Novel methods for multilinear data completion and de-noising based on tensor-SVD

Zemin Zhang, Gregory Ely, Shuchin Aeron
Department of ECE, Tufts University
Medford, MA 02155
zemin.zhang@tufts.com
gregoryely@gmail.com
shuchin@ece.tufts.edu

Ning Hao and Misha Kilmer
Department of Mathematics
Medford, MA 02155
ning.hao@tufts.edu
misha.kilmer@tufts.edu

Abstract

In this paper we propose novel methods for completion (from limited samples) and de-noising of multilinear (tensor) data and as an application consider 3-D and 4-D (color) video data completion and de-noising. We exploit the recently proposed tensor-Singular Value Decomposition (t-SVD)[11]. Based on t-SVD, the notion of multilinear rank and a related tensor nuclear norm was proposed in [11] to characterize informational and structural complexity of multilinear data. We first show that videos with linear camera motion can be represented more efficiently using t-SVD compared to the approaches based on vectorizing or flattening of the tensors. Since efficiency in representation implies efficiency in recovery, we outline a tensor nuclear norm penalized algorithm for video completion from missing entries. Application of the proposed algorithm for video recovery from missing entries is shown to yield a superior performance over existing methods. We also consider the problem of tensor robust Principal Component Analysis (PCA) for de-noising 3-D video data from sparse random corruptions. We show superior performance of our method compared to the matrix robust PCA adapted to this setting as proposed in [4].

1. Introduction

This paper focuses on several novel methods for robust recovery of multilinear signals or tensors (essentially viewed as 2-D, 3-D,..., N-D data) under limited sampling and measurements. Signal recovery from partial measurements, sometimes also referred to as the problem of data completion for specific choice of measurement operator being a simple downsampling operation, has been an important area of research, not only for statistical signal processing problems related to inversion, [6, 20, 15], but also in machine learning for online prediction of ratings, [9]. All of these applications exploit low structural and informational complexity of the data, expressed either as low rank for the 2-D matrices [6, 20], which can be extended to higher order data via *flattening or vectorizing* of the tensor data such as tensor N-rank [7], or other more general tensor-rank measures based on particular tensor decompositions such as higher oder SVD (HOSVD) or Tucker-3 and Canonical Decomposition (CANDECOMP). See [13] for a survey of these decompositions.

The key idea behind these methods is that under the assumption of low-rank of the underlying data thereby constraining the complexity of the hypothesis space, it should be feasible to recover data (or equivalently predict the missing entries) from number of measurements in proportion to the rank. Such analysis and the corresponding identifiability results are obtained by considering an appropriate complexity penalized recovery algorithm under observation constraints, where the measure of complexity, related to the notion of rank, comes from a particular *factorization* of the data. Such algorithms are inherently combinatorial and to alleviate this difficulty one looks for the tightest convex relaxations of the complexity measure, following which the well developed machinery of convex optimization as well as convex analysis can be employed to study the related problem. For example, rank of the 2-D matrix being relaxed to the Schatten 1-norm, [19] and tensor N-rank for order $N > 2$ tensors being relaxed to overlapped Schatten p-norms, [7].

Note that all of the current approaches to handle multilinear data extend the nearly optimal 2-D SVD[1] based vector space approach to the higher order ($N > 2$) case. This results in loss of optimality in the representation. In contrast, our approach is based upon recent results on decom-

[1] Optimality of 2-D SVD is based on the optimality of truncated SVD as the best k-dimensional ℓ_2 approximation.

position/factorization of tensors in [2, 12, 11] in which the authors refer to as tensor-SVD or t-SVD for short. Essentially the t-SVD is based on an operator theoretic interpretation of third-order tensors as linear operators on the space of oriented matrices [2]. This notion can be extended recursively to higher order tensors [16]. In this paper we will exploit this decomposition, the associated notion of tensor multi-rank and its convex relaxation to the corresponding Tensor Nuclear Norm (TNN) (see[18]) for completion and recovery of multilinear data.

This paper is organized as follows. Section 2 presents the notations and provide an overview and key results on t-SVD from [2, 12, 11] and illustrates the key differences and advantages over other tensor decomposition methods. We will then provide an over-view of the related structural complexity measures. In Section 3 we study the compression performance of the t-SVD based representation on several video data sets. Following that, in Section 4 we propose a tensor nuclear norm (TNN) penalized algorithm for 3-D and 4-D (color) video completion from randomly sampled data cube. In Section 5 we consider a tensor robust Principal Component Analysis (PCA) problem for videos with sparse data corruption and propose an algorithm to separate low multi-rank video from sparse corruptions. Finally we conclude in Section 6.

2. Brief overview of t-SVD

In this section, we describe the tensor decomposition as proposed in [2, 12, 11] and the notations used throughout the paper.

2.1. Notation and Indexing

A ***Slice*** of an N-dimensional tensor is a 2-D section defined by fixing all but two indices. A ***Fiber*** of an N-dimensional tensor is a 1-D section defined by fixing all indices but one [13]. For a third order tensor $\mathcal{A}$, we will use the Matlab notation $\mathcal{A}(k,:,:)$, $\mathcal{A}(:,k,:)$ and $\mathcal{A}(:,:,k)$ to denote the k_{th} horizontal, lateral and frontal slices, and $\mathcal{A}(:,i,j)$, $\mathcal{A}(i,:,j)$ and $\mathcal{A}(i,j,:)$ to denote the $(i,j)_{th}$ mode-1, mode-2 and mode-3 fiber. In particular, we use $\mathcal{A}^{(k)}$ to represent $\mathcal{A}(:,:,k)$.

One can view a 3-D tensor of size $n_1 \times n_2 \times n_3$ as an $n_1 \times n_2$ matrix of tubes. By introducing a commutative operation $*$ between the tubes $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{1\times1\times n_3}$ via $\mathbf{a} * \mathbf{b} = \mathbf{a} \circ \mathbf{b}$, where $\circ$ denotes the *circular convolution* between the two vectors, one defines the t-product between two tensors as follows.

Definition 2.1.1. t-product. *The t-product $\mathcal{C} = \mathcal{A} * \mathcal{B}$ of $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$ is a tensor of size $n_1 \times n_4 \times n_3$ where the $(i,j)_{th}$ tube denoted by $\mathcal{C}(i,j,:)$ for $i = 1,2,...,n_1$ and $j = 1,2,...,n_4$ of the tensor $\mathcal{C}$ is given by $\sum_{k=1}^{n_2} \mathcal{A}(i,k,:) * \mathcal{B}(k,j,:)$.*

The t-product is analogous to the matrix multiplication except that circular convolution replaces the multiplication operation between the elements, which are now tubes. Next we define related notions of tensor transpose and identity tensor.

Definition 2.1.2. Tensor Transpose. *Let $\mathcal{A}$ be a tensor of size $n_1 \times n_2 \times n_3$, then $\mathcal{A}^{\mathrm{T}}$ is the $n_2 \times n_1 \times n_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3.*

Definition 2.1.3. Identity Tensor. *The identity tensor $\mathcal{J} \in \mathbb{R}^{n_1\times n_1\times n_3}$ is a tensor whose first frontal slice is the $n_1 \times n_1$ identity matrix and all other frontal slices are zero.*

Definition 2.1.4. f-diagonal Tensor. *A tensor is called f-diagonal if each frontal slice of the tensor is a diagonal matrix.*

The t-product of $\mathcal{A}$ and $\mathcal{B}$ can be computed efficiently by performing the fast Fourier transformation (FFT) along the tube fibers of $\mathcal{A}$ and $\mathcal{B}$ to get $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, multiplying the each pair of the frontal slices of $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ to obtain $\hat{\mathcal{C}}$, and then taking the inverse FFT along the third mode to get the result. For details about the computation, see [12, 11].

Definition 2.1.5. Orthogonal Tensor. *A tensor $\mathcal{Q} \in \mathbb{R}^{n_1\times n_1\times n_3}$ is orthogonal if*

$$\mathcal{Q}^{\mathrm{T}} * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^{\mathrm{T}} = \mathcal{J} \tag{1}$$

where $*$ is the t-product.

2.2. Tensor Singular Value Decomposition (t-SVD)

The new t-product allows us to define a tensor Singular Value Decomposition (t-SVD).

Theorem 2.2.1. *For $\mathcal{M} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the t-SVD of $\mathcal{M}$ is given by*

$$\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{T}} \tag{2}$$

where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors of size $n_1 \times n_1 \times n_3$ and $n_2 \times n_2 \times n_3$ respectively. $\mathcal{S}$ is a rectangular f-diagonal tensor of size $n_1 \times n_2 \times n_3$, and $*$ denotes the t-product.

One can obtain this decomposition by computing matrix SVDs in the Fourier domain, see Algorithm 1. The notation in the algorithm can be found in [16]. Figure 1 illustrates the decomposition for the 3-D case.

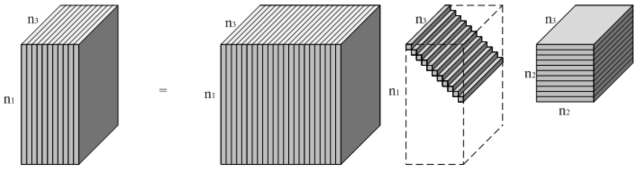


Figure 1. The t-SVD of an $n_1 \times n_2 \times n_3$ tensor.

Algorithm 1 t-SVD

```
Input: $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \ldots \times n_N}$
$\rho = n_3 n_4 \ldots n_N$
for $i = 3$ to N do
    $\mathcal{D} \leftarrow$ fft($\mathcal{M}$, [ ], i);
end for
for $i = 1$ to $\rho$ do
    $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = SVD(\mathcal{D}(:,:,i))$
    $\hat{\mathcal{U}}(:,:,i) = \mathbf{U}$; $\hat{\mathcal{S}}(:,:,i) = \mathbf{S}$; $\hat{\mathcal{V}}(:,:,i) = \mathbf{V}$;
end for
for $i = 3$ to $\rho$ do
    $\mathcal{U} \leftarrow$ ifft($\hat{\mathcal{U}}$, [ ], i); $\mathcal{S} \leftarrow$ ifft($\hat{\mathcal{S}}$, [ ], i); $\mathcal{V} \leftarrow$ ifft($\hat{\mathcal{V}}$, [ ], )i;
end for
```

2.3. t-SVD: Fundamental theorems and key results

The two widely used tensor decompositions, Tucker and PARAFAC[13] are usually seen as a higher order SVD for tensors. Both of these decompositions have several disadvantages. In particular, one cannot easily determine the rank-one components of the PARAFAC decomposition and given a fixed rank, calculation of an approximation can be numerically unstable. The tensor-train form of the Tucker decomposition is studied in [10] as an alternative form. Tucker decomposition can be seen as a generalization of PARAFAC decomposition, and the truncated decomposition doesn't yield the best fit of the original tensor. In contrast, the t-SVD can be easily computed by solving several SVDs in the Fourier domain. More importantly, it gives an optimal approximation of a tensor measured by the Frobenious norm of the difference, as stated in the following theorem[12, 11, 8].

Theorem 2.3.1. *Let the t-SVD of $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be given by $\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ and for $k < \min(n_1, n_2)$ define $\mathcal{M}_k = \sum_{i=1}^{k} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:,i,:)^T$, Then*

$$\mathcal{M}_k = \arg \min_{\tilde{\mathcal{M}} \in \mathbb{M}} ||\mathcal{M} - \tilde{\mathcal{M}}||_F$$

where $\mathbb{M} = \{\mathcal{C} = \mathcal{X} * \mathcal{Y} | \mathcal{X} \in \mathbb{R}^{n_1 \times k \times n_3}, \mathcal{Y} \in \mathbb{R}^{k \times n_2 \times n_3}\}$.

2.4. Measures of tensor complexity using t-SVD

We now define two measures of tensor complexity based on the proposed t-SVD: the tensor multi-rank, proposed in [11], and the novel tensor tubal rank.

Definition 2.4.1. ***Tensor multi-rank**. The multi-rank of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a vector $p \in \mathbb{R}^{n_3 \times 1}$ with the i_{th} element equal to the rank of the i_{th} frontal slice of $\hat{\mathcal{A}}$ obtained by taking the Fourier transform along the third dimension of the tensor.*

One can obtain a scalar measure of complexity as the ℓ_1 norm of the tensor multi-rank. We now define another measure motivated by the matrix SVD.

Definition 2.4.2. ***Tensor tubal-rank**. The tensor tubal rank of a 3-D tensor is defined to be the number of non-zero tubes of $\mathcal{S}$ in the t-SVD factorization.*

As in the matrix case, practical applications of these complexity measures require adequate convex relaxations. To this end we have the following result for the Tensor multi-rank.

Theorem 2.4.1. *The tensor-nuclear-norm (TNN) denoted by $||\mathcal{A}||_{TNN}$ and defined as the sum of the singular values of all the frontal slices of $\hat{\mathcal{A}}$ is a norm and is the tightest convex relaxation to ℓ_1 norm of the tensor multi-rank.*

Proof. The proof that TNN is a valid norm can be found in [18]. The ℓ_1 norm of the tensor multi-rank is equal to rank(blkdiag($\hat{\mathcal{A}}$)), for which the tightest convex relaxation is the the nuclear norm of blkdiag($\hat{\mathcal{A}}$) which is TNN of $\mathcal{A}$ by definition. Here blkdiag($\hat{\mathcal{A}}$) is a block diagonal matrix defined as follows:

$$\text{blkdiag}(\hat{\mathcal{A}}) = \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(n_3)} \end{bmatrix} \quad (3)$$

where $\hat{\mathcal{A}}^{(i)}$ is the i_{th} frontal slice of $\hat{\mathcal{A}}$, $i = 1, 2, ..., n_3$. □

Unlike the TNN is a relaxation for the tensor-nuclear-norm, there is no clear convex relaxation for the tensor tubal-rank. In the next section we will undertake a compressibility study for tensor data using two types of truncation strategies based on t-SVD and compare them with matrix SVD based approach on several video data sets.

3. Multilinear data compression using t-SVD

We outline two methods for compression based on t-SVD and compare them with the traditional truncated SVD based approach in this section. Note that we don't compare with truncated HOSVD or other tensor decompositions as there is no notion of optimality for these decompositions in contrast to truncated t-SVD and truncated SVD.

The use of SVD in matrix compression has been widely studied in [17]. For a matrix $A \in \mathbb{R}^{m \times n}$ with its SVD $A = USV^T$, the rank r approximation of A is the matrix $A_r = U_r S_r V_r^{\mathrm{T}}$, where S_r is a $r \times r$ diagonal matrix with $S_r(i,i) = S(i,i), i = 1, 2, ..., r$. U_r consists of the first r columns of U and V_r^{T} consists of the first r rows of V_r^{T}. The compression is measured by the ratio of the total number of entries in A, which is mn, to the total number of entries in U_r, S_r and V_r^{T}, which is equal to $(m + n + 1)r$. Extending this approach to a third-order tensor $\mathcal{M}$ of size $n_1 \times n_2 \times n_3$, we vectorize each frontal slice and save it as a column, so we get an $n_1 n_2 \times n_3$ matrix. Then the compression ratio of rank k_1 SVD approximation is

$$\text{ratio}_{\text{SVD}} = \frac{n_1 n_2 n_3}{n_1 n_2 k_1 + k_1 + n_3 k_1} = \frac{n_1 n_2 n_3}{k_1(n_1 n_2 + n_3 + 1)} \tag{4}$$

where $1 \leq k_1 \leq \min(n_1 n_2, n_3)$. Generally even with small k_1, the approximation M_{k_1} gets most of the information of $\mathcal{M}$.

Method 1 : Based on t-SVD our first method for compression, which we call t-SVD compression, basically follows the same idea of truncated SVD but in the *Fourier domain*. For an $n_1 \times n_2 \times n_3$ tensor $\mathcal{M}$, we use Algorithm 1 to get $\hat{\mathcal{M}}$, $\hat{\mathcal{U}}$, $\hat{\mathcal{S}}$ and $\hat{\mathcal{V}}^{\mathrm{T}}$. It is known that $\hat{\mathcal{S}}$ is a f-diagonal tensor with each frontal slice is a diagonal matrix. So the total number of f-diagonal entries of $\hat{\mathcal{S}}$ is $n_0 n_3$ where $n_0 = \min(n_1, n_2)$. We choose an integer k_2, $1 \leq k_2 \leq n_0 n_3$ and keep the k_2 largest f-diagonal entries of $\hat{\mathcal{S}}$ then set the rest to be 0. If $\hat{\mathcal{S}}(i,i,j)$ is set to be 0, then let the corresponding columns $\hat{\mathcal{U}}(:,i,j)$ and $\hat{\mathcal{V}}^{\mathrm{T}}(:,i,j)$ also be 0. We then call the resulting tensors $\hat{\mathcal{U}}_{k_2}$, $\hat{\mathcal{S}}_{k_2}$ and $\hat{\mathcal{V}}^{\mathrm{T}}_{k_2}$. So the approximation is $\mathcal{M}_{k_2} = \mathcal{U}_{k_2} * \mathcal{S}_{k_2} * \mathcal{V}^{\mathrm{T}}_{k_2}$ where $\mathcal{U}_{k_2}$, $\mathcal{S}_{k_2}$ and $\mathcal{V}^{\mathrm{T}}{}_{k_2}$ are the inverse Fourier transforms of $\hat{\mathcal{U}}_{k_2}$, $\hat{\mathcal{S}}_{k_2}$ and $\hat{\mathcal{V}}^{\mathrm{T}}_{k_2}$ along the third dimension. The compression ratio rate for this method is

$$\text{ratio}_{\text{t-SVD}} = \frac{n_1 n_2 n_3}{k_2(n_1 + n_2 + 1)}$$

where $1 \leq k_2 \leq n_0 n_3$.

Method 2: Our second method for compressing is called t-SVD-tubal compression and is also similar to truncated SVD but in the *t-product* domain. As in **Theorem 2.3.1**, we take the first k_3 tubes $(1 \leq k_3 \leq n_0)$ in $\mathcal{S}$ and get the approximation $\mathcal{M}_{k_3} = \sum_{i=1}^{k_3} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:,i,:)^{\mathrm{T}}$.

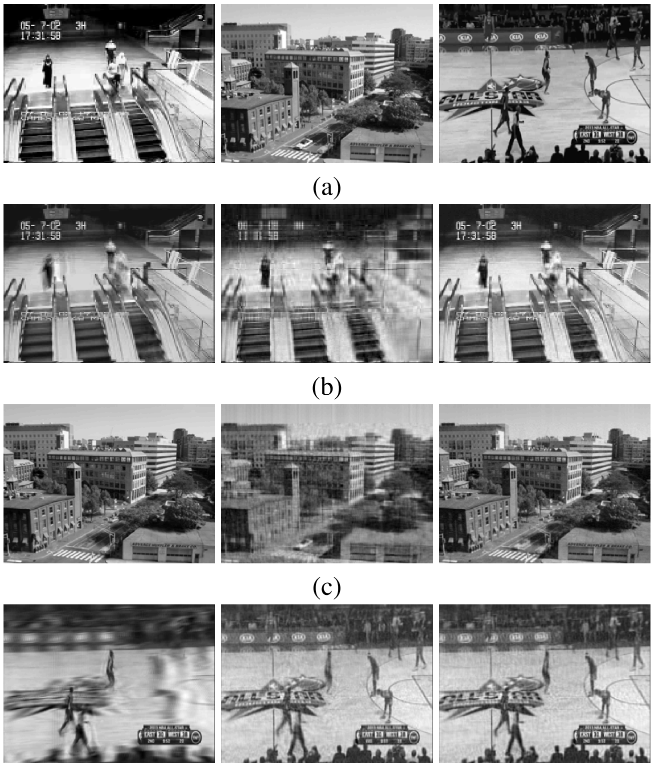
Compression ratio rate for the second method is

$$\text{ratio}_{\text{t-SVD-tubal}} = \frac{n_1 n_2}{k_3(n_1 + n_2 + 1)}$$

where $1 \leq k_3 \leq n_0$.

Video data representation and compression using t-SVD: We now illustrate the performance of SVD based compression, t-SVD compression and t-SVD tubal compression on 3 video datasets shown in Figure 2-(a).

1. The first video, referred to as the Escalator video, (source: http://www.ugcs.caltech.edu/ srbecker/rpca.shtml#2) of size $130 \times 160 \times 50$ (length $\times$ width $\times$ frames) from a stationary camera.
2. The second video, referred to as the MERL video, is a ***time lapse video*** of size $192 \times 256 \times 38$ also from a stationary camera (**data courtesy**: Dr. Amit Agrawal, Mitsubishi Electric Research Labs (MERL), Cambridge, MA).



(a)

(b)

(c)

(d)

Figure 2. (**a**) Three testing videos: escalator video, MERL video and basketball video. (b) (c) (d) are compression results under compression ratio 5. For (b) (c) (d) from left to right: SVD compression, t-SVD-tubal compression and t-SVD compression

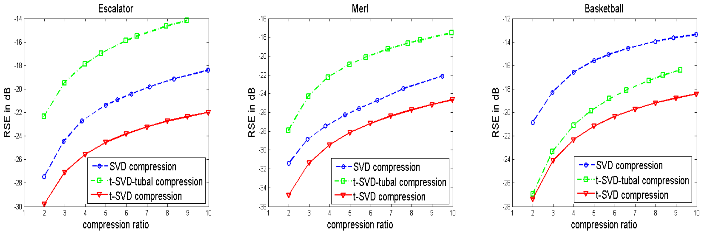


Figure 3. Compression ratio and RSE comparison for 3 videos.

3. The third video, referred to as the Basketball video is a $144 \times 256 \times 80$ video (source: YouTube) with a ***non-stationary panning camera*** moving from left to right horizontally following the running players.

Figure 2 (b) to (d) show the compression results for the 3 videos when truncated according to vectorized SVD and t-SVD compression (method 1) and t-SVD tensor tubal compression (method 2). In Figure 3 we show the relative square error (RSE) comparison for different compression ratio where RSE is defined in dB as RSE $= 20\log_{10}(\|\mathcal{X}_{\text{rec}} - \mathcal{X}\|_{\mathrm{F}} / \|\mathcal{X}\|_{\mathrm{F}})$. In all of the 3 results, the performance of t-SVD compression (method 1) is the best. This implies that tensor multi-rank fits very well for video datasets from both stationary and non-stationary cameras.

SVD compression method (based on vectorization) has a better performance over the t-SVD-tubal compression on the Escalator and MERL video. However, t-SVD tubal compression (method 2) works much better than SVD compression on the Basketball video. This is because in the videos where the camera is panning or in motion, one frontal slice of the tensor to the next frontal slice can be effectively represented as a shift and scaling operation which in turn is captured by a convolution type operation and t-SVD is based on such an operation along the third dimension.

4. Tensor completion from limited samples

We will show the case when the tensor data is simply decimated randomly or down sampled in this section. Specifically we consider the problem of data completion from missing entries for multilinear signals. Suppose there is an unknown tensor $\mathcal{M}$ of size $n_1 \times n_2 \times n_3$ which is ***assumed to have a low tubal-rank*** and we are given a subset of entries $\{\mathcal{M}_{ijk} : (i,j,k) \in \mathbf{\Omega}\}$ where $\mathbf{\Omega}$ is an indicator tensor of size $n_1 \times n_2 \times n_3$. Our objective is to recover the entire $\mathcal{M}$. This section develops an algorithm for addressing this problem via solving the following complexity penalized algorithm:

$$\begin{aligned} \min \quad & \|\mathcal{X}\|_{TNN} \\ \text{subject to} \quad & P_{\mathbf{\Omega}}(\mathcal{X}) = P_{\mathbf{\Omega}}(\mathcal{M}) \end{aligned} \tag{5}$$

where $P_{\mathbf{\Omega}}$ is the orthogonal projector onto the span of tensors vanishing outside of $\mathbf{\Omega}$. So the $(i,j,k)_{th}$ component of $P_{\mathbf{\Omega}}(\mathcal{X})$ is equal to $\mathcal{M}_{ijk}$ if $(i,j,k) \in \mathbf{\Omega}$ and zero otherwise. Let $\mathcal{Y}$ be the available (sampled) data: $\mathcal{Y} = P_{\mathbf{\Omega}}\mathcal{M}$. Define $\mathcal{G} = \mathcal{F}_3 P_{\mathbf{\Omega}} \mathcal{F}_3^{-1}$ where $\mathcal{F}_3$ and $\mathcal{F}_3^{-1}$ are the operators representing the Fourier and inverse Fourier transform along the third dimension of tensors. Then we have $\hat{\mathcal{Y}} = \mathcal{G}(\hat{\mathcal{M}})$ where $\hat{\mathcal{Y}}$ and $\hat{\mathcal{M}}$ are the Fourier transforms of $\mathcal{Y}$ and $\mathcal{M}$ along the third mode. So (5) is equivalent with the following:

$$\begin{aligned} \min \quad & ||\text{blkdiag}(\hat{\mathcal{X}})||_* \\ \text{subject to} \quad & \hat{\mathcal{Y}} = \mathcal{G}(\hat{\mathcal{X}}) \end{aligned} \tag{6}$$

where $\hat{\mathcal{X}}$ is the Fourier transform of $\mathcal{X}$ along the third dimension and blkdiag$(\hat{\mathcal{X}})$ is defined in (3). Noting that $\|\mathcal{X}\|_{TNN} = ||\text{blkdiag}(\hat{\mathcal{X}})||_*$. To solve the optimization problem, one can re-write (6) equivalently as follows:

$$\begin{aligned} \min \quad & ||\text{blkdiag}(\hat{\mathcal{Z}})||_* + \mathbf{1}_{\hat{\mathcal{Y}} = \mathcal{G}(\hat{\mathcal{X}})} \\ \text{subject to} \quad & \hat{\mathcal{X}} - \hat{\mathcal{Z}} = 0 \end{aligned} \tag{7}$$

where $\mathbf{1}$ denotes the indicator function. Then using the general framework of Alternating Direction Method of Multipliers(ADMM)[1] we have the following recursion,

$$\begin{aligned} &\mathcal{X}^{k+1} \\ &= \arg\min_{\mathcal{X}} \left\{ \mathbf{1}_{\mathcal{Y} = P_{\Omega}(\mathcal{X})} + \mathcal{X}(:)^{\mathrm{T}} \mathcal{Q}^k(:) + \frac{1}{2}||\mathcal{X} - \mathcal{Z}^k||_F^2 \right\} \\ &= \arg \min_{\mathcal{X}:\mathcal{Y} = P_{\Omega}(\mathcal{X})} \left\{ ||\mathcal{X} - (\mathcal{Z}^k - \mathcal{Q}^k)||_F^2 \right\} \end{aligned} \tag{8}$$

$$\begin{aligned} &\hat{\mathcal{Z}}^{k+1} \\ &= \arg\min_{\hat{\mathcal{Z}}} \left\{ \frac{1}{\rho}||\text{blkdiag}(\hat{\mathcal{Z}})||_* + \frac{1}{2}||\hat{\mathcal{Z}} - (\hat{\mathcal{X}}^{k+1} + \hat{\mathcal{Q}}^k)||_F^2 \right\} \end{aligned} \tag{9}$$

$$\mathcal{Q}^{k+1} = \mathcal{Q}^k + \left(\mathcal{X}^{k+1} - \mathcal{Z}^{k+1} \right) \tag{10}$$

where Equation (8) is least-squares projection onto the constraint and the solution to Equation (9) is given by the singular value thresholding[19, 3]. The $\mathcal{X}(:)$ and $\mathcal{Q}^k(:)$ means vectorizing the tensors which is Matlab notation.

4.1. Equivalence of the algorithm to iterative singular-tubal shrinkage via convolution

We will now show that the proposed algorithm for tensor completion has a very nice interpretation as an iterative singular tubal shrinkage using a convolution operation between the singular tubes and a tube of threshold vectors.

According to the particular format that (9) has, we can break it up into n_3 independent minimization problems. Let $\hat{\mathcal{Z}}^{k+1,(i)}$ denotes the i_{th} frontal slice of $\hat{\mathcal{Z}}^{k+1}$. Similarly define $\hat{\mathcal{X}}^{k+1,(i)}$ and $\hat{\mathcal{Q}}^{k,(i)}$. Then (9) can be separated as:

$$\begin{aligned} &\hat{\mathcal{Z}}^{k+1,(i)} \\ &= \arg\min_{W} \left\{ \frac{1}{\rho}||W||_* + \frac{1}{2}||W - (\hat{\mathcal{X}}^{k+1,(i)} + \hat{\mathcal{Q}}^{k,(i)})||_F^2 \right\} \end{aligned} \tag{11}$$

for $i = 1, 2, ..., n_3$. This means each i_{th} frontal slice of $\hat{\mathcal{Z}}^{k+1}$ can be calculated through (11).

In order to solve (11), we give out the following lemma.

Lemma 4.1.1. *Consider the singular value decomposition (SVD) of a matrix $X \in \mathbb{C}^{n_1 \times n_2}$ of rank r.*

$$X = U\Sigma V^*, \quad \Sigma = \textit{diag}(\{\sigma_i\}_{1 \le i \le r}), \tag{12}$$

where U and V are respectively $n_1 \times r$ and $n_2 \times r$ unitary matrices with orthonormal columns, and the singular values σ_i are real and positive. Then for all $\tau \ge 0$, define the soft-thresholding operator D_τ as follows [3] :

$$D_\tau(X) := UD_\tau(\Sigma)V^*, \quad D_\tau(\Sigma) = \textit{diag}\{(\sigma_i - \tau)_+\}, \tag{13}$$

where t_+ is the positive part of t, namely, $t_+ = max(0, t)$. Then, for each $\tau \geq 0$ and $Y \in C^{n_1 \times n_2}$, the singular value shrinkage operator (13) obeys

$$D_\tau(Y) = \arg\min_{X \in \mathbb{C}} \left\{ \frac{1}{2} \|X - Y\|_F^2 + \tau \|X\|_* \right\} \qquad (14)$$

The proof can be found in [3] for the case when the matrix is real valued. However, it can be easily extended to matrices with complex entries using the result on gradients of unitarily invariant norms in [14].

Now note that, if $USV^{\mathrm{T}} = (\hat{\mathcal{X}}^{k+1,(i)} + \hat{\mathcal{Q}}^{k,(i)})$ is the SVD of $(\hat{\mathcal{X}}^{k+1,(i)} + \hat{\mathcal{Q}}^{k,(i)})$, then the solution to (11) is $UD_\tau(S)V^{\mathrm{T}}$, where $D_\tau(S) = \text{diag}(S_{i,i} - \tau)_+$ for some positive constant τ and " $+$ " means keeping the positive part. This is equivalent to multiplying $(1 - \frac{\tau}{S_{i,i}})_+$ to the i_{th} singular value of S. So each frontal slice of $\hat{\mathcal{Z}}^{k+1}$ can be calculated using this shrinkage on each frontal slice of $(\hat{\mathcal{X}}^{k+1} + \hat{\mathcal{Q}}^k)$. Let $\mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{T}} = (\mathcal{X}^{k+1} + \mathcal{Q}^k)$ be the t-SVD of $(\mathcal{X}^{k+1} + \mathcal{Q}^k)$ and $\hat{\mathcal{S}}$ be the Fourier transform of $\mathcal{S}$ along the third dimension. Then each element of the singular tubes of $\hat{\mathcal{Z}}^{k+1}$ is the result of multiplying every entry $\hat{\mathcal{S}}(i,i,j)$ with $(1 - \frac{\tau}{\hat{\mathcal{S}}(i,i,j)})_+$ for some $\tau > 0$. Since this process is carried out in the Fourier domain, in the original domain it is equivalent to convolving each tube $\mathcal{S}(i,i,:)$ of $\mathcal{S}$ with a real valued tubal vector $\vec{\tau_i}$ which is the inverse Fourier transform of the vector $[(1 - \frac{\tau_i(1)}{\hat{\mathcal{S}}(i,i,1)})_+, (1 - \frac{\tau_i(2)}{\hat{\mathcal{S}}(i,i,2)})_+, ..., (1 - \frac{\tau_i(n_3)}{\hat{\mathcal{S}}(i,i,n_3)})_+]$. This operation can be captured by $\mathcal{S} * \mathcal{T}$, where $\mathcal{T}$ is an f-diagonal tensor with i_{th} diagonal tube to be $\vec{\tau_i}$. Then $\mathcal{Z}^{k+1} = \mathcal{U} * (\mathcal{S} * \mathcal{T}) * \mathcal{V}^{\mathrm{T}}$. In summary, the shrinkage operation in the Fourier domain on the singular values of each of the frontal faces is equivalent to performing a *tubal shrinkage* via convolution in the original domain.

Application to Video data completion - For experiments we test 3 algorithms for video data completion from randomly missing entries: TNN minimization of Section 4, Low Rank Tensor Completion (LRTC) algorithm in [15], which uses the notion of tensor-n-rank [7], and the nuclear norm minimization on the vectorized video data using the algorithm in [3]. As an application of the t-SVD to higher order tensor we also show performance on a 4-D color Basketball video data of size $144 \times 256 \times 3 \times 80$.

Figures 5 and 6 show the results of recovery using the 3 algorithms. Figure 7 shows the RSE (dB) plots for sampling rates ranging from 10% to 90% where the sampling rate is defined to be the percentage of pixels which are known. Results from the figures show that the TNN minimization algorithm gives excellent reconstruction over the LRTC and Nuclear norm minimization. These results are in line with the compressibility results in Section 3.

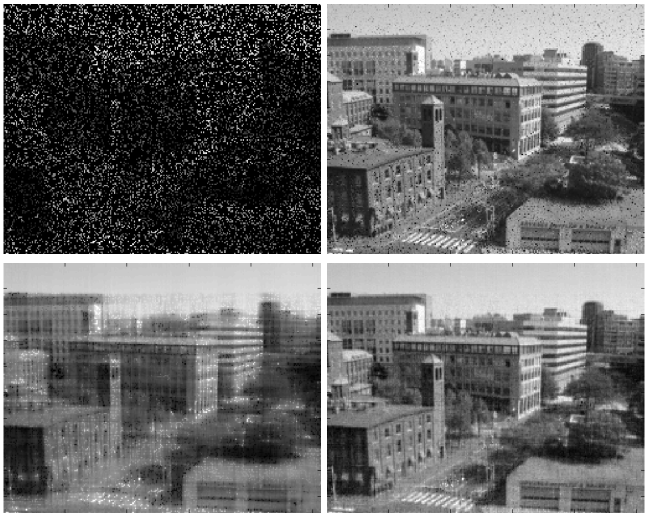


Figure 4. Tensor completion results for MERL video. **Upper left**: Sampled video(20%). **Upper right**: Nuclear norm minimization (vectorization and SVD based) result. **Lower left**: LRTC result. **Lower right**: TNN minimization result.

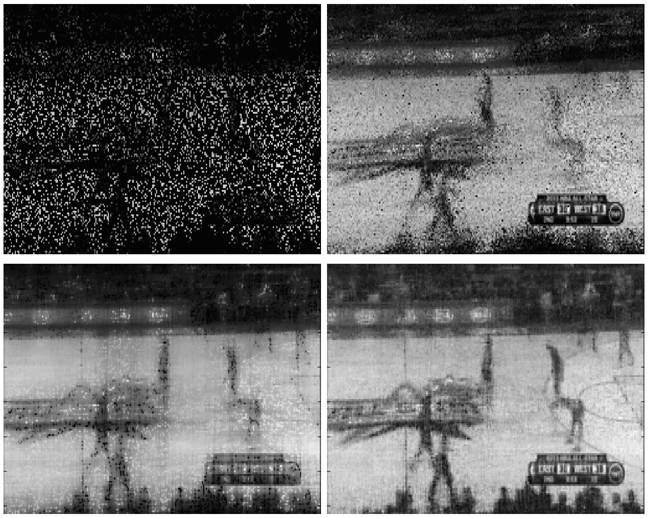


Figure 5. Tensor completion results for basketball video. **Upper left**: Sampled video(20%). **Upper right**: Nuclear norm minimization (vectorization and SVD based) result. **Lower left**: LRTC result. **Lower right**: TNN minimization result.

5. Tensor robust PCA

In this section we consider a "tensor robust principal component analysis" problem of recovering a low tensor-multi rank tensor $\mathcal{L}$ from a sparsely corrupted observation tensor. Similar to the matrix robust PCA case [4], suppose we have a third-order tensor $\mathcal{M}$ such that it can be decomposed as

$$\mathcal{M} = \mathcal{L} + \mathcal{S} \qquad (15)$$

where $\mathcal{L}$ has low tensor-multi-rank and $\mathcal{S}$ is sparse tensor. Here we focus on a case where the sparse tensor S is tube-

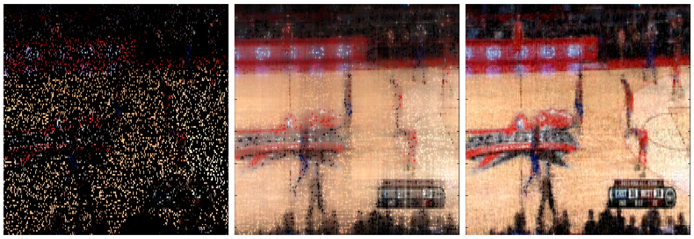


Figure 6. Recovery for color basketball video: **Left**: Sampled Video(10%). **Middle**: LRTC recovery. **Right**: Tensor-nuclear-norm minimization recovery

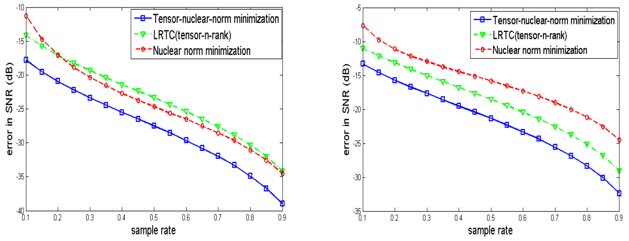


Figure 7. RSE (dB) plot against sampling rate **Left**: MERL video. **Right**: Basketball video

wise sparse as shown in Figure 8. To resolve the low rank and the sparse components given the observation $\mathcal{M}$ we consider the following optimization problem.

$$\begin{aligned} \min \quad & \|\mathcal{L}\|_{TNN} + \lambda \|\mathcal{S}\|_{1,1,2} \\ \text{subject to} \quad & \mathcal{M} = \mathcal{L} + \mathcal{S} \end{aligned} \tag{16}$$

where $\lambda > 0$ and the $\|\mathcal{S}\|_{1,1,2}$ for 3-D tensors is defined as $\sum_{i,j} \|\mathcal{S}(i,j,:)\|_F$.

An application where this is useful arises in multilinear imaging scenarios where some pixels have heavy noise on them and the task is to automatically locate such pixels and recover the video. Although this may be done by processing each frame but if the noise artifacts and video features are aligned, one needs to both detect the noise and estimate the corrupted video feature.

In order to solve the convex optimization problem of Equation (16) we use ADMM. Then we have the following recursion,

$$\mathcal{L}^{k+1} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_{TNN} + \frac{\rho}{2}\|\mathcal{L} + \mathcal{S}^k - \mathcal{M} + \mathcal{W}^k\|_F^2 \tag{17}$$

$$\mathcal{S}^{k+1} = \arg\min_{\mathcal{S}} \lambda \|\mathcal{S}\|_{1,1,2} + \frac{\rho}{2}\|\mathcal{L}^{k+1} + \mathcal{S} - \mathcal{M} + \mathcal{W}^k\|_F^2 \tag{18}$$

$$\mathcal{W}^{k+1} = \mathcal{W}^k + \mathcal{L}^{k+1} + \mathcal{S}^{k+1} - \mathcal{M} \tag{19}$$

where $\mathcal{W} = \rho \mathcal{Y}$. From section 4 we already have the solution to (17) if we transform this equation into the Fourier domain then the tensor-nuclear-norm of $\mathcal{L}$ will be the nuclear norm of blkdiag($\hat{\mathcal{L}}$). The closed form solution to Equation (18) is given by,

$$\mathcal{S}^{k+1}(i,j,:) = \left(1 - \frac{\lambda}{\rho \|\mathcal{S}^k(i,j,:)\|_F}\right)_+ \mathcal{S}^k(i,j,:) \tag{20}$$

where $i = 1, 2, ..., n_3$.

For experiment we consider a video, which is compressible in the t-SVD. We randomly corrupt video data by corrupting some pixels with heavy additive noise. We want to estimate the locations of such pixels using tensor robust PCA. The video used in this application is the basketball video with randomly chosen sparse pixels tubes along the third dimension. For each selected pixel we add random Gaussian noise on it. Figure (8) shows the original video(tensor) and the noise tensor. The size of each frame is 72 × 128 and the total number of frames is 80. The noisy pixel tubes within every 10 frames are consistent. We use the above ADMM algorithm to separate the original video and the noise. Our analysis (to be reported in a future paper) shows that the optimal choice of λ for tensor robust PCA is $\frac{1}{\sqrt{\max(n_1,n_2)}}$. We also perform matrix robust PCA on this noisy video data by vectorizing each frame, saving it as a column vector and then get a $n_1 n_2 \times n_3$ matrix. In this case the choice of λ is $\frac{1}{\sqrt{\max(n_1 n_2, n_3)}}$ [4].

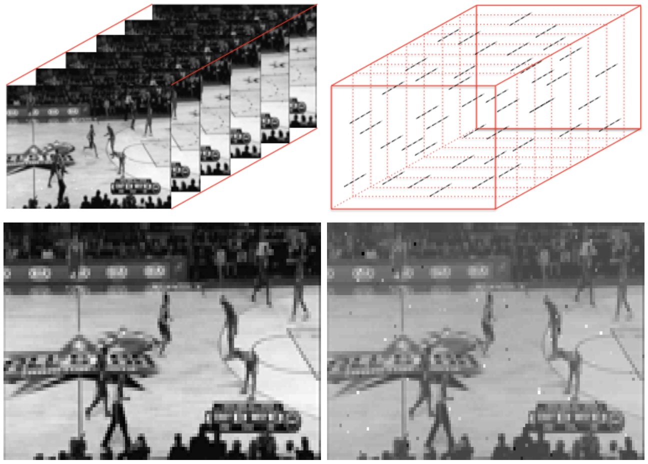


Figure 8. **Upper left**: Original video. **Upper right**: Noisy tensor. For 10 consecutive frames the locations of noisy pixels are the same and then selected randomly for the next 10 frames. **Lower left** 21st frame of the original video. **Lower right** 21st frame of the noisy video.

The result of both tensor robust PCA and matrix robust PCA is shown in Figure 9. From the results we can see that tensor robust PCA works very well on separating the noisy pixels from the video. However, the matrix robust PCA results in an almost fixed blurred background as the low rank part while some structure of the playground, moving people and the noise are recovered as the sparse part.

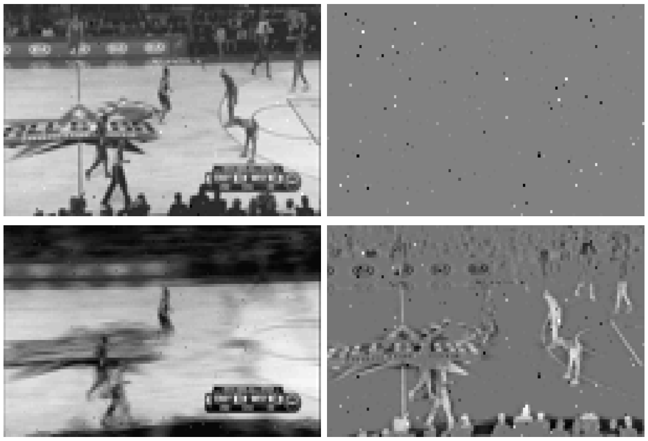


Figure 9. (21st frame shown) **Upper Left**: Low tensor multi-rank part recovered from tensor robust PCA. **Upper Right**: Sparse reconstruction from tensor robust PCA. **Lower left**: Low matrix rank part recovered from matrix robust PCA. **Lower right**: Sparse reconstruction from matrix robust PCA.

6. Conclusion and Future work

In this paper we presented novel methods for completion and de-noising (tensor robust PCA) of multilinear data using the recently proposed notion of tensor-SVD (t-SVD). As an application we considered the problem of video completion and de-noising from random sparse corruptions, and showed significant performance gains compared to the existing methods. The t-SVD based tensor analysis and methods can handle more general multilinear data as long as the data is shown to be compressible in the t-SVD based representation, as has been recently shown for pre-stack seismic data completion in [5]. Finding the necessary and sufficient conditions for recovery of low (multi)rank tensors using TNN from incomplete tensor data is an important theoretical problem and is an important area of future research.

7. Acknowledgements

This research was supported in part by the National Science Foundation grant NSF:1319653.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] K. Braman. Third-order tensors as linear operators on a space of matrices. *Linear Algebra and its Applications*, pages 1241–1253, 2010.

[3] J. F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[5] G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. 5d and 4d pre-stack seismic data completion using tensor nuclear norm (TNN). In *Society of Exploration Geophysicists (SEG) Expanded Abstracts*, 2013.

[6] G. Ely, S. Aeron, and E. L. Miller. Exploiting structural complexity for robust and rapid hyperspectral imaging. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2013.

[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[8] N. H. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover. Facial recognition with tensor-tensor decompositions. *SIAM Journal on Imaging Sciences*, 2012. Accepted Oct. 2012.

[9] E. Hazan, S. Kale, and S. Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *Journal of Machine Learning Research - Proceedings Track*, 23:38.1–38.13, 2012.

[10] I.V.Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[11] M. Kilmer, K. Braman, N. Hao, and R. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[12] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, Special Issue in Honor of G. W. Stewart's 70th birthday, Vol. 435(3):641–658, 2011. DOI: 10.1016/j.laa.2010.09.020.

[13] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[14] A. S. Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis Volume 2 (1995)*, 2(1/2):173–183, 1995.

[15] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):208–220, 2013.

[16] C. Martin, R. Shafer, and B. LaRue. An order-p tensor factorization with applications in imaging. *SIAM Journal on Scientific Computing*, 35(1):A474–A490, 2013.

[17] C. McGoldrick, W. J. Dowling, and A. Bury. Image coding using the singular value decomposition and vector quantization. In *Image Processing and its Applications, 1995., Fifth International Conference on*, pages 296–300, 1995.

[18] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller. An iterative reconstruction method for spectral CT with tensor-based formulation and nuclear norm regularization. In *The Second International Conference on Image Formation in X-Ray Computed Tomography*, 2012.

[19] G. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170(0):33 – 45, 1992.

[20] J. Wright, A. Ganesh, K. Min, and Y. Ma. Compressive principal component pursuit. preprint, `http://www.columbia.edu/~jw2966`, 2012.