Compressive Sensing of Sparse Tensor

Shmuel Friedland
Univ. Illinois at Chicago

Matheon Workshop CSA2013
December 12, 2013
joint work with Q. Li and D. Schonfeld, UIC

Abstract

Conventional Compressive sensing (CS) theory relies on data representation in the form of vectors.

Many data types in various applications such as color imaging, video sequences, and multi-sensor networks, are intrinsically represented by higher-order tensors.

We propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order spare tensors. Similar to Caiafa-Cichocki 2012-13

GTCS offers an efficient means for representation of multidimensional data by providing simultaneous acquisition and compression from all tensor modes.

We compare the performance of the proposed method with Kronecker compressive sensing (KCS, Duarte-Baraniuk), and multi-way compressive sensing (MWCS, Sidiropoulus-Kyrillidis). We demonstrate experimentally that GTCS outperforms KCS and MWCS in terms of both accuracy and speed.

Overview

- CS of vectors
- CS of matrices
- Simulations of CS for matrices
- CS of tensors
- Simulations of CS for tensors
- Conclusions

Compressive sensing of vectors: Noiseless

$\Sigma_{s,N}$ is the set of all $\mathbf{x} \in \mathbb{R}^N$ with at most s nonzero coordinates

Sparse version of CS: Given $\mathbf{x} \in \Sigma_{s,N}$ compress it to a short vector

$\mathbf{y} = (y_1, \ldots, y_M)^\top, M << N$ and send it to receiver

receiver gets $\mathbf{y}$, possible with noise, decodes to $\mathbf{x}$

Compressible version: coordinates of $\mathbf{x}$ have fast power law decay

Solution: $\mathbf{y} = A\mathbf{x}$, $A \in \mathbb{R}^{M \times N}$ a specially chosen matrix, e.g. s-n. p.

Sparse noiseless recovery: $\mathbf{x} = \arg\min\{\|\mathbf{z}\|_1, A\mathbf{z} = \mathbf{y}\}$

A has s-null property if for each $A\mathbf{w} = \mathbf{0}, \mathbf{w} \neq \mathbf{0}$, $\|\mathbf{w}\|_1 > 2\|\mathbf{w}_S\|_1$

$S \subset [N] := \{1, \ldots, N\}, |S| = s$,

$\mathbf{w}_S$ has zero coordinates outside S and coincides with $\mathbf{w}$ on S

Recovery condition $M \geq cs\log(N/s)$, noiseless reconstruction $O(N^3)$

Compressive sensing of vectors with noise

$A \in \mathbb{R}^{M \times N}$ satisfies restricted isometry property (RIP$_s$):

$(1-\delta_s)\|\mathbf{x}\|_2 \leq \|A\mathbf{x}\|_2 \leq (1+\delta_s)\|\mathbf{x}\|_2$

for all $\mathbf{x} \in \Sigma_{s,N}$ and for some $\delta_s \in (0,1)$

Recover with noise: $\hat{\mathbf{x}} = \arg\min\{\|\mathbf{z}\|_1, \|A\mathbf{z}-\mathbf{y}\}\|_2 \leq \varepsilon\}$

reconstruction $O(N^3)$

THM: Assume that A satisfies RIP$_{2s}$ property with $\delta_{2s} \in (0, \sqrt{2}-1)$.

Let $\mathbf{x} \in \Sigma_{s,N}, \mathbf{y} = A\mathbf{x} + \mathbf{e}, \|\mathbf{e}\|_2 \leq \varepsilon$. Then

$\|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq C_2\varepsilon$, where $C_2 = \frac{4\sqrt{1+\delta_{2s}}}{1-(1+\sqrt{2})\delta_{2s}}$

Compressive sensing of matrices I - noiseless

$X = [x_{ij}] = [\mathbf{x}_1 \dots \mathbf{x}_{N_1}]^\top \in \mathbb{R}^{N_1 \times N_2}$ is s-sparse.

$Y = U_1 X U_2^\top = [\mathbf{y}_1, \dots, \mathbf{y}_{M_2}] \in \mathbb{R}^{M_1 \times M_2}$, $U_1 \in \mathbb{R}^{M_1 \times N_1}$, $U_2 = \mathbb{R}^{M_2 \times N_2}$

$M_i \geq cs \log(N_i/s)$ and U_i has s-null property for $i = 1, 2$

Thm M: X is determined from noiseless Y.

Algo 1: $Z = [\mathbf{z}_1 \ \dots \mathbf{z}_{M_2}] = XU_2^\top \in \mathbb{R}^{N_1 \times M_2}$

each $\mathbf{z}_i$ a linear combination of columns of X hence s-sparse

$Y = U_1 Z = [U_1 \mathbf{z}_1, \dots, U_1 \mathbf{z}_{M_2}]$ so $\mathbf{y}_i = U_1 \mathbf{z}_i$ for $i \in [M_2]$

Recover each $\mathbf{z}_i$ to obtain Z

Cost: $M_2 O(N_1^3) = O((\log N_2) N_1^3)$

$Z^\top = U_2 X^\top = [U_2 \mathbf{x}_1 \dots U_2 \mathbf{x}_{N_1}]$

Recover each $\mathbf{x}_i$ from $i - th$ column of $Z^\top$

Cost: $N_1 O(N_2^3) = O(N_1 N_2^3)$, Total cost: $O(N_1 N_2^3 + (\log N_2) N_1^3)$

Compressive sensing of matrices II - noiseless

Algo 2: Decompose $Y = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\top$,

$\mathbf{u}_1, \ldots, \mathbf{u}_r$, $\mathbf{v}_1^\top, \ldots, \mathbf{v}_r^\top$ span column and row spaces of Y respectively

for example a rank decomposition of Y: $r = \operatorname{rank} Y$

Claim $\mathbf{u}_i = U_1 \mathbf{a}_i, \mathbf{v}_j = U_2 \mathbf{b}_j$, $\mathbf{a}_i, \mathbf{b}_j$ are s-sparse, $i, j \in [r]$.

Find $\mathbf{a}_i, \mathbf{b}_j$. Then $X = \sum_{i=1}^{r} \mathbf{a}_i \mathbf{b}_i^\top$

Explanation: Each vector in column and row spaces of X is s-sparse:

$Range(Y) = U_1 Range(X)$, $Range(Y^\top) = U_2 Range(X^\top)$

Cost: Rank decomposition: $O(rM_1M_2)$ using Gauss elimination or SVD

Note: $\operatorname{rank} Y \leq \operatorname{rank} X \leq s$

Reconstructions of $\mathbf{a}_i, \mathbf{b}_j$: $O(r(N_1^3 + N_2^3))$

Reconstruction of X: $O(rs^2)$

Maximal cost: $O(s \max(N_1, N_2)^3)$

Why algorithm 2 works

Claim 1: Every vector in Range X and Range $X^\top$ is s-sparse.

Claim 2: Let $X_1 = \sum_{i=1}^r \mathbf{a}_i \mathbf{b}_i^\top$. Then $X = X_1$.

Prf: Assume $0 \neq X - X_1 = \sum_{j=1}^k \mathbf{c}_j \mathbf{d}_j^\top$, $\mathbf{c}_1, \ldots, \mathbf{c}_k$ & $\mathbf{d}_1, \ldots, \mathbf{d}_k$ lin. ind.

as Range $X_1 \subset$Range X, Range $X_1^\top \subset$Range $X^\top$

$\mathbf{c}_1, \ldots, \mathbf{c}_k \in$Range X, $\mathbf{d}_1, \ldots, \mathbf{d}_k \in$Range $X^\top$

Claim: $U_1\mathbf{c}_1, \ldots U_1\mathbf{c}_k$ lin.ind..

Suppose $\mathbf{0} = \sum_{j=1}^k t_i U_1 \mathbf{c}_j = U_1 \sum_{j=1}^k t_j \mathbf{c}_j$.

As $\mathbf{c} := \sum_{j=1}^k t_j \mathbf{c}_j \in$ Range X, $\mathbf{c}$ is s-sparse.

As U_1 has null s-property $\mathbf{c} = \mathbf{0} \Rightarrow t_1 = \ldots = t_k = 0$.

$0 = Y - Y = U_1(X - X_1)U_2^\top = \sum_{j=1}^k (U_1\mathbf{c}_j)(\mathbf{d}_j^\top U_2^\top) \Rightarrow$

$U_2\mathbf{d}_1 = \ldots = U_2\mathbf{d}_k = \mathbf{0} \Rightarrow \mathbf{d}_1 = \ldots \mathbf{d}_k = \mathbf{0}$ as each $\mathbf{d}_i$ is s-sparse

So $X - X_1 = 0$ contradition

Sum.-Noiseless CS of matrices & vectors as matrices

1. Both algorithms are highly parallelizable
2. Algorithm 2 is faster by factor $s\min(N_1, N_2)$ at least
3. In many instances but not all algorithm 1 performs better.
4. Caveat: the compression is : $M_1 M_2 \geq C^2(\log N_1)(\log N_2)$.
5. Converting vector of length N to a matrix

Assuming $N_1 = N^{\alpha}, N_2 = N^{1-\alpha}$

the cost of vector compressing is $O(N^3)$

the cost of algorithm 1 is $O((\log N)N^{\frac{9}{5}})$, $\alpha = \frac{3}{5}$

the cost of algorithm 2 is $O(sN^{\frac{3}{2}})$, $\alpha = \frac{1}{2}$, $s = O(\log N)(?)$

Remark 1: The cost of computing Y from s-sparse X: $2sM_1M_2$

(Decompose X as sum of s standard rank one matrices)

Compressive sensing of matrices with noise - I

$Y = U_1 X U_2^\top + E, \quad \|E\|_F \le \varepsilon$

Algo 1: Recover each $\mathbf{z}_i$ by $\hat{\mathbf{z}}_i = \arg\min\{\|\mathbf{w}_i\|_1, \|U_1\mathbf{w}_i - \mathbf{c}_i(Y)\}\|_2 \le \varepsilon\}$

Form $\hat{Z} = [\mathbf{z}_1 \dots \mathbf{z}_{M_2}] \in \mathbb{R}^{N_1 \times m_2}$

$\|\mathbf{c}_i(\hat{Z}) - \mathbf{c}_i(XU_2^\top\|_2 \le C_2\varepsilon$ (optimistically $\frac{C_2\varepsilon}{\sqrt{M}}$)

$\|\hat{Z} - XU_2^\top\|_F \le \sqrt{M}C_2\varepsilon$ (optimistically $C_2\varepsilon$)

Obtain $\hat{X}$ by recovering each row of X:

$\hat{\mathbf{b}}_i = \arg\min\{\|\mathbf{w}_i\|_1, \|U_2\mathbf{w}_i - \mathbf{c}_i(\hat{Z}^\top)\}\|_2 \le \sqrt{M}C_2\varepsilon\}$

$\|\hat{\mathbf{b}}_i - \mathbf{c}_i(X^\top)\|_2 \le \sqrt{M}\varepsilon$ (optimistically $\|\hat{\mathbf{b}}_i - \mathbf{c}_i(X^\top)\|_2 \le \varepsilon$

Variation: Estimate s and take best rank s-approximation of Y: Y_s

Similarly, after computing $\hat{Z}$ from Y_s replace $\hat{Z}$ by Z_s.

Costlier and no estimates

Numerical simulations

We experimentally demonstrate the performance of GTCS methods on sparse and compressible images and video sequences.

Our benchmark algorithm is Duarte-Baraniuk 2010 named Kronecker compressive sensing (KCS)

Another method is multi-way compressed sensing of Sidoropoulus-Kyrillidis (MWCS) 2012

Our experiments use the ℓ_1-minimization solvers of Candes-Romberg.

We set the same threshold to determine the termination of ℓ_1-minimization in all subsequent experiments.

All simulations are executed on a desktop with 2.4 GHz Intel Core i5 CPU and 8GB RAM.

We set $M_i = K$

UIC logo

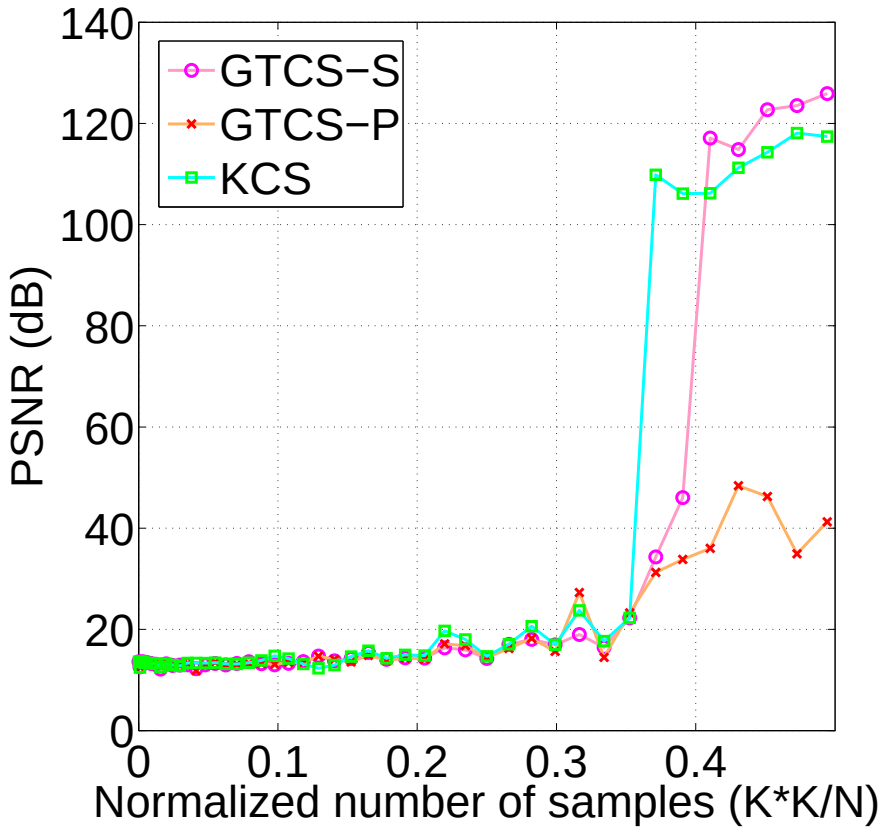
(a) The original sparse image

(b) GTCS-S recovered image

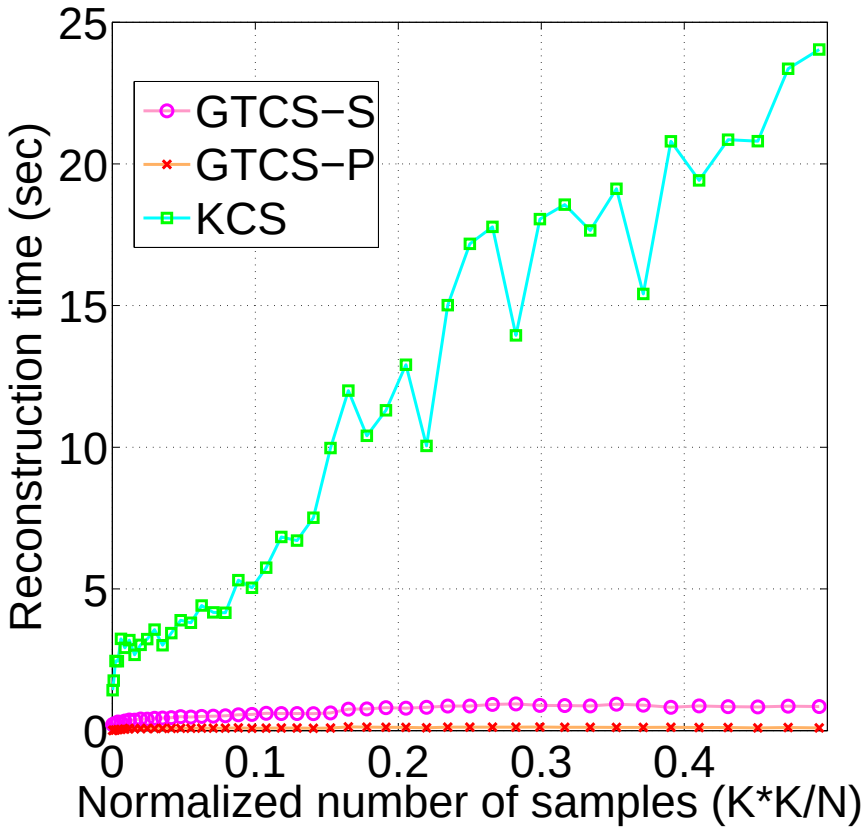
(c) GTCS-P recov-

(d) KCS recovered

PSNR and reconstruction times for UIC logo



(a) PSNR comparison



(b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on sparse image.

Explanation of UIC logo representation I

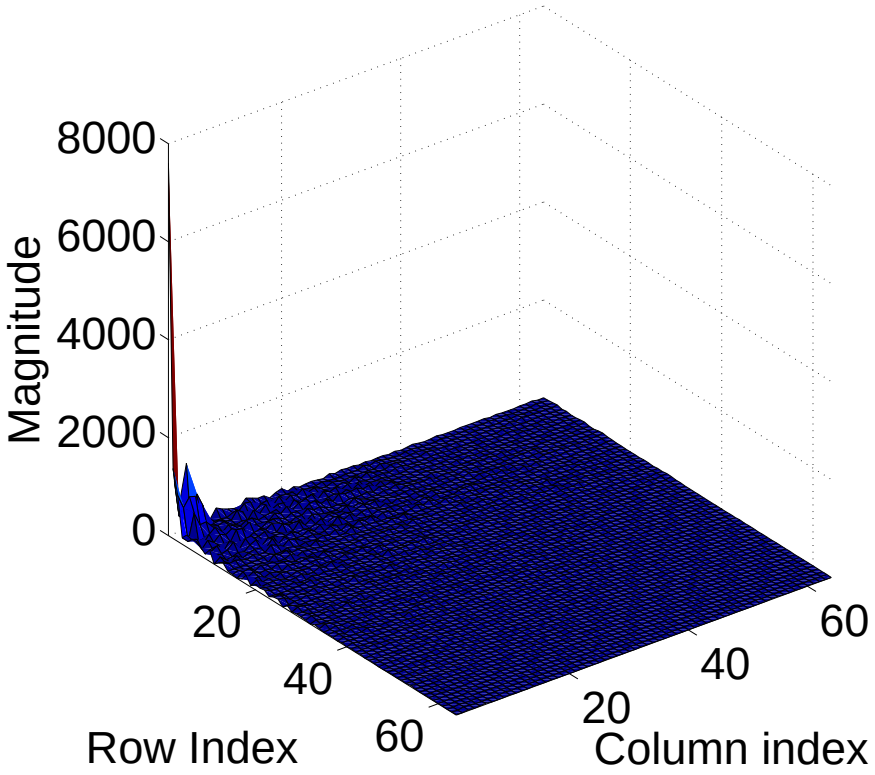
The original UIC black and white image is of size 64×64 ($N = 4096$ pixels). Its columns are 14-sparse and rows are 18-sparse. The image itself is 178-sparse. For each mode, the randomly constructed Gaussian matrix U is of size $K \times 64$. So KCS measurement matrix $U \otimes U$ is of size $K^2 \times 4096$. The total number of samples is K^2. The normalized number of samples is $\frac{K^2}{N}$. In the matrix case, GTCS-P coincides with MWCS and we simply conduct SVD on the compressed image in the decomposition stage of GTCS-P. We comprehensively examine the performance of all the above methods by varying K from 1 to 45.

Explanation of UIC logo representation II

Figure **??** and **??** compare the peak signal to noise ratio (PSNR) and the recovery time respectively. Both KCS and GTCS methods achieve PSNR over 30dB when $K = 39$. As K increases, GTCS-S tends to outperform KCS in terms of both accuracy and efficiency. Although PSNR of GTCS-P is the lowest among the three methods, it is most time efficient. Moreover, with parallelization of GTCS-P, the recovery procedure can be further accelerated considerably. The reconstructed images when $K = 38$, that is, using 0.35 normalized number of samples, are shown in Figure **??????**. Though GTCS-P usually recovers much noisier image, it is good at recovering the non-zero structure of the original image.

Cameraman simulations I

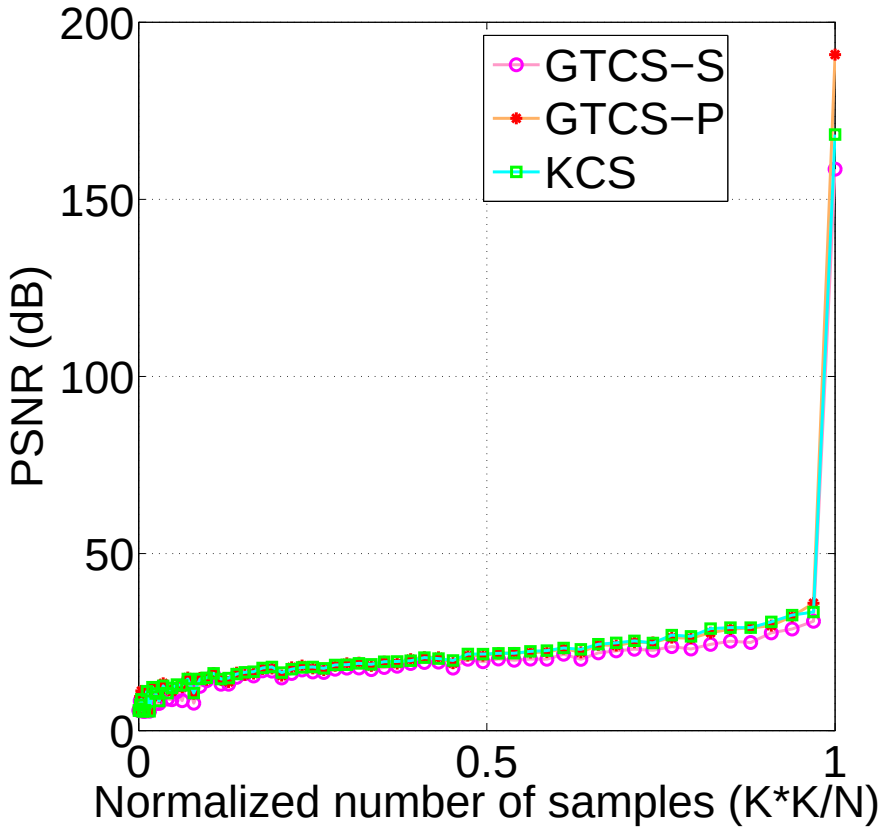
(a) Cameraman in space domain



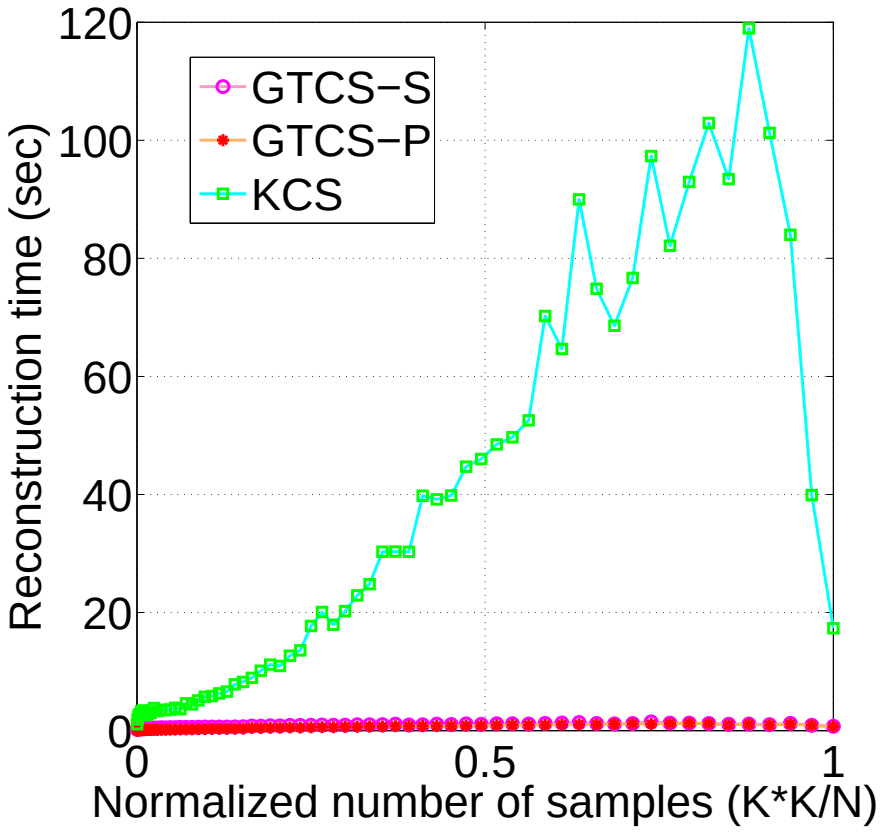
(b) Cameraman in DCT domain

Figure : The original cameraman image (resized to 64 $\times$ 64 pixels) in space domain and DCT domain.

Cameraman simulations II



(a) PSNR comparison

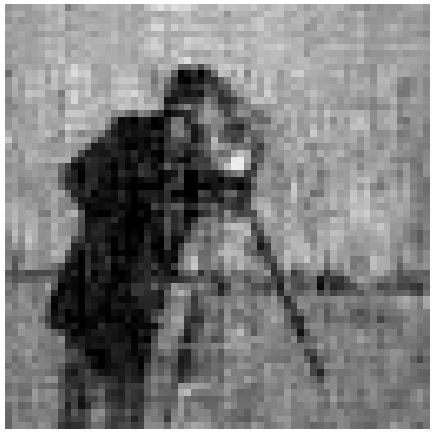


(b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on compressible image.

Cameraman simulations III

(a) GTCS-S, K = 46, PSNR = 20.21 dB



(b) GTCS-P/MWCS, K = 46, PSNR = 21.84 dB

(c) KCS, K = 46, PSNR = 21.79 dB

(d) GTCS-S,K = 63, PSNR = 30.88 dB

(e) GTCS-P/MWCS, K = 63, PSNR = 35.95 dB

(f) KCS, K = 63, PSNR = 33.46 dB

Cameraman explanations

As shown in Figure **??**, the cameraman image is resized to 64×64 ($N = 4096$ pixels). The image itself is non-sparse. However, in some transformed domain, such as discrete cosine transformation (DCT) domain in this case, the magnitudes of the coefficients decay by power law in both directions (see Figure **??**), thus are compressible. We let the number of measurements evenly split among the two modes. Again, in matrix data case, MWCS concurs with GTCS-P. We exhaustively vary K from 1 to 64.

Figure **??** and **??** compare the PSNR and the recovery time respectively. Unlike the sparse image case, GTCS-P shows outstanding performance in comparison with all other methods, in terms of both accuracy and speed, followed by KCS and then GTCS-S. The reconstructed images when $K = 46$, using 0.51 normalized number of samples and when $K = 63$, using 0.96 normalized number of samples are shown in Figure **??**.

Compressive sensing of tensors

$\mathbf{M} = (M_1, \ldots, M_d), \mathbf{N} = (N_1, \ldots, N_d) \in \mathbb{N}^d$, $J = \{j_1, \ldots, j_k\} \subset [d]$

Tensors: $\otimes_{i=1}^d \mathbb{R}^{N_i} = \mathbb{R}^{N_1 \times \ldots \times N_d} = \mathbb{R}^{\mathbf{N}}$

Contraction of $\mathcal{A} = [a_{i_{j_1}, \ldots, i_{j_k}}] \in \otimes_{j_p \in J} \mathbb{R}^{N_{j_p}}$ with $\mathcal{T} = [t_{i_1, \ldots, i_d}] \in \mathbb{R}^{\mathbf{N}}$:

$\mathcal{A} \times \mathcal{T} = \sum_{i_{j_p} \in [N_{j_p}], j_p \in J} a_{i_{j_1}, \ldots, i_{j_k}} t_{i_1, \ldots, i_d} \in \otimes_{l \in [d] \setminus J} \mathbb{R}^{N_l}$

$\mathcal{X} = [x_{i_1, \ldots, i_d}] \in \mathbb{R}^{\mathbf{N}}, \mathcal{U} = U_1 \otimes U_2 \otimes \ldots \otimes U_d \in \mathbb{R}^{(M_1, N_1, M_2, N_2, \ldots, M_d, N_d)}$

$U_p = [u^{(p)}_{i_p j_p}] \in \mathbb{R}^{M_p \times N_p}$, $p \in [d]$, $\mathcal{U}$ Kronecker product of $U_1, \ldots, U_d$.

$\mathcal{Y} = [y_{i_1, \ldots, i_d}] = \mathcal{X} \times \mathcal{U} := \mathcal{X} \times_1 U_1 \times_2 U_2 \times \ldots \times_d U_d \in \mathbb{R}^{\mathbf{M}}$

$y_{i_1, \ldots, i_p} = \sum_{j_q \in [N_q], q \in [d]} x_{j_1, \ldots, j_d} \prod_{q \in [d]} u_{i_q, j_q}$

Thm $\mathcal{X}$ is s-sparse, each U_i has s-null property

then $\mathcal{X}$ uniquely recovered from $\mathcal{Y}$.

Algo 1: GTCS-S

Algo 2: GTCS-P

Algo 1- GTCS-S

Unfold $\mathcal{Y}$ in mode 1: $Y_{(1)} = U_1 \mathcal{W}_1 \in \mathbb{R}^{M_1 \times (M_2 \cdot \ldots \cdot M_d)}$,

$\mathcal{W}_1 := X_{(1)}[\otimes_{k=d}^{2} U_k]^\top \in \mathbb{R}^{N_1 \times (M_2 \cdot \ldots \cdot M_d)}$

As for matrices recover the $\tilde{M}_2 := M_2 \cdots M_d$ columns of $\mathcal{W}_1$ using U_1

Complexity: $O(\tilde{M}_2 N_1^3)$.

Now we need to recover

$\mathcal{Y}_1 := \mathcal{X} \times_1 I_1 \times_2 U_2 \times \ldots \times_d U_d \in \mathbb{R} N_1 \times M_2 \ldots \times M_d$

Equivalently, recover N_1, $d-1$ mode tensors in $\mathbb{R}^{N_2 \times \ldots \times N_d}$ from $\mathbb{R}^{M_2 \times \ldots \times M_d}$ using $d-1$ matrices $U_2, \ldots, U_d$.

Complexity $\sum_{i=1}^{d} \tilde{N}_{i-1} \tilde{M}_{i+1} N_i^3$

$\tilde{N}_0 = \tilde{M}_{d+1} = 1, \quad \tilde{N}_i = N_1 \ldots N_i, \quad \tilde{M}_i = M_i \ldots M_d$

$d = 3$: $M_2 M_3 N_1^3 + N_1 M_3 N_2^3 + N_1 N_2 N_3^3$

Algo 2- GTCS-P

Unfold $\mathcal{X}$ in mode k: $X_{(k)} \in \mathbb{R}^{N_k \times \frac{N}{N_k}}$, $N = \prod_{i=1}^d N_i$.

As $\mathcal{X}$ is s-sparse $\operatorname{rank}_k \mathcal{X} := \operatorname{rank} X_{(k)} \leq s$.

$Y_{(k)} = U_k X_{(k)} [\otimes_{i \neq k} U_i]^\top \Rightarrow$ Range $Y_{(k)} \subset U_k$Range $X_{(k)}$, $\operatorname{rank} Y_{(k)} \leq s$.

$X_{(1)} = \sum_{j=1}^{R_1} \mathbf{u}_i \mathbf{v}_i^\top$, $\mathbf{u}_1, \ldots, \mathbf{u}_{R_1}$ spans range of $X_{(1)}$ so $R_1 \leq s$

Each $\mathbf{v}_i$ corresponds to $\mathcal{U}_i \in \mathbb{R}^{N_2 \times \ldots N_d}$ which is s-sparse

So (1) $\mathcal{X} = \sum_{j=1}^R \mathbf{u}_{1,j} \otimes \ldots \otimes \mathbf{u}_{d,j}$, $R \leq s^{d-1}$

$\mathbf{u}_{k,1}, \ldots, \mathbf{u}_{k,R} \in \mathbb{R}^{N_k}$ span Range $X_{(k)}$ and each is s-sparse

Compute decomposition $\mathcal{Y} = \sum_{j=1}^R \mathbf{w}_{1,j} \otimes \ldots \otimes \mathbf{w}_{d,j}$, $R \leq s^{d-1}$,

$\mathbf{w}_{k,1}, \ldots, \mathbf{w}_{k,R} \in \mathbb{R}^{M_k}$ span Range $Y_{(k)}$, Compl: $O(s^{d-1} \prod_{i=1}^d M_i)$

Find $\mathbf{u}_{k,j}$ from $\mathbf{w}_{k,j} = U_k \mathbf{u}_{k,j}$ and reconstruct $\mathcal{X}$ from (1)

Complexity $O(ds^{d-1} \max(N_1, \ldots, N_d)^3)$, $s = O(\log(\max(N_1, \ldots, N_d)))$

Summary of complexity converting linear data

$N_i = N^{\alpha_i}, M_i = O(\log N), \alpha_i > 0, \sum_{i=1}^{d} \alpha_i = 1, s = \log N$

$d = 3$

GTCS-S: $O((\log N)^2 N^{\frac{27}{19}})$

GTCS-P: $O((\log N)^2 N)$

GTCS-P: $O((\log N)^{d-1} N^{\frac{3}{d}})$ for any d.

Warning: the roundoff error in computing parfac decomposition of $\mathcal{Y}$ and then of $\mathcal{X}$ increases significantly with d.

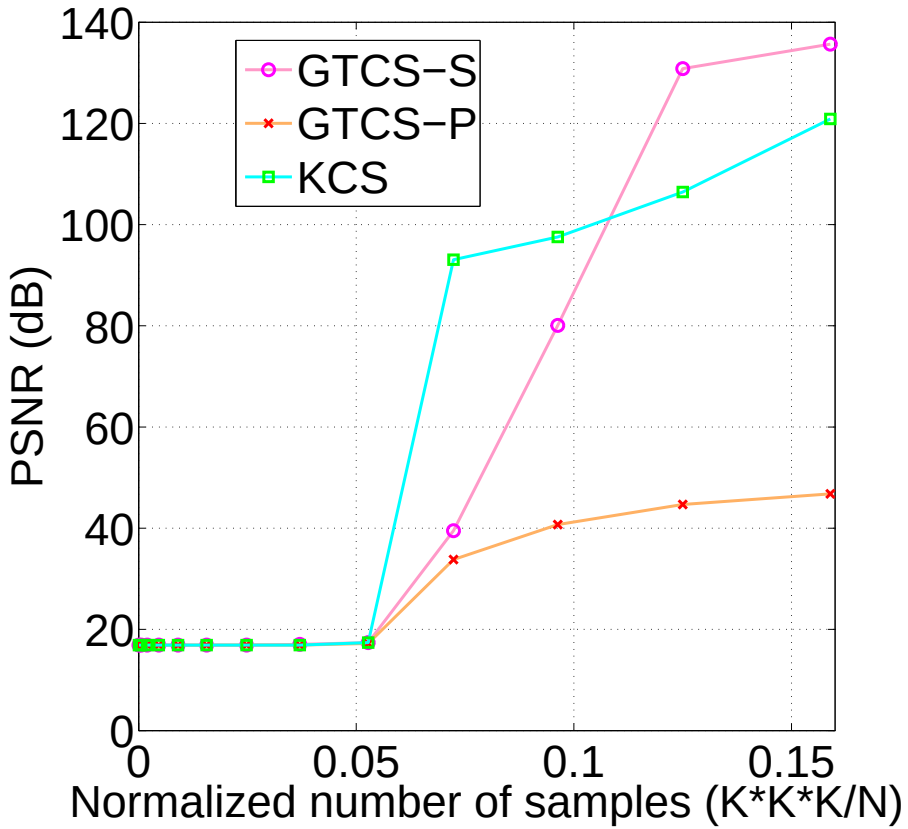
Sparse video representation

We compare the performance of GTCS and KCS on video data. Each frame of the video sequence is preprocessed to have size 24×24 and we choose the first 24 frames. The video data together is represented by a $24 \times 24 \times 24$ tensor and has $N = 13824$ voxels in total. To obtain a sparse tensor, we manually keep only $6 \times 6 \times 6$ nonzero entries in the center of the video tensor data and the rest are set to zero.

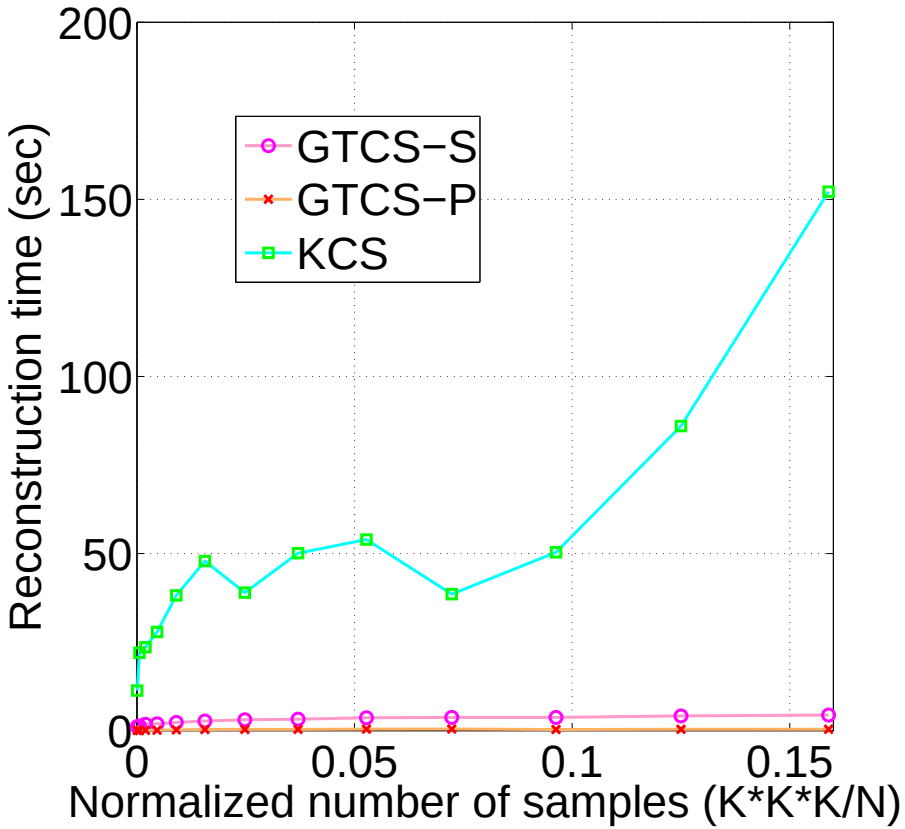
The video tensor is 216-sparse and its mode-i fibers are all 6-sparse $i = 1, 2, 3$. The randomly constructed Gaussian measurement matrix for each mode is now of size $K \times 24$ and the total number of samples is K^3. The normalized number of samples is $\frac{K^3}{N}$.

We vary K from 1 to 13.

PSNR and reconstruction time of sparse video



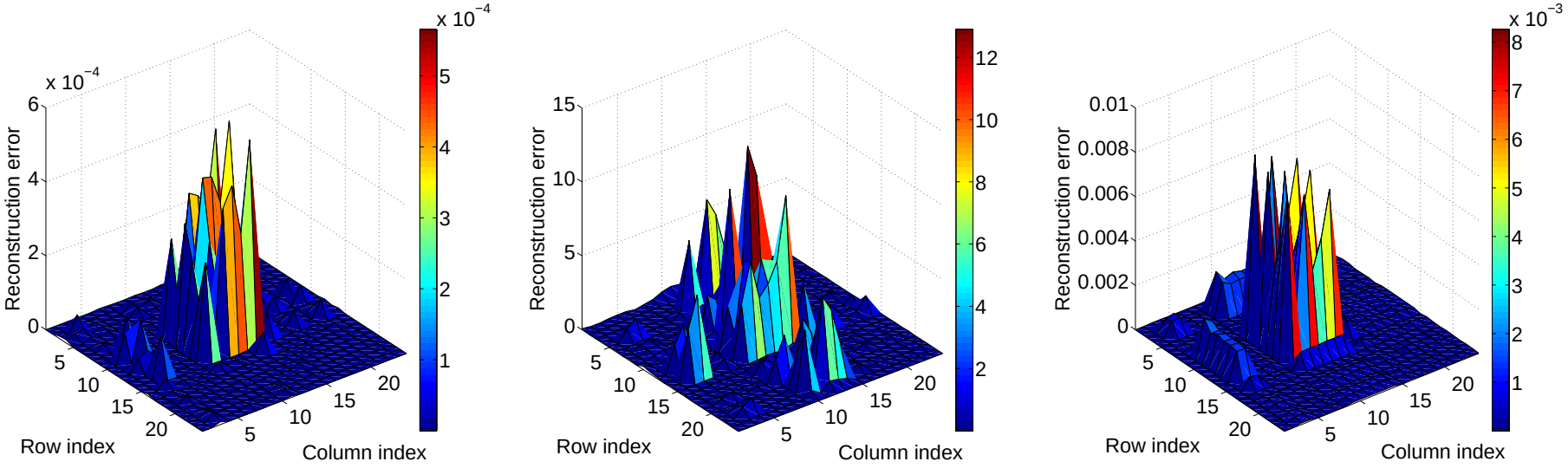
(a) PSNR comparison



(b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on sparse video.

Reconstruction errors of sparse video



(a) Reconstruction error of GTCS-S

(b) Reconstruction error of GTCS-P

(c) Reconstruction error of KCS

Figure : Visualization of the reconstruction error in the recovered video frame 9 by GTCS-S (PSNR = 130.83 dB), GTCS-P (PSNR = 44.69 dB) and KCS (PSNR = 106.43 dB) when $K = 12$, using 0.125 normalized number of samples.

Conclusion

Real-world signals as color imaging, video sequences and multi-sensor networks, are generated by the interaction of multiple factors or multimedia and can be represented by higher-order tensors. We propose Generalized Tensor Compressive Sensing (GTCS)-a unified framework for compressive sensing of sparse higher-order tensors. We give two reconstruction procedures, a serial method (GTCS-S) and a parallelizable method (GTCS-P). We compare the performance of GTCS with KCS and MWCS experimentally on various types of data including sparse image, compressible image, sparse video and compressible video. Experimental results show that GTCS outperforms KCS and MWCS in terms of both accuracy and efficiency. Compared to KCS, our recovery problems are in terms of each tensor mode, which is much smaller comparing with the vectorization of all tensor modes. Unlike MWCS, GTCS manages to get rid of tensor rank estimation, which considerably reduces the computational complexity and at the same time improves the reconstruction accuracy.

References 1

- C. Caiafa and A. Cichocki, Multidimensional compressed sensing and their applications, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(6), 355-380, (2013).
- E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete information, *Information Theory, IEEE Transactions on* 52 (2006), 489–509.
- E. Candes and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies, *Information Theory, IEEE Transactions on* 52 (2006), 5406–5425.
- D. Donoho, Compressed sensing, *Information Theory, IEEE Transactions on* 52 (2006), 1289–1306.
- M. Duarte and R. Baraniuk, Kronecker compressive sensing, *Image Processing*, IEEE Transactions on, 2 (2012), 494–504.

References 2

- Q. Li, D. Schonfeld and S. Friedland, Generalized tensor compressive sensing, *Multimedia and Expo (ICME)*, 2013 IEEE International Conference on, 15-19 July 2013, ISSN: 1945-7871, 6 pages.
- S. Friedland, Q. Li and D. Schonfeld, Compressive Sensing of Sparse Tensors, arXiv:1305.5777.
- N. Sidiropoulus and A. Kyrillidis, Multi-way compressive sensing for sparse low-rank tensors, *Signal Processing Letters*, IEEE, 19 (2012), 757–760.