

# Fast Convolutional Sparse Coding in the Dual Domain

Lama Affara, Bernard Ghanem, Peter Wonka

King Abdullah University of Science and Technology (KAUST), Saudi Arabia

lama.affara@kaust.edu.sa, bernard.ghanem@kaust.edu.sa, pwonka@gmail.com

## Abstract

*Convolutional sparse coding (CSC) is an important building block of many computer vision applications ranging from image and video compression to deep learning. We present two contributions to the state of the art in CSC. First, we significantly speed up the computation by proposing a new optimization framework that tackles the problem in the dual domain. Second, we extend the original formulation to higher dimensions in order to process a wider range of inputs, such as color inputs, or HOG features. Our results show a significant speedup compared to the current state of the art [8] in CSC.*

## 1. Introduction

Human vision is characterized by the response of neurons to stimuli within their receptive fields, which is realized mathematically by the convolution operator. Correspondingly for computer vision, coding the image based on a convolutional model has shown its benefits through the development and application of deep Convolutional Neural Networks. Such a model constitutes a strategy for unsupervised feature learning, and more specifically to patch-based feature learning also known as dictionary learning.

Convolutional Sparse Coding (CSC) is a special type of sparse dictionary learning algorithms. It uses the convolution operator in its image representation model rather than regular linear combinations. This results in diverse translation-invariant patches and maintains the latent structures of the underlying signal. CSC has recently been applied in a wide range of computer vision problems such as image and video processing [6, 1, 5, 19, 7], structure from motion [24], computational imaging [9], tracking [22], as well as the design of deep learning architectures [12].

Finding an efficient solution to the CSC problem however is a challenging task due to its high computational complexity and the non-convexity of its objective function. Seminal advances [3, 11, 8] in CSC have shown

computational speed-up by solving the problem efficiently in the Fourier domain where the convolution operator is transformed to element-wise multiplication. As such, the optimization is modeled as a biconvex problem with two convex subproblems that are solved iteratively and combined to form a coordinate descent solution.

Despite the performance boost attained by solving the CSC optimization problem in the Fourier domain, the problem is still deemed computationally heavy due to the dominating cost of solving large linear systems. More recent work [8, 11] makes use of the block-diagonal structure of the matrices involved and solves the linear systems in a parallel fashion.

Inspired by recent work on circulant sparse trackers [22], we model the CSC problem in the dual domain. The dual formulation casts the coding subproblem into an ADMM framework that involves solving a linear system with a lower number of parameters than previous work. This allows our algorithm to achieve not only faster convergence towards a feasible solution, but also a lower computational cost. The solution for the learning subproblem in the dual domain is achieved by applying coordinate ascent over the Lagrange multipliers and the dual parameters. Our evaluation experiments show that the dual framework achieves significant speedup over the state-of-the-art while converging to similar objective values.

In addition, due to its high computational cost, the CSC problem currently handles gray-scale images and doesn't adopt more representative features. Our formulation handles a deeper CSC model which allows learning more elaborate dictionaries such as color and HOG. This allows a richer image representation and would greatly benefit the related CSC application domains.

**Contributions.** This work makes two main contributions. (1) We formulate the CSC problem in the dual domain and show that this formulation leads to faster convergence and thus lower computation time. (2) We extend the original CSC problem to higher dimension allowing convolutional dictionary learning over richer features such as color and HOG.

## 2. Related Work

As discussed in Sec. 1, CSC has many applications and quite a few methods have been proposed for solving the complex optimization problems in CSC. In the following, we mainly review the works that focus on the computational complexity and efficiency aspects of the problem.

The seminal work of [21] proposes *Deconvolutional Networks*, a learning framework based on convolutional decomposition of images under a sparsity constraint. Unlike previous work in sparse image decomposition [17, 13, 15, 16] that builds hierarchical representations of an image on a patch level, *Deconvolutional Networks* perform a sparse decomposition over whole images. This strategy significantly reduces the redundancy among filters compared with those obtained by the patch-based approaches. Kavukcuoglu et al. [10] propose a convolutional extension to the coordinate descent sparse coding algorithm [14] to represent images using convolutional dictionaries for object recognition tasks. Following this path, Yang et al. [20] propose a supervised dictionary learning approach to improve the efficiency of sparse coding.

To efficiently solve the complex optimization problems in CSC, most of the existing approaches attempt to transform the problem into the frequency domain. Šorel and Šroubek [18] propose a non-iterative method for computing the inversion of the convolutional operator in the Fourier domain using the matrix inversion lemma. Bristow et al. [3] propose a quad-decomposition of the original objective into convex subproblems and they exploit the Alternating Direction Method of Multipliers (ADMM) approach to solve the convolution subproblems in the Fourier domain. In their follow-up work [4], a number of optimization methods for solving convolution problems and their applications are discussed. In the work of [11], the authors further exploit the separability of convolution across bands in the frequency domain. Their gain in efficiency is due to computing a partial vector (instead of a full vector). To further improve efficiency, Heide et al. [8] transform the original constrained problem into an unconstrained problem by encoding the constraints in the objective using some indicator functions. The new objective function is then further split into a set of convex functions that are easier to optimize separately. They also devise a more flexible solution by adding a diagonal matrix to the objective function to handle the boundary artifacts resulting from transforming the problem into the Fourier domain.

Various CSC methods have also been proposed for different applications. Zhang et al. [22] propose an efficient sparse coding method for sparse tracking. They also solve the problem in the Fourier domain, in which the  $\ell_1$  optimization is obtained by solving its dual problem. Unlike traditional sparse coding based image super resolution methods that divide the input image into overlapping patches, Gu et

al. [7] propose to decompose the image by filtering. Their method is capable of reconstructing local image structures. Similar to [3], the authors also solve the subproblems in the Fourier domain. The stochastic average and ADMM algorithms [23] are used for a memory efficient solution.

In this work, we attempt to provide a deeper CSC model and a more efficient solution by exploiting the optimization problem in its dual domain.

## 3. CSC Formulation and Optimization

In this section we present the mathematical formulation of the CSC problem and show our approach to the optimization solution that involves solving the dual CSC subproblems. There are multiple slightly different, but similar formulations for the CSC problem. Heide et al. [8] introduced a special case for boundary handling, but we use the more general formulation that is used by most authors. Thus, unlike [8], we assume circular boundary conditions in our derivation of the problem. Brisow et al. [3] verified that this assumption has a negligible effect for small support filters which is the case in dictionary learning where the learned patches are of a small size relative to the size of the image. In addition, they show that the Fourier transform can be replaced by the Discrete Cosine Transform when the boundary effects are problematic.

### 3.1. CSC Model

The CSC problem is generally expressed in the form

$$\begin{aligned} \arg \min_{\mathbf{d}, \mathbf{z}} \quad & \frac{1}{2} \left\| \mathbf{x} - \sum_{k=1}^K \mathbf{d}_k * \mathbf{z}_k \right\|_2^2 + \beta \sum_{k=1}^K \|\mathbf{z}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\} \end{aligned} \quad (1)$$

where  $\mathbf{d}_k \in \mathbb{R}^M$  are the vectorized 2D patches representing  $K$  dictionary elements, and  $\mathbf{z}_k \in \mathbb{R}^D$  are the vectorized sparse maps corresponding to each of the dictionary elements (see Figure 1). The dataterm represents the image  $\mathbf{x} \in \mathbb{R}^D$  modelled by the sum of convolutions of the dictionary elements with the sparse maps, and  $\beta$  controls the tradeoff between the sparsity of the feature maps and the reconstruction error. The inequality constraint on the dictionary elements assumes Laplacian distributed coefficients which ensures solving the problem at a proper scale for all elements since a larger value of  $\mathbf{d}_k$  would scale down the value of the corresponding  $\mathbf{z}_k$ . The above equation shows the objective function on a single image, and it can be easily extended to multiple images where for each image,  $K$  corresponding sparse maps are inferred, whereas all the images share the same  $K$  dictionary elements.

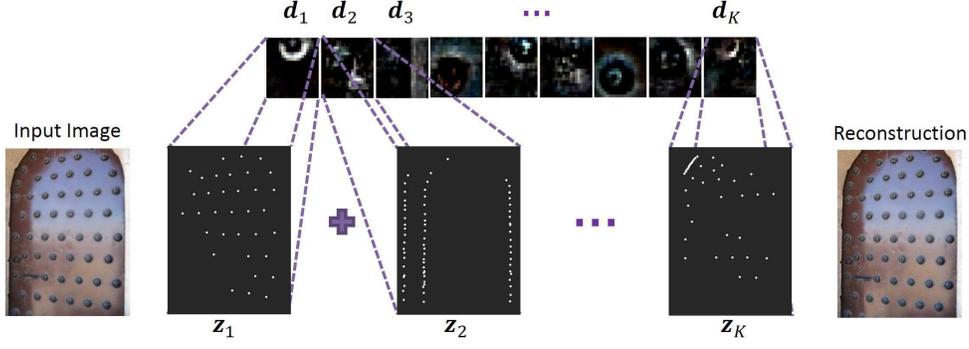


Figure 1: Convolutional Sparse Coding Model. An input image is represented by a sum of dictionary elements convolved with corresponding sparse maps and resulting in a reconstructed image that is similar to the input image.

### 3.1.1 CSC Subproblems

The objective in Eq. 1 is not jointly convex. However, solving it for one variable while keeping the other fixed leads to two convex subproblems, which we refer to by the coding subproblem and the dictionary learning subproblem. For ease of notation, we represent the convolution operations by multiplication of Toeplitz matrices with the corresponding variables.

**Coding Subproblem.** We infer the sparse maps for a fixed set of dictionary elements as shown in Eq. 2.

$$\arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \beta \|\mathbf{z}\|_1 \quad (2)$$

Here,  $\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_K]$  is of size  $D \times DK$  and is a concatenation of the convolution matrices of the dictionary elements, and  $\mathbf{z} = [\mathbf{z}_1^T \dots \mathbf{z}_K^T]^T$  is a concatenation of the vectorized sparse maps.

**Learning Subproblem.** We learn the dictionary elements for a fixed set of sparse feature maps as shown in Eq. 3.

$$\begin{aligned} \arg \min_{\mathbf{d}} \quad & \frac{1}{2} \|\mathbf{x} - \mathbf{Z}\mathbf{S}^T \mathbf{d}\|_2^2 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\} \end{aligned} \quad (3)$$

Similar to above,  $\mathbf{Z} = [\mathbf{Z}_1 \dots \mathbf{Z}_K]$  is of size  $D \times DK$  and is a concatenation of the sparse convolution matrices,  $\mathbf{d} = [\mathbf{d}_1^T \dots \mathbf{d}_K^T]^T$  is a concatenation of the dictionary elements, and  $\mathbf{S}$  projects the filter onto its spatial support.

The above two subproblems can be optimized iteratively using ADMM [3, 8], where at each iteration of ADMM, a large linear system with  $DK$  parameters needs to be solved for each of the two variables. Moreover, when applied to multiple images, solving the linear systems for the coding subproblem can be done separably, but should be done jointly for the learning subproblem since all images share

the same dictionary elements (see section 4.2 for more details on complexity analysis).

## 3.2. CSC Dual Optimization

In this section, we show our approach to solving the CSC subproblems in the dual domain. Formulating the problems in the dual domain reduces the number of parameters involved in the linear systems from  $DK$  to  $D$ , which leads to faster convergence towards a feasible solution and thus better computational performance. Since the two subproblems are convex, the duality gap is zero and solving the dual problem is equivalent to solving the primal objective. In addition, similar to [3], we also solve the convolutions efficiently in the Fourier domain as described below.

### 3.2.1 Coding Subproblem

To find the dual problem of Eq. 2, we first introduce a dummy variable  $\mathbf{r}$  with equality constraints to yield the following formulation

$$\min_{\mathbf{z}, \mathbf{r}} \frac{1}{2} \|\mathbf{r}\|_2^2 + \beta \|\mathbf{z}\|_1 \quad (4)$$

$$\text{subject to } \mathbf{r} = \mathbf{D}\mathbf{z} - \mathbf{x}$$

The Lagrangian of this problem would be

$$\mathcal{L}(\mathbf{z}, \mathbf{r}, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{r}\|_2^2 + \beta \|\mathbf{z}\|_1 + \boldsymbol{\lambda}^T (\mathbf{D}\mathbf{z} - \mathbf{x} - \mathbf{r}) \quad (5)$$

which results in the following dual function

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{r}} \frac{1}{2} \|\mathbf{r}\|_2^2 - \boldsymbol{\lambda}^T \mathbf{x} - \boldsymbol{\lambda}^T \mathbf{r} + \min_{\mathbf{z}} \beta \|\mathbf{z}\|_1 + \boldsymbol{\lambda}^T \mathbf{D}\mathbf{z} \quad (6)$$

Solving the minimizations over  $\mathbf{r}$  and  $\mathbf{z}$  and using the definition of the conjugate function to the  $l_1$  norm, we get the dual problem of Eq. 2 in the dual variable  $\boldsymbol{\lambda}$  as

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2} \boldsymbol{\lambda}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{D}^T \boldsymbol{\lambda}\|_{\infty} \leq \beta \end{aligned} \quad (7)$$

**Coding Dual Optimization.** Now we show how to solve the optimization problem in Eq. 7 using ADMM [2]. ADMM generally solves convex optimization problems by breaking the objective into two smaller pieces where each is easier to handle by iteratively updating the variables in an alternate fashion. To achieve this, we use an additional variable  $\theta = \mathbf{D}^T \lambda$  which allows us to write the problem in the general ADMM form as shown in Eq. 8. Since the dual solution to a dual problem is the primal solution, the Lagrange multiplier involved in the ADMM update step is the sparse map vector  $\mathbf{z}$  in Eq. 2.

$$\min_{\lambda, \theta} h(\lambda) + g(\theta) \text{ s.t. } \theta = \mathbf{D}^T \lambda \quad (8)$$

$$\text{where } h(\lambda) = \frac{1}{2} \lambda^T \lambda + \lambda^T \mathbf{x} \text{ and } g(\theta) = \text{ind}_C(\theta)$$

Here,  $\text{ind}_C(\cdot)$  is the indicator function defined on the convex set of constraints  $C = \{\theta \mid \|\theta\|_\infty \leq \beta\}$ . Deriving the augmented Lagrangian of the problem and solving for the ADMM update steps [2] yields the following iterative solution to the dual problem.

$$\begin{aligned} \lambda^{i+1} &= (\mathbf{D}\mathbf{D}^T + \frac{1}{\rho}\mathbf{I})^{-1}(\mathbf{D}\theta^i + \frac{1}{\rho}\mathbf{D}\mathbf{z}^i - \frac{1}{\rho}\mathbf{x}) \\ \theta^{i+1} &= \Pi_C(\mathbf{D}^T \lambda^{i+1} - \frac{1}{\rho}\mathbf{z}^i) \\ \mathbf{z}^{i+1} &= \mathbf{z}^i + \rho(\theta^{i+1} - \mathbf{D}^T \lambda^{i+1}) \end{aligned} \quad (9)$$

The variable  $\rho \in \mathbb{R}^+$  here denotes the step size for the ADMM iterations, and  $\Pi$  represents the projection operator onto the set  $C$ . The linear systems shown above do not require expensive matrix inversion or multiplication as they are transformed to elementwise divisions and multiplications when solved in the Fourier domain. This is possible because ignoring the boundary effects leads to a circulant structure for the convolution matrices, and thus they can be expressed by their base sample as

$$\mathbf{D}_k = \mathcal{F} \text{diag}(\hat{\mathbf{d}}_k) \mathcal{F}^H \quad (10)$$

where  $\hat{\mathbf{d}}$  denotes the Discrete Fourier Transform (DFT) of  $\mathbf{d}$ ,  $\mathcal{F}$  is the DFT matrix independent of  $\mathbf{d}$  and  $X^H$  is the Hermitian transpose.

In our formulation, the  $\lambda$ -update step requires solving a linear system of size  $D$ . Heide et al. [8] however solves the problem in the primal domain in which the  $\mathbf{d}$ -update step involves solving a system of size  $KD$ . This leads to faster ADMM convergence in the dual domain. Figure 2-left shows the coding subproblem convergence of our approach and that of Heide et al. Our dual formulation leads to convergence within less iterations compared to the primal domain. In addition, our approach achieves a lower objective in general at any feasible number of iterations.

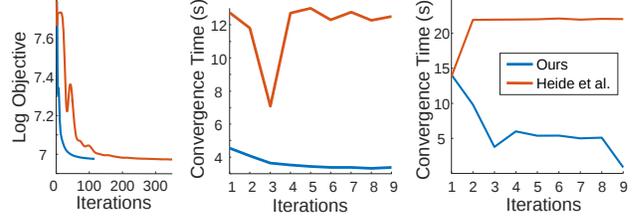


Figure 2: Convergence (left) and computation time (middle) of the coding subproblem. Computation time of the learning subproblem (right).

### 3.2.2 Learning Subproblem

To find a solution for Eq. 3, we minimize the Lagrangian of the problem (see Eq. 11) assuming optimal values for the Lagrange multipliers  $\mu_k$ . This results in the optimization problem shown in Eq. 12.

$$\mathcal{L}(\mathbf{d}, \mu) = \frac{1}{2} \|\mathbf{x} - \mathbf{Z}\mathbf{S}^T \mathbf{d}\|_2^2 + \sum_{k=1}^K \mu_k (\|\mathbf{d}_k\|_2^2 - 1) \quad (11)$$

$$\arg \min_{\mathbf{d}} \frac{1}{2} \|\mathbf{x} - \mathbf{Z}\mathbf{S}^T \mathbf{d}\|_2^2 + \sum_{k=1}^K \mu_k^* \|\mathbf{d}_k\|_2^2 \quad (12)$$

To find the dual problem of Eq. 12, we follow a similar approach to the inference subproblem by introducing a dummy variable  $\mathbf{r}$  with equality constraints such that  $\mathbf{r} = \mathbf{Z}\mathbf{S}^T \mathbf{d} - \mathbf{x}$ . Deriving the Lagrangian of the problem and minimizing over the primal variables yields the dual problem shown in 13.

$$\arg \min_{\gamma} \frac{1}{2} \gamma^T \gamma + \gamma^T \mathbf{x} + \sum_{k=1}^K \frac{1}{4\mu_k^*} \|\mathbf{S}_k \mathbf{Z}_k^T \gamma\|_2^2 \quad (13)$$

**Learning Dual Optimization.** The optimization problem in Eq. 13 has a closed form solution which when computed yields the optimal solution shown in Eq. 14.

$$\gamma^* = - \left( \mathbf{I} + \sum_{k=1}^K \frac{1}{2\mu_k^*} \mathbf{Z}_k \mathbf{S}_k^T \mathbf{S}_k \mathbf{Z}_k^T \right)^{-1} \mathbf{x} \quad (14)$$

Given the optimal value for the dual variable  $\gamma$ , we can compute the optimal value for the primal variable  $\mathbf{d}$ . By deriving the solution, the optimal value yields

$$\mathbf{d}_k^* = - \frac{1}{2\mu_k^*} \mathbf{S}_k \mathbf{Z}_k^T \gamma^* \quad \forall k = \{1, \dots, K\} \quad (15)$$

To find the optimal values for the Lagrange multipliers  $\mu_k$ , we need to assure that the KKT conditions are satisfied. At optimal  $(\mu_k^*, \mathbf{d}_k^*)$ , the solution to the primal problem 3

and its Lagrangian are equal. Thus, we end up with the below iterative update step for  $\mu_k$ .

$$\mu_k^{i+1} = \mu_k^i \| \mathbf{d}_k^i \|_2 \quad (16)$$

The learning subproblem is then solved iteratively by coordinating between updating  $\mathbf{d}_k$  as per Equations 14,15 and  $\mu_k$  as per Equation 16 until convergence is achieved.

We use conjugate gradient to solve the system involved in the  $\gamma$ -update step by applying the heavy convolution matrix multiplications in the Fourier domain. The computation cost for solving the system decreases as the number of iterations increase due to a better initialization of  $\gamma$  from the previous step. Figure 2-right shows the decreasing computation time of the learning subproblem of our approach.

For more details on the derivations of the coding and learning subproblems as well as the solutions to the equations in the Fourier domain, you may refer to the supplementary material.

### 3.2.3 Coordinate Descent

Now that we derived a solution to the two subproblems, we can use coordinate descent to solve the joint objective in Eq. 1 by alternating between the solutions for  $\mathbf{z}$  and  $\mathbf{d}$ . The full algorithm for the convolutional sparse coding problem is shown in Alg. 1.

---

#### Algorithm 1 Convolutional Sparse Coding

---

- 1: Set ADMM optimization parameter  $\rho > 0$
  - 2: Initialize variables  $\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\mu}$
  - 3: Apply FFT  $\rightarrow \hat{\mathbf{z}}, \hat{\mathbf{d}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}$
  - 4: **while** not converged **do**
  - 5:   Update  $\hat{\mathbf{z}}, \mathbf{z}, \hat{\boldsymbol{\theta}}, \boldsymbol{\theta}, \hat{\boldsymbol{\mu}}, \boldsymbol{\mu}$  iteratively by solving Eq. 9 in the Fourier domain when possible
  - 6:   Update  $\gamma, \mathbf{d}$  using Eq. 14, 15
  - 7:   Update  $\mu_k^{i+1} = \mu_k^i \| \mathbf{d}_k^i \|_2 \forall k \in \{1, \dots, K\}$
  - 8: Output solution variables by inverse FFT
- 

The coordinate descent algorithm above achieves a monotonically decreasing joint objective. We keep iterating until convergence is reached i.e. when the change in the objective value, or the solution for the optimization variables  $\mathbf{d}$  and  $\mathbf{z}$  reaches a user-defined threshold  $\tau = 10^{-3}$ . For solving the coding and learning subproblems, we also run the algorithms until convergence is achieved.

### 3.3. 3D CSC

We define 3D CSC as convolutional sparse coding over a set of 3D dictionary elements and 3D input images. Similar to before, given the input image, we seek to reconstruct the image using  $K$  patches convolved with  $K$  sparse maps. In

our formulation, unlike the input image, the sparse maps are 2D, and thus the convolution is not a complete 3D convolution, but a sum of convolutions over the third dimension of size  $J$ . Next, we show how to solve the two 3D CSC subproblems.

In the coding subproblem, the dictionary elements are 3D. Thus, each of the  $K$  convolution matrices is now of size  $DJ \times D$  where  $\mathbf{D}_k = [\mathbf{D}_{1,k}^T \dots \mathbf{D}_{J,k}^T]^T$ . The solution to the coding dual problem is shown in Eq. 9 where the inverse in the  $\lambda$ -update step is now a block diagonal matrix. Thus, the inversion can be done efficiently by parallelization over the  $D$  blocks while making use of the Woodbury inversion formula [3, 8].

The solution for the dictionary learning subproblem is straightforward since the sparse maps are 2D. We solve for  $\gamma_j^*$  and  $\mathbf{d}_{k,j}^*$  as previously shown in Eqs. 14,15 for all  $\mathbf{x}_j$  where  $j = 1 \dots J$ . Figures 1 and 8 show learned color dictionary elements from corresponding images.

## 4. Results

In this section, we give an overview of the implementation details and the parameters selected. We also show the complexity analysis and convergence of our approach compared to [8], the current state of the art. Finally, we show results on 3D CSC using color input images ( $J = 3$ ).

### 4.1. Implementation Details

We implemented the algorithm in MATLAB using the Parallel Computing Toolbox when possible, and we ran the experiments on an Intel 3.1GHz processor machine. We used the code provided by [8] in the comparisons. We evaluate our approach on the fruit and city datasets formed of 10 images each, and the house dataset containing 100 images.

We apply contrast normalization to the images prior to learning the dictionaries for both gray scale and color images; thus, the figures show normalized patches. We show results by varying the sparsity coefficient  $\beta$ , the number of dictionary elements  $K$ , and the number of images  $N$ . In our optimization, we choose a value of  $\rho = 0.1$  for the ADMM step size. We also initialize  $\mathbf{z}$  and  $\boldsymbol{\theta}$  with zeros for the first iteration of the learning subproblem, and  $\mathbf{d}$  with random numbers for the coding subproblem. Our results compare with Heide et al. [8] as it is the fastest among the published methods discussed in the related work section.

### 4.2. Complexity Analysis

In this section, we analyze the complexity of our approach compared to [8] with respect to each of the subproblems. In the below equations,  $D$  corresponds to the number of pixels in an input image,  $P$  is the number of ADMM inner iterations within the subproblems, and  $Q$  is the number of conjugate gradient iterations within the

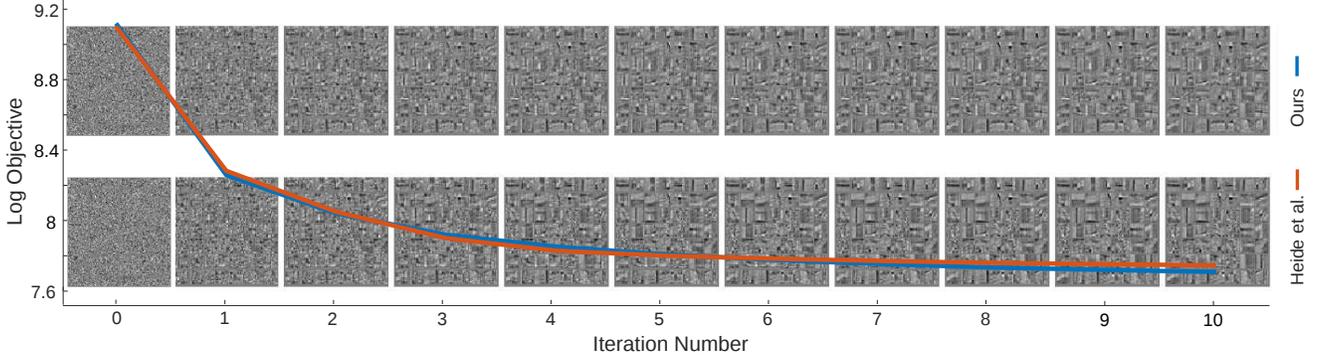


Figure 3: Dictionary learning convergence versus number of iterations, comparing our method (top) and Heide et al. [8] (bottom). Starting from the same initial dictionary shown at iteration 0, the objective value decreases as the number of iterations increase. Each point in the graph is in correspondence with a set of dictionary elements.

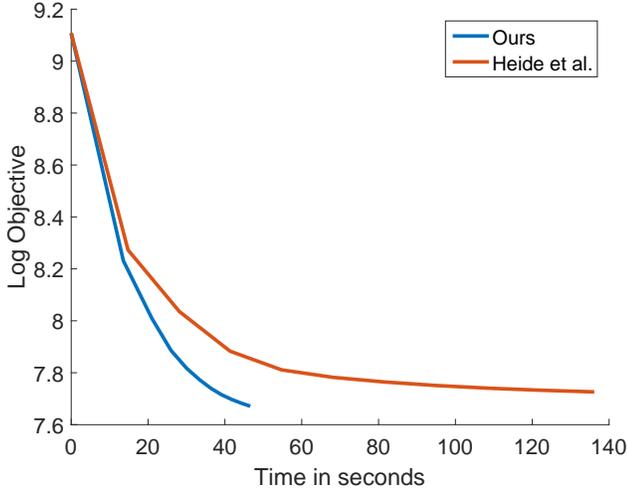


Figure 4: Objective function value as a function of time for fixed  $\beta$  and  $K$ .

learning subproblem.

**Coding Subproblem.** In the coding subproblem, the complexity of our approach is similar to that of [8] and is shown in Eq. 17. The computational complexity is dominated by solving the linear system by elementwise product and division operations because we’re solving in the Fourier domain, applying Fourier transforms and implementing the projection. It is true that computationally the two approaches are similar, but it is important to note here that the number of variables involved in solving the systems is much less in our approach, and thus we obtain a faster convergence within the subproblem as shown earlier in Figure 2.

$$PN( \underbrace{KD}_{\text{Elementwise Products}} + \underbrace{KD \log D}_{\text{Fourier Transforms}} \underbrace{KD}_{\text{Projection}} ) \quad (17)$$

**Learning Subproblem.** Here, our approach solves the problem iteratively using conjugate gradient to solve the linear system. Thus, the computational cost lies in solving elementwise products and divisions, with the additional cost of applying the Fourier transforms to the variables, and projecting to the filter support as shown in Eq. 18. Heide et al. [8] however solve the subproblem by applying ADMM which results in the computational cost shown in Eq. 19.

$$PQN( \underbrace{KD}_{\text{Elementwise Products}} + \underbrace{KD \log D}_{\text{Fourier Transforms}} + \underbrace{KD}_{\text{Projection}} ) \quad (18)$$

for  $K \leq N$ ,

$$\underbrace{K^3 D + (P-1)K^2 D}_{\text{Linear Systems}} + PN( \underbrace{KD \log D}_{\text{Fourier Transforms}} + \underbrace{KD}_{\text{Shrinkage}} )$$

for  $K > N$ ,

$$\underbrace{KN^2 D + (P-1)KND}_{\text{Linear Systems}} + PN( \underbrace{KD \log D}_{\text{Fourier Transforms}} + \underbrace{KD}_{\text{Shrinkage}} ) \quad (19)$$

We observe that the performance of our dictionary learning approach is upscaled by the number of conjugate gradient iterations involved in solving the linear system. This number decreases after each inner iteration and its cost becomes negligible within 4-6 iterations due to better initialization of  $\gamma$  at each step as shown earlier in Figure 2. In [8] however, there is the additional cost of solving the linear systems. Thus, our method has better scalability compared to Heide et al. [8] in which the linear systems solving step dominates for larger number of images and filters (see section 4.4).

### 4.3. Convergence

In this section, we analyse the convergence properties of our approach to convolutional sparse coding. In Figure 3,

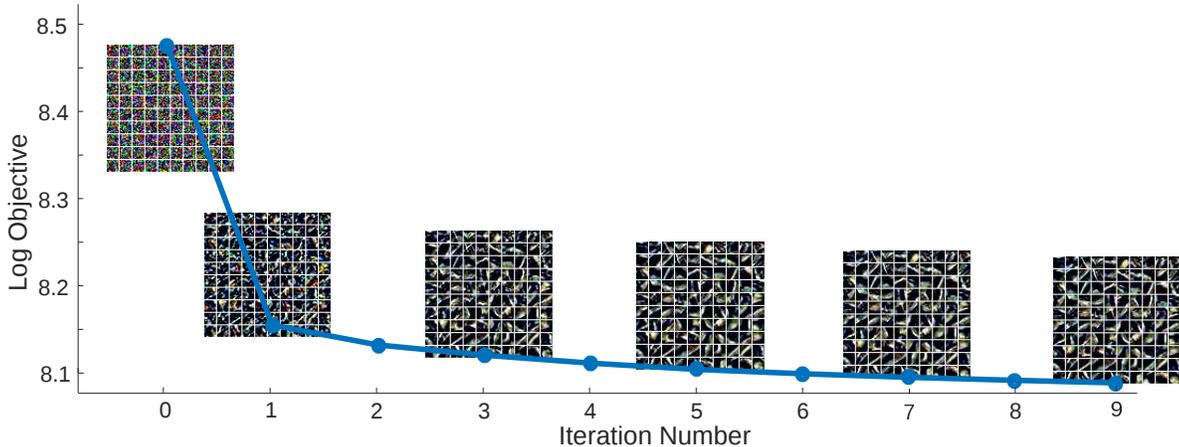


Figure 5: Color CSC convergence versus number of iterations. Starting from random initial dictionary shown at iteration 0, the objective value decreases as the number of iterations increase. Each point in the graph is in correspondence with a set of dictionary elements.

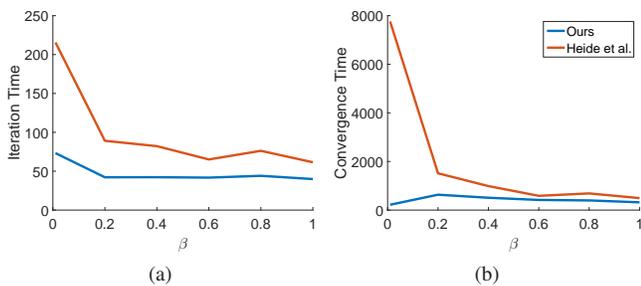


Figure 6: Convergence time as  $\beta$  is varied.

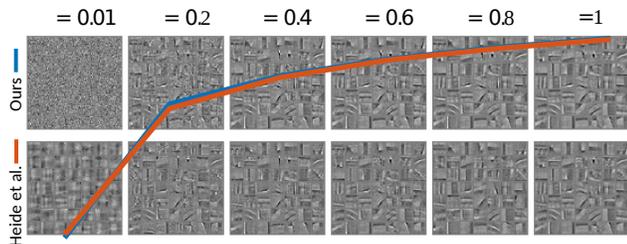


Figure 7: Dictionary element progression as a function of  $\beta$  for our method (top) and Heide et al. [8] (bottom). The curves also show the increasing objective value as  $\beta$  increases.

we plot the convergence of our method compared to the state of the art [8] on the City dataset for fixed  $\beta = 0.5$  and  $K = 49$ . We also show the progression of the learnt filters in correspondence with the curves. As shown in the figure, the two methods converge to the same solution. Figure 4 shows the plot with the same parameters as above, but vary-

ing the objective value as a function of time. This shows that our method converges significantly faster than [8].

We also plot in Figure 7 the objective value as a function of  $\beta$ . The plot shows how increasing the sparsity coefficient results in an increase in the objective value, and more importantly, it verifies that our method converges to an objective value similar to that of [8] even though we reach a solution in a lower computational time as shown in Figure 6. Figure 7 also shows how the dictionary elements vary with the variation of  $\beta$ .

#### 4.4. Scalability

In this section, we analyze the scalability of the CSC problem with increasing number of filters and images. We

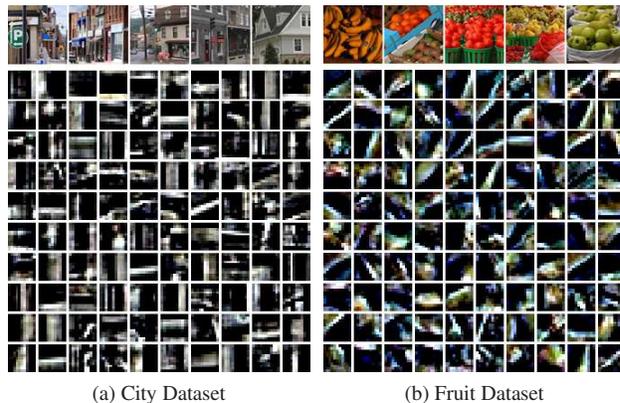


Figure 8: Learned color dictionary elements from the City and the Fruit datasets. The top row shows example images from each dataset

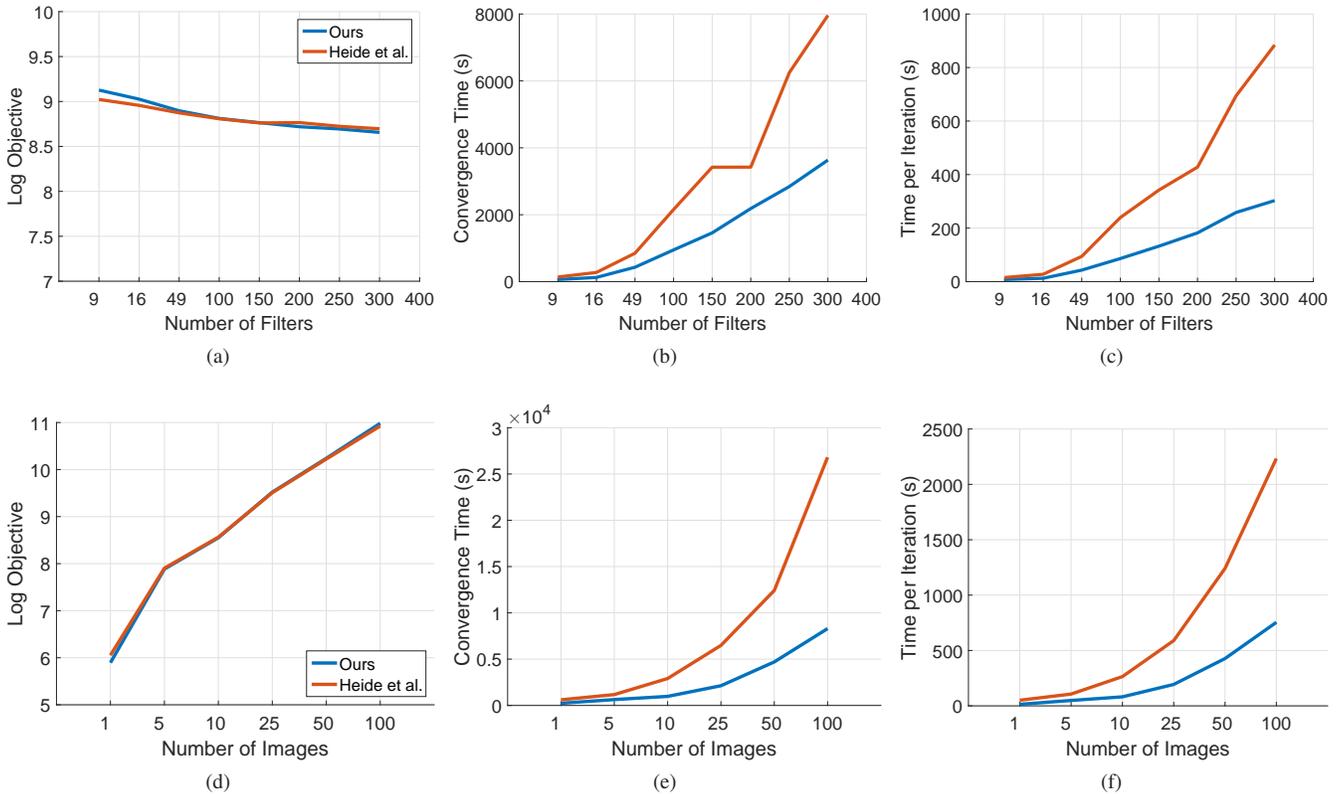


Figure 9: Objective function value (a,d), convergence time (b,e), and average iteration time in seconds (c,f) as a function of number of dictionary elements (top) and number of images (bottom).

compare our approach to Heide et. al. [8] in terms of the overall convergence time and average time per iteration for reaching the same final objective value. To assure that the two problems achieve similar overall objective value, we make sure that each of the methods runs until convergence within the coding and the learning convex subproblems given the same initial point. Figure 9 shows that the computation time of each of the methods increases with the increase of the number of filters and images. It also shows a significant speedup of an average 2.5 times for our approach over that of [8].

#### 4.5. Color CSC

In this section, we show results for 3D CSC on color images. We first show that our method achieves convergence when learning color dictionaries. Figure 5 shows the objective at each iteration for fixed  $\beta = 0.5$  and  $K = 100$ . We also show the learnt filters in correspondence with the convergence curves. Starting from random initial colored filters, the proposed method proceeds iteratively transforming random pixels to more meaningful filters reflecting the color and structure of corresponding images.

Figure 8 shows learnt color dictionaries over two differ-

ent datasets. By comparing the dictionaries, we can notice that the varying color range between the two datasets is reflected in the learnt dictionaries. In addition to difference in color, the learnt patches in the City dataset show patches with more prevalent vertical and horizontal directions. This also reflects the structure difference between the two datasets.

## 5. Conclusion and Future Work

We proposed our approach for solving the convolutional sparse coding problem by posing the optimization in the dual domain. This results in a lower computational complexity than previous work and leads to a practical algorithm that significantly speeds up convergence. We also extended the CSC problem to handle higher dimensional data. In future work, we would like to extend our algorithms to higher order tensors and experiment with additional regularizers for CSC. We could make use of the structure of the input signal and map the regularizer over the sparse maps to reflect this structure. For example, for images with a repetitive pattern, a nuclear norm can be added as a regularizer which is equivalent to making the sparse maps low rank.

## References

- [1] M. Aharon, M. Elad, and A. M. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear algebra and its applications*, 416(1):48–67, 2006. [1](#)
- [2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. 2011. [4](#)
- [3] H. Bristow, A. Eriksson, and S. Lucey. Fast Convolutional Sparse Coding. *CVPR*, 2013. [1](#), [2](#), [3](#), [5](#)
- [4] H. Bristow and S. Lucey. Optimization Methods for Convolutional Sparse Coding. *arXiv Prepr. arXiv1406.2407v1*, 2014. [2](#)
- [5] F. Couzinie-Devy, J. Mairal, F. Bach, and J. Ponce. Dictionary learning for deblurring and digital zoom. *arXiv preprint arXiv:1110.0957*, 2011. [1](#)
- [6] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. [1](#)
- [7] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. *ICCV*, 2015. [1](#), [2](#)
- [8] F. Heide, W. Heidrich, and G. Wetzstein. Fast and Flexible Convolutional Sparse Coding. *CVPR*, 2015. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [9] F. Heide, L. Xiao, A. Kolb, M. B. Hullin, and W. Heidrich. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Optics express*, 2014. [1](#)
- [10] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010. [2](#)
- [11] B. Kong and C. C. Fowlkes. Fast Convolutional Sparse Coding. *Tech. Rep. UCI*, 2014. [1](#), [2](#)
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. [1](#)
- [13] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006. [2](#)
- [14] Y. Li and S. Osher. Coordinate descent optimization for  $l_1$  minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging*, 3(3):487–503, 2009. [2](#)
- [15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009. [2](#)
- [16] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009. [2](#)
- [17] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. [2](#)
- [18] M. Šorel and F. Šroubek. Fast convolutional sparse coding using matrix inversion lemma. *Digital Signal Processing*, 55:44–51, 2016. [2](#)
- [19] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. [1](#)
- [20] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3517–3524. IEEE, 2010. [2](#)
- [21] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010. [2](#)
- [22] T. Zhang, A. Bibi, and B. Ghanem. In Defense of Sparse Tracking: Circulant Sparse Tracker. *CVPR*, 2016. [1](#), [2](#)
- [23] W. Zhong and J. T.-Y. Kwok. Fast stochastic alternating direction method of multipliers. In *ICML*, pages 46–54, 2014. [2](#)
- [24] Y. Zhu and S. Lucey. Convolutional sparse coding for trajectory reconstruction. *PAMI*, 2015. [1](#)