Compressive Tensor Sampling with Structured Sparsity

Yong Li*, Wenrui Dai†, and Hongkai Xiong*

*Department of Electronic Engineering
Shanghai Jiao Tong Univ.
Shanghai 200240, China
{marsleely,xionghongkai}@sjtu.edu.cn

†Department of Biomedical Informatics
University of California, San Diego
CA 92093, USA
wed004@ucsd.edu

Abstract

Conventional Compressive Sensing (CS) obscures the intrinsic structures of multidimensional signals with the vectorized representation. Although tensor-based CS methods can preserve the intrinsic multidimensional structures with reduced computational complexity, their sampling efficiency and recovery performance are degraded with the assumption of standard/simple sparsity. This paper proposes a general and adaptive model that incorporates structured sparsity into tensor representation to fit the varying nonstationary statistics of multidimensional signals. To guarantee the block sparsity, subspace clustering is adopted to adaptively generate the union of tensor subspaces with its basis of each tensor subspace learned for optimized representation. For sampled tensors, the stable recovery algorithm is developed to achieve desirable recovery performance using fewer degrees of freedom. Moreover, the proposed model inherits the merit from tensor-based CS to alleviate the computational and storage burden in sampling and recovery. Experimental results demonstrate that the proposed model can achieve better recovery performance in video sampling in comparison to the state-of-the-art tensor-based method.

1 Introduction

Compressive Sensing (CS)[1] has emerged as a promising framework for signal acquisition and recovery. Recently, CS has been applied to video acquisition and recovery [2]–[4], which relieves the burden of video encoder by relaxing the hardware limitations and reducing the number of measurements to be sampled. Given certain basis, effective reconstruction can guarantee the stable recovery from the sparse representation at the decoder side. However, vectorized representation of signals in conventional CS would destroy the intrinsic structures of multidimensional signals or tensors, e.g. images and videos. Thus, the sampling efficiency is degraded by the large-scale sampling matrix led by the vectorization of high-dimensional signals.

To overcome the deficiency of sampling high-order tensors, multidimensional CS techniques [6]–[9] have been developed for practical implementations. In [6] multidimensional signals were reconstructed by finding the best rank-R tensor approximation. Later, the Kronecker-CS model[7] assumes that multidimensional signals admit sparse representations over separable dictionaries constructed by Kronecker product matrices. In [8], multi-way CS adopted Kronecker sensing structure to sample tensors and fits a low-rank model in compressed domain. The low-rank CANDECOMP/PARAFAC(CP) tensor can be recovered by per-mode ℓ_0/ℓ_1 decompression. Generalized Tensor Compressive Sensing (GTCS) [9] utilized Kronecker sensing structure and ℓ_1-minimization approach per mode for recovery. However, these methods

1068-0314/16 $31.00 © 2016 IEEE
DOI 10.1109/DCC.2016.16

merely consider standard/simple sparsity by assuming that the signal of interest lives in a single tensor subspace, which would be inapplicable and inefficient for images and videos. To address this problem, structured sparsity [5, 11, 12] has been widely considered for CS to enhance the simple sparsity with additional prior knowledge of the underlying structures in signals, e.g. tree sparsity and block/group sparsity. For example, tree-sparsity model utilizes the prior information of successive relationship among wavelet coefficients based on the fact that nonzero wavelet coefficients tend to live in a rooted and connected tree. While block-sparsity model supposes that nonzero coefficients appear to cluster together in a few blocks. Inspired by block sparsity, the union of subspaces (UoS [10]) model was studied to significantly reduce the number of measurements, where sampled signals was supposed to live in a union of subspaces with fixed linear operators. [12] developed a robust block sparse recovery method with a mixed ℓ_2/ℓ_1 program constrained by block restricted isometry property (RIP). Derived from orthogonal basis like wavelet and DCT, however, they are rigid for signals with varying signal regularities, especially video sequences. Recently, union of data-driven subspaces (UoDS [5]) model leveraged block sparsity to obtain an adaptive and nonlocal basis for compressive video sampling. These models with structured sparsity can make a stable and efficient recovery with a reduced degrees of freedom. Thus, this paper proposes an adaptive general model with structured sparsity for video sampling and reconstruction.

In this paper, we propose a compressive tensor sampling model with structured sparsity for compressive video sampling, where Kronecker product framework is developed to admit block-sparse representation over the learned multilinear bases for tensor modes. The union of data-driven subspaces are adaptively generated by sparse subspace clustering to decompose and represent multidimensional signals based on their structured sparsity. To optimize the tensor subspace based representation, a multilinear subspace learning (MSL) method is adopted to derive the basis for each subspace. The proposed model maintains the intrinsic structures of tensors to be sampled with reduced computational complexity and storage cost. Furthermore, multidimensional signals can be recovered with a desirable performance using fewer degrees of freedom. Stable recovery is demonstrated to be guaranteed by block sparse representation over the derived adaptive bases.

2 Compressive Tensor Sampling with Structured Sparsity

2.1 Tensor representation with structured sparsity

Given arbitrary tensor $\mathbf{X}$, it can be represented according to standard multilinear algebra [15],

$$\mathbf{X} = \Theta \times_1 \Psi^{(1)} \times_2 \Psi^{(2)} \times \cdots \times_N \Psi^{(N)} = (\Psi^{(1)} \otimes \Psi^{(2)} \otimes \cdots \otimes \Psi^{(N)})\Theta(1)$$

where Θ is the tensor representation for $\mathbf{X}$, based on the n-mode orthogonal basis matrix$\Psi^{(n)} = (\psi_1^{(n)} \psi_2^{(n)} \cdots \psi_{I_n}^{(n)})$, $\times_n$ denotes the n-mode product of a tensor by a matrix, and $\otimes$ is the Kronecker product. Given the matrix$\Psi^{(n)}$, the n-mode product of Θ is defined as

$$\mathbf{X} = \Theta \times_n \Psi^{(n)} \Leftrightarrow \mathbf{X}_{(n)} = \Psi^{(n)} \Theta_{(n)}, \tag{2}$$

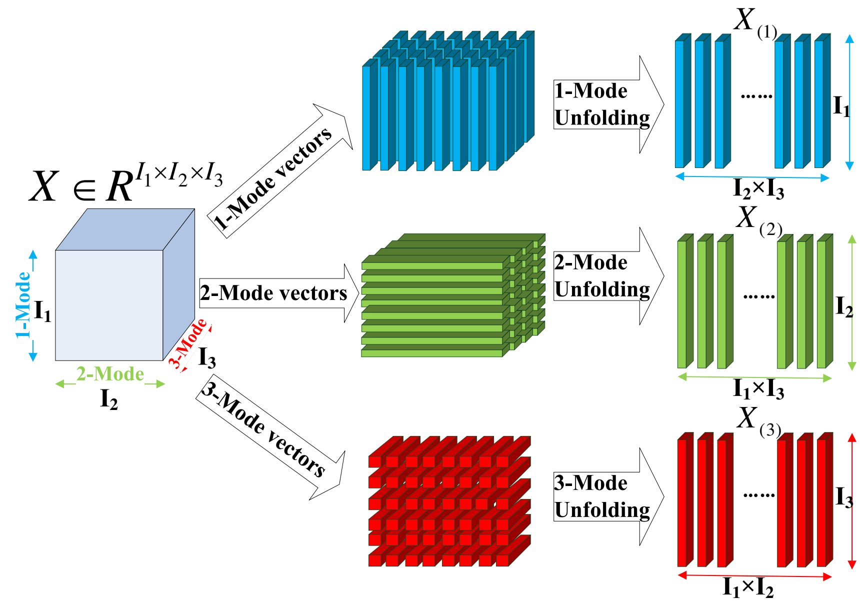


Figure 1: Illustrative diagram of the n-mode unfolding of Tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where $n = 1, 2, 3$. $X_{(n)}$ is the n-mode unfolding matrix for $n = 1, 2, 3$.

where $\mathbf{X}_{(n)}$ is the n-mode unfolding matrix of tensor $\mathbf{X}$. Fig.1 illustrates the n-mode unfolding of tensor $\mathbf{X}$ with $n \leq 3$.

When $\mathbf{X}$ lies in a tensor subspace $\mathcal{S}$, the n-mode vectors admit simple sparse representation over the n-mode basis. To incorporate the structured sparsity into tensor representation, we assume that the tensor $\mathbf{X}$ to be sampled is lying on a union of data-driven subspaces (UoDS[5]) $\mathcal{U}^* = \bigcup_i \mathcal{S}_i$. Given the training set of tensors $X = [X_1, X_2, \ldots, X_p]$, we first generate t groups $G = [G_1, \ldots, G_t]$ by subspace clustering (e.g. SSC[13]), with each group G_i containing p_i tensors belong to the same subspace $\mathcal{S}_i$, and $\sum_{i=1}^{t} p_i = p$.

Subsequently, the basisΨ_i of each tensor subspace $\mathcal{S}_i$ can be learned from training set $G = [G_1, G_2, \ldots, G_t]$ by using multilinear subspace learning. We adopt MPCA[17] for each cluster G_i to learn corresponding linear tensor subspace separately. MPCA obtains a tensor subspace that captures most of the variation in the original tensor objects and projects $\mathbf{X}_j \in G_i$, $j = 1, \ldots, p_i$ from the original tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ onto the tensor subspace $\mathcal{S}_i = \mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \cdots \otimes \mathbb{R}^{P_N}$ with $P_n \ll I_n$. The projection (representation)$\Theta_j = \mathbf{X}_j \times_1 \Psi_i^{(1)^T} \times_2 \Psi_i^{(2)^T} \times \cdots \times_N \Psi_i^{(N)^T}$, basis $\Psi_i^{(n)} \in R^{I_n \times P_n}$ spans the n-mode linear space $\mathbb{R}^{P_n}$ of $\mathcal{S}_i$. Thus, MPCA solves the problem:

$$\{\Psi_i^{(n)}, n = 1, \ldots, N\} = \arg \max_{\Psi_i^{(1)}, \ldots, \Psi_i^{(N)}} \sum_{j=1}^{p_i} \|\Theta_j - \bar{\Theta}_i\|_F^2 \quad (3)$$

where the mean tensor projection $\bar{\Theta} = (1/p_i) \sum_{j=1}^{p_i} \Theta_j$, p_i is the number of tensors in the i-th group G_i.

Therefore, we can obtain the basis $\hat{\Psi} = \Psi^{(1)} \otimes \Psi^{(2)} \otimes \cdots \otimes \Psi^{(N)}$ of the union of tensor subspaces $\mathcal{U}^*$, where $\Psi^{(n)} = [\Psi_1^{(n)}, \Psi_2^{(n)}, \ldots, \Psi_t^{(n)}]$ is the set of bases learned from the group G_i. In comparison to previous multidimensional CS models, the basis $\hat{\Psi}$ is adaptively derived so that it can capture the non-stationarity in multidimensional signals. Furthermore, each mode of tensors to be sampled admits block-sparse representation over$\Psi^{(n)}$, which can reduce the degrees of freedom for solution, and consequently, make a stable and efficient recovery. Remarkably, the basis$\Psi^{(n)} \in \mathbb{R}^{I_n \times r_n}$ should be over-complete with $r_n = tP_n > I_n$. In the following subsections,, we formulate the compressive tensor sampling and recovery for multidimensional signals.

2.2 Compressive tensor sampling and stable recovery

For multidimensional signals like images and videos, the efficiency of conventional compressive sampling is degraded by the large-scale sampling matrix led by the vectorization. Thus, this paper proposes the compressive tensor sensing (CTS) that directly samples the Nth-order tensor (multidimensional signal) in each of its mode under the assumption that the tensor lives in a union of tensor subspaces with lower dimensions.

Definition 1 (Compressive Tensor Sampling) *Given an Nth-order tensor* $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *who is* K_1*-*K_2*-...-*K_N*-block-sparse, the compressive tensor sampling (CTS) is defined by:*

$$\begin{aligned} \mathcal{Y} &= \hat{\Phi}\mathbf{X} = (\Phi_1 \otimes \Phi_2 \otimes \cdots \otimes \Phi_N)\mathbf{X} \\ &= \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \times \cdots \times_N \Phi_N, \end{aligned} \tag{4}$$

where $\otimes$ *is the Kronecker product,* $\mathcal{Y} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$*, and* $\Phi_n \in \mathbb{R}^{M_n \times I_n}$ *is the sensing matrix for n-mode. Here,* $K_n < M_n \ll I_n$*,* $n = 1, \ldots, N$*.*

Recalling the Kronecker product form of $\mathbf{X}$ in Eq. (1),

$$\mathbf{X} = \hat{\Psi}\Theta = (\Psi^{(1)} \otimes \Psi^{(2)} \otimes \cdots \otimes \Psi^{(N)})\Theta. \tag{5}$$

$\mathbf{X}$ can be sampled with Ψ and Φ by combining Eq. (4) and (5).

$$\begin{aligned} \mathcal{Y} &= \hat{\Phi}\mathbf{X} = \hat{\Phi}\hat{\Psi}\Theta = (\Phi_1 \otimes \Phi_2 \otimes \cdots \otimes \Phi_N)(\Psi^{(1)} \otimes \Psi^{(2)} \otimes \cdots \otimes \Psi^{(N)})\Theta \\ &= (\Phi_1\Psi^{(1)} \otimes \Phi_2\Psi^{(2)} \otimes \cdots \otimes \Phi_N\Psi^{(N)})\Theta = (A^{(1)} \otimes A^{(2)} \otimes \cdots \otimes A^{(N)})\Theta(6) \end{aligned}$$

Here, $A^{(n)}$ is the n-mode sensing matrix. Fig. 2 provides an example for compressive tensor sampling.

For self-containment of this paper, Proposition 1 demonstrates the uniqueness and stability conditions for the compressive tensor sampling. For each mode n, given the convex hull $\mathcal{S}_{ij}^{(n)}$ of two different data-driven tensor subspaces $\mathcal{S}_i^{(n)} \bigcup \mathcal{S}_j^{(n)}$, the maximum dimension of $\mathcal{S}_{ij}^{(n)}$ is defined as $k_{max} = \max_{i \neq j} \dim(\mathcal{S}_{ij}^{(n)})$. Thus, we derive the following conditions for arbitrary linear sampling operator.

Proposition 1 *Linear sampling operator* $\Phi_n : \mathcal{U}^{*(n)} \rightarrow \mathbb{R}^{M_n}$ *is invertible for* $\mathcal{U}^{*(n)}$ *if* $M_n \geq k_{max}$*.*

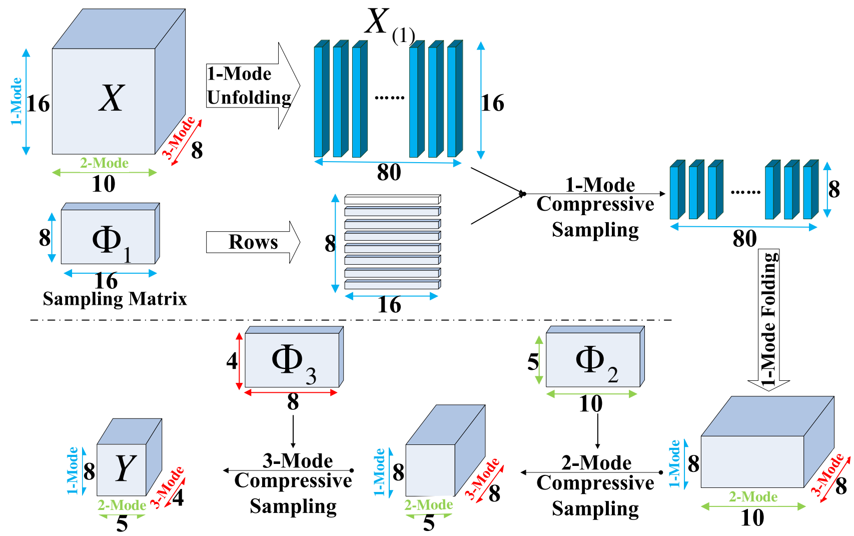


Figure 2: $1, 2, 3$-mode compressive sampling by projecting tensor $\mathbf{X} \in \mathbb{R}^{16\times10\times8}$ onto tensor $\mathcal{Y} \in \mathbb{R}^{8\times5\times4}$, where $\mathcal{Y} = \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \times_3 \Phi_3$ is obtained with sampling matricesΦ_1, Φ_2, andΦ_3.

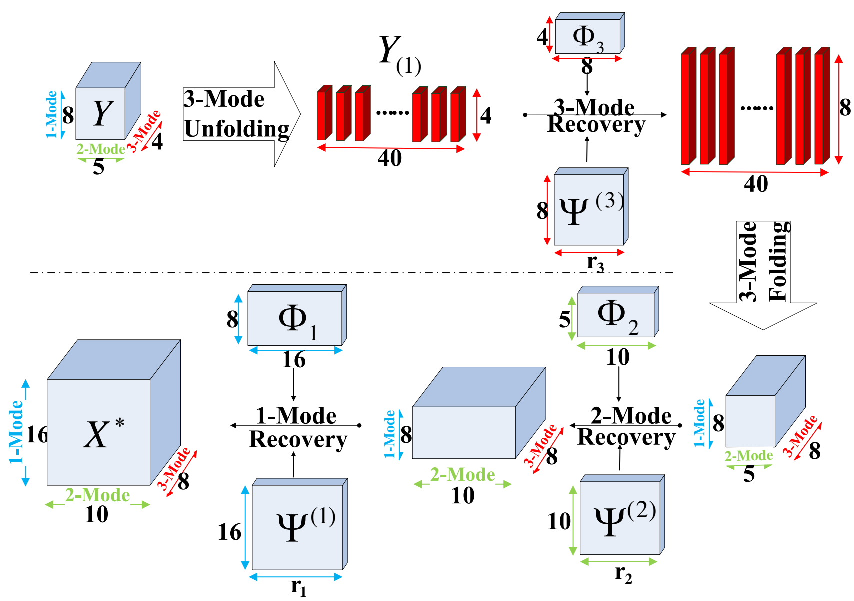


Figure 3: $1, 2, 3$ mode recovery of tensor $\mathbf{X}^* \in \mathbb{R}^{16\times10\times8}$ from tensor $\mathcal{Y} \in \mathbb{R}^{8\times5\times4}$, where $\mathcal{Y} = \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \times_3 \Phi_3$ with sampling matricesΦ_1, Φ_2, andΦ_3, and basis$\Psi^{(1)}$, $\Psi^{(2)}$, and $\Psi^{(3)}$ for the three modes, respectively.

Although $\mathcal{U}^*$ is data-driven, the proposed sampling method can still satisfy the properties similar to the UoS model. This conclusion can be proved similarly to the Proposition 3 in [10]. Proposition 1 implies that the minimum number of samples needed to guarantee a stable reconstruction for each mode.

Given the sensing matrix $A^{(n)}$ and the difference $u = \theta_1 - \theta_2$ of two k-block sparse column vectors of$\Theta_{(n)}$, Proposition 2 provides the conditions for stable recovery.

Algorithm 1 The proposed compressive tensor sampling

Input: Tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, training set $\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_t]$, random Gaussian sampling matrix$\Phi_1, \ldots, \Phi_N$.
Output: Tensor measurements $\mathcal{Y}$, recovered tensor $\mathbf{X}^*$, basis $\hat{\Psi}$.
1 Sampling: Sample $\mathbf{X}$ to derive measurements $\mathcal{Y}$ with Eq. (4).
2 Training:
  for $n = 1$ to N **do**
    for $i = 1$ to t **do**
      Apply MPCA to $\mathbf{G}_i$ and obtain the corresponding n-mode basis$\Psi_i^{(n)} \in \mathbb{R}^{I_n \times P_n}$
      Form the n-mode basis$\Psi^{(n)} \leftarrow [\Psi^{(n)}, \Psi_i^{(n)}]$
    end for
  end for
  Derive the overall basis $\hat{\Psi} = [\Psi^{(1)} \otimes \Psi^{(2)} \otimes \cdots \otimes \Psi^{(N)}]$
3 Recovering:
  for $n = 1$ to N **do**
    n-Mode Recovery: Calculate each column vector θ of$\Theta_{(n)}$ with Eq. (8)
    Calculate $X^*_{(n)} = \Psi^{(n)}\Theta_{(n)}$
  end for

Proposition 2 *The n-mode sensing matrix $A^{(n)}$ is stable for every $2k$-block sparse vector u if and only if there exists $C_1 > 0$ and $C_2 < \infty$ such that*

$$C_1\|u\|_2^2 \leq \|A^{(n)}u\|_2^2 \leq C_2\|u\|_2^2 \tag{7}$$

Proof: Eq. (6) shows that $A^{(n)} = \Phi_n\Psi^{(n)}$ for the orthonormal basis$\Psi^{(n)}$ of $\mathcal{U}^{*(n)}$ obtained by MPCA and the i.i.d. random matrixΦ_n. According to Proposition 4 and Proposition 5 in [10], we can easily obtain Proposition 2. ■

Definition 2 (n-Mode Recovery) *Provided that each column vector θ of $\Theta_{(n)}$ is k-block-sparse, θ can be recovered by solving:*

$$\min \|\theta\|_{1,2}, \quad s.t.\ y_{(n)} = \Phi_n\Psi^{(n)}\theta = A^{(n)}\theta, \tag{8}$$

Here $A^{(n)}$ satisfies the block-RIP condition with $\delta_{2k} \leq \sqrt{2} - 1$, and $y_{(n)}$ is the corresponding column vector of the n-mode unfolding matrix $Y_{(n)}$.

We can reconstruct θ by group-BP algorithm[14] for each mode. Fig. 3 provides an example for recovering a 3-order tensor. Algorithm 1 elaborates the proposed compressive tensor sampling method.

2.3 The proposed model for video sampling and recovery

Fig. 4 depicts the proposed model for video sampling and recovery with compressive tensor sampling. Given each group of pictures (GOP) in the video sequence, it is decomposed into a set of reference frames (RFs) and the remaining CS frames(CSFs). We first fully sample RF while CS patches x of CSFs are sampled with low sampling rate according to Eq.(4). Then, the recovered RF is decomposed into t data-driven

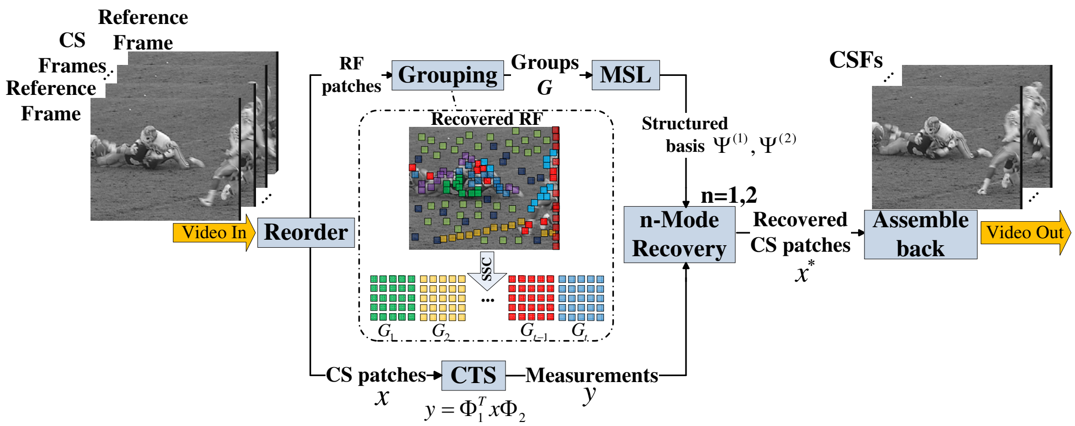


Figure 4: The proposed model for compressive tensor sampling with structured sparsity

groups $G = G_1, \ldots, G_t$ by SSC. The corresponding base$\Psi^{(1)}, \Psi^{(2)}$ can be derived by MSL. Finally, each block-sparse vector θ of the n-mode unfolding representation matrix$\Theta_{(n)}$ is derived according to Eq. (8), and utilized to recover the CS patches by $x^*_{(n)} = \Psi^{(n)}\Theta_{(n)}, n = 1, 2$.

3 Experimental Results and Discussion

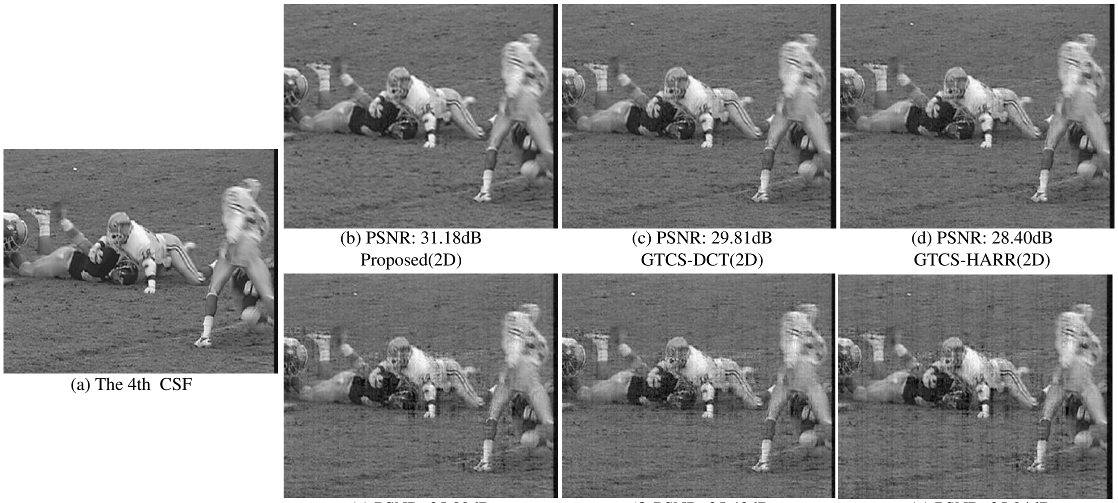
In this section, experiments are conducted on a variety of video sequences with CIF (352×288) resolution (i.e., *Akiyo, Bike, Bus, Football, Foreman, NBA*). The proposed method is compared with the state-of-the-art tensor-based method GTCS[9], which utilizes DCT basis to make video compressible in DCT domain. For 2D case, the video sequence is decomposed into 32×32 patches frame by frame. In the proposed method, we choose the first frame of each GOP and split the first Reference frame into overlapping 32×32 tensors to get the training set. The training set is partitioned into 10 groups by SSC. For each group, 1-mode and 2-mode basis are derived by MPCA with $P_1 = P_2 = 5$ and $\Psi^{(1)}, \Psi^{(2)} \in R^{32\times50}$. For GTCS, DCT and Harr basis are taken as 1-mode and 2-mode basis for validation, respectively. For both methods, we sample 32×32 non-overlapping patches from 4 consecutive frames with i.i.d. random Gaussian matrix$\Phi_1, \Phi_2 \in \mathbb{R}^{32SR\times32}$ with zero-mean and unit-variance for 1-mode and 2-mode sampling ,respectively. Therefore, the number of measurements for each patches is $SR^2 \cdot 1024$. For 3D case, each GOP in video sequence is represented by a $352 \times 288 \times 5$ tensor. Besides, 1-mode and 2-mode basis for the 2D case. DCT basis is adopted to derive the 3-mode basis in the proposed method. For both methods, we sample $32 \times 32 \times 4$ non-overlapping sub-video cubes with i.i.d. random Gaussian matrix$\Phi_1, \Phi_2 \in \mathbb{R}^{32SR\times32}$ with zero-mean and unit-variance for 1-mode and 2-mode sampling like 2D case respectively and use the 4×4 identity matrix for 3-mode full sampling. Similarly, the number of measurements for each sub-video cube is $SR^2 \cdot 16384$ for 3D case. In the experiments, we set SR from 0.4 to 0.8. For the recovery of each mode, the SPGL1 Matlab solver[14] is employed for both schemes.

Table 1: Average PSNR in dB for various video sequences obtained by the proposed method and GTCS with DCT and Harr basis with 32×32 block in both 2D and 3D cases, respectively.

Sequence	Algorithm	Sampling Rate ($M_i/I_i, i = 1, 2.$)				
		0.4	0.5	0.6	0.7	0.8
Akiyo	Proposed(2D)	**25.75**	**28.09**	**30.28**	**31.95**	**34.81**
	GTCS-DCT(2D)	21.88	24.55	26.86	28.64	32.84
	GTCS-HARR(2D)	21.40	24.34	26.80	29.00	33.48
	Proposed(3D)	23.69	25.33	26.25	29.28	32.42
	GTCS-DCT(3D)	21.62	25.42	25.94	29.26	32.32
	GTCS-HARR(3D)	22.07	24.94	26.07	29.17	32.26
Bike	Proposed(2D)	**16.96**	**19.05**	**21.00**	**23.15**	**26.92**
	GTCS-DCT(2D)	16.41	18.00	19.61	21.32	24.66
	GTCS-HARR(2D)	16.35	17.93	19.49	21.17	24.12
	Proposed(3D)	16.51	17.72	18.54	21.34	23.45
	GTCS-DCT(3D)	16.13	17.86	18.60	20.61	23.43
	GTCS-HARR(3D)	16.12	17.77	18.62	20.45	22.24
Bus	Proposed(2D)	**17.82**	**19.96**	**21.72**	**24.25**	**27.77**
	GTCS-DCT(2D)	15.82	18.44	19.79	21.85	25.07
	GTCS-HARR(2D)	16.03	18.52	19.97	22.13	25.16
	Proposed(3D)	17.00	17.72	19.88	21.41	23.35
	GTCS-DCT(3D)	16.12	17.77	19.91	21.43	22.99
	GTCS-HARR(3D)	16.26	17.95	20.15	21.75	22.57
Football	Proposed(2D)	**21.19**	**23.36**	**25.51**	**26.86**	**29.54**
	GTCS-DCT(2D)	18.12	20.53	22.37	24.14	28.16
	GTCS-HARR(2D)	17.97	20.16	21.87	23.53	27.07
	Proposed(3D)	17.63	20.24	20.47	23.51	27.28
	GTCS-DCT(3D)	17.67	20.50	21.22	23.02	24.96
	GTCS-HARR(3D)	17.83	19.69	20.85	22.72	23.99
Foreman	Proposed(2D)	**24.09**	**26.85**	**28.50**	**29.99**	**31.82**
	GTCS-DCT(2D)	19.12	20.99	23.23	26.03	30.11
	GTCS-HARR(2D)	19.10	20.89	23.26	26.05	30.28
	Proposed(3D)	19.96	21.87	22.97	26.17	28.65
	GTCS-DCT(3D)	19.47	21.52	22.43	25.50	28.42
	GTCS-HARR(3D)	18.01	20.81	22.87	25.09	28.52
NBA	Proposed(2D)	**16.42**	**17.76**	**20.13**	**22.31**	**25.41**
	GTCS-DCT(2D)	14.67	15.80	17.62	19.55	23.14
	GTCS-HARR(2D)	14.49	15.55	17.25	18.99	22.01
	Proposed(3D)	11.30	15.79	17.44	19.02	21.69
	GTCS-DCT(3D)	11.68	15.16	16.78	18.83	20.80
	GTCS-HARR(3D)	10.82	14.84	16.48	18.20	19.94

The proposed method is implemented based on the MATLAB Tensor Toolbox [18] in a workstation with 3.2-GHz CPU and 12-GB RAM.

Fig. 5 compares the visual quality of reconstructed frames obtained by the proposed method and GTCS. For both 2D and 3D cases, it shows that the proposed method can achieve better visual quality in comparison to GTCS with DCT and Harr basis. Table. 1 provides the average reconstruction performance in terms of PSNR under various sampling rates. When compared with GTCS, the proposed scheme can achieve a gain ranging from 0.5 to 5 dB in both 2D and 3D cases. This fact implies that the adaptive basis for each mode of the tensors to be sampled is more flexible and effective to capture varying nonstationary statistics in video sequences than DCT basis and Harr basis in GTCS. It should be noted that the proposed method can achieve perfect recovery with fewer necessary measurements based on the block sparsity for each tensor mode.



(a) The 4th CSF

(b) PSNR: 31.18dB Proposed(2D)

(c) PSNR: 29.81dB GTCS-DCT(2D)

(d) PSNR: 28.40dB GTCS-HARR(2D)

(e) PSNR: 25.89dB Proposed(3D)

(f) PSNR: 25.42dB GTCS-DCT(3D)

(g) PSNR: 25.04dB GTCS-HARR(3D)

Figure 5: The reconstructed frame in the sequence *Football* obtained by the proposed method and GTCS with DCT and Harr basis with 32×32 patches under SR=0.8 in both 2D and 3D cases, respectively.

4 Conclusions

This paper proposes a general and adaptive model that incorporates block sparsity into tensor representation to capture the varying statistics of nonstationary signals like images and video sequences. The proposed model inherits the merit from tensor-based CS to alleviate the computational and storage burdens in sampling and recovery. Furthermore, it can achieve desirable recovery quality for tensors using fewer degrees of freedom. To enable the block sparsity, a subspace clustering method is adopted to generate a union of tensor subspaces with their bases adaptively learned from the training set. The recovery algorithm is demonstrated to be stable with a provable performance. Experimental results show that the proposed model can achieve better performance in comparison to the state-of-the-art tensor-based sampling method. In future, we would investigate sophisticated subspace clustering methods and dictionary learning methods to improve the performance of the proposed method.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactoins on Information Theory*, vol. 52, pp. 1289-1306, April 2006.

[2] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "Compressive imaging for video representation and coding," in *the Proceedings of Picture Coding Symposium*, Beijing, China, April 2006, pp. 1-6.

[3] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Proceedings of the IEEE Data Compression Conference*, Snowbird, UT, March 2011, pp. 183-192.

[4] Ying Liu, Ming Li, and Dimitris A. Pados, "Motion-Aware Decoding of Compressed-Sensed Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 3, pp. 438-444, March 2013.

[5] Y. Li and H. Xiong, "Union of data-driven subspaces via subspace clustering for compressive video sampling," in *Proceedings of the IEEE Data Compression Conference*, Snowbird, UT, USA, March 2014, pp. 63-72.

[6] L. Lim and P. Comon, "Multiarray signal processing: Tensor decomposition meets compressed sensing," *Comptes Rendus Mecanique*, vol. 338, no. 6, pp. 311-320, June 2010.

[7] M. F. Duarte, R. G. Baraniuk, "Kronecker Compressive Sensing," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 494-504, February 2012.

[8] N. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 757-760, 2012.

[9] S. Friedland, Q. Li, D. Schonfeld, "Compressive Sensing of Sparse Tensors," *IEEE Transactions on Image Processing*, vol.23, no.10, pp. 4438-4447, October 2014.

[10] Yue M. Lu and Minh N. Do, "A Theory for Sampling Signals From a Union of Subspaces," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2334-2345, June 2008.

[11] J. Huang and T. Zhang, "The benefit of group sparsity," *Annals of Statistics*, vol. 38, no. 4, pp. 1978-2004, 2010.

[12] Yonina C. Eldar, and Moshe Mishali, "Robust Recovery of Signals From a Structured Union of Subspaces," *IEEE Transactions on Information*, vol. 55, no. 11, pp. 5302-5316, November 2009.

[13] E. Elhamifar and R. Vidal. "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765-2781, March 2013.

[14] E. van den Berg and M. P. Friedlander, "Sparse optimization with least-squares constraints," *SIAM Journal on Optimization*, vol.21, no.4, pp. 1201-1229, 2011.

[15] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1-3, pp. 167-191, November 2005.

[16] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, December 2005, pp. 499-506.

[17] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Network*, vol. 19, no. 1, pp. 18-39, January 2008.

[18] T. G. Kolda *et al.*, MATLAB Tensor Toolbox Version 2.5, January 2012. [Online]. Available: `http://www.sandia.gov/~tgkolda/TensorToolbox/`