# Compressive Imaging using Approximate Message Passing and a Markov-Tree Prior

Subhojit Som, Lee C. Potter, and Philip Schniter
Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210.
Email: som.4@osu.edu, potter@ece.osu.edu, schniter@ece.osu.edu

***Abstract*—We propose a novel algorithm for compressive imaging that exploits both the sparsity and persistence across scales found in the 2D wavelet transform coefficients of natural images. Like other recent works, we model wavelet structure using a hidden Markov tree (HMT) but, unlike other works, ours is based on loopy belief propagation (LBP). For LBP, we adopt a recently proposed "turbo" message passing schedule that alternates between exploitation of HMT structure and exploitation of compressive-measurement structure. For the latter, we leverage Donoho, Maleki, and Montanari's recently proposed approximate message passing (AMP) algorithm. Experiments on a large image database show that our turbo LBP approach maintains state-of-the-art reconstruction performance at half the complexity.**[1]

## I. Introduction

In compressive imaging [1], we aim to estimate an image $\boldsymbol{x} \in \mathbb{R}^N$ from $M \leq N$ noisy linear observations $\boldsymbol{y} \in \mathbb{R}^M$,

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta} + \boldsymbol{w}, \tag{1}$$

assuming that the image has a representation $\boldsymbol{\theta} \in \mathbb{R}^N$ in some wavelet basis $\boldsymbol{\Psi}$ (i.e., $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\theta}$) containing only a few ($K$) large coefficients (i.e., $K \ll N$). In (1), $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ is a known measurement matrix and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ is additive white Gaussian noise. Though $M < N$ makes the problem ill-posed, it has been shown that $\boldsymbol{x}$ can be recovered from $\boldsymbol{y}$ when $K$ is adequately small and $\boldsymbol{\Phi}$ is incoherent with $\boldsymbol{\Psi}$ [1].

The wavelet coefficients of natural images are known to have an additional structure known as *persistence across scales* (PAS) [2], which we now describe. For 2D images, the wavelet coefficients are naturally organized into quad-trees, where each coefficient at level $j$ acts as a parent for four child coefficients at level $j+1$. The PAS property says that, if a parent is very small, then all of its children are likely to be very small; similarly, if a parent is large, then it is likely that some (but not necessarily all) of its children will also be large.

Several authors have exploited the PAS property for compressive imaging [3]–[6]. The so-called "model-based" approach [3] is a deterministic incarnation of PAS that leverages a restricted union-of-subspaces and manifests as a modified CoSaMP [7] algorithm. Most other approaches are Bayesian in nature, exploiting the fact that PAS is readily modeled by a *hidden Markov tree* (HMT) [8]. The first work in this direction appears to be [4], where an iteratively re-weighted $\ell_1$ algorithm, generating an estimate of $\boldsymbol{x}$, was alternated with a Viterbi algorithm, generating an estimate of the HMT states.

[1]Supported in part by NSF CCF-1018368 and AFOSR FA9550-06-1-0324.

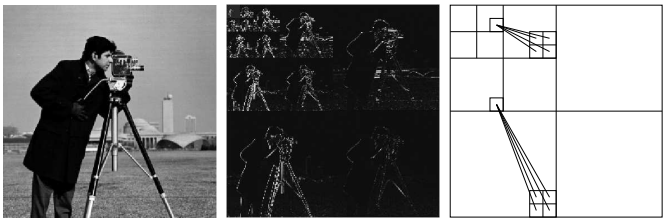

Fig. 1. Left: The camera-man image. Center: The corresponding transform coefficients, demonstrating PAS. Right: An illustration of quad-tree structure.

More recently, HMT-based compressive imaging has been attacked using modern Bayesian tools [9]. For example, [5] used Markov-chain Monte-Carlo (MCMC), which is known to yield correct posteriors after convergence. For practical image sizes, however, convergence takes an impossibly long time, and so MCMC must be terminated early, at which point its performance suffers. Variational Bayes (VB) can sometimes offer a better performance/complexity tradeoff, motivating the approach in [6]. Our experiments indicate that, while [6] indeed offers a state-of-the art performance/complexity tradeoff, it is possible to do significantly better.

In this paper, we propose a novel approach to HMT-based compressive imaging based on loopy belief propagation [10]. For this, we model $\boldsymbol{\theta}$ as Bernoulli-Gaussian with HMT structure on the coefficient state, and propagate beliefs on the corresponding factor graph. A recently proposed "turbo" schedule [11] suggests to iterate between exploitation of the HMT structure and exploitation of the observation structure from (1). For the former we use the standard sum-product algorithm [10], and for the latter the recently proposed *approximate message passing* (AMP) algorithm [12], which has been proven to yield asymptotically correct posteriors (as $M, N \to \infty$ with $M/N$ fixed) under i.i.d Gaussian $\boldsymbol{\Phi}$ [13], while being very fast—operating, essentially, as an iterative thresholding algorithm. In this paper, we further expand the factor graph to enable joint estimation of all statistical parameters (e.g., the AWGN variance $\sigma^2$ and the HMT parameters). Experiments on a large image database show that our turbo approach maintains state-of-the-art reconstruction performance at half the complexity.

## II. Signal Model

Since we use a 2D wavelet transform, the transform coefficients $\{\theta_n\}$ can be partitioned into so-called "wavelet" coefficients (at indices $n \in \mathcal{W}$) and "approximation" coefficients (at indices $n \in \mathcal{A}$). The wavelet coefficients can be

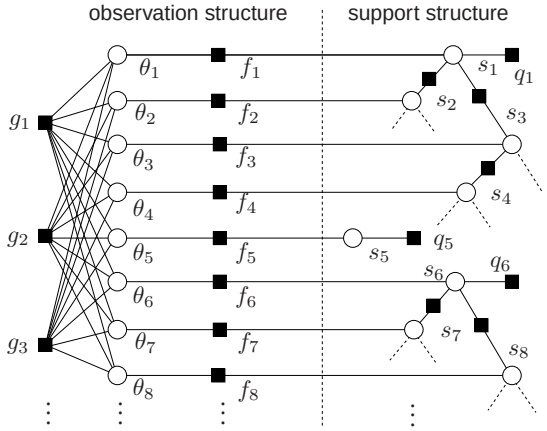


Fig. 2. Factor graph representation of the signal model. The variables $s_1$ and $s_6$ are wavelet states at the roots of two different Markov trees. The variable $s_5$ is an approximation state and hence is not part of any Markov tree. The remaining $s_n$ are wavelet states at levels $j > 0$. For visual simplicity, a binary-tree is shown instead of a quad-tree, and the nodes representing the statistical parameters $\rho, \{\rho_j\}, p^1_{-1}, p^1_0, \{p^{00}_j\}, \{p^{11}_j\}$ are not shown.

further partitioned into several quad-trees, each with $J$ levels (see Fig. 1). We denote the indices of coefficients at level $j \in \{0, \ldots, J-1\}$ of the wavelet trees by $\mathcal{W}_j$, where $j = 0$ is the root. With a slight abuse of notation, we sometimes refer to the approximation coefficients using level $j = -1$.

Each transform coefficient $\theta_n$ is modeled using a (conditionally independent) Bernoulli-Gaussian prior pdf of the form

$$p(\theta_n \,|\, s_n) \;=\; s_n \mathcal{N}(\theta_n; 0, \sigma_n^2) + (1 - s_n)\delta(\theta_n), \qquad (2)$$

with $\delta(.)$ denoting the Dirac delta, $\mathcal{N}(x; \mu, v) \triangleq (2\pi v)^{-1/2} e^{-(x-\mu)^2/(2v)}$, and $s_n \in \{0, 1\}$ a hidden binary state. The states $\{s_n\}_{n \in \mathcal{A}}$ are assigned an apriori activity rate $\Pr\{s_n = 1\} = p^1_{-1}$, which is discussed further below. Meanwhile, the root wavelet states $\{s_n\}_{n \in \mathcal{W}_0}$ are assigned $\Pr\{s_n = 1\} = p^1_0$. Within each quad-tree, the states have a Markov structure. In particular, the activities of states at level $j > 0$ are determined by their parent's activities (at level $j - 1$) and the transition matrix

$$T_{j-1} \;\triangleq\; \begin{bmatrix} p^{00}_j & 1 - p^{00}_j \\ 1 - p^{11}_j & p^{11}_j \end{bmatrix}, \qquad (3)$$

where $p^{00}_j$ denotes the probability that a child's state equals 0 given that his parent's state equals 0, and $p^{11}_j$ denotes the probability that a child's state equals 1 given given that his parent's state equals 1. Finally, the signal variances $\{\sigma^2_n\}_{n \in \mathcal{W}_j}$ are assigned the same prior within each level $j \in \{-1, 0, \ldots, J-1\}$. The corresponding factor graph is shown in Fig. 2.

We take a "fully Bayesian" approach, modeling all statistical parameters as random variables and assigning them non-informative hyperpriors. For the precisions (i.e., inverse variances), Gamma hyperpriors are assumed:

$$\rho_j \;\sim\; \text{Gamma}(\rho_j; a_j, b_j) = \frac{1}{\Gamma(a_j)} b_j^{a_j} \rho_j^{a_j - 1} \exp(-b_j \rho_j) \quad (4)$$

$$\rho \;\sim\; \text{Gamma}(\rho; a, b) = \frac{1}{\Gamma(a)} b^a \rho^{a-1} \exp(-b\rho). \qquad (5)$$

For the activity rates and transition parameters, Beta hyperpriors are assumed:

$$p^1_0 \;\sim\; \text{Beta}(p^1_0; c, d) = \frac{\Gamma(c+d)}{\Gamma(c)\Gamma(d)} (p^1_0)^{c-1} (1 - p^1_0)^{d-1} \quad (6)$$

$$p^1_{-1} \;\sim\; \text{Beta}(p^1_{-1}; \underline{c}, \underline{d}) = \frac{\Gamma(\underline{c}+\underline{d})}{\Gamma(\underline{c})\Gamma(\underline{d})} (p^1_{-1})^{\underline{c}-1} (1 - p^1_{-1})^{\underline{d}-1} \quad (7)$$

$$p^{00}_j \;\sim\; \text{Beta}(p^{00}_j; c_j, d_j) = \frac{\Gamma(c_j+d_j)}{\Gamma(c_j)\Gamma(d_j)} (p^{00}_j)^{c_j - 1} (1 - p^{00}_j)^{d_j - 1} \quad (8)$$

$$p^{11}_j \;\sim\; \text{Beta}(p^{11}_j; \underline{c}_j, \underline{d}_j) = \frac{\Gamma(\underline{c}_j+\underline{d}_j)}{\Gamma(\underline{c}_j)\Gamma(\underline{d}_j)} (p^{11}_j)^{\underline{c}_j - 1} (1 - p^{11}_j)^{\underline{d}_j - 1} \quad (9)$$

## III. Image Reconstruction

To infer $\boldsymbol{\theta}$, we would like to compute the posterior

$$p(\boldsymbol{\theta} \,|\, \boldsymbol{y}) \;\cong\; \sum_{\boldsymbol{s}} p(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{s}) p(\boldsymbol{\theta}, \boldsymbol{s}) \qquad (10)$$

$$= \sum_{\boldsymbol{s}} \underbrace{p(\boldsymbol{s})}_{\triangleq h(\boldsymbol{s})} \prod_{n=1}^{N} \underbrace{p(\theta_n \,|\, s_n)}_{\triangleq f_n(\theta_n, s_n)} \prod_{m=1}^{M} \underbrace{p(y_m \,|\, \boldsymbol{\theta})}_{\triangleq g_m(\boldsymbol{\theta})}, \qquad (11)$$

where $\cong$ denotes equality up to a normalization constant. Here, $f_n(\theta_n, s_n)$ is specified by (2) and, due to the white Gaussian noise model, $g_m(\boldsymbol{\theta}) = \mathcal{N}(y_m; \boldsymbol{a}_m^T \boldsymbol{\theta}, \sigma^2)$, where $\boldsymbol{a}_m^T$ denotes the $m^{\text{th}}$ row of the matrix $\boldsymbol{A} \triangleq \boldsymbol{\Phi}\boldsymbol{\Psi}$.

### A. Loopy Belief Propagation

While exact computation of $p(\boldsymbol{\theta} \,|\, \boldsymbol{y})$ is computationally prohibitive, the marginal posteriors $\{p(\theta_n \,|\, \boldsymbol{y})\}$ can be efficiently approximated using *loopy belief propagation* (LBP) [10] on the factor graph of Fig. 2. In doing so, we also obtain the marginal posteriors $\{p(s_n \,|\, \boldsymbol{y})\}$. In fact, we simultaneously infer the statistical parameters $\rho, \{\rho_j\}, p^1_{-1}, p^1_0, \{p^{00}_j\}, \{p^{11}_j\}$, but—for simplicity—we treat them here as if they were fixed and known and detail the procedure by which they are learned in Section III-D.

In LBP, messages are exchanged between the nodes of the factor graph until convergence. Messages take the form of either pdfs or pmfs, as will be clear from the context. Intuitively, the messages flowing to/from a variable node can be interpreted as local beliefs about that variable. According to the *sum-product algorithm* [10], the message emitted by a variable node along a given edge is (a scaled version of) the product of the incoming messages on all other edges. Meanwhile, the message emitted by a function node along a given edge is (a scaled version of) the integral (or sum) of the product of the node's constraint function and the incoming messages on all other edges. The integration (or summation) is performed over all variables other than the one directly connected to the edge along which the message travels.

If our factor graph had no loops, then exact marginal posteriors could be computed using two passes of the sum-product algorithm [10]. Since our factor graph has loops, exact inference is known to be NP hard and LBP is not guaranteed to produce correct posteriors. Still, LBP has been successfully applied to many other problems, such as as turbo decoding and LDPC decoding in communications, and in this paper we demonstrate that LBP can be successfully applied to compressive imaging as well.

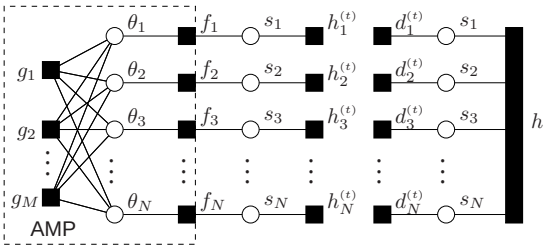


Fig. 3. The turbo approach yields a decoupled factor graph.

### *B. Message Scheduling: The Turbo Approach*

Given that our belief propagation is loopy, there exists considerable freedom as to how messages are scheduled. In this work, we adopt the "turbo" approach recently proposed in [11]. For this, we split the factor graph in Fig. 2 along the dashed line, noting that the left half exploits structure in the compressed observations and the right half exploits structure in the HMT. As a result, we obtain the two decoupled subgraphs in Fig. 3. We then alternate belief propagation on each of the two resulting sub-graphs, treating the likelihoods on $\{s_n\}$ generated from belief propagation on one sub-graph as priors on $\{s_n\}$ for the other sub-graph. In other words,

$$h_n^{(t)}(s_n) \;\triangleq\; \nu^{(t)}_{h\to s_n}(s_n) \tag{12}$$

$$d_n^{(t+1)}(s_n) \;\triangleq\; \nu^{t}_{f_n\to s_n}(s_n), \tag{13}$$

where $\nu^{(t)}_{A\to B}(.)$ denotes the message passed from node A to node B during the $t^{\text{th}}$ turbo iteration.

Borrowing terminology from the *turbo equalization* method used in communications [14], we refer to inference of $\{s_n\}$ using compressive-measurement structure as *sparsity pattern equalization* (SPE) and inference of $\{s_n\}$ using HMT structure as *sparsity pattern decoding* (SPD). Due to the tree structure of HMT, there are no loops in the SPD factor graph, and so SPD can be performed exactly using only two rounds of sum-product message passing [10]. The SPE factor graph is loopy, and so SPE performs several iterations of loopy belief propagation, internally, per turbo iteration. SPE details are described in the next subsection.

### *C. Sparsity Pattern Equalization via AMP*

We now discuss the message passing within SPE during a single turbo iteration $t$. The operations are invariant to $t$, and so we suppress the $t$-notation for brevity. As described above, SPE performs several iterations of loopy belief propagation per turbo iteration using the fixed priors $\lambda_n \triangleq h_n(s_n=1)$. Over the SPE iterations, the message $\nu_{f_n\to\theta_n}$ is fixed at

$$\nu_{f_n\to\theta_n}(\theta_n) \;=\; \lambda_n\mathcal{N}(\theta_n;0,\sigma_n^2)+(1-\lambda_n)\delta(\theta_n). \tag{14}$$

The dashed box in Fig. 3 shows the region of the factor graph on which messages are updated during the SPE iterations. This sub-graph can be recognized as the one that Donoho, Maleki, and Montanari used to derive their so-called *approximate message passing* (AMP) algorithms [12]. Although they mostly focus on the case of Laplacian (or least-favorable) $\boldsymbol{\theta}$-priors, they outlined the case of general priors in [15]. We derive the details of Bernoulli-Gaussian AMP below, using a superscript-$i$ to denote the SPE iteration.

According to the sum-product algorithm, the fact that $\nu_{f_n\to\theta_n}$ is non-Gaussian implies that $\nu^i_{\theta_n\to g_m}$ is also non-Gaussian, and this complicates the calculation of the subsequent messages $\nu^i_{g_m\to\theta_n}$. However, for large $N$, the combined effect of $\{\nu^i_{\theta_n\to g_m}\}_{n=1}^N$ can be approximated as Gaussian, at which point it becomes sufficient to parameterize each message $\nu^i_{\theta_n\to g_m}$ by its mean and variance $(\mu^i_{mn}, v^i_{mn})$ [15]:

$$\mu^i_{mn} \;\triangleq\; \textstyle\int_{\theta_n}\theta_n\,\nu^i_{\theta_n\to g_m}(\theta_n) \tag{15}$$

$$v^i_{mn} \;\triangleq\; \textstyle\int_{\theta_n}(\theta_n-\mu^i_{mn})^2\nu^i_{\theta_n\to g_m}(\theta_n). \tag{16}$$

Combining the fact that

$$\prod_q \mathcal{N}(\theta;\mu_q,v_q) \;\cong\; \mathcal{N}\left(\theta;\frac{\sum_q \mu_q/v_q}{\sum_q v_q^{-1}},\frac{1}{\sum_q v_q^{-1}}\right) \tag{17}$$

with $g_m(\boldsymbol{\theta})=\mathcal{N}(y_m;\boldsymbol{a}_m^T\boldsymbol{\theta},\sigma^2)$, it can be shown that

$$\nu^i_{g_m\to\theta_n}(\theta_n) \;=\; \mathcal{N}\left(\theta_n;\frac{z^i_{mn}}{A_{mn}},\frac{c^i_{mn}}{A^2_{mn}}\right) \tag{18}$$

$$z^i_{mn} \;\triangleq\; y_m-\textstyle\sum_{q\neq n}A_{mq}\mu^i_{qm} \tag{19}$$

$$c^i_{mn} \;\triangleq\; \sigma^2+\textstyle\sum_{q\neq n}A^2_{mq}v^i_{qm}. \tag{20}$$

The quantities $\mu^{i+1}_{mn}$ and $v^{i+1}_{mn}$ are then calculated from

$$\nu^{i+1}_{\theta_n\to g_m}(\theta_n) \;\cong\; \nu_{f_n\to\theta_n}(\theta_n)\textstyle\prod_{l\neq m}\nu^i_{g_l\to\theta_n}(\theta_n), \tag{21}$$

where, using (17), the product term in (21) becomes

$$\cong\mathcal{N}\left(\theta_n;\frac{\sum_{l\neq m}A_{ln}z^i_{ln}/c^i_{ln}}{\sum_{l\neq m}A^2_{ln}z^i_{ln}/c^i_{ln}},\frac{1}{\sum_{l\neq m}A^2_{ln}/c^i_{ln}}\right). \tag{22}$$

Assuming that the entries of $\boldsymbol{A}$ were generated from an i.i.d distribution with unit-normalized columns, and $M$ is adequately large, we have $\sum_{l\neq m}A^2_{ln}\approx\sum_{l=1}^M A^2_{ln}=1$ and $c^i_{ln}\approx c^i_n\triangleq\frac{1}{M}\sum_{m=1}^M c^i_{mn}$. In this case, (21) reduces to

$$\nu^{i+1}_{\theta_n\to g_m}(\theta_n) \;\cong\; \big(\lambda_n\mathcal{N}(\theta_n;0,\sigma_n^2)+(1-\lambda_n)\delta(\theta_n)\big) \times\mathcal{N}(\theta_n;\xi^i_{nm},c^i_n) \tag{23}$$

$$\xi^i_{nm} \;\triangleq\; \textstyle\sum_{l\neq m}A_{ln}z^i_{ln}, \tag{24}$$

and the mean and variance of $\nu^{i+1}_{\theta_n\to g_m}(\theta_n)$ become

$$\mu^{i+1}_{nm} \;=\; \alpha_n(c^i_n)\xi^i_{nm}/(1+\gamma^i_{nm}) \tag{25}$$

$$v^{i+1}_{nm} \;=\; \gamma^i_{nm}(\mu^{i+1}_{nm})^2+\mu^{i+1}_{nm}c^i_n/\xi^i_{nm} \tag{26}$$

$$\gamma^i_{nm} \;\triangleq\; \beta_n(c^i_n)\exp(-\zeta_n(c^i_n)(\xi^i_{nm})^2), \tag{27}$$

where

$$\alpha_n(c)\triangleq\tfrac{\sigma_n^2}{c+\sigma_n^2},\;\beta_n(c)\triangleq\tfrac{1-\lambda_n}{\lambda_n}\sqrt{\tfrac{c+\sigma_n^2}{c}},\;\zeta_n(c)\triangleq\tfrac{\sigma_n^2}{2c(c+\sigma_n^2)}.$$

For the first turbo iteration, we set $z^0_{mn}=y_m$ and $c^0_n\gg\sigma_n^2$ for all $m,n$. For subsequent turbo iterations, SPE is initialized using the final SPE values from the previous turbo iteration.

According to the sum-product algorithm, the estimated $\theta_n$-posterior at the $i^{\text{th}}$-SPE iteration is

$$\hat{p}^i(\theta_n\,|\,\boldsymbol{y}) \;\cong\; \nu_{f_n\to\theta_n}(\theta_n)\textstyle\prod_{l=1}^M\nu^i_{g_l\to\theta_n}(\theta_n), \tag{28}$$

whose mean and variance determine the $i^{\text{th}}$-iteration MMSE estimate of $\theta_n$ and its variance, respectively. Noting that the difference between (28) and (21) is only the inclusion of the $m^{\text{th}}$ product term, the mean and variance become

$$\mu_n^{i+1} \;=\; \alpha_n(c_n^i)\xi_n^i/(1+\gamma_n^i) \tag{29}$$
$$v_n^{i+1} \;=\; \gamma_n^i(\mu_n^{i+1})^2 + \mu_n^{i+1}c_n^i/\xi_n^i \tag{30}$$
$$\xi_n^i \;\triangleq\; \textstyle\sum_{l=1}^M A_{ln} z_{ln}^i \tag{31}$$
$$\gamma_n^i \;\triangleq\; \beta_n(c_n^i)\exp(-\zeta_n(c_n^i)(\xi_n^i)^2). \tag{32}$$

Similarly, the $i^{\text{th}}$-SPE iteration $s_n$-posterior estimate is

$$\hat{p}^i(s_n \,|\, \boldsymbol{y}) \;\cong\; \nu_{f_n\to s_n}^{i-1}(s_n)\nu_{h_n\to s_n}(s_n), \tag{33}$$

where

$$\nu_{f_n\to s_n}^i(s_n) \;\cong\; \int_{\theta_n} f_n(\theta_n, s_n)\prod_{l=1}^M \nu_{g_l\to\theta_n}^i(\theta_n). \tag{34}$$

Since $f_n(\theta_n, s_n) = s_n\mathcal{N}(\theta_n; 0, \sigma_n^2) + (1-s_n)\delta(\theta_n)$, it can be seen that SPE's extrinsic log-likelihood ratio (LLR) is

$$L_n^i \triangleq \ln\frac{\nu_{f_n\to s_n}^i(s_n=1)}{\nu_{f_n\to s_n}^i(s_n=0)} = \frac{1}{2}\ln\frac{c_n^i}{c_n^i+\sigma_n^2} + \zeta_n(c_n^i)(\xi_n^i)^2. \tag{35}$$

The message update equations derived thus far updates $\mathcal{O}(MN)$ variables per iteration, which is inconvenient for large $M$ and $N$. We now summarize the "first-order" AMP algorithm [15] for the non-identical Bernoulli-Gaussian prior (14), which updates only $\mathcal{O}(N)$ variables per iteration:

$$\xi_n^i \;=\; \textstyle\sum_{m=1}^M A_{mn} z_m^i + \mu_n^i \tag{36}$$
$$\mu_n^{i+1} \;=\; F_n(\xi_n^i; c^i) \tag{37}$$
$$v_n^{i+1} \;=\; G_n(\xi_n^i; c^i) \tag{38}$$
$$z_m^{i+1} \;=\; y_m - \textstyle\sum_{n=1}^N A_{mn}\mu_n^i + \frac{z_m^i}{M}\sum_{n=1}^N F_n'(\xi_n^i; c^i) \tag{39}$$
$$c^{i+1} \;=\; \sigma^2 + \tfrac{1}{M}\textstyle\sum_{n=1}^N v_n^{i+1}, \tag{40}$$

where $F_n(.;.)$, $G_n(.;.)$ and $F_n'(.;.)$ are defined as

$$F_n(\xi; c) \;=\; \frac{\alpha_n(c)}{1+\beta_n(c)\exp(-\zeta_n(c)(\xi)^2)}\xi \tag{41}$$
$$G_n(\xi; c) \;=\; \beta_n(c)\exp(-\zeta_n(c)(\xi)^2)F_n(\xi; c)^2 + c\,\xi^{-1}F_n(\xi; c) \tag{42}$$
$$F_n'(\xi; c) \;=\; \frac{\alpha_n(c)}{[1+\beta_n(c)\exp(-\zeta_n(c)(\xi)^2)]^2}\big[1 + \beta_n(c)\exp(-\zeta_n(c)(\xi)^2)\left(1+2\zeta(c_n)(\xi)^2\right)\big]. \tag{43}$$

### *D. Learning the Statistical Parameters*

We now briefly summarize how the precisions $\{\rho_j\}$ are learned. Say that, just after the $t^{\text{th}}$ turbo iteration, $\mathcal{S}_j \triangleq \{n \in \mathcal{W}_j : L_n^\infty > 0\}$ contains indices of the level-$j$ coefficients though to be "large," and $K_j \triangleq |\mathcal{S}_j|$ is its cardinality. Here, "$\infty$" is used to denote the final SPE iteration. Then, level-$j$'s precision hyper-parameters and mean are updated via

$$\hat{a}_j^{(t+1)} \;=\; a_j + K_j/2 \tag{44}$$
$$\hat{b}_j^{(t+1)} \;=\; b_j + \tfrac{1}{2}\textstyle\sum_{n\in\mathcal{S}_j}(\mu_n^\infty)^2 \tag{45}$$
$$\mathrm{E}[\hat{\rho}_j^{(t+1)}] \;=\; \hat{a}_j^{(t+1)}/\hat{b}_j^{(t+1)}, \tag{46}$$

and, for turbo iteration $t+1$, the variances $\{\sigma_n^2\}_{n\in\mathcal{W}_j}$ are set to $1/\mathrm{E}[\hat{\rho}_j^{(t+1)}]$. The precision $\rho$ (and noise variance $\sigma^2$) are learned similarly from the SPE-estimated residual.

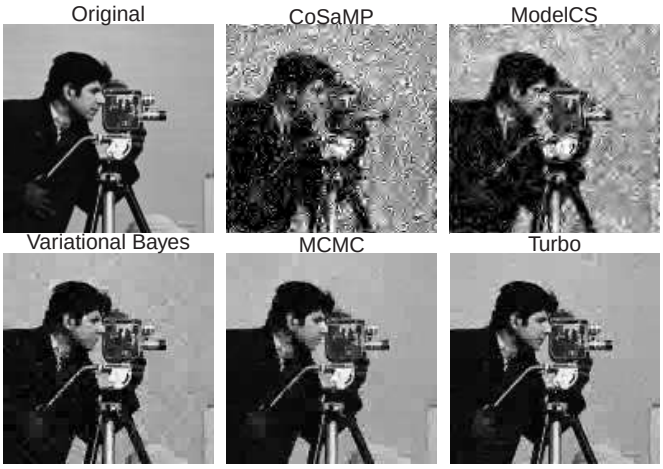


Fig. 4. Reconstruction from $M = 5000$ observations of a $128 \times 128$ (i.e., $N = 16384$) section of the cameraman image using i.i.d Gaussian $\boldsymbol{\Phi}$.

We now briefly summarize how the transition probabilities $\{p_j^{11}\}$ are learned. Say that, just after the $t^{\text{th}}$ turbo iteration, of the $K_j$ level-$j$ coefficients thought to be "large," $C_j$ have children that are also thought to be large (based on the SPE-generated LLRs). Then the corresponding hyperparameters and transition probability are updated as follows:

$$\hat{\underline{c}}_j^{(t+1)} \;=\; \underline{c}_j + C_j \tag{47}$$
$$\hat{\underline{d}}_j^{(t+1)} \;=\; \underline{d}_j + K_j - C_j \tag{48}$$
$$(\hat{p}_j^{11})^{(t+1)} \;=\; \hat{\underline{c}}_j^{(t+1)}\big(\hat{\underline{c}}_j^{(t+1)} + \hat{\underline{d}}_j^{(t+1)}\big)^{-1}. \tag{49}$$

The parameters $p_0^1$, $p_{-1}^1$, and $\{p_j^{00}\}$ are learned similarly.

## IV. Numerical Results

The proposed turbo approach to compressive imaging was compared to several others: CoSaMP [7], ModelCS [3], variational Bayes [6], and MCMC [5]. All numerical experiments were performed on $128\times128$ (i.e., $N = 16384$) grayscale images using a wavelet decomposition with $4$ quad-tree levels, yielding $8^2 = 64$ approximation coefficients and $3\times8^2 = 192$ Markov trees. In all cases, the matrix $\boldsymbol{\Phi}$ had i.i.d Gaussian entries, and $M = 5000$ noiseless measurements were used.

For our turbo scheme, we learned the statistical parameters as described in Section III-D and applied the same values to all Markov trees. The Gamma hyperparameters were set as $a = 1 = a_j\ \forall j$, $b = 1\times10^{-10}$, and $[b_0, \ldots, b_4] = [50, 4, 4.5, 1.5, 0.45]$. The Beta parameters were chosen with $c{+}d{=}1$ and $c_j{+}d_j{=}1\ \forall j$ such that $\mathrm{E}\{p_j^{00}\}{=}0.9$, $\mathrm{E}\{p_j^{11}\}{=}0.5$, $\mathrm{E}\{p_0^1\}{=}0.9$, and $\mathrm{E}\{p_{-1}^1\}{=}0.9$.

Fig. 4 shows a $128{\times}128$ section of the "cameraman" image along with the recovered images of the various algorithms. CoSaMP, which leverages only simple sparsity, and ModelCS, which models PAS deterministically, both perform poorly. The HMT-based schemes (VB, MCMC, and turbo) all perform significantly better, with MCMC and turbo performing best.

For a quantitative comparison, we measured average reconstruction performance over a suite of images from a *Microsoft*

Fig. 5. A sample image from each of the 20 types in the Microsoft database. Image statistics were found to vary significantly from one type to another.

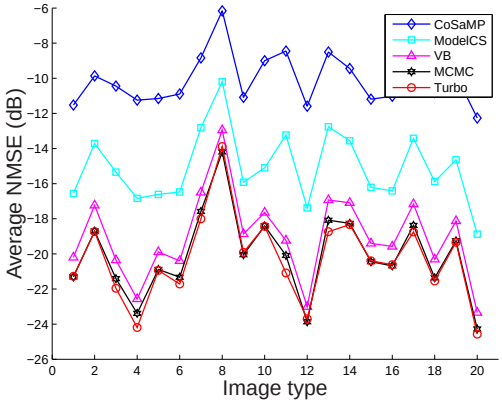


Fig. 6. Average NMSE for each image type.

*Research Object Class Recognition* database[2] that contains 20 types of images (see Fig. 5) with roughly 30 images of each type. For each image type, we computed the average normalized mean squared error (NMSE) $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2/\|\boldsymbol{x}\|_2^2$ as well as the average computation time on a 2.5 GHz PC. These results are reported in Figures 6 and 7, and the global averages (over all 591 images) are reported in Table I. We observe that the proposed turbo algorithm outperforms all the other algorithms in both reconstruction NMSE and computation time. Relative to the turbo algorithm, CoSaMP and ModelCS both have significantly higher NMSE and require significantly higher computation. MCMC yields NMSE that is very close to that of the turbo algorithm, but is 14 times slower. Variational Bayes yields NMSE that is 1 dB worse than that of the turbo algorithm, while taking twice as long to run.

Although the experiments reported here assume noiseless observations, we have performed successful noisy compressive imaging as well. Even in the noiseless case, though, we find $\sigma^2 > 0$ useful in conjunction with our sparse signal model (2); natural images are never truly sparse, and so the very small coefficients get absorbed into the noise vector $\boldsymbol{w}$.

## V. Conclusion

We proposed a new approach to HMT-based compressive imaging based on loopy belief propagation, leveraging a turbo message passing schedule and the AMP algorithm of Donoho, Maleki, and Montanari. We then tested our algorithm on a suite of 591 natural images and found that it outperformed the state-of-the-art approach while halving its runtime.

[2]From each image in the "Pixel-wise labelled image database v2" at *http://research.microsoft.com/en-us/projects/objectclassrecognition*, we extracted the center $128\times128$ section. What we refer to as an "image type" is referred to as a "row" in the database.

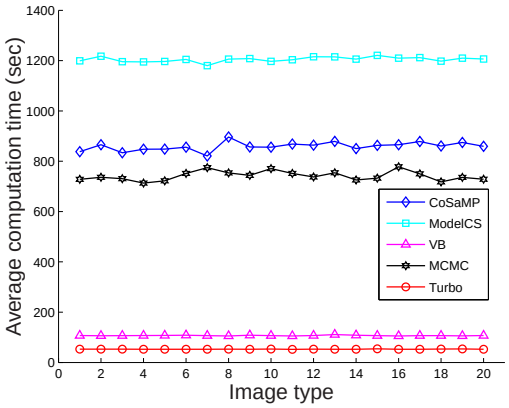


Fig. 7. Average computation time for each image type.

| Algorithm | Computation Time (sec) | NMSE (dB) |
|---|---|---|
| CoSaMP | 859 | -10.13 |
| ModelCS | 1205 | -15.10 |
| Variational Bayes | 107 | -19.04 |
| MCMC | 742 | -20.10 |
| Turbo | 53 | -20.31 |

TABLE I
NMSE AND COMPUTATION TIME AVERAGED OVER 591 IMAGES.

## References


[1] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008.

[2] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed. San Diego, CA: Academic Press, 2008.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[4] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal recontruction using a hidden Markov tree model," in *Proc. IEEE Int. Conf. Acoust. Speech & Signal Process.*, Las Vegas, NV, Apr. 2008, pp. 5137–5140.

[5] L. He and L. Carin, "Exploiting structure in wavelet-based Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3488–3497, Sep. 2009.

[6] L. He, H. Chen, and L. Carin, "Tree-structured compressive sensing with variational Bayesian analysis," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 233–236, Mar. 2010.

[7] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Computational Harmonic Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

[8] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

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

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[11] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. Conf. Inform. Science & Syst.*, Princeton, NJ, Mar. 2010.

[12] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[13] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *arXiv:1001.3448*, Jan. 2010.

[14] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 67–80, Jan. 2004.

[15] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inform. Theory Workshop*, Cairo, Egypt, Jan. 2010.