Pak. J. Engg. Appl. Sci. Vol. 23 July, 2018 (p. 46–54)

Tensors for Improving Compressed Sensing Reconstruction

Saad Jamal, Khurram Khurshid*

Department of Electrical Engineering, Institute of Space Technology, Islamabad, Pakistan

* **Corresponding Author:** Email: khurram.khurshid@ist.edu.pk

Abstract

Compressed Sensing (CS) proposes a framework that any signal can be efficiently reconstructed by observing fewer measurements than required by the famous Shannon-Nyquist theorem, the observing signal must be sparse in some transform domain. Many methods have been proposed to further improve the quality of reconstructed images such as Group Sparsity, Structural Group Sparsity, Total-Variation etc. In this paper we unify tensor based approach with compressed sensing to efficiently reconstruct the original signal. Tensor based approach helps in preserving the intrinsic structure of the signal. Two methods of tensor decomposition Tucker and CANDE/PARAFAC (CP) are unified to give the best low-rank representation of the sparse signal.

Keywords: Compressed Sensing, Tensor Decomposition, Sparse Coding, Tucker-Decomposition, CP Decomposition

1. Introduction

Aim of compressed sensing is to reconstruct the original signal from set of many fewer required/sensed measurements then the proposed Shannon-Nyquist theorem, which states that in order to efficiently represent an analogue signal digitally one must keep the sampling at least twice the highest frequency of the signal [1, 2]. The sensed signal can only be reconstructed with the prior knowledge that signal was sparse (a signal having fewer non-zero elements) in some transform domain. Having low encoding complexity CS theory has gained quite a lot interest in the field of image processing. Compressed sensing differs from current method of compression as it introduces sparsity in a signal in some domain in which most of the coefficients are zero's, it also introduces a framework for linear sampling operator in which sampling and compression happens simultaneously to avoid any oversampling. By introducing sparsity in the signal, it can be again reconstructed with high probability from few linear projections (measurements) by convex optimization techniques.

CS based compression has strength in its asymmetric design i.e. by having simple computationally inexpensive encoder and a complex decoder, this asymmetric design is desirable in many signal and image processing applications such as taking satellite imaging where the encoder is less power consuming, the transmitted images can be decoded at the receiving station where computational resources have no constraints, this design is also desirable in situations where exposure to radiations over long period of time can harm the object such as X-Ray imaging [3], other applications include Magnetic Resonance Imaging (MRI) [4], radar imaging [5] and neutron imaging [6] and so on.

In CS framework a signal having higher degree of sparsity (having only few measurements) in one basis will lead to higher quality of recovery in another basis but both the basis will be incoherent with each other [7]. For stationary signals transforms such as discrete wavelet transform (DWT) [8] or discrete cosine transform (DCT) [9] etc. can be used to find sparsity bases but since natural images are nonstationary and there exists no universal basis that gives sufficient sparsity for the entire signal, in some other basis a signal contain different degree of sparsity but that basis may not ensure higher quality of recovery, this remains one of the main challenge in CS to find a basis function for a signal that contains a high degree of sparsity. Using transforms such as wavelets or gradient DCT are non-adaptive and are signal dependent that leads to poor rate-distortion.

CS framework has many obstacles in reconstruction, challenges from being computationally expensive to requiring huge memory for storing random sampling operators. Many algorithms have been introduced that take prior knowledge (such as statistical dependencies, structure, etc.) into account to improve reconstruction quality. Landweber iteration was proposed in [10] for fast reconstruction and removing blocking artifacts by imposing smoothness, in [11] Mun et al. combined the same framework of blocked CS with iterative projection

driven CS that added sparsity in the directional transform domain called BCS-SPL. Chen et al. in [12] extended the concept by combining residual and prediction to use the multihypothesis (MH) prediction, it takes the structural similarities (SSIM) from BCS-SPL reconstruction and use it as a stopping criterion. Zhang et al. [13] proposed structural group sparse representation (SGSR) by dividing an image into patches and taking non-local self-similarity, the patches that are most similar are grouped together and are used for adaptive dictionary to sparsely represent the nonlocal patches.

In this paper the approach used for [12] is used as initialization and coupled it with low-rank tensor decomposition. The reconstruction quality is improved by first dividing the image into patches with an overlap and stacking the patches that are most similar to each other, then using optimization problem we can find the true nonlocal self-similarity (NSS) from its initially reconstructed image. To achieve the tensor based reconstruction alternating direction method of multiplier (ADMM) algorithm is used to solve the above underdetermined problem.

The remainder of this paper is organized as follows. In Section 2 a brief overview of CS and tensor related work has been presented. Design implementation of tensor based approach is provided in Section 3. Section 4 gives the design implementation of the proposed method with mathematical preliminaries, in Section 5 performance of the proposed method is evaluated and Section 6 concludes this paper.

2. Background

2.1 Compressed Sensing

A signal $X \in R^N$ of size N dimension is said to be sparse in the Ψc domain (where Ψ is transform domain or basis and c is the vector that has sparse coefficients) if most of its elements in the transform domain are equal to zero, or it is nearly sparse if most of the elements are close to zero. The signal is reduced to M dimensions (such that $M < N$) by using a sensing matrix Φ of dimensions $M \times N$. Mathematically, $y = \Phi X$, where y is the measurement vector of size M dimensions. The goal is to recover X from y but recovering x is an ill-posed problem as the number unknowns exceeds the number of equations. By exploiting the fact that the original signal X was sparse then recovery of X is possible from these fewer linear measurements by solving the optimization problem.

$$\min_{x} \|\Psi X\|_p \quad \text{s.t } \|y\text{-}\Phi x\|_2 \qquad (1)$$

Note that x is sparse i.e. it has fewer non-zero elements ($k \ll N$). The sensing matrix Φ is random matrix, $\|\cdot\|_p$ is known as the ℓ^p norm, usually set to 0 or 1. The ℓ^0 norm to find the sparsest solution calculates all possible combinations of the non-zero elements in the vector mathematically $\min_x \|\Psi X\|_0 \quad \text{s.t } \|y\text{-}\Phi x\|_0$, this type of optimization is combinatorial complex i.e. NP-hard. To solve this system of non-linear equations ℓ^1 norm is used in which rather than minimizing the number of non-zeros entries instead take a vector and calculate its magnitude sum of absolute values for each entry x subject to same constraint. The exact reconstruction of x with high probability is only possible if X is sufficiently sparse and all other criteria of CS framework are met by both the transform domain and the sensing matrix.

2.2 Tensor Based Approach

A tensor is higher order generalization of scalars, vectors and matrices, the basic idea behind tensor factorization is to observe a tensor trough some latent representation. The order of a tensor r is the rank of a matrix which represents dimensions of a tensor (scalar is rank 0 tensor x, vectors are rank 1 tensors represented as $\{X_i\}$ while matrices are rank 2 tensors $X = \{X_{ij}\}$), hence multidimensional matrix is also known as multidimensional matrix or n-way array, Similar to vectors of size N which are represented as an array of one-dimension having the length N with any given basis, tensors can be represented as multidimensional array with respect to given basis. The intuition behind tensors is that if you have some data in the form of tensors the goal is to find explanation for the data as simple as possible.

Tensors assume that images are formed due to some multifactor concurrence that are content to linear analysis. Mathematically, tensors can be defined as multidimensional array $\mathbb{A} \in R^{I_1+I_2+\cdots+I_N}$ of order N, where each element of $\mathbb{A}$ is denoted as $a_{i_1\cdots i_n\cdots i_N}$ where $1 \le i_n \le I_n$. n is the nth index of order N tensor of size I_n in mode n (in a tensor different dimensions are called modes). Tensors are defined over a set of vectors that are associated with different quantities that can be length height etc. in case of image these components are rows and columns.

3. Intrinsic Tensors Based CS Reconstruction

After initial reconstructing of the image we want to improve our results, this can be achieved with help of tensors. To represent our 2D image in higher dimensions non-local self-similarity (NSS) can be used. NSS is another key feature of natural images in image reconstruction besides sparsity [14-16]. The concept behind using NSS is to gather together patches that have similar pattern but are spatially far from each other and stack them together in a form of 3D and to use high dimensional filtering of each group fragments to estimate the true signal. In natural images self-similarity holds repetitiveness of the texture and its structure. This is useful in retaining sharpness and edges of the images in a nonlocal consistency.

Usually tensor decomposition takes two form Tucker and Canonical Polyadic (CP) decompositions [27]. NSS will be combined with Tucker decomposition [17] [18] as well as CP decomposition [19] both are having insightful understanding of tensor sparsity will be blended together into tensor decomposition.

First, we will briefly introduced the NSS framework in which the image x with size N is divided into n overlapped patches $\Omega = \{y_i \in \mathrm{R}^d\}_{i=1}^N$ (where d is the pixel number of each patch), using block matching in [20] the patches that are most similar to a particular patch y_i are stacked togeather in a matrix form Y_i. The original nonlocal similar patch matrix X_i is recovered as

$$X_i = \underset{X}{\operatorname{argmin}} \|X\|_2 + \frac{\gamma}{2}\|Y_i - X\|_2^2 \qquad (2)$$

γ is the tradeoff parameter, $\|X\|_2$ is the order-2 sparsity of the true matrix X. By combining all X_i's we can reconstruct the estimated image x. The main challenge now lies in how to design a fitting sparsity measure for X.

Designing a sparsity for vectors/matrices is an easy task where only few non-zero elements exists however implementing sparsity in tensors is relatively complicated because in tensors we also must define the intrinsic correlation along different tensor modes.

Tucker decomposition (TD) also known as high order singular value decomposition factorizes a tensor X into multilinear polynomials such that it allows us to examine different factorization methods through some additional constraints, for an N order tensor. Tucker decomposition will factorize a tensor with one small core tensor X and N set of latent factor matrices (also known as bases) one in each mode, which can be mathematically defined as

$$\hat{X} = S \times_1 U_1 \times_2 U_2 \times_3 U_3 \times \cdots \times_N U_N \qquad (3)$$

where $S \in R^{r_1 \times r_2 \times \cdots N^{r_N}}$ is the core tensor which is dense but has dimensions less than the original tensor X the elements of core tensor shows the relation between components, $\times_1$ indicate the product between core tensor S and the bases matrix U_i in the i^{th} mode, and $U_i \in R^{I_i \times r_i} (1 \leq i \leq N)$ are the corresponding r_i orthogonal bases known as Tucker rank (or n-rank) along the i^{th} mode matrix unfolding of X of $N-$dimenson if all the unfolded vector bases are full rank then the tensor S is said to be a full-rank tensor, tensor unfolding of X is computed by arrange all elements belonging to X into corresponding rows or columns. The core tensor is calculated as

$$S = X \times_1 (U_1)^{\mathrm{T}} \times_2 (U_2)^{\mathrm{T}} \times_3 (U_3)^{\mathrm{T}} \times \cdots \times_N (U_N)^{\mathrm{T}} \qquad (4)$$

In TD the core tensor unlike the matrix sigma (Σ) in SVD (which is diagonal) does not have a special structure to it, it is a dense tensor that shows interaction between vectors U_1, U_2 and U_1.During unfolding process, the tensor is flattened into a matrix, for tensor unfolding TD uses the low-rank property for each of its modes, the mode i vector subspace are arranged to be the columns of the resulting matrix. Just like SVD the coefficients in the core tensor of TD are sorted in descending order along each tensor mode based on their importance to the tensor. The problem with such representation is that some of modes that might have strong correlation to the data but the coefficients in that mode will be decreasing very fast to zero, while some modes that are placed farther from zero because of low correlation to the data may contain a lot of non-zeros. Such representation of data throughs away a lot of useful information. Such understanding is useful for data compression they are less effective for signal reconstruction.

CP decomposition has been well studied for a low-rank tensor decomposition. CP decomposition takes an N order tenor $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$ and decomposes it into linear combination of sum of rank one tensors each rank one matrix is the vector loading in all modes, CP decomposition for a multivariate function can be defined as:

$$X = \sum_{i=1}^{r} c_i V_i = \sum_{i=1}^{r} c_i V_{i1} \circ V_{i2} \circ \cdots V_{iN} \qquad (5)$$

Where ∘ the Kronecker product, here r is known as tensor rank, and c is the vector of the weighs that imposes coefficients on Kronecker bases V_i. In CP decomposition the core tensor is super diagonal, but CP decomposition suffers from ill-posedness i.e. low-rank approximation does not exists. The core tensor being diagonal is desirable because only combination of a few affiliated tensor bases will be used, though the core tensor will be highly sparse but along each tensor mode the subspace will not be unique. Combining the sparsity understanding from both techniques, taking low-rank tensor property for core tensor from Tucker decomposition and sparse Kronecker bases from CP decomposition. Mathematically

$$S(X) = t\|S\|_0 + (1-t)\prod\nolimits_{i=1}^{N}\left(X_{(i)}\right) \tag{6}$$

The core tensor S is obtained by first taking CP decomposition to constraint the number of tensor modes representing objective tensor, once the number of modes restrained then we will take Tucker decomposition to regularize the low rank property of the subspace spanned on each tensor mode. Inspired by the work of Xie *et al.* [21] used for denoising of multi-spectral images similar concept will be used in this paper for CS recovery and to the best of our knowledge this method has not been adapted for 2D images.

4. Numerical Algorithm Implementation

Here we discuss the implementation of the numerical algorithms.

4.1 Alternating Direction Method of Multiplier (ADMM) Algorithm

ADMM decomposes large global problem into parallel small sub-problems and by coordinating local solution it reaches to global convergence it merges converging property of multiple multiplier with dual ascent decomposition. Note that l_0 in eq. (6) only can have discrete values which essentially makes it a combinatorial convex optimization problem which makes it quite difficult to solve due to non-differentiability. We therefore will introduce first the penalty parameter λ which is an effective way to solve the optimization problem

$$\min_X P_{ls}(S) + \lambda\prod\nolimits_{j=1}^{2} P_{ls}^{*}\left(X_{(j)}\right) + \frac{\beta}{2}\|y_i - X\|_F \tag{7}$$

The penalty parameter $\lambda = {1-t}/{t}$ controls the tradeoff between the two terms in (7). Then invoking alternating direction method of multipliers (ADMM) [22] [23] [28] to eq. (7) and $P_{ls}(A) = \sum_{i_1,i_2}\log\left(\left|a_{i_1,i_2}+\varepsilon\right|\right), P_{ls}^{*}(A) = \sum_j \log\left(\sigma_j(A)+\varepsilon\right)$ an effective way to tackle large scale optimization problem. For optimizing eq. (7) we first split it by introducing two auxiliary tensors M_j $(j = 1,2)$. The optimization problem now becomes

$$\min_{S,M_j,U_j} P_{ls}(S) + \lambda\, P_{ls}^{*}\prod\nolimits_{i=1}^{N}\left(M_{j(j)}\right) + \frac{\beta}{2}\|y_i - S\times_1 U_1\times_2 U_2\|_2^2 \quad s.t.\ S\times_1 U_1\times_2 U_2 - M_j = 0,\ U_j^T U_j = I,\ j = 1,2 \tag{8}$$

$M_{j(j)} = \text{unfold}\, M_j$ ADMM algorithm uses a variant of augmented Lagrangian function [24] for X minimization. The augmented Lagrangian function with this form is:

$$L_\mu(S, M_1, M_2, U_1, U_2) = P_{ls}(S) + \lambda\, P_{ls}^{*}\prod\nolimits_{i=1}^{2}\left(M_{j(j)}\right) + \frac{\beta}{2}\|y_i - S\times_1 U_1\times_2 U_2\|_2^2 + \sum\nolimits_{j=1}^{2}\langle S\times_1 U_1\times_2 U_2 - M_j, P_j\rangle + \sum\nolimits_{j=1}^{2}\frac{\mu}{2}\left\|S\times_1 U_1\times_2 U_2 - M_j\right\|_2^2$$

Where $P_{j's}$ being Lagrangian multipliers, μ is the non-negative dual update scalar step, U_j is the dual variable that must satisfy $U_j^T U_j = I$. Now we can use the ADMM algorithm to minimize $P_{ls}(S)$. Now that other parameters are fixed we can update S as a sub-problem by solving

$$L_\rho(S, M_1, M_2, U_1, U_2).$$

$$\min_S bP_{ls}(S) + \frac{1}{2}\|S\times_1 U_1\times_2 U_2 - \mathcal{O}\|_2^2 \tag{9}$$

where $b = \frac{1}{\beta+3\mu}$ and $\mathcal{O} = \frac{\beta y_i + \sum_j(\mu M_j - P_j)}{\beta+3\mu}$. Since

$$\|D\times V\|_2^2 = \|D\|_2^2, \forall\, V^T V = I, \tag{10}$$

On each mode, mode j produces U_j^T, now eq. (9) becomes:

$$\min_S bP_{ls}(S) + \frac{1}{2}\|S - \mathcal{Q}\|_2^2 \tag{11}$$

$$\text{where } \mathcal{Q} = \mathcal{O}\times_1 U_1^T\times_2 U_2^T$$

$\mathcal{Q}$ has a closed-form solution which can be solved as proven in [25]. As the first term is non-differentiable we will use a closed-form solution

$$S^{+} = D_{b.\epsilon}(\mathcal{Q}) \tag{12}$$

Where $D_{b.\epsilon}(.)$ is a soft thresholding operator which is defined as:

$$D_{b.\epsilon}(a) = \begin{cases} 0, & if\ c_2 \le 0 \\ \text{sign}(a)\left(\frac{c_1 + \sqrt{c_2}}{2}\right), & if\ c_2 > 0 \end{cases} \tag{13}$$

where $c_1 = |a| - \varepsilon, c_1 = (c_1)^2 - 4(b - \varepsilon)|a|$. In ADMM algorithm U_j is updated while keeping the other parameters fixed, we can also obtain the closed-form as follows:

$$\min_{U_1} \|S \times_1 U_1 \times_2 U_2 - \mathcal{O}\|_2^2 \tag{14}$$

Here taking U_1 as an example, U_1 is updated while keeping U_2, and other parameters fixed. Using eq. (10) we can say (14) is equivalent to

$$\max_{U_1^T U_1 = I} \langle A_1, U_1 \rangle \tag{15}$$

where $A_1 = \left(O_1 \text{unfold}_1(S \times_2 U_2)\right)$.Similarly, U_2 can be updated in similar fashion

$$\max_{U_k^T U_k = I} \langle A_k, U_k \rangle \tag{16}$$

For eq. (16) we can obtain its closed-form solution by using the following conclusion

Using eq. (12) we can update U_k as:

$$U_k^{+} = B_k C_k^T \tag{17}$$

where the SVD decomposition of $A_k = B_k D C_k^T$.

Keeping other parameters and $M_j$where $(j \neq k)$ fixed M_k can be updated using the following equation

$$\min_{M_k} \partial_k P_{ls}^{*}\left(M_{k(k)}\right) + \frac{1}{2}\left\|\mathcal{L} + \frac{1}{\mu}\mathcal{P}_k - M_k\right\|_2^2 \tag{18}$$

Where $\partial_k = \left(\frac{\lambda}{\mu}\prod_{j \neq k} P_{ls}^{*}\left(M_{j(j)}\right)\right)$ and

$\mathcal{L} = S \times_1 U_1 \times_2 U_2$. Now we can update M_k.

$$M_k^{+} = fold_k\left(V_1 \sum\nolimits_{\partial_k} V_2^T\right) \tag{19}$$

where

$$\sum\nolimits_{\partial_k} = \text{diag}\left(D_{\partial_k,\varepsilon}(\sigma_1), D_{\partial_k,\varepsilon}(\sigma_2), \cdots, D_{\partial_k,\varepsilon}(\sigma_n)\right)$$

and $V_1 \text{diag}(\sigma_1, \sigma_2, \cdots \sigma_n) V_2^T$ is the SVD of $\text{unfold}_k\left(\mathcal{L} + \frac{1}{\mu}\mathcal{P}_k\right)$.

5. Performance Evaluation

In this section, the performance of the proposed tensor based CS reconstruction is evaluated and compared with other existing CS recovery methods in terms of image reconstruction quality. Performance was evaluated on twelve test images shown in Fig. 4, results were compared to four CS recovery methods, block-based compressed-sensing with smoothed projected-Landweber (BCS-SPL) [11] [29], multi-hypothesis-BCS-SPL (MH-BCS-SPL) [12] [30] and total variance (TV) [26] [31] the codes used were provided by their authors. To evaluate the quality of reconstructed image peak signal-to-noise ratio (PSNR) is used which measures the magnitude of noise to the peak value of the signal (for an 8-bit image the peak value is 255). Mathematically PSNR is defined as

$$PSNR = 10 \log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

where MAX_f stands for maximum peak value of the signal, and MSE stands for Mean Squared Error. In image processing we want to keep MSE as little as possible, therefore by having small MSE we get high PSNR which means our reconstructed image is nearer to the original image. Images used were of 256×256 pixels in size. Quality of the reconstructed images might vary due to random generation of the sensing matrix. Parameter λ is used as tradeoff between two parameters is set to 10, the other parameter β depends on cv, where c in our experiment was set to 1.5×10^{-3}. It is to be noted that the values assigned to the parameters are general and can be changed depending on the type of image used. The evaluating metric used was PSNR to compare the results of the data set by varying sampling rate at 0.1, 0.3 and 0.5 (0.1, 0.3 and 0.5 mean that 90%, 70% and 50% percent of the samples were missing) for all 12 test images as shown in Table 1-3. MATLAB was used to conduct the experiments.

In our experiment, results show that using tensors in compressed sensing reconstruction shows quite promising results even at lower sampling rate Fig. (1-3). In Fig. 5 when sampling rate is 10 the reconstructed image of Lena is quite clearer, smooth and contain well preserved boundaries, where as in TV based algorithm at this sampling rate performed quite poor as it introduced staircase artifacts and over-smoothed the image neither does it suffer from blocking

artifacts which is the shortcoming BCS-SPL suffer due to its block based approach. Visual quality comparison of the recovered images is provided at the end of this section.

Table 1: Comparing PSNR of different CS recovery method (dB) at sampling rate 10%

Method	BCS-SPL	MH	TV	Tensor
Airplane	23.15	23.67	22.47	23.53
Barbara	24.69	25.21	21.75	25.42
Boats	23.39	23.96	22.71	24.16
Cameraman	21.87	22.39	23.40	22.79
Couple	23.30	23.73	22.65	24.03
Cub	20.67	20.68	23.02	20.70
Einstein	26.91	27.87	25.73	28.42
House	25.28	25.87	23.97	26.10
Lena	25.98	26.41	22.92	26.76
Mandrill	21.49	21.43	23.35	21.73
Peppers	26.62	26.91	22.20	27.77
Soyuz	19.86	23.72	24.44	23.70
Avg.	23.60	24.32	23.22	24.59

Table 2: Comparing PSNR of different CS recovery method (dB) at sampling rate 30%

Method	BCS-SPL	MH	TV	Tensor
Airplane	29.68	29.68	28.23	30.52
Barbara	28.79	31.65	27.67	33.39
Boats	28.06	28.99	27.75	30.23
Cameraman	27.88	27.75	31.29	28.91
Couple	27.28	28.50	27.33	30.36
Cub	23.71	24.06	25.99	24.52
Einstein	31.37	32.19	31.63	33.65
House	29.84	30.98	28.66	32.38
Lena	31.87	32.23	29.3	33.03
Mandrill	24.44	25.16	26.05	25.92
Peppers	33.47	33.47	28.86	35.59
Soyuz	29.06	29.70	31.93	30.20
Avg.	28.79	29.53	28.72	30.73

Table 3: Comparing PSNR of different CS recovery method (dB) at sampling rate 50%

Method	BCS-SPL	MH	TV	Tensor
Airplane	33.64	33.64	33.80	35.16
Barbara	31.70	34.81	32.11	37.03
Boats	31.03	32.36	32.04	34.71
Cameraman	31.34	30.48	32.32	32.66
Couple	30.03	32.17	31.35	34.80
Cub	25.81	26.29	27.08	27.19
Einstein	33.91	34.81	35.66	36.62
House	32.74	34.09	32.70	36.19
Lena	35.49	35.55	33.78	37.10
Mandrill	26.63	27.7	28.90	29.01
Peppers	37.02	36.92	33.71	39.33
Soyuz	34.07	34.13	37.83	34.74
Avg.	31.95	32.75	32.61	34.55

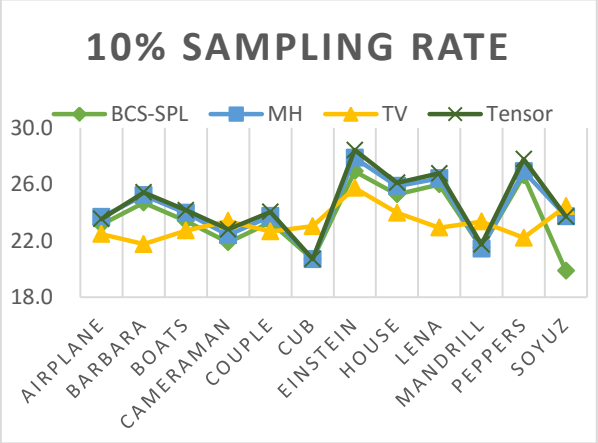

Fig. 1: PSNR comparison at 10% sampling rate

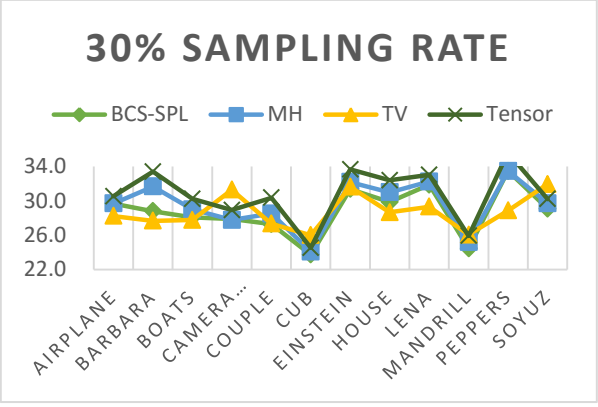

Fig. 2: PSNR comparison at 30% sampling rate

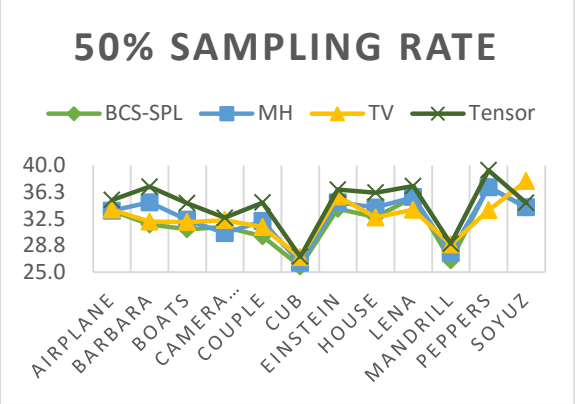

Fig. 3: PSNR comparison at 50% sampling rate

Fig. 4: Test Images (256x256) L-R: 'Lena', 'cub', 'boats', 'barbara', 'airplane', 'cameraman', 'couple', 'mandrill', 'goldhill', 'peppers', 'soyuz', 'einstein'

Fig. 5: Reconstruction of 'Lena': at subsampling 0.1 (PSNR [dB]) L-R: (a) BCS-SPL (25.98), (b) MH (26.41), (c) TV (22.92), (d) Tensor (26.76)

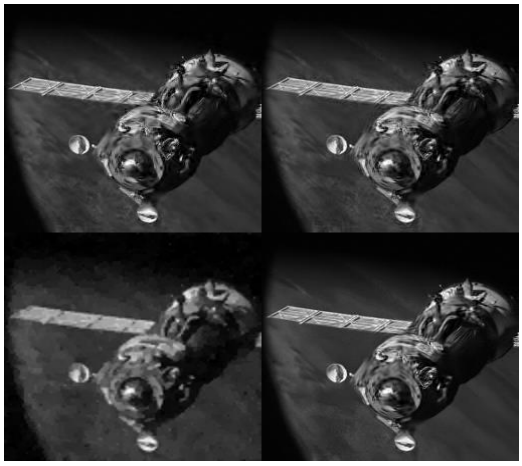

Fig. 6: Reconstruction of 'soyuz': at subsampling 0.3 (PSNR [dB]) L-R: (a) BCS-SPL (29.06), (b) MH (29.70), (c) TV (31.93), (d) Tensor (29.73)

Fig. 7: Reconstruction of 'cub': at subsampling 0.5 (PSNR [dB]) L-R: (a) BCS-SPL (25.81), (b) MH (26.29), (c) TV (27.08), (d) Tensor (27.19)

6. Conclusion

Compressed sensing is an alternative way to how we compress our data there is a lot of room for new research to improve the quality of CS reconstruction. In [1] and [2] Candes and Donoho proposed the idea that images can be reconstructed from only few observations with by using ℓ^1 minimization. Lu [10] used Landweber algorithm to improve his result. In [11] Fowler introduced block based approach to enhance quality of reconstructed CS images.

In this paper tensor based CS reconstruction was proposed that considers both Tucker and CP decompositions along with NSS of natural images. The model was solved through means of ADMM algorithm. The concept of tensor decomposition is to represent the data in the form of tensor and seek

as simple explanation for it as possible. We do this by first finding similar patches in the initially reconstructed image by NSS and then decompose each patch as a sum of few tensor products. The idea behind this to have very concise representation of our tensors that makes subsequent computation efficient and gives intrinsic information about the data. Experimental results showed that tensor can be used to increase the quality of sparse image reconstruction.

7. References

[1] Candes, E., & Wakin, M. (2008). An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, *25*(2), 21-30.

[2] Donoho, D. (2006). Compressed Sensing. *IEEE Transaction On Information Theory*, *52*(4), 1289-1306.

[3] Masoum, A., Meratnia, N., & Havinga, P. (2013). A Distributed Compressive Sensing Technique for Data Gathering in Wireless Sensor Networks. *Procedia Computer Science*, *21*, 207-216.

[4] Lustig, M., Donoho, D., Santo, J., & Pauly, J. (2008). Compressed Sensing MRI. *IEEE Signal Processing Magazine*, (25(2)), 72-82...

[5] Potter, L., Ertin, E., Parker, J., & Cetin, M. (2010). Sparsity and Compressed Sensing in Radar Imaging. *Proceedings Of The IEEE*, *98*(6), 1006-1020.

[6] Jin, W., Liu, Z., & Li, G. (2015). Block-Based Compressed Sensing for Neutron Radiation Image Using WDFB. *Advances In Optoelectronics*, *2015*, 1-5.

[7] Candes, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions On Information Theory*, *52*(2), 489-509.

[8] M. B. Wakin *et al.*, "Compressive imaging for video representation and coding," *Pict. Coding Symp.*, vol. 1, p. 13, 2006.

[9] Stankovic, V., Stankovic, L., & Cheng, S. (2008). Compressive video sampling. In *16th European Signal Processing Conference*. Lausanne, Switzerland.

[10] Gan, L. (2007). Block Compressed Sensing of Natural Images. In *15th International Conference on Digital Signal Processing* (pp. 403-406).

[11] Mun, S., & Fowler, J. (2009). Block compressed sensing of images using directional transforms. In *16th IEEE International Conference on Image Processing* (pp. 3021-3024).

[12] C. Chen, E. W. Tramel, & J. E. Fowler. (2011). Compressed-sensing recovery of images and video using multihypothesis predictions. In *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)* (pp. 1193-1198).

[13] Zhang, J., Zhao, D., Jiang, F., & Gao, W. (2013). Structural Group Sparse Representation for Image Compressive Sensing Recovery. In *Data Compression Conference* (pp. 331-340).

[14] Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2009). Non-local sparse models for image restoration. In *IEEE 12th International Conference on Computer Vision* (pp. 2272-2279).

[15] Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., & Zhang, B. (2014). Decomposable Nonlocal Tensor Dictionary Learning for Multispectral Image Denoising. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2949-2956).

[16] Zha, Z., Liu, X., Zhang, X., Chen, Y., Tang, L., & Bai, Y. et al. (2016). Compressed sensing image reconstruction via adaptive sparse nonlocal regularization. *The Visual Computer*, *34*(1), 117-137.

[17] Sidiropoulos, N., Lathauwer, L., Fu, X., Huang, K., Papalexakis, E., & Faloutsos, C. (2017). Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Transaction On Signal Processing*, *65*(13), 3551-3582.

[18] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., & PHAN, H. (2015). Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, *32*(2), 145-163.

[19] Kolda, T., & Bader, B. (2009). Tensor Decompositions and Applications. *SIAM Review*, *51*(3), 455-500.

[20] Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions*

On Image Processing, *16*(8), 2080-2095.

[21] Qi Xie, Qian Zhao, Deyu Meng, Zongben Xu, Shuhang Gu, Wangmeng Zuo, & Lei Zhang. (2016). Multispectral Images Denoising by Intrinsic Tensor Sparsity Regularization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1692-1700).

[22] Ghadimi, E., Teixeira, A., Shames, I., & Johansson, M. (2015). Optimal Parameter Selection for the Alternating Direction Method of Multipliers (ADMM): Quadratic Problems. *IEEE Transactions On Automatic Control*, *60*(3), 644-658.

[23] Boyd, S. (2010). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations And Trends® In Machine Learning*, *3*(1), 1-122.

[24] Sedghi, H., Anandkumar, A., & Jonckheere, E. (2012). Multi-Step Stochastic ADMM in High Dimensions: Applications in Sparse Optimization and Noisy Matrix Decomposition, 1-33. Retrieved from http://Newport.Eecs.Uci.Edu.

[25] Gong, P., Zhang, C., Lu, Z., Huang, J., & Ye, J. (2013). A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* (pp. II-37-II-45). Atlanta, GA, USA.

[26] C. Li, W. Yin, and Y. Zhang, "TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Algorithms," 2009.

[27] Rabanser, S., Shchur, O., & Günnemann, S. (2017). Introduction to Tensor Decompositions and their Applications in Machine Learning, 1-13.

[28] Teixeira, A., Ghadimi, E., Shames, I., Sandberg, H., & Johansson, M. (2016). The ADMM Algorithm for Distributed Quadratic Problems: Parameter Selection and Constraint Preconditioning. *IEEE Transactions On Signal Processing*, *64*(2), 290-305.

[29] Retrieved from http://my.ece.msstate.edu/faculty/fowler/Publications/SourceCode/bcs-spl-1.5-1.tar.gz

[30] Retrieved from http://my.ece.msstate.edu/faculty/fowler/Publications/SourceCode/mh-bcs-spl-1.0-1.tar.gz

[31] Retrieved from http://www.caam.rice.edu/~optimization/L1/TVAL3/