A hierarchical Bayesian-MAP approach to inverse problems in imaging

To cite this article: Raghu G Raj 2016 Inverse Problems 32 075003

View the article online for updates and enhancements.

Related content

- Expectation propagation for continuous time stochastic processes
  Botond Cseke, David Schnoerr, Manfred Opper et al.

- Inverse problems with Poisson data: statistical regularization theory, applications and algorithms
  Thorsten Hohage and Frank Werner

- Estimating the underlying signal waveform, noise covariance and synchronization
  J Martinsson
A hierarchical Bayesian-MAP approach to inverse problems in imaging

Raghu G Raj
US Naval Research Laboratory, Washington DC, 20375, USA
E-mail: raghu.raj@nrl.navy.mil

Received 30 April 2015, revised 11 January 2016
Accepted for publication 11 February 2016
Published 12 May 2016

Abstract
We present a novel approach to inverse problems in imaging based on a hierarchical Bayesian-MAP (HB-MAP) formulation. In this paper we specifically focus on the difficult and basic inverse problem of multi-sensor (tomographic) imaging wherein the source object of interest is viewed from multiple directions by independent sensors. Given the measurements recorded by these sensors, the problem is to reconstruct the image of the object with a high degree of fidelity. We employ a probabilistic graphical modeling extension of the compound Gaussian distribution as a global image prior into a hierarchical Bayesian inference procedure. Since the prior employed by our HB-MAP algorithm is general enough to subsume a wide class of priors including those typically employed in compressive sensing (CS) algorithms, HB-MAP algorithm offers a vehicle to extend the capabilities of current CS algorithms to include truly global priors. After rigorously deriving the regression algorithm for solving our inverse problem from first principles, we demonstrate the performance of the HB-MAP algorithm on Monte Carlo trials and on real empirical data (natural scenes). In all cases we find that our algorithm outperforms previous approaches in the literature including filtered back-projection and a variety of state-of-the-art CS algorithms. We conclude with directions of future research emanating from this work.

Keywords: inverse problems, imaging, compressive sensing, Bayesian, probabilistic graphical models, hierarchical Bayes, sparse reconstruction

(Some figures may appear in colour only in the online journal)
1. Introduction

Imaging typically refers to a class of inverse problems wherein the central objective is to form a two-dimensional representation (the image) of an object or scene of interest that is being sensed by one or more sensors each furnishing complementary but correlated sources of information, and each of which is potentially contaminated by noise and distortion. What distinguishes this from the more general class of inverse and regression problems is that images originating from typical empirical sources (such as natural scenes) have a special structure that makes it possible to inject informed prior models into the inference process that can considerably enhance the quality of the reconstructed image. Accordingly a wide variety of analytical and statistical approaches have proliferated over the past several decades in diverse imaging applications such as radar [1–3], medical [4, 5], acoustic [45] and astronomical imaging [6].

Figure 1 shows a high-level view of an imaging system which consists of a central receiver that receives the signals resulting from the reflection of waveforms (transmitted and received at diverse sensing angles) from the object/scene of interest, followed by a preprocessor (that suitably conditions the signal by factoring out signal variations that are due to various systemic factors)–the output of which is finally processed by the central imaging algorithm. The pre-processing unit encapsulates much of the system- (and domain-) level expertise associated with the application domain of interest. For example, in radar imaging [1–3] typical pre-processing steps include pulse compression (generalized matched filtering) and range alignment (which corrects for the different time delays associated with the transmission of pulse sequences through the propagation media) which collectively ensure that the effective system can be approximately modeled as a linear shift-invariant transformation—thus rendering the resulting signals amenable to efficient Fourier analysis tools.

In this paper we abstract out these and other system-level issues (such as synchronization of different sensors for the case of multistatic radar systems) and focus instead on the basic problem of optimum image reconstruction conditioned on the aforementioned pre-processing operations. In this setting, the effective forward model can be captured surprisingly effectively by the following linear system model:

$$y = \Psi \Phi c + v,$$

where:
\( y \in \mathbb{R}^m \) is the vectorized measurements,
\( \Psi \in \mathbb{R}^{m \times d} \) is the effective measurement matrix,
\( I = \Phi c \in \mathbb{R}^n \) is the vectorized (unknown) image of interest,
\( \Phi \in \mathbb{R}^{d \times n} \) is the dictionary in which we choose to represent the unknown image \( I \),
\( c \in \mathbb{R}^n \) is the unknown coefficient vector which is the object of our inference algorithm,
\( v \in \mathbb{R}^m \) is the effective measurement noise.

The central goal of this paper is to solve equation (1) i.e. to estimate the coefficient vector \( c \) given the observed measurements \( y \). The dictionary (matrix) \( \Phi \), with respect to (w.r.t.) which the image is represented, is chosen \textit{a priori}. In this paper we choose \( \Phi \) to be any class of wavelet dictionaries [30]; in section 6 we describe how this can be extended to much more general class of overcomplete dictionaries.

We now explain the action of the sensing matrix \( \Psi \) in more detail. In an imaging system the measurement matrix \( \Psi \) is effectively (and metaphorically) the ‘lens’ through which the source object is sensed/viewed. For optical imaging systems, for example, \( \Psi \) is typically a blurring operator. Although the techniques described in this paper can be fruitfully deployed to invert these and indeed arbitrary sensing matrices—we demonstrate an example involving a random sensing matrix in section 5.1 wherein our techniques show superior results—in this paper we primarily focus on inverting the Radon transform since it underlies most imaging applications including microwave, acoustic and medical imaging modalities [1, 4, 7].

The Radon transform of a function \( I(x) \) (i.e. in our case, the underlying source object to be imaged) with respect to angle \( \theta \) can be defined as follows:

\[
I(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x) \delta(r - \xi_\theta \cdot x) \, dx \, \equiv \mathcal{R}_\theta \{I\},
\]

where \( \xi_\theta = [\cos \theta, \sin \theta]^T \) is a unit vector in the direction at which the sensor is pointing, and \( x = [x_1, x_2]^T \) is the local image coordinate of the source object.

An imaging system typically interacts with the environment by transmitting a waveform (pulse) \( \psi(t) \) in various directions \( \theta = [\theta_m]_{m=1}^M \) and receiving the backscattered signal. The transmitted waveform interacts with the source object after which the backscattered waveform is measured and recorded at the sensor (assumed to be operating in duplex mode). For the imaging modalities described above, the received waveform w.r.t. the sensing angle \( \theta_m \) (under suitable conditions such as ‘far-field approximation’ and suitable pre-processing operations such as pulse compression in radar imaging [2, 23]) can be modeled as follows:

\[
y_m = \psi^* \mathcal{R}_{\theta_m} \{I\},
\]

where \( ^* \) denotes the convolution operator. We observe that for the special case in which \( \psi(t) = \delta(t) \) (i.e. Dirac delta function), equation (3) reduces to (2). Given this, the aggregation of the backscattered responses from the various sensing angles \( \theta = [\theta_m]_{m=1}^M \) can thus be succinctly encapsulated in equation (1).

A plethora of different approaches have been proposed to solve for \( c \) in equation (1). The traditional approach is the Fourier backprojection (FBP) algorithm [7] (wherein the underlying dictionary \( \Phi \) comprises of Fourier atoms). More recently, a variety of compressive-sensing (CS)/sparsity-based-reconstruction approaches including 11-Ls [8], OMP [9], ROMP [10], CoSaMP [11], and Bayesian compressive sensing (BCS) [41] have been successfully applied for this purpose.

In this paper we present novel algorithmic techniques and framework which encompasses a large class of such CS and sparse reconstruction approaches [8–18, 39–41] as special cases and which, as we demonstrate in section 5, yields state-of-the-art performance.
We expound our hierarchical Bayesian-MAP (HB-MAP) paradigm in detail in section 2; describe our HB-MAP algorithm and details of its derivation in section 3. In section 4 we delve into some of the implementation details for enabling efficient gradient and Newton calculations for Type-II inference; and provide detailed experimental results in section 5.

Although the focus of this paper is on imaging, our formulation (including our choice of prior model) is broad enough to make it an attractive and effective tool for a much wider class of statistical inverse problems. We discuss this in more depth in section 6.

2. Hierarchical Bayes and the graphical compound Gaussian (CG) model

From (1) we have that the Bayesian-MAP\(^1\) estimate of \(c\) is given as follows:

\[
c^* = \argmax_c \log P(c|y), \quad (4)
\]

\[
\propto \argmax_c \{ \log P(y|c) + \log P(c) \}, \quad (5)
\]

\[
= \argmin_c \{ ||y - \Psi \Phi c||_2^2 - \log P(c) \}, \quad (6)
\]

where in (6) we assume that \(v \sim \mathcal{N}(0, \Sigma_v = I)\).

The distribution \(P(c)\) encapsulates the statistical prior knowledge that we have about the scene structure. For the specific choice of a Laplacian, i.e. \(P(c) = \exp(-\lambda ||c||_1)\), (4) reduces to the well-known LASSO [39] (closely related to Basis Pursuit [40]) algorithm:

\[
c^* = \argmin_c \{ ||y - \Psi \Phi c||_2^2 + \lambda \ ||c||_1 \}. \quad (7)
\]

The various CS algorithms in the literature [8–18, 39–41], which amount to different computational implementations of this idea, have enjoyed tremendous success because images derived from empirical sources typically admit sparse structure in wavelet-like dictionaries which, when promoted by the Laplace prior, provably result in robust estimates of the underlying sparse coefficients [8–18].

In reality however the Laplacian distribution is not rich enough for modeling the statistics of wavelet coefficients of natural scenes when sensed by optical [19–21] or radar [22, 23] sensors. In this paper we propose to use a probabilistic graphical modeling extension [21] of the CG distribution as a candidate for \(P(c)\). Doing so allows us to subsume many of the well-known distributions in the literature including the Laplacian, generalized Gaussian, alpha-stable distribution etc [21]. We show how the Bayesian-MAP estimation problem, under the CG prior, naturally lends itself to a hierarchical Bayes formulation that can yield superior performance to traditional CS algorithms.

First we briefly describe the global CG model and then proceed to discuss its incorporation into the HB framework for the purposes of solving equation (1).

\(^1\) MAP refers to maximum a posteriori estimation [42]. The prefix Bayesian is added to emphasize the fundamental Bayesian interpretation in this paper wherein probabilities are assigned without a corresponding notion of sampling or frequency [24, 25]. Furthermore although our method, as explained in section 3, also produces covariance estimates (which could be used to determine ‘Bayesian/confidence intervals’) we confine ourselves to utilizing point estimates in this paper.
2.1. Graphical CG model

We model \( c \) (in equation (1)) as a random vector\(^2\) that can be decomposed into the following (Hadamard) product form [20, 21]:

\[
c = z \odot u
\]

such that:

(i) \( u \sim \mathcal{N}(0, \, P_u) \)

\( z = h(x) \)

\( x \) follows a multi-scale Gaussian tree structure described in more detail below,

(ii) \( u \) and \( z \) are independent random variables,

(iii) \( \mathbb{E}[z^2] = 1 \),

(iv) \( h \) is a nonlinearity (which ultimately controls the sparse structure of \( c \)).

The multi-scale Gaussian tree structure in (i) is based on a quad-tree decomposition of the image space [21, 26] as shown in figure 2. Each level of the quad-tree decomposition corresponds to a different scale of the wavelet representation [30] of the image wherein each pixel at level \( i \) corresponds to \( d \) different wavelet coefficients corresponding to that location. For example, when choosing dictionary \( \Phi \) to be any family of orthogonal wavelets, each node of the multi-scale Gaussian tree contains a vector of size \( d = 4 \) coefficient corresponding to the familiar LL, LH, HL, HH sub-bands of the wavelet decomposition. In this paper, for simplicity, we confine ourselves to the multi-scale tree structure which is defined by the following equations [26]:

\(^2\) Since we espouse a Bayesian interpretation in this paper, we take some liberties in abusing notation by not carefully distinguishing between upper and lower case scripts—the former of which is usually reserved for random variables.
\[ x(s) = A(s)x(par(s)) + B(s)w(s), \]  
(9)

\[ P_s(s) = A(s)P_{par}(s)A(s)^T + Q(s), \]  
(10a)

\[ P_s(s, t) = \Gamma(s, s\Lambda t)P_s(s\Lambda t)\Gamma^T(t, s\Lambda t), \]  
(10b)

where \( s \) is a node of the tree and \( par(s) \) refers to the parent of \( s; \ w(s) \sim \mathcal{N}(0, I), \)
\( Q(s) = B(s)B^T(s); \) the symbol \( s\Lambda t \) refers to the closest common ancestor to nodes \( s \) and \( t \) on a tree; and where \( \Gamma \) is a state transition matrix defined recursively as follows:

\[ \Gamma(s, s) = I, \]
\[ \Gamma(s, t) = A(s)\Gamma(par(s), t). \]

These coarse-to-fine dynamics defined by (9) implicitly define a probabilistic graphical model in which edges between parent-child nodes represent a jointly Gaussian dependency structure.

The structure of the above model (8)–(10) has been proposed [21] in order to accommodate the salient properties of natural image statistics (stemming from the seminal observations of Simoncelli [19]):

(a) The statistics of wavelet coefficients can be modeled effectively by the CG model [20, 21] (and related observations in sea-clutter [22] and radar images [23]). This property is of course ensured by the model (8).

(b) Given the model (8) and property (ii), we have that the covariance of a coefficient and its parent is as follows:

\[ \text{cov}[c(s), c(par(s))] = E[h(x(s))h^T(par(x(s)))] \odot \text{cov}[u(s)u^T(par(s))]. \]

(c) The fact that the wavelet coefficients of images across scales tend to be decorrelated can be enforced by constraining the Gaussian field \( u \) to be corresponding white noise process across the multiscale tree. Furthermore, the variance of the wavelet coefficients across scales follows a self-similar structure. These properties can be accommodated by modeling:

\[ u(s) = D(s)\zeta(s), \quad \zeta(s) \sim \mathcal{N}(0, I) \]

such that:

\[ D(s) = 2^{-\gamma m(s)} \]

where \( m(s) \) is the scale corresponding to node \( s \), and \( \gamma > 0 \) is a constant. From properties (i)–(iii) it follows that the variance of \( c \) is controlled by the \( u \)-field.

(d) In spite of the above decorrelation across scales, the strong non-Gaussian local dependence of wavelet coefficients that is observed in natural images [20, 21] is captured by the tree dependency structure in \( x \) (and correspondingly, via (i), the \( z \) random field).

(e) Finally as mentioned above, the sparse structure of the wavelet coefficients is enforced by the nonlinearity \( h \).

In our deployment of the graphical CG model for the purposes of solving equation (1), we make a number of simplifying assumptions similar to those in [21]:

Firstly we assume that \( A(s) = A \cdot I_d \) and \( B(s) = B \cdot I_d \forall s \) where, \( A, B \in \mathbb{R}, \) and \( I_d \) denotes the identity matrix of size \( d \). We assume therefore that the tree structure corresponding to \( x \) is homogeneous. Given this we have from (9) that:

\[ P_s(s) = \frac{B^2}{1 - A^2}. \]  
(11)

Letting \( A \equiv \mu \) and \( B \equiv \sqrt{1 - \mu^2} \), we have that \( P_s(s) = I_d \forall s \). Thus a single parameter \( \mu \) controls the inter-scale dependency structure of the wavelet coefficients.
Secondly, although many different sparsity-inducing nonlinearities $h$ can be incorporated, we have found the following nonlinearity to be particularly useful in terms of analytical properties such as smoothness with respect to the sparsity controlling parameter $\alpha$:

$$h(x) = \sqrt{\exp(x/\alpha)}.$$  

From (12) we can see that the sparsity level of the signal is inversely proportional to $\alpha$.

As mentioned above the parameter $\gamma$ controls the exponential distribution of signal variance (power) across different scales. In this paper for simplicity we set $\gamma$ to zero i.e. assuming a flat distribution of power across scales. We leave the optimization of $\gamma$ for future work.

Thus we employ two parameters, $\alpha$ and $\mu$, to control the properties of the graphical CG distribution. Note that when $\mu = 0$, our model reduces to a i.i.d. CG model which subsumes the naïve Bayes statistical model underlying standard CS algorithms. We remark that the BCS approach of Ji–Carin [41] also employs a naïve Bayes model and is therefore subsumed as a special case of our more general global CG model.

We further remark that for the special case of a latent $z$-vector that comprises of unity values at as sparse number of locations and zero everywhere else; the resulting CG model reduces to yet another popular class of statistical models underlying sparsity based reconstruction algorithms [12].

Thus the multiscale graphical CG model employed in this paper is general enough to accommodate a wide class of prior models.

The novel contribution of this paper is to algorithmically solve, for general sensing matrices $\Psi$ (especially for imaging applications), the fundamental equation (1) subject to the above global CG prior placed on the signal to be estimated. This work therefore represents a generalization of the contributions in [21] which only focuses on the denoising problem (i.e. where $\Psi$ is an identity matrix). Parameter estimation is described in detail in section 3. We next describe the hierarchical Bayes-MAP methodology and its application to the CG distribution.

**Algorithm 1.** Steepest descent algorithm for Type-II estimation

1. Initialize $x^0 = \text{abs}(h^{-1}(\hat{\Psi}^T y))$ (where $\hat{\Psi} = \Psi \Phi$) and $n = 0$

2. Calculate the descent direction $d^n$ either by gradient descent or by Newton descent methods; where:
   
   $d^n = -\nabla f(x^n)$ for gradient descent
   
   $d^n = -M^{-1}\nabla f(x^n)$ for Newton descent
   
   where, $M \equiv \text{Closest PSD approx of the Hessian } \nabla^2 f(x^n)$

3. Update the $x$ field:
   
   $x^{n+1} = x^n + \lambda d^n$

   where, $\lambda$ is chosen by an Armijo line search

4. $n \leftarrow n + 1$

5. Repeat steps (2)–(3) until convergence

6. Return $x^*$

The high-level algorithm for performing Type-II estimation for inferring the $x$-field is described. The equations for calculating the gradient and Hessian of the cost function $f$ is given in appendix. Further details are in section 3.2.
Algorithm 2. Type-I estimation

1. Let \( z^* \) be the solution of the Type-II estimation procedure given the measurements \( y \), the sampling matrix \( \Psi \), and dictionary \( \Phi \).
2. Calculate \( \bar{\Lambda}_L = \text{diag}(\Lambda_{\Psi, \Phi}) \) for a threshold \( \lambda \) in equation (27) and let \( \Lambda_R = \text{diag}(z^*) \).
3. Solve for \( u^* \):
   \[
   u^* = L^{-1}R
   \]
   where,
   \[
   L = \bar{\Lambda}_L (\Psi^T \Sigma^{-1} \Psi + \Lambda_R) \Lambda_R U
   \]
   \[
   R = \bar{\Lambda}_U \Psi^T \Sigma^{-1}
   \]
4. Return \( u^* \).

The high-level algorithm for performing Type-I estimation for inferring the \( u \)-field is described. The determination of the threshold \( \lambda \) from the histogram of \( z^* \) is described in section 5.

2.2. Hierarchical Bayes-MAP strategy

Given the CG model in (8), the probability density function (pdf) of the wavelet field \( c \) is given by:
\[
P(c) = \int \frac{1}{\sqrt{2\pi |\Sigma|}} \exp \left(-\frac{(c/z)^T \Sigma^{-1} (c/z)}{2}\right) P_z(z) \, dz.
\] (13)

In other words, the structure of the CG distribution results from the summation of a continuum of different Gaussians with different scales (variances)—each of which is weighted by the profile \( P_z(z) \). Different choices of the weighting profile \( P_z(z) \) result in synthesis of different kinds of pdfs (each with different kinds of heavy-tailed behavior)—including many of the well-known distributions in statistics including the Laplacian, generalized Gaussian, alpha-stable distributions etc [21].

A statistical model with prior distribution \( \pi(\theta) \) is said to be hierarchical Bayes [24, 25] if it can be decomposed in terms of conditional distributions \( \pi(\theta|\theta_i), \pi(\theta|\theta_2), \ldots, \pi(\theta_{n-1}|\theta_1) \), and marginal \( \pi(\theta_i) \):
\[
\pi(\theta) = \int \pi(\theta|\theta_1) \pi(\theta|\theta_2) \ldots \pi(\theta_{n-1}|\theta_1) \pi(\theta_1) \, d\theta_1 \ldots d\theta_n.
\] (14)

The CG distribution inherently has a hierarchical structure in that conditioning on the latent variable \( z \), Gaussianizes the wavelet field \( c \).

A fully Bayesian non-parametric approach to statistical inference under a hierarchical Bayesian prior model (14) involves drawing samples from the posterior distribution:
\[
P(c, z|y) \propto P(c, z) P(y|c) = P(c|z) P(z) P(y|c)
\] (15)
via MCMC and other sampling methods [27, 44]. The intermediate estimation of the latent variable, in our case the \( z \)-field, is called Type-II estimation [24, 25]; whereas the estimation of the primary parameter of interest (in our case \( c \)) is called Type-I estimation [24, 25]. The resulting posterior (15) can give us rich information not just about optimal point estimates that solve equation (1), but also the Bayesian intervals (i.e. Bayesian counterpart of frequentist ‘confidence intervals’) associated with them [27]. A fully non-parametric approach is
however computationally very intensive given the high-dimensionality of the signals that we are typically interested in.

To overcome this obstacle we explicitly focus on a particular family of prior distributions i.e. the graphical CG model described above. Our approach is nevertheless Bayesian not only due to the interpretation of the probability assignments of the parameters, but because the corresponding covariance estimates are also calculated as described in section 3.

In this paper, however, we do not use the resulting covariance matrix to further refine the effective sensing matrix and estimates and leave these issues for future work. Instead we focus on producing point estimates via a \emph{hierarchical Bayes-MAP strategy} wherein we perform inference by a sequence of MAP estimates starting with the latent variable:

1. **Step 1:** Perform Type-II MAP estimation problem:
   
   \[ z^* = \arg\max_z \log P(z|y). \]

2. **Step 2:** Perform Type-I MAP estimation problem conditioned on \( z^* \):
   
   \[ c^* = \arg\max_c \log P(c|y, z^*). \]

In the next section we describe our resulting HB-MAP (hierarchical Bayes-MAP) algorithm in detail. The resulting hierarchical Bayesian strategy yields an elegant optimization framework that reveals new structural information beyond sparsity and that is amenable to theoretical analysis. The focus of this paper is exclusively on the details of our algorithmic framework and its empirical performance on real and simulated data.

### 3. HB-MAP algorithm

As mentioned above, our goal is to solve the optimization problem \((6)\) wherein a graphical CG prior is placed on the unknown coefficients \(c\). Following the strategy described in section 2.2, we first estimate the non-Gaussian component \(z\) (referred to as Type-II estimation), followed by the estimation of \(u\) (referred to as Type-I estimation).

#### 3.1. Type-II estimation

Given that \(z = h(x)\), it suffices to estimate the multi-scale Gaussian tree random process \(x\). This can be performed by recourse, once again, to the Bayesian-MAP strategy:

\[
\begin{align*}
    x^* &= \arg\max_x \log P(x|y), \\
    &= \arg\max_x \log P(y|x) + \log P(x).
\end{align*}
\]

From (1) and (8) we have that:

\[y = \tilde{\Psi}(h(x) \odot u) + \nu,\]

where \(\tilde{\Psi} \equiv \Psi \Phi \)

\[\Rightarrow y = A_x u + \nu,\]

where

\[
\begin{align*}
    A_x &= \tilde{\Psi} H_x, \\
    H_x &= \text{diag}(h(x)).
\end{align*}
\]
From (18) and section 2.1, we have that:

\[ y \mid x \sim N(y; 0, A_x P_x A_x^T + \Sigma_y), \]
\[ x \sim N(x; 0, P_x), \]

where \( N(w; \mu, \Sigma) \) denotes a Gaussian with mean \( \mu \) and covariance matrix \( \Sigma \).

Thus we have that (18) is equivalent to:

\[ x^* = \arg\max_x f(x), \]  
(19)

where:

\[ f(x) = y^T (A_x P_x A_x^T + \Sigma_y)^{-1} y + \log \det (A_x P_x A_x^T + \Sigma_x) + x^T P_x^{-1} x. \]  
(20)

To solve (19) we employ a steepest descent approach as summarized in algorithm 1. There are two principle types of steepest descent strategies that we consider (and which are the basis of other hybrid approaches such as quasi-Newton [28] etc): gradient and Newton based steepest descent. The former involves a first-order Taylor series expansion of the cost function (20) wherein only the gradient vector of \( f \) has to be computed in each iteration. The Newton descent approach, which involves the second order Taylor approximation of \( f \), in addition entails the calculation of the Hessian matrix. The derivation of the gradient and Hessian equations is given in appendix.

For Newton iterations, once the Hessian is calculated via equation (A7), we find its closest positive semi-definite (psd) approximation and use this for calculating the descent direction as described in algorithm 1. Given a matrix \( X \), an approximation to its closest psd can be efficiently calculated as follows:

\[ M = Q L Q^T, \]  
(21)

where

\[ L \equiv \text{diag}(\max(\text{real}(\Lambda_X), 0)), \]
\[ Q \equiv \sqrt{Q_X^T Q_X}, \]
\[ \Lambda_X \equiv \text{diagonal matrix of eigenvalues of } X, \]
\[ Q_x \equiv \text{matrix of eigenvectors of } X. \]

**Algorithm 3.** HB-MAP algorithm for solving equation (1)

1. Initialize parameters
2. Perform Type-II estimation to obtain \( z^* \), the MAP estimate of the \( z \) field
3. Given \( z^* \) perform the approximate EM algorithm to estimate the optimum parameter \( \mu \)
4. Iterate between steps (2)–(3) until convergence
5. Perform Type-I estimation to obtain \( \nu^* \)
6. Return the optimum estimate of the image: \( I^* = \Phi c^* \); where \( c^* = z^* \odot \nu^* \)

The high-level pseudo-code of the HB-MAP algorithm. The details of the approximate EM algorithm is given in section 3.3. The details of parameter initialization is given in section 5.
Algorithm 4. Gradient calculation

1. for \( i = 1 \) to \( n \)
2. \( r = \tilde{G}_x(i, :) \)
3. \( \text{index} = \text{find locations } k \text{ where } r(k) \neq 0 \)
4. \( g(i) = 0 \)
5. for \( j = 1 \) to \( \text{index} \)
6. \( k = \text{index}(j), k_{\text{mod}} = \text{mod}(k, n), k_s = (k - k_{\text{mod}})/n \)
7. \( v_1 = \tilde{\Psi}(k, :) \quad v_2 = \tilde{\Psi}(k_{\text{mod}}, :) \)
8. \( g(i) = g(i) + v_1^T \text{vec}(v_2 v_1^T) \)
9. end for
10. end for
11. \( g = g + 2P_x^{-1}x \)
12. Return \( g \)

The pseudo-code for calculating the gradient vector in (35). The quantities \( G_x \) and \( v_x \) are defined in (36) and (37) respectively. Further details are in section 4.

Performing the closest psd approximation (21) of the Hessian matrix \( X = \nabla^2 f(x^*) \), which effectively constrains the search direction to an elliptical ball defined by the matrix \( M \), is extremely important for furnishing descent directions that can improve upon the gradient direction.

3.2. Type-I estimation

Let \( x^* \) be the solution of (19) obtained via algorithm 1. Thus the estimate of the non-Gaussian component of the unknown coefficients \( c \) is \( z^* = h(x^*) \). The goal of the Type-I estimation is how to estimate the corresponding \( u \) field. As explained in section 2.2, we once again resort to a Bayesian-MAP strategy for generating the optimality criterion to solve:

\[
\begin{align*}
    u &= \arg\max_u \log P(u|y), \\
    &= \arg\max_u \log P(y|u) + \log P(u), \\
    &= \arg\min_u (y - A_s^* u)^T \Sigma_u^{-1}(y - A_s^* u) + u^T P_u^{-1} u.
\end{align*}
\]  

We can easily verify that (24) reduces to solving the following equation w.r.t. \( u \):

\[
\Lambda_L \left( \tilde{\Psi}^T \Sigma_v^{-1} \tilde{\Psi} + \Lambda_u \right) \Lambda_R u = \Lambda_L \tilde{\Psi}^T \Sigma_v^{-1} y,
\]

where

\[
\begin{align*}
\Lambda_L &= \Lambda_R = \text{diag}(\epsilon^*), \\
\Lambda_u &= \Lambda_L^{-1} P_u^{-1} \Lambda_R^{-1}.
\end{align*}
\]
At this juncture, we explicitly exploit the sparse structure of \( \mathbf{z}^* \) by effectively pruning out the number of equations to be solved:
\[
\hat{\Lambda}_L \left( \tilde{\Psi}^T \Sigma_{\mathbf{w}}^{-1} \tilde{\Psi} + \Lambda_{\mathbf{y}} \right) \Lambda_{\mathbf{R}} \mathbf{u} = \hat{\Lambda}_L \tilde{\Psi}^T \Sigma_{\mathbf{w}}^{-1} \mathbf{y},
\]
where
\[
\hat{\Lambda}_L = \text{diag} (\mathcal{S}_{\mathbf{z}} [\mathbf{z}^*]),
\]
\[
\mathcal{S}_{\mathbf{z}} [z(i)] = \begin{cases} 1 & z(i) > \lambda, \\ 0 & z(i) \leq \lambda. \end{cases}
\]
\( \mathcal{S}_{\mathbf{z}} \) is a variable selection operator w.r.t. threshold \( \lambda \). We determine threshold \( \lambda \) from the histogram of \( \mathbf{z}^* \) as detailed in section 5. The resulting Type-I estimation algorithm is summarized in algorithm 2. We remark that the matrix \( \hat{\Lambda}_L \) prunes out the rows corresponding to values of the \( \mathbf{z} \) that are too small (as determined by threshold \( \lambda \)).

The operations of Type-I estimation in (26) together with the (quality of our) Type-II estimation are both crucial for enabling state-of-the-art performance of the resulting HB-MAP algorithm. While \( \hat{\Lambda}_L \) destroys the \( \mathbf{z} \) field information, \( \Lambda_{\mathbf{R}} \) preserves it. As can be easily verified, the preservation of \( \mathbf{z} \) field in \( \Lambda_{\mathbf{R}} \) information does confer some performance benefits over and above the sparsity structure in the \( \mathbf{z} \)-field.

3.3. The HB-MAP algorithm

Algorithm 3 summarizes the high-level pseudo-code of our HB-MAP algorithm. In addition to employing the Type-I and Type-II estimation procedures described above, the HB-MAP algorithm also employs an approximate EM algorithm that enables the refinement of parameter \( \mu \). We now derive the EM algorithm which generalizes the one proposed in [21].

The parameter that we aim to optimize is \( \mu \) that controls the non-Gaussian inter-scale dependencies between the wavelet coefficients. Let \( \mu_k \) be the value of \( \mu \) in the \( k \)th iteration; and let \( x_k^* \) be the corresponding Type-II MAP estimate that solves (16). We then can perform a second-order Taylor expansion of the cost function \( f \) about \( x_k^* \) as follows:
\[
f(x) \approx f(x_k^*; \mu_k) + [\nabla f(x_k^*; \mu_k)]^T (x - x_k^*) + 0.5(x - x_k^*)^T [\nabla^2 f(x_k^*; \mu_k)]^{-1} (x - x_k^*)
\]
\[
\approx 0.5(x - x_k^*)^T [\nabla^2 f(x_k^*; \mu_k)]^{-1} (x - x_k^*)
\]
where \( \nabla f(x_k^*; \mu_k) = 0 \) by definition of the fact that \( x_k^* \) is a local optimum point; and \( f(x_k^*; \mu_k) \approx 0 \) by assumption. Thus we have that:
\[
p(x_k^*; \mu_k) \approx \mathcal{N}(x; x_k^*, \nabla^2 f(x_k^*; \mu_k)).
\]

Given this a natural definition of the \( Q \)-function of the EM algorithm [29] is:
\[
Q(\mu; \mu_k) = -\mathbb{E}_{p(x|y; \mu_k)} \{ \log p(x|y, \mu) \}.
\]

It is straightforward to verify that:
\[
Q(\mu; \mu_k) = \frac{n}{2} \log (2\pi) + \frac{1}{2} \log \det (P_x(\mu)) + \frac{1}{2} (x_k^*)^T P_x^{-1}(\mu) x_k^*
\]
\[
+ \frac{1}{2} \text{tr} (P_x^{-1}(\mu) \nabla^2 f(x_k^*; \mu_k))
\]
\[
= \frac{n}{2} \log (2\pi) + \frac{1}{2} \log \det (P_x(\mu)) + \frac{1}{2} (x_k^*)^T P_x^{-1}(\mu) x_k^* + \frac{1}{2} \text{tr} (P_x^{-1}(\mu) \nabla^2 f(x_k^*; \mu_k)).
\]
and that the gradient function of $Q$ is as follows:

$$
\nabla_{\mu}Q(\mu; \mu_k) = -\frac{1}{2}(\nabla_{\mu}P_t(\mu))\text{vec}(P_t^{-T}(\mu)) + \frac{1}{2}(\nabla_{\mu}P_t(\mu))\text{vec}(P_t^{-T}(\mu))
\times x_k^T(x_k^T P_t^{-T}(\mu)) + \frac{1}{2}(\nabla_{\mu}P_t(\mu))\text{vec}(P_t^{-T}(\mu))\nabla^2 f(x_k^T; \mu_k)
\times P_t^{-T}(\mu)).
$$

(33)

In order to find the optimum parameter $\mu$, we solve the following optimization problem:

$$
\mu^* = \arg\min_{\mu} Q(\mu; \mu_k),
$$

(34)

where (34) can be solved by a simple gradient descent by using the gradient expression given in (33).

4. Some computational details

The computationally most intensive part of our algorithm is the calculation of the Hessian matrix and (to a lesser extent) the gradient vector that is required in the steepest descent algorithm given in algorithm 1. While the equations for the calculation of these quantities is given in the appendix, direct implementation of these equations without exploiting the sparse structure of the matrices involved will be infeasible (using any conceivable digital computer either at present or anytime in the future) due to the presence of massive tensor products resulting from Kronecker operations on high dimensional matrices. In this section we give some indications as to how this apparent computational bottleneck can be systematically overcome.

From (A2) in the appendix, the gradient vector is given by the following equation:

$$
\nabla f(x) = 2P_t^{-1}x + G_x(\Psi^T \otimes \Psi^T)v_x,
$$

(35)

where

$$
\tilde{G}_x = \nabla_t H_x \{ \{P_tH_t \otimes I_n \} + \{I_n \otimes (P_tH_t) \} \},
$$

(36)

$$
v_x = \text{vec}((M_x^{-1})^{-1} - [(M_x^{-1})y] \otimes (M_x^{-1})y].
$$

(37)

The quantities $M_x$ and $H_x$ are defined in appendix. Algorithm 4 describes an efficient procedure for calculating the gradient in (35).

**Algorithm 5.** Sparsity preserving operations

1. $[A_i \in \mathbb{R}^{n_i \times n_{i+1}}]_{i=1}^{k}$ are matrices to be multiplied; $n_1 = n$
2. for $i = 1 : n$
3. $r = A_i(i, :)$
4. for $j = 1 : k$
5. $r = \text{prod}(r, A_j)$
6. end
7. $D(i, :) = \text{devec}(r)^{\text{vec}(N)}$
8. end
9. Return $D$
High-level pseudo-code for sparsity preserving operations that are employed, in part, in the calculation of the Hessian. The matrix $N$ is either $N_1$ or $N_2$ (in (35–36)). The operator $\text{prod}()$ refers to specialized matrix products that use algorithmic techniques similar to that in algorithm 4. Finally, $\text{devec}(r) \in \mathbb{R}^{m^2}$. Further details of the Hessian calculation are given in section 4.

The Hessian form, given equation (A7) in the appendix, can be expressed as follows:

$$
\nabla^2 f(x) = 2P^{-1}x + \sum_{i=1}^{4} M_i [N_1 + N_2] + Q_1 + Q_2,
$$

(38)

where

$$
M_1 = \nabla^2 H_x [(P_H H_x) \otimes I_n] (\tilde{\Psi}^T \otimes \tilde{\Psi}^T \otimes I_n),
$$

(39)

$$
M_2 = \nabla^2 H_x [I_n \otimes (P_H H_x) \otimes I_n] (\tilde{\Psi}^T \otimes \tilde{\Psi}^T \otimes I_n),
$$

(40)

$$
M_3 = \nabla H_x [I_n \otimes (P_H \otimes \tilde{K}_m)] [I_{n^2} \otimes (\nabla H_x)^T] (\tilde{\Psi}^T \otimes \tilde{\Psi}^T \otimes I_n),
$$

(41)

$$
M_4 = \nabla H_x (I_n \otimes P_{H}^T) (\tilde{K}_m \otimes I_n) [I_{n^2} \otimes (\nabla H_x)^T] (\tilde{\Psi}^T \otimes \tilde{\Psi}^T \otimes I_n).
$$

(42)

Furthermore

$$
N_1 = \text{vec}(M^{-1}_1) \otimes I_n,
$$

(43)

$$
N_2 = (M^{-1}_1 y) \otimes (M^{-1}_1 y) \otimes I_n,
$$

(44)

$$
Q_1 = \frac{1}{2} \tilde{G}_x \left[ (\tilde{\Psi}^T (M^{-1}_1 y)(M^{-1}_1 y)^T \tilde{\Psi}) \right] G_x^T,
$$

(45)

$$
Q_2 = \tilde{G}_x \left[ K_m \left( (\tilde{\Psi}^T (M^{-1}_1 y)(M^{-1}_1 y)^T \tilde{\Psi}) \right) \right] G_x^T.
$$

(46)

Matrices $K_m$ and $\tilde{K}_m$ are defined in the appendix.

As mentioned above, due to the high dimensionality of the matrices involved extreme caution must be exercised when carrying out the Kronecker products in the above equations. The key property that we exploit to enable efficient calculation of the above matrix products is to preserve sparsity of the intermediate matrices products calculated from left to right.

Firstly by construction the matrices $\nabla H_x$ and $\nabla^2 H_x$, given in (A5) and (A12) respectively, are sparse. Given this algorithm 5 allows the sparsity of the intermediate matrix products to be preserved while producing the desired matrix products (for e.g. $M_i N_j \forall i, j$) in the end. Apart from this, algorithmic techniques similar to that in algorithm 4 are exploited to calculate many of the intermediate and other matrix products that are involved in the Hessian calculation.

In terms of flop count the gradient calculation in algorithm 4 takes $O(n S)$ and the Hessian calculation takes $O(n^2 S)$ (where integer $S$ depends on matrix $P_H$; for a block-diagonal matrix $P_H$, $S$ is bounded by a small constant) on a single-core central processing unit (CPU). We point out however that a tremendous amount of parallelism exists in the structure of the gradient and Hessian computations. Each of the iterations in algorithms 4 and 5 are independent of each other and thus can be performed in any order. This key property allows for parallel processing to be efficiently exploited in multicore CPU or graphical processing unit platforms.

In spite of the linear (and quadratic) complexity described above for the case of the gradient- (and Newton-) HB-MAP algorithm, it turns out that the presence of big hidden
constants under the big-O notation render the current implementation of the HB-MAP very slow in practice. This will be explained in greater detail towards the end of section 5 below.

5. Experiments and results

5.1. HB-MAP performance w.r.t. the oracle

We first demonstrate the performance of the HB-MAP and competing algorithms w.r.t. the theoretically best performance that is achievable by any algorithm. The latter is furnished by means of the oracle HB-MAP strategy. The oracle HB-MAP corresponds to the case where we know the z-field exactly; because of this, it only applies the Type-I estimation in algorithm 2 in order to estimate the image. The performance of the oracle HB-MAP is therefore a lower bound which no other algorithm can surpass. To demonstrate this we perform Monte-Carlo trials on images sampled from our multiscale CG tree model. In this situation the z-field is, indeed, by construction, known exactly and is thus exploited by the oracle HB-MAP algorithm.

Importantly, these Monte-Carlo simulations give us the ability to compare the performance of the HB-MAP and other competing algorithms in various controlled situations—such as varying levels of sparsity in the data etc—which otherwise are not directly observable when applying the algorithms on empirical data.

In this section we compare this theoretically best performance to the Newton-HBMAP, the gradient HB-MAP, 11-ls [8], CoSaMP [11], ROMP [10], OMP [9], BCS [41], and FBP [7] algorithms. In all the simulations we choose \( \Phi \) to be a dictionary of Biorthogonal 1.1 wavelets [30]. In all the simulation results shown below, the parameters of each of the algorithms have been adjusted to give the best performance that is possible. For each Monte-Carlo trial, we calculate the mean-squared error (MSE) between the reconstructed and original wavelet coefficients. The wavelet coefficient vectors are all scaled to unity norm when calculating the MSE (for the case of FBP, the effective wavelet coefficients are determined easily from the reconstructed image).

We first consider the performance of the various algorithms for different levels of sparsity (for a fixed number of channels) from low to high-sparsity cases. We recall in section 2.1 that the amount of sparsity in our multiscale CG model is inversely related to the parameter \( \alpha \). For the Monte-Carlo trials we generate 16 \( \times \) 16 images on which the Radon transform at 8 equally spaced angles is performed (which simulates in effect the signals received by eight equally spaced sensors after perfect matched filtering). In all cases we inject 60 dB measurement noise into the received signal.

Given the received signal, the various algorithms above are applied to reconstruct the image which is then compared with the original image. Figures 3–6 show the performance of the various algorithms for the cases \( \alpha = 0.2, 0.5, 0.75, \) and 1 respectively. As expected the oracle HB-MAP gives a lower bound of the performance of any algorithm that solved equation (1) for this scenario. In all cases we find that the Newton and the gradient HB-MAP consistently outperform all algorithms. For the case of high-sparsity (which is more typical of natural images), the Newton HB-MAP outperforms the gradient HB-MAP algorithm. For the low-sparsity case on the other hand (which is not typical of natural images), the performance of Newton and gradient HB-MAP algorithms is virtually the same. While BCS, ROMP and CoSaMP performing relatively well compared to the FBP approach for the high-sparsity, their performance dramatically drops, unlike 11-ls, in the low-sparsity case. The HB-MAP algorithms on the other hand consistently outperform the various CS algorithms in both cases.
Figures 7–9 show the performance of the various algorithms for a different number of sensing channels (for a fixed level of sparsity). While Figures 3–6 already show the performance of HB-MAP for 8 channels in detail, figures 7–9 show the performance of the various algorithms for cases where the number of channels is 6, 10, and 15 respectively. Once again we see that in all cases, the Newton HB-MAP and the gradient HB-MAP outperform all other algorithms.

In all the above we have demonstrated the performance of the HB-MAP for the case of inverting the Radon transform. In figure 10 we also demonstrate below the performance of the HB-MAP for the cases of random sensing matrices which is ubiquitously used in CS literature. Here the number of rows of the sensing matrix has been taken to be equivalent to the case of eight sensing channels. From figure 10 we can see that HB-MAP outperforms the various state-of-art algorithms even in this situation. Random sensing matrices however are not at all realistic for imaging applications which is why we focus on the more difficult problem of inverting the Radon transform under the constraint of sparse sensing in this paper.

It is important to note that in all the cases above (i.e. figures 3–10), the HB-MAP outperforms all other algorithms in every single Monte-Carlo trial across a variety of different experimental conditions. These results give an indication of the power of the HB-MAP paradigm in handling a variety of inverse problem applications in imaging. This will be further substantiated in the performance on empirical data below.
Now we are ready to examine performance of HB-MAP for empirical data consisting of natural images. Here again we compare the performance of the Newton HB-MAP and the gradient HB-MAP against $l_1$-$l_s$ \cite{8}, CoSaMP \cite{11}, ROMP \cite{10}, BCS \cite{41}, and FBP \cite{7} algorithms. As before in all the experiments we choose $\Phi$ to be a dictionary of Biorthogonal 1.1 wavelets \cite{30}. We consider progressively more challenging experimental conditions below from Figure 11 to 14.4

In Figure 11 we show the performance of the Newton HB-MAP algorithm, the gradient HB-MAP, the FBP, and the various CS algorithms when applied to inverting the Radon transform, taken at 15 uniformly spaced angles of a $32 \times 32$ segment of the Barbara image—Figure 4. Performance for the case where $16 \times 16$ images are reconstructed based on measurements from a Radon transform at eight different uniformly-spaced angles. The simulated images have sparsity level $\alpha = 0.5$. (i) Our algorithm (Newton, gradient descent, and oracle). (ii) Sparsity based convex opt. algorithms ($l_1$-$l_s$, CoSaMP, ROMP and BCS). (iii) Standard Fourier-based reconstruction. Our HB-MAP algorithm consistently shows the best performance in terms of MSE across all trials. Newton HB-MAP outperforms gradient HB-MAP. The oracle HB-MAP plot is the result of the Type-I estimation for the case when we know the true $z$-field and thus gives the best performance as expected.

For all the empirical experiments in this paper, we chose a particular 64x64 (and 32x32) sub-image of the Barbara image because, from an image processing point of view, this sub-image contains a nice combination of many features—such as different textures, edges at different scales, smooth regions, contrasts and spatial transitions between gray level shadings (or textures) etc—that likely renders this a good candidate for testing the mettle of the imaging algorithms considered here.

For all the empirical experiments below we used the following parameters for HB-MAP and the other algorithms (the parameters for the competing algorithms were chosen to try to maximize their respective performance). For the Newton- and gradient-HB-MAP we set $\alpha = 0.5$, $\mu = 0.99$, and the matrix $P$ to be the identity matrix. Finally, the threshold $\lambda$ in (28) is chosen such that the fractional density, below $\lambda$, of the histogram of the estimated $z$ field is 0.1 (we have empirically found most threshold values below 0.1 give equivalent performance). For the $l_1$-$s$ algorithm the sparsity regularization parameter is set to 0.05 and the relative tolerance is set to 0.2. The sparsity level for the greedy algorithms CoSaMP, ROMP and OMP were set to 60, 50 and 50 respectively. For the BCS algorithm we found the default values (sparsity threshold of $10^{-6}$) give best performance. We intend to post the full Matlab implementation of the HB-MAP algorithm on-line (in a website associated with the author) in the near future.
and where 60 dB of measurement noise is added at each observation channel (sensor). The quality of the reconstruction has been measured using the structural similarity index (SSIM) quality measure\(^5\) \cite{31}. We can see that the Newton- and gradient-HB-MAP algorithms are significantly better than all of the CS algorithms and especially the traditional FBP approach to imaging.

Figure 12 shows the performance of the Newton- and gradient-HB-MAP algorithms when applied to inverting the Radon transform, taken at 15 uniformly spaced angles, of a 64 \times 64 segment of the Barbara image—and, again, in which 60 dB of measurement noise is added at each observation channel (sensor). When compared to other algorithms we see that both the Newton- and gradient-HB-MAP outperforms every other CS algorithm and is also once again much better than the standard FBP algorithm.

Similarly, figures 13 and 14 show the performance of the HB-MAP for the case where the number of channels is 6 and 10 respectively. Once again we see that in all cases, the HB-MAP algorithm outperforms all other algorithm in terms of visual and quantitative performance measures.

Furthermore we remark that the full power of the HB-MAP has not yet been exploited in this paper because we have set \(\gamma = 0\) (i.e. we have not exploited the self-similarity of variance of wavelet coefficients across scales which is present in natural images). Incorporation of this important parameter will also be left for future work.

\(^5\) SSIM [31] is an image quality measure that monotonically maps visual image quality to a scale ranging from zero to unity; where unity corresponds to best perceptual match with the reference image.
Figure 6. Performance for the case where $16 \times 16$ images are reconstructed based on measurements from a Radon transform at eight different uniformly-spaced angles. The simulated images have sparsity level $\alpha = 1$ (low level of sparsity). (i) Our algorithm (Newton, gradient descent, and oracle). (ii) Sparsity based convex opt. algorithms (l1-ls, CoSaMP, ROMP and BCS). (iii) Standard Fourier-based reconstruction. Our HB-MAP algorithm consistently shows the best performance in terms of MSE across all trials. Newton HB-MAP outperforms gradient HB-MAP. The oracle HB-MAP plot is the result of the Type-I estimation for the case when we know the true $z$-field and thus gives the best performance as expected.

Figure 7. Performance for the case where $16 \times 16$ images are reconstructed based on measurements from a Radon transform at six different uniformly-spaced angles. The simulated images have sparsity level $\alpha = 0.2$. Higher levels of sparsity are more typical of natural images. (i) Our algorithm (Newton, gradient descent, and oracle). (ii) Sparsity based convex opt. algorithms (l1-ls, CoSaMP, ROMP and BCS). (iii) Standard Fourier-based reconstruction. Our HB-MAP algorithm consistently shows the best performance in terms of MSE across all trials. Newton HB-MAP outperforms gradient HB-MAP. The oracle HB-MAP plot is the result of the Type-I estimation for the case when we know the true $z$-field and thus gives the best performance as expected.
5.3. Current limitations of the HB-MAP algorithm

The experimental results above clearly demonstrate the advantages that our hierarchical Bayesian approach to inverse problems has over conventional counterparts. There is, however, a significant limitation of HB-MAP compared to these algorithms viz. computational complexity. As described towards the end of section 4, the presence of a big constant under the big-O notation renders HB-MAP very slow in practice. To execute the Newton-HB-MAP algorithm on a $64 \times 64$ image takes on the order of 3 days (when using parallel processing on a state-of-the-art desktop computer). Although the gradient HB-MAP brings this down to around a couple of hours while maintaining reconstructed image quality in a manner that is very much comparable to Newton HB-MAP as shown above (and while outperforming all other algorithms in the literature), it is still much slower than typical CS algorithms which take on the order of seconds to a couple of minutes, and are much slower than the conventional FBP algorithm which calculates the reconstructed image almost instantaneously for these problem sizes.

It turns out that the reason for slow computational speed of the HB-MAP algorithm is the storage and transfer of large matrices (manipulated by HB-MAP) in between the hard-disk and RAM. These transfer times can be very significant and dominate the computational cost per iterations.

One way to overcome this issue would be to devise customized matrix algorithms that handle sparse matrices in a way that is better than off-the-shelf approaches available in numerical toolboxes. Another approach is to perform wavelet and other transforms involved via fast algorithms rather than via matrix multiplication as we are doing in our current
implementation. Further ways of speeding up the algorithm includes the incorporation of alternative optimization approaches such as stochastic gradient descent \cite{43} and stochastic simulation approaches \cite{44}.

Importantly, the demonstration of the HB-MAP algorithm and its superior performance compared to conventional counterparts, as revealed in this paper, can open up significant research challenges in the signal processing and inverse problems communities for devising various iterative convex relaxations of the HB-MAP algorithm that can give a similar performance with bounded performance guarantees, while significantly reducing the computational complexity per iteration.

In the next section we describe further avenues of future work emanating from this paper.

### 6. Discussion

We have offered novel algorithmic techniques, embodied in HB-MAP, for solving, at this level of generality, equation (1) under a global multiscale tree CG prior. Since the global CG model subsumes a very broad class of priors in the literature—including the naïve Bayes Laplacian model which is the basis of most CS and sparsity based reconstruction algorithms \cite{8–18, 39–41}—our HB-MAP algorithm offers a vehicle to extend the capabilities of current CS algorithms to include truly global priors.

We have previously shown that probabilistic graphical models yield state-of-the-art performance for signal classification problems \cite{33, 34}. In this paper we deploy a CG-based probabilistic graphical model for the purposes of regression (i.e. inverse problems) and we have shown how state-of-the-art results can likewise be obtained. An immediate thrust of our
future work, as discussed in section 5.3 above, is to investigate faster algorithms for carrying out the calculations of HB-MAP which can help to significantly broaden its range of applications. In our paper we have focused on the basic problem of inverting the Radon transform—which underlies most imaging applications [1–7]. Our formulation and the resulting HB-MAP algorithm is however very general and can be applied to any sampling matrix $\Psi$ whatsoever. Important future work arising from this includes the application of our algorithm to different application scenarios arising in radar, remote sensing, acoustics and medical domains in addition to theoretical investigations into the properties of the HB-MAP algorithm.

Although we have limited ourselves in this paper, to any class of wavelet dictionaries, our techniques are capable of being extended to a more general class of overcomplete dictionaries. An important degree of freedom which we have not exploited in this paper is the dimensionality of the nodes of the graphical model. In the current instantiation we stored the LH, HL, HH coefficients, associated with the wavelet transform, at each node. For a more general overcomplete dictionary, this can be arbitrarily changed and can also be spatially varying to account for non-stationary behavior across the image. We recall once again that when $\mu = 0$ the graphical model reduces to the naive Bayes case i.e. standard CS algorithms operating on overcomplete dictionaries. Significant algorithmic challenges and problems emerge from this which when overcome can potentially yield powerful ways to accommodate Bayesian (CG) and structural (dictionary) prior models into the inference process.

Figure 10. Performance for the case where $16 \times 16$ images are reconstructed based on measurements from a Random sensing matrix (with number of rows equivalent to a channel consisting of eight different uniformly-spaced angles). The simulated images have sparsity level $\alpha = 0.2$. Higher levels of sparsity are more typical of natural images. (i) Our algorithm (Newton, gradient descent, and oracle). (ii) Sparsity based convex opt. algorithms (I1-Is, CoSaMP, ROMP and BCS). (iii) Standard Fourier-based reconstruction. Our HB-MAP algorithm consistently shows the best performance in terms of MSE across all trials. Newton HB-MAP outperforms gradient HB-MAP. The oracle HB-MAP plot is the result of the Type-I estimation for the case when we know the true $z$-field and thus gives the best performance as expected.
Another important direction of generalization is the structure of the graph itself. Tree-based representations of global probability distributions over images are known to be limited in expressiveness \cite{35} to overcome which loops must be introduced into the graphs. However, several computational difficulties emerge with performing inference over the resulting loopy graphs \cite{35}. Over the past decade a lot of important progress has been made in increasing our understanding of this situation \cite{35–38}. An important research direction is therefore to extend the capabilities of the HB-MAP algorithm to incorporate arbitrary graphical structures.

Figure 11. Here performance of our Newton- and gradient-HB-MAP and other algorithms is shown for the case where the original image is a $32 \times 32$ image segment for which a Radon transform is performed at 15 uniformly spaced angles plus 60 dB of measurement noise at each channel. The HB-MAP algorithm has the best visual performance and in terms of SSIM \cite{31}.

Another important direction of generalization is the structure of the graph itself. Tree-based representations of global probability distributions over images are known to be limited in expressiveness \cite{35} to overcome which loops must be introduced into the graphs. However, several computational difficulties emerge with performing inference over the resulting loopy graphs \cite{35}. Over the past decade a lot of important progress has been made in increasing our understanding of this situation \cite{35–38}. An important research direction is therefore to extend the capabilities of the HB-MAP algorithm to incorporate arbitrary graphical structures.
Another natural direction of future work is to investigate how the Bayesian/confidence intervals resulting from our covariance structure estimates can be utilized to further refine the effective sensing matrix. Although a fully non-parametric Bayesian extension of our work can also be pursued, the advantage of our approach is that it results in an elegant optimization framework that reveals novel structural cues (specifically $\mu$ and $\gamma$) which point us to a direction beyond sparsity (which in our case is controlled by $\alpha$).

Figure 12. Here performance of our Newton- and gradient-HB-MAP and other algorithms is shown for the case where the original image is a $64 \times 64$ image segment for which a Radon transform is performed at 15 uniformly spaced angles plus 60 dB of measurement noise at each channel. The HB-MAP algorithm has the best visual performance and in terms of SSIM [31].

Another natural direction of future work is to investigate how the Bayesian/confidence intervals resulting from our covariance structure estimates can be utilized to further refine the effective sensing matrix. Although a fully non-parametric Bayesian extension of our work can also be pursued, the advantage of our approach is that it results in an elegant optimization framework that reveals novel structural cues (specifically $\mu$ and $\gamma$) which point us to a direction beyond sparsity (which in our case is controlled by $\alpha$).
Acknowledgments

This work (RGR) was supported by the Office of Naval Research through the US Naval Research Laboratory Base Program.

---

Figure 13. Here performance of our Newton- and gradient-HB-MAP and other algorithms is shown for the case where the original image is a 64 × 64 image segment for which a Radon transform is performed at ten uniformly spaced angles plus 60 dB of measurement noise at each channel. The HB-MAP algorithm has the best visual performance and in terms of SSIM [31].

Acknowledgments

This work (RGR) was supported by the Office of Naval Research through the US Naval Research Laboratory Base Program.
Appendix

In this section we derive the gradient and Hessian equations of the cost function $f$ in (20) that are used in the Type-II estimation procedure in algorithm 1. We employ many basic identities of the matrix calculus [32] to derive these expressions:

Figure 14. Here performance of our Newton- and gradient-HB-MAP and other algorithms is shown for the case where the original image is a 64 $\times$ 64 image segment for which a Radon transform is performed at six uniformly spaced angles plus 60 dB of measurement noise at each channel. The HB-MAP algorithm has the best visual performance and in terms of SSIM [31].
From (20) we have that the gradient of cost function $f$ is given by:

$$
\nabla f(x) = \nabla_x \{x^T P_1^{-1} x\} + \nabla_x \{y^T M_x^{-1} y\} + \nabla_x \{\logdet(M_x)\},
$$

(A1)

$$
= 2P_x^{-T} x - G_x [(M_x^{-1} y) \otimes (M_x^{-1} y)] + G_x \vec{\text{vec}}(M_x^{-T}),
$$

(A2)

where

$$
\otimes \equiv \text{Kronecker product of matrices}
$$

$$
X^{-T} \equiv (X^T)^{-1},
$$

$$
M_x = A_x P_x A_x^T + \Sigma_v,
$$

(A3)

$$
\Sigma_v \equiv \text{defined in section 3.1}
$$

$A_x$ is defined in (18)

$$
G_x \equiv \nabla_x M_x = \nabla_x H_x \left\{ \left( P_x H_x \otimes I_n \right) + \left\{ I_n \otimes (P_x H_x) \right\} \right\} (\Psi^T \otimes \hat{\Psi}^T)
$$

(A4)

such that

$$
\nabla_x H_x(i, j) = \begin{cases} \frac{\partial h(x_i)}{\partial x_i} & j = i(n + 1) - n \\ 0 & \text{else} \end{cases}
$$

(A5)

$P_x^{-T} = P_x^{-1}$ and $M_x^{-T} = M_x^{-1}$.

The Hessian can be derived in a similar manner:

$$
\nabla^2 f(x) = \nabla^2_x \{x^T P_x^{-1} x\} + \nabla^2_x \{\logdet(M_x)\} + \nabla^2_x \{y^T M_x^{-1} y\},
$$

(A6)

$$
= 2P_x^{-1} + L_x + R_x,
$$

(A7)

where

$$
L_x \equiv \nabla^2_x \{\logdet(M_x)\} = \nabla_x G_x (\text{vec}(M_x^{-1}) \otimes I_n) - \nabla_x \left[ M_x^{-1} \otimes M_x^{-1} \right] G_x^T,
$$

(A8)

$$
R_x \equiv \nabla^2_x \{y^T M_x^{-1} y\} = - (\nabla_x G_x) [(M_x^{-1} y) \otimes (M_x^{-1} y) \otimes I_n] + G_x \left[ ((M_x^{-1} y)^T \otimes M_x^{-1}) \times \left[ ((M_x^{-1} y)^T \otimes I_m + ((M_x^{-1} y)^T \otimes I_m) K_{mm} \right] \right] G_x^T,
$$

(A9)

$$
\nabla_x G_x = E_x \left( \Psi^T \otimes \hat{\Psi}^T \otimes I_n \right),
$$

(A10)

$$
E_x = \nabla^2 H_x \left\{ (P_x H_x) \otimes I_n + I_n \otimes (P_x H_x) \otimes I_n \right\} + \nabla H_x \left\{ I_n \otimes (P_x \hat{K}_{mm}) + (I_n \otimes (P_x \hat{K}_{mm}) \otimes \nabla H_x^T \right\}
$$

(A11)

$I_p \in \mathbb{R}^{p \times p}$ is a $p \times p$ identity matrix

$K_{mm} \in \mathbb{R}^{m \times m}$ permutation matrix such that:

$$
K_{mm} \text{vec}(A) = \text{vec}(A^T)
$$

$$
\tilde{K}_{mm} = [I_n \otimes e_{1,n}^T, \ldots, I_n \otimes e_{1,n}^T] \in \mathbb{R}^{m \times m}
$$
\( e_{k,n} \in \mathbb{R}^n \) is a Euclidean unit vector with unity at the \( k \)th location

\[
\nabla^2 H_k \in \mathbb{R}^{n \times n}
\]

such that

\[
\nabla^2 H_k(i, j) = \begin{cases} 
\frac{\partial^2 h(x_i)}{\partial^2 x_j} & j = i(n^2 + n + 1) - (n^2 + n), \\
0 & \text{else}.
\end{cases}
\tag{A12}
\]

References

[12] Elad M 2009 Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (Berlin: Springer)
[22] Raj R G and Bovik A C 2012 A nonlinear compound representation of sea clutter IEEE Radar Conf. (Atlanta, GA)
[33] Srinivas U, Monga V and Raj R G 2011 ATR using discriminative graphical models IEEE Int. Conf. on Image Processing (ICIP)