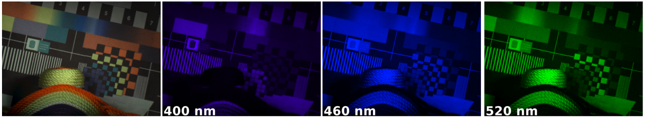
Tensor Sparse Representation Learning for Single-Snapshot Compressive Spectral Video Reconstruction*

Kareth LEÓN
Postdoctoral Researcher
INP-IRIT/ENSEEIHT, Université de Toulouse

TéSA Seminar
Toulouse, France

May, 2022

`kareth.leon@irit.fr`

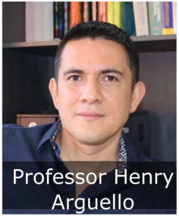


*Research Paper of the Talk

Online Tensor Sparsifying Transform Based on Temporal Superpixels From Compressive Spectral Video Measurements.

Authors: K. León and H. Arguello.

IEEE Transactions on Image Processing. (2020)
Vol. 29, pp. 5953-5963, DOI: 10.1109/TIP.2020.2985871.



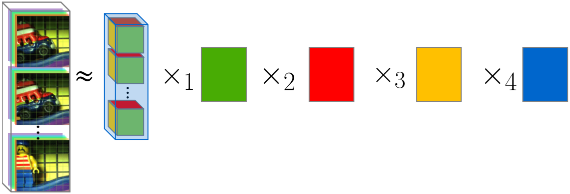
Universidad Industrial de Santander

High Dimensional Signal Processing Research Group

Outline of Talk

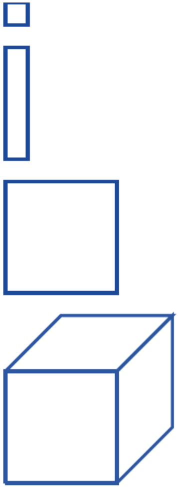
1. Motivation
2. Matrix Decomposition: SVD
3. Tensor Decomposition
4. Application of Tensors for Spectral videos Sparsifying
6. Results

$$\underline{\mathbf{F}} \approx \underline{\mathbf{\Theta}} \times_1 \mathbf{D}^{(1)} \times_2 \mathbf{D}^{(2)} \times_3 \mathbf{D}^{(3)} \times_4 \mathbf{D}^{(4)}$$



A **tensor** is a multidimensional array (or multi-way array, $N-$way array).

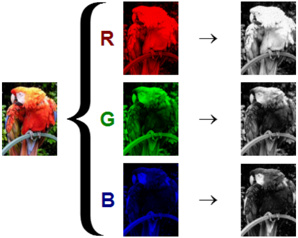
$d = 0$	scalar	$a =$
$d = 1$	vector	$\mathbf{a} =$
$d = 2$	matrix	$A =$
$d \geq 3$	tensor	$\mathcal{A} =$

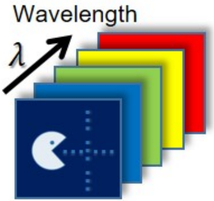


The "order" of a tensor is the *number of dimensions* d.

Motivation

- Much real-world data is inherently **multidimensional**:
  - Colour (RGB) and hyperspectral images are 3-order tensors



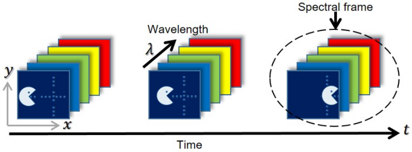


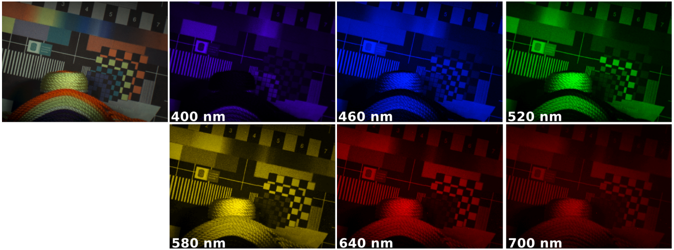
  - Color depth images[1] are 3-order tensors

[1] https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Motivation

- ◦ Hyperspectral videos are 4-order arrays





- ◦ Medical images, times series, light fields, etc.

- Tensors serve to compress or constrain data in the multiples dimensions.
- Matrix-based methods rely on the data vectorization, where the higher-order structure is lost!

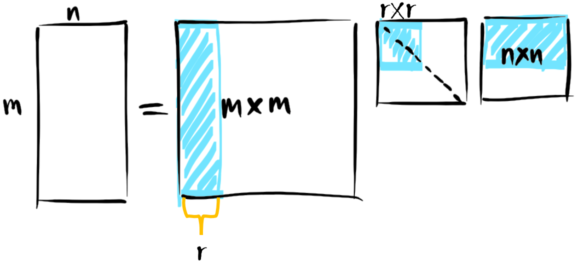
Singular Value Decomposition (for Matrices)

The SVD of $\mathbf{X} \in \mathbb{R}^{m \times n}$ is given by:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{1}$$

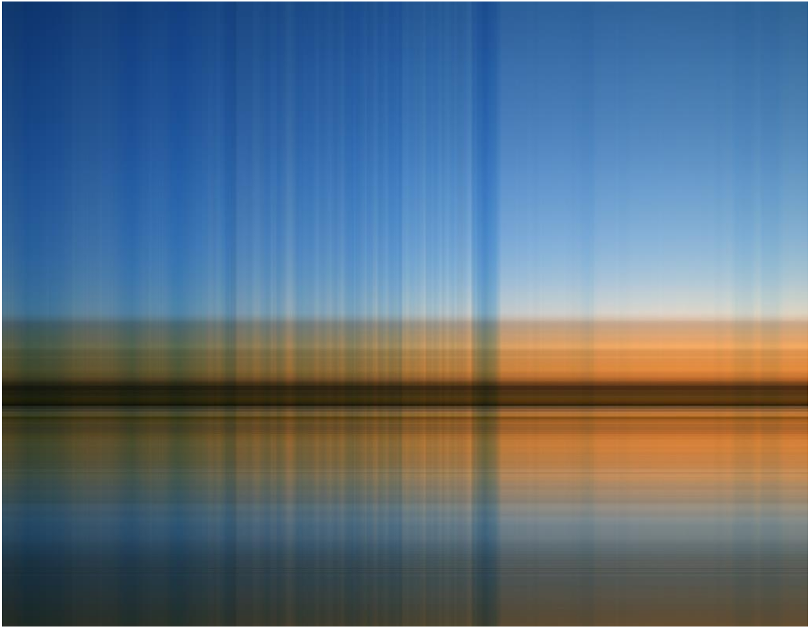
- where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonals matrices
- $\mathbf{S} \in \mathbb{R}^{m \times n}$ diagonal matrix whose elements are the *singular values* with decreasing order.

SVD Truncation[2]: $\mathbf{X}_r = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^T$;
r (rank): maximum number of linearly independent vectors in the matrix[3]

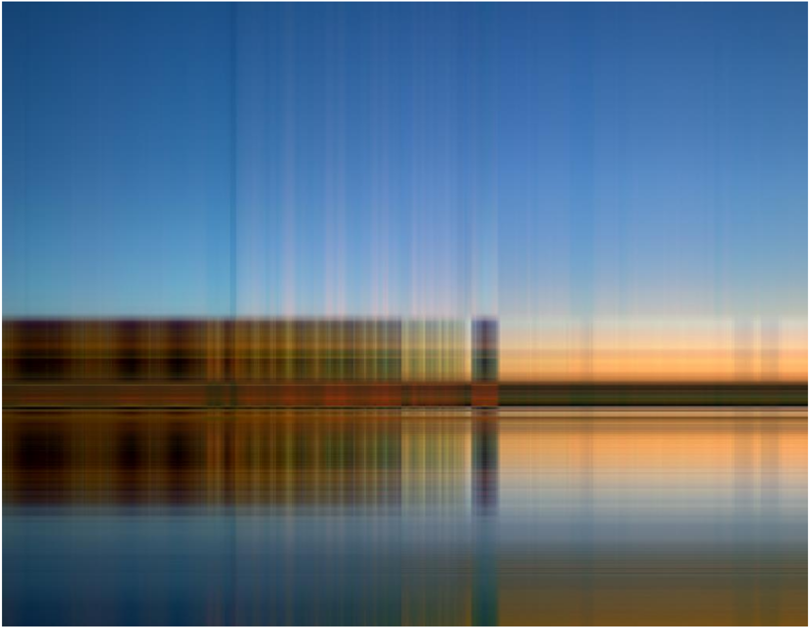


[2] `https://csiu.github.io/blog/update/2017/04/16/day51.html`

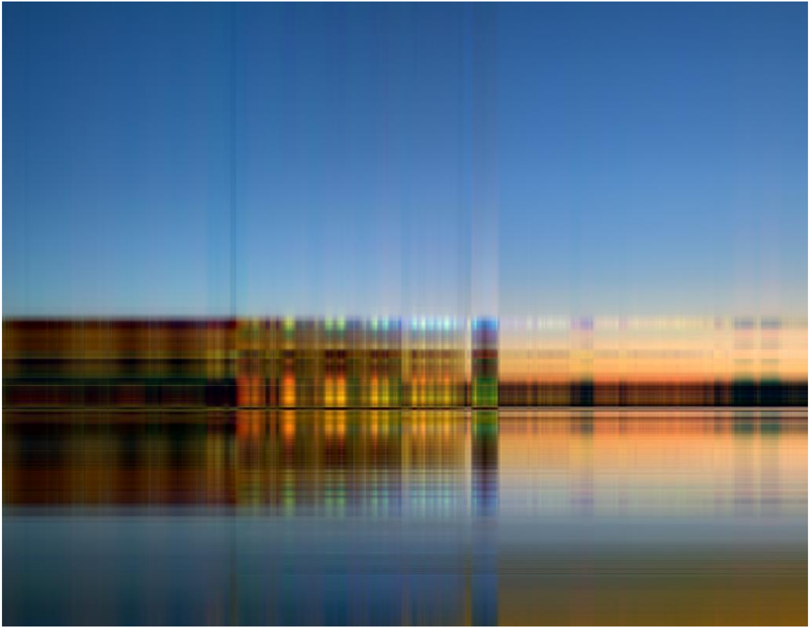
[3] When vectors are linearly independent and span a whole space we say they are a 'basis' of that space.



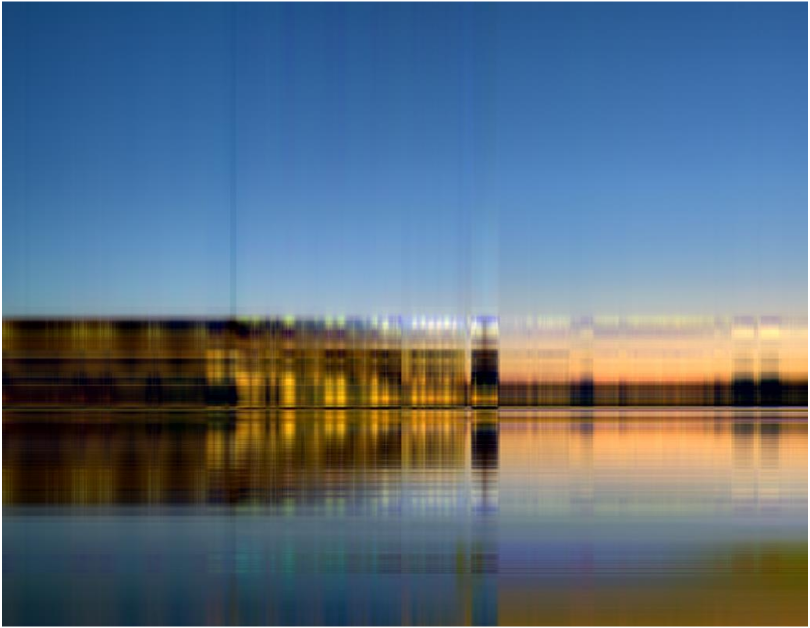
$r = 1$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$



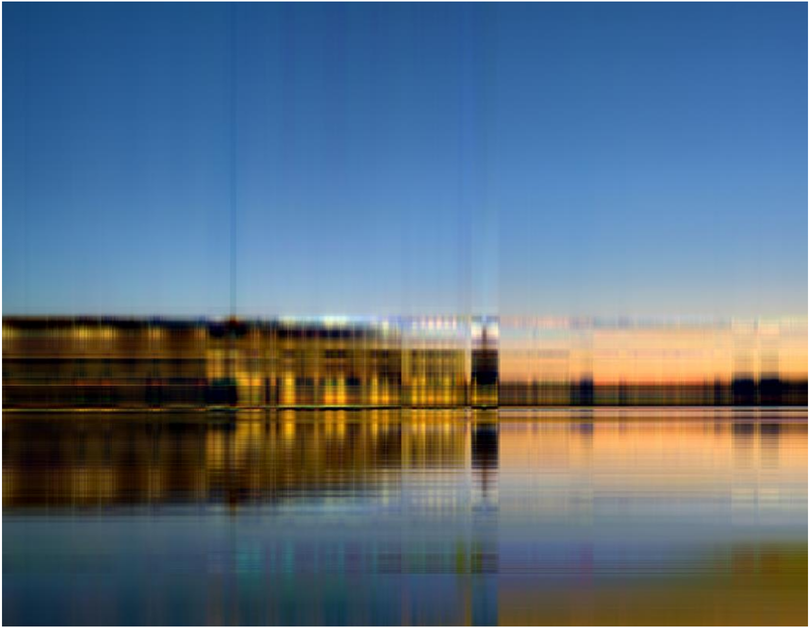
$r = 2$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$



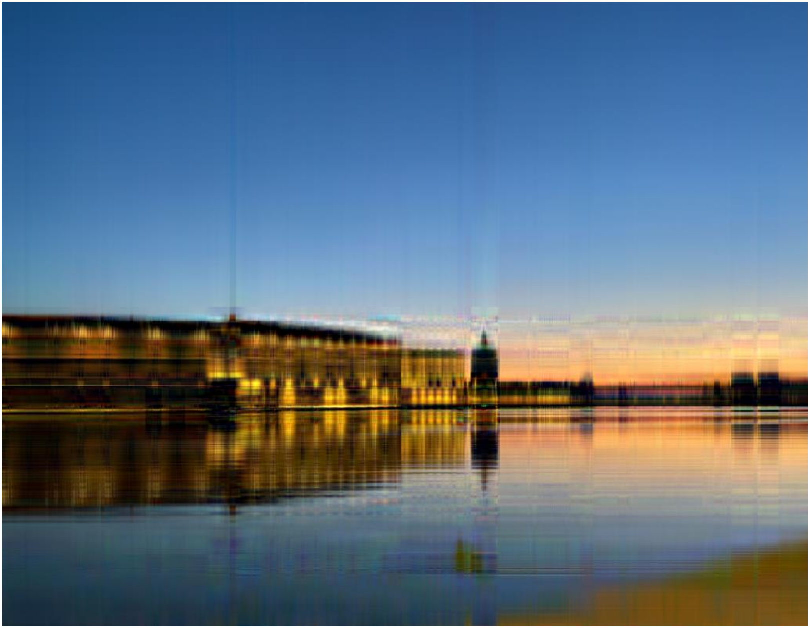
$r = 3$ in $\mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^T$



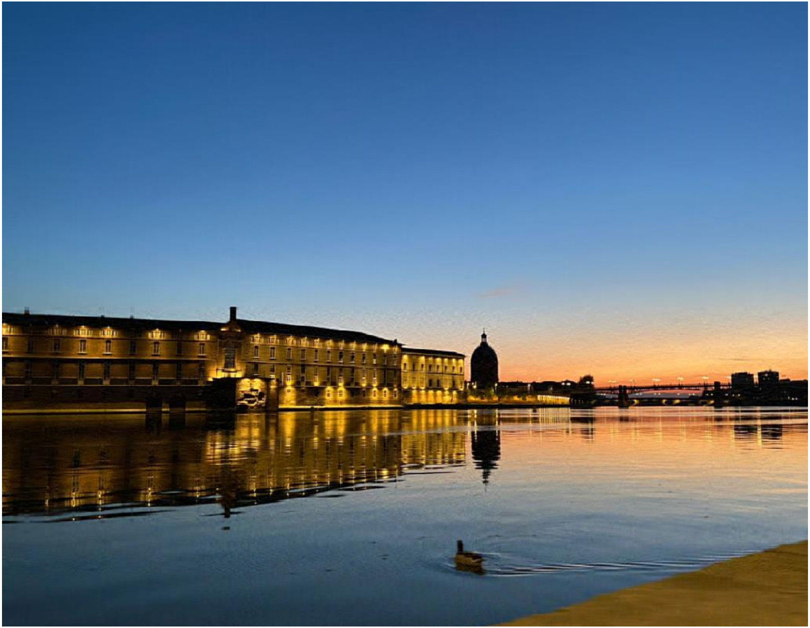
$r = 4$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$



$r = 5$ in $\mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^T$



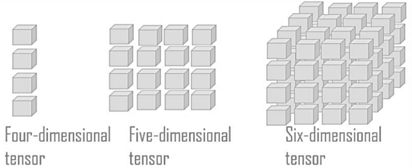
$r = 10$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$



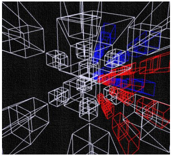
$r = 100$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$

$r = 200$ in $\mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T$

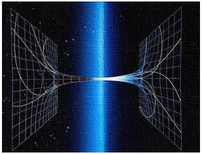
Is there a natural analogy of SVD for higher-order arrays $(d \geq 3)$?



...

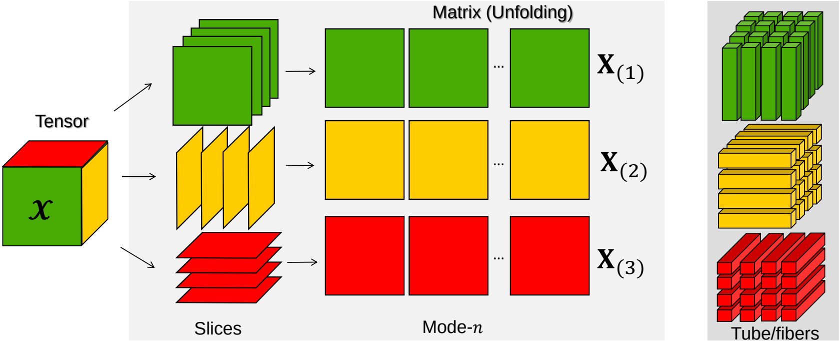


...



Basic Notation - Multilinear Algebra

Mode-n matrix representation / Sub-arrays



Mode-n product: product between a tensor and a matrix.

$$\mathcal{Z} = \mathcal{X} \times_n \mathbf{A} \Leftrightarrow \mathbf{Z}_{(n)} = \mathbf{A}\mathbf{X}_{(n)} \tag{2}$$

Outer product

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \ldots \circ \mathbf{a}^{(N)} \tag{3}$$

Ex: $\mathbf{X} = \mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^T$

Tensors Rank Decomposition

Canonical Polyadic Decomposition-CANDECOMP/Parallel Factors-PARAFAC[4]

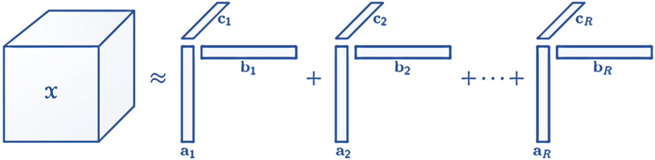
▷ Factorize a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ into a SUM of a finite number of rank-one tensors as:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(1)} \circ \ldots \mathbf{a}_r^{(N)}$$

Ex: Suppose a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the CP is written as:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{4}$$

where $R \in \mathbb{N}$, $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$, with $r = 1, \ldots, R$.



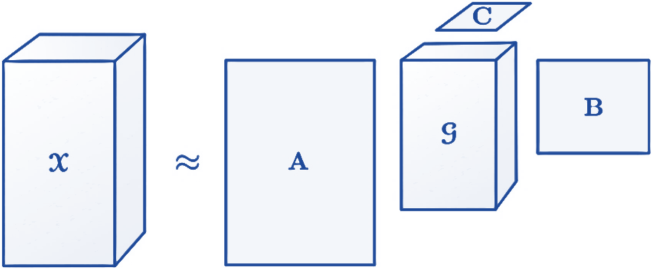
[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM Rev.m 2009.

Tucker Decomposition (TD)

▷ TD is multilinear transformation of a core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \ldots \times R_N}$ by a set of factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $n = 1, \ldots, N$ as

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A}^{(1)} \ldots \times_N \mathbf{A}^{(N)} \quad (5)$$

Ex: for the 3-order tensor: $\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$



▷ TD is a form of higher-order SVD.

Higher-Order SVD (HOSVD)

The **HOSVD**[5] of a given 3-order tensor $\mathcal{F}$ can be written as:

$$\mathcal{F} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \quad (6)$$

where $\mathcal{S} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_{\min(I_1,I_2,I_3)})$, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{\min(I_1,I_2,I_3)} \geq 0$.

A suitable tensor-decomposition-based sparsifying transform $\mathcal{U}$ can be constructed by using the unitary matrices as

$$\mathcal{U}(\mathcal{F}) = \mathcal{F} \times_1 {\mathbf{U}^{(1)}}^T \times_2 {\mathbf{U}^{(2)}}^T \times_3 {\mathbf{U}^{(3)}}^T, \quad (7)$$

where
- $\mathcal{U}(\cdot)$ **induces sparsity on the signal** and $\mathcal{S} \leftarrow \mathcal{U}(\mathcal{F})$
- $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, with $n = 1, 2, 3$ is found as the R_n **left singular vectors** on the n-mode of $\mathcal{F}$.

[5] L. De Lathauwer, et al, 'A multilinear singular value decomposition,' SIAM journal on Matrix Analysis and Applications, 2000

Higher-Order SVD (HOSVD)

The **HOSVD**[5] of a given 3-order tensor $\mathcal{F}$ can be written as:

$$\mathcal{F} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \tag{6}$$

where $\mathcal{S} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_{\min(I_1,I_2,I_3)})$, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{\min(I_1,I_2,I_3)} \geq 0$.

A suitable tensor-decomposition-based sparsifying transform $\mathcal{U}$ can be constructed by using the unitary matrices as

$$\mathcal{U}(\mathcal{F}) = \mathcal{F} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times_3 \mathbf{U}^{(3)^T}, \tag{7}$$

where

- $\mathcal{U}(\cdot)$ **induces sparsity on the signal** and $\mathcal{S} \leftarrow \mathcal{U}(\mathcal{F})$
- $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, with $n = 1, 2, 3$ is found as the R_n **left singular vectors** on the n-mode of $\mathcal{F}$.

Remark: A HOSVD transformation is useful in a *compressive sensing based scenario* due to that the coefficients decay is faster and lead sparsest solutions!

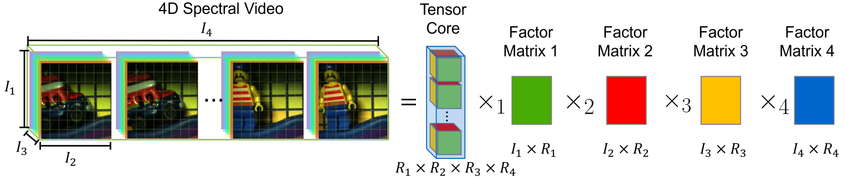
[5] L. De Lathauwer, et al, 'A multilinear singular value decomposition,' SIAM journal on Matrix Analysis and Applications, 2000

Application

Task: Exploit tensors for sparse transform compressive learning!

Multilinear transformation

A fourth-order (4D) tensor spectral video $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ can be decomposed as:



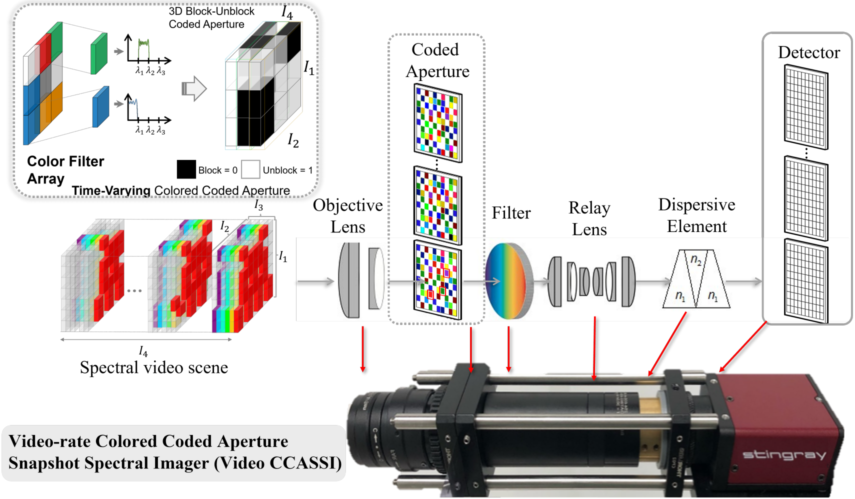
$$\mathcal{F} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)}$$

- $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times R_4}$ is the core tensor.
- $\{\mathbf{U}^{(n)}\}_{n=1}^{4} \in \mathbb{R}^{I_n \times R_n}$ is a dictionary basis for each n-mode.
- $\times_n$ is the mode-n product[6].

[6] Example: mode-3 of $\mathcal{F}$ is $\mathbf{F}_{(3)} = \mathbf{U}^{(3)} \mathbf{B}_{(3)} (\mathbf{U}^{(4)} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T$

Compressed Measurements

Single-Shot Compressive Spectral Video Sensing (CSVS)



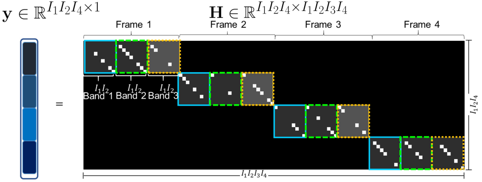
Given a spectral video $\mathcal{F}$, the sensing process is written as:

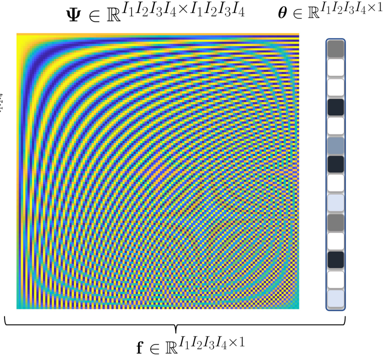
$$\mathcal{Y}_{i_1,i_2,i_4} = \sum_{i_3=1}^{I_3} \mathcal{F}_{i_1,(i_2-i_3),i_3,i_4} \circ \mathcal{T}_{i_1,(i_2-i_3),i_3,i_4} + \mathcal{W}_{i_1,i_2,i_4}. \tag{8}$$

Traditional Matrix-based Formulation

Spatial-spectral coded compressive spectral imager extended to video acquisition

Given a spectral video $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$, then, $n = I_1 I_2 I_3 I_4$ and $m = I_1 I_2 I_4$. In matrix form:





Commonly used transform bases:

- Wavelet, Discrete Cosine
- Kronecker product
  $\mathbf{\Psi} = \mathbf{\Psi}_1 \otimes \mathbf{\Psi}_2 \otimes \mathbf{\Psi}_3 \otimes \mathbf{\Psi}_4$
- Dictionary Learning

Acquisition and Recovery Problem

Tensor-based Model for the 3D-CASSI

- The CSVS acquisition procedure can be then expressed as

$$\mathcal{Y} = \mathcal{H}(\mathcal{F}) + \mathcal{W}, \tag{9}$$

where $\mathcal{H}(\mathcal{F}) : \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4} \rightarrow \mathbb{R}^{I_1 \times J_2 \times I_4}$ represents the CSVS operator and establishes the modulation and compression of the incoming signal.

- **Recovery Problem:** For a fixed basis $\{\mathbf{\Psi}^{(z)}\}_{z=1}^{4}$,

$$\begin{aligned} &\underset{\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}}{\text{minimize}} \left\| \mathcal{Y} - \mathcal{H}\big(\mathcal{G} \times_1 \mathbf{\Psi}^{(1)} \times_2 \mathbf{\Psi}^{(2)} \times_3 \mathbf{\Psi}^{(3)} \times_4 \mathbf{\Psi}^{(4)}\big) \right\|_F^2 \\ &\text{subject to } ||\text{vec}(\mathcal{G})||_1 \leq S, \end{aligned} \tag{10}$$

where the constant S denotes the sparsity level of the core tensor.

Acquisition and Recovery Problem

Tensor-based Model for the 3D-CASSI

- The CSVS acquisition procedure can be then expressed as

$$\mathcal{Y} = \mathcal{H}(\mathcal{F}) + \mathcal{W}, \tag{9}$$

where $\mathcal{H}(\mathcal{F}) : \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4} \rightarrow \mathbb{R}^{I_1 \times J_2 \times I_4}$ represents the CSVS operator and establishes the modulation and compression of the incoming signal.

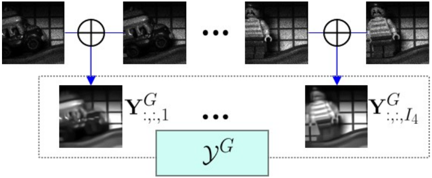
- **Recovery Problem:** For a fixed basis $\{\boldsymbol{\Psi}^{(z)}\}_{z=1}^{4}$,

$$\begin{aligned} &\underset{\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}}{\text{minimize}} \left\| \mathcal{Y} - \mathcal{H}\left(\mathcal{G} \times_1 \boldsymbol{\Psi}^{(1)} \times_2 \boldsymbol{\Psi}^{(2)} \times_3 \boldsymbol{\Psi}^{(3)} \times_4 \boldsymbol{\Psi}^{(4)}\right) \right\|_F^2 \\ &\text{subject to } ||\text{vec}(\mathcal{G})||_1 \leq S, \end{aligned} \tag{10}$$

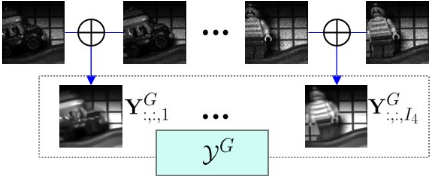
where the constant S denotes the sparsity level of the core tensor.

How the basis can be learned from the compressed measurements $\mathcal{Y}$?

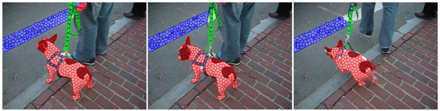
▶ IDEA: A spatial approximation of the frame t can be obtained by the summation of two consecutive measurement frames as $\mathbf{Y}^G_{:,:,t} = \mathcal{Y}(:,:,t) + \mathcal{Y}(:,:,t+1)$, where for the last frame is assigned the $(I_4 - 1)$-th estimated spatial approximation, i.e., $\mathbf{Y}^G_{:,:,I_4} = \mathbf{Y}^G_{:,:,I_4-1}$.

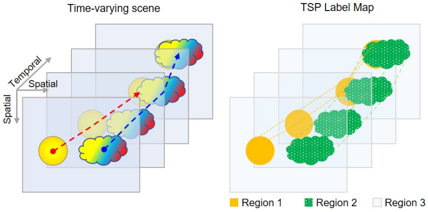


- **IDEA:** A spatial approximation of the frame t can be obtained by the summation of two consecutive measurement frames as $\mathbf{Y}^G_{:,:,t} = \mathcal{Y}(:,:,t) + \mathcal{Y}(:,:,t+1)$, where for the last frame is assigned the $(I_4 - 1)$-th estimated spatial approximation, i.e., $\mathbf{Y}^G_{:,:,I_4} = \mathbf{Y}^G_{:,:,I_4-1}$.



- Temporal Superpixels (TSP) from the Measurements





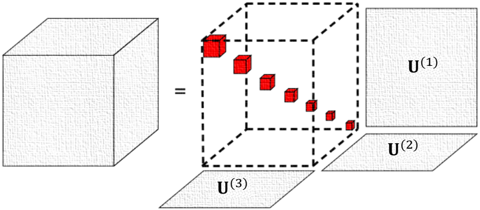
Proposed Formulation

Joint Dictionary and Recovery Problem Formulation

- Let $\mathbf{U}^{(z)} \in \mathbb{R}^{I_z \times I_z}$, for $z = 1, ..., 4$, be the factor matrices that sparsify the core tensor $\mathcal{G}$, then the **joint sparse transform and reconstruction** estimation can be expressed as

$$\{\hat{\mathbf{U}}^{(z)}, \hat{\mathcal{G}}\} \in \underset{\{\mathbf{U}^{(z)}\}_{z=1}^{4}, \mathcal{G}}{\operatorname{argmin}} \left\| \mathcal{Y} - \mathcal{H}\Big(\mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)}\Big) \right\|_F^2$$
$$\text{subject to } ||\text{vec}(\mathcal{G})||_1 \leq S,$$
$$\mathbf{U}^{(z)T}\mathbf{U}^{(z)} = \mathbf{I}^{(z)},\ z = 1, .., 4, \tag{11}$$

where $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ is the core tensor, and $\mathbf{I}^{(z)}$ is an identity matrix.



Based on TSP, the Eq. (11) can be rewritten as

$$\begin{aligned}\{\hat{\mathcal{G}}_d, \hat{\mathbf{U}}_d^{(z)}, \hat{\mathbf{U}}^{(3)}\} \in \underset{\substack{\mathcal{G}_d,\, \mathbf{U}^{(3)} \\ \{\mathbf{U}_d^{(z)}\}_{z=1}^{2,4}}}{\operatorname{argmin}} & \left\| \mathcal{Y}_d - \mathcal{H}_d\Big(\mathcal{G}_d \times_1 \mathbf{U}_d^{(1)} \times_2 \dots \times_4 \mathbf{U}_d^{(4)}\Big) \right\|_F^2 \\ \text{subject to } & ||\mathrm{vec}(\mathcal{G}_d)||_1 \leq S, \\ & \{{\mathbf{U}_d^{(z)}}^T \mathbf{U}_d^{(z)} = \mathbf{I}^{(z)}\}_{z=1,2,4},\ {\mathbf{U}^{(3)}}^T \mathbf{U}^{(3)} = \mathbf{I}^{(3)},\end{aligned} \tag{12}$$

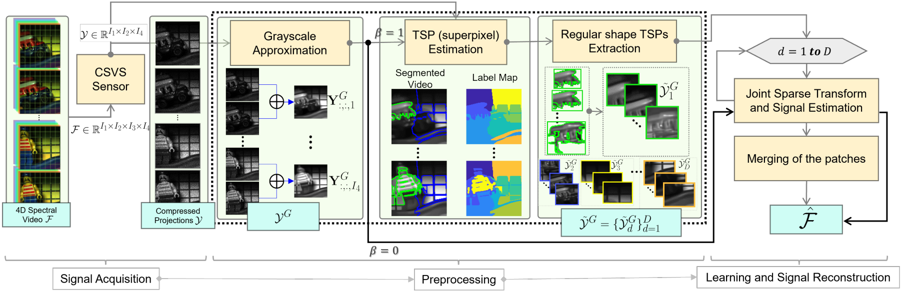
where $\mathcal{Y}_d = y^d_{i_1 i_2 i_4}$ is a TSP patch computed from the measurements.

Based on TSP, the Eq. (11) can be rewritten as

$$\{\hat{\mathcal{G}}_d, \hat{\mathbf{U}}_d^{(z)}, \hat{\mathbf{U}}^{(3)}\} \in \underset{\substack{\mathcal{G}_d, \mathbf{U}^{(3)} \\ \{\mathbf{U}_d^{(z)}\}_{z=1}^{2,4}}}{\operatorname{argmin}} \left\| \mathcal{Y}_d - \mathcal{H}_d\left(\mathcal{G}_d \times_1 \mathbf{U}_d^{(1)} \times_2 \ldots \times_4 \mathbf{U}_d^{(4)}\right) \right\|_F^2$$
$$\text{subject to } ||\text{vec}(\mathcal{G}_d)||_1 \leq S,$$
$$\{\mathbf{U}_d^{(z)T} \mathbf{U}_d^{(z)} = \mathbf{I}^{(z)}\}_{z=1,2,4}, \ \mathbf{U}^{(3)T} \mathbf{U}^{(3)} = \mathbf{I}^{(3)}, \quad (12)$$

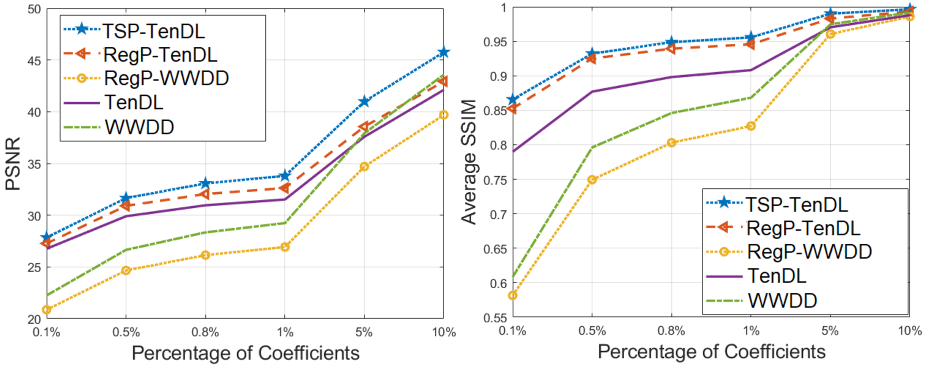
where $\mathcal{Y}_d = y^d_{i_1 i_2 i_4}$ is a TSP patch computed from the measurements.

Approach Summary



Sparsity Analysis on the Proposed Basis

TenDL:	Tensor-based sparsifying transform
TSP-TenDL:	TenDL performed in temporal superpixels
RegP-TenDL:	TenDL performed in regular patches
WWDD:	Kronecker of 2D Wavelet, 1D discrete cosine (DCT), and 1D DCT



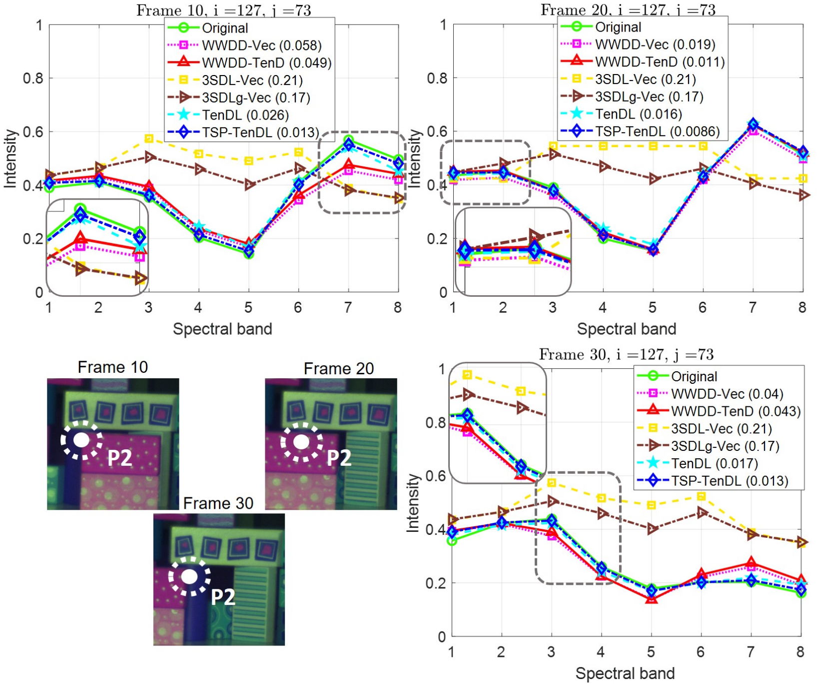
Evaluation of the compression capabilities of different sparse representations respect to the percentage of coefficients used for represent a spectral video.

Numerical Experiments

- Methods to be compared:
  - ▷ **WWDD-Vec**: vector-form recovery + fixed basis
  - ▷ **WWDD-TenD**: proposed tensor modeling + fixed basis.
  - ▷ **3SDL-Vec**: Dictionary learning (simultaneous sparse model) + PanCam.
  - ▷ **3SDLg-Vec**: Dictionary learning + grayscale approximation.
  - ▷ **TenDL**: proposed tensor model on full data.
  - ▷ **TSP-TenDL**: proposed tensor model + Temporal Superpixels.
- CA used: Temporal colored CA.
- Dataset Size:

	Spatial pixels		**Bands**	**Frames**
Size	I_1	I_2	I_3	I_4
Video 1	128	128	8	8
Video 2	256	256	8	32
Video 3	128	128	24	16

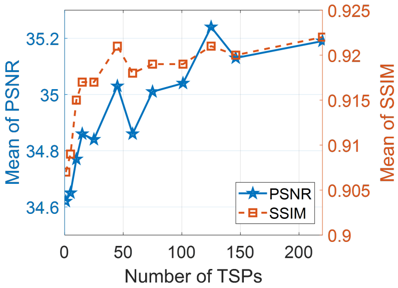
RGB profile of the originals (1st column) and the reconstructed frames 1, 5 and 10 of each video.

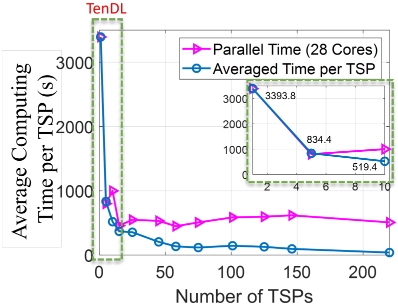

Frame 10, i =127, j =73
Original
WWDD-Vec (0.058)
WWDD-TenD (0.049)
3SDL-Vec (0.21)
3SDLg-Vec (0.17)
TenDL (0.026)
TSP-TenDL (0.013)
Intensity
Spectral band
Frame 20, i =127, j =73
Original
WWDD-Vec (0.019)
WWDD-TenD (0.011)
3SDL-Vec (0.21)
3SDLg-Vec (0.17)
TenDL (0.016)
TSP-TenDL (0.0086)
Intensity
Spectral band
Frame 10
P2
Frame 20
P2
Frame 30
P2
Frame 30, i =127, j =73
Original
WWDD-Vec (0.04)
WWDD-TenD (0.043)
3SDL-Vec (0.21)
3SDLg-Vec (0.17)
TenDL (0.017)
TSP-TenDL (0.013)
Intensity
Spectral band

Overall Accuracy and Computing Time

Table 1: Mean of PSNR, SSIM and RMSE of the Reconstructed Videos from the Different Approaches.

	Video 1		Video 2		Video 3		Video 1	Video 2	Video 3
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	RMSE		
WWDD-Vec	31.26	0.933	30.31	0.923	30.70	0.851	0.0206	0.0299	0.0221
WWDD-TenD	30.31	0.931	30.56	0.930	32.08	0.843	0.0228	0.0241	0.0196
3SDL-Vec	29.84	0.915	27.47	0.854	30.59	0.832	0.0252	0.0371	0.0229
3SDLg-Vec	29.30	0.907	26.64	0.835	30.35	0.823	0.0269	0.0411	0.0236
TenDL (Proposed)	**35.61**	**0.978**	**33.97**	**0.962**	**34.62**	**0.907**	**0.0136**	**0.0175**	**0.0137**
TSP-TenDL (Proposed)	**37.17**	**0.980**	**33.44**	**0.960**	**34.77**	**0.915**	**0.0110**	**0.0176**	**0.0130**



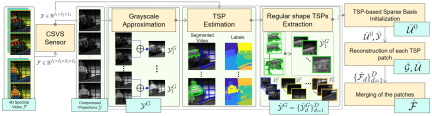


Impact of the number of TSPs in the reconstruction process and computing time using the video 3.

Summary

Design based on Tensor Representation (Basis and Recovering)

1. The **sparse representation is learned** from the compressed measurements **while the video is estimated**. The CA is fixed.
2. The method allows the higher-order correlations to be exploited in the recovery procedure.
3. TSP speeds-up the recovery!



References I

Tensor Decomposition Surveys

- L. De Lathauwer, et al, A multilinear singular value decomposition, *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- T. G. Kolda and B. W. Bader, Tensor decompositions and applications, SIAM Rev., vol. 51, no. 3, pp. 455–500, Aug. 2009.

Algorithms Used

- J. Chang, D. Wei, and J. W. Fisher III, A video representation using temporal superpixels, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2051–2058.
- K. M. L. Lopez, L. V. G. Carreño, and H. A. Fuentes, Temporal colored coded aperture design in compressive spectral video sensing, IEEE *Trans. Image Process.*, vol. 28, no. 1, pp. 253–264, Jan. 2019

Thank you for your attention!

Questions?

Merci/Gracias/Thanks to

High Dimensional Signal Processing Research Group, Universidad Industrial de Santander

TéSA Laboratory

How to solve the problem? BCD-based Formulation I

▷ The augmented Lagrangian can be written as

$$\mathcal{L}_A(\mathcal{G}, \mathcal{F}, \{\mathbf{U}^{(z)}\}_{z=1}^4, \mathcal{Q}) = \|\mathcal{Y} - \mathcal{H}(\mathcal{F})\|_F^2 + \lambda||\mathrm{vec}(\mathcal{G})||_1 \\ + (\lambda/2)\left\|\mathcal{F} - [\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}, \mathbf{U}^{(4)}]\!] + \mathcal{Q}\right\|_F^2 + \sum_{z=1}^{4} \mathcal{I}_{\mathcal{U}}(\mathbf{U}^{(z)}), \tag{13}$$

where $\mathcal{Q}$ is the Lagrange multiplier and $\mathcal{I}_{\mathcal{U}}\left(\mathbf{U}^{(z)}\right)$ is an indicator function defined as

$$\mathcal{I}_{\mathcal{U}}\left(\mathbf{U}^{(z)}\right) = \begin{cases} 1, & \text{if } \mathbf{U}^{(z)} \in \mathcal{U} \\ 0, & otherwise \end{cases}, \tag{14}$$

where $\mathcal{U} = \left\{\mathbf{U} \in \mathbb{R}^{I_z \times I_z} | \mathbf{U}^T\mathbf{U} = \mathbf{I}\right\}$, $z = 1, ..., 4$. Equation (13) can be iteratively solved by the following three steps, where each variable is updated while the others are fixed:
$1)$ $\tilde{\mathcal{F}}^{k+1}$ sub-problem:

$$\tilde{\mathcal{F}}^{k+1} \in \underset{\mathcal{F}}{\mathrm{argmin}}\ \frac{\lambda}{2}\left\|\mathcal{F}_k - [\![\mathcal{G}_k; \mathbf{U}_k^{(1)}, \mathbf{U}_k^{(2)}, \mathbf{U}_k^{(3)}, \mathbf{U}_k^{(4)}]\!] + \mathcal{Q}_k\right\|_F^2 + \frac{1}{2}\|\mathcal{Y} - \mathcal{H}(\mathcal{F}_k)\|_F^2. \tag{15}$$

▷ Solution:

$$\tilde{\mathbf{f}} = \lambda\mathrm{vec}([\![\mathcal{G}_k; \mathbf{U}_k^{(1)}, \mathbf{U}_k^{(2)}, \mathbf{U}_k^{(3)}, \mathbf{U}_k^{(4)}]\!]) + \mathbf{H}^T(\mathrm{vec}(\mathcal{Y})) = \lambda\mathbf{f} + \mathbf{H}^T(\mathbf{H}\mathbf{f}), \tag{16}$$

How to solve the problem? BCD-based Formulation II

where $\mathbf{f}$ is zero-initialized, $\mathbf{H}$ is the sensing matrix that encloses the projection operation performed by the camera, $\mathbf{H}^T$ denotes the transpose operation for $\mathbf{H}$, and $\tilde{\mathbf{f}}$ can be found from the conjugate gradient (CG) method.

▷ $\tilde{\mathcal{G}}^{k+1}$ sub-problem:

$$\tilde{\mathcal{G}}^{k+1} \in \underset{\mathcal{G}}{\operatorname{argmin}} \ \frac{\lambda}{2} \left\| \mathcal{F}_{k+1} - [\![\mathcal{G}_k; \mathbf{U}_k^{(1)}, \mathbf{U}_k^{(2)}, \mathbf{U}_k^{(3)}, \mathbf{U}_k^{(4)}]\!] + \mathcal{Q}_k \right\|_F^2 + \tau ||\mathrm{vec}(\mathcal{G}_k)||_1, \quad (17)$$

▷ Solution: This subproblem-update is a proximal operator evaluation, whose closed-form solution can be obtained from the well-known soft shrinkage operator given by

$$\tilde{\mathcal{G}}^{k+1} = \mathrm{vec}^{-1}\{\mathcal{S}_{\lambda/\tau}(\mathrm{vec}(\mathcal{F}^{k+1} + \mathcal{G}^k), \lambda/\tau)\}, \quad (18)$$

with $\mathcal{S}_{\lambda/\tau}(\mathbf{x}, \beta) := \mathrm{sgn}(\mathbf{x})\max(|\mathbf{x}| - \beta, 0)$ as the soft thresholding operator, and $\lambda, \tau > 0$ are regularization parameters.

▷ Online Transform Refinement $\tilde{\mathbf{U}}_{k+1}^{(z)}$ for $z = 1, ..., 4$ After estimating $\tilde{\mathcal{F}}^{k+1}$ and $\tilde{\mathcal{G}}^{k+1}$, the sparse transform is refined for each dimension as

$$\tilde{\mathbf{U}}_{k+1}^{(z)} \in \underset{\{\mathbf{U}^{(z)}\}_{z=1}^4}{\operatorname{argmin}} \ \frac{\lambda}{2} \left\| \mathcal{F}_{k+1} - [\![\mathcal{G}_{k+1}; \mathbf{U}_k^{(1)}, \mathbf{U}_k^{(2)}, \mathbf{U}_k^{(3)}, \mathbf{U}_k^{(4)}]\!] + \mathcal{Q}_k \right\|_F^2 + \mathcal{I}_z(\mathbf{U}^{(z)}). \quad (19)$$

How to solve the problem? BCD-based Formulation III

Note that this subproblem can be alternatively written in a general form for the mode-z as

$$\tilde{\mathbf{U}}_{k+1}^{(z)} \in \underset{\mathbf{U}^{(z)}}{\operatorname{argmin}} \frac{\lambda}{2}||\mathbf{F}_{(z)} - \mathbf{U}^{(z)}\mathbf{G}_{(z)}(\mathbf{U}^{(Z)} \otimes ...\mathbf{U}^{(z-1)} \otimes \mathbf{U}^{(z+1)}... \\ ... \otimes \mathbf{U}^{(1)})^T + \mathbf{Q}_{(z)}||_F^2 + \mathcal{I}_z\big(\mathbf{U}^{(z)}\big), \tag{20}$$

for $z = 1, ..., Z$, where $Z = 4$, are the modes of the tensor, such that each matrix $\mathbf{U}^{(z)}$ can be refined by using the mode-z of the Eq. (19). Problem in Eq. (20) is known as the *Orthogonal Procrustes* problem, whose closed-form solution is given by

$$\tilde{\mathbf{U}}_{k+1}^{(z)} = \mathbf{S}\mathbf{V}^T, \tag{21}$$

where $\mathbf{S}$ and $\mathbf{V}^T$ are obtained from the matrix-based singular value decomposition (SVD) of the factor $\big(\mathbf{F}_{(z)} + \mathbf{Q}_{(z)}\big)\big(\mathbf{G}_{(z)}\big(\mathbf{U}^{(Z)} \otimes ...\mathbf{U}^{(z-1)} \otimes \mathbf{U}^{(z+1)}... \otimes \mathbf{U}^{(1)}\big)^T\big)^T$, i.e.

$$\mathbf{S}\mathbf{\Sigma}\mathbf{V}^T = \mathrm{SVD}\big((\mathbf{F}_{(z)} + \mathbf{Q}_{(z)})\big(\mathbf{G}_{(z)}(\mathbf{U}^{(Z)} \otimes \otimes \mathbf{U}^{(z-1)} \otimes \mathbf{U}^{(z+1)}... \otimes \mathbf{U}^{(1)})^T\big)^T\big). \tag{22}$$

Thus, the sparse transform update is reduced to the computation of Eq. (21) for $n = 1, 2, 3, 4$. ▷ The multiplier is updated as

$$\tilde{\mathcal{Q}}_{k+1} = \mathcal{Q}_k + \tilde{\mathcal{F}}_{k+1} - [\![\tilde{\mathcal{G}}_{k+1}; \tilde{\mathbf{U}}_{k+1}^{(1)}, \tilde{\mathbf{U}}_{k+1}^{(2)}, \tilde{\mathbf{U}}_{k+1}^{(3)}, \tilde{\mathbf{U}}_{k+1}^{(4)}]\!]. \tag{23}$$

K. M. León-López and H. A. Fuentes, "Online tensor sparsifying transform based on temporal superpixels from compressive spectral video measurements," IEEE Transactions on Image Processing, vol. 29, pp. 5953-5963, Apr. 2020.