{X}$.

As an illustration of tensor operations, we consider a 3-D seismic survey with the spatial x and y directions as the first two dimensions respectively and time as the third. A *slice* of a tensor refers to a submatrix obtained by fixing one of the three indices. A *frontal slice* refers to a slice for a fixed value of the third dimension. In this example, a frontal slice would be a cross-section in the x–y directions for a fixed t value. A *tube* is a vector oriented along the third dimension, for example a tube would correspond to a single time trace (see Fig. 1b). To refer to a scalar entry of a tensor we use subscripts, for example $\mathcal{X}_{i,j,k}$ is the entry of $\mathcal{X}$ in the ith row, jth column and kth frontal slice.

The *orientation* of a tensor is the order of its dimensions (Kilmer & Martin 2011). By permuting the dimensions of a tensor it is possible to obtain different orientations. The *most-square orientation* is the orientation for which the frontal slices are the closest to square matrices out of all possible orientations (Popa *et al.* 2021). For 3-D seismic data, the time dimension is often significantly larger than the spatial dimensions. In this case the most-square orientation corresponds with ordering the spatial dimensions first and second, with time third.

Martin *et al.* (2013) define the tensor SVD (tSVD) as a generalization of the matrix SVD, which is a critically important part of low rank reconstruction. Given a real valued tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tSVD decomposes $\mathcal{X}$ into a product of three tensors as follows,

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \tag{1}$$

such that $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a diagonal tensor. We use the symbol $*$ to denote the tensor product and $\mathcal{V}^T$ to denote the transpose of $\mathcal{V}$. For definitions of tensor product, transpose, orthogonal and diagonal tensors, see the Appendix. The diagonal entries of $\hat{\mathcal{S}}$ (the Fourier transform of $\mathcal{S}$ in the tubal direction) are the singular values of $\mathcal{X}$. Using tSVD we can define the TNN as the sum of singular values of the tensor. For example, for order-3 tensors the TNN is defined by:

$$\|\mathcal{X}\|_{\mathrm{TNN}} = \sum_{k=1}^{n_3} \sum_{i=1}^{\min(n_1,n_2)} \hat{\mathcal{S}}_{i,i,k}. \tag{2}$$

The TNN is motivated by the matrix nuclear norm, that is the sum of the singular values, which is NP-hard to minimize (Candès & Recht 2009). For a matrix X let s denote a vector comprised of the singular values of X. The rank and nuclear norm of X are equivalent to the l_0 and l_1 vector norms of s, respectively. The l_0 norm counts the number of non-zero entries, whereas the l_1 norm sums the magnitude of each entry. Since the l_1 norm is the tightest convex relaxation to the l_0 norm, minimizing the matrix nuclear norm does a good job of approximating rank minimization (Candès & Tao 2005; Cai *et al.* 2010). Likewise, TNN provides a convex relation to tensor tubal rank (Semerci *et al.* 2014; Zhang & Aeron 2017), which is defined as the number of non-zero tubes of $\hat{\mathcal{S}}$ or more specifically the maximal rank among all the frontal slices $\hat{\mathcal{S}}_{:,:,k}$.

2.2 Regularizing data

Given a seismic survey, we seek to form a tensor containing the seismic traces while preserving the locations of the receivers. That is, we map the physical coordinates of a receiver to indices of a matrix. By regularizing data in this manner, we can apply a tensor-based completion method to fill in missing data for coordinates where no receiver is present.

First we discretize the region over which the survey is taken. This process enables us to use discrete data structures such as matrices and tensors to organize real data. We illustrate this concept in Fig. 1(a). The three solid lines represent streamers, with nodes indicating receivers. We discretize this region into a 5 × 5 grid, resulting in each receiver being contained within a unique grid cell. We colour these grid cells, illustrating how we use the discretization to map receiver locations to indices of a matrix. Since receivers are not necessarily located at the centres of grid cells, there is an

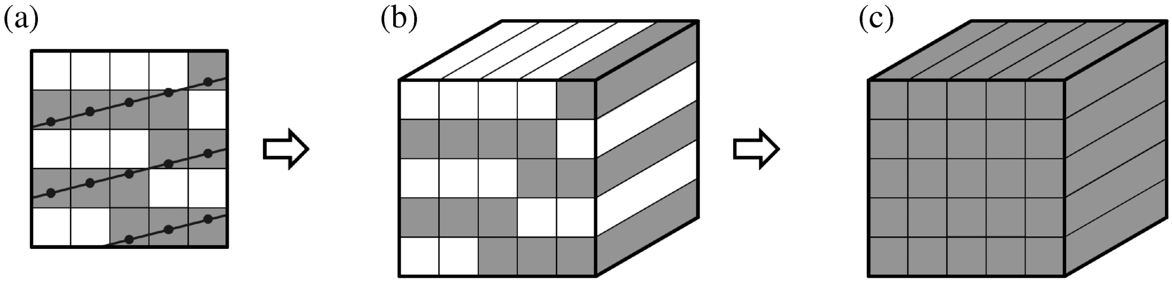

Figure 1. (a) The receiver locations (shown as black circles) are mapped to grid squares. Each shaded grid square contains a receiver. (b) Tensor with seismic traces shown as tubes. The indices of each binned receiver in (a) are used to determine the indices of the tube. (c) A completed tensor.

inherent error to naively binning receivers at the cell centres as we have here. This binning error is the difference between the recorded trace and an ideal trace recorded at the centre of the cell. In general, this error cannot be calculated for real data. However, the grid size spacing bounds the furthest distance a trace can be from a grid cell centre, and hence bounds the binning error. We note that our approach is to work with data after it has been recorded and processed. Thus, while the frequency bands of the sources and receivers and the acquisition geometry could certainly affect the accuracy of data binning and completion, we have not undertaken a careful study of when these effects would negatively impact our algorithm. In Fig. 1(b) we display an order-3 tensor with a 5 × 5 frontal face. Each shaded tube of this tensor contains a recorded trace from the receiver whose binned indices match the tube's indices. The white tubes of this tensor are empty (zero entries), as no data was recorded at these locations. Now that we have formed a tensor with incomplete data, incomplete in the sense we do not have data for every tube, we can apply a completion method to fill in the missing data. In Fig. 1(c), we colour the formerly white tubes to indicate the data has been completed for all grid locations.

With a completed tensor we have estimated data at each grid location. Performing this process for both baseline and monitor surveys results in complete tensors for each survey with estimated data at every grid location. Thus the surveys can be compared pointwise to calculate the 4-D difference. We use TNN-ADMM as our choice of completion method, discussed in detail in the following section. Other completion algorithms such as transform or frequency based methods (projection onto convex sets or MSSA) could also be used.

For our examples we choose a grid spacing such that each receiver is binned into a unique grid square. When considering time-lapse data the grid spacing should be chosen carefully as it is possible for one survey to satisfy the unique binning assumption while the other does not. Having multiple receivers binned to the same grid cell creates ambiguity as each tube can contain only one trace. A solution to this problem is to use an interpolation method to produce a single trace from the binned traces. For simplicity we will avoid the case of binning multiple receivers to the same grid cell in this work.

2.3 TNN-ADMM

Data completion methods are used to reconstruct missing or incomplete data. The relation between observed data and completed data can be represented by,

$$\mathcal{Y} = \mathcal{A}(\mathcal{X}), \tag{3}$$

where $\mathcal{Y}$ is the observation, $\mathcal{A}$ is a sampling operator and $\mathcal{X}$ represents the complete data. The sampling operator $\mathcal{A}$ can be considered as a tensor with ones where data are recorded and zeros where it is not. The sampling operator acts on $\mathcal{X}$ by the Hadamard or entry-wise product (Kreimer *et al.* 2013). Observed seismic data often contains redundancies or similarities. When representing data as a matrix or tensor, repeated rows or columns is an indicator of low rank.

Using TNN as the objective function, one can formulate the low rank minimization problem as follows,

$$\min ||\mathcal{X}||_{\text{TNN}} \quad \text{s.t. } \mathcal{Y} = \mathcal{A}(\mathcal{X}). \tag{4}$$

We can solve (4) using the ADMM (Boyd *et al.* 2010). ADMM alternates between reducing the TNN objective function and satisfying the data matching constraint, $\mathcal{Y} = \mathcal{A}(\mathcal{X})$. This formulation of the objective function with an exact equality constraint is used for completion without denoising, which is the focus of this paper. The denoising case requires an additional regularization term in the objective function $\mu\|\mathcal{Y} - \mathcal{A}(\mathcal{X})\|_{\text{F}}^2$ in place of the equality constraint and requires tuning of the denoising parameter μ.

In order to implement this method for data completion without denoising, we introduce an auxiliary variable $\mathcal{Z}$ to decouple the TNN term and the constraint, expressing the problem (4) equivalently as,

$$\min ||\mathcal{Z}||_{\text{TNN}} + 1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})} \quad \text{s.t. } \mathcal{X} = \mathcal{Z}. \tag{5}$$

Here the term $1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})}$ is the indicator function, taking a value of zero when the subscript equation is satisfied and a value of ∞ otherwise.

Using the decoupled eq. (5), we define the augmented Lagrangian as,

$$L_{\rho}(\mathcal{X}, \mathcal{Z}; \mathcal{B}) = ||\mathcal{Z}||_{\text{TNN}} + 1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})} + \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z} + \mathcal{B}\|_F^2 - \frac{\rho}{2}\|\mathcal{B}\|_F^2, \tag{6}$$

where $\mathcal{B}$ is the dual variable, the parameter $\rho > 0$ is the step size and $\| \cdot \|_F$ is the Frobenius norm (for details see the Appendix). The dual variable $\mathcal{B}$ is also referred to as the Lagrange multiplier, which is introduced to deal with the equality constraint $\mathcal{X} = \mathcal{Z}$.

ADMM is an iterative method that alternates between minimizing the augmented Lagrangian (6) with respect to $\mathcal{X}$ and $\mathcal{Z}$, and updating the dual variable $\mathcal{B}$ by using gradient ascent. The ADMM iterations are given by (Ely *et al.* 2015),

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X}} \left(1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})} + \tfrac{\rho}{2}||\mathcal{X} - \mathcal{Z}^k + \mathcal{B}^k||_{\text{F}}^2\right), \tag{7}$$

$$\mathcal{Z}^{k+1} = \arg\min_{\mathcal{Z}} \left(||\mathcal{Z}||_{\text{TNN}} + \tfrac{\rho}{2}||\mathcal{X}^{k+1} - \mathcal{Z} + \mathcal{B}^k||_{\text{F}}^2\right), \tag{8}$$

$$\mathcal{B}^{k+1} = \mathcal{B}^k + \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}, \tag{9}$$

where the superscript k denotes iteration number. Eq. (7) represents the solution to the data matching constraint. Eq. (8) represents the solution to minimizing the TNN objective function. Eq. (9) is the

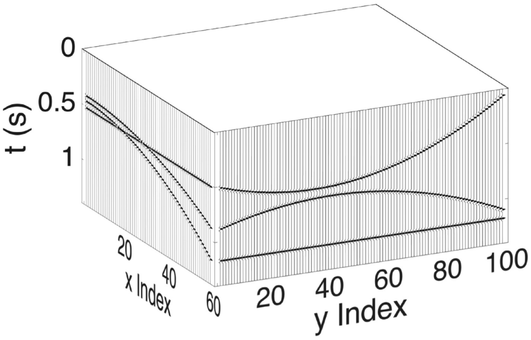

Figure 2. Tensor of synthetic data containing parabolic events. Tensor has dimension $(x, y, t) = 60 \times 100 \times 700$. The x–t and y–t faces illustrate a single gather or slice of the tensor.

dual variable update. In our numerical experiments we apply TNN-ADMM to reconstruct incomplete tensors formed by binning data onto a regular grid.

3 NUMERICAL EXPERIMENTS

3.1 Synthetic Example

We first demonstrate our method of regularization and completion on a synthetic data set. We will see that for this example completing baseline and monitor surveys on a regular grid results in accurate estimation of the time-lapse difference. We choose to use TNN-ADMM as our completion method. In Popa *et al.* (2021) TNN-ADMM is shown to be more accurate and efficient than two other completion methods, POCS and MSSA. On a synthetic example all three methods successfully reconstruct the data, with TNN-ADMM and MSSA having errors of less than 1 per cent, and POCS having an error of less than 4 per cent. Additionally, TNN-ADMM converges with a runtime that is one order of magnitude less than the time taken by POCS and two orders of magnitude less than MSSA. On a real data example TNN-ADMM results in the best recovery, having an error 5 per cent less than MSSA and 10 per cent less than POCS. The runtime of TNN-ADMM for the real data example is an order of magnitude less than POCS and three orders of magnitude less than MSSA. While these runtime comparisons and errors are only from a small subset of possible experiments, we believe they are representative of some of the advantages of TNN-ADMM over competing methods.

We generate synthetic data using a modified version of a code made publicly available by [data set] Chen *et al.* (2016). The complete synthetic baseline data consists of a trace for every bin on a regular grid of size 60×100. Each trace is recorded for 1.4 s with a 2 ms time sampling, resulting in a tensor of data with dimensions $60 \times 100 \times 700$. Each trace recording contains three Gaussian pulses, each corresponding to an event. Fig. 2 illustrates the tensor of data with gathers in the x–t and y–t dimensions. Along these faces it can be seen that one of the events is linear in x and parabolic in y, one is linear in y and parabolic in x, and one is parabolic in both x and y. For this synthetic experiment we assume during production the event which is parabolic in x and linear in y undergoes an amplitude increase by a factor of two from the initial time to the time the monitor data is recorded.

Figs 3(a) and (c) show the baseline and monitor streamers, respectively. Figs 3(b) and (d) display the gather corresponding to the bold highlighted streamer in (a) and (c), respectively. In both surveys there are 50 streamers, each with 60 receivers. The streamers in the baseline survey are designed with greater feathering than the monitor survey, resulting in differing receiver locations between the surveys. Additionally, the monitor streamers are designed to be slightly shorter, with smaller spacing between consecutive receivers.

We have binned each receiver into a unique grid cell, and hence, the trace of each receiver is binned to a unique tube of a tensor. We assign each receiver location an integer (x, y) coordinate pair corresponding to the index of the grid square for the receiver, which is then used for assigning the receiver's trace to a tube of a tensor. This binning process results in a tensor that contains non-zero traces only at receiver locations. Fig. 4(a) illustrates the receivers on the grid and Fig. 4(b) shows the gather that corresponds to the highlighted row in Fig. 4(a). This gather is comprised of the traces of receivers that intersect the designated row. The empty space between receiver locations results in observable gaps in this gather. Similar gathers can be produced by plotting the binned traces along any row or column. Each such gather can be considered as a slice of the tensor containing the regularized traces. The ratio of receivers (i.e. non-zero tubes) to the total number of grid cells can be calculated to reveal the percent of the tensor that is filled with non-zero data. For this synthetic example 50 per cent of the tensor is filled and 50 per cent is missing.

The regularized tensor represents a limited observation, with data available only at the receivers. This observation is produced by applying a sampling operator to a tensor containing the data on the entire grid. We reconstruct the baseline and monitor data independently by applying TNN-ADMM to the sampled data. Due to the differences in streamer length and feathering, the monitor data covers a subset of the baseline survey. This subdomain consists of baseline rows 5–99 (out of 1–100), and excludes the first and last columns due to the initial offset of the streamers and the spacing between receivers. On this subdomain, approximately 46 per cent of the tubes of the tensor are empty or missing. In the TNN-ADMM algorithm we use a step size of $\rho = 10^{-2}$ for both baseline and monitor data sets. This value is selected by running a fixed number of iterations over a range of ρ values and determining the value of ρ which produces the least error relative to the ground-truth. We discuss our strategy for finding ρ in real data examples where the ground-truth is unknown in the next subsection. We calculate the error as $\|\mathcal{X}^* - \mathcal{X}^k\|_{\mathrm{F}}/\|\mathcal{X}^*\|_{\mathrm{F}}$, where $\mathcal{X}^*$ represents the ground-truth and $\mathcal{X}^k$ represents the ADMM solution at the kth iteration. We multiply the error by 100 to express it as a percentage. We consider the algorithm to have reached convergence when the relative change in the reconstructed tensor between consecutive iterations is below a threshold $\epsilon = 10^{-4}$. Additionally, we use the most square orientation of the tensor to yield optimal results (Popa *et al.* 2020).

The baseline and monitor completion problems converge in 147 and 84 iterations, respectively. Figs 5 and 6 show the TNN value and relative error as a function of iteration. We observe that the TNN of the observation is larger than the TNN value of the underlying true data, which is significant since TNN-ADMM will not succeed when the observation has lower TNN value than the original data.

Figs 7(a) and (b) display a reconstructed gather for the baseline and monitor data sets, respectively. The baseline is reconstructed with error around 7.27 per cent over the entire domain. The majority of the error occurs near the boundary, that is, in the first and last slices of the tensor. The difference between the true data and these

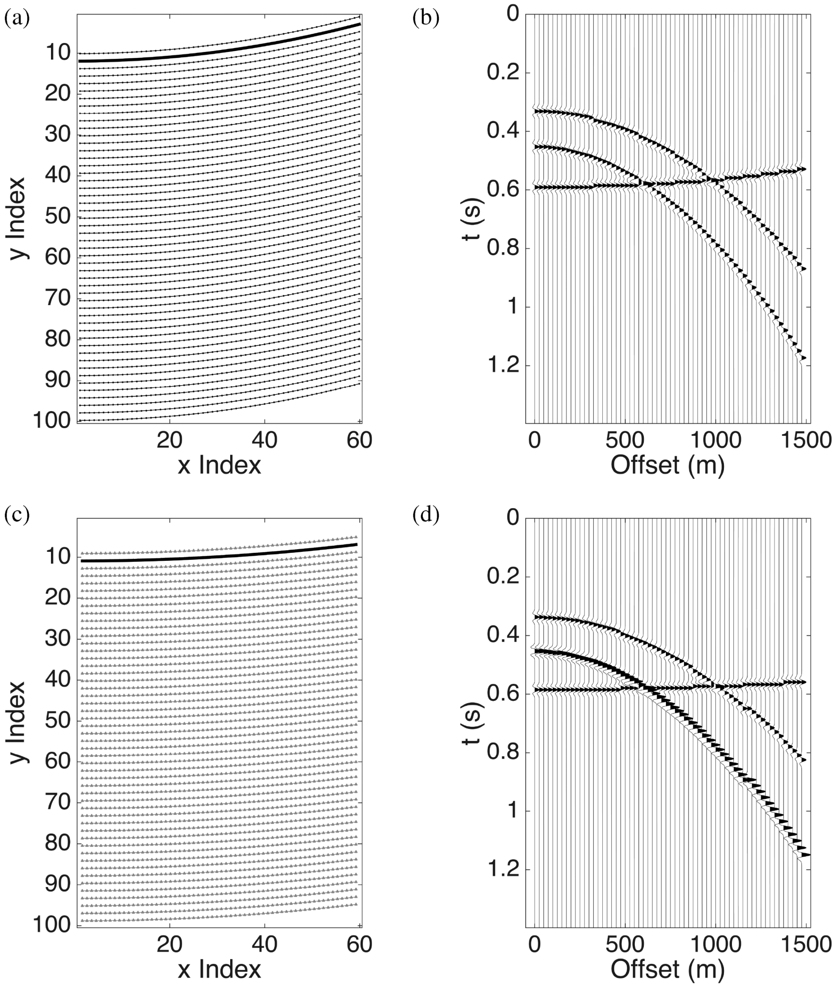

Figure 3. The streamers and receiver locations for the (a) baseline and (c) monitor surveys. For each survey we highlight a streamer and plot the corresponding gather in (b) and (d). The time lapse difference is contained in the later parabolic event which has twice the amplitude in the monitor survey compared to the baseline.

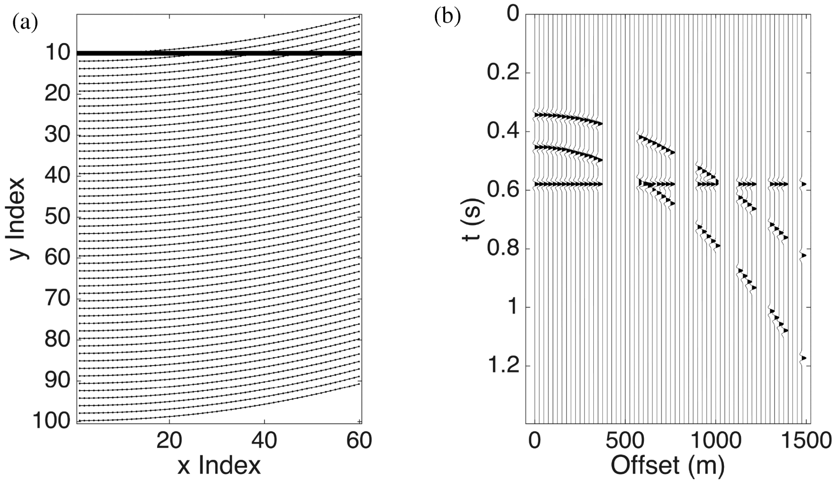

Figure 4. (a) The baseline survey on a regular grid. (b) A single gather (horizontal slice of the tensor) from receivers along the row highlighted in (a). Note the gaps between receivers.

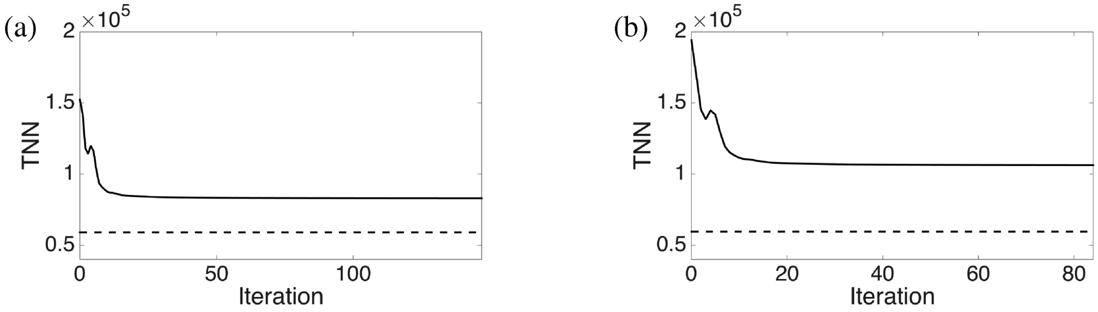

Figure 5. TNN objective function for the reconstructed (solid line) and ground truth (dashed) data as a function of iteration for the (a) baseline and (b) monitor data sets.

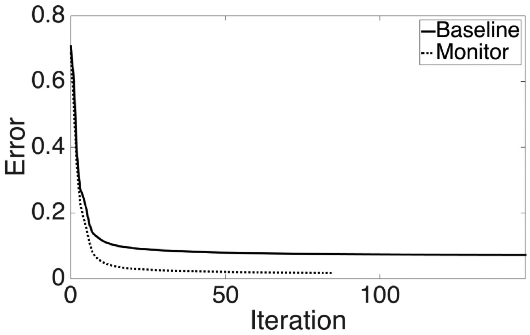

Figure 6. The relative error in the reconstructed data as a function of iteration for TNN-ADMM for the baseline (solid) and monitor (dotted) data sets. The method converges in fewer iterations for the monitor survey than for the baseline.

reconstructed edge slices is shown in Fig. 8. These slices have errors of 57.64 and 34.82 per cent, respectively. If these slices are removed, then the error over the rest of the tensor is 2.78 per cent. The large errors at the boundaries are likely due to undersampling at these locations.

The monitor survey is reconstructed with error around 1.8 per cent over the whole tensor. While the reconstructed monitor data has the largest error at its boundaries, it is not as severe as for the baseline. In Fig. 9 we show the difference between the true and reconstructed monitor data for the first and last slices. These slices have errors around 14.68 and 5.63 per cent, respectively. The interior (excluding these two slices of the tensor) has an error of 0.81 per cent. We attribute better performance for the monitor's reconstruction to a smaller feathering angle than the baseline, leading to more samples per row near the boundary.

After reconstruction, we compute the time-lapse difference. We observe an accurate difference between the reconstructed data sets, with the lower parabolic event clearly identified in Fig. 10. This difference is within 4.52 per cent of the difference in the true data between baseline and monitor data sets, demonstrating the success of this method.

3.2 Real data example

We now illustrate the regularization and reconstruction method on real time-lapse data from offshore Western Australia. For this example we use the data from only a single shot location. After selecting a shot in the baseline survey, we select the closest shot in the monitor survey. The source locations in our example are are within 1.18 m of each other.

Fig. 11 shows the acquisition geometry for our selected baseline and monitor shots, as well as a streamer gather from each. The baseline gather includes 12 streamers, each with 288 receivers. The monitor gather includes 12 streamers as well, but each contains only 204 receivers. For both surveys the receivers are spaced approximately 12.5 m apart and the streamers are separated by approximately 51 m. The recording duration is 3.462 and 3.514 s for the baseline and monitor data sets, respectively, with 2 ms sampling.

Before regularizing the surveys, we rotate the streamers to align horizontally. We refer to the axes of this rotated coordinate system as *u* and *v*. By aligning the receivers horizontally we seek to avoid the undersampling issue that occurred at the boundaries of the synthetic problem. Furthermore, if there is redundancy in the data across the streamers, that is the rows are redundant, then the frontal slices will have low rank. Due to the differences in the survey geometries, we seek the rotation angle that results in the best compromise between these surveys. For these data the baseline streamers experienced greater feathering than the monitor streamers. Thus, we choose our rotation angle such that the first and last receiver for the first streamer of the baseline survey are close to horizontal so that multiple receivers are binned to each row. Other rotation angles result in some horizontal rows near the top and bottom of the domain having very few baseline receivers.

To regularize the data, we choose a discretization such that no grid cell contains more than one receiver. To find this discretization we compute the differences between the coordinates of consecutive receivers in both the horizontal and vertical directions, then take the minimum values of these differences for our grid spacing in each direction. This selection of grid spacing paired with the rotation of the data aims to minimize the total number of grid cells under the constraint of binning receivers uniquely. Using smaller cells increases the percent of dead traces, which consequently increases the error. Using larger cells can cause ambiguities as multiple traces may be binned into the same cell. The dimensions of our discretized grid are 304 × 18. We plot the streamers in the rotated coordinate system with the receivers on the regularized grid shown in Fig. 12. The receiver locations on the grid are then used to bin the traces at those locations into a tensor. This binning process is performed for both baseline and monitor surveys. In Fig. 13, we display a slice of the tensor for the baseline and monitor surveys. Each slice corresponds to a single row of the grid and contains all the traces from any receivers that are binned

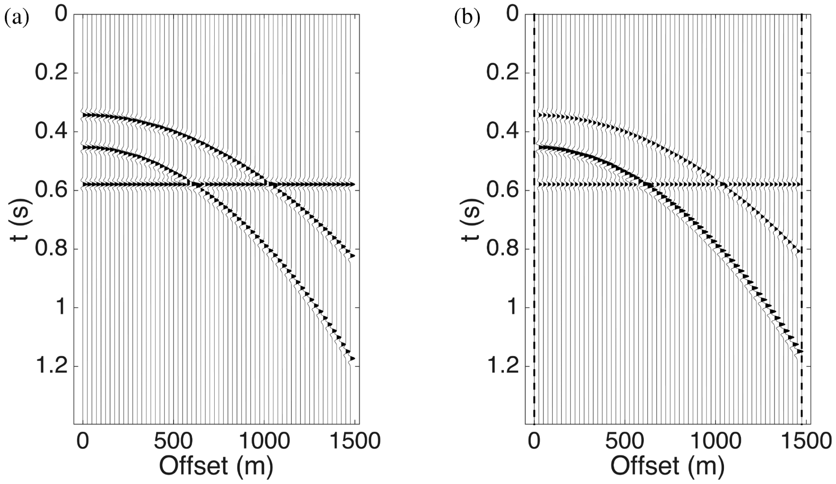

Figure 7. A gather from the reconstructed tensor for (a) the baseline and (b) monitor. The vertical dashed lines in (b) mark the edges of the domain, as the monitor streamers are slightly shorter than the baseline streamers.

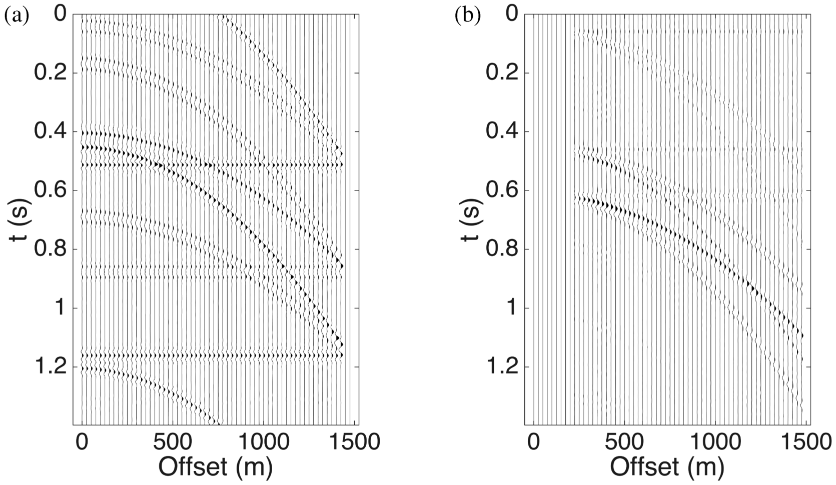

Figure 8. The baseline difference for the (a) first and (b) last slices of the 3-D tensor which contain the largest errors. These slices have relative errors of 57.64 and 34.82 per cent, respectively.

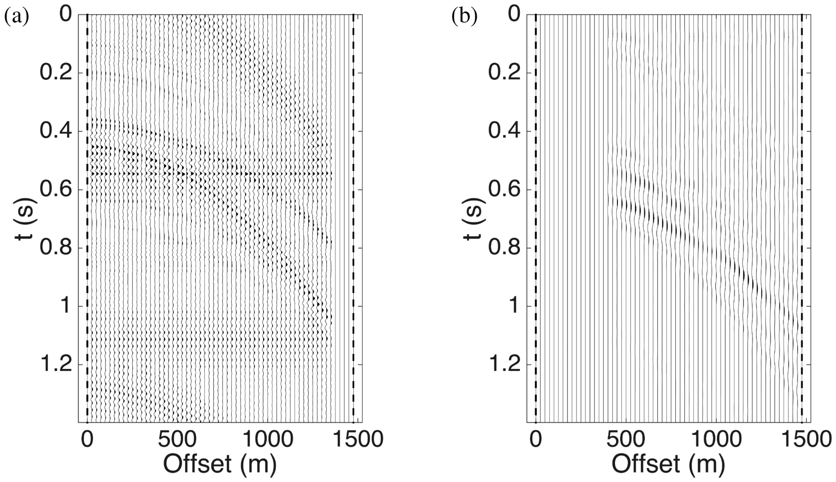

Figure 9. The monitor difference for the (a) first slice and (b) last slice of the monitor domain (corresponding to slices 5 and 99 in the baseline tensor). These plots have been amplified by a factor of five to better show the difference. These slices have errors of 14.68 and 5.63 per cent, respectively.

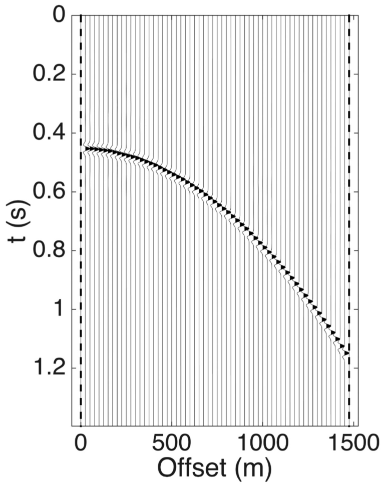

Figure 10. A slice of the time-lapse difference between the reconstructed baseline and monitor surveys on their common domain. The true time-lapse difference is concentrated in the lower parabolic event as is clearly observed.

to this row, with gaps between receivers regarded as zeros in the tensor.

The slices in Fig. 13 are plotted on the same range in u. However we use dashed lines to indicate the subregion over which each survey resides. While the baseline survey spans the entire grid horizontally, covering all 304 discretized columns, the monitor survey is contained within a subset, spanning only 218 of these columns. This difference in span is due to the shorter cable lengths in the monitor survey. Furthermore, the monitor survey has a greater initial offset and does not contain data for the first column of the grid. The difference in the vertical alignment of these surveys in this coordinate system causes them to span different sets of rows on the grid. The monitor survey spans the first 17 rows of the grid. The baseline survey spans rows 2–18. The baseline and monitor survey data tensors have approximately 33 and 34 per cent of the tubes empty for each respective survey's subdomain. Additionally, the difference in the recording duration results in the baseline data having a smaller time duration than the monitor data. These domain differences need to be considered when examining the time-lapse difference post data completion.

After binning the data from each survey, we apply TNN-ADMM to complete each survey independently on their respective domains. For the step size parameter in TNN-ADMM eqs (7) and (8) we use $\rho = 2 \times 10^{-4}$ and a convergence threshold of $\epsilon = 10^{-3}$. In real data examples where the ground-truth is unknown, we approximate the best value for ρ among a pre-determined set of values from the procedure described in Popa *et al.* (2021). In that paper, we completed the same baseline data discussed here and considered values of ρ from the set $P = \{10^{-5}, 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$. The procedure is as follows: data traces are arranged into a tensor $\mathcal{T}$ with dimensions for receiver number, streamer number, and time. Without binning, the tensor $\mathcal{T}$ is considered complete and can thus be regarded as the ground-truth. We then apply a sampling operator to decimate 60 per cent of the traces. For a specific $\rho \in P$, we apply TNN-ADMM to fill in the missing data and calculate the error between the reconstructed result and the original tensor $\mathcal{T}$. We choose as our value for ρ the one that produces the least error. The ρ that yields minimal error by this procedure may not be the best ρ for completion of the tensor obtained by binning the data, but in the absence of the ground-truth it is our best estimate. In the experiments described in this paper we repeat this process for the monitor data. Fig. 14 shows the slices of the reconstructed tensor corresponding to the same slices shown in Fig. 13 for the baseline and monitor surveys. We observe the recovered result appears reasonable, with events that are continuous although somewhat noisy.

Real data requires further processing to see a time-lapse difference. Post-stack processing techniques such as deghosting, phase rotation, time-shift and amplitude correction are beyond the scope of this work. The reconstruction of data in the pre-stack domain aims to provide an improved starting point for these processing methods.

4 DISCUSSION

When applying this binning and recovery method to data in general several factors must be considered. Low rank methods require the observation to have higher rank than the underlying data, otherwise minimizing an approximation of rank can not be expected to succeed. In the synthetic case, the rank of the observation and underlying data can be compared to ensure low rank reconstruction is suitable. For real data while the rank of the observation can be computed, the rank of the ideal reconstruction or ground truth is not known. However, the norms of operators are correlated with the norms of the observations they produce, that is for two operators A and B if the norm of A is greater than the norm of B then it is expected that the norm of $A(X)$ will be greater than the norm of $B(X)$ for most X (Popa *et al.* 2019). Hence comparing the norm of a sampling operator, $\mathcal{A}$, to the identity operator can give an indication of whether the observation produced by the sampling operator, $\mathcal{A}(\mathcal{X})$, is likely to have a lower norm value than the (unknown) true $\mathcal{X}$.

For our real data example, the regularized baseline data had a sampling operator with a TNN value of 2.71×10^5 while the identity operator for this data has a TNN value of 1.25×10^5. For the regularized monitor data the sampling operator and identity operator have TNN values of 2.2×10^5 and 1.07×10^5, respectively. In both cases since the TNN of the sampling operators are greater than the identity we expect the observation has greater rank than the (unknown) ground truth, hence low rank reconstruction is a suitable approach for recovery.

When discretizing surveys there is freedom in the choice of the mesh size. The size of the discretization should be selected based on the needs of the experiment. For our experiments we choose the mesh to be sufficiently small such that receivers are binned uniquely. In general, finer discretizations are more computationally expensive and increase the percent of empty bins, creating a more challenging reconstruction problem. Any mesh coarser than the one we used may bin multiple receivers into the same grid cell. This binning creates ambiguity as to which trace to assign to the corresponding tube for that bin in the tensor. In such a case one could apply interpolation to produce a single interpolated trace at the centre of the cell from the binned traces (Carozzi & Sacchi 2021).

For the synthetic problem we observed significant error near the boundaries where undersampling is most pronounced. These errors appear to be the time-shifted duplicates of the correct events. Since the events in this example should be smooth with respect to time, we hypothesize that a temporal smoothing operator could help to reduce these errors.

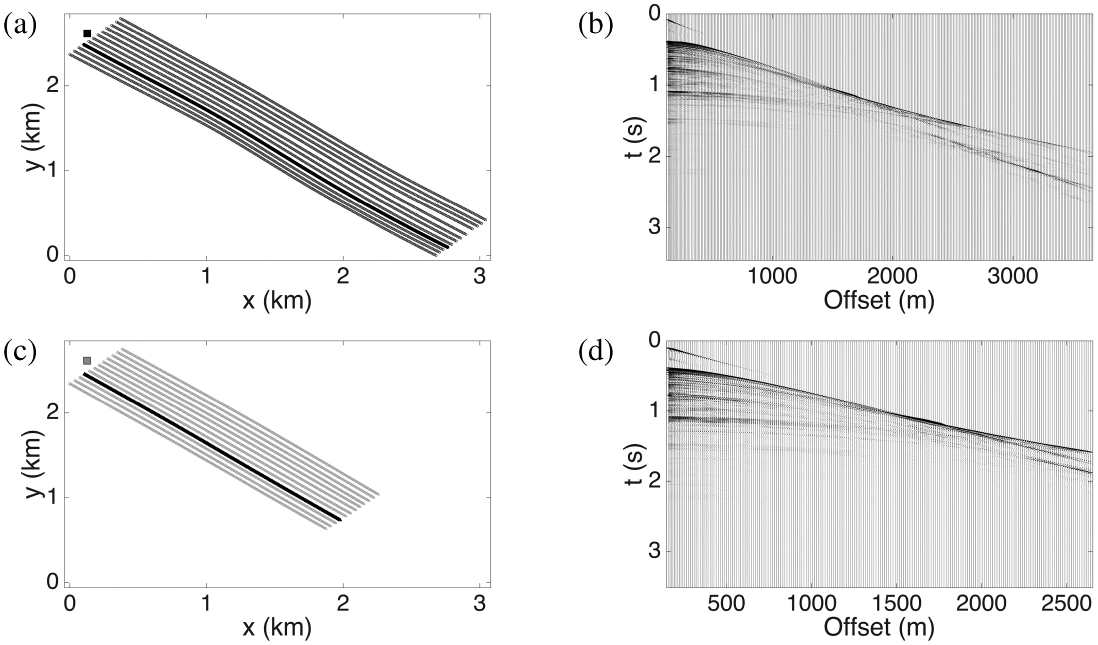

Figure 11. The acquisition geometry of (a) a baseline shot and (c) the monitor shot with matching source locations. The receiver locations are marked with black (baseline) and grey (monitor) nodes. The source location is marked with a square in the top left for each survey. For the highlighted baseline and monitor streamers, we plot the corresponding gathers in (b) and (d), respectively.

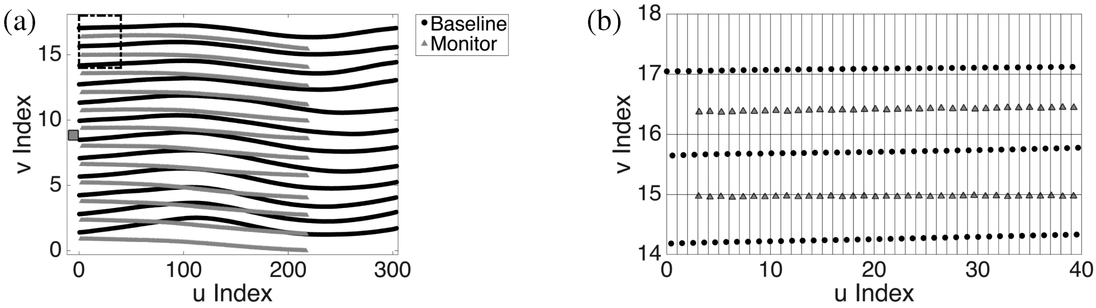

Figure 12. We rotate the streamers such that the receivers align horizontally. We plot (a) the receiver locations in this rotated coordinate system, using black and grey to distinguish the baseline and monitor surveys, respectively, and mark the source location by a square on the left edge of the domain. Panel (b) shows a blow up of the region in the upper left dashed box in panel (a). We use *u* and *v* to denote the horizontal and vertical axes of this rotated coordinate system, respectively.

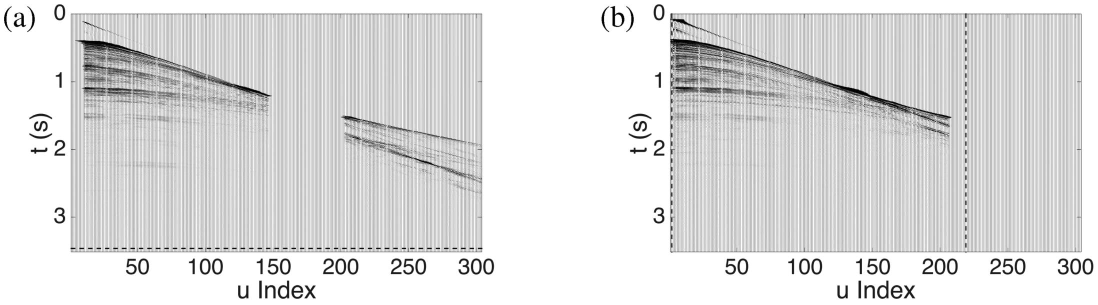

Figure 13. Slices of the (a) baseline and (b) monitor tensors for the row $v = 11$. Each slice is composed of traces from all receivers binned into this row. The horizontal dashed line in (a) marks the end of the recording duration for the baseline traces, as the monitor data was recorded for a slightly longer interval. The vertical dashed lines in (b) mark the edges of the monitor domain, as the baseline streamers are longer and cover a larger region for a single shot gather.

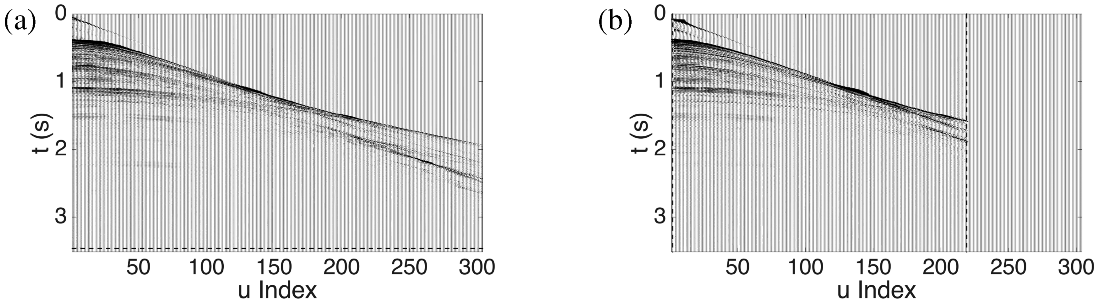

Figure 14. The slices of the reconstructed tensor for (a) baseline and (b) monitor, corresponding to the same regularized tensor slices shown in Fig. 13. The dashed lines indicate the boundary of each survey.

5 CONCLUSIONS

The low-rank tensor-based reconstruction method TNN-ADMM is useful for reconstructing seismic data. We show how one can apply this method to time-lapse data. In order to create comparable tensors for two sets of data recorded over the same region, a regularization method is required to bin traces. We bin receivers to unique grid cells and record gaps as zero entries in the tensor to generate observed data. By applying TNN-ADMM to this tensor for both baseline and monitor surveys, we are able to reconstruct the data at every grid location. This reconstruction allows the surveys to be compared.

In the synthetic example we showed how this method produced an accurate time-lapse difference. For the real data example additional processing is required before taking a time-lapse difference. However, the reconstructed results are encouraging as recovered events appear continuous, and reconstructed pre-stack data is beneficial for post-stack processing. Future studies can be conducted to analyse the effects of differing mesh sizes, binning multiple traces to grid cells, adjusting traces for centre offsets, implementing denoising into the reconstruction and applying a smoothing operator to smooth events near the boundary of the domain.

ACKNOWLEDGMENTS

This research is partially supported by the sponsors of the UT Dallas ‘3D+4D Seismic FWI’ research consortium and by NSF CAREER Award no. 1846690. We thank BHP, and its partners, Santos and INPEX, for providing the offshore Western Australia data and James Cai for his expertise with this data set and time-lapse data processing in general. We are very grateful to David Lumley for many helpful conversations in which he shared technical advice on the need for time-lapse regularization, and on the topics of non-repeatability, time-lapse data processing, and construction of synthetic examples. We also thank Kai Xiong and Wei Zhou for their help in preparing the data used in our experiments. Finally, we express our gratitude to the editor Andrew Valentine, the assistant editor Fern Storey, reviewer Mauricio Sacchi, as well as an anonymous reviewer for their helpful feedback and comments.

DATA AVAILABILITY

The synthetic data used in this study is available upon request. The real data in this study is strictly confidential and not available for public use.

REFERENCES

Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.,* **3,** 1–122.

Cai, J.-F., Candès, E.J. & Shen, Z., 2010. A singular value thresholding algorithm for matrix completion, *SIAM J. Opt.,* **20**(4), 1956–1982.

Candès, E. & Recht, B., 2009. Exact matrix completion via convex optimization, *Found. Comput. Math.,* **9**(6), 717–772.

Candès, E. & Tao, T., 2005. Decoding by linear programming, *IEEE Trans. Inform. Theory,* **51**(12), 4203–4215.

Carozzi, F. & Sacchi, M.D., 2019. Robust tensor-completion algorithm for 5D seismic-data reconstruction, *Geophysics,* **84**(2), V97–V109.

Carozzi, F. & Sacchi, M.D., 2021. Interpolated multichannel singular spectrum analysis: a reconstruction method that honors true trace coordinates, *Geophysics,* **86**(1), V55–V70.

Chen, Y., Bai, M. & Chen, Y., 2019. Obtaining free USArray data by multi-dimensional seismic reconstruction, *Nat. Commun.,* **10**(4434), doi:10.1038/s41467-019-12405-0.

Cheng, J., Sacchi, M. & Gao, J., 2019. Computational efficient multidimensional singular spectrum analysis for prestack seismic data reconstruction, *Geophysics,* **84**(2), V111–V119.

[dataset] Chen, Y., Huang, W., Zhang, D. & Chen, W., 2016. An open-source Matlab code package for improved rank-reduction 3D seismic data denoising and reconstruction, *Comput. Geosci.,* **95,** 59–66.

De Lathauwer, L., De Moor, B. & Vandewalle, J., 2000. A multilinear singular value decomposition, *SIAM J. Matrix Anal. Appl.,* **21**(4), 1253–1278.

Eiken, O., Brevik, I., Arts, R., Lindeberg, E. & Fagervik, K., 2000. Seismic monitoring of CO_2 injected into a marine acquifer, in *SEG Technical Program Expanded Abstracts 2000,* pp. 1623–1626, Society of Exploration Geophysicists.

El-Baz, A.S., Acharya, U.R., Laine, A.F. & Suri, J.S., 2011a. *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies,* Vol. **1,** Springer Publishing Company.

El-Baz, A.S., Acharya, U.R., Laine, A.F. & Suri, J.S., 2011b. *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies,* Vol. **2,** Springer Publishing Company.

Ely, G., Aeron, S., Hao, N. & Kilmer, M.E., 2015. 5D seismic data completion and denoising using a novel class of tensor decompositions, *Geophysics,* **80,** V83–V95.

Gao, J., Stanton, A. & Sacchi, M.D., 2015. Parallel matrix factorization algorithm and its application to 5D seismic reconstruction and denoising, *Geophysics,* **80,** V173–V187.

Gao, J., Cheng, J. & Sacchi, M.D., 2017. Five-dimensional seismic reconstruction using parallel square matrix factorization, *IEEE Trans. Geosci. Remote Sens.,* **55**(4), 2124–2135.

Gierse, G., Otto, D., Berhorst, A., Trappe, H. & Pruessmann, J., 2010. CRS technique for advanced prestack merging and regularisation of vintage 3D seismic data, in *SEG Technical Program Expanded Abstracts 2010,* pp. 3624–3628, Society of Exploration Geophysicists.

Hill, D.L., Batchelor, P.G., Holden, M. & Hawkes, D.J., 2001. Medical image registration, *Phys. Med. Biol*., **46**(3), R1–45.

Kilmer, M.E. & Martin, C.D., 2011. Factorization strategies for third-order tensors, *Linear Algebra Appl.,* **435**(3), 641–658.

Kreimer, N., Stanton, A. & Sacchi, M.D., 2013. Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction, *Geophysics,* **78**(6), 273–284.

Lane, M.C., 1994. An implementation of 3-D seismic binning, *CREWES Res. Rep.,* **6**(24), 5.

Li, H., Li, W., Han, G. & Liu, F., 2018. Coupled tensor decomposition for hyperspectral pansharpening, *IEEE Access,* **6,** 34 206–34 213.

Liu, X.-Y., Aeron, S., Aggarwal, V. & Wang, X., 2020. Low-tubal-rank tensor completion using alternating minimization, *IEEE Trans. Inform. Theory,* **66**(3), 1714–1737.

Lou, Y., Niu, T., Jia, X., Vela, P.A., Zhu, L. & Tannenbaum, A.R., 2013. Joint CT/CBCT deformable registration and CBCT enhancement for cancer radiotherapy, *Med. Image Anal.,* **17**(3), 387–400.

Lumley, D.E., 2001. Time-lapse seismic reservoir monitoring, *Geophysics,* **66**(1), 50–53.

Martin, C.D., Shafer, R. & Larue, B., 2013. An order-*p* tensor factorization with applications in imaging, *SIAM J. Scient. Comput.,* **35**(1), A474–A490.

Mu, C., Huang, B., Wright, J. & Goldfarb, D., 2014. Square deal: lower bounds and improved relaxations for tensor recovery, *Proc. Mach. Learn. Res.,* **32,** 73–81.

Nguyen, P.K.T., Nam, M.J. & Park, C., 2015. A review on time-lapse seismic data processing and interpretation, *Geosci. J. (Seoul),* **19**(2), 375–392.

Oghenekohwo, F., Wason, H., Esser, E. & Herrmann, F.J., 2017. Low-cost time-lapse seismic with distributed compressive sensing—part 1: exploiting common information among the vintages, *Geophysics,* **82**(3), P1–P13.

Oropeza, V. & Sacchi, M., 2011. Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis, *Geophysics,* **76,** V25–V32.

Pevzner, R., Shulakova, V., Kepic, A. & Urosevic, M., 2011. Repeatability analysis of land time-lapse seismic data: CO2CRC Otway pilot project case study, *Geophys. Prospect.,* **59**(1), 66–77.

Popa, J., Minkoff, S. & Lou, Y., 2019. Improving seismic data completion and efficiency using tensors, in *SEG Technical Program Expanded Abstracts 2019,* pp. 4034–4038, Society of Exploration Geophysicists.

Popa, J., Minkoff, S. & Lou, Y., 2020. Improving seismic data completion via low-rank tensor optimization, in *SEG Technical Program Expanded Abstracts 2020,* pp. 2774–2778, Society of Exploration Geophysicists.

Popa, J., Minkoff, S.E. & Lou, Y., 2021. An improved seismic data completion algorithm using low-rank tensor optimization: cost reduction and optimal data orientation, *Geophysics,* **86**(3), V219–V232.

Rickett, J.E. & Lumley, D.E., 2001. Cross-equalization data processing for time-lapse seismic reservoir monitoring: a case study from the Gulf of Mexico, *Geophysics,* **66**(4), 1015–1025.

Sacchi, M.D. & Cheng, J., 2017. 5D reconstruction via robust tensor completion, in *Proceedings of the GeoConvention,* Calgary, Canada, May 15–19, pp. 1–5.

Schonewille, M., 2005. A modeling study on seismic data regularization for time-lapse applications, in *SEG Technical Program Expanded Abstracts 2003,* pp. 1537–1540, Society of Exploration Geophysicists.

Semerci, O., Hao, N., Kilmer, M.E. & Miller, E.L., 2014. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography, *IEEE Trans. Image Proc.,* **23**(4), 1678–1693.

Stanton, A., Kreimer, N., Bonar, D., Naghizadeh, M. & Sacchi, M., 2012. A comparison of 5D reconstruction methods, in *SEG Technical Program Expanded Abstracts 2012,* pp. 1–5, Society of Exploration Geophysicists.

Vivone, G., Dalla Mura, M., Garzelli, A., Restaino, R., Scarpa, G., Ulfarsson, M.O., Alparone, L. & Chanussot, J., 2021. A new benchmark based on recent advances in multispectral pansharpening: revisiting pansharpening with classical and emerging pansharpening methods, *IEEE Geosci. Remote Sens. Mag.,* **9**(1), 53–81.

Xu, Y., Hao, R., Yin, W. & Su, Z., 2015. Parallel matrix factorization for low-rank tensor completion, *Inverse Problems Imag.,* **9**(2), 601–624.

Zhang, Z. & Aeron, S., 2017. Exact tensor completion using T-SVD, *IEEE Trans. Signal Proc.,* **65**(6), 1511–1526.

Zhang, Z., Ely, G., Aeron, S., Hao, N. & Kilmer, M., 2014. 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,* 23–28 June 2014, Columbus, OH, USA, pp. 3842–3849.

Zhou, W. & Lumley, D., 2021. Nonrepeatability effects on time-lapse 4D seismic full-waveform inversion for ocean-bottom node data, *Geophysics,* **86**(4), R547–R561.

APPENDIX: TENSOR ALGEBRA

To define the product between two tensors we use the t-product described by Kilmer & Martin (2011) for order-3 tensors and later generalized by Martin *et al.* (2013). For two order-3 tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times m \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{m \times n_2 \times n_3}$, the product $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be computed in three steps. First, perform the Fourier transform along each tube of both tensors and denote the transformed tensors as $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, respectively. Secondly, compute the matrix product between each frontal slice of $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, and denote this resulting tensor as $\hat{\mathcal{C}}$. In the last step, apply the inverse Fourier transform to each tube of $\hat{\mathcal{C}}$, and denote this resulting tensor as $\mathcal{C}$, representing the product between $\mathcal{A}$ and $\mathcal{B}$.

We now generalize matrix properties that are relevant to the tSVD. A *diagonal tensor* has the property that each frontal slice is a diagonal matrix. The *transpose* of a tensor is denoted as $\mathcal{X}^T$ and is defined for order-3 tensors by taking the matrix transpose of each frontal slice, and then reversing the order of the transposed slices except for the first slice. The *identity* tensor, $\mathcal{I}$, has the identity matrix as its first frontal slice, with all other slices containing only zeros. The product of the identity tensor with any other tensor $\mathcal{X}$ results in the same tensor $\mathcal{X}$. A real-valued tensor is *orthogonal* if the product with its transpose results in the identity tensor, that is $\mathcal{X} * \mathcal{X}^T = \mathcal{I}$ and $\mathcal{X}^T * \mathcal{X} = \mathcal{I}$.

The *Frobenius norm* can be generalized from matrices to tensors (Kilmer & Martin 2011). It is defined as the square root of the sum of the absolute values of the entries squared,

$$||\mathcal{X}||_{\mathrm{F}} = \sqrt{\sum_{i,j,k} |\mathcal{X}_{i,j,k}|^2}. \tag{A1}$$

The Frobenius norm is used when comparing the relative error in reconstruction and for the regularization terms in the TNN-ADMM equations presented in the TNN-ADMM subsection.