Exploiting Structure in Wavelet-Based Bayesian Compressive Sensing

Lihan He and Lawrence Carin
Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708-0291 USA
{lihan,lcarin}@ece.duke.edu

EDICS: DSP-RECO, MAL-BAYL

Abstract

Bayesian compressive sensing (CS) is considered for signals and images that are sparse in a wavelet basis. The statistical structure of the wavelet coefficients is exploited explicitly in the proposed model, and therefore this framework goes beyond simply assuming that the data are compressible in a wavelet basis. The structure exploited within the wavelet coefficients is consistent with that used in wavelet-based compression algorithms. A hierarchical Bayesian model is constituted, with efficient inference via Markov chain Monte Carlo (MCMC) sampling. The algorithm is fully developed and demonstrated using several natural images, with performance comparisons to many state-of-the-art compressive-sensing inversion algorithms.

Index Terms

Bayesian signal processing, wavelets, sparseness, compression

I. Introduction

Over the last two decades there has been significant research directed toward development of transform codes, with the discrete-cosine and wavelet transforms [1] constituting two important examples. The discrete cosine transform (DCT) is employed in the JPEG standard [2], with wavelets employed in the JPEG2000 standard [3]. Wavelet-based transform coding [4] explicitly exploits the structure [5] manifested in the wavelet coefficients of typical data. Specifically, for most natural data (signals and images) the wavelet coefficients are compressible, implying

that a large fraction of the coefficients may be set to zero with minimal impact on the signal-reconstruction accuracy.

A discrete wavelet transform may be implemented via a series of high- and low-pass filters, with decimation performed after each such filtering [1]. This naturally yields a quadtree structure of the wavelet coefficients for an image [1], with each wavelet coefficient generally serving as a "parent" for four "children" coefficients. The wavelet coefficients at the coarsest scale serve as "root nodes" for the quadtrees, with the finest scale of coefficients constituting the "leaf nodes". For most natural images the negligible wavelet coefficients tend to be clustered together; specifically, if a wavelet coefficient at a particular scale is negligible, then its children are also generally (but not always) negligible. This leads to the concept of "zero trees" [4] in which a tree or subtree of wavelet coefficients are all collectively negligible. The structure of the wavelet coefficients, and specifically zero trees, are at the heart of most wavelet-based compression algorithms, and specifically JPEG2000.

Transform coding, particularly JPEG and JPEG2000, are now widely used in digital media. One observes, however, that after the digital data are measured and then transform compressed, one often "throws away" a large fraction of the transform coefficients, while still achieving accurate data reconstruction. This seems wasteful, since there are many applications for which data collection is expensive. For example, the collection of magnetic-resonance imagery (MRI) is time consuming and often uncomfortable for the patient, and hyperspectral cameras require measurement of images at a large number of spectral bands. Since collection of such data is expensive, and because after transform encoding a large fraction of the data are ultimately discarded in some sense, this suggests the following question: Is it possible to measure the informative part of the data directly, thereby reducing measurement costs, while still retaining all of the informative parts of the data? This goal has spawned the new field of compressive sensing (or compressed sensing) [6], [7], [8], in which it has been demonstrated that if the signal of interest is sparse in some basis, then with a relatively small number of appropriately designed projection measurements the underlying signal may be recovered *exactly*. If the data are compressible but not exactly sparse in a particular basis (many coefficients are negligibly small, but not exactly zero), one may still employ compressive sensing (CS) to recover the data up to an error proportional to the energy in the negligible coefficients [9]. Two of the early important applications of CS are in MRI [10] and in development of new hyperspectral cameras

[11].

Details on how to design the compressive-sensing projection vectors, and requirements on the (typically relatively small) number of such projections, may be found in [6], [7], [8], [9]. Assume that the set of N CS projection measurements are represented by the vector $\boldsymbol{v}$, and that these measurements may be represented as $\boldsymbol{v} = \boldsymbol{\Phi\theta}$, where $\boldsymbol{\theta}$ represents an $M \times 1$ vector of transform coefficients, and $\boldsymbol{\Phi}$ is an $N \times M$ matrix constituted via the compressive-sensing measurements; in CS it is desired that $N \ll M$. Given $\boldsymbol{\theta}$, one may recover the desired underlying signal via an inverse transform ($e.g.$, an inverse DCT or wavelet transform, depending on which basis is employed, in which $\boldsymbol{\theta}$ is sparse or compressible). Note that in CS we do not measure $\boldsymbol{\theta}$ directly, but rather projections on $\boldsymbol{\theta}$. One must infer $\boldsymbol{\theta}$ from $\boldsymbol{v}$, this generally an ill-posed inverse problem because $N < M$. To address this problem, almost all practical CS inversion algorithms seek to solve for $\boldsymbol{\theta}$ with the following ℓ_1-regularized optimization problem:

$$\boldsymbol{\theta} = \underset{\tilde{\boldsymbol{\theta}}}{argmin} \|\tilde{\boldsymbol{\theta}}\|_{\ell_1} \quad s.t. \quad \boldsymbol{v} = \boldsymbol{\Phi}\tilde{\boldsymbol{\theta}}. \tag{1}$$

If $\boldsymbol{\theta}$ is sparse, with S non-zero coefficients ($S \ll M$), then CS theory indicates that if $\boldsymbol{\Phi}$ is constructed properly (more on such constructions in Section III) then with "overwhelming probability" [6], [7], [8], [9] one may recover $\boldsymbol{\theta}$ *exactly* if $N > O(S \cdot log(M/S))$; similar relationships hold when $\boldsymbol{\theta}$ is compressible but not exactly sparse.

The aforementioned ℓ_1 inversion may be viewed as a maximum *a posteriori* estimate for $\boldsymbol{\theta}$ under the assumption that each component of $\boldsymbol{\theta}$ is drawn i.i.d. from a Laplace prior [12]. This i.i.d. assumption also implies that (1) assumes that the S non-zero or important coefficients may exist among any of the M components in $\boldsymbol{\theta}$. While this leads to development of many algorithms for CS inversion (see [12], [13], [14], [15], [16], among many others), such a formulation does not exploit all of the prior information available about the transform coefficients $\boldsymbol{\theta}$. For example, as discussed above with respect to the wavelet transform, there is anticipated structure in $\boldsymbol{\theta}$ that may be exploited to further constrain or regularize the inversion, ideally reducing the number of required CS measurements N. This concept has been made rigorous recently for sparse $\boldsymbol{\theta}$ [17], as well as for compressible $\boldsymbol{\theta}$ [18]; these papers demonstrate that one may achieve accurate CS inversions with substantially fewer projection measurements (smaller N) if known properties of the structure of $\boldsymbol{\theta}$ are exploited properly.

As indicated above, if a signal is compressible in a wavelet basis, typically $\boldsymbol{\theta}$ will be charac-

terized by structure within the quadtrees, that may be exploited when performing CS inversion. In [18] the authors leveraged the structure of the wavelet transform to substantially improve the quality of CS inversions, also providing important theoretical foundations for the required number of measurements N. The work reported here is different from that in [18] in the following respect. In [18] the authors "hard-code" a set of models for the wavelet coefficients, exploiting the aforementioned properties of how negligible wavelet coefficients are typically distributed in quadtrees; this yields a single deterministic estimate for the underlying transform coefficients $\boldsymbol{\theta}$, and hence for the underlying signal. In this paper we address this problem from a statistical setting, building upon the Bayesian CS framework introduced in [12]. The structure in the wavelet coefficients is imposed within a Bayesian prior, and the analysis yields a full posterior density function on the wavelet coefficients. Consequently, in addition to estimating the underlying $\boldsymbol{\theta}$, we also provide "error bars" which provide a measure of confidence in the inversion. Such error bars are useful for at least two reasons: (i) when inference is performed subsequently on $\boldsymbol{\theta}$, one may be able to place that inference within the context of the confidence in the CS inversion; and (ii) typically one may not know *a priori* how many transform coefficients are important in a signal of interest, and therefore one will generally not know in advance the proper number of CS measurements N – one may use the error bars on the inversion to infer when enough CS measurements have been performed to achieve a desired accuracy.

The remainder of the paper is organized as follows. In Section II we review the structure inherent to wavelet coefficients in natural images, and in Section III we describe how this structure may be exploited in a Bayesian CS inversion framework. Example results are presented in Section IV, with comparisons to many of the state-of-the-art CS algorithms, which are based primarily on (1). Conclusions and discussions of future work are provided in Section V.

II. Wavelet Tree Structure

The discrete wavelet transform may be represented in matrix form as [1]

$$\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\theta} \tag{2}$$

where $\boldsymbol{x}$ is an $M \times 1$ real vector of data, $\boldsymbol{\Psi}$ is an $M \times M$ matrix with columns corresponding to orthonormal scaling and wavelet basis vectors, and $\boldsymbol{\theta}$ represents the $M \times 1$ vector of wavelet-transform coefficients. The wavelet coefficients that constitute $\boldsymbol{\theta}$ may be represented in terms of

a tree structure, as depicted in Figure 1 for an image. The coefficients at scale $s = 1$ correspond to "root nodes", and the coefficients at the largest scale $s = L$ ($L = 3$ in Figure 1) correspond to "leaf nodes"; the top-left block in Figure 1 corresponds to the scaling coefficients, denoted as $s = 0$, which capture the coarse-scale representation of the image. Each wavelet coefficient at scales $1 \leq s \leq L - 1$ has four "children" coefficients at corresponding scale $s + 1$, and it is the statistical relationships between the parent and children coefficients that is exploited in the proposed CS inversion model.

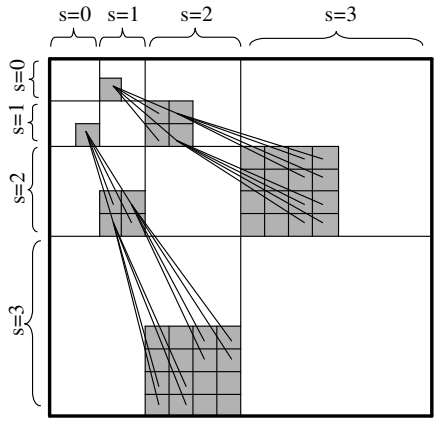


Fig. 1. Wavelet decomposition of an image, with the tree structure depicted across scales. The wavelet transform is performed with three wavelet decomposition levels, and two wavelet trees are shown in the figure. The top-left block ($s = 0$) represents scaling coefficients, and other regions are wavelet coefficients.

The statistics of the wavelet coefficients may be represented via the hidden Markov tree (HMT), in which the structure of the wavelet tree is exploited explicitly. In an HMT model [5] each wavelet coefficient is assumed to be drawn from one of two zero-mean Gaussian distributions, these distributions defining the observation statistics for two hidden states. One of the states is a "low" state, defined by a small Gaussian variance, and the "high" state is defined by a large variance. Intuitively, if a wavelet coefficient is relatively small it is more likely to reside in the "low" state; by contrast, a large wavelet coefficient has a high probability of coming from the "high" state. The probability of a given state is conditioned on the state of the parent coefficient, yielding a Markov representation across scales. The Markov transition property is represented by a 2×2 matrix P, with $P(i, j)$ representing the probability that children coefficients are in state j given that the associated parent coefficient is in state i; $i = 1$ and $j = 1$ (arbitrarily)

correspond to the "low" state, and $i = 2$ and $j = 2$ correspond to the "high" state. Typically $P(1,1) = 1-\epsilon$ and $P(1,2) = \epsilon$, where $\epsilon > 0$ satisfies $\epsilon \ll 1$. This form of P imposes the belief that if a parent coefficient is small, its children are also likely to be small. We also note that P generally varies between different scales and across different wavelet quadtrees. For the root nodes, P is a 1×2 vector, representing an initial-state distribution.

When modeling the statistics of wavelet coefficients for a given signal, the observation is the wavelet coefficient, and for each of the two states the observation is drawn from a zero-mean Gaussian with associated variance (small variance for the "low" state and a relatively large variance for the "high" state). In compressive sensing we do not observe the wavelet coefficients directly, but rather observe projections of these coefficients. The form of the HMT will be employed within the compressive-sensing inversion, thereby explicitly imposing the belief that if a given coefficient is negligible, then its children coefficients are likely to be so as well. This imposes important structure into the form of the wavelet coefficients across scales, and it is consistent with state-of-the-art wavelet-based compression algorithms that are based upon "zero trees" (subtrees of wavelet coefficients that may all be set to zero with negligible effect on the reconstruction accuracy) [4], [19]. The motivation for the HMT construct is discussed in detail in [5].

III. Tree-Structured Wavelet Compressive Sensing

A. Compressive Sensing with Wavelet-Transform Coefficients

Assume a discrete signal/image is represented by the M-dimensional vector $\boldsymbol{x}$, and that it is compressible in a wavelet basis represented by the $M \times M$ matrix $\boldsymbol{\Psi}$ (defined as above). The CS measurements $\boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{\Psi}^T\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\theta}$, where $\boldsymbol{\Phi}$ is an $N \times M$ dimensional matrix ($N < M$), and $\boldsymbol{\theta}$ denotes an M-dimensional vector of wavelet-transform coefficients ($\boldsymbol{\theta} = \boldsymbol{\Psi}^T\boldsymbol{x}$). The rows of $\boldsymbol{\Phi}$ correspond to randomly defined projection vectors [7], [8]. For most natural signals $\boldsymbol{\theta}$ is compressible, meaning that a large fraction of the coefficients in $\boldsymbol{\theta}$ may be set to zero with minimal impact on the reconstruction of $\boldsymbol{x}$; this compressibility property makes it possible to infer $\boldsymbol{\theta}$ based on a small number of projection measurements, assuming that $\boldsymbol{\Phi}$ is designed properly. The theoretical justification for compressive sensing, for design of $\boldsymbol{\Phi}$, and for defining an appropriate N for a given M may be found in [7], [8].

Assume only m transform coefficients in $\boldsymbol{\theta}$ are significant, and the other $M-m$ coefficients are negligibly small. We rewrite $\boldsymbol{\theta} = \boldsymbol{\theta}_m + \boldsymbol{\theta}_e$, where $\boldsymbol{\theta}_m$ represents the original $\boldsymbol{\theta}$ with the $M-m$ smallest coefficients set to zero, and $\boldsymbol{\theta}_e$ represents $\boldsymbol{\theta}$ with the largest m coefficients set to zero. We therefore have

$$\boldsymbol{v} = \boldsymbol{\Phi\theta} = \boldsymbol{\Phi\theta}_m + \boldsymbol{\Phi\theta}_e = \boldsymbol{\Phi\theta}_m + \boldsymbol{n}_e, \tag{3}$$

where $\boldsymbol{n}_e = \boldsymbol{\Phi\theta}_e$. According to Section II, each element of $\boldsymbol{\theta}_e$ can be modeled by a zero-mean Gaussian with small variance (as being drawn from a "low" state), and thus each element of $\boldsymbol{n}_e$, which is a linear combination of elements in $\boldsymbol{\theta}_e$, can also be modeled by a zero-mean Gaussian with appropriate variance. Further, if we also assume the CS measurements are noisy, with zero-mean Gaussian noise $\boldsymbol{n}_0$, we have

$$\boldsymbol{v} = \boldsymbol{\Phi\theta}_m + \boldsymbol{n}_e + \boldsymbol{n}_0 = \boldsymbol{\Phi\theta}_m + \boldsymbol{n}, \tag{4}$$

where the elements of $\boldsymbol{n}$ can be represented by a zero-mean Gaussian noise with unknown variance σ^2, or unknown precision $\alpha_n = \sigma^{-2}$ (to be inferred in the CS inversion).

For the wavelet-based CS reconstruction problem, given measurements $\boldsymbol{v}$ and the random projection matrix $\boldsymbol{\Phi}$, the objective is to estimate the values and the locations of the nonzero elements in the transform coefficients $\boldsymbol{\theta}_m$. For simplicity we henceforth use $\boldsymbol{\theta}$ to replace $\boldsymbol{\theta}_m$ in (4), with the understanding that $\boldsymbol{\theta}$ is now sparse (a large fraction of coefficients are exactly zero).

B. Tree-Structured Wavelet CS Model

Baraniuk *et al.* [18] demonstrate that it is possible to improve compressive-sensing reconstruction performance by leveraging signal models (structure within the transform coefficients), by introducing dependencies between values and locations of the signal coefficients. Two greedy CS algorithms, CoSaMP [20] and iterative hard thresholding (IHT) [21], are implemented in [18], with the wavelet tree structure incorporated into the inversion models.

In this paper the proposed tree-structured wavelet compressive sensing (TSW-CS) model is constructed in a hierarchical Bayesian learning framework. In this setting we infer a full posterior density function on the wavelet coefficients, and therefore we may quantify our confidence in the inversion ($e.g.$, the variance about the mean inverted signal). Within the Bayesian framework

we impose a prior belief for the model parameters, represented in terms of prior distributions on the model parameters. The posterior distribution for all model parameters and for the wavelet coefficients are inferred based on the observed data $\boldsymbol{v}$. The structural information embodied by the wavelet tree (the parent-children relationship and the propagation of small coefficients across scales) is incorporated in the prior, and is therefore imposed statistically.

We utilize a spike-and-slab prior, which has been used recently in Bayesian regression and factor models [22], [23], [24], [25], [26]. The prior for the ith element of $\boldsymbol{\theta}$ (corresponding to the ith transform coefficient) has the form

$$\theta_i \sim (1-\pi_i)\delta_0 + \pi_i \mathcal{N}(0, \alpha_i^{-1}), \quad i = 1, 2, ...M, \tag{5}$$

which is a mixture of two components. The first component δ_0 is a point mass concentrated at zero, and the second component is a zero-mean Gaussian distribution with (relatively small) precision α_i; the former represents the zero coefficients in $\boldsymbol{\theta}$ and the latter the non-zero coefficients. This is a two-component mixture model, and the two components are associated with the two states in the HMT. Related models of this type have been employed previously for wavelet-based clustering [27]. The form of this model is different from an HMT [5] in that the coefficient associated with the "low" state is now explicitly set to zero, such that the inferred wavelet coefficients are explicitly sparse (many coefficients exactly zero). However, like in the HMT, we impose the belief that if a parent coefficient is zero, its children coefficients are likely to also be zero. To achieve this goal, the key to the model is imposition of dependencies in the π_i across scales, in the form discussed above.

The mixing weight π_i, the precision parameter α_i, as well as the unknown noise precision α_n, are learned from the data. The proposed Bayesian tree-structured wavelet (TSW) CS model is

summarized as follows:

$$\begin{aligned}
\boldsymbol{v}|\boldsymbol{\theta}, \alpha_n &\sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{\theta}, \alpha_n^{-1}\boldsymbol{I}), && \text{(6a)}\\
\theta_{s,i} &\sim (1-\pi_{s,i})\delta_0 + \pi_{s,i}\mathcal{N}(0, \alpha_s^{-1}), \text{ with } \pi_{s,i} = \begin{cases} \pi_r, \text{ if } s=1, \\ \pi_s^0, \text{ if } 2\le s\le L, \theta_{pa(s,i)}=0, \\ \pi_s^1, \text{ if } 2\le s\le L, \theta_{pa(s,i)}\neq 0, \end{cases} && \text{(6b)}\\
\alpha_n &\sim \text{Gamma}(a_0, b_0), && \text{(6c)}\\
\alpha_s &\sim \text{Gamma}(c_0, d_0), \quad s=1,...,L, && \text{(6d)}\\
\pi_r &\sim \text{Beta}(e_0^r, f_0^r), && \text{(6e)}\\
\pi_s^0 &\sim \text{Beta}(e_0^{s0}, f_0^{s0}), \quad s=2,...,L, && \text{(6f)}\\
\pi_s^1 &\sim \text{Beta}(e_0^{s1}, f_0^{s1}), \quad s=2,...,L, && \text{(6g)}
\end{aligned}$$

where $\theta_{s,i}$ denotes the ith wavelet coefficient (corresponding to the spatial location) at scale s, for $i = 1, ..., M_s$ (M_s is the total number of wavelet coefficients at scale s), $\pi_{s,i}$ is the associated mixing weight, and $\theta_{pa(s,i)}$ denotes the parent coefficient of $\theta_{s,i}$. In (6b) it is assumed that all the nonzero coefficients at scale s share a common precision parameter α_s. It is also assumed that all the coefficients at scale s with a zero-valued parent share a common mixing weight π_s^0, and the coefficients at scale s with a nonzero parent share a mixing weight π_s^1. We may also let each coefficient maintain its own $\pi_{s,i}$, but we found from the experiments that the performance is very similar to that from the model presented in (6) (sharing common π_s^0 and π_s^1 for each scale), and (6) is much simpler because there are less parameters in the model. Gamma priors are placed on the noise precision parameter α_n and the nonzero coefficient precision parameter α_s, and the posteriors of these precisions are inferred according to the data. The mixing weights π_r, π_s^0 and π_s^1 are also inferred, by placing Beta priors on them. To impose the structural information, depending on the scale and the parent value of the coefficients, different Beta priors are placed. For the coefficients at the root node, a prior preferring a value close to one is set in (6e), because at the low-resolution level many wavelet coefficients are nonzero; for the coefficients with a zero-valued parent, a prior preferring zero is considered in (6f), to represent the propagation of zero coefficients across scales; finally, (6g) is for the coefficients with a nonzero parent, and hence no particular preference is considered since zero or nonzero values are both possible (the

hyperparameters e_0^{s1} and f_0^{s1} impose a uniform prior on π_s^1). Note that the model presented in (6) does not include the scaling coefficients (coefficients at scale $s = 0$); in Section III-D we extend the model to also estimate the scaling coefficients.

The prior imposed in (6f) implies that if a parent node is zero, with high (but not unity) probability its children coefficients will also be zero. The form of the model reduces the degrees of freedom *statistically* in the solution space $\mathcal{R}^M$, since we impose the belief that particular forms of wavelet coefficients are more probable. As opposed to the work in [18], we do not make an explicit ("hard") imposition of the structure of the coefficients, but rather impose the structure statistically.

The desired structural information is naturally integrated into the proposed TSW-CS model. It can be seen that the two components in the spike-and-slab prior are analogous to the two states in the HMT model, and the zero-mean Gaussian distributions are analogous to the observation functions of the HMT. The transition-probability matrix P at scale s ($s > 1$) in the HMT is now represented by the mixing weights π_s^0 and π_s^1, with $P(1,1) = 1 - \pi_s^0, P(1,2) = \pi_s^0, P(2,1) = 1 - \pi_s^1$, and $P(2,2) = \pi_s^1$ (the initial-state distribution is represented by $[1 - \pi_r,\ \pi_r]$). Note that π_s^0 and π_s^1 represent the summary of the overall (Markovian) statistical properties for all the wavelet coefficients at scale s, while for each particular coefficient $\theta_{s,i}$, an associated posterior of mixing weight, $\tilde{\pi}_{s,i}$, will be inferred (see Section III-C for the inference).

We also note that the HMT has recently been employed explicitly within CS inversion, for wavelet-based CS [28]. In that previous work one must first train an HMT model on representative example data, and then that model is used within the CS inversion. The difficulty of such an approach is that one must have access to training data that is known *a priori* to be appropriate for the CS data under test. By contrast, in the proposed inference engine, in addition to inferring a posterior distribution on the wavelet coefficients, posterior distributions are jointly inferred on the underlying model parameters as well. There is therefore no need for *a priori* training data. In this sense the proposed method infers the wavelet coefficients *and* a statistical model for these coefficients, with the model consistent with the expected statistical structure typically inherent to the wavelet transform.

C. MCMC Inference

We implement the posterior computation by an Markov chain Monte Carlo (MCMC) method [29] based on Gibbs sampling, where the posterior distribution is approximated by a sufficient number of samples. These samples are collected by iteratively drawing each random variable (model parameters and intermediate variables) from its conditional posterior distribution given the most recent values of all the other random variables. The priors of the random variables are set independently as

$$p(\alpha_n, \{\alpha_s\}_{s=1:L}, \pi_r, \{\pi_s^0, \pi_s^1\}_{s=2:L}) = \\ \text{Gamma}(a_0, b_0) \left\{ \prod_{s=1}^{L} \text{Gamma}(c_0, d_0) \right\} \text{Beta}(e_0^r, f_0^r) \left\{ \prod_{s=2}^{L} \text{Beta}(e_0^{s0}, f_0^{s0}) \text{Beta}(e_0^{s1}, f_0^{s1}) \right\}. \quad (7)$$

Under this setting the priors are conjugate to the likelihoods, and the conditional posteriors used to draw samples can be derived *analytically*. At each MCMC iteration, the samples are drawn from the following conditional posterior distributions:

- $p(\theta_{s,i}|-) = (1 - \tilde{\pi}_{s,i})\delta_0 + \tilde{\pi}_{s,i}\mathcal{N}(\tilde{\mu}_{s,i}, \tilde{\alpha}_{s,i}^{-1})$.

  Assume $\theta_{s,i}$ is the jth element in the M-dimensional vector $\boldsymbol{\theta}$, denoted by $\theta_{(j)}$, then

  $$\tilde{\alpha}_{s,i} = \alpha_s + \alpha_n \Phi_{(j)}^T \Phi_{(j)},$$

  $$\tilde{\mu}_{s,i} = \tilde{\alpha}_{s,i}^{-1} \alpha_n \Phi_{(j)}^T \tilde{\boldsymbol{v}}_{(j)}, \text{ with } \tilde{\boldsymbol{v}}_{(j)} = \boldsymbol{v} - \sum_{\substack{k=1 \\ k \neq j}}^{M} \Phi_{(k)} \theta_{(k)},$$

  $$\frac{\tilde{\pi}_{s,i}}{1 - \tilde{\pi}_{s,i}} = \frac{\pi_{s,i}}{1 - \pi_{s,i}} \frac{\mathcal{N}(0|0, \alpha_s^{-1})}{\mathcal{N}(0|\tilde{\mu}_{s,i}, \tilde{\alpha}_{s,i})},$$

  where $\Phi_{(j)}$ denotes the jth column of the $N \times M$ random projection matrix $\boldsymbol{\Phi}$.
- $p(\alpha_s|-) = \text{Gamma}(c_0 + \frac{1}{2}\sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} \neq 0),\ \ d_0 + \frac{1}{2}\sum_{i=1}^{M_s} \theta_{s,i}^2)$.

  where $\mathbf{1}(y)$ denotes an indicator function such that $\mathbf{1}(y) = 1$ if y is true and 0 otherwise.
- $p(\pi_r|-) = \text{Beta}(e_0^r + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} \neq 0),\ \ f_0^r + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} = 0))$, for $s = 1$.
- $p(\pi_s^0|-) = \text{Beta}(e_0^{s0} + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} \neq 0, \theta_{pa(s,i)} = 0),\ \ f_0^{s0} + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} = 0, \theta_{pa(s,i)} = 0))$, for $2 \leq s \leq L$.
- $p(\pi_s^1|-) = \text{Beta}(e_0^{s1} + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} \neq 0, \theta_{pa(s,i)} \neq 0),\ \ f_0^{s1} + \sum_{i=1}^{M_s} \mathbf{1}(\theta_{s,i} = 0, \theta_{pa(s,i)} \neq 0))$, for $2 \leq s \leq L$.
- $p(\alpha_n|-) = \text{Gamma}(a_0 + \frac{N}{2},\ \ b_0 + \frac{1}{2}(\boldsymbol{v} - \boldsymbol{\Phi\theta})^T(\boldsymbol{v} - \boldsymbol{\Phi\theta}))$.

At each MCMC iteration, $\boldsymbol{\theta}$ can be sampled in a block manner; alternatively, $\theta_{s,i}$ can also be sampled sequentially for all s and i. We observed in our experiments that sequential sampling typically achieves faster convergence, *i.e.*, less iterations are required to achieve MCMC convergence compared to block sampling. This is because in block sampling, computing the conditional posterior of $\theta_{(j)}$ uses all the other elements of $\boldsymbol{\theta}$ ($\theta_{(k)}$ for $k \neq j$) from the last MCMC iteration; however, by sequential sampling, computing the conditional posterior of $\theta_{(j)}$ can use $\theta_{(k)}$ for $k < j$ from the current iteration (updated before $\theta_{(j)}$ in the current iteration) and $\theta_{(k)}$ for $k > j$ from the last iteration. We observed fast convergence of this model for the problems considered; typically a burn-in period of 200 iterations is enough for an image of size 128×128, and the collection period corresponds to 100 samples. It is very unlikely that this small number of MCMC iterations is sufficient to accurately represent the full posterior on all model parameters; however, based on many experiments, the mean wavelet coefficients have been found to provide a good practical CS inversion, and the collection samples also provide useful "error bars" (discussed further below). Figure 2 shows the convergence curve for a 128×128 image *cameraman* with 5000 measurements (see Figure 3(a) for the image); we employed the sequential sampling approach in this example, as well as in all results presented below. The use of approximate (truncated) MCMC sampling is distinct from but related to particle filters, which have recently seen significant utility in signal-processing applications [30].

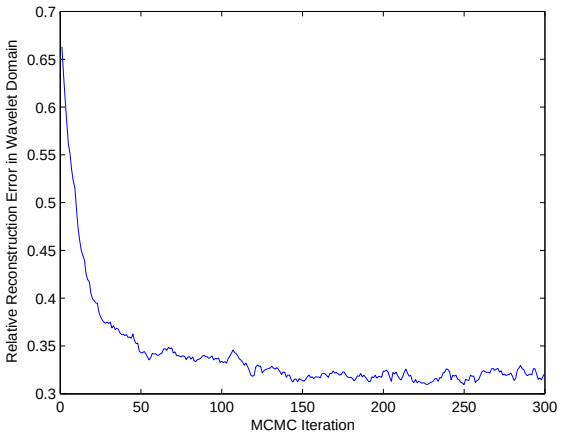


Fig. 2. Example of an MCMC convergence curve for the image *cameraman* of size 128×128; see Figure 3(a) for the image. The vertical axis is evaluated as $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2 / \|\boldsymbol{\theta}\|_2$, where $\hat{\boldsymbol{\theta}}$ denotes the reconstructed wavelet coefficients.

The variational Bayesian (VB) method [31] is often considered for fast but approximate Bayesian inference. However, the VB approach is not appropriate in our model because of the speciality of our mixing components. In the VB inference, when updating variational posterior $q(\boldsymbol{\theta})$, similar to a Gaussian mixture model (GMM), we have difficulty dealing with the logarithm of a summation, $log[\pi_{s,i}\delta_0 + (1-\pi_{s,i})\mathcal{N}(0,\alpha_s^{-1})]$. The VB GMM solves this problem by introducing an indicator variable to denote which component the current datum comes from, so that the summation is removed. But the indicator variable cannot be used here because in our model an indicator variable is redundant with regard to $\theta_{s,i}$. Specifically, assuming $z_{s,i}=0$ indicates $\theta_{s,i}=0$, we then have $p(z_{s,i}=0|\theta_{s,i}=0)=1$, and $p(\theta_{s,i}=0|z_{s,i}=0)=1$. Consequently, $z_{s,i}$ will never be updated for those $\theta_{s,i}=0$, and similarly, $\theta_{s,i}$ will never be updated for those $z_{s,i}=0$.

Given the fast MCMC convergence for the problems considered, with non-optimized programming in Matlab^{TM}, we feel that this may be a practical inference engine for the applications of interest. Specifically, as indicated below, the computational requirements of the TSW-CS model are competitive with many of the existing compressive-sensing inversion algorithms in the literature.

D. Extension with Scaling Coefficients

The TSW-CS model presented in (6) only performs inversion for the wavelet coefficients, assuming that the scaling-function coefficients are measured separately (this has been assumed in many previous compressive-sensing studies [15]). However, it is also of interest in many applications to also infer the scaling coefficients. We may readily extend our TSW-CS model to include reconstruction of the scaling coefficients as follows. Specifically, the model is the same as (6), except with

$$\theta_{s,i} \sim (1-\pi_{s,i})\delta_0 + \pi_{s,i}\mathcal{N}(0,\alpha_s^{-1}), \text{ with } \pi_{s,i} = \begin{cases} \pi_{sc}, \text{ if } s=0, \\ \pi_r, \text{ if } s=1, \\ \pi_s^0, \text{ if } 2\leq s\leq L, \theta_{pa(s,i)}=0, \\ \pi_s^1, \text{ if } 2\leq s\leq L, \theta_{pa(s,i)}\neq 0, \end{cases} \tag{8a}$$

$$\pi_{sc} \sim \text{Beta}(e_0^{sc}, f_0^{sc}). \tag{8b}$$

Compared to (6), the extended model in (8) includes the scale $s=0$ for the scaling coefficients, with an associated mixing weight π_{sc}, which is drawn from a prior distribution Beta(e_0^{sc}, f_0^{sc}).

Considering that the scaling coefficients are usually nonzero, the hyperparameters e_0^{sc} and f_0^{sc} are specified such that $\pi_{sc} = 1$ is almost always true (since π_{sc} is only one more parameter, we perform inference on it, but our experience is that it may be set $\pi_{sc} = 1$ with minimal change on the results, for the examples considered). All scaling coefficients share a common precision parameter α_0, which is learned from the inference.

IV. Experimental Results

We test the performance of the TSW-CS framework on three example images: *cameraman*, *peppers* and *pirate*. The original sizes of the three images are 256×256, 256×256, and 512×512, respectively; these images are available online at `http://decsai.ugr.es/cvg/dbimagenes/`. We compare the performance of TSW-CS to six recently developed CS reconstruction algorithms: basis pursuit (BP) [13], Bayesian compressive sensing (BCS) [12], fast-BCS [12], orthogonal matching pursuit (OMP) [14], stagewise orthogonal matching pursuit (StOMP) [15], and Lasso-modified least angle regression (LARS/Lasso) [16]. For the BP implementation, we use the `l1_eq` solver from the ℓ_1-*Magic* toolbox available at `http://www.acm.caltech.edu/l1magic/`; the fast-BCS solver is available at `http://www.ece.duke.edu/~shji/BCS.html`; for the OMP, StOMP and LARS/Lasso algorithms, we use the solvers `SolveOMP`, `SolveStOMP`, and `SolveLasso`, respectively, from the *SparseLab* toolbox available at `http://sparselab.stanford.edu/`. The BCS algorithm can be implemented via the relevance vector machines (RVM) [12]; we implemented it using a variational RVM [32]. All software are written in MATLAB^{TM}, and run on PCs with 3.6GHz CPU and 4GB memory.

Because of limitations of computation time and memory, it is difficult to reconstruct an image of size 256×256 or larger. To make the performance comparisons practical, we use three formats to reduce the number of the wavelet coefficients to be estimated. (Later, we show the performance comparisons for larger images, with the projection matrix $\mathbf{\Phi}$ expressed implicitly and not stored in the memory; also, the comparisons are only among selected – relatively fast – methods and for a limited number of compressive sensing measurements.) The three formats are:

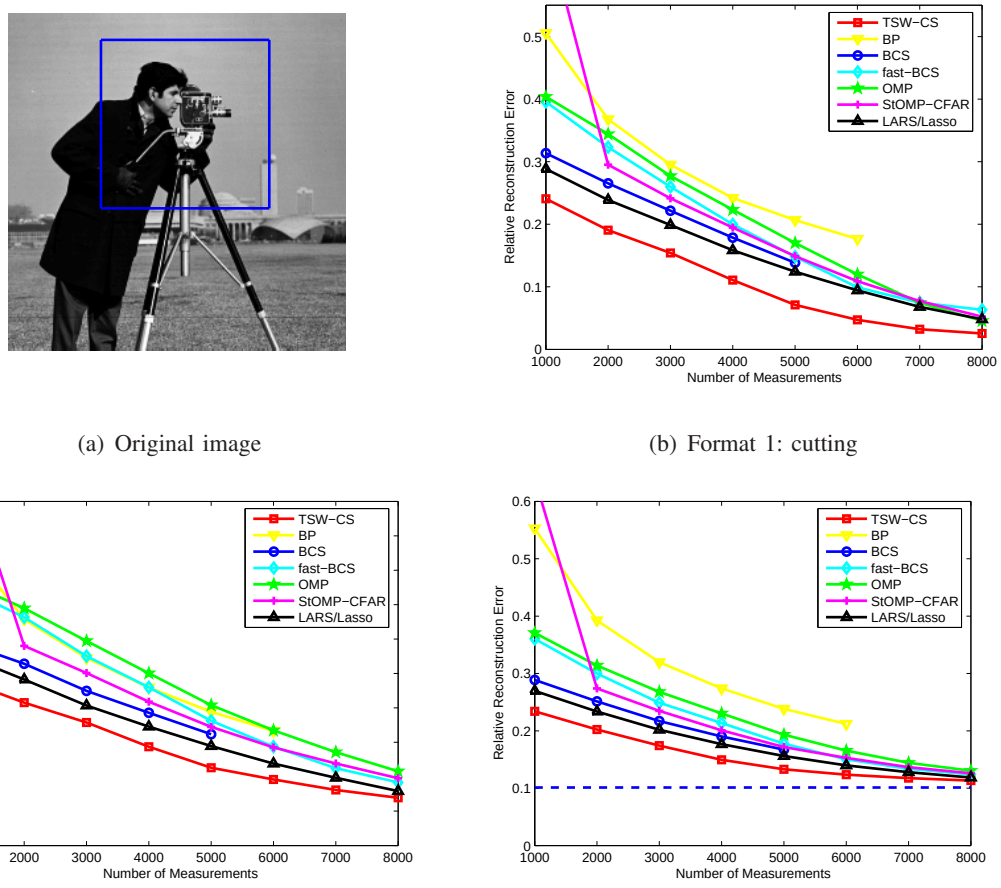
1) cut a patch of size 128×128 from the original image, and reconstruct all the wavelet coefficients (Format 1);
2) resize the original image to 128×128 using MATLAB^{TM} function `imresize`, and reconstruct all the wavelet coefficients (Format 2);

3) using the hybrid CS scheme [33], only estimate wavelet coefficients at relatively lower-resolution scales, and assume the wavelet coefficients at other scales are zero. For the images *cameraman* and *peppers*, which are of size 256×256, we assume the wavelet coefficients at the highest scale, $s = 5$, are zero and not estimated; for the image *pirate*, which is of size 512×512, the wavelet coefficients at the highest two scales, $s = 5$ and $s = 6$, are assumed zero and not estimated (Format 3).

For all three formats, we estimate around $16K$ wavelet coefficients using the TSW-CS model presented in (6); the scaling coefficients constitute a block of size 8×8, and we here assume the scaling coefficients are measured directly (later we will show the example that the scaling coefficients are also inferred based on the extended model in (8)). Our objective is to estimate the wavelet coefficients of size $128^2 - 8^2 = 16320$, based on a given number of CS measurements. We use the Haar wavelet for *cameraman*, since the wavelet coefficients are sparser for the Haar basis than for other wavelet bases, and choose the Daubechies-8 wavelet as our orthonormal basis for *peppers* and *pirate*.

For each CS algorithm, image, and image format, we produce a curve of relative reconstruction error as a function of number of measurements N (the 64 scaling coefficients are not included in N). The relative reconstruction error is defined as $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 / \|\boldsymbol{x}\|_2$, where $\boldsymbol{x}$ is the original image, and $\hat{\boldsymbol{x}}$ is the recovered image based on the wavelet coefficients reconstructed by a particular CS algorithm. The scaling coefficients are assumed measured accurately, and are used during image recovery. Figure 3, 4 and 5 show the performance comparisons for the three images, with three formats for each image, and Figure 6 shows the time comparison for the image *cameraman* with Format 1 (see above). Time comparison for other images/formats are similar to Figure 6. All results for TSW-CS are based on MCMC inference with 200 burn-in iterations followed by 100 MCMC collection samples (we show results for the mean coefficients, but also have access to "error bars", as discussed further below). More burn-in iterations and more samples collected may be considered, but we have found the mean results to be similar to results shown here. The hyperparameters for the priors in the TSW-CS model are as follows: $a_0 = b_0 = c_0 = d_0 = 10^{-6}$, $[e_0^r,\ f_0^r] = [0.9,\ 0.1] \times M_1$, $[e_0^{s0},\ f_0^{s0}] = [\frac{1}{M},\ 1 - \frac{1}{M}] \times M_s$, and $[e_0^{s1},\ f_0^{s1}] = [0.5,\ 0.5] \times M_s$. Note that the form $[u,\ 1-u] \times V$ is used to represent the hyperparameters e_0 and f_0 in the Beta priors, where u represents the prior mean of the mixing weight π, and V represents the confidence of the prior (larger V means more confidence). We

set as $\frac{1}{M}$ the prior probability of the rare event that a child is not zero given that its parent is zero (recall that M is the total number of estimated wavelet coefficients), and use M_s (number of coefficients at scale s) for the confidence so that the strength of the prior is comparable to the likelihood. For the other CS algorithms, default parameters (if required as input arguments) are used. The StOMP algorithm with CFDR thresholding is not stable; consequently, we use the StOMP algorithm with CFAR thresholding, with the false-alarm rate specified as 0.01.



(a) Original image

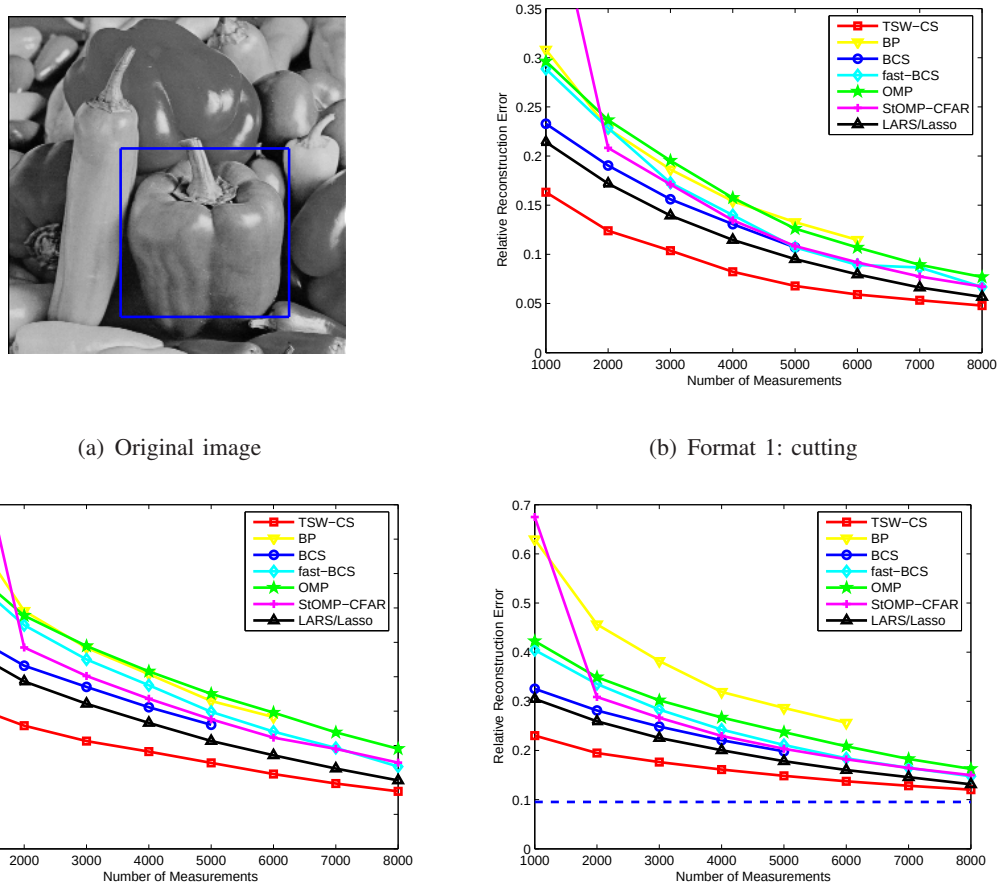
(b) Format 1: cutting

(c) Format 2: resizing

(d) Format 3: hybrid CS

Fig. 3. Performance comparisons for the image *cameraman*. (a) Original image; the square in the image denotes the small patch of size 128×128 used by Format 1. (b) Performance for Format 1. (c) Performance for Format 2. (d) Performance for Format 3. The dashed line denotes the best performance that could be achieved since we assume the wavelet coefficients at scale $s = L$ are zero and not estimated.

Each evaluated point in the curves in Figure 3, 4, 5 and 6 is computed based on the average of five trials. For each trial, a random projection matrix $\mathbf{\Phi}$ is generated, with each entry sampled



(a) Original image

(b) Format 1: cutting

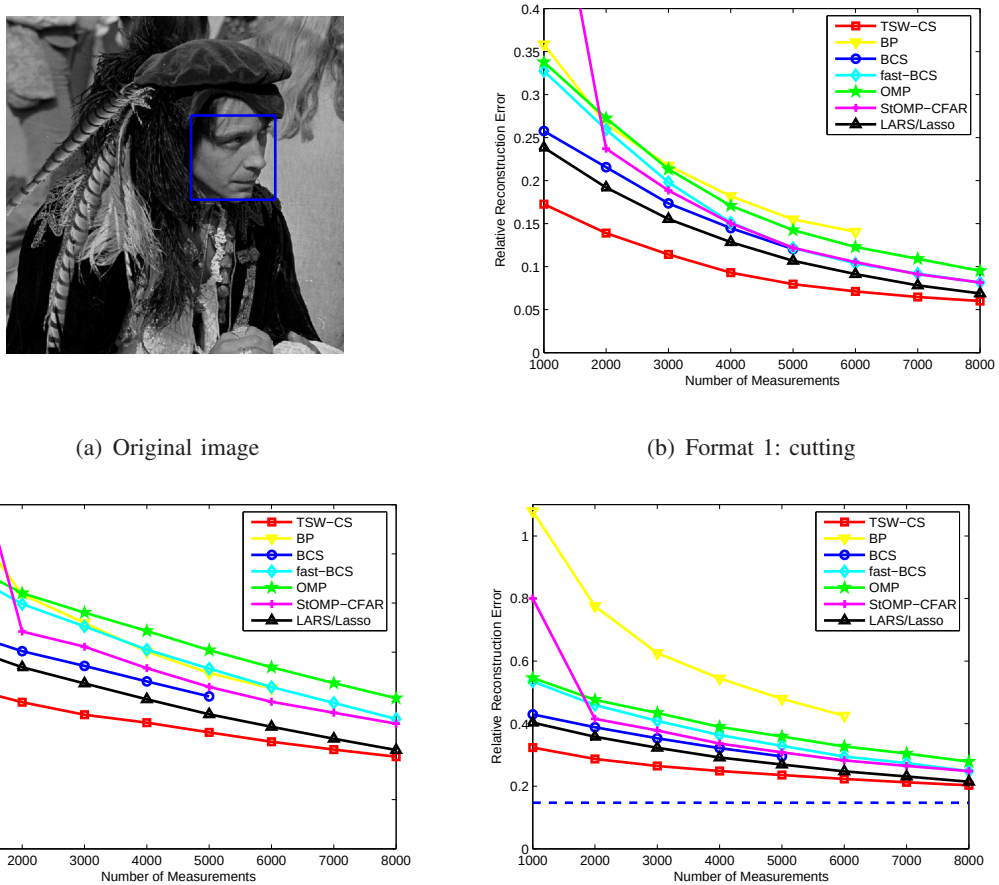
(c) Format 2: resizing

(d) Format 3: hybrid CS

Fig. 4. Performance comparisons for the image *peppers*. (a) Original image; the square in the image denotes the small patch of size 128×128 used by Format 1. (b) Performance for Format 1. (c) Performance for Format 2. (d) Performance for Format 3. The dashed line denotes the best performance that could be achieved since we assume the wavelet coefficients at scale $s = L$ are zero and not estimated.

i.i.d. from $\mathcal{N}(0, 1)$, and normalized along the column to have unit energy. This $\mathbf{\Phi}$ is then shared by all the CS algorithms compared at the current evaluation point. The absence of an evaluated point in a curve (particularly, for BP and BCS) means that the memory requirement at that point exceeds the computer capability.

From these figures, our TSW-CS model consistently outperforms the other algorithms, with a relatively small computational time. Especially, when the number of measurements is very limited, the relative reconstruction error for TSW-CS is consistently much less than that for the other algorithms. In addition, for a given reconstruction error, the TSW-CS model requires



(a) Original image

(b) Format 1: cutting

(c) Format 2: resizing

(d) Format 3: hybrid CS

Fig. 5. Performance comparisons for the image *pirate*. (a) Original image; the square in the image denotes the small patch of size 128×128 used by Format 1. (b) Performance for Format 1. (c) Performance for Format 2. (d) Performance for Format 3. The dashed line denotes the best performance that could be achieved since we assume the wavelet coefficients at scales $s = L - 1, L$ are zero and not estimated.

much less CS measurements than the other algorithms ($e.g.$, in Figure 3(b), to achieve a relative reconstruction error of 0.05, LARS/Lasso requires around 8000 measurements, while TSW-CS requires only around 6000 measurements). Figure 7 presents an example of the reconstructed wavelet coefficients $\hat{\boldsymbol{\theta}}$ for all CS algorithms under comparison, for *cameraman* with Format 1 and 1000 measurements. We see that when the number of measurements is relatively small, the TSW-CS model concentrates more energy on the low-resolution scales, and so estimates the coefficients in the low-resolution bands better. To make this point clearer, Figure 7(b) shows zoom-in plots of the first 960 coefficients at the low-resolution scales $s = 1$ and $s = 2$. When the

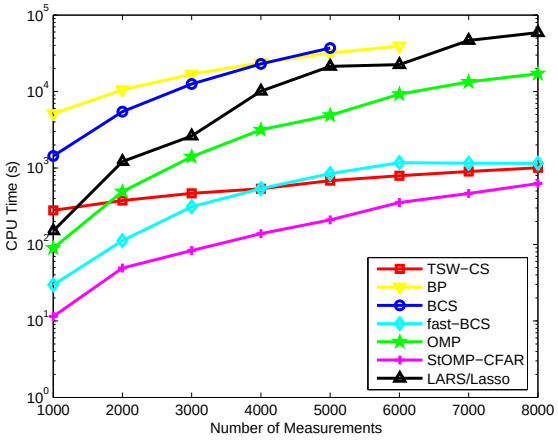
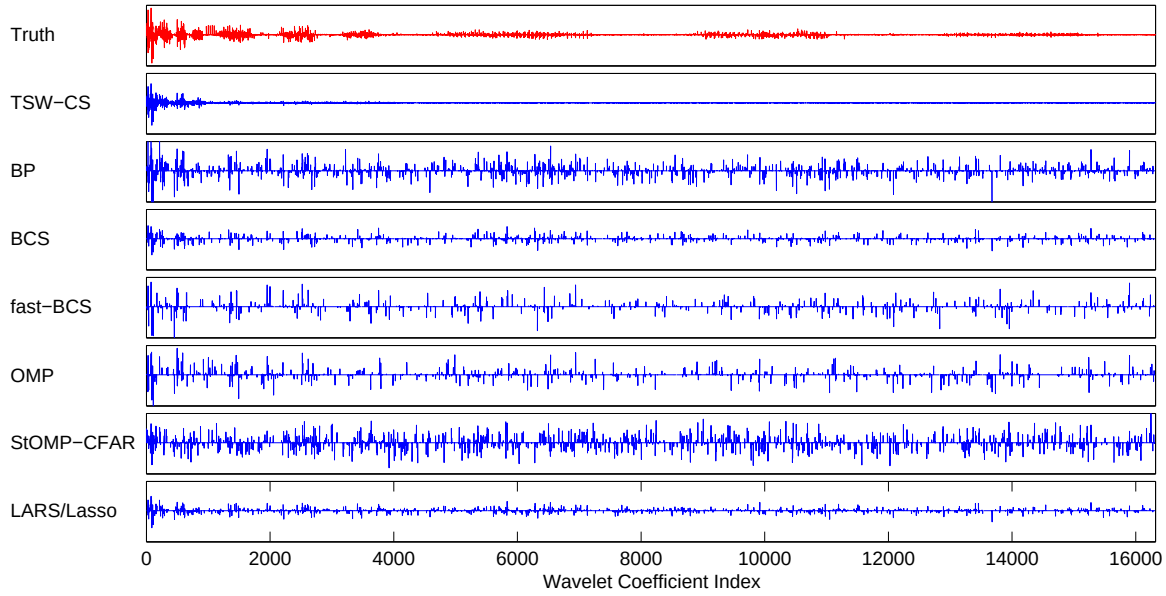


Fig. 6. Comparison of CPU time for the CS algorithms. The results are based on the image *cameraman* of size 128×128 with Format 1.

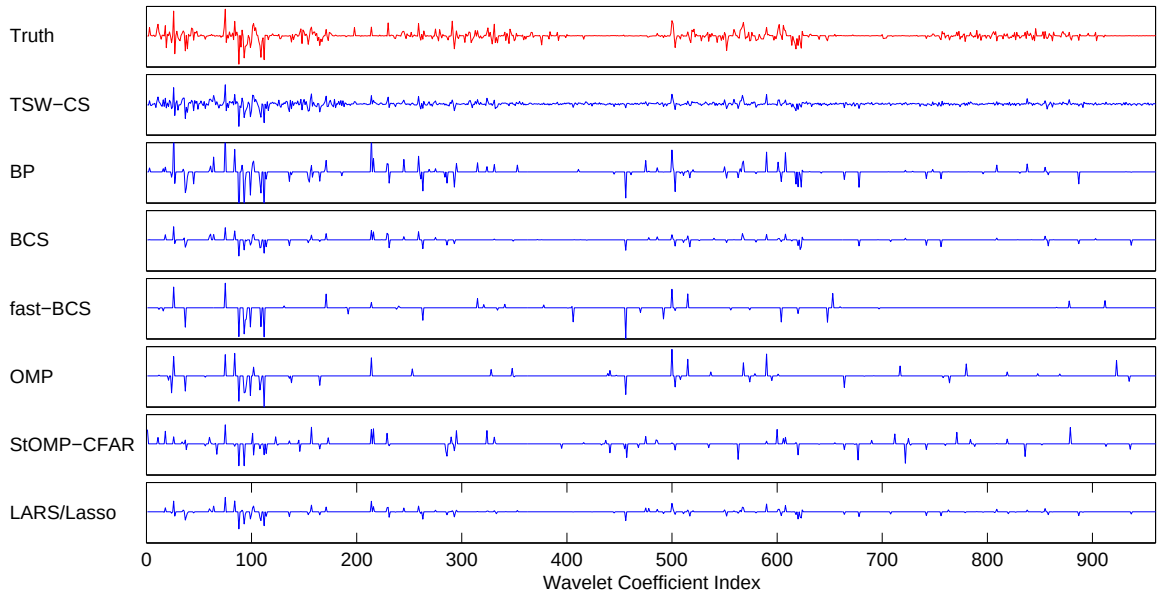
number of measurements increases, details of an image are then revealed. With the δ_0 component and the parent-child relationships in the prior setting, the TSW-CS model provides a much sparser solution, in the sense of less high frequency noise in the reconstruction compared to the other algorithms, and so estimates large coefficients at higher-resolution scales more accurately. Figure 8 shows the comparisons of the reconstructed $\hat{\boldsymbol{\theta}}$ for the situation of more measurements, for *cameraman* with Format 1 and 5000 measurements. It can be seen that in the regions of zero or small coefficients in the "truth" (usually at high-resolution scales), the TSW-CS model infers exact zero values, while the other algorithms often give noisy estimations (small non-zero values). These noisy values impair the accurate estimation of large coefficients at high-resolution scales.

To show the quality of the reconstruction further, Figure 9 also presents examples of the recovered images for the CS algorithms. We see that for a given number of measurements, TSW-CS achieves a better image quality. A careful examination of the TSW-CS relative to the other approaches reveals that the former more accurately renders details, particularly at a relatively small number of measurements (see, for example, results at 4000 measurements, and the ability of TSW-CS to recover details in the "cameraman's" face as well as in the buildings at the lower right portion of the image).

As mentioned above, with the Bayesian learning framework, the TSW-CS infers a posterior



(a) All wavelet coefficients



(b) Wavelet coefficients at scales $s = 1, 2$

Fig. 7. Comparison of the reconstructed wavelet coefficients by the CS algorithms, for the image *cameraman* with Format 1 and 1000 measurements. (a) All the wavelet coefficients, $M = 16320$. (b) A zoom-in version of (a), showing the first 960 wavelet coefficients (*i.e.*, coefficients at scales $s = 1, 2$).

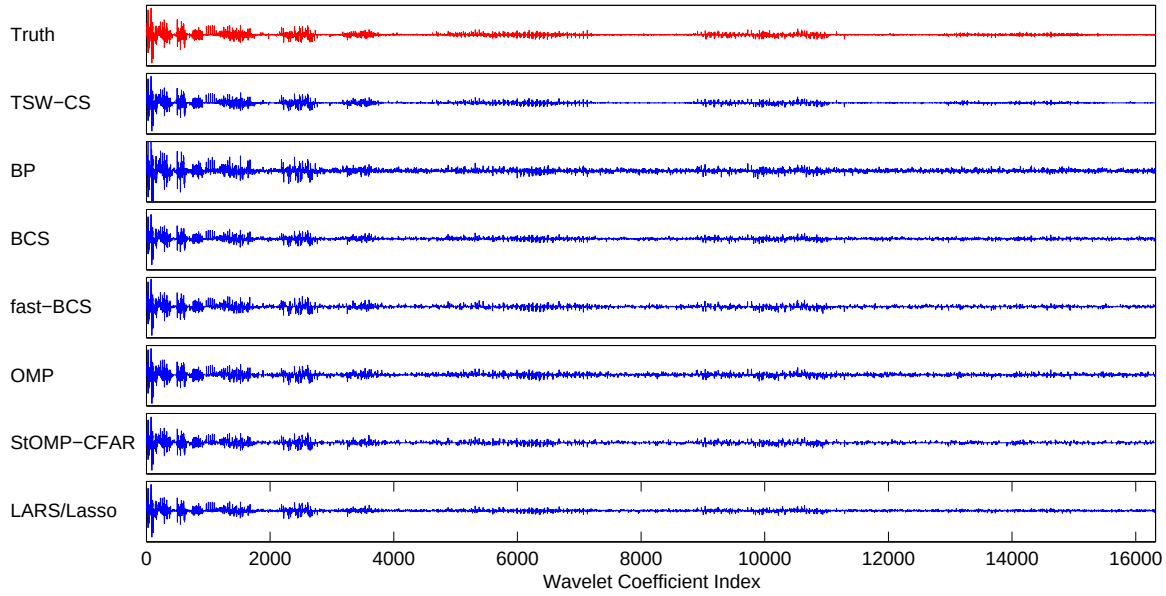


Fig. 8. Comparison of the reconstructed wavelet coefficients by the CS algorithms, for the image *cameraman* with Format 1 and 5000 measurements.

distribution for the wavelet coefficients (and other model parameters), so it yields "error bars" for the estimated wavelet coefficients, indicating the confidence for the current estimation. This level of confidence may be of interest for placing confidences on inferences made from particular portions of the image. Further, if the TSW-CS may be constituted in fast software or (better) in hardware, it may be fast enough to adaptively infer when a sufficient number of CS measurements have been performed. As an example, Figure 10 plots the error bars of the first 192 estimated wavelet coefficients (corresponding to the coefficients at scale $s = 1$) for the image *cameraman* with Format 1; this subset of coefficients are selected to make the figure easy to read, with error bars inferred for all coefficients. From Figure 10 one observes that the error bars on the reconstructed wavelet coefficients become tighter (and the reconstructed coefficients approach to the "truth") as the number of measurements N increases.

In addition to the reconstructed wavelet coefficients, TSW-CS also infers other parameters of the underlying statistical model. For example, the posterior mean of the transition-probability matrix P_s for scale s (equivalent to π_s^0 and π_s^1, see Section III-B), for the image *cameraman* with Format 1, is inferred as

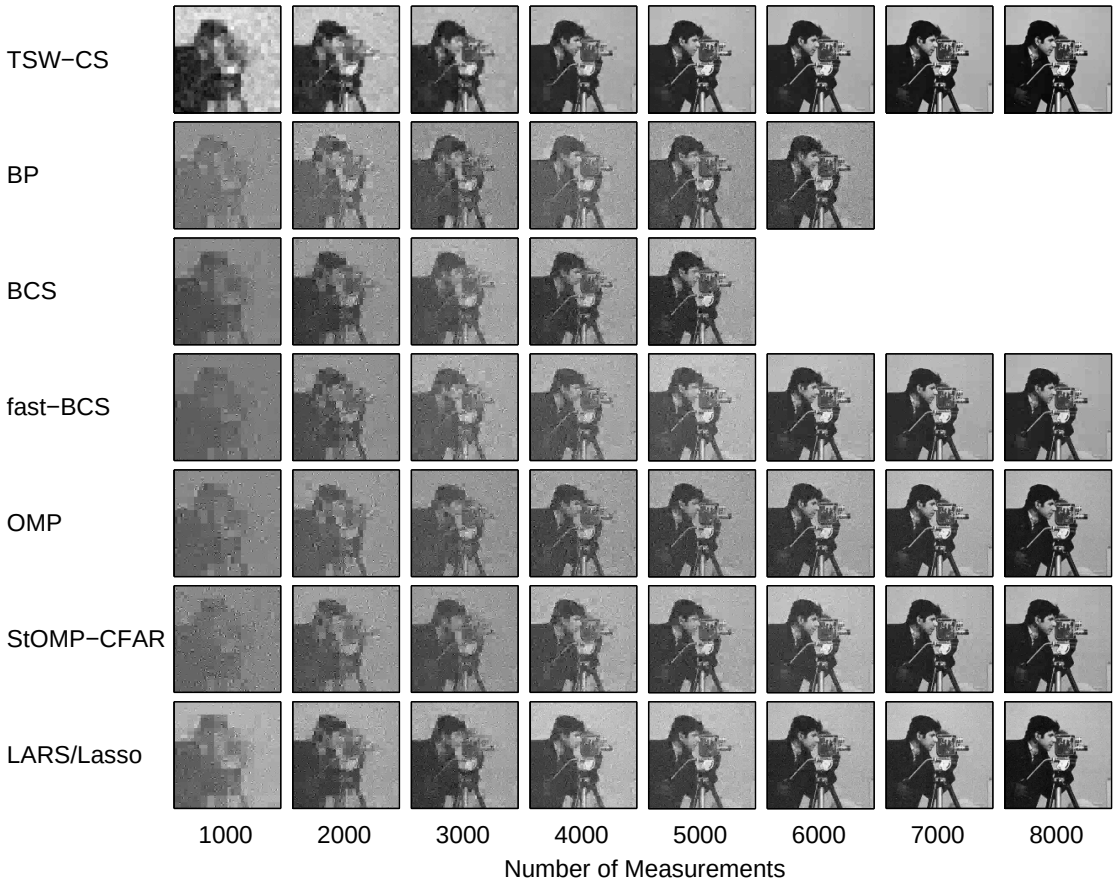


Fig. 9. Comparison of the recovered images by the CS algorithms, for the image *cameraman* with Format 1.

$$P_1 = \begin{bmatrix} 0.1737 & 0.8263 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0.9783 & 0.0217 \\ 0.4771 & 0.5229 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 0.9886 & 0.0114 \\ 0.4755 & 0.5245 \end{bmatrix}, \quad P_4 = \begin{bmatrix} 0.9902 & 0.0098 \\ 0.4940 & 0.5060 \end{bmatrix},$$

with $P_s(i,j)$ defined in Section II. We see that $P_s(1,1) = p(\theta s, i = 0 | \theta_{pa(s,i)} = 0)$ is close to one and slightly increases with the increase of s.

We note that all results presented above are based on images of size 128×128, or an estimation of only about 128×128 coefficients (in Format 3, hybrid CS). For an image of size 256×256 or larger, because of memory limitations, storing and applying $\mathbf{\Phi}$ matrix is often intractable. Fortunately, it has been proven [6] that as an alternative to drawing elements of $\mathbf{\Phi}$ randomly from a zero-mean Gaussian or from Bernoulli distribution, one may consider an arbitrary orthogonal basis and randomly select basis vectors to constitute rows of $\mathbf{\Phi}$. Furthermore,

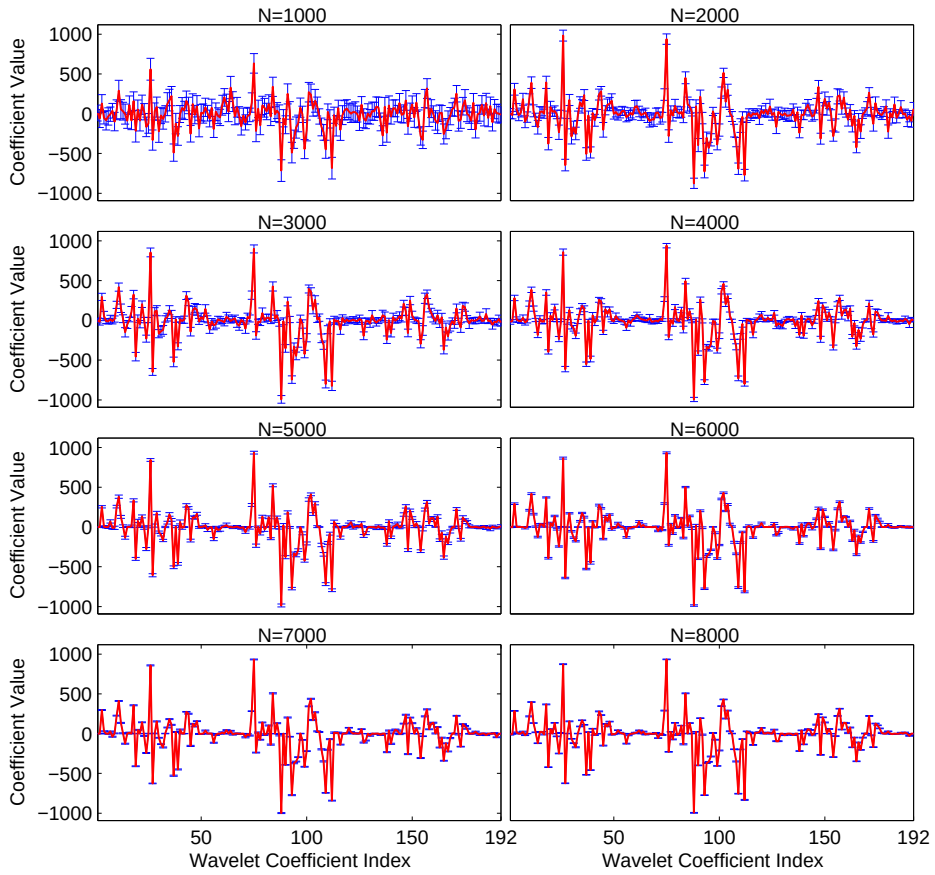


Fig. 10. Error bars of the first 192 estimated wavelet coefficients (corresponding to the coefficients at scale $s = 1$) for the image *cameraman* with Format 1. The error bars are computed as the standard deviation of the posterior distribution approximated by the MCMC samples for each estimated coefficient.

a complex "noiselet" basis set [34] is perfectly incoherent with the Haar wavelet representation (*i.e.*, motivating the random selection of noiselet basis vectors as rows of $\mathbf{\Phi}$). The advantages of constructing $\mathbf{\Phi}$ using orthogonal bases are (i) $\mathbf{\Phi}$ is not stored in the memory explicitly, so it is possible to handle large images, and (ii) all matrix products $\mathbf{\Phi}\boldsymbol{\theta}$ can be replaced by fast transform algorithms ($e.g.$, Fourier transform, wavelet transform, etc.), depending on the orthogonal basis set used for constructing $\mathbf{\Phi}$.

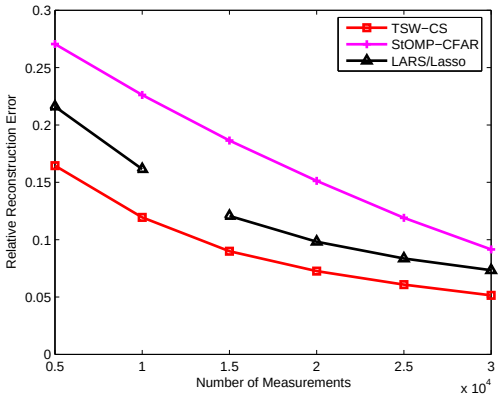
We now test our TSW-CS algorithm on larger images. Assuming the CS measurements are real numbers, we choose discrete cosine transform (DCT) as the orthogonal basis set to construct $\mathbf{\Phi}$ (we may also use noiselets). For an experiment with N measurements, the N rows of $\mathbf{\Phi}$ are selected at random from an $M \times M$ matrix of DCT bases, with the entry in row i and column

j expressed as $C(i)\cos[\pi(i-1)(2j-1)/2M]$, where $C(i)$ is a normalization constant such that $C(i) = 1/\sqrt{M}$ for $i = 1$ and $C(i) = \sqrt{2/M}$ for $2 \leq i \leq M$. After choosing N rows randomly from the DCT bases, $\mathbf{\Phi}$ is constructed by random permutation of columns and rows selected. Note that the indexes of the row selection and the random permutation are stored, but the $\mathbf{\Phi}$ matrix itself is not stored.

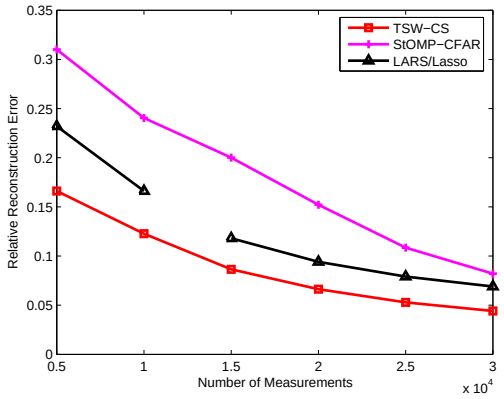
The reconstruction of larger images is performed on the three images employed above with the original sizes (*cameraman* and *peppers* are of size 256×256 and *pirate* is 512×512). Using the same wavelet as above, a wavelet decomposition is performed for all three images such that the scaling coefficients are a 8×8 block (assuming they are measured accurately), and we estimate all the wavelet coefficients (65472 coefficients for *cameraman* and *peppers*; 262080 coefficients for *pirate*). The hyperparameters $a_0, b_0, c_0, d_0, e_0, f_0$ are specified as above. We compare the performances with LARS/Lasso and StOMP-CFAR algorithms, because based on the precious results on smaller images of size 128×128, LARS/Lasso has good performance among all the other algorithms under comparison, and StOMP-CFAR is the fastest algorithm with relatively good performance. BCS also performs well, but estimating and storing the covariance matrix of $\boldsymbol{\theta}$ (an $M \times M$ matrix) in the algorithm makes the reconstruction intractable for large images. The solvers for LARS/Lasso and StOMP-CFAR are modified based on `SolveStOMP` and `SolveLasso`, respectively, with randomly selected DCT bases as rows of $\mathbf{\Phi}$. The false alarm rate for StOMP-CFAR is specified as 0.01.

The performance comparisons for the three images are plotted in Figure 11. We also plot the time comparison for *cameraman* in Figure 12. The results from TSW-CS are, as above, based on the MCMC inference with 200 burn-in iterations and then 100 samples collected. For the cases with the number of measurements $N \geq 15000$, the LARS/Lasso algorithm fails to yield a solution within 72 hours when using the default setting for the maximum iteration number, so the results presented in Figure 11(a), (b) and Figure 12 for LARS/Lasso with $N \geq 15000$ are generated by setting the maximum iteration number to 10000. It is possible that by using the default setting and waiting longer, the performance for LARS/Lasso will be better. The corresponding LARS/Lasso curves are disconnected in Figure 11(a), (b) and Figure 12 to indicate that the two parts are generated using different settings. Figure 11(c) does not show the performance of LARS/Lasso since the maximum iteration number 10000 for the larger image *pirate* is clearly not enough, resulting in reconstruction errors larger than StOMP. Examples of the recovered images are also

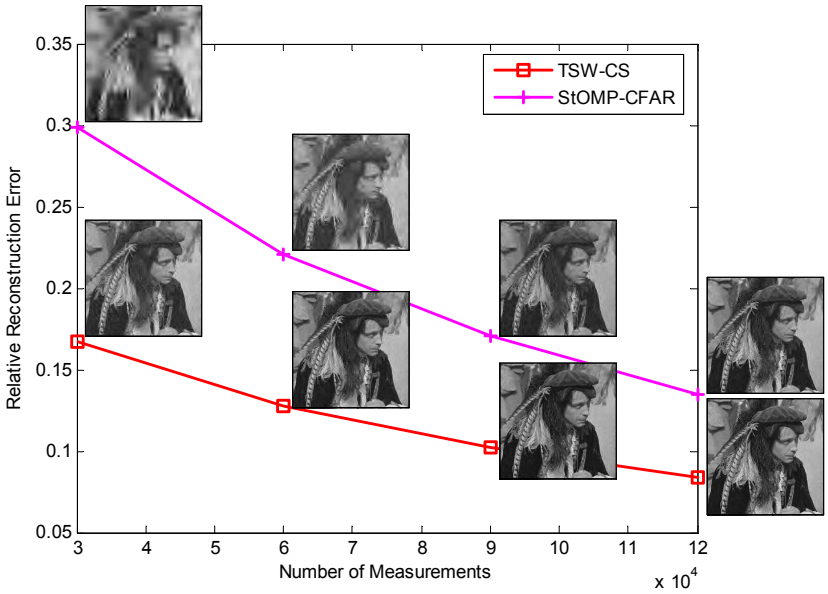
plotted in Figure 11(c). Similar to the small-image cases, we see that the TSW-CS algorithm achieves a low construction error compared to the other algorithms (this is because it is using more information; specifically, not just that the wavelet decomposition is compressible, but also concerning the structure of these compressible coefficients).



(a) Performance comparison for *cameraman*



(b) Performance comparison for *peppers*



(c) Performance comparison for *pirate*

Fig. 11. Performance comparisons for three large images. The LARS/Lasso curves are disconnected in (a) and (b) to indicate that the two parts are generated using different settings. For the number of measurements $N \leq 10000$, the maximum iteration number is specified by default, while for $N \geq 15000$, the maximum iteration number is specified as 10000. Examples of the recovered images are plotted in (c), located at the corresponding evaluated points in the curves.

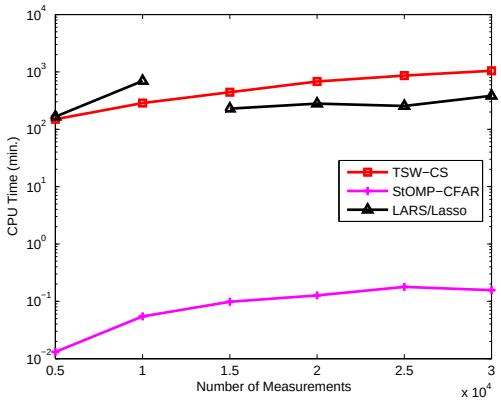


Fig. 12. Time comparison for *cameraman*. The LARS/Lasso curve is disconnected to indicate that the two parts are generated using different settings. For the number of measurements $N \leq 10000$, the maximum iteration number is specified by default, while for $N \geq 15000$, the maximum iteration number is specified as 10000.

Finally we examine the performance of the TSW-CS model with the estimation of the scaling coefficients and wavelet coefficients simultaneous (*not* assuming that the scaling coefficients are measured separately). Using the model presented in (8), we reconstruct $M = 16384$ transform coefficients in total, including both scaling and wavelet coefficients, for the image *cameraman* with Format 1 (this is one example, presented for brevity, with similar results realized for all cases studied to date). The hyperparameters $[e_0^{sc},\ f_0^{sc}] = [1,\ 0] \times M_0$ ($M_0 = 64$ is the number of the scaling coefficients), and the other hyperparameters are specified as above. For comparison, we also perform the experiment using the other CS algorithms, with the same settings and input arguments as above.

Figure 13 plots in solid lines the performance of the CS algorithms when reconstructing all transform coefficients. As reference, we also re-plot the curves of Figure 3(b) with dashed lines, in which the scaling coefficients were assumed known. We see that for TSW-CS there is only a slight degradation in reconstruction accuracy when all transform coefficients are estimate, relative to assuming that the scaling coefficients are known. For some other CS inversion algorithms, such as BP, BCS and LARS/Lasso, estimating scaling coefficients introduces relatively large increased reconstruction errors, particularly when the number of measurements is limited. In Figure 14 we show the recovered images by TSW-CS and by LARS/Lasso (the best among all the other algorithms) when all transform coefficients are estimated. TSW-CS captures the

coarse information of the image successfully with very limited measurements, and the recovered images are less blurry than the images recovered by LARS/Lasso with the same number of measurements. Note that for all images presented in Figure 14, LARS/Lasso has more high-frequency noise than TSW-CS.

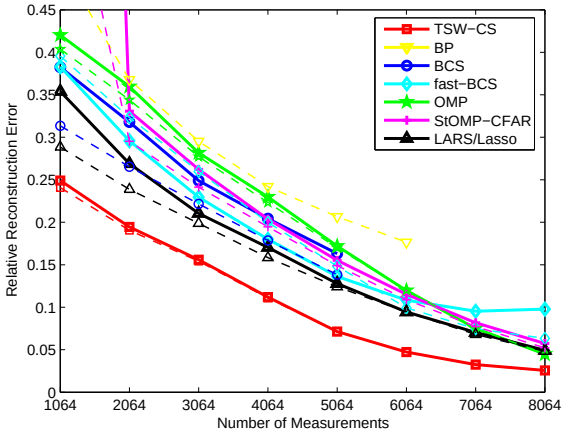


Fig. 13. Performance comparison for the image *cameraman* with Format 1, with and without the estimation of the scaling coefficients. The solid lines represent the performance when scaling and wavelet coefficients are both reconstructed; the dashed line with the same color/marker as a solid line represents the performance of the corresponding algorithm when only wavelet coefficients are reconstructed, assuming scaling coefficients are measured directly and accurately (the measurements of the scaling coefficients are included in the total number of measurements N). The solid curve for BP is beyond the vertical scale of the figure, with relative reconstruction error of 2.98 when $N = 1064$ and 0.71 when $N = 6064$.

V. Conclusions and Future Work

A new statistical model has been developed for Bayesian inverse compressive sensing (CS), for situations in which the signal of interest is compressible in a wavelet basis. The formulation explicitly exploits the structure in the wavelet coefficients of typical/natural signals [5], and related structure is exploited in conventional wavelet-based compression algorithms [4]. The advantage of CS, relative to conventional measure-and-then-compress approaches [4], is that the number of (projection) measurements may be significantly smaller than the number of measurements in traditional sampling methods.

Conventional CS research has assumed that the signal of interest is sparse or compressible in a particular basis ($e.g.$, wavelets), but it assumes no further structure in the transform coefficients.

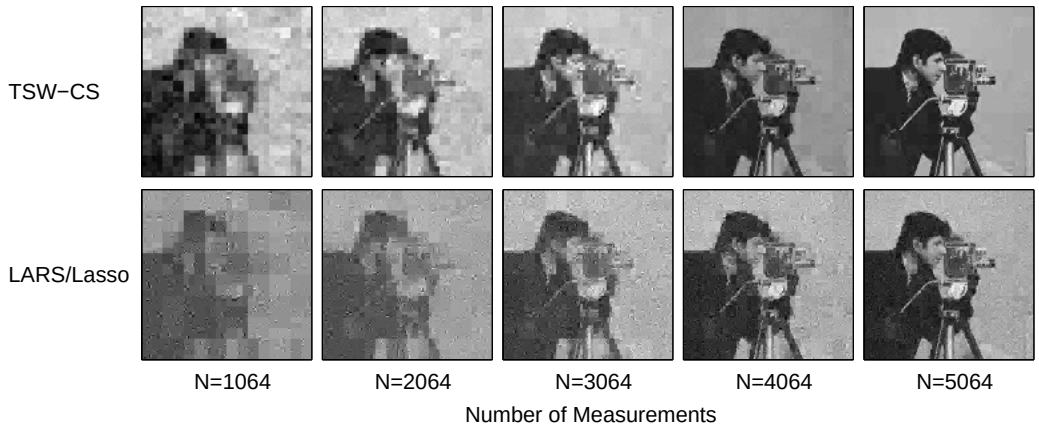


Fig. 14. Comparison of the recovered images by TSW-CS and by LARS/Lasso for the image *cameraman* with Format 1, when both scaling and wavelet coefficients are reconstructed.

Recent research has demonstrated that if one exploits the structure in the transform coefficients characteristic of typical data or imagery, one often may significantly reduce the number of required CS measurements [17], [18]. In this paper we have assumed the signals of interest are compressible in a wavelet basis. The structure associated with typical wavelet coefficients has been utilized in a statistical setting, building on recent research on Bayesian CS [12].

The proposed method utilizes ideas related to the hidden Markov tree statistical representation of wavelet coefficients [5], and an efficient MCMC inference engine has been constituted. On all examples considered to date, considering real imagery, we have observed very fast convergence of the MCMC algorithm; the inference yields an estimate of the wavelet-transform coefficients as well as "error bars" on the coefficients, reflecting a level of confidence in the inference based on the available CS measurements. In this paper, using a set of canonical images that are widely used in the literature, the proposed method has demonstrated competitive computational cost, while consistently providing superior performance, as compared to traditional CS algorithms that do not exploit the structure inherent to the wavelet coefficients.

Concerning future research, there has recently been interest in the simultaneous inversion of multiple distinct CS measurements [35], [36] (by sharing information between these different measurements, the total number of CS measurements may be reduced). The Bayesian setting proposed here is particularly amenable to the joint processing of data from multiple images [37], and this will be investigated in future research. It is also of interest to examine the statistical

leveraging of structure in other popular transforms, such as the DCT.

REFERENCES

[1] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. Academic Press, 1998.

[2] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 18–34, 1992.

[3] C. Christopoulos, "JPEG2000 tutorial," in *IEEE International Conference on Image Processing (ICIP)*, Kobe, Japan, 1999. [Online]. Available: http://www.dsp.toronto.edu/ dsp/JPEG2000/

[4] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.

[5] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov model," *IEEE Transactions on Signal Processing*, vol. 46, pp. 886–902, 1998.

[6] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–985, 2007.

[7] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.

[8] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.

[9] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, 2008.

[10] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[11] A. Wagadarikar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Applied Optics*, vol. 47, pp. B44–B51, 2008.

[12] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346–2356, 2008.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.

[14] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, 2007.

[15] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," March 2006, preprint.

[16] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics (with discussion)*, vol. 32, pp. 407–499, 2004.

[17] T. Blumensath and M. E. Davies, "Sampling theorems for signals from the union of linear subspaces," *IEEE Transactions on Information Theory*, 2007, submitted.

[18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, 2008, submitted.

[19] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, 1993.

[20] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, June 2008, to be published.

[21] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," July 2008, preprint.

[22] H. Ishwaran and J. S. Rao, "Spike and slab variable selection : Frequentist and Bayesian strategies," *Annals of Statistics*, vol. 33, pp. 730–773, 2005.

[23] E. I. George and R. E. McCulloch, "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, vol. 88, pp. 881–889, 1993.

[24] H. Chipman, "Bayesian variable selection with related predictors," *Canadian Journal of Statistics*, vol. 24, pp. 17–36, 1996.

[25] C. Carvalho, J. Chang, J. Lucas, Q. Wang, J. Nevins, and M. West, "High-dimensional sparse factor modelling: Applications in gene expression genomics," *Journal of the American Statistical Association*, 2008, to be published.

[26] M. West, "Bayesian factor regression models in the "large p, small n" paradigm," in *Bayesian Statistics 7*, J. M. Bernardo, A. P. Dawid, J. O. Berger, M. West, D. Heckerman, M. J. Bayarri, and A. F. M. Smith, Eds. Oxford University Press, 2003, pp. 723–732.

[27] S. Ray and B. Mallick, "Functional clustering by Bayesian wavelet methods," *Journal of the Royal Statistical Society. Series B, statistical methodology*, vol. 68, pp. 305–332, 2006.

[28] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden Markov tree model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Press, 2008, pp. 5137–5140.

[29] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer, 2004.

[30] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[31] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[32] C. Bishop and M. Tipping, "Variational relevance vector machines," in *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 46–53.

[33] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Processing*, vol. 86, pp. 549–571, 2006.

[34] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, pp. 27–44, 2001.

[35] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," in *Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*. IEEE Press, 2005, pp. 1537–1541.

[36] S. Ji, D. Dunson, and L. Carin, "Multi-task compressive sensing," *IEEE Transactions on Signal Processing*, 2009, to appear.

[37] Y. Qi, D. Liu, L. Carin, and D. Dunson, "Multi-task compressive sensing with dirichlet process priors," in *International Conference on Machine Learning (ICML)*, 2008.