Bayesian Method for Image Recovery from Block Compressive Sensing

U. L. Wijewardhana*, *Student Member, IEEE,* M. Codreanu, *Member, IEEE,* and M. Latva-aho, *Senior Member, IEEE*

***Abstract*—We consider the problem of recovering an image using block compressed sensing (BCS). Traditional BCS algorithms recovers each image block independently and utilizes post-processing methods for removing the blocking artifacts. In contrast, we propose an image recovery method free of post-processing, where we utilize a *lapped transform* (LT) for the sparse representation of the image in order to reduce the blocking artifacts. Specifically, we derive an iterative image reconstruction method, where a small number of adjacent measurement blocks are jointly processed for recovering an image block. For this purpose, we propose a novel sparse Bayesian learning (SBL) algorithm.**

***Index Terms*—Block compressed sensing, image processing, sparse Bayesian learning, Bayesian compressive sensing**

I. Introduction

Compressive sensing (CS) [1], [2] framework enables the reconstruction of a high dimensional but compressible (i.e., sparse in a known basis) signal from a small number of random linear projections, utilizing computationally efficient recovery algorithms. Such conventional CS signal, or image, recovery methods require processing the entire set of compressive measurements at once. While this is entirely feasible for relatively small images (e.g., below 1 megapixel), as the image size increases, this approach become quickly prohibitively complex [3], [4]. A block compressed sensing (BCS) [3]–[5] method is usually utilized for larger images: the image is divided into small disjoint blocks and each block is acquired independently. However, if the image is recovered by reconstructing independently each image block [3]–[5], significant blocking artifacts are introduced in the reconstructed image [3]–[7].

In [7]–[9], the authors have studied this effect in the case of one-dimensional (1-D) signal recovery and proposed computationally efficient solutions, based on lapped transforms, to mitigate the blocking artifacts. Building on these results, in this work we develop an iterative image reconstruction method, based on 2-D lapped transforms and sparse Bayesian learning (SBL) [10]. Specifically, to suppress the blocking artifacts, we jointly process a small number of adjacent measurement blocks for recovering an image block. Further, we develop a Kalman-like implementation, where the information obtained in the preceding processing intervals is efficiently reused to refine the signal estimation in the current estimation interval.

This research was supported by Academy of Finland and Infotech Oulu Doctoral Program.

The authors are with the Centre for Wireless Communications, Department of Communications Engineering, University of Oulu, 90014 Oulu, Finland. (e-mail: {uditha, codreanu, matti.latva-aho}@ee.oulu.fi)

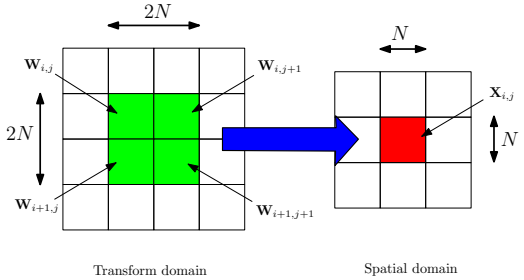

Fig. 1: Synthesis of image $\mathbf{X}$ using a lapped transform. The relationship between the image blocks $\mathbf{X}_{i,j}$ and the transform domain coefficient blocks $\mathbf{W}_{i,j}$. The image $\mathbf{X}$ has a sparse representation in a LT with $L = 2N$.

II. System model and problem formulation

We consider the recovery of an image $\mathbf{X}$ from block compressed sensing (BCS) measurements. Specifically, we divide the image $\mathbf{X}$ into $H \times W$ non-overlapping blocks with each block having size $N \times N$. Suppose that $\mathbf{X}_{i,j} \in \mathbb{R}^{N\times N}$ denotes the (i,j)-th block of the image $\mathbf{X}$. Then the linear acquisition model is given by

$$\mathbf{y}_{i,j} = \mathbf{\Phi}_{i,j}\mathbf{x}_{i,j} + \mathbf{e}_{i,j}, \quad i = 1,\dots,H \quad j = 1,\dots,W \quad (1)$$

where $\mathbf{x}_{i,j} = \text{vec}(\mathbf{X}_{i,j}) \in \mathbb{R}^{N^2}$; $\mathbf{y}_{i,j} \in \mathbb{R}^M$ contains $M < N^2$ linear measurements of $\mathbf{X}_{i,j}$; the measurement matrix is denoted by $\mathbf{\Phi}_{i,j} \in \mathbb{R}^{M\times N^2}$ and the measurement noise vector is denoted by $\mathbf{e}_{i,j} \in \mathbb{R}^M$, where each entry is assumed to be an independent Gaussian random variable with zero mean and known variance σ^2, i.e., $\mathbf{e}_{i,j} \sim \mathcal{N}(0, \sigma^2 I_M)$.

To reduce the blocking artifacts that occur with traditional block transforms, we use a *lapped transform* (LT) for the sparse representation of the image $\mathbf{X}$ [11]–[14]. By utilizing the *synthesis (or inverse)* transform matrix denoted by $\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_0\end{bmatrix} \in \mathbb{R}^{N\times 2N}$, the image block $\mathbf{X}_{i,j}$ can be synthesized as [12], [14] [1]

$$\mathbf{X}_{i,j} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_0\end{bmatrix} \begin{bmatrix}\mathbf{W}_{i,j} & \mathbf{W}_{i,j+1} \\ \mathbf{W}_{i+1,j} & \mathbf{W}_{i+1,j+1}\end{bmatrix} \begin{bmatrix}\mathbf{Q}_1^{\mathrm{T}} \\ \mathbf{Q}_0^{\mathrm{T}}\end{bmatrix}, \quad (2)$$

where $\mathbf{W}_{i,j} \in \mathbb{R}^{N\times N}$ contains the transform domain coefficients that we expected to be sparse. Thus, an arbitrary portion of the image $\mathbf{X}$ can be represented using the transform domain coefficients as in Fig. 1.

Using block matrix multiplication on (2) and then utilizing $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\text{vec}(\mathbf{B})$, we obtain

$$\mathbf{x}_{i,j} = \mathbf{\Psi}_{11}\mathbf{w}_{i,j} + \mathbf{\Psi}_{10}\mathbf{w}_{i+1,j} + \mathbf{\Psi}_{01}\mathbf{w}_{i,j+1} + \mathbf{\Psi}_{00}\mathbf{w}_{i+1,j+1}, \quad (3)$$

[1]The LT satisfy the perfect reconstruction property if $\mathbf{P}_1\mathbf{Q}_1^{\mathrm{T}} + \mathbf{P}_0\mathbf{Q}_0^{\mathrm{T}} = \mathbf{I}$ and $\mathbf{P}_1\mathbf{Q}_0^{\mathrm{T}} = \mathbf{P}_0\mathbf{Q}_1^{\mathrm{T}} = \mathbf{0}$, where $[\mathbf{P}_1 \;\; \mathbf{P}_0]$ is the forward transform matrix.

where, $\mathbf{w}_{i,j} = \text{vec}(\mathbf{W}_{i,j}) \in \mathbb{R}^{N^2}$ and $\mathbf{\Psi}_{kl} = \mathbf{Q}_k \otimes \mathbf{Q}_l \in \mathbb{R}^{N^2 \times N^2}$ for $k, l = 0, 1$. Now, by substituting (3) in the measurement model (1), we obtain the linear relationship between the measurement vector $\mathbf{y}_{i,j}$ and the transform domain coefficient vectors $\mathbf{w}_{i,j}$ as

$$\mathbf{y}_{i,j} = \mathbf{\Phi}_{i,j} \begin{bmatrix} \mathbf{\Psi}_{11} & \mathbf{\Psi}_{10} & \mathbf{\Psi}_{01} & \mathbf{\Psi}_{00} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{i,j} \\ \mathbf{w}_{i+1,j} \\ \mathbf{w}_{i,j+1} \\ \mathbf{w}_{i+1,j+1} \end{bmatrix} + \mathbf{e}_{i,j}. \tag{4}$$

Consider the problem of estimating $\mathbf{x}_{i,j}$ (or equivalently the image block $\mathbf{X}_{i,j}$) using the measurement vectors $\mathbf{y}_{i,j}$ for $i = 1, \ldots, H$ $j = 1, \ldots, W$. It is clear from (3) that $\mathbf{x}_{i,j}$ can be synthesized with the knowledge of the sparse vectors $\mathbf{w}_{i,j}$, $\mathbf{w}_{i+1,j}$, $\mathbf{w}_{i,j+1}$ and $\mathbf{w}_{i+1,j+1}$. However, estimating $\mathbf{x}_{i,j}$ independently from $\mathbf{y}_{i,j}$ (by estimating $\mathbf{w}_{i,j}$, $\mathbf{w}_{i+1,j}$, $\mathbf{w}_{i,j+1}$ and $\mathbf{w}_{i+1,j+1}$ from $\mathbf{y}_{i,j}$) is not optimal since multiple $\mathbf{y}_{i,j}$ contain measurements of a transform domain coefficient block $\mathbf{w}_{i,j}$ (see (4)). The optimal solution would be to jointly estimate all sparse vectors $\mathbf{w}_{i,j}$ for $i = 1, \ldots, H+1$ $j = 1, \ldots, W+1$ from the measurement vectors $\mathbf{y}_{i,j}$ for $i = 1, \ldots, H$ $j = 1, \ldots, W$. For a large image, the memory requirement and the computational complexity of the recovery algorithm grows exceedingly high for such approach.

III. Recovery of the Image Using Sliding Window Technique

In order to reduce the memory burden as well as the complexity of the recovery algorithm, we propose a reconstruction algorithm based on sliding window processing. We consider that a fairly accurate estimate of the sparse vector $\mathbf{w}_{i,j}$ can be obtained by processing only the measurement vectors $\mathbf{y}_{k,l}$ for $k = i-d, \ldots, i+d-1$, $l = j-d, \ldots, j+d-1$. Thus, in this section we look into the problem of reconstructing the sparse vectors $\mathbf{w}_{i,j}$ for $i = 1, \ldots, H+1$ $j = 1, \ldots, W+1$ in terms of sliding window processing and propose a recovery algorithm based on sparse Bayesian learning (SBL). In the proposed algorithm d is a design parameter, and the simulation results in section IV show that even the modest values such as $d = 1, 2$ provide a good signal recovery performance.

A. Processing window

Let a fairly accurate estimation of $\mathbf{w}_{i,j}$, given by $\hat{\mathbf{w}}_{i,j}$, can be obtained by processing the measurement vectors $\mathbf{y}_{k,l}$ for $k = i-d, \ldots, i+d-1$, $l = j-d, \ldots, j+d-1$. Thus, we denote the measurement vector inside the processing window (or the active interval) for estimating $\mathbf{w}_{i,j}$ by $\tilde{\mathbf{y}}_{i,j} \in \mathbb{R}^{4d^2M}$ which consists of the measurement blocks $\mathbf{y}_{k,l}$ for $k = i-d, \ldots, i+d-1$, $l = j-d, \ldots, j+d-1$. The linear measurement model for the active interval can be expressed as

$$\tilde{\mathbf{y}}_{i,j} = \tilde{\mathbf{B}}_{i,j} \tilde{\mathbf{w}}_{i,j} + \tilde{\mathbf{e}}_{i,j}, \tag{5}$$

where $\tilde{\mathbf{B}}_{i,j} \in \mathbb{R}^{4d^2M \times (2d+1)^2N^2}$ denotes the system matrix, which models the composite effect of the measurement and the representation matrices, inside the current active interval; $\tilde{\mathbf{w}}_{i,j} \in \mathbb{R}^{(2d+1)^2N^2}$ denotes the sparse vector inside the active interval and the measurement noise vector is denoted by $\tilde{\mathbf{e}}_{i,j} \in \mathbb{R}^{4d^2M}$. The sparse vector $\tilde{\mathbf{w}}_{i,j}$ consists of the transform domain coefficient blocks $\mathbf{w}_{k,l}$ for $k = i-d, \ldots, i+d$, $l = j-d, \ldots, j+d$ (see Fig. 2) and $\tilde{\mathbf{e}}_{i,j}$ contains the error vectors $\mathbf{e}_{k,l}$ $k = i-d, \ldots, i+d-1$, $l = j-d, \ldots, j+d-1$.

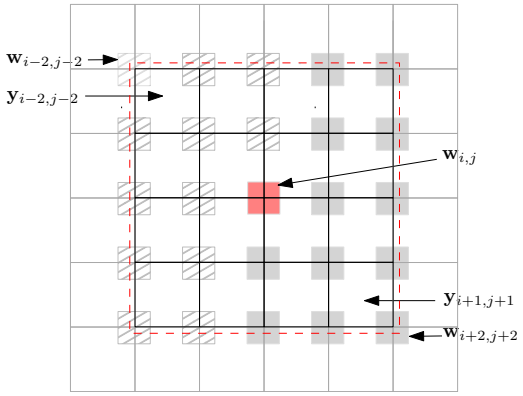

Fig. 2: Processing window. The small squares illustrates the sparse transform domain coefficient blocks inside the active interval designed to output an estimation for $\mathbf{w}_{i,j}$ (marked in red color) in case of $d = 2$. The large squares shows the measurement blocks inside the active interval.

Since multiple intervals are overlapping each other, they share some common transform domain coefficient blocks $\mathbf{w}_{i,j}$. Hence, we can utilize the preliminary information on these sparse vectors (i.e., the estimates obtained from the preceding intervals) to improve the performance of the recovery algorithm. Let $\breve{\mathbf{w}}_{i,j}$ denote the subvector of $\tilde{\mathbf{w}}_{i,j}$ containing the sparse vectors $\mathbf{w}_{i,j}$ estimated from the preceding intervals. In order to single out the contribution of $\breve{\mathbf{w}}_{i,j}$ on $\tilde{\mathbf{y}}_{i,j}$, we express the measurement model in (5) as

$$\tilde{\mathbf{y}}_{i,j} = \breve{\mathbf{B}}_{i,j} \breve{\mathbf{w}}_{i,j} + \bar{\mathbf{B}}_{i,j} \bar{\mathbf{w}}_{i,j} + \tilde{\mathbf{e}}_{i,j}, \tag{6}$$

where we partitioned the system matrix as $\tilde{\mathbf{B}}_{i,j} = \begin{bmatrix} \breve{\mathbf{B}}_{i,j} & \bar{\mathbf{B}}_{i,j} \end{bmatrix}$; $\breve{\mathbf{B}}_{i,j}$ denotes the submatrix of $\tilde{\mathbf{B}}_{i,j}$ obtained by keeping the columns of $\tilde{\mathbf{B}}_{i,j}$ corresponding to the indices of $\breve{\mathbf{w}}_{i,j}$, $\bar{\mathbf{B}}_{i,j}$ denotes the submatrix of $\tilde{\mathbf{B}}_{i,j}$ obtained by removing the columns of $\tilde{\mathbf{B}}_{i,j}$ corresponding to the indices of $\breve{\mathbf{w}}_{i,j}$ and we have partitioned the sparse vector $\tilde{\mathbf{w}}_{i,j} = \begin{bmatrix} \breve{\mathbf{w}}_{i,j}^{\mathrm{T}} & \bar{\mathbf{w}}_{i,j}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The goal of the recovery algorithm is to obtain an estimation for the sparse vector $\bar{\mathbf{w}}_{i,j}$ by utilizing the measurement vector $\tilde{\mathbf{y}}_{i,j}$ while removing the effect of the estimate of $\breve{\mathbf{w}}_{i,j}$ on $\tilde{\mathbf{y}}_{i,j}$. Note that, in case we have an point estimate of $\breve{\mathbf{w}}_{i,j}$, we can use $\tilde{\mathbf{y}}_{i,j} - \breve{\mathbf{B}}_{i,j} \breve{\mathbf{w}}_{i,j}$ as the new measurement vector and utilize a sparse signal recovery algorithm for the estimation of $\bar{\mathbf{w}}_{i,j}$. However, such an approach is prone to error propagation. Hence, next we derive a recovery algorithm based on SBL, which takes into account the uncertainty of the reconstruction of $\breve{\mathbf{w}}_{i,j}$.

B. Recovery algorithm based on SBL

In the SBL framework, we try to estimate the posterior density of the unknown vector $\bar{\mathbf{w}}_{i,j}$ given the measurement vector $\tilde{\mathbf{y}}_{i,j}$ by employing a sparsity promoting prior on $\bar{\mathbf{w}}_{i,j}$. We utilize the standard 2-stage hierarchical prior model [10], [15] in the sequel. For the notational convenience, let $\bar{\mathbf{w}}_{i,j} \in \mathbb{R}^K$, i.e., $K = cN^2$ where c is the number of transform domain coefficient blocks in $\bar{\mathbf{w}}_{i,j}$, label the elements of $\bar{\mathbf{w}}_{i,j}$ as

$w_1, \ldots, w_K$ and denote the set of all elements in $\bar{\mathbf{w}}_{i,j}$ by $\mathcal{N}_{i,j}$, i.e., $\mathcal{N}_{i,j} = \{1, \ldots, K\}$. In the first stage, we model each element of the vector $\bar{\mathbf{w}}_{i,j}$ as an independent Gaussian random variable with zero mean and individual precision parameters, i.e.,

$$p\big(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}\big) = \prod_{k=1}^{K} \mathcal{N}\big(w_k|0, \alpha_k^{-1}\big) \tag{7}$$

where α_k denotes the precision of w_k, which is the individual hyperparameter associated with w_k; and $\bar{\boldsymbol{\alpha}}_{i,j} = \begin{bmatrix}\alpha_1 & \ldots & \alpha_K\end{bmatrix}^{\mathrm{T}}$ is a vector containing K hyperparameters controlling the precision of each element of $\bar{\mathbf{w}}_{i,j}$. In the second stage, hyperpriors over $\bar{\boldsymbol{\alpha}}_{i,j}$ are defined. Each precision parameter α_k is modeled as an independent random variable with a Gamma distribution[2] [10], with a common shape parameter a and rate parameter b, as $p(\bar{\boldsymbol{\alpha}}_{i,j}|a,b) = \prod_{k=1}^{K} \mathrm{Gamma}(\alpha_k|a,b)$. By marginalizing over the hyperparameter α_k, we can obtain the true prior over w_k analytically, and it corresponds to the density of a Student-t distribution [10], [15]. Hence, the overall prior over the vector $\bar{\mathbf{w}}_{i,j}$ becomes the product of independent Student-t distributions over w_k. For appropriate choice of a and b, the Student-t distribution is highly peaked around $w_k = 0$, and hence this prior favors most of w_k to be zero promoting sparsity.

Having defined the priors, we proceeds with the Bayesian inference by calculating the posterior density $p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$ by utilizing the following decomposition[3]

$$p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j}) = p(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}, \tilde{\mathbf{y}}_{i,j}) p(\bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j}). \tag{8}$$

The first term in the right-hand-side of (8) represents the (conditional) posterior distribution over the sparse coefficient vector $\bar{\mathbf{w}}_{i,j}$, and it can be expressed as

$$p(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}, \tilde{\mathbf{y}}_{i,j}) = \frac{p(\tilde{\mathbf{y}}_{i,j}|\bar{\mathbf{w}}_{i,j}) p(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j})}{p(\tilde{\mathbf{y}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j})}. \tag{9}$$

Now, by using the measurement model (6), we can evaluate the likelihood function

$$p(\tilde{\mathbf{y}}_{i,j}|\bar{\mathbf{w}}_{i,j}) = \mathcal{N}(\tilde{\mathbf{y}}_{i,j}|\boldsymbol{\eta}_{i,j}, \boldsymbol{\Omega}_{i,j}) \tag{10}$$

with

$$\boldsymbol{\eta}_{i,j} = \breve{\mathbf{B}}_{i,j} \mathbf{E}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j}) + \bar{\mathbf{B}}_{i,j} \bar{\mathbf{w}}_{i,j} \tag{11}$$

$$\boldsymbol{\Omega}_{i,j} = \sigma^2 \mathbf{I}_{4d^2M} + \breve{\mathbf{B}}_{i,j} \mathbf{cov}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j}) \breve{\mathbf{B}}_{i,j}^{\mathrm{T}}. \tag{12}$$

In order to compute the conditional expectation $\mathbf{E}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j})$ and covariance $\mathbf{cov}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j})$, we utilize the preliminary information available on $\tilde{\mathbf{w}}_{i,j} = \begin{bmatrix}\breve{\mathbf{w}}_{i,j}^{\mathrm{T}} & \bar{\mathbf{w}}_{i,j}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, i.e., the current estimation of the probability density of $\tilde{\mathbf{w}}_{i,j}$ denoted as $p(\tilde{\mathbf{w}}_{i,j}|\mathbf{y}^{\mathrm{old}}) = \mathcal{N}\big(\tilde{\mathbf{w}}_{i,j}|\boldsymbol{\mu}_{\tilde{\mathbf{w}}}^{\mathrm{old}}, \boldsymbol{\Sigma}_{\tilde{\mathbf{w}}}^{\mathrm{old}}\big)$. For the notational convenience, let us partition $\boldsymbol{\mu}_{\tilde{\mathbf{w}}}^{\mathrm{old}}$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{w}}}^{\mathrm{old}}$ as

$$\boldsymbol{\mu}_{\tilde{\mathbf{w}}}^{\mathrm{old}} = \begin{bmatrix}\boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b\end{bmatrix}, \quad \boldsymbol{\Sigma}_{\tilde{\mathbf{w}}}^{\mathrm{old}} = \begin{bmatrix}\boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}\end{bmatrix}, \tag{13}$$

where $\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}$ are the mean and covariance of $\breve{\mathbf{w}}_{i,j}$; similarly $\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}$ are the mean and covariance of $\bar{\mathbf{w}}_{i,j}$. Now, with the help of (13), the conditional expectation and covariance of $\breve{\mathbf{w}}_{i,j}$ can be evaluated[4] as [16, pp. 87]

$$\mathbf{E}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j}) = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\big(\boldsymbol{\Sigma}_{bb}\big)^{\dagger}(\bar{\mathbf{w}}_{i,j} - \boldsymbol{\mu}_b), \tag{14}$$

$$\mathbf{cov}(\breve{\mathbf{w}}_{i,j}|\bar{\mathbf{w}}_{i,j}) = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\big(\boldsymbol{\Sigma}_{bb}\big)^{\dagger}\boldsymbol{\Sigma}_{ba}. \tag{15}$$

By substituting (14) and (15) in (11) and (12), we can rewrite the mean $\boldsymbol{\eta}_{i,j}$ and the covariance $\boldsymbol{\Omega}_{i,j}$ of the likelihood function as

$$\boldsymbol{\eta}_{i,j} = \bar{\boldsymbol{\eta}}_{i,j} + \bar{\mathbf{U}}_{i,j} \bar{\mathbf{w}}_{i,j}, \tag{16}$$

$$\boldsymbol{\Omega}_{i,j} = \sigma^2 \mathbf{I} + \breve{\mathbf{B}}_{i,j}\big(\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\big(\boldsymbol{\Sigma}_{bb}\big)^{\dagger}\boldsymbol{\Sigma}_{ba}\big)\breve{\mathbf{B}}_{i,j}^{\mathrm{T}}, \tag{17}$$

where,

$$\bar{\boldsymbol{\eta}}_{i,j} = \breve{\mathbf{B}}_{i,j}\big(\boldsymbol{\mu}_a - \boldsymbol{\Sigma}_{ab}\big(\boldsymbol{\Sigma}_{bb}\big)^{\dagger}\boldsymbol{\mu}_b\big), \tag{18}$$

$$\bar{\mathbf{U}}_{i,j} = \bar{\mathbf{B}}_{i,j} + \breve{\mathbf{B}}_{i,j}\boldsymbol{\Sigma}_{ab}\big(\boldsymbol{\Sigma}_{bb}\big)^{\dagger}. \tag{19}$$

Now, we can evaluate the (conditional) posterior distribution over $\bar{\mathbf{w}}_{i,j}$ for given CS measurements $\tilde{\mathbf{y}}_{i,j}$ in (9) analytically using the Bayes' theorem for Gaussian variables [16, pp.90] by noting that the likelihood function $p(\tilde{\mathbf{y}}_{i,j}|\bar{\mathbf{w}}_{i,j})$ and the prior $p(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j})$ are both Gaussian (see (10) and (7)) as $p\big(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}, \tilde{\mathbf{y}}_{i,j}\big) = \mathcal{N}(\bar{\mathbf{w}}_{i,j}|\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}, \boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}})$ with

$$\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}} = \bar{\mathbf{A}}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\big(\boldsymbol{\Omega}_{i,j} + \bar{\mathbf{U}}_{i,j}\bar{\mathbf{A}}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\big)^{-1}\bar{\mathbf{y}}_{i,j}, \tag{20}$$

$$\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}} = \bar{\mathbf{A}}_{i,j}^{-1} - \bar{\mathbf{A}}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\big(\boldsymbol{\Omega}_{i,j} + \bar{\mathbf{U}}_{i,j}\bar{\mathbf{A}}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\big)^{-1}\bar{\mathbf{U}}_{i,j}\bar{\mathbf{A}}_{i,j}^{-1}, \tag{21}$$

where $\bar{\mathbf{A}}_{i,j} = \mathrm{diag}(\bar{\boldsymbol{\alpha}}_{i,j})$ and $\bar{\mathbf{y}}_{i,j} = \tilde{\mathbf{y}}_{i,j} - \bar{\boldsymbol{\eta}}_{i,j}$.

The second term in the right-hand-side of (8), posterior density of the hyperparameters $p(\bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$, cannot be evaluated analytically. Instead, we utilize the *evidence procedure* [10], [15], [17] and obtain a point estimate for $\bar{\boldsymbol{\alpha}}_{i,j}$ by computing the maximum a-posteriori estimation of $\bar{\boldsymbol{\alpha}}_{i,j}$, i.e., $\bar{\boldsymbol{\alpha}}_{i,j}^{map} = \mathrm{argmax}\ p(\bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$. By noting that, $p(\bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j}) \propto p(\bar{\boldsymbol{\alpha}}_{i,j}, \tilde{\mathbf{y}}_{i,j})$, we estimate the value of $\bar{\boldsymbol{\alpha}}_{i,j}^{map}$ by maximizing the joint distribution $p(\bar{\boldsymbol{\alpha}}_{i,j}, \tilde{\mathbf{y}}_{i,j})$, or equivalently, its logarithm with respect to $\bar{\boldsymbol{\alpha}}_{i,j}$, i.e.,

$$\bar{\boldsymbol{\alpha}}_{i,j}^{map} = \mathrm{argmax}\ \bar{\mathcal{L}}_{i,j}, \tag{22}$$

where

$$\bar{\mathcal{L}}_{i,j} = \log p(\tilde{\mathbf{y}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}) + \log p(\bar{\boldsymbol{\alpha}}_{i,j}), \tag{23}$$

$$= -\frac{1}{2}\log|\bar{\mathbf{C}}_{i,j}| - \frac{1}{2}\bar{\mathbf{y}}_{i,j}^{\mathrm{T}}\bar{\mathbf{C}}_{i,j}^{-1}\bar{\mathbf{y}}_{i,j} + \log p(\bar{\boldsymbol{\alpha}}_{i,j}), \tag{24}$$

with $\bar{\mathbf{C}}_{i,j} = \boldsymbol{\Omega}_{i,j} + \bar{\mathbf{U}}_{i,j}\bar{\mathbf{A}}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}$.

Now, we can obtain the posterior distribution of the sparse vector, i.e., $p(\bar{\mathbf{w}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$, by integrating out the nuisance parameters $\bar{\boldsymbol{\alpha}}_{i,j}$ from $p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$ in (8). However, the posterior density of $\bar{\mathbf{w}}_{i,j}$ can be approximated by $p(\bar{\mathbf{w}}_{i,j}|\tilde{\mathbf{y}}_{i,j}) \approx p(\bar{\mathbf{w}}_{i,j}|\bar{\boldsymbol{\alpha}}_{i,j}^{map}, \tilde{\mathbf{y}}_{i,j})$, since we obtain a point estimate for $\bar{\boldsymbol{\alpha}}_{i,j}$. Thus, we set $p\big(\bar{\mathbf{w}}_{i,j}|\tilde{\mathbf{y}}_{i,j}\big) = \mathcal{N}(\bar{\mathbf{w}}_{i,j}|\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}, \boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}})$ and output $\hat{\mathbf{w}}_{i,j} = \boldsymbol{\mu}_{i,j}^{\mathbf{w}}$ (i.e., the obtained posterior mean for $\mathbf{w}_{i,j}$) as the point estimation of $\mathbf{w}_{i,j}$.

[2]Gamma prior is conjugate to the Gaussian likelihood with a known mean.

[3]Note that evaluating $p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j})$ from the standard decomposition, i.e., $p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}|\tilde{\mathbf{y}}_{i,j}) = p(\tilde{\mathbf{y}}_{i,j}|\bar{\mathbf{w}}_{i,j}) p(\bar{\mathbf{w}}_{i,j}, \bar{\boldsymbol{\alpha}}_{i,j}) / p(\tilde{\mathbf{y}}_{i,j})$, is intractable, since the the normalization factor $p(\tilde{\mathbf{y}}_{i,j})$ cannot be evaluated.

[4]Since $\boldsymbol{\Sigma}_{\tilde{\mathbf{w}}}^{\mathrm{old}}$ is a positive semi-definite matrix, we consider the pseudo inverse of the matrix.

C. Fast SBL algorithm with warm-start

Unfortunately, the value of $\bar{\boldsymbol{\alpha}}_{i,j}^{map}$ in (22) cannot be obtained in closed form. Thus, in this subsection we derive an algorithm, which is fast and computationally efficient, that maximizes $\bar{\mathcal{L}}_{i,j}$ based on the following decomposition:

$$\begin{aligned}\bar{\mathbf{C}}_{i,j} &= \boldsymbol{\Omega}_{i,j} + \sum_{k=1,k\neq l}^{K} \alpha_k^{-1}\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}} + \alpha_l^{-1}\mathbf{u}_l\mathbf{u}_l^{\mathrm{T}}, \\ &= \bar{\mathbf{C}}_{-l} + \alpha_l^{-1}\mathbf{u}_l\mathbf{u}_l^{\mathrm{T}}, \end{aligned} \tag{25}$$

where α_l is the lth element of $\bar{\boldsymbol{\alpha}}_{i,j}$, $\mathbf{u}_l$ is the lth column of $\bar{\mathbf{U}}_{i,j}$ and $\bar{\mathbf{C}}_{-l}$ is $\bar{\mathbf{C}}_{i,j}$ with the contribution of vector $\mathbf{u}_l$ removed.

Substituting (25) in (24) and then equating the derivative of $\bar{\mathcal{L}}_{i,j}$ with respect to α_l to zero, we can show that the maximum of $\bar{\mathcal{L}}_{i,j}$ with respect to α_l, when all the others are fixed, is achieved [18] when

$$\alpha_l = \begin{cases} \dfrac{s_l^2}{q_l^2 - s_l} & \text{if } q_l^2 - s_l > 0 \\ \infty & \text{otherwise}\ , \end{cases} \tag{26}$$

where $s_l = \mathbf{u}_l^{\mathrm{T}}\bar{\mathbf{C}}_{-l}^{-1}\mathbf{u}_l$ and $q_l = \mathbf{u}_l^{\mathrm{T}}\bar{\mathbf{C}}_{-l}^{-1}\bar{\mathbf{y}}_{i,j}$. Note that in the case of $\alpha_l = \infty$, the column $\mathbf{u}_l$ is pruned out from the model and the corresponding $\mu_k^{\bar{\mathbf{w}}}$ is set to zero.

Thus, we can develop an iterative algorithm to estimate $\bar{\boldsymbol{\alpha}}_{i,j}^{map}$, $\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}$ and $\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}$ by noting that (26) provides a systematic method of deciding which columns of $\bar{\mathbf{U}}_{i,j}$ should be included in the model and which should be excluded. In the algorithm, first we choose a candidate $\mathbf{u}_l$ and add/remove it to/from the model or re-estimate the value of α_l following (26)[5]. Then, the quantities $\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}$ and $\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}$ can be computed[6] by (20) and (21) respectively. However, we can utilize the fast update formulae similar to [9] for add, remove and re-estimation operations for a more efficient computation of $\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}$ and $\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}$. Finally, we calculate the quantities s_k and q_k for $k = 1, \ldots, K$. For this purpose, we maintain and update the quantities

$$S_k = \mathbf{u}_k^{\mathrm{T}}\bar{\mathbf{C}}_{i,j}^{-1}\mathbf{u}_k = \mathbf{u}_k^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\mathbf{u}_k - \mathbf{u}_k^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\mathbf{u}_k, \tag{27}$$

$$Q_k = \mathbf{u}_k^{\mathrm{T}}\bar{\mathbf{C}}_{i,j}^{-1}\bar{\mathbf{y}}_{i,j} = \mathbf{u}_k^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\bar{\mathbf{y}}_{i,j} - \mathbf{u}_k^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\bar{\mathbf{U}}_{i,j}\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}\bar{\mathbf{U}}_{i,j}^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\bar{\mathbf{y}}_{i,j}, \tag{28}$$

for all $k = 1, \ldots, K$. Then the new values of s_k and q_k follow from [19]

$$s_k = \frac{\alpha_k S_k}{\alpha_k - S_k}, \qquad q_k = \frac{\alpha_k Q_k}{\alpha_k - S_k}. \tag{29}$$

Using the new values of s_k and q_k, we choose a new candidate $\mathbf{u}_k$ and the hyperparameter α_k is updated according to (26). This iterative process is carried out until the increment in $\bar{\mathcal{L}}_{i,j}$ for each potential update is smaller than a predefined threshold ϵ.

[5]We start with an empty model and $\mathbf{u}_l$ with the largest $\|\mathbf{u}_l^{\mathrm{T}}\boldsymbol{\Omega}_{i,j}^{-1}\bar{\mathbf{y}}_{i,j}\|^2/\|\mathbf{u}_l\|^2$ value is included to the model in the first iteration.

[6]Note that, only the $\alpha_l \neq \infty$ values and corresponding $\mathbf{u}_l$ are used in computing $\boldsymbol{\mu}_{i,j}^{\bar{\mathbf{w}}}$ and $\boldsymbol{\Sigma}_{i,j}^{\bar{\mathbf{w}}}$.

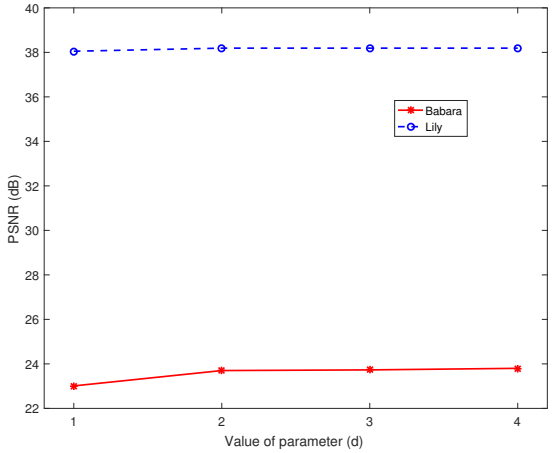

Fig. 3: Average PSNR of the reconstructed image using the SBL algorithm for different values of d ($N = 8$, $\delta = 0.4$, SNR $= 45$ dB).

IV. Numerical Results

First, we evaluate the effect of d (size of the processing window) on the recovery performance of the algorithm. For this purpose, we acquire the 512×512 image *Babara* and 1496×1496 image *Lily* utilizing BCS and reconstructed them using the proposed algorithm, named BCS-SBL. In the acquisition process, we divided the image into $N \times N$ disjoint blocks $\mathbf{X}_{i,j}$ and each block was acquired using the linear measurement model (1), $\mathbf{y}_{i,j} = \boldsymbol{\Phi}_{i,j}\mathbf{x}_{i,j} + \mathbf{e}_{i,j}$, with the same measurement operator $\boldsymbol{\Phi}$ at a measurement rate $\delta = M/N^2$. We added Gaussian noise to the measurements where the measurement noise vector $\mathbf{e}_{i,j}$ follows the distribution $\mathbf{e}_{i,j} \sim \mathcal{N}(0, \sigma^2 I_M)$. The noise variance σ^2 is selected such that the expected SNR with respect to the measurements $\boldsymbol{\Phi}_{i,j}\mathbf{x}_{i,j}$ is ρ dB, i.e., $\rho = 10\log\left(\frac{\|\boldsymbol{\Phi}_{i,j}\mathbf{x}_{i,j}\|_2^2}{M\sigma^2}\right)$.

Fig. 3 presents the quality of the reconstructed image for different values of d in terms of the average peak signal-to-noise ratio (PSNR) in dB. We choose $N = 8$ as the block size, $\delta = 0.4$ as the measurement rate, the sub-sampling operator as the measurement matrix $\boldsymbol{\Phi}$, measurement noise is generated with SNR $= 45$ dB and the stopping threshold of SBL algorithm is set to $\epsilon = 10^{-6}$. We utilize a variable length lapped bi-orthogonal transform (VLLBT) [14], [20], with length $L = 16(2N)$ and 4 long basis functions for the sparse representation of the image. For all the simulations, we performed 5 independent trials for the reconstruction of the image from compressive measurements and the results are averaged over all trials. Results show that the PSNR performance increases with d, as expected, for all images.

Next, we compare the performance of the proposed algorithm with two different recovery methods based on BCS, namely BCS-DCT and BCS-TV. The BCS-DCT, reconstruct each $N \times N$ image block independently from its measurements assuming that the image block is sparse in 2D-DCT. We utilize the standard SBL algorithm in [19] to recover the sparse coefficients in BCS-DCT. In BCS-TV, we utilize the total variation (TV) minimization[7] to reconstruct each $N \times N$ image

[7]We use the MATLAB implementation provided in l_1-magic toolbox for the TV minimization [21].

Algorithm	Measurement rate (δ)			
	0.2	0.3	0.4	0.5
Barbara (512×512)				
BCS-DCT	21.77	23.95	25.42	26.43
BCS-TV	22.75	23.30	23.65	23.95
BCS-SBL	22.73	25.15	26.40	27.20

TABLE I: Comparison of the average PSNR (in dB).

Fig. 4: Reconstruction of Barbara image for $N = 16$, $\delta = 0.2$, SNR $= 45$ dB, $d = 2$. **Top left:** Original image. **Top right:** BCS-DCT. **Bottom left:** BCS-TV. **Bottom right:** BCS-SBL.

block independently from its measurements.

Table I compares PSNR for several images at several measurement rates (δ). We choose $N = 16$ as the block size and the rest of the parameters are chosen same as the previous experiment. A VLLBT with $L = 2N$ containing 8 long basis functions is utilized for the sparse representation of the images. We performed 5 independent trials for the reconstruction of the images from compressive measurements and the results are averaged over all trials. Results show that, the proposed BCS-SBL algorithm clearly outperforms the BCS-DCT algorithm, which only exploits the sparsity of the individual image blocks in the DCT domain. Further, the proposed BCS-SBL algorithm outperforms the BCS-TV algorithm. Even in the case where the proposed method have a close PSNR value compared to BCS-TV, the proposed method provide a more visually pleasing image (see Fig. 4). From Fig. 4, we can see that the reconstructed images based on BCS-DCT and BCS-TV contains blocking artifacts while they have been mitigated in the reconstruction based on the proposed BCS-SBL algorithm. Further, BCS-SBL based reconstruction provide more details such as the stripe patterns on the dress and the table cloth. We believe that, better results can be obtained for the proposed BCS-SBL algorithm by utilizing more complex LTs, such as the once exploits the directional information of the images [22], [23].

V. Conclusions

We addressed the problem of recovering an image using block compressed sensing. We proposed an image recovery method free of post processing, where the blocking artifacts are mitigated by utilizing a lapped transform for the sparse representation of the image. The proposed iterative algorithm is based on sparse Bayesian learning and utilize a Kalman-like implementation.

References

[1] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[3] L. Gan, "Block compressed sensing of natural images," in *International Conference on Digital Signal Processing*, Jul. 2007, pp. 403–406.

[4] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. IEEE Int. Conf. on Image Proc.*, Nov. 2009, pp. 3021–3024.

[5] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends in Signal Processing*, vol. 4, no. 4, pp. 297–416, 2010.

[6] H. T. Kung and S. J. Tarsa, "Partitioned compressive sensing with neighbor-weighted decoding," in *Proc. IEEE Military Commun. Conf.*, Nov. 2011, pp. 149–156.

[7] M. S. Asif and J. Romberg, "Sparse recovery of streaming signals using ℓ_1-homotopy," *IEEE Trans. Signal Processing*, vol. 62, no. 16, pp. 4209–4223, Aug 2014.

[8] U. L. Wijewardhana and M. Codreanu, "Streaming signal recovery using sparse bayesian learning," in *Proc. Annual Asilomar Conf. Signals, Syst., Comp.*, Nov 2014, pp. 1225–1230.

[9] ——, "A bayesian approach for online recovery of streaming signals from compressive measurements," *IEEE Trans. Signal Processing*, vol. 65, no. 1, pp. 184–199, Jan. 2017.

[10] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. 1, pp. 211 – 244, Jun. 2001.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. New York, NY, USA: Academic Press, 1999.

[12] R. L. de Queiroz and T. D. Tran, "Lapped transforms for image compression," in *The Handbook on Transforms and Data Compression*, K. R. Rao and P. C. Yip, Eds. CRC Press, Inc., 2000.

[13] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA, USA: Artech House, Inc., 1992.

[14] T. Tanaka, "Lapped transforms and their applications in image processing," Ph.D. dissertation, Tokyo Institute of Technology, Tokyo, Japan, 2002.

[15] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[17] M. E. Tipping, "Bayesian inference: An introduction to principles and practice in machine learning," in *Advanced Lectures on Machine Learning*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Springer Berlin Heidelberg, 2004, pp. 41–62.

[18] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Adv. Neural Inform. Processing Syst.* MIT Press, 2002, pp. 383–389.

[19] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. Int. Workshop on AI and Stat.*, 2003, pp. 3–6.

[20] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 1043–1053, Apr. 1998.

[21] E. Candès and J. Romberg, "Practical signal recovery from random projections," Jan. 2005.

[22] J. Xu, F. Wu, J. Liang, and W. Zhang, "Directional lapped transforms for image coding," *IEEE Trans. Image Processing*, vol. 19, no. 1, pp. 85–97, Jan. 2010.

[23] S. Muramatsu, D. Han, T. Kobayashi, and H. Kikuchi, "Directional lapped orthogonal transform: theory and design," *IEEE Trans. Image Processing*, vol. 21, no. 5, pp. 2434–2448, May 2012.