High Order Tensor Formulation for Convolutional Sparse Coding

Adel Bibi and Bernard Ghanem
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
adel.bibi@kaust.edu.sa, bernard.ghanem@kaust.edu.sa

In this supplementary material, we start first by re-deriving the solver for our proposed TCSC for an arbitrary number of training examples N. We later discuss some further details on the computational complexity. Then, we discuss the some details of the parameters using for the experiments with some extra experiments. Lastly, we show some possible extensions to another useful regularizers that were not possible with the standard SCSC.

The problem in hand to be solved is given as follows:

$$\begin{aligned} \min_{\mathcal{D},\vec{\mathcal{X}}} \quad & \frac{1}{2}\sum_{n}^{N} \|\vec{\mathcal{Y}}_n - \mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n\|_F^2 + \lambda \|\vec{\mathcal{X}}_n\|_{\underbrace{1,\ldots,1}_{d+1}} \\ \text{s.t.} \quad & \|\vec{\mathcal{D}}_k\|_F^2 \leq 1 \; \forall k = 1,\ldots,K \end{aligned} \tag{1}$$

where $\vec{\mathcal{Y}}_n \in \mathbb{R}^{n_1 \times 1 \times n_2 \times \cdots \times n_d}$, $\mathcal{D} \in \mathbb{R}^{n_1 \times K \times n_2 \times \cdots \times n_d}$, $\vec{\mathcal{X}}_n \in \mathbb{R}^{K \times 1 \times n_2 \times \cdots \times n_d}$, and $\vec{\mathcal{D}}_k = \mathcal{D}(:,k,:,\ldots,:) \in \mathbb{R}^{n_1 \times 1 \times n_2 \times \cdots \times n_d}$.

As discussed in the main manuscript, Problem (11) will be solved by alternating between the sparse codes and the dictionaries where each subproblem is solved using ADMM.

Subproblem (1): Sparse Coding. Fixing $\mathcal{D}$, we solve:

$$\begin{aligned} \arg\min_{\vec{\mathcal{X}},\vec{\mathcal{Z}}} \quad & \frac{1}{2}\sum_{n}^{N} \|\vec{\mathcal{Y}}_n - \mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n\|_F^2 + \lambda \|\vec{\mathcal{Z}}\|_{\underbrace{1,\ldots,1}_{d+1}} \\ \text{s.t.} \quad & \vec{\mathcal{X}}_n = \vec{\mathcal{Z}}_n \;\; \forall n = 1,\ldots,N \end{aligned} \tag{2}$$

The augmented lagrangian is given as follows:

$$\begin{aligned} \mathcal{L}(\vec{\mathcal{X}}_{n\forall n}, \vec{\mathcal{Z}}_{n\forall n}, \vec{\mathcal{U}}_{n\forall n}) = & \frac{1}{2}\sum_{n}^{N} \|\vec{\mathcal{Y}}_n - \mathcal{D} \circledast \vec{\mathcal{X}}_n\|_F^2 + \lambda \sum_{n}^{N} \|\vec{\mathcal{Z}}_n\|_{\underbrace{1,\ldots,1}_{d+1}} \\ & + \frac{\rho_1}{2}\sum_{n}^{N} \|\vec{\mathcal{X}}_n - \vec{\mathcal{Z}}_n\|_F^2 + \sum_{n}^{N} < \vec{\mathcal{U}}_n, (\vec{\mathcal{X}}_n - \vec{\mathcal{Z}}_n) > \end{aligned} \tag{3}$$

It is clear that the subproblem (1), and thereafter its augmented lagrangian, is in fact separable in each of the training examples $\vec{\mathcal{Y}}_n$. Thus, the ADMM updates are exactly as presented in the main manuscript for every training sample. Moreover, and as discussed in the main manuscript, the subproblems can be solved in the Fourier domain.

Update $\vec{\mathcal{X}}$:

$$\hat{\mathcal{X}}_n^{(i)} \leftarrow (\hat{\mathcal{D}}^{(i)\top}\hat{\mathcal{D}}^{(i)} + \rho \mathbf{I}_K)^{-1}\Big(\hat{\mathcal{D}}^{(i)\top}\hat{\mathcal{Y}}_n^{(i)} + \rho\hat{\mathcal{Z}}_n^{(i)} - \hat{\mathcal{U}}_n^{(i)}\Big) \tag{4}$$

where $\{\hat{\mathcal{X}}_n^{(i)}\}_{\forall n=1,\ldots,N}$ can be concatenated along the second dimension as $\hat{\mathcal{X}} \in \mathbb{C}^{K \times N \times n_2 \times \cdots \times n_d}$.

Update $\vec{\hat{\mathcal{Z}}}$:

$$\vec{\hat{\mathcal{Z}}}_n \leftarrow \underset{\vec{\hat{\mathcal{Z}}}_n}{\arg\min}\, \lambda \|\vec{\hat{\mathcal{Z}}}_n\|\underbrace{_{1,\ldots,1}}_{d+1} + \frac{\rho}{2}\|\vec{\hat{\mathcal{Z}}}_n - \overbrace{\left(\vec{\hat{\mathcal{X}}}_n + \frac{1}{\rho}\vec{\hat{\mathcal{U}}}_n\right)}^{\vec{\mathcal{A}}_n}\|_F^2 \;\; \forall n = 1, \ldots, N \tag{5}$$

This is the proximal operator to the ℓ_1 norm, popularly known as the soft thresholding operator, $S_{\frac{\lambda}{\rho_1}}(a) = sign(a)\max(0, |a| - \frac{\lambda}{\rho_1})$. It is applied in an element-wise fashion to tensor $\vec{\mathcal{A}}_n$.

Update $\vec{\mathcal{U}}$:

$$\vec{\mathcal{U}}_n \leftarrow \vec{\mathcal{U}}_n + \rho(\vec{\mathcal{X}}_n - \vec{\mathcal{Z}}_n) \tag{6}$$

where the update for $\vec{\mathcal{U}}$ is the standard dual ascent on the dual variables.

It is clear now that the sparse coding subproblem (1) is the same as in the main manuscript except that it mush be solved for each training sample n. This is exactly analgous to all other CSC problems as in [1].

Subproblem (2): Dictionary Learning. Fixing $\vec{\mathcal{X}}_n \; \forall n$ from subproblem (1), we solve for $\mathcal{D}$ using ADMM, where the augmented Lagrangian is:

$$\begin{aligned}\mathcal{L}(\mathcal{D}, \mathcal{T}, \mathcal{G}) := &\frac{1}{2}\sum_n^N \|\vec{\mathcal{Y}}_n - (\mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n)\|_F^2 + \frac{\rho_2}{2}\|\mathcal{D} - \mathcal{T}\|_F^2 \\ &+ \langle \mathcal{G}, \mathcal{D} - \mathcal{T}\rangle + \sum_{k=1}^K \mathbb{1}_{\{\|\vec{\mathcal{T}}_k\|_F^2 \le 1\}}\end{aligned} \tag{7}$$

Note that all operations in Equation (7) are preserved under unitary matrix multiplication. Unlike the sparse coding step, the problem can be entirely solved in the Fourier domain. By using the diagonalization property of $\circledast_{HO}$, the augmented Lagrangian in the Fourier domain is rewritten as:

$$\begin{aligned}\mathcal{L}(\hat{\mathcal{D}}, \hat{\mathcal{T}}, \hat{\mathcal{G}}) := &\frac{1}{2}\sum_n^N \|\vec{\hat{\mathcal{Y}}}_n - (\hat{\mathcal{D}} \circledast_{HO} \vec{\hat{\mathcal{X}}}_n)\|_F^2 + \frac{\rho_2}{2}\|\hat{\mathcal{D}} - \hat{\mathcal{T}}\|_F^2 \\ &+ \langle \hat{\mathcal{G}}, \hat{\mathcal{D}} - \hat{\mathcal{T}}\rangle + \sum_{k}^K \mathbb{1}_{\|\vec{\hat{\mathcal{T}}}_k\|_F^2 \le 1}\end{aligned} \tag{8}$$

Update $\mathcal{D}$:

$$\hat{\mathcal{D}}^{(i)} \leftarrow \Big((\sum_n^N \hat{\mathcal{Y}}_n^{(i)} \hat{\mathcal{X}}_n^{(i)\mathbf{H}}) + \rho\hat{\mathcal{T}}^{(i)} - \hat{\mathcal{G}}^{(i)}\Big)\Big((\sum_n^N \hat{\mathcal{X}}_n^{(i)} \hat{\mathcal{X}}_n^{(i)\mathbf{H}}) + \rho\mathbf{I}_K\Big)^{-1} \tag{9}$$

This is solved again for a linear index $i = 1, \ldots, n_2 n_3 \ldots n_d$.

Update $\hat{\mathcal{T}}$: $\hat{\mathcal{T}}$ is updated using the proximal operator for the ℓ_2 unit ball as follows:

$$\hat{\mathcal{T}} \leftarrow \underset{\|\vec{\hat{\mathcal{T}}}_i\|_F^2 \le 1}{\arg\min} \;\; \frac{\rho_2}{2}\|\hat{\mathcal{T}} - \Big(\hat{\mathcal{D}} + \frac{1}{\rho_2}\hat{\mathcal{G}}\Big)\|_F^2 \tag{10}$$

As discussed in the main manuscript, the filters $\vec{\mathcal{D}}_k$ have smaller spatial support than the training images $\vec{\mathcal{Y}}$. To allow for this, the problem is re-written as follows (refer to main manuscript for details) and is solved exactly similar to the case $N = 1$.

Update $\hat{\mathcal{G}}$: $\hat{\mathcal{G}} \leftarrow \hat{\mathcal{G}} + \rho_2(\hat{\mathcal{D}} - \hat{\mathcal{T}})$

Extensions to in painting and video completion problems. In here, we show how our TCSC formulation can be extended to handle boundary conditions as proposed by [1] with the $\mathbf{M}$ matrix. Since our reconstruction images $\mathbf{y}_n$ are high order tensors, $\mathcal{M} \in \mathbb{R}^{n_1 \times 1 \times n_2 \times \cdots \times n_d}$. The problem can thus be reformulated as follows:

$$\begin{aligned} \min_{\mathcal{D},\vec{\mathcal{X}}} \quad & \frac{1}{2}\sum_{n}^{N} \|\vec{\mathcal{Y}}_n - \mathcal{M} \odot (\mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n)\|_F^2 + \lambda \|\vec{\mathcal{X}}_n\|\underbrace{_{1,\ldots,1}}_{d+1} \\ \text{s.t.} \quad & \|\vec{\mathcal{D}}_k\|_F^2 \leq 1 \; \forall k = 1, \ldots, K \end{aligned} \tag{11}$$

where the operation $\odot$ is element wise product. The updates are all exactly the same except for $\hat{\mathcal{D}}^{(i)}$ and $\hat{\mathcal{X}}_n^{(i)}$. As for the $\hat{\mathcal{X}}_n^{(i)}$ the updates are given as follows.

$$\begin{aligned} & \vec{\mathcal{X}}_n = \arg\min_{\vec{\mathcal{X}}_n} \frac{1}{2}\|\vec{\mathcal{Y}}_n - \mathcal{M} \odot (\mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n)\|_2^2 + \frac{\rho_1}{2}\|\vec{\mathcal{X}} - \vec{\mathcal{Z}}\|_2^2 + < \mathcal{U}_n, \mathcal{X}_n > \\ & \Leftrightarrow \\ & \hat{\mathcal{X}}_n^{(i)} = \arg\min_{\hat{\mathcal{X}}_n^{(i)}} \frac{1}{2}\|\hat{\mathcal{Y}}_n^{(i)} - \mathcal{M}^{(i)} \odot (\hat{\mathcal{D}}^{(i)}\hat{\mathcal{X}}_n^{(i)})\|_2^2 + \frac{\rho_1}{2}\|\hat{\mathcal{X}}_n^{(i)} - \hat{\mathcal{Z}}_n^{(i)}\|_2^2 + < \hat{\mathcal{U}}_n^{(i)}, \hat{\mathcal{X}}_n^{(i)} > \\ & \Leftrightarrow \\ & \hat{\mathcal{X}}_n^{(i)} = \arg\min_{\hat{\mathcal{X}}_n^{(i)}} \frac{1}{2}\|\hat{\mathcal{Y}}_n^{(i)} - \text{diag}(\mathcal{M}^{(i)})\hat{\mathcal{D}}^{(i)}\hat{\mathcal{X}}_n^{(i)}\|_2^2 + \frac{\rho_1}{2}\|\hat{\mathcal{X}}_n^{(i)} - \hat{\mathcal{Z}}_n^{(i)}\|_2^2 + < \hat{\mathcal{U}}_n^{(i)}, \hat{\mathcal{X}}_n^{(i)} > \\ & \Leftrightarrow \\ & \Big(\hat{\mathcal{D}}^{(i)\mathbf{H}}\text{diag}(\mathcal{M}^{(i)} \odot \mathcal{M}^{(i)})\hat{\mathcal{D}}^{(i)} + \rho_1 \mathbf{I}_K\Big)\hat{\mathcal{X}}_n^{(i)} = \hat{\mathcal{D}}^{(i)\mathbf{H}}(\mathcal{M}^{(i)} \odot \hat{\mathcal{Y}}_n^{(i)}) + \\ & \qquad \rho_1 \hat{\mathcal{Z}}_n^{(i)} - \hat{\mathcal{U}}_n^{(i)} \end{aligned} \tag{12}$$

This follows naturally since $\mathbf{a} \odot \mathbf{b} \Leftrightarrow \text{diag}(\mathbf{a})\mathbf{b} \Leftrightarrow \text{diag}(\mathbf{b})\mathbf{a}$ where $\mathcal{M}^{(i)} \in \mathbb{R}^{n_1 \times 1}$. Lastly, and in a very similar fashion, the update for $\hat{\mathcal{D}}^{(i)}$ is given as follows:

$$\hat{\mathcal{D}}^{(i)} = \arg\min_{\hat{\mathcal{D}}^{(i)}} \frac{1}{2}\sum_{n}^{N} \|\hat{\mathcal{Y}}_n^{(i)} - \text{diag}(\mathcal{M}^{(i)})(\hat{\mathcal{D}}^{(i)}\hat{\mathcal{X}}_n^{(i)})\|_F^2 + \frac{\rho_2}{2}\|\hat{\mathcal{D}}^{(i)} - \hat{\mathcal{T}}^{(i)}\|_F^2 + < \hat{\mathcal{G}}^{(i)}, \hat{\mathcal{D}}^{(i)} > \tag{13}$$

$$\Leftrightarrow \tag{14}$$

$$\text{diag}(\mathcal{M}^{(i)})\sum_{n}^{N}\Big(\text{diag}(\mathcal{M}^{(i)})\hat{\mathcal{D}}^{(i)}\hat{\mathcal{X}}_n^{(i)} - \hat{\mathcal{Y}}_n^{(i)}\Big)\hat{\mathcal{X}}_n^{(i)\mathbf{H}} + \rho_2\hat{\mathcal{D}}^{(i)} = \rho_2\hat{\mathcal{T}}^{(i)} \tag{15}$$

Complexity. Computational Complexity. Now, we discuss the computational complexity of our TCSC formulation as compared to SCSC. As for the sparse coding step, the most expensive part is solving for $\vec{\mathcal{X}}$, which involves taking n_1 2D Fourier transforms of size $n_2 \times n_3$ and solving $n_2 n_3$ linear systems each of size $K \times K$. Therefore, the total cost of updating the sparse codes can be estimated to be $\mathcal{O}(n_2 n_3 K^3) + \mathcal{O}(n_1 n_2 n_3 \log(n_2 n_3)$. Similarly, the dictionary learning subproblem involves solving $n_2 n_3$ linear systems each of size $K \times K$. The dictionary learning subproblem is solved completely in the Fourier domain and no FFTs are required. That brings the total complexity for TCSC to $\mathcal{O}(n_2 n_3 K^3) + \mathcal{O}(n_1 n_2 n_3 \log(n_2 n_3))$. As for the single channel case (SCSC), the linear system can be solved more efficiently by using the Shermon-Morrison formula. This leads to a computational complexity of $\mathcal{O}(n_2 n_3 K^2) +$
$\mathcal{O}(n_2 n_3 \log(n_2 n_3))$ per channel. Since SCSC has to be done on each channel (n_1 in total) independently, the total complexity is $\mathcal{O}(n_1 n_2 n_3 K^2) + \mathcal{O}(n_1 n_2 n_3 \log(n_2 n_3))$. TCSC is computationally more expensive than SCSC (unless $n_1 \gg K$); however, it is more attractive memory-wise. As for the memory efficiency, TCSC has $n_1 K n_3 n_4$ parameters in the dictionary and $K n_2 n_3$ in the sparse codes. On the other hand, SCSC has the same number of parameters in the dictionary and $n_1 K n_2 n_3$ parameters for the sparse codes. This means that TCSC is much more memory efficient than SCSC in general.

Memory Complexity. More importantly, TCSC has n_1 times fewer parameters as compared to SCSC, thus, making it much more memory efficient as it encodes higher order correlations.

Table 1. TCSC's training sparsity with varying λ.

λ	1	10	20
Training Sparsity	98.17%	98.63 %	99.55%

Table 2. SCSC's training sparsity with varying λ.

λ	1	5	10	20
Training Sparsity	97.77%	98.53%	98.77%	99.22%

Parameters. As for the parameters, we first list the values of λs sued in the training for both methods and their corresponding sparsity level. Tables 1 and 2 list the varying sparsity levels in the training over varying λ for both TCSC and SCSC, respectively.

Tables (3) and (4) list the optimization parameters used in training and testing for experiments (1) and (2). As for tables (5) and (6) list the optimization parameters used in training and testing for experiment (3) of video completion.

Table 3. TCSC's and SCSC's training parameters for experiments (1) and (2).

ρ_1	γ_1	$\rho_{\max_1}$	ρ_2	γ_2	$\rho_{\max_2}$	Filter size
1	10^{-2}	600	1	10^{-2}	600	11×11

Table 4. SCSC's and TCSC testing parameters for experiments (1) and (2).

	ρ_1	γ_1	$\rho_{\max_1}$	Filter size
SCSC	10^{-3}	10^{-1}	100	11×11
TCSC	10^{-4}	2×10^{-2}	100	11×11

Table 5. TCSC's training parameters for the colored video completion experiment.

ρ_1	γ_1	$\rho_{\max_1}$	ρ_2	γ_2	$\rho_{\max_2}$	Filter size	N	K
$\frac{1}{2}$	0.015	600	2	10^{-2}	600	11×11	1	100

Table 6. TCSC's testing parameters for the colored video completion experiment.

ρ_1	γ_1	$\rho_{\max_1}$	filter size	N	K
$\frac{1}{2}$	0.015	600	11×11	1	100

Other Regularizes with TCSC. A different set of regularizes that are now made possible with TCSC is the tube norm $\|.\|_{1,1,F}$ on the sparse codes $\vec{\mathcal{X}}$. This norm will encourage the optimizer to look for a sparse subset of the K filters to best reconstruct the tensors. Thus, the tube norm acts as a prior on model (or dictionary) complexity. The overall objective can now be written as follows:

$$\begin{aligned} \min_{\mathcal{D},\vec{\mathcal{X}}} \quad & \frac{1}{2}\sum_{n}^{N} \|\vec{\mathcal{Y}}_n - \mathcal{D} \circledast_{HO} \vec{\mathcal{X}}_n\|_F^2 + \lambda \|\vec{\mathcal{X}}_n\|\underbrace{_{1,\ldots,1}}_{d+1} + \beta \|\vec{\mathcal{X}}_n\|_{1,1,F} \\ \text{s.t.} \quad & \|\vec{\mathcal{D}}_k\|_F^2 \leq 1 \; \forall k = 1,\ldots,K \end{aligned} \tag{16}$$

where $\|\vec{\mathcal{X}}\|_{1,1,F}$ essentially applies the $\|.\|_F$ norm along all the dimensions $\{3, dots, d+1\}$ and applying $\|.\|_{1,1}$ on the resultant matrix. This norm induces the choice of sparse number of filters K which is a model complexity prior. A similar norm has been for 3^{rd}-order tensors as in [2]. The overall problem has two priors, one over the sparse codes $\|\vec{\mathcal{X}}_n\|_{1,\ldots,1}$ and the other over the model complexity $\|\vec{\mathcal{X}}_n\|_{1,1,F}$. The trade off between both priors can be controlled by the parameters λ and β.

Filters' Visualization. In here we show an example of how the colored filters learnt evolve through the iterations. The presented results in figure are from experiment 1 from the main manuscript over the city dataset.
Lastly, we advise the reader to refer to the video to see an example of the complete video reconstruction of experiment 3 from the main manuscript.

References

[1] F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5135–5143, 2015. 2, 3

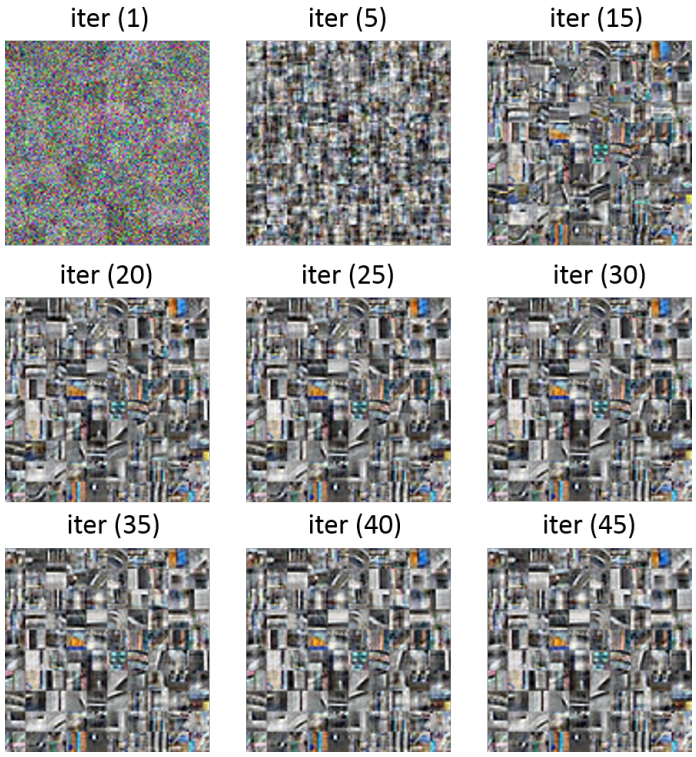

Figure 1. This figure shows the 100 3D filters learnt in consecutive iterations. The filter sizes are 11×11.

[2] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849, 2014. 4