GEOPHYSICS, VOL. 86, NO. 3 (MAY-JUNE 2021); P. V219–V232, 18 FIGS., 5 TABLES.
10.1190/GEO2020-0539.1

DOI:10.1190/geo2020-0539.1

An improved seismic data completion algorithm using low-rank tensor optimization: Cost reduction and optimal data orientation

Jonathan Popa[1], Susan E. Minkoff[1], and Yifei Lou[1]

ABSTRACT

Seismic data are often incomplete due to equipment malfunction, limited source and receiver placement at near and far offsets, and missing crossline data. Seismic data contain redundancies because they are repeatedly recorded over the same or adjacent subsurface regions, causing the data to have a low-rank structure. To recover missing data, one can organize the data into a multidimensional array or tensor and apply a tensor completion method. We can increase the effectiveness and efficiency of low-rank data reconstruction based on tensor singular value decomposition (tSVD) by analyzing the effect of tensor orientation and exploiting the conjugate symmetry of the multidimensional Fourier transform. In fact, these results can be generalized to any order tensor. Relating the singular values of the tSVD to those of a matrix leads to a simplified analysis, revealing that the most square orientation gives the best data structure for low-rank reconstruction. After the first step of the tSVD, a multidimensional Fourier transform, frontal slices of the tensor form conjugate pairs. For each pair, a singular value decomposition can be replaced with a much cheaper conjugate calculation, allowing for faster computation of the tSVD. Using conjugate symmetry in our improved tSVD algorithm reduces the runtime of the inner loop by 35%–50%. We consider synthetic and real seismic data sets from the Viking Graben Region and the Northwest Shelf of Australia arranged as high-dimensional tensors. We compare the tSVD-based reconstruction with traditional methods, projection onto convex sets and multichannel singular spectrum analysis, and we see that the tSVD-based method gives similar or better accuracy and is more efficient, converging with runtimes that are an order of magnitude faster than the traditional methods. In addition, we verify that the most square orientation improves recovery for these examples by 10%–20% compared with the other orientations.

INTRODUCTION

Seismic data collection is limited by acquisition obstacles and financial constraints that can prevent complete source/receiver coverage. These limitations restrict the placement of receivers at extreme offsets and constrain our ability to acquire complete crossline data in marine environments. In addition, receiver malfunction can result in data with gaps between traces. Missing data negatively affect seismic processing methods, such as migration, full-waveform inversion, and amplitude variation with offset analysis (Sacchi and Liu, 2005; Hunt et al., 2010; Chi et al., 2015). The data completion problem refers to recovering the underlying true data given only partial observations. Seismic data often contain redundancies. Mathematically, redundant traces can be described as a set of linearly *dependent* vectors. By treating traces as columns of a matrix, data containing redundancies can be regarded as a low-rank matrix because the rank of a matrix is equal to the number of linearly *independent* rows or columns.

Candès and Recht (2009) demonstrate that signals, images, and other data represented as a matrix can be recovered by optimizing a low-rank model. They observe that directly minimizing rank is NP-hard and therefore infeasible. To circumvent this issue, they propose a model that minimizes the sum of singular values, also known as the nuclear norm, as an approximation or relaxation of rank. This approximation stems from the fact that the rank of a matrix also corresponds to the number of nonzero singular values given by its singular-value decomposition (SVD). Since the work of Candès and Recht (2009), many researchers have generalized low-rank models

Manuscript received by the Editor 24 July 2020; revised manuscript received 21 January 2021; published ahead of production 6 February 2021; published online 21 April 2021.
[1]The University of Texas at Dallas, 800 W. Campbell Road, Richardson, Texas 75080, USA. E-mail: jonathan.popa1@utdallas.edu (corresponding author); sminkoff@utdallas.edu; yifei.lou@utdallas.edu.

DOI:10.1190/geo2020-0539.1

and methods for the reconstruction of high-dimensional data sets. We consider seismic data organized into a high-dimensional array or tensor. For instance, prestack seismic data with x, y midpoints, x, y offsets, and time samples can be arranged into an order-5 tensor with a dimension for each of the five variables. Arranging data as a higher order tensor allows for efficient and more accurate recovery (Kilmer and Martin, 2011; Ng et al., 2017). Studies show that tensor completion can succeed at reconstructing seismic data, recovering hyperspectral images, completing videos, and denoising computerized tomography images (Semerci et al., 2014; Zhang et al., 2014; Kumar et al., 2015; Mohd Sagheer and George, 2019; Liu et al., 2020a).

Low-rank reconstruction techniques for tensors include transform-based and rank reduction methods. Examples of transform-based methods include projection onto convex sets (POCS) and minimum weighted norm interpolation (Abma and Kabir, 2006; Stanton et al., 2012; Sacchi et al., 2017). Additional transform-based methods have been developed to better analyze and interpolate irregularly spaced data such as least-squares wavelet analysis and interpolation by matching pursuit (Øzbek et al., 2009; Ghaderpour and Pagiatakis, 2017). Other methods such as the antileakage Fourier transform and antileakage least-squares spectral analysis were designed to mitigate spectral leakage caused by irregular sampling (Xu et al., 2005; Ghaderpour et al., 2018).

We focus on investigating rank reduction methods. For higher order tensors, there are several definitions of rank that stem from the choice of decomposition, such as the canonical decomposition/parallel factor rank, Tucker rank, tubal rank, tensor train rank, tensor ring rank, and tensor tree rank (Kolda and Bader, 2009; Ely et al., 2015; Long et al., 2019). These types of rank can be difficult to compute, let alone minimize. Two popular generalizations of matrix SVD are higher order SVD (HOSVD) and tensor SVD (tSVD), both of which have associated norms that generalize the nuclear norm and can be used as a relaxation for the notion of rank (Kilmer and Martin, 2011; Kreimer and Sacchi, 2012). There are alternative methods and algorithms that seek to reduce rank while avoiding SVD computations such as parallel matrix factorization, randomized QR decomposition, and tubal altmin (Cheng and Sacchi, 2015; Gao et al., 2015; Carozzi and Sacchi, 2017; Sacchi and Cheng, 2017; Liu et al., 2020b). Multichannel singular spectrum analysis (MSSA), also known as Cadzow reconstruction, aims to reduce the rank of block Hankel matrices formed from frequency slices (Trickett et al., 2010; Oropeza and Sacchi, 2011). A variant, multidimensional dealiased Cadzow reconstruction, has been shown to reduce the effects of aliasing (Naghizadeh and Sacchi, 2013). Kumar et al. (2015) discuss how singular values decay at different rates for different data domains, indicating that rank reduction techniques can be made more effective when combined with appropriate transforms.

Returning to the previously mentioned SVD generalization, the term HOSVD was introduced by De Lathauwer et al. (2000), who show that the Tucker (1966) model generalizes the matrix SVD to tensors of any dimension. Kreimer and Sacchi (2012) develop a reconstruction technique to recover missing data using low-rank approximations of an observation by truncating its HOSVD. They verify their model by reconstructing 4D patches of prestack data (x, y midpoints and x, y offsets). Several works have continued to build upon and improve the HOSVD model (Ng et al., 2017; Gao and Sacchi, 2018).

Distinct from HOSVD, Kilmer and Martin (2011) develop a framework for multiplying order-3 tensors via the tensor product (tProduct), leading to a generalization of the matrix SVD known as tSVD. A tensor's representation by tSVD depends on its orientation, that is, the ordering of its dimensions. This dependence is a distinction from HOSVD, which is orientation-independent (Ely et al., 2015). Martin et al. (2013) further generalize the tSVD to tensors of any dimension. Ely et al. (2013) generalize the nuclear norm, defining the tensor nuclear norm (TNN) as the sum of singular values given by the tSVD. Furthermore, they formulate a low-rank tensor model that minimizes TNN and solve the model via the alternating direction method of multipliers (ADMM) algorithm, successfully reconstructing 4D and 5D prestack seismic data. Ely et al. (2015) improve the TNN model to reconstruct and denoise, showing improvement on 5D prestack seismic data. Various studies have defined variations of the TNN, still using the tSVD framework, and have applied these norm variations for low-rank reconstruction. Mu et al. (2020) propose a low-rank model using a weighted TNN and apply their model to image and video completion. Liu et al. (2020a) replace the nuclear norm by the L_p norm (for $p < 1$) referred to as the p-shrinkage nuclear norm, and they demonstrate recovery of color and hyperspectral images. Lu et al. (2020) develop a tensor robust principal component analysis model to recover low-rank tensors corrupted by sparse errors. Su et al. (2019) examine summing the TNN over each orientation of a 3D tensor, defining the sum of TNN to avoid the issue of tensor orientation.

Because the tSVD is orientation-dependent, Popa et al. (2020) study how orientation of order-3 tensors impacts the error in recovery. The orientation affects the shape of the frontal slices, submatrices within a tensor formed by fixing all but the first and second dimensions. The closer the ratio of the first and second dimensions is to one, the closer the frontal slices are to square matrices. They empirically find that the most square orientation produced the tensor of lowest rank based on the normalized cumulative sum of singular values. Popa et al. (2020) apply the low-rank TNN model to reconstruct 3D seismic data and find that this orientation produced the smallest error in recovery. Apart from tSVD, Mu et al. (2014) use a model based on reshaping a tensor into a matrix and find that square matrices improved low-rank recovery. Gao et al. (2017) use this idea of reshaping tensors into approximately square matrices to improve parallel matrix factorization, developing parallel square matrix factorization.

Popa et al. (2019) examine the effect of an order-3 tensor's orientation on the runtime of the tSVD algorithm. They find that the most square orientation results in an optimal runtime. Lu et al. (2020) and Popa et al. (2020) demonstrate that, for order-3 tensors, the tSVD runtime can be further reduced, independent of orientation, by using the conjugate symmetry produced by the Fourier transform in the first step of the tSVD algorithm.

In this work, we improve the TNN-ADMM model for seismic data reconstruction based on low-rank tSVD. We focus on the reconstruction problem without denoising. We demonstrate the advantages of TNN-ADMM over other reconstruction methods, analyze the impact of orientation on recovery for tensors of arbitrary order, and improve the runtime of tSVD by using conjugate symmetry. Specifically, our contributions include the following:

1) Compared with two conventional reconstruction methods, POCS and MSSA, for synthetic and real data, we find that TNN-ADMM provides comparable or improved accuracy and is the most efficient among these methods.

2) We discover that, for tensors of any order, the most square orientation provides the best data structure for the low-rank model, which allows for improved recovery by permuting the dimensions of input data.
3) We speed up tSVD-based reconstruction algorithms for any order of tensor by using the conjugate symmetry of the multidimensional Fourier transform. This conjugate symmetry can be applied to other tensor operations computed in the Fourier domain such as the tensor product.

THEORY AND METHOD

Tensor algebra background

When seismic data are recorded, acquisition limitations may restrict where they may be acquired. For example, the placement of near- and far-offset receivers or constraints on the spacing between receivers will lead to recorded data gaps. Seismic streamer surveys for marine acquisition may have missing crossline data. Additional measuring device issues such as receiver malfunction can also contribute to missing data. Figure 1a and 1b displays a gather from the Northwest Shelf (NWS) of Australia and the decimated observation after 60% of the data have been manually removed, respectively. Given such partial observations, we seek to recover the missing data samples and traces.

Mathematically, matrices provide a natural means of representing data in two dimensions. Seismic data can also be organized into higher dimensional structures (e.g., time-lapse data or data acquired using multiple sources and receivers can be considered to have more than two dimensions). Such high-dimensional data can naturally be arranged into a multidimensional array or tensor. We use script letters to denote tensors, such as $\mathcal{X}$, and capital letters for matrices, such as X. The *order* of a tensor refers to its dimension. For example, we can consider a sail line of seismic streamer data as an order-4 tensor, with dimensions corresponding to time, offset, streamer position, and shot position. Figure 2 illustrates an order-4 tensor corresponding to the NWS data, in which each colored cube represents the data collected at a different shot position. Submatrices within a cube contain the gathers from each streamer.

A partial observation is modeled mathematically using a *sampling operator*. A sampling operator can be represented as a matrix or tensor with dimensions matching the data's dimensions. The sampling operator contains ones (zeros) where data are recorded (missing). Letting $\mathcal{X}$ represent the underlying true seismic data, we consider the partial observation $\mathcal{Y}$ as the result of sampling operator $\mathcal{A}$ acting on $\mathcal{X}$; that is,

$$\mathcal{Y} = \mathcal{A}(\mathcal{X}). \tag{1}$$

A sampling operator acts by Hadamard or entry-wise product, where we multiply each entry of $\mathcal{X}$ by the corresponding entry of $\mathcal{A}$ (Kreimer and Sacchi, 2012).

Seismic data contain many redundancies, which results in a low-rank array in 2D. To recover the true data $\mathcal{X}$, we solve the optimization problem

$$\min h(\mathcal{X}) \quad \text{s.t. } \mathcal{Y} = \mathcal{A}(\mathcal{X}), \tag{2}$$

where h denotes a complexity measure. The complexity measure serves as a constraint on $\mathcal{X}$ because the inverse problem to find $\mathcal{X}$ given $\mathcal{Y}$, e.g., $\mathcal{Y} = \mathcal{A}(\mathcal{X})$, is ill-posed. If we wish to recover the data $\mathcal{X}$ of lowest rank that satisfy the data-matching constraint $\mathcal{Y} = \mathcal{A}(\mathcal{X})$, the complexity measure h should be chosen as rank. However, minimizing rank is NP-hard; hence, an approximation of rank is needed (Candès and Recht, 2009). To approximate rank, one often makes use of the SVD. The singular values are significant in that the largest singular values contain the most important information about the matrix (Strang, 1980). Low-rank matrices are

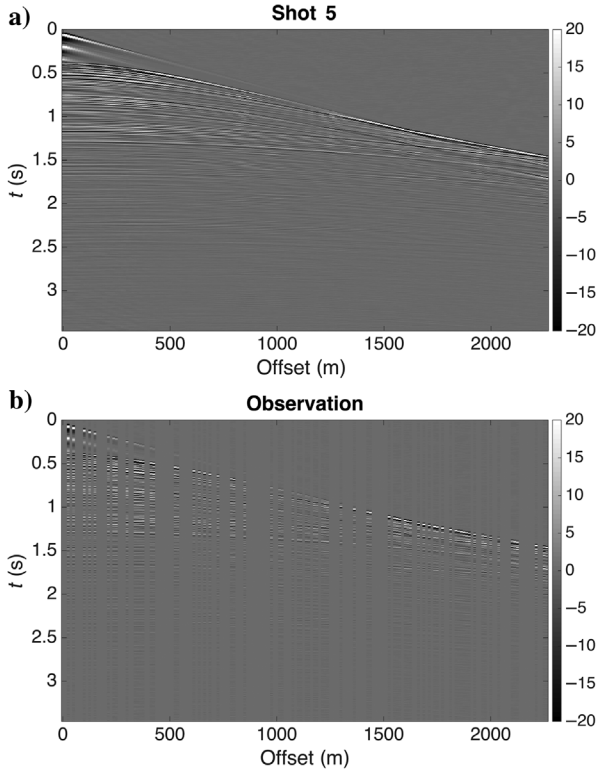


Figure 1. (a) A gather of the NWS interpolated to have a single zero-feather streamer and (b) the decimated observation with 60% of the columns removed at random.

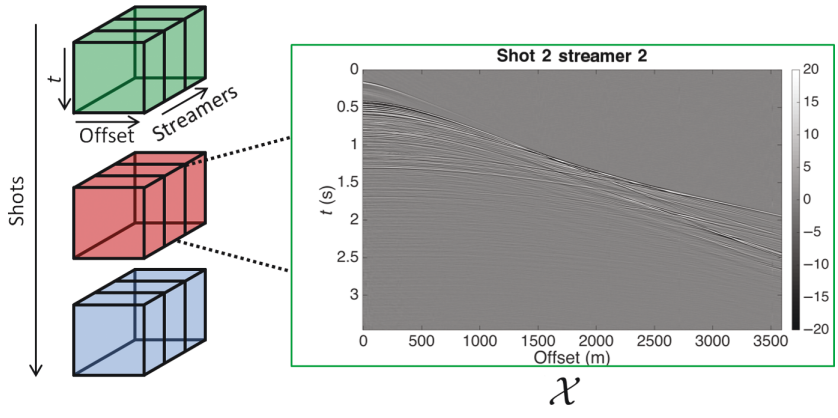


Figure 2. Illustration of the NWS data arranged as an order-4 tensor with dimensions for time, offset (receiver position), streamer position, and shot position.

characterized by a small percentage of the singular values containing most of the matrices' information. Thus, the sum of the singular values, also known as the nuclear norm, denoted $\|\cdot\|_{\text{nuc}}$, is commonly used to approximate rank (Candès and Recht, 2009).

We next discuss the theory for arbitrary order-p tensors for $p \geq 3$, which allows us to explain the generalizations of SVD and the nuclear norm for tensors of any order. For an order-p tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$, the elements are indexed by $\mathcal{X}_{k_1,k_2,\ldots,k_p}$, with $k_j \in \{1, \ldots, n_j\}$ for all dimensions $j = 1, 2, \ldots, p$. A *frontal slice* is a submatrix formed by fixing all but the first two dimensions. We denote a frontal slice as $\mathcal{X}_i$ for $i = 1, 2, \ldots, N$, where $N = n_3 n_4 \ldots n_p$ is the total number of frontal slices.

The tSVD generalizes the matrix SVD and decomposes a tensor into a product of three tensors (Martin et al., 2013),

$$\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^*. \tag{3}$$

The tensors $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \cdots \times n_p}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3 \times \cdots \times n_p}$ are unitary. The term $\mathcal{S}$ is a diagonal tensor, where each frontal slice is a diagonal matrix. The multidimensional Fourier transform of $\mathcal{S}$ over dimensions 3 to p is denoted as $\hat{\mathcal{S}}$. The diagonal of each frontal slice of $\hat{\mathcal{S}}$ is positive and descending. Due to the similarity to matrix SVD, we will refer to the diagonal values in $\hat{\mathcal{S}}$ as the singular values of $\mathcal{X}$ (for more details, see Appendix A).

TNN generalizes the matrix nuclear norm and is defined by summing the nuclear norms of each frontal slice of the tensor in the Fourier domain,

$$\|\mathcal{X}\|_{\text{TNN}} = \sum_{i=1}^{N} \|\hat{\mathcal{X}}_i\|_{\text{nuc}}. \tag{4}$$

Because the nuclear norm provides a relaxation of rank for matrices, the TNN analogously provides a relaxation of rank for tensors and produces a convex objective function (Candès and Recht, 2009; Ely et al., 2015). Hence, using TNN as our choice of complexity measure in equation 2, we consider the optimization problem

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

We solve equation 5 using ADMM because it converges quickly and can be implemented relatively easily (Boyd et al., 2010; Ely et al., 2015). We show that ADMM performs favorably relative to two traditional data completion methods in the "Numerical results" section, providing accurate results at a reduced computational expense.

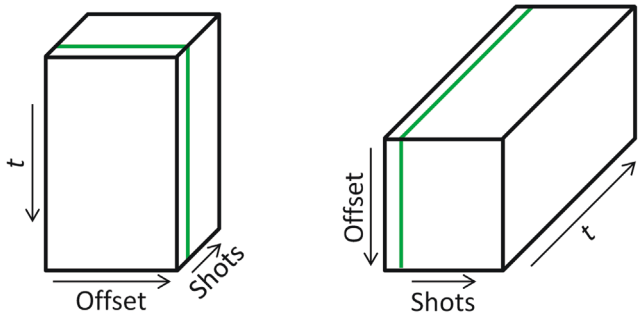


Figure 3. Illustration of two orientations of an order-3 tensor with dimensions for time, offset, and shot position. The green line highlights the same gather in each orientation.

ADMM alternates between satisfying the low-rank requirement imposed by the TNN penalty term and satisfying the data-matching constraint by splitting the variable $\mathcal{X}$ into two. Applying this method to solve equation 5 results in the following iterative updates (Ely et al., 2015):

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X}} \left\{ 1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})} + \frac{\rho}{2} \|\mathcal{X} - (\mathcal{Z}^k - \mathcal{B}^k)\|_{\text{F}}^2 \right\}, \tag{6}$$

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

$$\mathcal{B}^{k+1} = \mathcal{B}^k + (\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}), \tag{8}$$

where k denotes the iteration, $\rho > 0$ is a step-length parameter, and $\|\cdot\|_{\text{F}}$ is the Frobenius norm. The indicator function is denoted by $1_{\mathcal{Y}=\mathcal{A}(\mathcal{X})}$, which takes the value zero if the relation in the subscript is satisfied and ∞ otherwise. The variable $\mathcal{X}$ is required to satisfy the data-matching constraint $\mathcal{Y} = \mathcal{A}(\mathcal{X})$ by the indicator function. The variable $\mathcal{Z}$ captures the TNN penalty, and $\mathcal{B}$ is the dual variable. In the next section, we seek to improve TNN-ADMM by understanding the impact of orientation on arbitrary order-p tensors.

Relating tSVD to SVD and the best orientation for low-rank recovery

We examine how the accuracy of low-rank data recovery is affected by tensor orientation. Permuting the dimensions of a tensor, similar to a matrix transpose, results in different orientations. Figure 3 illustrates two orientations of an order-3 tensor with dimensions for time, offset, and shot position. We relate tensor $\mathcal{X}$ to a matrix sharing the same singular values and analyze how the orientation affects this matrix. In doing so, we determine the optimal orientation for the low-rank TNN model.

For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$, let $\bar{X} \in \mathbb{R}^{n_1 N \times n_2 N}$ be a block diagonal matrix, with the diagonal elements corresponding to the frontal slices of $\hat{\mathcal{X}}$ indexed from $i = 1, \ldots, N$, e.g.,

$$\bar{X} = \begin{bmatrix} \hat{\mathcal{X}}_1 & & & \\ & \hat{\mathcal{X}}_2 & & \\ & & \ddots & \\ & & & \hat{\mathcal{X}}_N \end{bmatrix}. \tag{9}$$

For a tensor with tSVD $\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^*$, it has been shown in the construction of the tSVD by Martin et al. (2013) and Lu et al. (2020) that

$$\bar{X} = \bar{U}\,\bar{S}\,\bar{V}^*. \tag{10}$$

This equation gives the SVD of $\bar{X}$. Thus, tensor $\mathcal{X}$ and its corresponding block diagonal matrix $\bar{X}$ share the same singular values. Hence, minimizing the TNN of $\mathcal{X}$ is equivalent to minimizing the

nuclear norm of $\bar{X}$. This connection simplifies the tensor analysis to analyze a matrix.

The dimensions of the block diagonal matrix are dependent on the orientation of the tensor, as $\bar{X} \in \mathbb{R}^{n_1 N \times n_2 N}$. Hence, if $n_1 \gg n_2$ or $n_2 \gg n_1$, then $\mathcal{X}$ has very rectangular or skinny frontal slices and $\bar{X}$ is a skinny matrix. Likewise, if $\mathcal{X}$ has approximately square frontal slices, then $\bar{X}$ is approximately square. The most square orientation occurs when the ratio n_1/n_2 is the closest to one among all possible pairs of dimensions. Skinny matrices are often of low rank; hence, applying a low-rank model does not improve data reconstruction. Consequently, square matrices provide the best data structure for low-rank models. In addition, this orientation coincides with the most cost efficient orientation when n_1 and n_2 are the two smallest dimensions and their ratio is the closest to one (Popa et al., 2019).

We also consider the physical meaning of this orientation. For seismic data, time is often the largest dimension by a significant margin resulting in the most square orientation having frontal slices formed over spatial dimensions such as the receiver position and shot location. If data are collected from an area with lateral continuity, we can expect the data to be of lower rank along the spatial dimensions than with respect to time. In such cases, we would expect the most square orientation to have the lowest rank frontal slices among all the orientations. We provide empirical evidence to support the advantages of the most square orientation and compare the recovery under different orientations in the "Numerical results" section.

Improving runtime

We now discuss a means of speeding up the runtime of the tSVD algorithm independent of orientation. Lu et al. (2020) and Popa et al. (2019) demonstrate for order-3 tensors that operations in the Fourier domain acting on all frontal slices, such as tSVD and tProduct, can be implemented more efficiently by taking conjugates of the appropriate slices to avoid redundant calculations. Here, we show how this conjugate symmetry extends to the general case for tensors of any order.

The tSVD algorithm can be implemented in three steps as shown in Algorithm 1. In the first loop of the algorithm, one computes the multidimensional Fourier transform of $\mathcal{X}$ over dimensions 3 to p, resulting in $\hat{\mathcal{X}}$. We use "fft$(\hat{\mathcal{X}}, j)$" to denote the Fourier transform of $\hat{\mathcal{X}}$ along dimension j and "ifft" for the inverse Fourier transform. In the second loop, one computes the SVD for each frontal slice of $\hat{\mathcal{X}}$ and stores the results in $\hat{\mathcal{U}}$, $\hat{\mathcal{S}}$, and $\hat{\mathcal{V}}$. In the last loop, we compute the multidimensional inverse Fourier transform of $\hat{\mathcal{U}}$, $\hat{\mathcal{S}}$, and $\hat{\mathcal{V}}$ to obtain the three tensors that comprise the tSVD of $\mathcal{X}$.

To speed up this algorithm, we can make use of the conjugate symmetry property of the multidimensional Fourier transform (for the 2D definition of conjugate symmetry, see Briggs and Henson, 1995). If $\mathcal{X}$ is real-valued, then values of $\hat{\mathcal{X}}$ form conjugate pairs by the relation

$$\hat{\mathcal{X}}_{k_1,k_2,k_3,\ldots,k_p} = \hat{\mathcal{X}}^*_{k_1,k_2,m_3,\ldots,m_p} \quad \forall\, k_1, k_2, \tag{11}$$

where $m_j = [-(k_j - 1) \bmod n_j] + 1$ for $j = 3, 4, \ldots, p$. This conjugate symmetry results in *most* frontal slices having an entry-wise conjugate pair; that is, for frontal slice $\hat{\mathcal{X}}_K$ there exists a frontal slice $\hat{\mathcal{X}}_M$ such that $\hat{\mathcal{X}}_M = \text{conj}(\hat{\mathcal{X}}_K)$, where "conj" denotes the entry-wise conjugate. A slice does not have a conjugate pair if $m_j = k_j$ for all $j = 3, 4, \ldots, p$. Figure 4 illustrates the conjugate symmetry for a 4D $n_1 \times n_2 \times 3 \times 3$ tensor using three cubes, each with three frontal slices. Slices of the same color are entry-wise conjugate pairs. For conjugate pairs of slices, we can avoid redundant calculations in the tSVD and tProduct algorithms, replacing an SVD and matrix product, respectively, with cheaper conjugate computations.

For computing the tSVD, we consider a conjugate pair of frontal slices $\hat{\mathcal{X}}_K$ and $\hat{\mathcal{X}}_M$, $K < M$. For the first frontal slice of the pair, $\hat{\mathcal{X}}_K$, we calculate the matrix SVD as $\hat{\mathcal{X}}_K = \hat{\mathcal{U}}_K \hat{\mathcal{S}}_K \hat{\mathcal{V}}^*_K$. For the second slice of the pair, $\hat{\mathcal{X}}_M$, we can take the entry-wise conjugate of each term of the decomposition, $\hat{\mathcal{X}}_M = \text{conj}(\hat{\mathcal{U}}_K)\hat{\mathcal{S}}_K \text{conj}(\hat{\mathcal{V}}^*_K)$ (because the singular values are real, taking the conjugate of $\hat{\mathcal{S}}_K$ is unnecessary).

To implement tSVD efficiently, we first compute a mapping between conjugate slices. We store this mapping in a vector $v \in \mathbb{R}^N$. Each element of v contains an index of a frontal slice. For the first slice of a conjugate pair $\hat{\mathcal{X}}_K$, we set $v(K) = K$. For the second slice in a pair $\hat{\mathcal{X}}_M$, we set $v(M) = K$ to signify that we can take the conjugate of the SVD of slice $\hat{\mathcal{X}}_K$. For a slice $\hat{\mathcal{X}}_i$ without a conjugate pair, we set $v(i) = i$. We note that the mapping vector v needs to be computed only once for a set of dimensions $k_3, k_4, \ldots, k_p$, allowing for efficient implementation when computing the tSVD for multiple tensors of the same size.

Applying conjugate symmetry and the mapping vector, we present our improved tSVD in Algorithm 2. In the second loop of the algorithm, we check if a slice is the first in a pair (or if it

Algorithm 1. Original tSVD algorithm.

1: **Input:** $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$
2: $N = n_3 n_4 \ldots n_p$
3: $\hat{\mathcal{X}} = \mathcal{X}$
4: **for** j = 3 to p **do**
5: $\quad \hat{\mathcal{X}} = \text{fft}(\hat{\mathcal{X}}, j)$
6: **end for**
7: **for** i = 1 to N **do**
8: $\quad [\hat{\mathcal{U}}_i, \hat{\mathcal{S}}_i, \hat{\mathcal{V}}_i] = \text{svd}(\hat{\mathcal{X}}_i)$
9: **end for**
10: **for** j = 3 to p **do**
11: $\quad \hat{\mathcal{U}} = \text{ifft}(\hat{\mathcal{U}}, j)$
12: $\quad \hat{\mathcal{S}} = \text{ifft}(\hat{\mathcal{S}}, j)$
13: $\quad \hat{\mathcal{V}} = \text{ifft}(\hat{\mathcal{V}}, j)$
14: **end for**

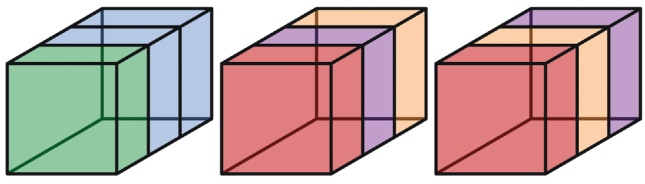


Figure 4. We illustrate an order-4 tensor with dimensions $n_1 \times n_2 \times 3 \times 3$ using three cubes, each with three frontal slices. Slices of the same color are entry-wise conjugate.

has no pair) using the mapping vector v, in which case we compute and store its SVD. Otherwise, we store the conjugate of the previously computed SVD of the appropriate slice, thus replacing an SVD computation with a much cheaper conjugate computation. The tProduct can be implemented similarly, using the same mapping vector v to replace matrix product computations with conjugates of appropriate slices.

To estimate the cost savings of replacing SVD calculations with conjugates, we examine the cost of each operation and how many slices have a conjugate pair. As previously stated, a slice does not have a conjugate pair if $m_j = k_j$ for all $j = 3, 4, \ldots, p$. This equality occurs for $k_j = 1$ and $k_j = n_j/2 + 1$ when n_j is even. Taking all combinations gives the number of slices that do not have a conjugate pair as 2^α, where α is the number of dimensions with an even value among $n_3, n_4, \ldots, n_p$. Hence, the number of slices with a conjugate pair is $N - 2^\alpha$. For half of these, $(N - 2^\alpha)/2$ many slices, we can perform an inexpensive conjugate computation in place of an expensive SVD calculation in the second loop of the tSVD algorithm. Thus, the cost of the inner loop of the tSVD algorithm can be reduced by approximately 50%. We demonstrate the improvement of our algorithm on synthetic and real data in the "Numerical results" section.

NUMERICAL RESULTS

We illustrate our theoretical results numerically using synthetic and field data sets. The synthetic data contain three linear events and consist of 100 gathers, each with 60 traces and 700 time steps, and they are shown in Figure 5a. For the field data experiments, we use data sets from the Viking Graben (VG) Region and NWS of Australia. We show a schematic of the acquisition geometry of the VG seismic streamer data in Figure 6. From the VG data set, we use a subset of 100 common-midpoint gathers with a bin size of 12.5 m. Each gather has 60 traces covering offsets from 262 to 3212 m with a 6 s time window (4 ms sampling).

The NWS data that we use contain gathers from 20 shot locations along one sail line with 25 m spacing. Gathers were recorded with 12 streamers per shot and 288 traces per streamer. Traces contain 1732 samples with a 2 ms time sampling. Figure 7 illustrates the acquisition geometry of the NWS data. For our 4D tensor experiments, we use gathers from five shot locations and refer to these data as NWS4D. In addition, we examine a subset of the NWS, interpolated from the full data set to have a single zero-feather streamer for each shot. We refer to this zero-feather data as NWS3D and use it for additional 3D tensor experiments. For our experiments, we operate on each of these data sets arranged into a single high-dimensional tensor. In general, splitting the data allows for efficient parallel computing but increases the error in reconstruction.

In the remainder of this section, we compare TNN-ADMM with two other standard methods used for seismic data reconstruction, POCS and MSSA (Abma and Kabir, 2006; Chen et al., 2016). We find that these three methods are comparable when applied to synthetic data, whereas TNN-ADMM performs better on real field data. In addition, we find that TNN-ADMM has the fastest

Algorithm 2. Improved tSVD algorithm using conjugate symmetry.

Inputs: $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$, $v \in \mathbb{R}^N$
$N = n_3 n_4 \ldots n_p$
$\hat{\mathcal{X}} = \mathcal{X}$
for j = 3 to p **do**
  $\hat{\mathcal{X}} = \text{fft}(\hat{\mathcal{X}}, j)$
end for
for i = 1 to N **do**
  if v(i) == i **then**
    $[\hat{\mathcal{U}}_i, \hat{\mathcal{S}}_i, \hat{\mathcal{V}}_i] = \text{svd}(\hat{\mathcal{X}}_i)$
  else
    $\hat{\mathcal{S}}_i = \hat{\mathcal{S}}_{v(i)}$
    $\hat{\mathcal{U}}_i = \text{conj}(\hat{\mathcal{U}}_{v(i)})$
    $\hat{\mathcal{V}}_i = \text{conj}(\hat{\mathcal{V}}_{v(i)})$
  end if
end for
for j = 3 to p **do**
  $\hat{\mathcal{U}} = \text{ifft}(\hat{\mathcal{U}}, j)$
  $\hat{\mathcal{S}} = \text{ifft}(\hat{\mathcal{S}}, j)$
  $\hat{\mathcal{V}} = \text{ifft}(\hat{\mathcal{V}}, j)$
end for

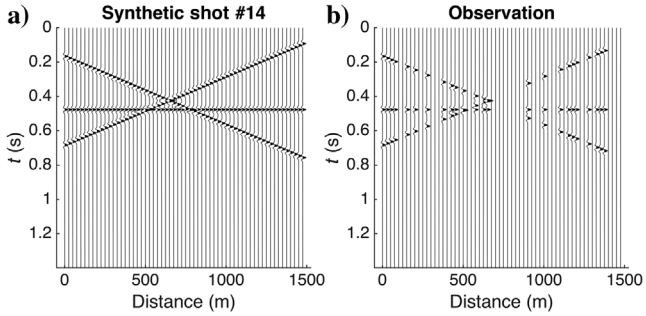


Figure 5. (a) A synthetic gather and (b) the decimated observation with 60% of the columns removed at random.

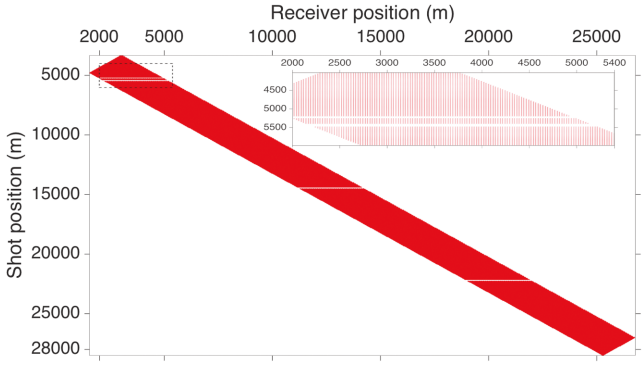


Figure 6. Acquisition map of the VG data (D. Lumley, personal communication, 2018). The horizontal and vertical axes correspond with the receiver and shot positions, respectively. A red point is placed at each location where data are recorded. The section of the plot marked with a dashed box in the upper left corner is shown enlarged in the upper right. We see a few horizontal bands for which no data are recorded at specific shot locations.

runtime to convergence among these methods. Then, we compare TNN-ADMM for each orientation of the order-3 and order-4 tensors. For the synthetic and real data examples, the most square orientation produces the least error. Finally, we compare the runtime of the original tSVD algorithm and our improved algorithm using conjugate symmetry. We observe significant runtime reduction by using conjugate symmetry.

Comparison of methods

In this paper, our focus is on improving the data reconstruction via TNN-ADMM. However, because this method is less well-known in the seismic community, we start by comparing TNN-ADMM with two other, more standard methods for seismic data reconstruction, namely, POCS and MSSA. For POCS, we implement the iterative algorithm described in Abma and Kabir (2006). The POCS method performs a Fourier transform over each dimension of a data tensor and removes the smallest amplitude frequencies in the *f-k* domain via thresholding. Although TNN-ADMM is dependent on data orientation, POCS is orientation independent.

The MSSA method transforms the data from the time to the frequency domain and then creates a block Hankel matrix from each frequency slice. In the noise-free case, these block Hankel matrices have rank equal to the number of linear events. Noise increases the rank of these matrices, hence by applying rank reduction the data can be reconstructed. Given the physical motivation behind MSSA to operate on frequency slices, we consider this method to be orientation-independent. For MSSA, we use the existing code made available by Chen et al. (2016).

First, we compare these three methods on the synthetic data set shown in Figure 5a. Using the sampled observation, Figure 5b, we attempt to reconstruct the original data. Each method has parameters tuned independently to optimize performance and is run until convergence. For TNN-ADMM, the parameter ρ affects the convergence speed (see equations 6 and 7). Smaller values of ρ can cause the algorithm to blow up or diverge, whereas, for larger values of ρ, the algorithm converges more slowly. Testing ρ values over the set $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, we determine the optimal ρ value to be 10^{-2} based on the least error after 50 initial iterations. For MSSA, the parameter N corresponds with the number of singular values in the block Hankel matrix kept at each iteration. Varying N over the set {2, 3, 5, 10}, we determine an optimal value of three based on the relative error after the first iteration. We define convergence here to be when the relative change in the recovered data between consecutive iterations is below a specified tolerance. For all of our experiments, we use a tolerance of 10^{-4}. For TNN-ADMM, we show results for the most square orientation. In Figure 8, we plot the recovered results and relative errors for each method. In Figure 8a–8c, the methods appear to have reconstructed the data. We amplify the relative error plots by a factor of 50 to better observe the error (Figure 8d–8f). TNN-ADMM and MSSA perform well with error of 0.3% or less. POCS does not perform as well, reconstructing with error of approximately 4%. In Figure 9, we show

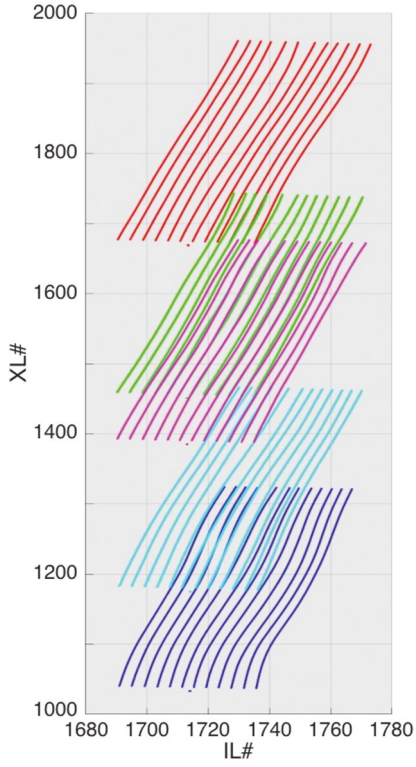


Figure 7. Acquisition map of the NWS (W. Zhou and D. Lumley, personal communication, 2020). The horizontal and vertical axes are labeled for the inline and crossline numbers, respectively. The diagonal lines represent streamers with each color corresponding to a shot location. The boat pulling the streamers is represented by a dot traveling in the negative crossline direction along a fixed inline number.

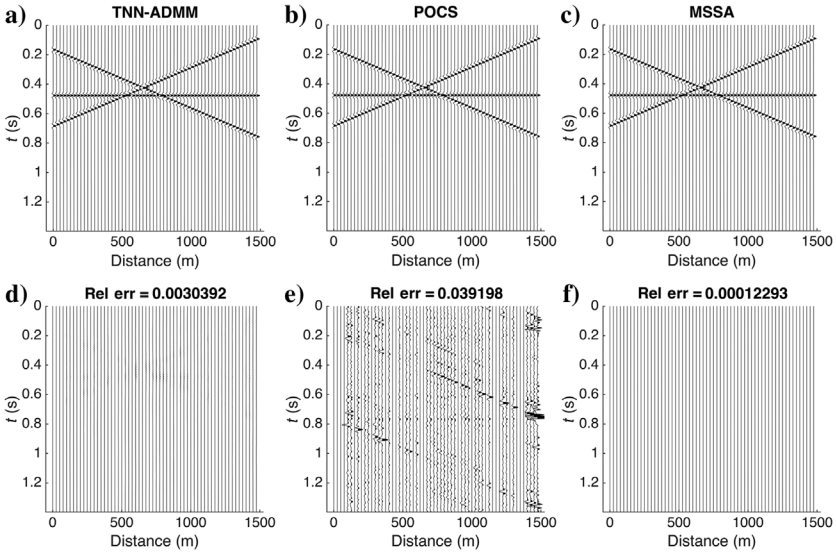


Figure 8. Comparison of recovery methods applied to synthetic data. (a–c) The recovered results. (d–f) The relative error. The relative error plots have been magnified by a factor of 50.

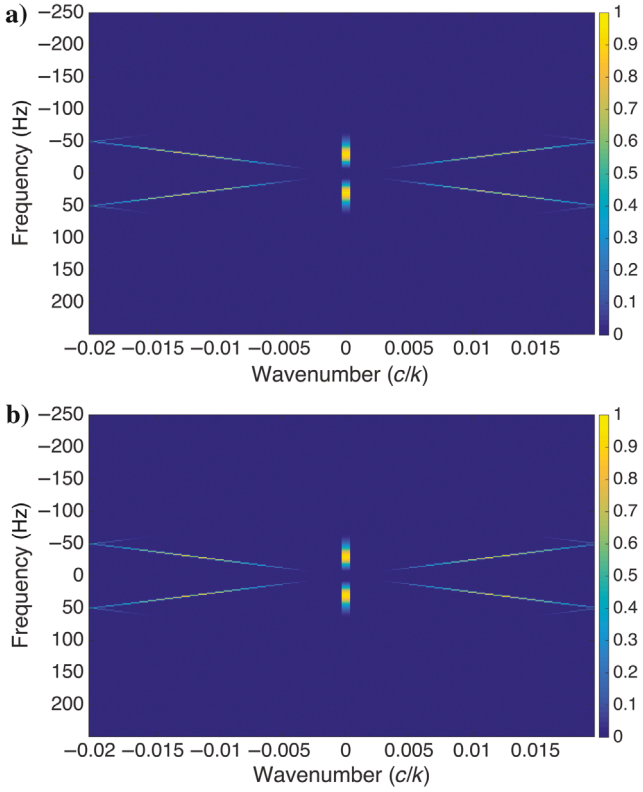


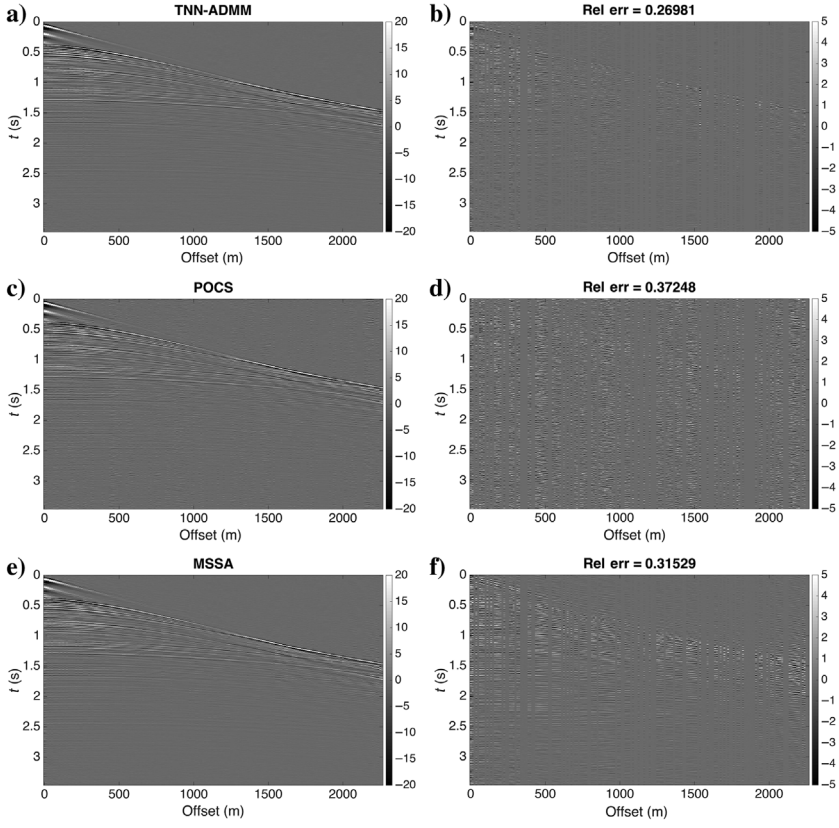
Figure 9. The *f-k* spectra of (a) the generated synthetic gather and (b) the recovered result for TNN-ADMM. The recovered result for ADMM, POCS, and MSSA give nearly identical *f-k* spectra; therefore, we only show the *f-k* spectrum for recovery using ADMM.

the *f-k* spectra of the generated data and the recovered result for TNN-ADMM. We observe that the *f-k* domain is successfully reconstructed in this case. The recovered result for ADMM, POCS, and MSSA gives nearly identical *f-k* spectra. Therefore, we only show the *f-k* spectrum for recovery using ADMM.

Then, we compare these three methods on field data using the NWS3D data shown in Figure 1. For this data set, we tune TNN-ADMM's ρ parameter over the set $\{10^{-5}, 10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 10^{-3}\}$ in the same manner as described above for the synthetic data set and determine 5×10^{-4} as the optimal value. We vary MSSA's parameter (N) over the set {2, 3, 5, 10, 15, 20} and determine an optimal value of 10 based on the relative error after two initial iterations. We present the recovered and relative error plots in Figure 10. We observe that POCS yields a noisy recovery and has the largest error among the methods, of approximately 37%. MSSA recovers the data with error of approximately 32%, whereas TNN-ADMM performs the best with error of approximately 27%. Figure 11 shows the relative error in reconstruction of the *f-k* spectra. We observe that TNN-ADMM has the least residual in this domain.

In Table 1, we compare the number of iterations and runtime until convergence for each method when applied to the synthetic and NWS3D data sets. We observe that TNN-ADMM is faster by at least an order of magnitude compared with POCS and MSSA. When factoring together algorithm efficiency and accuracy, TNN-ADMM has considerable advantages over other standard data reconstruction techniques. Having shown that TNN-ADMM is a suitable method for seismic data completion, we now turn to discussing how to best apply this method for recovery by taking orientation into account.

Figure 10. Comparison of recovery methods applied to the NWS3D field data: (a and b) TNN-ADMM, (c and d) POCS, and (e and f) MSSA. (a, c, and e) The recovered results. (b, d, and f) The relative error plots amplified by a factor of four.



Orientation results

As discussed previously, we expect the most square orientation of a data tensor to result in the best recovery. First, we present numerical experiments using the synthetic data shown in Figure 5. In Figure 12, we compare the normalized cumulative sum of singular values for each orientation. The cumulative sum is a partial sum calculated by adding the largest singular values of the tensor. From this plot, we can determine the percentage of data contained in the first few (largest) singular values. We also note the rate of decay in the singular values from this plot. The faster the singular values decay, the faster the sum approaches one. Because tensors with the first two dimensions transposed contain the same singular values, we only need to compare three orientations with distinct singular values out of the six possible choices. The most square orientation (the magenta curve) requires only a small percent of its singular values to describe 100% of its data, indicating that it has low rank. The other orientations do not exhibit this behavior, so we expect low-rank reconstruction to not perform as well in these cases.

To further support our hypothesis about how best to orient the tensor to recover data, we calculate the coherence for each orientation of the synthetic data. The coherence of a tensor measures the spread of its singular vectors with respect to the standard basis. Tensors with singular vectors that contain combinations of standard basis vectors are likely to have more entries observed by the sampling operator. Having more observed entries increases the probability of successful recovery (Candès and Recht, 2009; Zhang and Aeron, 2017). The more spread out that the singular vectors are, the lower the coherence value is (see Appendix B for details on computing tensor coherence as well as examples illustrating the correlation between coherence and the probability of recovery). In Table 2, we compare the coherence for each orientation. We observe that the most square orientation has the lowest coherence value, indicating that this orientation is the most likely to be successfully recovered given an observation with a limited number of entries.

Then, we apply a sampling operator that decimates 60% of the columns of data chosen at random in each gather. We apply ADDM to the decimated data and compare the recovered results for each orientation. In Figure 13, we compare the recovered gathers and relative error after convergence of ADMM for each orientation

Table 1. The number of iterations and total runtime until convergence for each method applied to the synthetic data set shown in Figure 5 and the NWS3D data shown in Figure 1.

	Synthetic		NWS3D	
Method	Iterations	Runtime (s)	Iterations	Runtime (s)
TNN-ADMM	115	1.146×10^2	98	1.7448×10^2
POCS	400	5.2639×10^3	400	2.3026×10^3
MSSA	18	2.5654×10^4	253	1.828×10^5

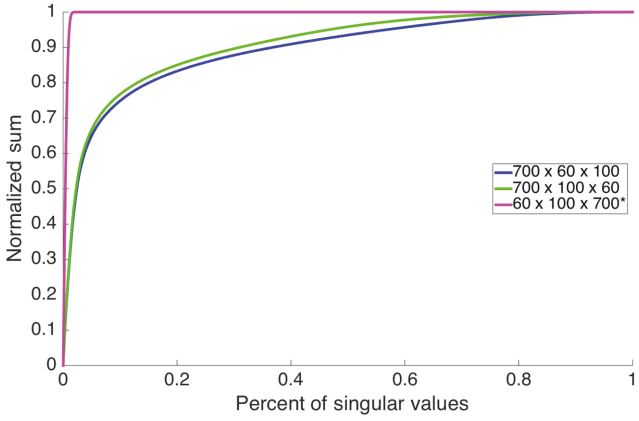


Figure 12. Normalized cumulative sum of singular values for three orientations of the synthetic data set shown in Figure 5. The most square orientation is denoted with an asterisk. Transposing the first two dimensions results in the same singular values; hence, we only compare three distinct orientations out of all six possible orientations.

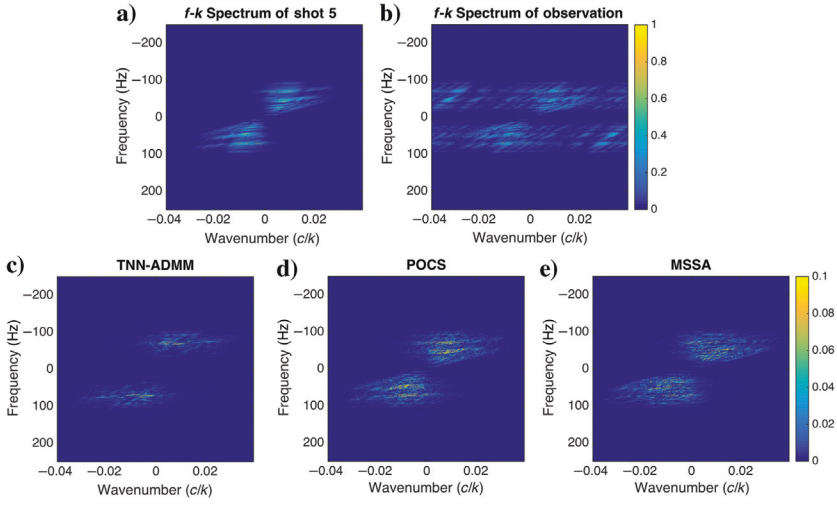


Figure 11. The *f-k* spectra of (a) a gather from the NWS3D data set and (b) the decimated observation. The relative error in the *f-k* spectra of the recovered result versus true for (c) TNN-ADMM, (d) POCS, and (e) MSSA. The relative error plots have been amplified by a factor of 10.

of synthetic data. The error is computed over the entire 3D volume relative to the original data before decimation. We observe that the most square orientation $60 \times 100 \times 700$ does significantly better in this example, with an error two orders of magnitude smaller than the other orientations. Although the error for the most square orientation is negligible, the other orientations have noticeable artifacts in their relative error plots. In particular, the second orientation $700 \times 100 \times 60$ does poorly near the right edge of the gather.

We next perform a similar experiment using the NWS3D and NWS4D data sets. For a 4D tensor, there are 24 different possible orientations. However, any orientations for which the first two dimensions are either the same or transposed, such as $12 \times 5 \times 288 \times 1732$ and $12 \times 5 \times 1732 \times 288$, give tensors that have the same singular values. Thus, we only need to consider six orientations with distinct singular values out of the 24 possible orientations for the 4D data. We plot the normalized cumulative sum of singular values for each distinct orientation of NWS3D and NWS4D in Figure 14. For the NWS3D data, we observe for the most square orientation that 10% of the largest singular values contains more than 70% of the information in the data, a greater percentage than the other two orientations. For the NWS4D data, we observe that the most square orientation exhibits a similar behavior. These observations indicate that the most square orientation has the lowest rank structure among all possible orientations because a small percent of singular values contains most of the information in the data. Furthermore, we compare the coherence of each orientation of NWS4D in the second column of Table 3. We observe that the most square orientation has the lowest coherence value, providing additional evidence that this orientation should result in the best recovery.

Table 2. The coherence for each orientation of the synthetic data set. We have marked the most square orientation with an asterisk.

Size	$\mu(\mathcal{X})$
$700 \times 60 \times 100$	1.5819
$700 \times 100 \times 60$	1.2288
$60 \times 100 \times 700^*$	1.0641

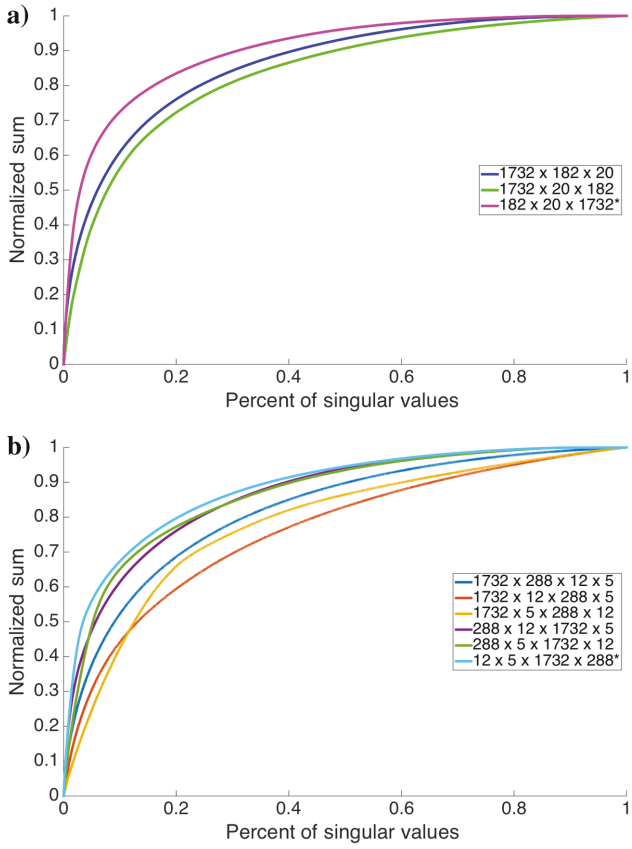


Figure 14. Normalized cumulative sum of singular values for orientations with distinct singular values of (a) NWS3D and (b) NWS4D. The most square orientation of each data set is denoted with an asterisk.

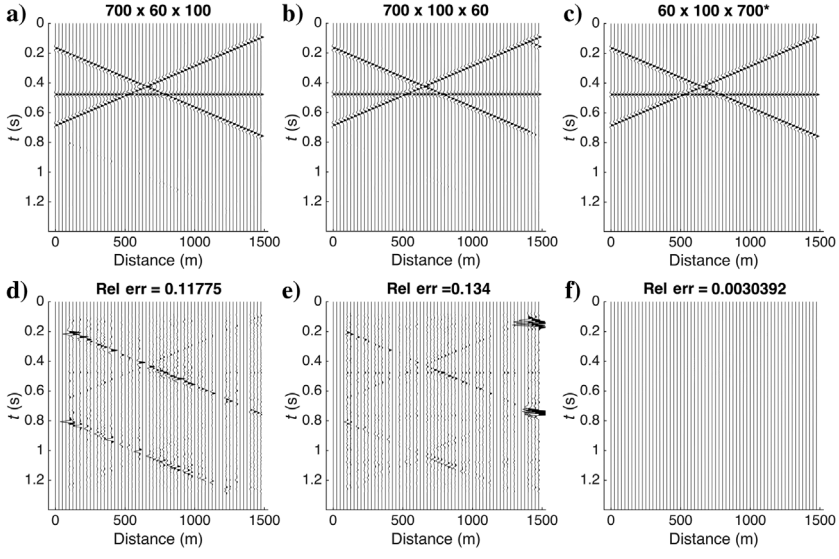


Figure 13. (a–c) Recovered and (d–f) relative error plots resulting from applying TNN-ADMM to each of the distinct orientations of the synthetic data shown in Figure 5. The final relative error, computed over the entire 3D volume, is displayed above each relative error plot. The relative error plots have been amplified by a factor of 10. Plots (c and f) correspond to the most square orientation.

We apply a sampling operator that decimates 60% of the columns of data chosen at random in each gather, producing observations like the gather shown in Figure 1b. Applying ADDM to the decimated data for each orientation, we compare the relative error in recovery after convergence of ADMM for NWS3D and NWS4D in Figure 15 and the third column of Table 3, respectively. The error is computed relative to the original data before decimation over the entire 3D or 4D volume. For ADMM, we set the parameter $\rho = 5 \times 10^{-4}$ for NWS3D and use $\rho = 2 \times 10^{-4}$ for NWS4D (see equations 6 and 7). In Figure 16, we present the recovered and relative error plots of a single gather for the three distinct orientations of NWS3D. For NWS4D, we present the recovered results in the left column of Figure 17 for the orientations $1732 \times 288 \times 12 \times 5$ and $12 \times 5 \times 1732 \times 288$. In the right column, we show the relative error plots for these two orientations, respectively, contrasting the best and worst recovered results. We observe that the most square orientations once again allow for the best low-rank recovery, more than 10% and 20% improvement over the worst case for NWS3D and NWS4D, respectively.

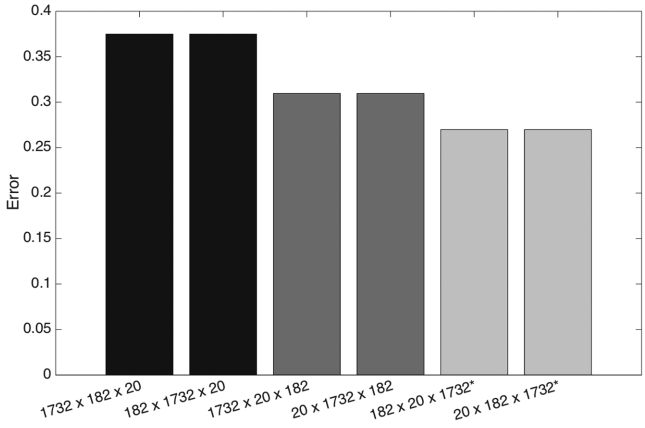


Figure 15. Comparison of the final relative error in recovery for each orientation of the NWS3D data. The most square orientation (denoted by the asterisk) gives the smallest error.

Table 3. The coherence and relative data completion error for each orientation of the NWS4D data listed in order of descending error. The most square orientation is denoted with an asterisk.

Size	$\mu(\mathcal{X})$	Relative error
$1732 \times 288 \times 12 \times 5$	2.6268	0.5162
$1732 \times 5 \times 288 \times 12$	17.8015	0.3846
$1732 \times 12 \times 288 \times 5$	16.2087	0.3795
$288 \times 5 \times 1732 \times 12$	18.6898	0.36424
$288 \times 12 \times 1732 \times 5$	13.3627	0.32741
$12 \times 5 \times 1732 \times 288^*$	1.3759	0.29232

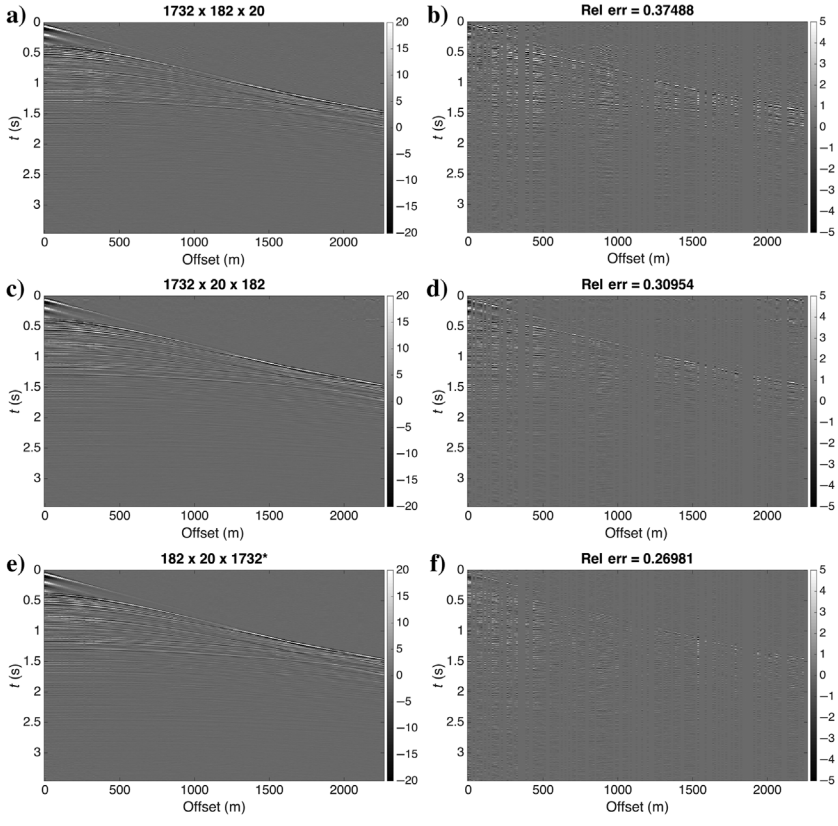


Figure 16. Recovered plots of a single NWS3D gather for the orientations (a) $1732 \times 182 \times 20$, (c) $1732 \times 20 \times 182$, and (e) $182 \times 20 \times 1732$. The relative error plots for each of these orientations are (b, d, and f), respectively. The final relative error, computed over the entire 3D volume, is displayed above each relative error plot. Plots (e and f) correspond to the most square orientation, denoted by an asterisk.

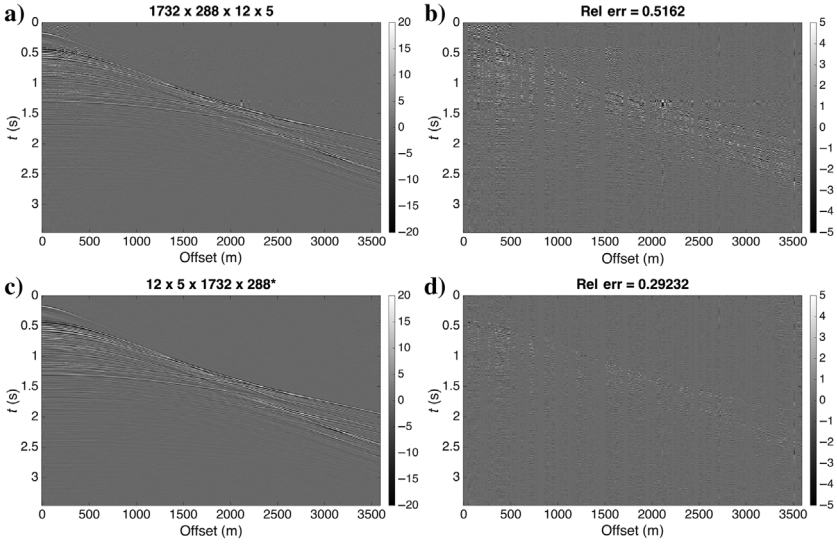


Figure 17. Recovered plots of a representative gather from the NWS for the orientations: (a) $1732 \times 288 \times 12 \times 5$ (worst case) and (c) $12 \times 5 \times 288 \times 1732$ (best case). The relative error plots for these orientations are (b and d), respectively. The relative error plots have been amplified by a factor of four. The final relative error, computed over the entire 4D volume, is displayed above each relative error plot. Plots (c and d) correspond to the most square orientation.

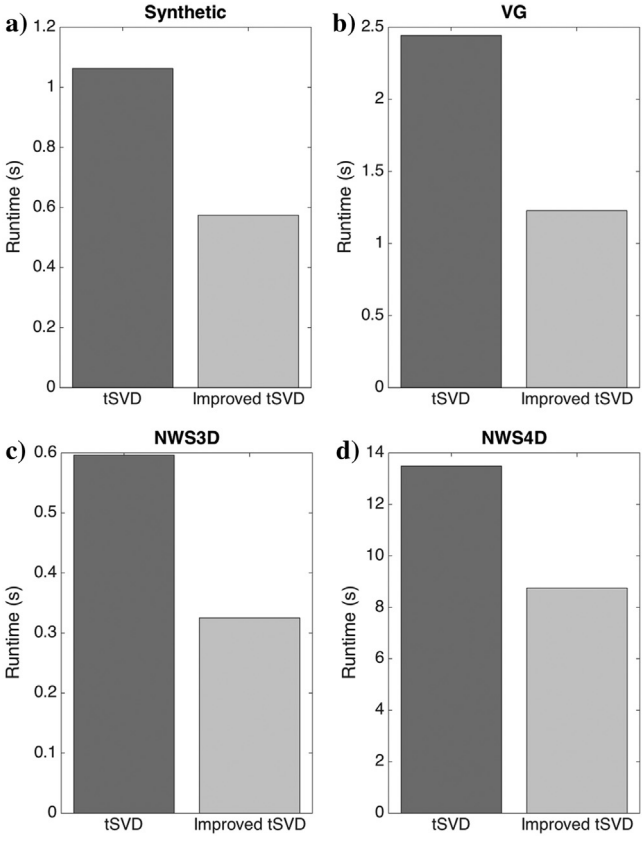


Figure 18. Runtime comparison for the inner loop of the original and improved tSVD given in Algorithm 2 applied to (a) the synthetic data, (b) VG, (c) NWS3D, and (d) NWS4D. The runtime decreased by 46%, 50%, 45%, and 35%, respectively.

tSVD runtime results

We compare the runtime of the inner loop of the original tSVD to our improved algorithm using conjugate symmetry. The runtime of our algorithm includes the time to create the mapping vector, which only needs to be performed once per set of dimensions. In Figure 18, we see that, for the synthetic data, we obtain an improvement in the runtime of 46%. We also test the algorithms using the VG, NWS3D, and NWS4D field data sets. For the VG data, the runtime improves by approximately 50% by using conjugate symmetry, close to the theoretical limit. We observe improvements of 45% and 35% for the NWS3D and NWS4D data sets, respectively. These results indicate that the algorithm is robust with respect to the dimension of the input data, increasing the efficiency for the order-3 and order-4 cases.

CONCLUSION

We find that TNN-ADMM is an accurate and efficient method for data completion. In this work, we give mathematical justification for the most cost-efficient and accurate way to complete seismic data for tensors of arbitrary order. We find that, for any tensor $\mathcal{X}$, the corresponding block diagonal matrix $\bar{X}$ shares the same singular values. Furthermore, the orientation of the tensor $\mathcal{X}$ affects the shape of the matrix $\bar{X}$. In particular, the most square orientation, for which the ratio of the first and second dimensions is closest to one, gives the block diagonal matrix $\bar{X}$ with the best data structure for the low-rank model. For the 3D and 4D examples, we observe that the most square orientation provides the closest approximation to a low-rank data structure and produces the best recovered result. We show that the runtime of the tSVD and tProduct algorithms can be improved, independent of tensor orientation, by using the conjugate symmetry produced by the multidimensional Fourier transform. This symmetry causes frontal slices to form entry-wise conjugate pairs. Taking advantage of this symmetry, costly operations on frontal slices, such as matrix SVD in the tSVD algorithm, can be replaced by cheaper conjugate calculations for the appropriate slices.

Being able to accurately fill in missing data has implications for more cost-efficient survey design. One could further investigate whether the most square orientation provides the best recovery for the additional denoising problem.

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 for providing the NWS 3D marine streamer data. We thank Mobil Oil (now ExxonMobil) for making publicly available the seismic data set from the VG Region. We also thank W. Zhou and D. Lumley for their help in preparing the data used in our experiments. We thank the four anonymous reviewers and the editors for their comments that greatly improved the manuscript.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are confidential and cannot be released.

APPENDIX A

MULTIDIMENSIONAL FOURIER TRANSFORM AND TENSOR PRODUCT

For a tensor $\mathcal{X}$, we denote the result of the multidimensional Fourier transform as $\hat{\mathcal{X}}$. For the multidimensional Fourier transform, we compute the Fourier transform for each dimension from 3 to p. To compute a Fourier transform over the third dimension, we consider *tubes*, or vectors oriented along the third dimension. For each tube, we compute its Fourier transform treating the tube as a vector.

The *tensor product* (tProduct) is used to define multiplication between two tensors (Martin et al., 2013). The tProduct of $\mathcal{A} \in \mathbb{R}^{n_1 \times m \times n_3 \times \cdots \times n_p}$ and $\mathcal{B} \in \mathbb{R}^{m \times n_2 \times n_3 \times \cdots \times n_p}$ results in a tensor $\mathcal{C} = \mathcal{A}\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$. The tProduct can be computed in three steps. First, take the multidimensional Fourier transform of $\mathcal{A}$ and $\mathcal{B}$, denoted $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$. Then, multiply the frontal slices of $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, storing the result of each matrix product as a frontal slice in $\hat{\mathcal{C}}$. Fianlly, we take the inverse multidimensional Fourier transform of $\hat{\mathcal{C}}$ to obtain $\mathcal{C}$. For the tSVD, we need to define unitary tensors. Unitary tensors have the property that their conjugate transpose is their inverse, i.e., $\mathcal{U}\mathcal{U}^* = \mathcal{I}$ and $\mathcal{U}^*\mathcal{U} = \mathcal{I}$. The *identity* tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \cdots \times n_p}$ has the identity matrix as its first frontal slice and zeros elsewhere. Transposing a tensor is computed recursively by transposing each subtensor and reversing the order of all but the first subtensor.

APPENDIX B

COHERENCE

Let $e_i \in \mathbb{R}^{n_1 \times 1 \times n_3 \times \cdots \times n_p}$ denote the **tensor column basis**, with an entry of 1 in the ith row of first frontal slice and zeros elsewhere.

Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$ have tSVD given by $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \cdots \times n_p}$, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$, and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3 \times \cdots \times n_p}$. Let $r = \max_{i=1,\ldots,N}(\text{rank}(\mathcal{X}_i))$. The **economic tSVD** is given by truncating $\mathcal{S}$, keeping only the first r nonzero rows and columns in each slice and similarly truncating $\mathcal{U}$ and $\mathcal{V}$ to keep only the first r columns in each slice.

For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$ with economic tSVD, $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3 \times \cdots \times n_p}$, $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3 \times \cdots \times n_p}$, and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3 \times \cdots \times n_p}$, the **coherence** of $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{X}$ is defined as (Zhang and Aeron (2017))

$$\begin{aligned} \mu(\mathcal{U}) &= \frac{n_1}{r} \max_{1 \le i \le n_1} \|\mathcal{U}^T * e_i\|_F^2 \in \left[1, \frac{n_1}{r}\right], \\ \mu(\mathcal{V}) &= \frac{n_2}{r} \max_{1 \le j \le n_2} \|\mathcal{V}^T * e_j\|_F^2 \in \left[1, \frac{n_2}{r}\right], \\ \mu(\mathcal{X}) &= \max(\mu(\mathcal{U}), \mu(\mathcal{V})). \end{aligned} \tag{B-1}$$

The more concentrated the singular vectors of the underlying data $\mathcal{X}$ are, the more likely it is that $\mathcal{X}$ will fall in the null space of the sampling operator. For example, consider the rank 1 matrix with a singular vector concentrated in a single standard basis element $u = e_1 \in \mathbb{R}^5$:

$$X = uu^T = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{B-2}$$

Any partial observation is likely to observe only zeros, making recovery unlikely. This matrix has coherence of $\mu(X) = 5$.

Now, consider another rank 1 matrix, with singular vector $u = (e_1 + e_2 + e_3 + e_4)/2 \in \mathbb{R}^5$ and singular value $\sigma_1 = 4$:

$$X = \sigma_1 uu^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{B-3}$$

In this example, the singular vector is spread across four standard basis vectors and the matrix has coherence of $\mu(X) = 1.25$. Here, recovery is possible with only a partial observation. The following theorem by Zhang and Aeron (2017) supports the idea that tensors of low coherence are more likely to be recovered.

Theorem: Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ represent the underlying data with economic tSVD $\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T$, $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$, $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$, and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$. Let p be the probability that an individual entry of $\mathcal{X}$ is observed. Then, there exists constants $c_0, c_1, c_2 > 0$ such that if

$$p \ge c_0 \frac{\mu(\mathcal{X}) r \log(n_3(n_1 + n_2))}{\min(n_1, n_2)}, \tag{B-4}$$

then $\mathcal{X}$ is the unique minimizer of equation 5 with probability $1 - c_1((n_1 + n_2)n_3)^{-c_2}$.

Because p is linearly dependent on $\mu(\mathcal{X})$, smaller coherence allows for data completion to be possible with more entries missing.

REFERENCES

Abma, R., and N. Kabir, 2006, 3D interpolation of irregular data with a POCS algorithm: Geophysics, **71**, no. 6, E91–E97, doi: 10.1190/1.2356088.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein, 2010, Distributed optimization and statistical learning via the alternating direction method of multipliers: Foundations and Trends in Machine Learning, **3**, 1–122, doi: 10.1561/2200000016.

Briggs, W. L., and V. E. Henson, 1995, The DFT: An owners' manual for the discrete Fourier transform: SIAM, 143–178.

Candès, E., and B. Recht, 2009, Exact matrix completion via convex optimization: Foundations of Computational Mathematics, **9**, 717–772, doi: 10.1007/s10208-009-9045-5.

Carozzi, F., and M. D. Sacchi, 2017, 5D seismic reconstruction via parallel matrix factorization with randomized QR decomposition: 87th Annual International Meeting, SEG, Expanded Abstracts, 4251–4256, doi: 10.1190/segam2017-17782228.1.

Chen, Y., W. Huang, D. Zhang, and W. Chen, 2016, An open-source Matlab code package for improved rank-reduction 3D seismic data denoising and reconstruction: Computers & Geosciences, **95**, 59–66, doi: 10.1016/j.cageo.2016.06.017.

Cheng, J., and M. D. Sacchi, 2015, A fast rank-reduction algorithm for 3D deblending via randomized QR decomposition: 85th Annual International Meeting, SEG, Expanded Abstracts, 3830–3835, doi: 10.1190/segam2015-5850767.1.

Chi, B., L. Dong, and Y. Liu, 2015, Correlation-based reflection full-waveform inversion: Geophysics, **80**, no. 4, R189–R202, doi: 10.1190/geo2014-0345.1.

De Lathauwer, L., B. De Moor, and J. Vandewalle, 2000, A multilinear singular value decomposition: SIAM Journal on Matrix Analysis and Applications, **21**, 1253–1278, doi: 10.1137/S0895479896305696.

Ely, G., S. Aeron, N. Hao, and M. E. Kilmer, 2013, 5D and 4D pre-stack seismic data completion using tensor nuclear norm (TNN): 83rd Annual International Meeting, SEG, Expanded Abstracts, 3639–3644, doi: 10.1190/segam2013-1143.1.

Ely, G., S. Aeron, N. Hao, and M. E. Kilmer, 2015, 5D seismic data completion and denoising using a novel class of tensor decompositions: Geophysics, **80**, no. 4, V83–V95, doi: 10.1190/geo2014-0467.1.

Gao, J., J. Cheng, and M. D. Sacchi, 2017, Five-dimensional seismic reconstruction using parallel square matrix factorization: IEEE Transactions on Geoscience and Remote Sensing, **55**, 2124–2135, doi: 10.1109/TGRS.2016.2636864.

Gao, J., A. Stanton, and M. D. Sacchi, 2015, Parallel matrix factorization algorithm and its application to 5D seismic reconstruction and denoising: Geophysics, **80**, no. 6, V173–V187, doi: 10.1190/geo2014-0594.1.

Gao, W., and M. D. Sacchi, 2018, PP-wave and PS-wave 5D reconstruction and denoising to assist registration: 88th Annual International Meeting, SEG, Expanded Abstracts, 2397–2401, doi: 10.1190/segam2018-2997862.1.

Ghaderpour, E., W. Liao, and M. P. Lamoureux, 2018, Antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation: Geophysics, **83**, no. 3, V157–V170, doi: 10.1190/geo2017-0284.1.

Ghaderpour, E., and S. Pagiatakis, 2017, Least-squares wavelet analysis of unequally spaced and non-stationary time series and its applications: Mathematical Geosciences, **49**, 819–844, doi: 10.1007/s11004-017-9691-0.

Hunt, L., J. Downton, S. Reynolds, S. Hadley, D. Trad, and M. Hadley, 2010, The effect of interpolation on imaging and AVO: A Viking case study: Geophysics, **75**, no. 6, WB265–WB274, doi: 10.1190/1.3475390.

Kilmer, M. E., and C. D. Martin, 2011, Factorization strategies for third-order tensors: Linear Algebra & Its Applications, **435**, 641–658, doi: 10.1016/j.laa.2010.09.020.

Kolda, T. G., and B. W. Bader, 2009, Tensor decompositions and applications: SIAM Review, **51**, 455–500, doi: 10.1137/07070111X.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, no. 3, V113–V122, doi: 10.1190/geo2011-0399.1.

Kumar, R., C. Da Silva, O. Akalin, A. Y. Aravkin, H. Mansour, B. Recht, and F. J. Herrmann, 2015, Efficient matrix completion for seismic data reconstruction: Geophysics, **80**, no. 5, V97–V113, doi: 10.1190/geo2014-0369.1.

Liu, C., H. Shan, and C. Chen, 2020a, Tensor p-shrinkage nuclear norm for low-rank tensor completion: Neurocomputing, **387**, 255–267, doi: 10.1016/j.neucom.2020.01.009.

Liu, X.-Y., S. Aeron, V. Aggarwal, and X. Wang, 2020b, Low-tubal-rank tensor completion using alternating minimization: IEEE Transactions on Information Theory, **66**, 1714–1737, doi: 10.1109/TIT.2019.2959980.

Long, Z., Y. Liu, L. Chen, and C. Zhu, 2019, Low rank tensor completion for multiway visual data: Signal Processing, **155**, 301–316, doi: 10.1016/j.sigpro.2018.09.039.

Lu, C., J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, 2020, Tensor robust principal component analysis with a new tensor nuclear norm: IEEE Transactions on Pattern Analysis and Machine Intelligence, **42**, 925–938, doi: 10.1109/TPAMI.2019.2891760.

Martin, C. D., R. Shafer, and B. Larue, 2013, An order-p tensor factorization with applications in imaging: SIAM Journal on Scientific Computing, **35**, A474–A490, doi: 10.1137/110841229.

Mohd Sagheer, S. V., and S. N. George, 2019, Denoising of low-dose CT images via low-rank tensor modeling and total variation regularization: Artificial Intelligence in Medicine, **94**, 1–17, doi: 10.1016/j.artmed.2018.12.006.

Mu, C., B. Huang, J. Wright, and D. Goldfarb, 2014, Square deal: Lower bounds and improved relaxations for tensor recovery: Proceedings of the Machine Learning Research, PMLR, 73–81.

Mu, Y., P. Wang, L. Lu, X. Zhang, and L. Qi, 2020, Weighted tensor nuclear norm minimization for tensor completion using tensor-SVD: Pattern Recognition Letters, **130**, 4–11, doi: 10.1016/j.patrec.2018.12.012.

Naghizadeh, M., and M. Sacchi, 2013, Multidimensional de-aliased Cadzow reconstruction of seismic records: Geophysics, **78**, no. 1, A1–A5, doi: 10.1190/geo2012-0200.1.

Ng, M. K.-P., Q. Yuan, L. Yan, and J. Sun, 2017, An adaptive weighted tensor completion method for the recovery of remote sensing images with missing data: IEEE Transactions on Geoscience and Remote Sensing, **55**, 3367–3381, doi: 10.1109/TGRS.2017.2670021.

Oropeza, V., and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, no. 3, V25–V32, doi: 10.1190/1.3552706.

Øzbek, A., A. K. Øzdemir, and M. Vassallo, 2009, Interpolation by matching pursuit: 79th Annual International Meeting, SEG, Expanded Abstracts, 3254–3258, doi: 10.1190/1.3255534.

Popa, J., S. Minkoff, and Y. Lou, 2019, Improving seismic data completion and efficiency using tensors: 89th Annual International Meeting, SEG, Expanded Abstracts, 4034–4038, doi: 10.1190/segam2019-3215431.1.

Popa, J., S. Minkoff, and Y. Lou, 2020, Improving seismic data completion via low-rank tensor optimization: 90th Annual International Meeting, SEG, Expanded Abstracts, 2774–2778, doi: 10.1190/segam2020-3423985.1.

Sacchi, M. D., and J. Cheng, 2017, 5D reconstruction via robust tensor completion: Presented at the GeoConvention, CSPG/CSEG/CWLS.

Sacchi, M. D., G. Garabito, H. Schots, and J. Caldeira, 2017, Preconditioning and denoising prestack onshore seismic data via 5D reconstruction: Application to 3D data from the Parnaba Basin: SBGf, 1384–1388.

Sacchi, M. D., and B. Liu, 2005, Minimum weighted norm wavefield reconstruction for AVA imaging: Geophysical Prospecting, **53**, 787–801, doi: 10.1111/j.1365-2478.2005.00503.x.

Semerci, O., N. Hao, M. E. Kilmer, and E. L. Miller, 2014, Tensor-based formulation and nuclear norm regularization for multienergy computed tomography: IEEE Transactions on Image Processing, **23**, 1678–1693, doi: 10.1109/TIP.2014.2305840.

Stanton, A., N. Kreimer, D. Bonar, M. Naghizadeh, and M. Sacchi, 2012, A comparison of 5D reconstruction methods: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi: 10.1190/segam2012-0269.1.

Strang, G., 1980, Linear algebra and its applications: Academic Press.

Su, Y., X. Wu, and W. Liu, 2019, Low-rank tensor completion by sum of tensor nuclear norm minimization: IEEE Access, **7**, 134943–134953, doi: 10.1109/ACCESS.2019.2940664.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation: 80th Annual International Meeting, SEG, Expanded Abstracts, 3829–3833, doi: 10.1190/1.3513645.

Tucker, L. R., 1966, Some mathematical notes on three-mode factor analysis: Psychometrika, **31**, 279–311, doi: 10.1007/BF02289464.

Xu, S., Y. Zhang, D. Pham, and G. Lambar, 2005, Antileakage Fourier transform for seismic data regularization: Geophysics, **70**, no. 4, V87–V95, doi: 10.1190/1.1993713.

Zhang, Z., and S. Aeron, 2017, Exact tensor completion using T-SVD: IEEE Transactions on Signal Processing, **65**, 1511–1526, doi: 10.1109/TSP.2016.2639466.

Zhang, Z., G. Ely, S. Aeron, N. Hao, and M. Kilmer, 2014, Novel methods for multilinear data completion and de-noising based on tensor-SVD: IEEE Conference on Computer Vision and Pattern Recognition, 3842–3849.

Biographies and photographs of the authors are not available.