

Tree-Structured Compressive Sensing with Variational Bayesian Analysis

Lihan He, Haojun Chen and Lawrence Carin
 Department of Electrical and Computer Engineering
 Duke University
 Durham, NC 27708-0291 USA
 {lihan, hc44, lcarin}@ece.duke.edu
 EDICS: DSP-RECO, MAL-BAYL

Abstract

In compressive sensing (CS) the known structure in the transform coefficients may be leveraged to improve reconstruction accuracy. We here develop a hierarchical statistical model applicable to both wavelet and JPEG-based DCT bases, in which the tree structure in the sparseness pattern is exploited explicitly. The analysis is performed efficiently via variational Bayesian (VB) analysis, and comparisons are made with MCMC-based inference, and with many of the CS algorithms in the literature. Performance is assessed for both noise-free and noisy CS measurements, based on both JPEG-DCT and wavelet representations.

Index Terms

variational Bayesian signal processing, discrete cosine transform, sparseness, compression

I. INTRODUCTION

Compressive sensing (CS) [1], [2], [3] constitutes a framework whereby n linear measurements \mathbf{v} may be performed to recover an N -dimensional sparse or compressible signal $\boldsymbol{\theta}$, with $n < N$. The CS measurements may be represented as $\mathbf{v} = \boldsymbol{\Phi}\boldsymbol{\theta}$, where $\boldsymbol{\Phi} \in \mathfrak{R}^{n \times N}$ is a random projection matrix constituted according to CS theory [2], [4]. In practice $\boldsymbol{\theta}$ represents the transform coefficients associated with a basis in which the signal of interest may be sparsely rendered. The *structure* in $\boldsymbol{\theta}$ due to the properties of the orthonormal transform (beyond just sparseness) may be exploited to reduce the number of required CS measurements [5], [6], [7], [8], [9]. Most of these algorithms employ a wavelet transform, and exploit structure in $\boldsymbol{\theta}$ associated with wavelet quadrees [10].

In this letter we employ Bayesian CS inversion to exploit structure in $\boldsymbol{\theta}$, and in this sense the work reported here is most closely related with a recent tree-structured Bayesian CS inversion algorithm [9]. However, the approach in [9] employs Markov chain Monte Carlo (MCMC) inference, which may not be

computationally efficient, and for which convergence may be difficult to diagnose. Variational Bayesian (VB) inference [11] has been widely considered recently in Bayesian analysis, because it often provides a good approximation to the exact posterior, and VB convergence is readily monitored. In the work presented here we propose a modified model to that in [9], allowing inference via VB analysis; the CS inversion time may be significantly reduced compared to the MCMC-based inference in [9].

We also demonstrate how one may exploit the structure in JPEG-based DCT transforms within CS inversion. Two of the most widely used transform encoders are block-DCT-based JPEG [12] and wavelet-based JPEG2000 [13]. Although most of the existing CS inversion algorithms consider a wavelet basis, we here show that related algorithms may be employed for wavelet-based *and* JPEG-based DCT representations of θ .

The contributions of this letter are therefore twofold: (i) development of a hierarchical Bayesian CS inversion algorithm in which VB inference may be performed, yielding computational efficiency while providing state-of-the-art performance (for noise-free and noisy CS measurements); and (ii) the placement of wavelet-based and JPEG-based DCT representations of θ within a single setting, allowing direct CS-inversion compatibility with JPEG or JPEG2000, while exploiting the structure (beyond sparseness) inherent to each construction.

II. EXPLOITING TRANSFORM-COEFFICIENT TREE STRUCTURE

A. Tree Structure in Discrete Cosine Transform

Xiong *et al.* introduced a tree-based representation to characterize the block-DCT transform associated with JPEG [14]. The characteristics of this representation are related to the tree structure exploited in wavelet-based encoding [15], [16]. Using this tree structure in a JPEG block-DCT signal construction, [14] demonstrated state-of-the-art compression quality. As employed in the JPEG standard, the input image is divided into contiguous 8×8 blocks, and each block is transformed to the DCT domain. In [14] each block defines a *single* DCT tree, while in our CS model we adopt a modified version of this structure. Specifically, we consider each 8×8 block in the DCT domain as three depth-three trees, plus one DC component; the three trees (rooted by coefficients 1, 2 and 3 in [14]) are analogous to the high-low, low-high and high-high trees associated with the wavelet transform [10], and hence our developed model may be readily applied directly to a wavelet-based transform instead of the block-DCT-based transform. Each of the three trees consist of 21 dependent coefficients, and the parent-children relationships are defined in [14]. The key idea behind the tree structure is that if a parent coefficient is small, its children are also likely to be small. As discussed in Section II-B, the three trees associated with each 8×8 block

are modeled via a Markovian statistical model, and the DC components are modeled separately. Our CS inversion algorithm is used to infer all transform components including the DC components.

B. Tree-Structured Bayesian Compressive Sensing

In a Bayesian CS framework [17], the CS measurements are represented as $\mathbf{v} = \Phi\boldsymbol{\theta} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \in \mathbb{R}^n$, where $\boldsymbol{\theta} \in \mathbb{R}^N$ represents the large nonzero components in the original transform coefficients, and the small-amplitude coefficients are modeled by the noise $\boldsymbol{\epsilon}$. Any additive measurement noise is added to $\boldsymbol{\epsilon}$.

To model the sparseness of $\boldsymbol{\theta}$, the previous work in [9] employed a ‘‘spike-and-slab’’ prior defined as $(1 - \pi_i)\delta_0 + \pi_i\mathcal{N}(0, \alpha_i^{-1})$ for the component θ_i , with π_i a mixing weight and δ_0 a point mass concentrated at zero. The MCMC method was used in [9] for inference. Although the algorithm achieves good inversion accuracy, it may be slow in general, and MCMC convergence may be difficult to diagnose. The model in [9] is not directly applicable to VB inference.

In this letter we propose an equivalent form to the ‘‘spike-and-slab’’ prior, where the δ_0 term is removed, so that the VB inference can be implemented. The equivalent form represents the sparseness of the coefficients as $\boldsymbol{\theta} = \mathbf{w} \odot \mathbf{z}$, where $\mathbf{w} \in \mathbb{R}^N$ are non-sparse coefficients, $\mathbf{z} \in \mathbb{R}^N$ are zero/one indicators, and ‘‘ \odot ’’ denotes Hadamard product; accordingly, we let $w_i \sim \mathcal{N}(0, \alpha_i^{-1})$, and $z_i \sim \text{Bernoulli}(\pi_i)$. For nonzero components in $\boldsymbol{\theta}$, the associated components in \mathbf{z} will be one; otherwise, the associated components in \mathbf{z} will be zero. The model is summarized as follows:

$$\begin{aligned}
\mathbf{v}|\boldsymbol{\theta}, \alpha_0 &\sim \mathcal{N}(\Phi\boldsymbol{\theta}, \alpha_0^{-1}\mathbf{I}), & \boldsymbol{\theta} &= \mathbf{w} \odot \mathbf{z}, & w_{k,s,i} &\sim \mathcal{N}(0, \alpha_s^{-1}), \\
z_{k,s,i} &\sim \text{Bernoulli}(\pi_{k,s,i}), & \text{with } \pi_{k,s,i} &= \begin{cases} \pi^s, & \text{if } s = 0, 1, \\ \pi^{s0}, & \text{if } 2 \leq s \leq L, z_{pa(k,s,i)} = 0, \\ \pi^{s1}, & \text{if } 2 \leq s \leq L, z_{pa(k,s,i)} = 1, \end{cases} \\
\alpha_0 &\sim \text{Gamma}(a_0, b_0), & \alpha_s &\sim \text{Gamma}(c_0, d_0), & 0 \leq s \leq L, \\
\pi^s &\sim \text{Beta}(e_0^s, f_0^s), & s &= 0, 1, \\
\pi^{s0} &\sim \text{Beta}(e_0^{s0}, f_0^{s0}), & \pi^{s1} &\sim \text{Beta}(e_0^{s1}, f_0^{s1}), & 2 \leq s \leq L,
\end{aligned} \tag{1}$$

where the subscript $\{k, s, i\}$ indexes the i th components at level s of a DCT-tree in block k , and $pa(k, s, i)$ represents the associated parent component, when present. Let $s = 0$ denote DC components, and let $s = 1, \dots, L$ denote AC components from root level ($s = 1$) to leaf level ($s = L$) of a tree; for an 8×8 DCT-block, $L = 3$. The tree-structural information is imposed in the second row of (1). Depending on the parent values of the coefficients, different prior parameters π are imposed for the associated indicators \mathbf{z} : for the coefficients with a zero-valued parent, a prior preferring zero is considered for π^{s0} (the last

row of (1)), and for the coefficients with a nonzero parent, a uniform prior is imposed for π^{s1} . Readers may refer to [9] for a detailed explanation of the model and the specification of hyperparameters (which are readily set, and for which no tuning has been performed for the results presented below).

Note that in the model θ are block-DCT coefficients, and the tree structure is constructed in the block-DCT representation, as discussed in Section II-A. However, one may readily replace the block-DCT coefficients with wavelet coefficients, and replace the DCT trees with wavelet trees as defined in [9], with minor change of the model (the only change happens on the definition of the parent-children relationship $pa(k, s, i)$). When presenting results, we compare the performance of DCT-based CS inversion to wavelet-based inversion.

The posterior distributions of the model parameters Θ are inferred from observed data – CS measurements \mathbf{v} , via the VB inference [11], which converges fast and is computationally efficient (by contrast, MCMC convergence may be difficult to assess). The VB inference employs a set of distributions $q(\Theta)$ to approximate the true posterior distributions $p(\Theta|\mathbf{v})$, and uses a lower bound \mathcal{F} to approximate the true log-likelihood of the model $\log p(\mathbf{v}|\Theta)$. The algorithm iteratively updates $q(\Theta)$ so that \mathcal{F} approaches to $\log p(\mathbf{v}|\Theta)$ until convergence. In our model, $\Theta = \{\mathbf{w}, \mathbf{z}, \alpha_0, \{\alpha_s\}_{s=0:L}, \pi^0, \pi^1, \{\pi^{s0}, \pi^{s1}\}_{s=2:L}\}$.

The update equations for the posterior distributions of Θ are as follows:

$$\begin{aligned} q(z_{k,s,i}) &= \text{Beronulli}(z_{k,s,i}|p_{k,s,i}), & q(w_{k,s,i}) &= \mathcal{N}(w_{k,s,i}|\mu_{k,s,i}, \sigma_{k,s,i}^2), \\ q(\alpha_0) &= \text{Gamma}(\alpha_0|a, b), & q(\alpha_s) &= \text{Gamma}(\alpha_s|c_s, d_s), \\ q(\boldsymbol{\pi}) &= \prod_{s=0}^1 \text{Beta}(\pi^s|e^s, f^s) \prod_{s=2}^L [\text{Beta}(\pi^{s0}|e^{s0}, f^{s0})\text{Beta}(\pi^{s1}|e^{s1}, f^{s1})], \end{aligned} \quad (2)$$

with

$$\begin{aligned} p_{k,s,i} &= [1 + \exp\{\langle \ln(1 - \pi_{k,s,i}) \rangle - \langle \ln \pi_{k,s,i} \rangle \\ &\quad + \frac{\langle \alpha_0 \rangle}{2} [\langle w_{k,s,i}^2 \rangle \boldsymbol{\Phi}_{(j)}^T \boldsymbol{\Phi}_{(j)} - 2 \langle w_{k,s,i} \rangle \boldsymbol{\Phi}_{(j)}^T (\mathbf{v} - \sum_{\substack{l=1 \\ l \neq j}}^N \boldsymbol{\Phi}_{(l)} \langle z_{(l)} \rangle \langle w_{(l)} \rangle)]\}]^{-1}, \end{aligned} \quad (3)$$

$$\begin{aligned} \sigma_{k,s,i}^2 &= (\langle \alpha_s \rangle + \langle \alpha_0 \rangle \langle z_{k,s,i}^2 \rangle \boldsymbol{\Phi}_{(j)}^T \boldsymbol{\Phi}_{(j)})^{-1}, \\ \mu_{k,s,i} &= \sigma_{k,s,i}^2 \langle \alpha_0 \rangle \langle z_{k,s,i} \rangle \boldsymbol{\Phi}_{(j)}^T (\mathbf{v} - \sum_{\substack{l=1 \\ l \neq j}}^N \boldsymbol{\Phi}_{(l)} \langle z_{(l)} \rangle \langle w_{(l)} \rangle), \end{aligned} \quad (4)$$

$$a = a_0 + n/2,$$

$$\begin{aligned} b &= b_0 + \|\mathbf{v} - \boldsymbol{\Phi}(\langle \mathbf{z} \rangle \odot \langle \mathbf{w} \rangle)\|_2^2/2 \\ &\quad + \text{trace}(\boldsymbol{\Phi}^T \boldsymbol{\Phi} [\langle \mathbf{w} \mathbf{w}^T \rangle \odot \langle \mathbf{z} \mathbf{z}^T \rangle - (\langle \mathbf{w} \rangle \odot \langle \mathbf{z} \rangle)(\langle \mathbf{w} \rangle \odot \langle \mathbf{z} \rangle)^T])/2, \end{aligned} \quad (5)$$

$$c_s = c_0 + N_s/2, \quad d_s = d_0 + \sum_k \sum_i \langle w_{k,s,i}^2 \rangle / 2, \quad s = 0, \dots, L, \quad (6)$$

$$e^s = e_0^s + \sum_k \sum_i \langle z_{k,s,i} \rangle, \quad f^s = f_0^s + N_s - \sum_k \sum_i \langle z_{k,s,i} \rangle, \quad s = 0, 1,$$

$$e^{s0} = e_0^{s0} + \sum_k \sum_i (1 - \langle z_{pa(k,s,i)} \rangle) \langle z_{k,s,i} \rangle, \quad s = 2, \dots, L,$$

$$f^{s0} = f_0^{s0} + \sum_k \sum_i (1 - \langle z_{pa(k,s,i)} \rangle) (1 - \langle z_{k,s,i} \rangle), \quad s = 2, \dots, L,$$

$$e^{s1} = e_0^{s1} + \sum_k \sum_i \langle z_{pa(k,s,i)} \rangle \langle z_{k,s,i} \rangle, \quad s = 2, \dots, L,$$

$$f^{s1} = f_0^{s1} + \sum_k \sum_i \langle z_{pa(k,s,i)} \rangle (1 - \langle z_{k,s,i} \rangle), \quad s = 2, \dots, L, \quad (7)$$

where we assume $z_{k,s,i}$ corresponds to the j th entry in the N -dimensional vector \mathbf{z} , denoted by $z_{(j)}$, $\Phi_{(j)}$ represents the j th column of Φ , and N_s denotes the total number of coefficients at tree level s . The angle bracket $\langle \rangle$ represents the expectation with respect to the random variable in it. They can be evaluated as:

$$\begin{aligned} \langle z_{k,s,i} \rangle &= p_{k,s,i}, & \langle w_{k,s,i} \rangle &= \mu_{k,s,i}, & \langle \alpha_0 \rangle &= a/b, \\ \langle z_{k,s,i}^2 \rangle &= p_{k,s,i}, & \langle w_{k,s,i}^2 \rangle &= \mu_{k,s,i}^2 + \sigma_{k,s,i}^2, & \langle \alpha_s \rangle &= c_s/d_s, \\ \langle \ln(1 - \pi_{k,s,i}) \rangle - \langle \ln \pi_{k,s,i} \rangle &= \\ &\begin{cases} \psi(f^s) - \psi(e^s), & \text{if } s = 0, 1, \\ \langle z_{pa(k,s,i)} \rangle [\psi(f^{s1}) - \psi(e^{s1})] + (1 - \langle z_{pa(k,s,i)} \rangle) [\psi(f^{s0}) - \psi(e^{s0})], & \text{if } 2 \leq s \leq L, \end{cases} \end{aligned}$$

where ψ is the *digamma* function [11] defined as $\psi(x) = \frac{\partial}{\partial x} \ln \Gamma(x)$.

Note that in the CS problem the matrix $\Phi^T \Phi$ is approximately diagonal, so (5) may be approximated as

$$b \approx b_0 + \|\mathbf{v} - \Phi(\langle \mathbf{z} \rangle \odot \langle \mathbf{w} \rangle)\|_2^2 / 2 + \sum_{l=1}^N \Phi_{(l)}^T \Phi_{(l)} [\langle w_{(l)}^2 \rangle \langle z_{(l)}^2 \rangle - \langle w_{(l)} \rangle^2 \langle z_{(l)} \rangle^2] / 2, \quad (8)$$

which further reduces the computational burden, with minimal impact on the reconstruction accuracy.

III. EXPERIMENTAL RESULTS

We compare CS inversion with a block-DCT-based (JPEG) representation vis-a-vis a wavelet-based (JPEG2000) representation. For the latter the same basic inversion model (1) is employed, with the trees now corresponding to wavelet coefficients. The Haar wavelet was employed, but any discrete wavelet may be used. We also perform an analysis of performance in the presence of noise. The proposed model is referred to below as TS-BCS-VB, for tree-structured Bayesian CS implemented by VB inference, with detailed distinctions made depending on whether a block-DCT or wavelet decomposition is used. In all

examples presented below the matrix $\Phi \in \mathfrak{R}^{n \times N}$ is constructed randomly, with components drawn i.i.d. from $\mathcal{N}(0, 1/n)$. Any of the other constructions for Φ discussed in [1], [2], [3] may also be employed.

A. Comparisons to other CS algorithms

We compare the performance of TS-BCS-VB to eight recently developed CS reconstruction algorithms: basis pursuit (BP) [18], Bayesian compressive sensing (BCS) [17], fast-BCS [17], orthogonal matching pursuit (OMP) [19], stagewise orthogonal matching pursuit (StOMP) [20], Lasso-modified least angle regression (LARS/Lasso) [21], total variation (TV) [22], and TS-BCS-MCMC [9]. For the TV and BP implementations, we used solvers from the ℓ_1 -*Magic* toolbox¹; for the OMP, StOMP and LARS/Lasso algorithms, we use the solvers `SolveOMP`, `SolveStOMP`, and `SolveLasso`, respectively, from the *Sparse-Lab* toolbox²; for the BCS, fast-BCS, and TS-BCS-MCMC, we use the packages `bcs_vb`, `bcs_ver0.1`, and `tswcs` (for wavelet) and `tsdctcs` (for block-DCT), respectively, from the BCS website³. All software are written in MATLABTM.

All examples considered below are for 128×128 images. Our objective is to estimate $N = 128^2 = 16384$ DCT coefficients, or scaling and wavelet coefficients. To provide a fair comparison, we let wavelet decomposition depth $L = 3$ such that the DCT-tree and the wavelet-tree have the same depth. The hyperparameters for both TS-BCS-VB and TS-BCS-MCMC are set as in [9]. For the other CS algorithms, default parameters are used. For the StOMP algorithm, we use the CFAR thresholding, with the false-alarm rate specified as 0.01.

For each CS algorithm we produce a curve of relative reconstruction error as a function of number of CS measurements n . The relative reconstruction error is defined as $\|\mathbf{x} - \hat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$, where \mathbf{x} is the original image, and $\hat{\mathbf{x}}$ is the recovered image based on the DCT coefficients or wavelet coefficients reconstructed by a particular CS algorithm.

We consider the same dataset used in [9]. The dataset contains five image classes and ten images from each class. In Figure 1(a) a comparison of reconstruction error is performed for all algorithms discussed above, for a JPEG-DCT basis, using the first image in the dataset, and the comparison for computation time is presented in Figure 1(b). The results for wavelet-based representation and for all other images are quite similar to this example, and are omitted for brevity (tabulated results are presented below). The BP

¹<http://www.acm.caltech.edu/l1magic/>

²<http://sparselab.stanford.edu/>

³<http://www.ece.duke.edu/~lihan/cs/>

results were found to yield errors significantly larger than those associated with the other methods (the relative errors are greater than 0.7 when $n = 6000$ in Figure 1(a)), and therefore they are not explicitly presented in Figure 1(a) and in other results below.

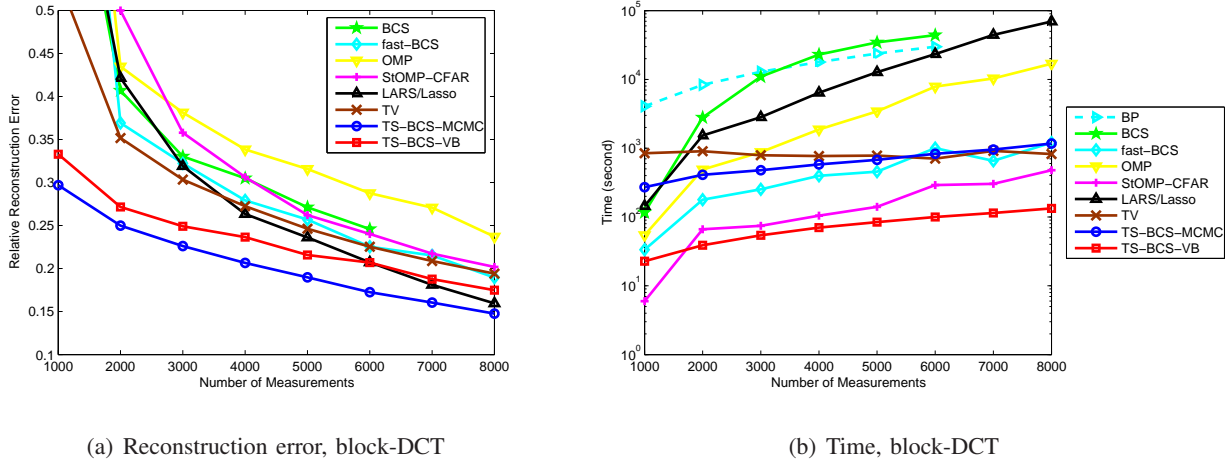


Fig. 1. Comparisons of relative reconstruction error and computation time, for a 128×128 image. Note that in (b) the y -axis is in the logarithmic scale.

From Figure 1 we observe that the TS-BCS algorithm benefits from exploiting the tree-based structure information, especially with a small number of measurements. With MCMC inference, the TS-BCS algorithm achieves the best accuracy under comparison, with competitive computation time. The VB inference reduces the computation time significantly, with a relatively small cost of reconstruction quality. Typically the VB inference converges after 30 to 40 iterations, and it takes about 2 minutes to reconstruct a 128×128 image with 8000 CS measurements. Consequently, one may consider TS-BCS with MCMC inference if the reconstruction accuracy is the main concern, or consider TS-BCS with VB inference if the computation time is more important.

In Table I is presented average performance of all algorithms, for block-DCT and wavelet-based representation, based on *all* 50 images in the dataset, for $n = 2000$ (corresponding to a relatively small number of CS measurements) and $n = 6000$ (corresponding to a relatively large number of measurements). From these results, we observe that the TS-BCS consistently performs better than the other algorithms, both under the block-DCT basis and with the wavelet basis. In addition, the reconstruction error by VB inference is a bit larger than that by MCMC inference, but we should note that the VB inference is much faster. These results also indicate that CS reconstruction performance under the block-DCT and wavelet basis are very similar for all the algorithms.

(a) block-DCT, number of measurements $n = 2000$									(b) block-DCT, $n = 6000$										
Algorithm		BCS	fast-BCS	OMP	StOMP- CFAR	LARS/ Lasso	TV	TS-BCS -MCMC	TS-BCS -VB	Algorithm		BCS	fast-BCS	OMP	StOMP- CFAR	LARS/ Lasso	TV	TS-BCS -MCMC	TS-BCS -VB
Urban	MEAN	0.3459	0.2898	0.3419	0.3987	0.3664	0.3073	0.2026	0.2189	Urban	MEAN	0.1817	0.1715	0.2180	0.1825	0.1586	0.1920	0.1335	0.1547
	STD	0.0426	0.0504	0.0545	0.0624	0.0416	0.0287	0.0260	0.0275		STD	0.0289	0.0260	0.0326	0.0275	0.0233	0.0158	0.0189	0.0238
Buildings	MEAN	0.3713	0.3295	0.3833	0.4366	0.4022	0.3281	0.2217	0.2417	Buildings	MEAN	0.1963	0.1856	0.2314	0.1937	0.1684	0.2033	0.1383	0.1607
	STD	0.1217	0.1249	0.1305	0.1209	0.1011	0.0763	0.0597	0.0727		STD	0.0746	0.0765	0.0922	0.0738	0.0646	0.0498	0.0556	0.0675
Office	MEAN	0.4119	0.3741	0.4271	0.4630	0.4180	0.3551	0.2368	0.2534	Office	MEAN	0.1858	0.1737	0.2191	0.1857	0.1610	0.1965	0.1236	0.1379
	STD	0.0798	0.0976	0.0880	0.0816	0.0511	0.0612	0.0429	0.0471		STD	0.0367	0.0376	0.0428	0.0370	0.0329	0.0260	0.0268	0.0287
Cows	MEAN	0.2594	0.2203	0.2693	0.3219	0.3087	0.2671	0.1653	0.1726	Cows	MEAN	0.2345	0.1362	0.1710	0.1430	0.1238	0.1722	0.1016	0.1188
	STD	0.1041	0.0901	0.0944	0.0960	0.0933	0.0630	0.0552	0.0633		STD	0.2081	0.0641	0.0790	0.0634	0.0562	0.0386	0.0499	0.0618
Flowers	MEAN	0.4054	0.3609	0.4115	0.4855	0.4299	0.3438	0.2248	0.2531	Flowers	MEAN	0.2599	0.2025	0.2439	0.2090	0.1815	0.2122	0.1362	0.1640
	STD	0.1429	0.1481	0.1516	0.1604	0.1208	0.0898	0.0734	0.0944		STD	0.0720	0.0956	0.1349	0.1008	0.0853	0.0626	0.0794	0.0970

(c) wavelets, $n = 2000$									(d) wavelets, $n = 6000$										
Algorithm		BCS	fast-BCS	OMP	StOMP- CFAR	LARS/ Lasso	TV	TS-BCS -MCMC	TS-BCS -VB	Algorithm		BCS	fast-BCS	OMP	StOMP- CFAR	LARS/ Lasso	TV	TS-BCS -MCMC	TS-BCS -VB
Urban	MEAN	0.3582	0.3012	0.3531	0.3993	0.3748	0.2690	0.2123	0.2335	Urban	MEAN	0.1870	0.1744	0.2246	0.1875	0.1607	0.1865	0.1383	0.1588
	STD	0.0600	0.0460	0.0524	0.0565	0.0455	0.0289	0.0267	0.0299		STD	0.0270	0.0270	0.0337	0.0272	0.0228	0.0156	0.0201	0.0261
Buildings	MEAN	0.3928	0.3311	0.3847	0.4470	0.4128	0.2840	0.2296	0.2535	Buildings	MEAN	0.2073	0.1970	0.2456	0.2050	0.1772	0.1999	0.1490	0.1752
	STD	0.1151	0.1118	0.1184	0.1182	0.0919	0.0669	0.0608	0.0682		STD	0.0753	0.0812	0.0958	0.0772	0.0652	0.0489	0.0569	0.0675
Office	MEAN	0.4441	0.3920	0.4465	0.4838	0.4386	0.3133	0.2539	0.2767	Office	MEAN	0.1935	0.1838	0.2273	0.1930	0.1674	0.1898	0.1311	0.1465
	STD	0.0701	0.0985	0.0976	0.0743	0.0551	0.0521	0.0490	0.0489		STD	0.0425	0.0479	0.0543	0.0442	0.0351	0.0230	0.0319	0.0355
Cows	MEAN	0.2705	0.2369	0.2825	0.3251	0.3189	0.2306	0.1773	0.1912	Cows	MEAN	0.1814	0.1415	0.1790	0.1487	0.1304	0.1696	0.1088	0.1252
	STD	0.1008	0.0899	0.0941	0.1025	0.0943	0.0546	0.0542	0.0622		STD	0.0634	0.0660	0.0826	0.0669	0.0560	0.0375	0.0502	0.0582
Flowers	MEAN	0.4193	0.3770	0.4240	0.4879	0.4462	0.3092	0.2415	0.2648	Flowers	MEAN	0.2948	0.2238	0.2773	0.2275	0.1975	0.2173	0.1598	0.1847
	STD	0.1405	0.1434	0.1422	0.1570	0.1233	0.0775	0.0702	0.0814		STD	0.1270	0.0967	0.1212	0.0942	0.0801	0.0614	0.0705	0.0834

TABLE I

MEAN AND STANDARD DEVIATION OF THE RECONSTRUCTION ERROR FOR EACH CLASS IN THE DATASET. THE BOLD NUMBERS REPRESENT THE BEST TWO AMONG ALL THE CS ALGORITHMS UNDER COMPARISON.

B. Performance with noisy measurements

In all experiments presented above we assumed the CS measurements \mathbf{v} were noise free. We now test the reconstruction performance with noisy measurements, and therefore within TS-BCS now the Gaussian precision α_0 accounts for negligibly small coefficients *and* additive noise.

Assume additive white Gaussian noise $\boldsymbol{\eta} \in \mathbb{R}^n$ is introduced in the CS measurements. The noisy measurements $\bar{\mathbf{v}} = \mathbf{v} + \boldsymbol{\eta}$, where \mathbf{v} is the noise-free measurements, and each component of $\boldsymbol{\eta}$ is i.i.d. drawn from a Gaussian distribution $\mathcal{N}(0, \alpha^{-1})$. The noise variance $1/\alpha$ defines the signal-to-noise ratio (SNR), represented as $SNR(\text{dB}) = 20 \log_{10} (\|\mathbf{v}\|_{\ell_2} / \|\boldsymbol{\eta}\|_{\ell_2})$. Note that when performing CS inversion, the noise variance is *not* assumed known, and α_0 is inferred in the CS inversion.

Figure 2 shows the reconstruction error vs. SNR when number of measurements $n = 4000$. It is observed that when $SNR \geq 30$ dB, all the algorithms perform well. However, when the SNR becomes lower, the TS-BCS algorithm for both DCT coefficients and wavelet coefficients are less sensitive to the noise, and performs particularly better than the other algorithms when the SNR is very low. This may be attributed to the fact that noisy measurements are naturally accounted for within the Bayesian linear-

regression model. In addition, we observe that the performances for a block-DCT and wavelet-based reconstruction are comparable.

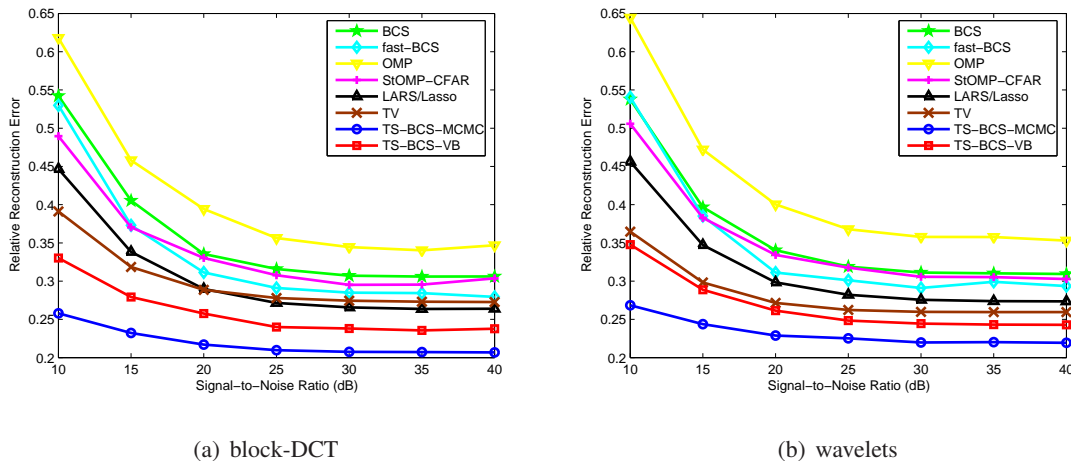


Fig. 2. Comparison of the reconstruction error given number of measurements $n = 4000$, under noisy measurements.

IV. CONCLUSIONS

A hierarchical Bayesian model has been developed to characterize the statistics of the coefficients in a block-DCT or wavelet transform of natural images, and this model has been employed within a CS inversion algorithm. The framework explicitly imposes sparseness in the inferred coefficients, also constituting a Markovian structure between different spectral components. The model is particularly designed for the implementation of the variational Bayesian inference, so that the computation time may be significantly reduced compared to the MCMC-based model in a previous study [9]. The inversion algorithm allows direct utilization of a signal decomposition that is compatible with JPEG, and therefore the inferred coefficients may be directly employed within a subsequent JPEG encoder.

Based upon extensive tests with natural images, we observe that exploitation of the statistical structure in the block-DCT and wavelet representations yields state-of-the-art CS inversion quality, relative to many of the existing algorithms in the CS literature. The VB inference greatly reduces the computation time, with minimal increase of the reconstruction error, compared to the MCMC inference.

REFERENCES

- [1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.
- [2] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–985, 2007.

- [3] 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.
- [4] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry principle for random matrices,” *Constructive Approximation*, 2007.
- [5] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a union of subspaces,” *Preprint*, 2008.
- [6] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” *IEEE Transactions on Information Theory*, 2008, submitted.
- [7] C. La and M. Do, “Signal reconstruction using sparse tree representations,” *Proc. SPIE Wavelets XI*, 2005.
- [8] 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.
- [9] L. He and L. Carin, “Exploiting structure in wavelet-based Bayesian compressive sensing,” *IEEE Trans. Signal Processing*, Sept. 2009.
- [10] 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.
- [11] M. J. Beal, “Variational algorithms for approximate Bayesian inference,” Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.
- [12] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 18–34, 1992.
- [13] C. Christopoulos, “JPEG2000 tutorial,” in *IEEE International Conference on Image Processing (ICIP)*, Kobe, Japan, 1999. [Online]. Available: <http://www.dsp.toronto.edu/dsp/JPEG2000/>
- [14] Z. Xiong, O. G. Gulerguz, and M. T. Orchard, “A DCT-Based embedded image coder,” *IEEE Signal Processing Letters*, vol. 3, pp. 289–290, 1996.
- [15] 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.
- [16] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, 1993.
- [17] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346–2356, 2008.
- [18] 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.
- [19] 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.
- [20] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit,” *Stanford Statistics Technical Report 2006-2*, April 2006.
- [21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of Statistics (with discussion)*, vol. 32, pp. 407–499, 2004.
- [22] 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.