# Spectral Video Compression Using Convolutional Sparse Coding

C. Barajas-Solano*, J.M. Ramirez†, and H.Arguello*

*Universidad Industrial de Santander
Bucaramanga, Colombia
crisostomo.barajas@correo.uis.edu.co

†Universidad Rey Juan Carlos
Móstoles, España
juanmarcos.ramirez@urjc.es

## Abstract

Spectral Videos (SV) are datasets containing spatial-spectral-and-temporal information of a moving scene and this kind of information have been successfully used in medicine, remote sensing, and military application. However, expensive acquisition processes and difficulties in equipment manufacture lead to low-resolution datasets. Therefore, super-resolution (SR) techniques have emerged as a processing tool that recovers a high-resolution dataset by expressing the measurements as compressed versions of the desired data. Furthermore, the Convolutional Sparse Coding (CSC) has been developed as a signal model that learns a dictionary directly from the target signal, improving the reconstruction quality. This work proposes to extend the CSC formulation to consider temporal correlations in SVs, exploiting the shifting invariance property of the CSC model. The simulation results show a PSNR improvement in up to 2.5dB with respect to the state-of-the-art methods, preserving the edges and textures of the spectral video frames.

***Keywords***— spectral video, convolutional sparse coding, data compression

## 1 Introduction

Spectral Images (SI) capture 2D scenes in a specific wavelength range leading to a 3-D data representation [1]. Although SIs have been widely used in fields such as medical diagnosis [2], remote sensing [1], military operations [3], in many applications the changes in the discrete-time dimension are also of high interest. In this regard, Spectral Video (SV) has emerged as an image modality that has been used with great interest in object or human tracking [4–6], cancer detection [7], bile duct inspection [8], and several types of surgery [9].

Traditional spectral imaging techniques, such as push-broom [10] or optical band-pass filters [11], scan the scene per spatial line or for each required spectral band, respectively. This process is quite slow compared to the required speed for capturing spectral videos. This generate either major costs in manufacturing fast and reliable sensing equipment or low spatial-and-spectral resolutions in order to maintain manufacturing costs.

In this regard, Super-Resolution (SR) techniques [12, 13] have emerged has a processing tool for recovering high-resolution information from low-resolution measurements. In general, the SR problem describes the low-resolution measurements as a compressed version of the high-resolution data, which is recovered solving an inverse problem. In order to recover the high-resolution data, the state-of-the-art methods use the sparse signal representation model (SSR) [14, 15] as a signal framework for recovering the information of interest. In this regard, the SSR model uses predefined dictionaries, specifically orthonormal basis, for sparsely representing any signal.

On the other hand, synthesis frameworks, like the convolutional sparse coding (CSC), have emerged as an alternative approach for sparsely representing multidimensional signals

[16], and recently this signal model has been expanded in order to represent the spatial-spectral correlations of spectral images [17]. More precisely, CSC model characterizes a signal as a sum of cyclic convolutions between coefficient maps with their corresponding dictionary filters [18]. The collection of dictionary filters is learned from the signal to be represented, improving the quality of the reconstruction. These specificity in the signal representation, and the attached invariance to shifting and deformation property [18], represents an attractive framework for sparsely representing SVs.

The main contributions of this work are twofold: first, we present a model for representing a SV using a collection of 4D convolutionary dictionary elements and sparse coefficient maps; second, we incorporate the proposed model into a SR scheme for recovering high-resolution SVs. The results show an improvement in the recovery quality with respect to the state-of-the-art methods, where the recovered frame preserves the edges and textures of the spectral video frames.

This work is organized as follows: Section 2.1 presents the mathematical formulation of the proposed CSC4D model. Sections 2.2-2.4 presents the application and numerical solution of the CSC4D model to the SR-SV problem. Section 3 presents the results of the numerical performance tests. Finally, Section 4 presents the conclusions of the proposed method.

## 2 Proposed Work

### *2.1 Proposed CSC4D Model*

This work proposes to expand the CSC formulation to a full 4D CSC framework, in order to represent the spatial-spectral-temporal correlations in SVs. Let $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{D}}_m|_{m=1}^{M_d} \mid \boldsymbol{\mathcal{D}}_m \in \mathbb{R}^{d_M \times d_N \times d_L \times d_T}\}$ be a collection of 4D convolutional dictionary elements and $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\mathcal{X}}_m|_{m=1}^{M_d} \mid \boldsymbol{\mathcal{X}}_m \in \mathbb{R}^{M \times N \times L \times T}\}$ a collection of sparse coefficient maps, where $d_M, d_N, d_L, d_T \ll M, N, L, T$. Then a SV $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{M \times N \times L \times T}$ can be represented as

$$\boldsymbol{\mathcal{S}} = \sum_{m=1}^{M_d} \boldsymbol{\mathcal{D}}_m \overset{4}{*} \boldsymbol{\mathcal{X}}_m = \bar{\mathbf{D}}\mathbf{x} = \bar{\mathbf{X}}\mathbf{d}, \tag{1}$$

where $\overset{4}{*}$ denotes the 4D cyclic convolution. The equivalent lineal operator $\bar{\mathbf{D}} \in \mathbb{R}^{MNLT \times MNLTM_d}$ is defined as $\bar{\mathbf{D}} = [\bar{\mathbf{D}}_1 ... \bar{\mathbf{D}}_{M_d}]$ where each $\bar{\mathbf{D}}_m \in \mathbb{R}^{MNLT \times MNLT}$ is designed such that $\bar{\mathbf{D}}_m \mathbf{x}_m = \boldsymbol{\mathcal{D}}_m \overset{4}{*} \boldsymbol{\mathcal{X}}_m$, with $\mathbf{x}_m \in \mathbb{R}^{MNLT}$ as the vectorization of $\boldsymbol{\mathcal{X}}_m$. Consequently, $\mathbf{x} = [\mathbf{x}_1^T ... \mathbf{x}_{M_d}^T]^T \in \mathbb{R}^{MNLTM_d}$. The matrix $\bar{\mathbf{X}}$ and vector $\mathbf{d}$ are created in the same way profiting the convolution's commutativity property.

### *2.2 Proposed CSC4D model for SR*

In order to recover a high-resolution dataset from low-resolution measurements, we propose to express the low-resolution measurements as a decimated version of the objective dataset: $\dot{\mathbf{s}} = \mathbf{H}\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^{MNLT}$ is the vectorized version of $\boldsymbol{\mathcal{S}}$, $\dot{\mathbf{s}} \in \mathbb{R}^{M_1 N_1 L_1 T}$ is the vector containing the low-resolution measurements and $\mathbf{H} \in \mathbb{R}^{M_1 N_1 L_1 T \times MNLT}$ is a decimation matrix with $M_1 = M/\alpha$, $N_1 = N/\beta$, $L_1 = L/\gamma$, and $\alpha, \beta, \gamma \in \mathbb{N}$. Using the CSC model then $\dot{\mathbf{s}} = \mathbf{H}\bar{\mathbf{D}}\mathbf{x} = \mathbf{H}\bar{\mathbf{X}}\mathbf{d}$. For simplicity of the mathematical formulation we propose two linked optimization problems, solved alternately, for obtaining the convolutional dictionary

elements and sparse coefficient maps. Then, the general optimization problems can be formulated as

$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{H}\bar{\mathbf{D}}\mathbf{x} - \dot{\mathbf{s}} \right\|_2^2 + \lambda \left\| \mathbf{x} \right\|_1, \tag{2}$$

$$\underset{\mathbf{d}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{H}\bar{\mathbf{X}}\mathbf{d} - \dot{\mathbf{s}} \right\|_2^2 + \iota_{C_Z}(\mathbf{d}). \tag{3}$$

Considering that convolutional dictionaries have a low performance for representing the low-frequency components of multidimensional signals [19], this work uses the high-frequencies versions of a dataset of interest. This version is obtained by performing a high-pass filtering stage to the image data. On the other hand, the low-frequencies version of the original dataset is interpolated up to scale, and added to the recovered high-resolution high-frequencies version, completing the recovered high-resolution SV $\boldsymbol{\mathcal{S}}_1$, as shown in Fig 1.

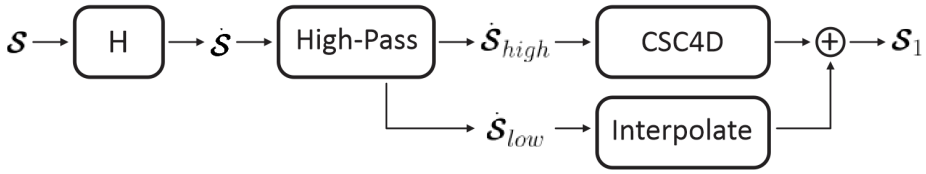


Figure 1: Proposed SR-CSC4D scheme

### *2.3 Coefficients Learning Subproblem*

Eq. (2) is known as the *Coefficients Update Problem* (CUP), and aims to learn a collection of sparse coefficient maps, hence the $\ell_1$ restriction. CUP can be solved using the Alternating Directions Multiplier Method (ADMM) [20] approach by introducing the auxiliary variables $\mathbf{u} = \bar{\mathbf{D}}\mathbf{x}$ and $\mathbf{v} = \mathbf{x}$. Then, the update steps can be written as

$$\mathbf{x}^{j+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{\rho}{2} \left\| \bar{\mathbf{D}}\mathbf{x} - \mathbf{u}^j + \mathbf{f}^j \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{x} - \mathbf{v}^j + \mathbf{g}^j \right\|_2^2, \tag{4}$$

$$\mathbf{u}^{j+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{H}\mathbf{u} - \dot{\mathbf{s}} \right\|_2^2 + \frac{\rho}{2} \left\| \bar{\mathbf{D}}\mathbf{x}^{j+1} - \mathbf{u} + \mathbf{f}^j \right\|_2^2, \tag{5}$$

$$\mathbf{v}^{j+1} = \underset{\mathbf{v}}{\operatorname{argmin}} \frac{\rho}{2} \left\| \mathbf{x}^{j+1} - \mathbf{v} + \mathbf{g}^j \right\|_2^2 + \lambda \left\| \mathbf{v} \right\|_1, \tag{6}$$

$$\mathbf{f}^{j+1} = \mathbf{f}^j + \bar{\mathbf{D}}\mathbf{x}^{j+1} - \mathbf{v}^{j+1}, \tag{7}$$

$$\mathbf{g}^{j+1} = \mathbf{g}^j + \mathbf{x}^{j+1} - \mathbf{v}^{j+1}. \tag{8}$$

Note that the product $\bar{\mathbf{D}}\mathbf{x}$ in (4) denotes the sum of 4D cyclic convolutions, which can be solved efficiently as the sum of Hadamard products in the 4D Fourier domain, as stated in the Discrete Fourier Theorem (DFT). Equation (4) is then rewritten in the Fourier domain, derived and equaled to zero with solution

$$\hat{\mathbf{x}}^{j+1} = \left( \hat{\bar{\mathbf{D}}}^H \hat{\bar{\mathbf{D}}} + \mathbf{I} \right)^{-1} \left( \hat{\bar{\mathbf{D}}}^H \hat{\mathbf{w}} + \hat{\mathbf{z}} \right), \tag{9}$$

where $\hat{\mathbf{w}} = \mathcal{F}_{4D}\left\{\mathbf{u}^j - \mathbf{f}^j\right\}$ and $\hat{\mathbf{z}} = \mathcal{F}_{4D}\left\{\mathbf{v}^j - \mathbf{g}^j\right\}$. The operator $\hat{\bar{\mathbf{D}}} = [\hat{\bar{\mathbf{D}}}_1 ... \hat{\bar{\mathbf{D}}}_{M_d}] \in \mathbb{C}^{MNLT \times MNLTM_d}$ is built analog to $\bar{\mathbf{D}}$ using the diagonalized Fourier transforms $\mathcal{F}_{4D}\{\boldsymbol{\mathcal{D}}_m\}$.

The update $\mathbf{x}^{j+1}$ is obtained from folding $\hat{\mathbf{x}}^{j+1} \in \mathbb{C}^{MNLTM_d}$ into $\{\hat{\boldsymbol{\mathcal{X}}}_m^{j+1} \in \mathbb{C}^{M\times N\times L\times T}\}$, applying the inverse 4D Fourier transform and vectorizing. Eq. (5) is also solved by deriving and equaling to zero, but solving in the spatial domain as

$$\mathbf{u}^{j+1} = \left(\mathbf{H}^T\mathbf{H} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{H}^T\mathbf{w} + \rho\mathbf{z}\right), \tag{10}$$

with $\mathbf{w} = \dot{\mathbf{s}}$ and $\mathbf{z} = \bar{\mathbf{D}}\mathbf{x}^{j+1} + \mathbf{f}^j$. Finally, Eq. (6) is solved via shrinkage/soft thresholding as

$$\mathbf{v}^{j+1} = \mathcal{S}_{\frac{\lambda}{\rho}}\left(\mathbf{x}^{j+1} + \mathbf{g}^j\right). \tag{11}$$

The variable updates for CUP are summarized in the Algorithm 1:

**Algorithm 1** CUP for CSC4D

**Input:** $\dot{\mathbf{s}}, \mathbf{H}, \mathbf{d}^j, \mathbf{u}^j, \mathbf{v}^j, \mathbf{f}^j, \mathbf{g}^j, \lambda$
**Output:** $\mathbf{u}^{j+1}, \mathbf{v}^{j+1}, \mathbf{f}^{j+1}, \mathbf{g}^{j+1}$

1: create $\bar{\mathbf{D}}$ and $\hat{\bar{\mathbf{D}}}$ from $\mathbf{d}^j$
2: create $\hat{\mathbf{w}} = \mathcal{F}_{4D}\left\{\mathbf{u}^j - \mathbf{f}^j\right\}$ and $\hat{\mathbf{z}} = \mathcal{F}_{4D}\left\{\mathbf{v}^j - \mathbf{g}^j\right\}$
3: solve $\hat{\mathbf{x}}^{j+1}$ using Eq. (9), fold it into $\{\hat{\boldsymbol{\mathcal{X}}}_m^{j+1}\}$, and $\mathbf{x}^{j+1} = \text{vec}\left(\mathcal{F}_{4D}^{-1}\left\{\{\hat{\boldsymbol{\mathcal{X}}}_m^{j+1}\}\right\}\right)$
4: solve $\mathbf{u}^{j+1}$ using Eq. (10)
5: solve $\mathbf{v}^{j+1}$ using Eq. (11)
6: update $\mathbf{f}^{j+1}$ and $\mathbf{g}^{j+1}$ using Eq. (7) and (8)
7: **return** the updated split and dual variables

### *2.4 Dictionary Learning Subproblem*

Eq. (3) is known as the *Dictionary Update Problem* (DUP), and aims to learn a collection of convolutional dictionary elements. Each dictionary element is desired to be small and one-norm, in order to avoid the scaling ambiguity between dictionary filters and coefficients. In this order of ideas we must do some annotations. First, we zero-pad each of the $M_d$ dictionary elements using the operator $\mathbf{Z}_p : \mathbb{R}^{d_M d_N d_L d_T} \to \mathbb{R}^{MNLT}$, in order to match the dimensions of the operator $\bar{\mathbf{X}} \in \mathbb{R}^{MNLT\times MNLTM_d}$. Second, the constraint set [21]

$$\mathcal{C}_{Z_p} = \left\{\mathbf{x} \in \mathbb{R}^{MNLT} : (\mathbf{I} - \mathbf{Z}_p\mathbf{Z}_p^T)\mathbf{x} = 0, \|\mathbf{x}\|_2 = 1\right\}, \tag{12}$$

guarantees that the obtained dictionary elements are normalized and the zero-padding is removed. Finally, the indicator function of the constrained set is introduced as

$$\iota_{\mathcal{C}_Z}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{C}_{Z_p} \\ \infty & \text{if } \mathbf{x} \notin \mathcal{C}_{Z_p}. \end{cases}, \tag{13}$$

and applied over each vectorized individual convolutional dictionary, but for notation simplicity it will be applied over the whole collection. Finally, DUP can be solved using ADMM by introducing the auxiliary variables $\mathbf{p} = \bar{\mathbf{X}}\mathbf{d}$ and $\mathbf{q} = \mathbf{d}$, with update steps

$$\mathbf{d}^{j+1} = \underset{\mathbf{d}}{\text{argmin}}\ \frac{\sigma}{2}\left\|\bar{\mathbf{X}}\mathbf{d} - \mathbf{p}^j + \mathbf{a}^j\right\|_2^2 + \frac{\sigma}{2}\left\|\mathbf{d} - \mathbf{q}^j + \mathbf{b}^j\right\|_2^2, \tag{14}$$

$$\mathbf{p}^{j+1} = \underset{\mathbf{p}}{\text{argmin}}\ \frac{1}{2}\left\|\mathbf{H}\mathbf{p} - \dot{\mathbf{s}}\right\|_2^2 + \frac{\sigma}{2}\left\|\bar{\mathbf{X}}\mathbf{d}^{j+1} - \mathbf{p} + \mathbf{a}^j\right\|_2^2, \tag{15}$$

$$\mathbf{q}^{j+1} = \underset{\mathbf{q}}{\operatorname{argmin}} \frac{\sigma}{2} \left\| \mathbf{d}^{j+1} - \mathbf{q} + \mathbf{b}^{j} \right\|_2^2 + \iota_{C_z}(\mathbf{q}), \tag{16}$$

$$\mathbf{a}^{j+1} = \mathbf{a}^{j} + \bar{\mathbf{X}}\mathbf{d}^{j+1} - \mathbf{p}^{j+1}, \tag{17}$$

$$\mathbf{b}^{j+1} = \mathbf{b}^{j} + \mathbf{d}^{j+1} - \mathbf{q}^{j+1}. \tag{18}$$

Again, the product $\bar{\mathbf{X}}\mathbf{d}$ in Eq. (14) denotes the sum of 4D cyclic convolutions. Eq. (14) is solved in the 4D Fourier domain, just as Eq. (4), as

$$\hat{\mathbf{d}}^{j+1} = \left( \hat{\bar{\mathbf{X}}}^{H} \hat{\bar{\mathbf{X}}} + \mathbf{I} \right)^{-1} \left( \hat{\bar{\mathbf{X}}}^{H} \hat{\mathbf{w}} + \hat{\mathbf{z}} \right), \tag{19}$$

where $\hat{\mathbf{w}} = \mathcal{F}_{4D}\left\{ \mathbf{p}^{j} - \mathbf{a}^{j} \right\}$ and $\hat{\mathbf{z}} = \mathcal{F}_{4D}\left\{ \mathbf{q}^{j} - \mathbf{b}^{j} \right\}$. Again, the update $\mathbf{p}^{j+1}$ is obtained by folding $\hat{\mathbf{p}}^{j+1}$, applying the inverse 4D Fourier transform and vectorizing. Eq. (16) is solved just as Eq. (5) in the spatial domain as

$$\mathbf{p}^{j+1} = \left( \mathbf{H}^{T}\mathbf{H} + \rho\mathbf{I} \right)^{-1} \left( \mathbf{H}^{T}\mathbf{w} + \rho\mathbf{z} \right), \tag{20}$$

with $\mathbf{w} = \dot{\mathbf{s}}$ and $\mathbf{z} = \bar{\mathbf{X}}\mathbf{d}^{j+1} + \mathbf{a}^{j}$. Finally, the close solution for Eq. (16) is the proximal [21]

$$\operatorname{prox}_{\iota_{\mathcal{C}_Z}} = \frac{\mathbf{Z}_p \mathbf{Z}_p^{T}(\mathbf{d}^{j+1} + \mathbf{b}^{j})}{\left\| \mathbf{Z}_p \mathbf{Z}_p^{T}(\mathbf{d}^{j+1} + \mathbf{b}^{j}) \right\|}. \tag{21}$$

The variable updates for DUP are summarized in the Algorithm 2. Problems CUP and DUP are solved alternately, with CUP's updates feeding DUP and vice versa, per iteration, as explained in Algorithm 3. The process continues until a desired reconstruction error, desired sparsity threshold or a maximum number of iterations is achieved.

**Algorithm 2** DUP for CSC4D

**Input:** $\dot{\mathbf{s}}, \mathbf{H}, \mathbf{x}^{j}, \mathbf{p}^{j}, \mathbf{q}^{j}, \mathbf{a}^{j}, \mathbf{b}^{j}, \lambda$
**Output:** $\mathbf{p}^{j+1}, \mathbf{q}^{j+1}, \mathbf{a}^{j+1}, \mathbf{b}^{j+1}$

1: create $\bar{\mathbf{X}}$ and $\hat{\bar{\mathbf{X}}}$ from $\mathbf{x}^{j}$
2: create $\hat{\mathbf{w}} = \mathcal{F}_{4D}\left\{ \mathbf{p}^{j} - \mathbf{a}^{j} \right\}$ and $\hat{\mathbf{z}} = \mathcal{F}_{4D}\left\{ \mathbf{q}^{j} - \mathbf{b}^{j} \right\}$
3: solve $\hat{\mathbf{d}}^{j+1}$ using Eq. (19), fold it into $\{\hat{\boldsymbol{\mathcal{D}}}_m^{j+1}\}$, and $\mathbf{d}^{j+1} = \operatorname{vec}\left( \mathcal{F}_{4D}^{-1}\left\{ \{\hat{\boldsymbol{\mathcal{D}}}_m^{j+1}\} \right\} \right)$
4: solve $\mathbf{p}^{j+1}$ using Eq. (20)
5: solve $\mathbf{q}^{j+1}$ using Eq. (21)
6: update $\mathbf{a}^{j+1}$ and $\mathbf{b}^{j+1}$ using Eq. (17) and (18)
7: **return** the updated split and dual variables

**Algorithm 3** CSC4D for SV

**Input:** $\dot{\mathbf{s}}, \mathbf{H}, \{\boldsymbol{\mathcal{D}}_m^0\}, \{\boldsymbol{\mathcal{X}}_m^0\}, \lambda$

**Output:** $\boldsymbol{\mathcal{S}}_1$

*Initialization:*

build the vectorization $\mathbf{d}^0$ from $\{\boldsymbol{\mathcal{D}}_m^0\}$

set $\{\boldsymbol{\mathcal{V}}_m^0\} = \{\boldsymbol{\mathcal{X}}_m^0\}$ and build the vectorization $\mathbf{v}^0$

set $\{\boldsymbol{\mathcal{Q}}_m^0\} = \{\boldsymbol{\mathcal{D}}_m^0\}$ and build the vectorization $\mathbf{q}^0$

set $\boldsymbol{\mathcal{U}}^0 = \boldsymbol{\mathcal{P}}^0 = \sum_{m=1}^{M_d} \boldsymbol{\mathcal{D}}_m^0 \overset{4}{*} \boldsymbol{\mathcal{X}}_m^0$ and build de vectorizations $\mathbf{u}^0$ and $\mathbf{p}^0$

set j=0;

*Iterative Process:*

**repeat**

  solve $(\mathbf{u}^{j+1}, \mathbf{v}^{j+1}, \mathbf{f}^{j+1}, \mathbf{g}^{j+1}) = \text{CUP}(\dot{\mathbf{s}}, \mathbf{H}, \mathbf{d}^j, \mathbf{u}^j, \mathbf{v}^j, \mathbf{f}^j, \mathbf{g}^j, \lambda)$

  solve $(\mathbf{p}^{j+1}, \mathbf{q}^{j+1}, \mathbf{a}^{j+1}, \mathbf{b}^{j+1}) = \text{DUP}(\dot{\mathbf{s}}, \mathbf{H}, \mathbf{v}^{j+1}, \mathbf{p}^j, \mathbf{q}^j, \mathbf{a}^j, \mathbf{b}^j, \lambda)$

  set $\mathbf{d}^j = \mathbf{q}^{j+1}$

**until** the residuals meet a given tolerance, or completed a number of iterations.

create $\{\boldsymbol{\mathcal{D}}_m\}$ and $\{\boldsymbol{\mathcal{X}}_m\}$ from $\mathbf{q}^{j+1}$ and $\mathbf{v}^{j+1}$, respectively

**return** $\boldsymbol{\mathcal{S}}_1 = \sum_{m=1}^{M_d} \boldsymbol{\mathcal{D}}_m \overset{4}{*} \boldsymbol{\mathcal{X}}_m$

## 3 Simulation Testing and Results

### *3.1 Test Datasets*

The proposed CSC4D method was tested using two laboratory-captured spectral videos, Boxes and Toy Truck (see Fig 2) [15]. Both data sets have spatial resolution $128 \times 128$, 16 spectral bands and 8 frames long.

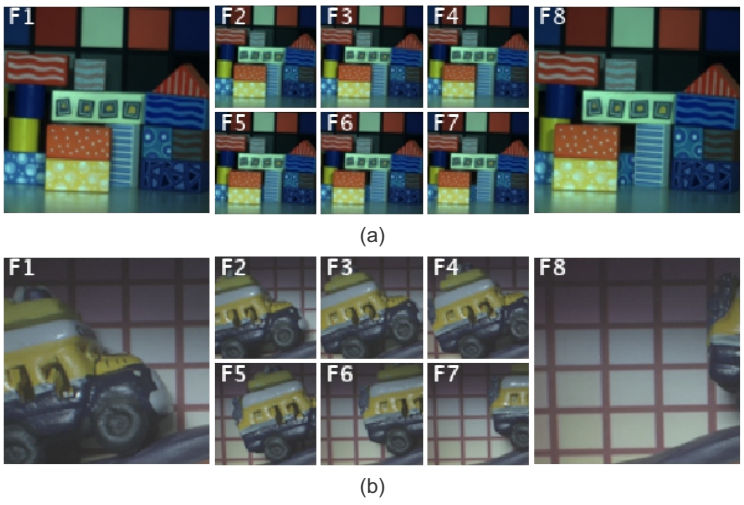


Figure 2: False RGB of the test datasets (a) Boxes and (b) Toy Truck SVs.

### *3.2 Comparison Metrics*

The CSC approach is compared against the state-of-the-art SSR approach. Particularly, we use the Kronecker basis for spatial-spectral sparse representation [22] and both the DCT

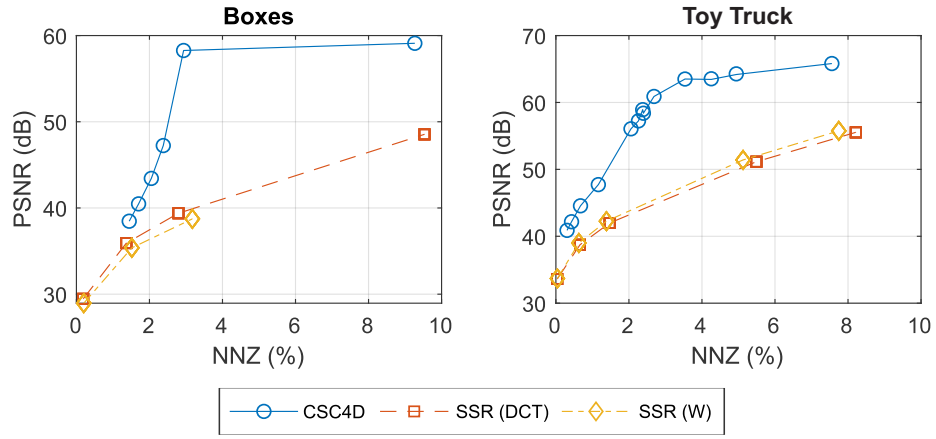


Figure 3: Reconstruction quality performance (PSNR) for various levels of sparsity (NNZ), for the proposed CSC4D and SSR approach with different time-compression schemes.

and Wavelet basis as temporal compression [14, 15]. As comparison metrics we used the Peak Signal-to-Noise Ratio (PSNR) for measuring the overall reconstruction quality, the Structural Similarity Index (SSIM) for edge sharpness, and the % of non-zero elements (NNZ) for measuring sparsity. Considering that the convolutional coefficient maps are in fact a collection of $M_d$ sparse tetrahedrons $\in \mathbb{R}^{M\times N\times L\times T}$, compared to the single sparse tetrahedron of the SSR model, then the sparsity of the convolutional coefficient maps will be measured as

$$\text{sparsity} = \max_{m=1}^{M_d} \frac{\|\boldsymbol{\mathcal{X}}_m\|_0}{MNLT}. \tag{22}$$

This is, the sparsity of the convolutional solutions will be the maximum sparsity of the individual coefficient maps.

### *3.3 Sparse Representation Quality Performance*

First we test the performance of the proposed CSC4D method for sparsely representing a SV, without decimation. For this case, $\mathbf{H} = \mathbf{I}$ in Eq. (2) and (3). The number of convolutional elements was fixed to $M_d = 30$ and the dictionary sizes were fixed to $d_M = d_N = d_L = 8$ and $d_T = 3$. For both methods, CSC4D and SSR, the parameter $\lambda$ was varied in order to get the reconstruction quality in function of the sparsity level. All other regularizer parameters were fixed to optimal values.

Fig 3 shows the sparse representation performance, for both datasets, for both frameworks. However, the main tendency is conserved. The proposed CSC4D method outperforms the SSR approach by up to 20dB, with with little evident difference between using the DCT or W temporal compression.

### *3.4 Super Resolution Performance*

The second test was to evaluate the performance of the proposed CSC4D method at recovering SVs from a spatially decimated version. For this work we set $\alpha = 2$, $\beta = 2$ and $\gamma = 1$ so $\dot{\mathbf{s}} \in \mathbb{R}^{64.64.16.8}$. All the regularizer parameters were fixed to optimal values. Finally, we only used the DCT time compression for the SSR model, for simplicity of the comparisons.

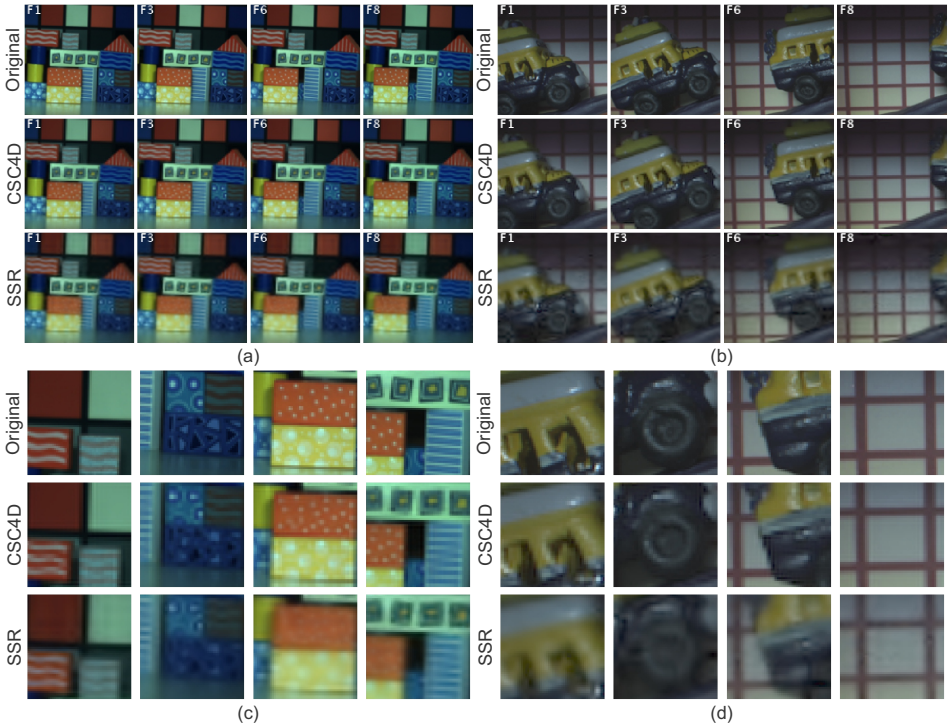


Figure 4: Example reconstructed frames with the CSC4D and SSR methods, for the datasets (a) Boxes and (b) Toy Truck, and some close-up details, (c) and (d)


Figure 5: Example reconstructed frames with the CSC4D and SSR methods, for the datasets (a) Boxes and (b) Toy Truck, $256 \times 256$ version, and some close-up details, (c) and (d)

.

For the Boxes dataset, the best reconstruction qualities were 30.07dB for CSC4D and 29.23dB for SSR. For the Toy Truck dataset, the best reconstruction qualities were 39.52dB for CSC4D and 37.08dB for SSR (see Fig. 4(a) and 4(b)). Besides outperforming the state-of-the-art method in PSNR values, the proposed CSC4D also outperforms SSR in edge reconstruction, as seen in Fig. 4(c) and 4(d). Note the sharp edges in the images reconstructed with the CSC4D method, while the SSR's reconstructed images have blurred edges.

The same experiment was recreated with higher resolution versions of the same datasets, $256 \times 256$ of spatial resolution. For the Boxes dataset, the best reconstruction qualities were 35.93dB for CSC4D and 33.48dB for SSR. For the Toy Truck dataset, the best reconstruction qualities were 46.48dB for CSC4D and 44.53dB for SSR (see Fig. 5(a) and 5(b)). Again, the proposed CSC4D generates sharper edges than the state-of-the-art approach (see Fig. 5(c) and 5(d)). The edge sharpness can be measured using SSIM, as shown in Table 1.

| | CSC4D | | SSR | |
|---|---|---|---|---|
| | Mean | Std | Mean | Std |
| Boxes | 0.97 | $5.8 \times 10^{-4}$ | 0.94 | $1.0 \times 10^{-3}$ |
| Toy Truck | 0.97 | $2.1 \times 10^{-3}$ | 0.93 | $8.3 \times 10^{-3}$ |
| Boxes $256 \times 256$ | 0.98 | $2.7 \times 10^{-4}$ | 0.92 | $8.4 \times 10^{-4}$ |
| Toy Truck $256 \times 256$ | 0.99 | $1.0 \times 10^{-3}$ | 0.97 | $2.9 \times 10^{-3}$ |

Table 1: Mean SSIM and standard deviation for the 8-frame collection, using both frameworks and four datasets.

## 4 Conclusions

The CSC3D approach, created for sparsely representing SIs, can be easily scaled to represent the SVs as 4D datasets, without introducing new temporal operators. The proposed CSC4D method profits on CSC's invariance to shifting and deformation, leading to better qualities when sparsely representing SVs and recovering SVs from decimated versions. The proposed CSC4D method improves the definition of the reconstructed edges, outperforming the state-of-the-art. This improved sharpness improves the use and processing of the recovered SVs.

## 5 References


[1] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln laboratory journal*, vol. 14, no. 1, pp. 3–28, 2003.

[2] G Lu and B Fei, "Medical hyperspectral imaging: A review," *J. Biomed. Opt*, vol. 19, no. 1, pp. 010901, 2014.

[3] D Manolakis, D Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Lab. J*, vol. 14, no. 1, pp. 79–116, 2003.

[4] S.Y. Cheng, S. Park, and M.M. Trivedi, "Multi-spectral and multiperspective video arrays for driver body tracking and activity analysis," *Comput. Vis. Image Underst.*, vol. 106, no. 2-3, pp. 245–257, 2007.

[5] H. Van-Nguyen, A. Banerjee, and R. Chellappa, "Tracking via object reflectance using a hyperspectral video camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 44–51.

[6] A. Banerjee, P. Burlina, and J. Broadwater, "Hyperspectral video for illumination-invariant tracking," in *WHISPERS '09 - 1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2009.

[7] R. Leitner, M. De-Biasio, T. Arnold, C.V. Dinh, M. Loog, and R.P.W. Duin, "Multi-spectral video endoscopy system for the detection of cancerous tissue," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 85–93, 2013.

[8] K.J. Zuzak, S.C. Naik, G. Alexandrakis, D. Hawkins, K. Behbehani, and E. Livingston, "Intraoperative bile duct visualization using nearinfrared hyperspectral video imaging," in *Proceedings of the International Conference on Sensing Technology, ICST*, 2013, pp. 145–150.

[9] D. Yi, L. Kong, F. Wang, F. Liu, S. Sprigle, and A. Adibi, "Instrument an off-shelf CCD imaging sensor into a handheld multispectral video camera," *Photonics Technology Letters, IEEE*, vol. 23, no. 10, pp. 606–608, 2011.

[10] R.G. Sellar and G.D. Boreman, "Classification of imaging spectrometers for remote sensing applications," *Opt. Eng.*, vol. 44, no. 1, pp. 013602–1 – 013602–2, 2005.

[11] N. Gat, "Imaging Spectroscopy using tunable filters," in *Proc. SPIE vol 4056*, 2000, pp. 50–64.

[12] Bård Buttingsrud and Bjørn K. Alsberg, "Superresolution of hyperspectral images," *Chemometrics and Intelligent Laboratory Systems*, vol. 84, no. 1-2 SPEC. ISS., pp. 62–68, 2006.

[13] Chiman Kwan, Joon Hee Choi, Stanley Chan, Jin Zhou, and Bence Budavari, "Resolution enhancement for hyperspectral images: A super-resolution and fusion approach," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017, pp. 6180–6184.

[14] Claudia Victoria Correa Pugliese, Diana Fernanda Galvis Carreño, and Henry Arguello Fuentes, "Sparse representations of dynamic scenes for compressive spectral video sensing," *Dyna*, vol. 83, no. 195, pp. 42–51, 2016.

[15] Kareth M. Leon-Lopez, Laura V. Galvis Carreno, and Henry Arguello Fuentes, "Temporal colored coded aperture design in compressive spectral video sensing," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 253–264, 2019.

[16] Brendt Wohlberg, "Convolutional Sparse Representation of Color Images," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, Santa Fe, NM, USA, 2016, pp. 57–60.

[17] C. Barajas-Solano, J.M. Ramirez, H. Garcia, and H. Arguello, "Tridimensional Convolutional Sparse Coding of Spectral Images," in *Optical Sensors and Sensing Congress 2019*, 2019, vol. 2019.

[18] Vardan Papyan, Jeremias Sulam, and Michael Elad, "Working locally thinking globally: Theoretical guarantees for convolutional sparse coding," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5687–5701, 2017.

[19] Brendt Wohlberg, "Boundary handling for convolutional sparse representations," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1833–1837.

[20] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 2, pp. 1–122, 2010.

[21] Brendt Wohlberg, "Efficient Algorithms for Convolutional Sparse Representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2016.

[22] G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle, "Compressive coded aperture spectral imaging: An introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 105–115, 2014.