Sparse Signal Representation: Image Compression using Sparse Bayesian Learning

Galen Reeves, Vijay Ullal, Eric Battenberg

I. Introduction and Motivation

In this paper we investigate methods of finding sparse representations of a signal $\boldsymbol{t} = [t_1, \dots, t_N]^T$, i.e. representations with the fewest non-zero coefficients. We assume that $\boldsymbol{t}$ has a sparse representation in some possibly over-complete dictionary of basis functions Φ. We represent the signal as

$$\boldsymbol{t} = \Phi \boldsymbol{w} + \boldsymbol{\epsilon} \tag{1}$$

with $\Phi \in \mathbb{R}^{N \times M}$, $M \geq N$, and some noise $\boldsymbol{\epsilon}$. The challenge is to determine the sparsest representation of reconstruction coefficients $\boldsymbol{w} = [w_1, \dots, w_M]^T$.

Finding a sparse representation of a signal in an over-complete dictionary is equivalent to solving a regularized linear inverse. For a given dictionary Φ, finding the maximally sparse $\boldsymbol{w}$ is an NP-hard problem [1]. A great deal of recent research has focused on computationally feasible methods for determining highly sparse representations and is fueled by applications in signal processing, compression and feature extraction [2].

In section II of this paper we formulate the problem of finding a sparse inverse solution. In section III we give an overview of several popular techniques: *Method of Frames* (MOF), *Matching Pursuits* (MP), *Basis Pursuit* (BP), *Focal Underdetermined System Solution* (FOCUSS), and *Sparse Bayesian Learning* (SBL). We give a general comparison of problems solved by each method and the strengths and weaknesses of each approach. In sections IV and V we apply these techniques to two applications: image compression and medical image reconstruction. Each application highlights one of the two goals of sparse signal representation: sparsity and hyper-resolution.

A. Sparsity: Image Compression

We can define the sparsity of a signal $D(\boldsymbol{t}) \approx ||\boldsymbol{t}||_0$ as the number of significant coefficients, i.e. coefficients whose values will not be quantized to zero. The goal of compression is to find a representation of $\boldsymbol{w}$ such that $D(\boldsymbol{w}) < D(\boldsymbol{t})$. For images we consider 1D representations in $\mathbb{R}^N$ where N is the product of the image dimensions. Compression is paramount to the usability of image and video signals, but the computational complexity of the sparse basis selection methods has severely limited the size of the signals that can be compressed. In this paper we apply BP, MP, and SBL to images up to 32x32 ($N = 1024$) using over-complete dictionaries consisting of either the DCT or a steerable pyramid.

B. Hyper Resolution: MEG Reconstruction

If we view $\boldsymbol{t}$ as some sampled version of $\boldsymbol{w}$ then we can consider finding $\boldsymbol{w}$ as an interpolation, i.e. we are increasing the resolution of our signal. Since this is an under-constrained problem we must use *a priori* knowledge of $\boldsymbol{w}$ to choose from the infinite number of solutions. If we know that $\boldsymbol{w}$ is sparse we can try to reconstruct it with a high resolution. In this paper we present MOF, MP, and SBL applied to the inverse problem in magnetoencaphalography (MEG) where our signal is a 1D representation of 4D signal (3D location in space over time). At each point in time the electric currents at 9,981 locations on the surface of the brain are determined by 273 magnetic sensors located on the surface of the head.

II. Problem Formulation

If we assume some sparsity inducing objective function $D(\boldsymbol{w})$ (not necessarily the L_0 norm) then our constrained linear inverse takes on the following form

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} ||\boldsymbol{t} - \Phi \boldsymbol{w}||^2 + \lambda D(\boldsymbol{w}) \tag{2}$$

where λ is a tradeoff between sparsity and reconstruction error. In much of our analysis in this paper we will look at the zero noise case ($\lambda \to 0$). The problem then becomes

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} D(\boldsymbol{w}) \qquad \text{s.t. } \boldsymbol{t} = \Phi \boldsymbol{w}. \tag{3}$$

In the next few sections we discuss the choices we must make in $D(\cdot)$ and Φ and show how to represent the solution as a weighted pseudo-inverse.

A. Sparsity Measure

Ideally we want to choose the L_0 norm as a measure of our sparsity. That is

$$D(\boldsymbol{w}) = \sum_{i=l}^{M} \mathbf{1}(|w_i| \neq 0) \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function. Unfortunately, since finding the maximally sparse solution is NP-hard we will see that many techniques try instead to minimize the L_p norm defined as

$$D(\boldsymbol{w}) = \left(\sum_{i=l}^{M} |w_i|^p \right)^{1/p}. \tag{5}$$

The authors of [3] argue that a smaller value of p leads to more sparse solutions. We offer the following intuition. For $p < 1$ the know that the slope of $|w_i|^p$ is steepest for values of w_i whose magnitude are near zero. Accordingly the penalty is given more to the number of $w_i \not\approx 0$ rather than the magnitude of w_i. This results in a large number of coefficients with negligible values.

B. Dictionary Selection

Dictionary selection directly affects our ability to represent the signals. When implementing each of the algorithms discussed, it is sensible to choose an overcomplete dictionary $\Phi \in \mathbb{R}^{N\times M}$ that will give the sparsest set of reconstruction coefficients while maintaining an accurate representation of the original signal. Possible dictionaries include wavelet dictionaries, Gabor dictionaries, cosine packets, and chirplets. Another possible choice is that of the steerable pyramid, which is a multi-scale, multi-orientation set of basis functions that closely resembles wavelets.

According to [3], choosing a random dictionary can provide an unbiased means of comparing various basis selection algorithms. By random, we mean that the entries of Φ are selected from a standard Gaussian distribution. Since our goal is sparse image coding, it is best to pick a dictionary that can represent natural images effectively.

While dictionary learning algorithms do exist and random dictionaries may be useful, in this paper, we focus on using a predetermined set of basis functions.

C. Weighted Pseudo-Inverse

For an over-complete dictionary Φ we can construct the pseudo-inverse solution as

$$\boldsymbol{w} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{t} = (\Phi)^{\dagger}\boldsymbol{t}. \tag{6}$$

This solution is referred to as *method of frames* (MOF) and results in the reconstruction with the minimum L_2 norm. As discussed, such solutions are known to have very poor sparsity.

We can also represent all possible solutions in terms of a weighted pseudo-inverse. For any matrices $G \in \mathbb{R}^{M\times M}, G^{\ddagger} \in \mathbb{R}^{M\times M}$ such that

$$GG^{\ddagger} = I_{\boldsymbol{w}} \tag{7}$$

where $I_{\boldsymbol{w}}$ is a diagonal matrix with ones at every diagonal element corresponding to a non-zero element in $\boldsymbol{w}$, we can equivalently show

$$\boldsymbol{t} = \Phi\boldsymbol{w} = (\Phi G)(G^{\ddagger}\boldsymbol{w}) \tag{8}$$

$$\boldsymbol{w} = G\left(\Phi G\right)^{\dagger}\boldsymbol{t}. \tag{9}$$

This allows us to represent the problem as choosing G such that it satisfies (7) and minimizes $D(\boldsymbol{w})$. We will find a description of the weight matrix G imposed by several of the methods we discuss and use it as a means of comparison.

III. Overview of Techniques

In this section we describe some of the recent techniques used in sparse signal reconstruction: MP, BP, FOCUSS, and SBL. At the end, we give a general comparison of all methods.

A. Matching Pursuits

Matching pursuits [4] is a greedy algorithm that attempts to identify the bases that "match" the signal best, i.e. are most correlated with the signal. Reconstruction coefficients are built one at a time by first initializing $\boldsymbol{w}^{(0)} = 0$ and then iterating for K steps

$$i_k = \arg\max_i < (\boldsymbol{t} - \Phi\boldsymbol{w}^{(k)}), \phi_i > \tag{10}$$

$$w_{i_k} =< (\boldsymbol{t} - \Phi\boldsymbol{w}^{(k)}), \phi_{i_k} > \tag{11}$$

where i_k corresponds to the best matching basis at each iteration.

It is important to note that for non-orthogonal dictionaries (i.e. any over-complete dictionary) the algorithm may revisit the same basis function ϕ_i multiple times. This occurs when the use of another basis function ϕ_j, $j \neq i$ projects the error back into the dimension of ϕ_i. Accordingly the number of basis functions used after K iterations is $L \leq K$. After K iterations we know that, by construction, at least $M - K$ coefficients in $\boldsymbol{w}$ are zero. A sparse representation is found if the error becomes sufficiently small for $L < N$.

For over-complete dictionaries the method runs into difficulties if it makes an error on an initial choice of basis and wastes subsequent iterations trying to correct the error. Also, the algorithm can become hampered by revisiting the same basis functions multiple times.

We can recast MP as an iterative inverse problem of the form

$$\boldsymbol{w} = \bar{G}^T\Phi^T\boldsymbol{t} \tag{12}$$

with $\bar{G} \in \mathbb{R}^{M\times M}$. Without loss of generality we may assume that the bases are numbered in the same order that they are included in the solution by matching pursuits. Then we know that after k iterations $\bar{G}^{(k)}$ is of the form

$$\bar{G}^{(k)} = \begin{bmatrix} \bar{G}_L^{(k)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tag{13}$$

where $\bar{G}_L^{(k)} \in \mathbb{R}^{L^{(k)}\times L^{(k)}}$ consists of a series of weighted projections. At the k^{th} iteration only the i_k^{th} column of $\bar{G}_L^{(k)}$ is updated via

$$\bar{g}_{i_k}^{(k)} = \bar{g}_{i_k}^{(k-1)} + e_{i_k} - \sum_{j=0}^{L^{(k)}} < \phi_{i_k}, \phi_j > \bar{g}_j^{(k-1)} \tag{14}$$

where e_{i_k} is the i_k^{th} column of the identity matrix. We note that L is the number of basis selected by the algorithm. If a new basis is used L is incremented by one; if a previous basis is revisited L stays the same.

It is interesting to observe what occurs if we find a zero error solution with $L < M$ non-zero coefficients. We present the following theorem where $\boldsymbol{w}_L$ corresponds to the non-zero elements of a solution $\boldsymbol{w}$ found using MP and $\Phi_L = [\phi_1, \cdots, \phi_L]$..

Theorem 3.1: If $\boldsymbol{w}_L \in \mathbb{R}^L$ with $L < M$ has zero reconstruction error, then it is the minimum L_2 solution of $\boldsymbol{t} = \Phi_L\boldsymbol{w}_L$

Proof: If we have zero reconstruction error we know that any subsequent updates will not alter $\bar{G}$. This means that for any column i we have

$$\bar{g}_i = \bar{g}_i + e_i - \sum_{j=1}^{L} < \phi_i, \phi_j > \bar{g}_j \tag{15}$$

$$e_i = \sum_{j=1}^{L} < \phi_i, \phi_j > \bar{g}_j. \tag{16}$$

Writing the equation out in matrix form gives us

$$I_L = \bar{G}\Phi_L^T\Phi_L \tag{17}$$

$$\bar{G} = \left(\Phi_L^T\Phi_L\right)^{-1} \tag{18}$$

$$\boldsymbol{w}_L = \left(\Phi_L^T\Phi_L\right)^{-1}\Phi_L\boldsymbol{t} \tag{19}$$

■

In light of theorem 3.1 we can see that in the case of zero reconstruction error the the weight matrix G from equation 9 is of the form

$$G = \begin{bmatrix} I_L & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}. \tag{20}$$

Thus the weight matrix simply selects L basis functions to use in a pseudo-inverse solution.

B. Basis Pursuit

Basis Pursuit, developed by [2], uses methods in linear programing (LP) to find an optimal solution to a constrained linear inverse (3) when the constraint is the L_1 norm, i.e. $D(\boldsymbol{w}) = ||\boldsymbol{w}||_1$. As we saw previously, the L_2 norm used in MOF led to a quadratic optimization with linear equality constraints. The L_1 norm is a convex nonquadratic optimization problem given explicitly as

$$\min_{\boldsymbol{w}} ||\boldsymbol{w}||_1 \quad \text{s.t. } \boldsymbol{t} = \Phi\boldsymbol{w} \tag{21}$$

The standard LP formalization for a variable $\boldsymbol{x} \in \mathbb{R}^L$ is

$$\min_{\boldsymbol{x}} c^T\boldsymbol{x} \quad \text{s.t. } \boldsymbol{b} = A\boldsymbol{x}, \quad \boldsymbol{x} \geq 0 \tag{22}$$

where $c^T\boldsymbol{x}$ specifies the objective function, A and $\boldsymbol{b}$ specify equality constraints, and the additional boundary constraint $\boldsymbol{x} \geq 0$ is imposed. To show that the two problems are equivalent we write $\boldsymbol{w} = \boldsymbol{w}^{pos} + \boldsymbol{w}^{neg}$ where

$$w_i^{pos} = \begin{cases} w_i & \text{if } w_i \geq 0 \\ 0 & \text{if } w_i < 0 \end{cases} \tag{23}$$

and $\boldsymbol{w}^{neg}$ is likewise defined for the negative values of $\boldsymbol{w}$. Then we can write the L_1 norm as an inner product between $c^T = [\mathbf{1}, \mathbf{1}]^T$ and $\boldsymbol{x}^T = [\boldsymbol{w}^{pos}, -\boldsymbol{w}^{neg}]^T$. With $A = [\Phi, -\Phi]$ the problem becomes

$$\min_{\boldsymbol{w}} [\mathbf{1}, \mathbf{1}] \begin{bmatrix} \boldsymbol{w}^{pos} \\ -\boldsymbol{w}^{neg} \end{bmatrix} \qquad \text{s.t. } \boldsymbol{t} = [\Phi, -\Phi] \begin{bmatrix} \boldsymbol{w}^{pos} \\ -\boldsymbol{w}^{neg} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{w}^{pos} \\ -\boldsymbol{w}^{neg} \end{bmatrix} \geq 0 \tag{24}$$

Once this connection has been established BP can be implemented using some of the sophisticated methods developed in LP. Two highly efficient methods are the simplex algorithm and the interior-point method [2]. These methods find the global minimum and can take advantage of dictionaries with fast implicit algorithms.

C. FOCUSS

The FOCUSS algorithm [1] is formulated implicitly as a recursive weighted minimum norm of the form in (9). Although several variations have been studied, the basic form updates a diagonal weight matrix $G = \text{diag}(\boldsymbol{g})$ at every step based on the previous weight matrix and reconstruction coefficients. Two possible update rules are

$$g_i^{(k)} = w_i^{(k-1)} \tag{25}$$

and

$$g_i^{(k)} = g_i^{(k-1)}w_i^{(k-1)}. \tag{26}$$

According to [1], experimentation showed that the later formalization had faster convergence times and was more faithful to the initialization.

Because the algorithm does not have the nice convex properties of the SBL model, the choice of initial weights is critical to the sparsity of the solution. A poor initialization can result in the algorithm getting stuck in a very non-sparse local minimum. A common choice for the initial weights is min L_2 solution.

D. Sparse Bayesian Learning

Sparse Bayesian Learning takes advantage of the properties of Gaussian probability distributions to develop a convergent recursive method of finding a sparse $\boldsymbol{w}$. In this section we describe the methodology developed in [5] following the general presentation given in [3].

We are trying to solve equation (1) and begin by making the assumption that that the noise is i.i.d. Gaussian with some variance σ^2 (possibly unknown), i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$. Thus the conditional distribution of $\boldsymbol{t}$ can be written as.

$$p(\boldsymbol{t}|\boldsymbol{w};\sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{t} - \Phi\boldsymbol{w}\|^2\right) \tag{27}$$

For a given σ^2, the ML rule for selecting $\boldsymbol{w}$ is given by

$$\boldsymbol{w}_{ML} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{t};\sigma^2) \tag{28}$$

$$= \arg\max_{\boldsymbol{w}} p(\boldsymbol{t}|\boldsymbol{w};\sigma^2) \quad \text{(ML)} \tag{29}$$

$$= \arg\min_{\boldsymbol{w}} \|\boldsymbol{t} - \Phi\boldsymbol{w}\|^2. \tag{30}$$

Unfortunately, the ML rule is still under-constrained. We can use the non-sparsity inducing minimum L_2 solutions. We instead endeavor to use the MAP rule of the form

$$\boldsymbol{w}_{MAP} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{t};\sigma^2) \tag{31}$$

$$= \arg\max_{\boldsymbol{w}} p(\boldsymbol{t}|\boldsymbol{w};\sigma^2)p(\boldsymbol{w}) \quad \text{(MAP)}. \tag{32}$$

Here we must know the prior $p(\boldsymbol{w})$. SBL assumes a parametric form of the prior $p(\boldsymbol{w};\boldsymbol{\gamma}) \sim \mathcal{N}(0, \Gamma)$ with $\Gamma = \text{diag}(\boldsymbol{\gamma})$ and

$\boldsymbol{\gamma} = [\gamma_0, \cdots, \gamma_M]$, where the hyper-paremter γ_i is the variance of w_i.

$$p(\boldsymbol{w}; \boldsymbol{\gamma}) = \prod_{i=1}^{M} (2\pi\gamma_i)^{-\frac{1}{2}} \exp\left(\frac{w_i^2}{2\gamma_i}\right) \tag{33}$$

It is important to note here that this zero mean independent prior does not necessarily reflect the true nature of our ideal $\boldsymbol{w}$, but is an assumption we make to gain computational feasibility. Due to the Gaussian nature of the distributions we can calculate the pdf of $\boldsymbol{t}$ as

$$p(\boldsymbol{t}; \sigma^2, \boldsymbol{\gamma}) = \int p(\boldsymbol{t}|\boldsymbol{w}; \sigma^2) p(\boldsymbol{w}; \boldsymbol{\gamma}) d\boldsymbol{w} \tag{34}$$

$$= (2\pi)^{-\frac{N}{2}} |\Sigma_t|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\boldsymbol{t}^T \Sigma_t^{-1} \boldsymbol{t}\right] \tag{35}$$

$$= \mathcal{N}(0, \Sigma_t) \tag{36}$$

where

$$\Sigma_t = \sigma^2 I + \Phi\Gamma\Phi^T. \tag{37}$$

Ideally we want to choose parameters $(\sigma_{ML}^2, \boldsymbol{\gamma}_{ML})$ that maximize the probability of $p(\boldsymbol{t}; \sigma^2, \boldsymbol{\gamma})$. Unfortunately this problem, referred to as type-II maximum likelihood [5], does not have a simple solution. We will show shortly how we iteratively learn the most likely model parameters along with our optimal $\boldsymbol{w}$.

Now that we have defined all the necessary probability functions in terms of fixed parameters σ^2 and $\boldsymbol{\gamma}$ we can write the conditional probability of $\boldsymbol{w}$ given $\boldsymbol{t}$ as

$$p(\boldsymbol{w}|\boldsymbol{t}; \sigma^2, \boldsymbol{\gamma}) = \frac{p(\boldsymbol{t}; \sigma^2) p(\boldsymbol{w}; \boldsymbol{\gamma})}{p(\boldsymbol{t}; \sigma^2, \boldsymbol{\gamma})} \tag{38}$$

$$= \mathcal{N}(\boldsymbol{\mu}, \Sigma_w) \tag{39}$$

with

$$\boldsymbol{\mu} = \sigma^{-2} \Sigma_w \Phi^T \boldsymbol{t} \tag{40}$$

$$\Sigma_w = (\sigma^{-2}\Phi^T\Phi + \Gamma^{-1})^{-1} \tag{41}$$

At this point we have a description of our probability model $(\sigma^2, \boldsymbol{\gamma})$ and a probability of $\boldsymbol{w}$ conditioned on our assumed model and the observed signal $\boldsymbol{t}$. We employ the iterative *Expectation-Maximization* (EM) algorithm to find the best choices of $(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma})$ given $\boldsymbol{t}$. The two steps of the EM algorithm are outlined below.

E-step: We use the MAP rule to update our estimate of $\boldsymbol{w}^{(k)}$ based on our model parameters $(\sigma^2, \boldsymbol{\gamma})$. Since the conditional probability is a Gaussian, the MAP rule is simply the mean.

$$\boldsymbol{w}_{(k)} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{t}; (\sigma^2)^{(k)}, \boldsymbol{\gamma}^{(k)}) \tag{42}$$

$$= \boldsymbol{\mu}^{(k)} \tag{43}$$

M-step: We use the ML rule to update our new model parameters $((\sigma^2)^{(k+1)}, \boldsymbol{\gamma}^{(k+1)})$. Although we have defined $p(\boldsymbol{t}; \sigma^2, \boldsymbol{\gamma})$ in (34), it is insufficient to find $(\sigma_{ML}^2, \boldsymbol{\gamma}_{ML})$. To make the problem solvable we assume that our current guess $\boldsymbol{w}^{(k)}$ is correct, .i.e. we treat $\boldsymbol{w}$ as hidden variables in our problem, and then maximize the parameters over the complete data $\{\boldsymbol{t}, \boldsymbol{w}^{(k)}\}$. We use $p(\boldsymbol{t}, \boldsymbol{w}; \sigma^2, \boldsymbol{\gamma}) = p(\boldsymbol{t}|\boldsymbol{w}; \sigma^2) p(\boldsymbol{w}; \boldsymbol{\gamma})$ and then compute

$$\gamma_i^{(k+1)} = \arg\max_{\gamma_i} p(\boldsymbol{t}, \boldsymbol{w}^{(k)}; \sigma^2, \boldsymbol{\gamma}) \tag{44}$$

$$= (\Sigma_{w^{(k)}})_{i,i} + (\mu_i^{(k)})^2 \tag{45}$$

and

$$(\sigma^2)^{(k+1)} = \arg\max_{\sigma^2} p(\boldsymbol{t}, \boldsymbol{w}^{(k)}; \sigma^2, \boldsymbol{\gamma}) \tag{46}$$

$$= \frac{||\boldsymbol{t} - \Phi\boldsymbol{\mu}^{(k)}||^2 + (\sigma^2)^{(k)} \sum_{i=1}^{M} \left[1 - (\Sigma_{w^{(k)}})_{i,i}/\gamma_i^{(k)}\right]}{N}. \tag{47}$$

In [3] a variational formulation of the SBL method is used to show why sparse solutions are encouraged. We offer the following intuition: The solution is naturally sparse because we assume a zero mean prior for each w_i. There must be strong conditional evidence for non-zero w_i in (38) to overcome the prior (33) in the MAP estimate. It is also apparent that $w_i \to 0$ as $\gamma_i \to 0$.

We can also write the updates for SBL in the form of the weighted pseudo-inverse. If we let $G = \Gamma^{1/2}$ then we can represent the M step by the weighted pseudo-inverse (9) and the weight updates by

$$g_i^{(k)} = \sqrt{(\Sigma_w)_{i,i} + w_i^2} \tag{48}$$

with

$$\Sigma_w = \left(\Phi^T\Phi + (G^T G)^{-1}\right)^{-1}. \tag{49}$$

E. Comparison of Techniques

All of the techniques we have presented must make some compromises in order to find a tractable solution to the constrained linear inverse. In the following we highlight the assumptions and constraint modifications used.

Use Parametric Solutions: SBL drastically simplifies the search of a sparse $\boldsymbol{w}$ by assuming a probabilistic model of the data and using the EM algorithm to learn to the parameters along with the MAP $\boldsymbol{w}$.

Alter the Sparsity Constraint: Some methods relax the L_0 sparsity constraint by allowing $D(\cdot)$ to be the L_p norm. With $p = 1$ the constraint is the L_1 norm and as $p \to 0$ the constraint approaches the L_0 norm. Methods which use this constraint have no guarantee of being maximally sparse but gain ease and/or optimality (with respect to $D(\cdot)$) of implementation. BP is able to optimally solve the L_1 norm constraint and FOCUSS uses a recursive, non-optimal method for solving the generalized constraint for $p \in (0, 1]$.

Use a Heuristic: MP is an example of using a heuristic to greatly reduce complexity of implementation. As discussed, it is a bottom-up approach which has very low probability of being maximally sparse but can elegantly provide a balance between high sparsity and reconstruction error with very little computation.

In Table I we present several of the properties of the discussed methods. One important issue in evaluating a technique is how well it converges within its own framework, i.e.

TABLE I
COMPARISON OF TECHNIQUES

Method	$D(\boldsymbol{w})$	Globally Converg.[a]	Maximally Sparse[b]	Complexity / Iteration	Iterations	Complexity
MOF	L_2	yes	no	medium	one	very low
MP	L_2	no	no	low	many	low
BP	L_1	yes	no	medium	NA	medium
FOCUSS	L_p	no	no	medium	NA	medium
SBL	L_0	no	yes	high	few	high

[a] Globally convergent with respect to the chosen model
[b] Maximally sparse at the global minimum

with respect to its assumptions and choice of constraint. Of the algorithms presented only BP can guarantee that it will not become stuck in local minima. The FOCUSS algorithm, assuming a good initialization, is more likely to be convergent when $p = 1$ and loses optimality as $p \rightarrow 0$. Between SBL and FOCUSS (with a small value of p) experiments have shown that SBL is far less likely to be caught in a local minima [3].

Another important issue is if the global minimum of the technique corresponds to one of the maximally sparse solutions (the solution may not be unique). The quality depends on the norm being minimized and only SBL can guarantee a maximally sparse solution at its global minima. Consequently any errors made by SBL are a result of becoming stuck in local minima.

Finally it is important to consider the affects of parametric methods. When we assume a particular parametric model to find our solution we may be deviated from the true nature of our data. This is a major limitation of SBL and the variations of FOCUSS.

Although we try to provide a general feeling for the complexity of the various methods more quantitative analysis requires investigating the details of implementation, such as dictionary and method formulations, as well as the properties of the signals. We will present more specific comparisons in our application sections).

IV. APPLICATION I: IMAGE COMPRESSION

One of our objectives is to represent natural images with as few non-zero coefficients $\boldsymbol{w}_i$ as possible. We want to choose the optimal over-complete dictionary and algorithm that will achieve this goal. Additionally, we should acheive better performance than when using the optimal method with a complete dictionary.

A. Steerable Pyramid Dictionary

Steerable pyramids are used in several computer vision applications and in noise removal and enhancement techniques in image processing. Applying the steerable pyramid transform to an image decomposes the image into several unaliased subbands. Two useful properties of the steerable pyramid are its translation-invariance and rotation-invariance. This simply means that as a given image is translated or rotated in space, the information represented within each subband remains in that subband. The "steerable" name is derived from this rotation-invariance property [6]. Such a decomposition technique is very efficient when representing natural images, making the steerable pyramid basis functions a good dictionary choice. Since these basis functions are directional derivatives, each can be viewed as a translated, scaled, and rotated version of a single kernel.

The steerable pyramid transform is a multi-scale, multi-orientation decomposition. The transform can be represented as a series of filter banks. First an image is decomposed into a residual highpass subband and a lowpass subband. The lowpass subband is then divided into k bandpass subbands and a residual lowpass subband, where each subband represents a certain orientation. The residual lowpass subband is subsampled by a factor of 2 in both the x and y directions and then divided into another set of k orientation subbands and a lowpass band, and the process continues recursively. The number of orientation subbands k is equal to one more than the order of the directional derivative used. Thus, a set of third order directional derivatives will create four subbands.

In our filter bank, let us refer to the initial high pass filter as $H_0(\omega)$ and the initial low pass filter as $L_0(\omega)$. As mentioned earlier, the image passed through $L_0(\omega)$ is then passed through a set of bandpass filters, $B_k(\omega)$, and a low pass filter, $L_1(\omega)$. We can write the Fourier magnitude of $B_k(\omega)$ in terms of an angular component, $A(\theta)$ and a radial one, $B(\omega)$:

$$B_k(\vec{\omega}) = A(\theta - \theta_k)B(\omega) \tag{50}$$

where $\theta = \tan^{-1}(\omega_y/\omega_x)$, $\theta_k = 2\pi/k$, and $\omega = |\vec{\omega}|$. The angular component can be described by the formula $A(\theta) = cos(\theta)^n$, where n indicates the order of the directional derivative used. Using the notation above, there are three constraints we must impose:

$$L_1(\omega) = 0 \qquad \text{for } |\omega| > \pi/2 \tag{51}$$

$$|H_0(\omega)|^2 + |L_0(\omega)|^2 \left[|L_1(\omega)|^2 + |B(\omega)|^2\right] = 1 \tag{52}$$

$$|L_1(\omega/2)|^2 = |L_1(\omega/2)|^2 \left[|L_1(\omega)|^2 + |B(\omega)|^2\right]. \tag{53}$$

These constraints ensure that no aliasing takes place when subsampling, that the system response is unity, and that the recursive process can occur [7].

1) Example: Decomposing a 128×128 pixel image: A 128×128 bitmap image of a white circle on a gray background was decomposed into 2 pyramid levels and 4 subbands using Matlab code obtained from [8]. This can be seen in Figure 1(a). Figure 1(b) shows the residual high pass information along with the four orientation bands at the finest scale, or first pyramid level(128×128 pixels). Figure 1(c) shows the

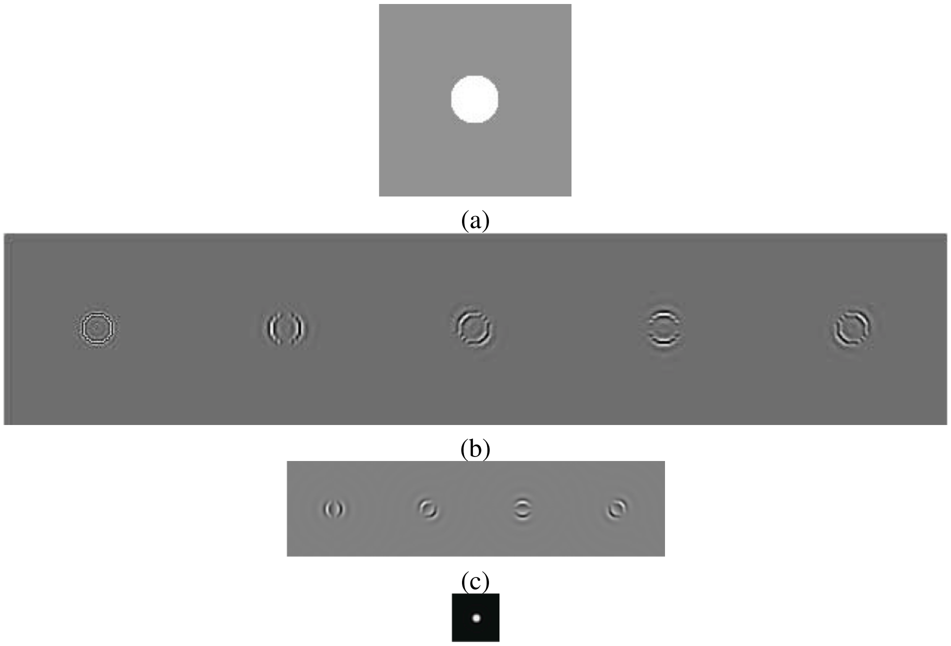


Fig. 1. Steerable Pyramid Decomposition

four orientation bands at the second pyramid level (64×64 pixels) while Figure 1(d) shows the residual subsampled low pass information (32×32 pixels).

2) Building a Set of Steerable Pyramid Basis Functions: The size of a dictionary of steerable pyramid basis functions depends on the number of scales and orientations desired. To create a dictionary of c basis functions for an $r \times r$ image, we can create a $c \times 1$ vector, where each element represents a single pixel within all subbands of the image decomposition. We can generate one basis function by creating an impulse within this vector and then reconstructing an $r \times r$ matrix with this impulse, using a Matlab function obtained from [8]. By repeating the process for all elements within the vector, an $r^2 \times c$ matrix Φ is created, where each column in Φ represents one basis function within the dictionary.

B. Implementation

Implementing the SBL algorithm using the steerable pyramid dictionary required large matrix multiplications and inversions. We performed these matrix calculations using the CLAPACK (Linear Algebra Package for C) and ATLAS (Automatically Tuned Linear Algebra Software) software packages. These packages included BLAS (Basic Linear Algebra Subprograms) subroutines that performed optimal calculations for various types of matrices.

The SBL algorithm was run on the PSI Fast Storage Cluster here at the University of California at Berkeley. All nodes and frontends on the cluster contained dual 3.0 GHz Pentium 4 Xeon chips and 3 GB of RAM. Within our implementation of the SBL algorithm, matrix storage was $O(M^2)$ and matrix multiplication was $O(M^3)$. Memory issues were encountered for a 64×64 image with a 4.8 times overcomplete dictionary, which contained 2 scales and 4 orientation subbands. 700 MB were required to store the dictionary while all matrices used in the algorithm required more than 5.3 GB of RAM. When performing the algorithm on a 32×32 image, only 340 MB of RAM were required to store all matrices. Looking at the computation time, each SBL iteration on a 32×32 image took approximately 13 seconds. We estimate that one iteration on a 64×64 image would take approximately 830 seconds. As a result of the memory and computational issues, the largest image on which SBL was performed was 48x48 pixels.

C. Results

We performed a series of experiments on three 32×32 pixel images, which we called lena100-32, einstein32, and firefox32. The Lena image was originally 512×512 pixels; however, it was downsampled to 100×100 and then cropped. The Einstein image we used was simply a cropped version of a 256×256 pixel image of Einstein and the Firefox logo was originally 32×32.

In our first experiment, we ran 50 iterations of SBL on the einstein32 image using a 4.8 times overcomplete steerable pyramid dictionary (2 scales and 4 orientations) and a 4 times overcomplete DCT dictionary. We then retained only the most significant n coefficients, where n varied from 10 to 1020, with a step size of 10. The mean squared error corresponding to the reconstructed image was taken for each step size. A graph of the MSE versus the number of retained coefficients can be seen in Figure 2. We also used a complete DCT dictionary as a baseline. Since that dictionary is complete, the coefficients were computed using $\boldsymbol{w} = \Phi^{-1}\boldsymbol{t}$. The steerable pyramid dictionary performed better when between 200 and 600 coefficients were retained. Figure 3 shows the original Lena image along with reconstructed versions using the two dictionaries when 250 coefficients were retained. The image

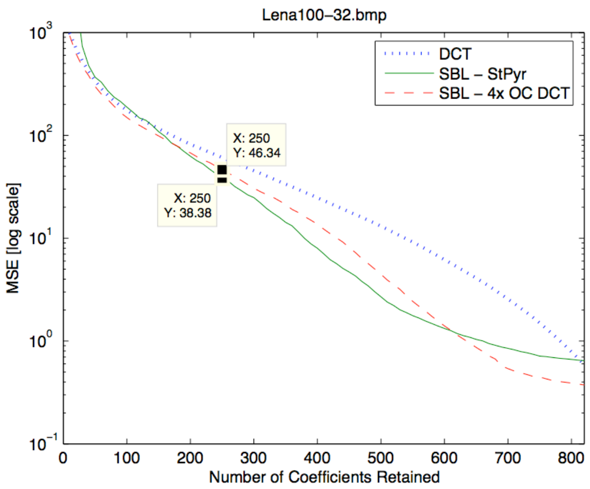
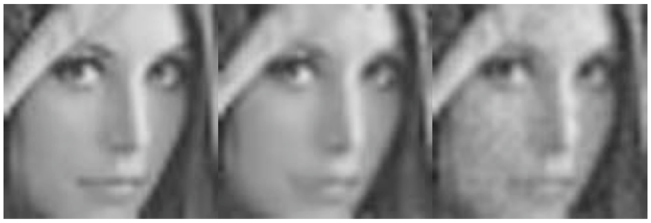


Fig. 2. Comparison of Dictionaries



(a) (b) (c)

Fig. 3. Lena100-32: (a) Original, (b) SBL with StPyr, (c) SBL with 4x OC DCT

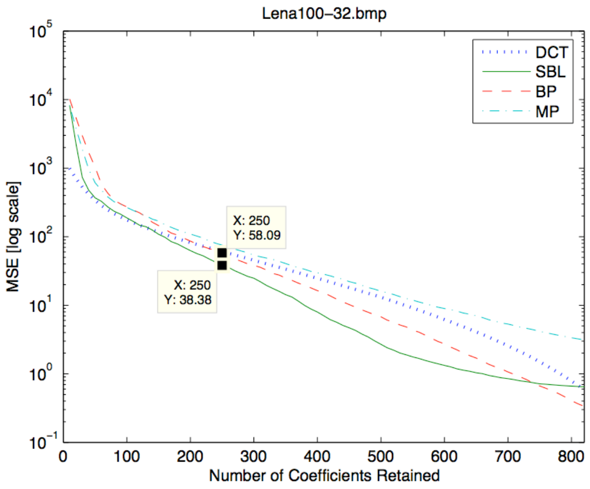
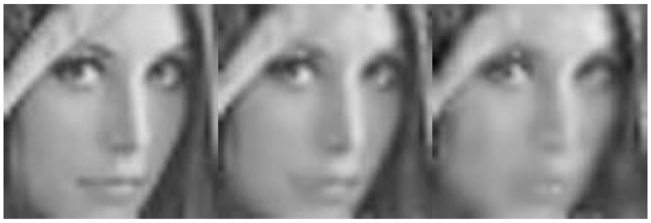


Fig. 4. Comparison of Methods on lena100-32.bmp



(a) (b) (c)

Fig. 5. Lena100-32: (a) Original, (b) SBL, (c) Basis Pursuit

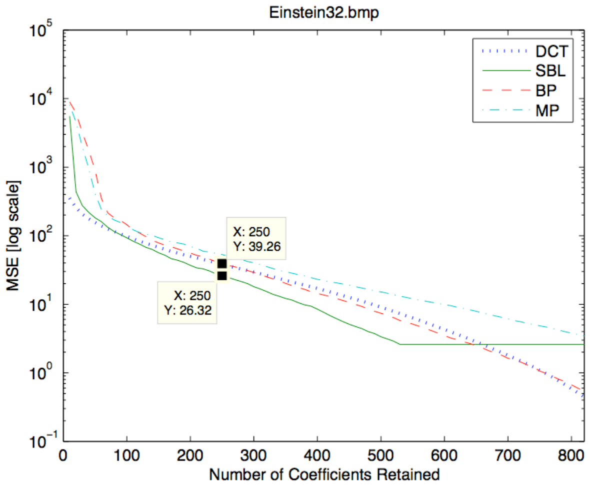
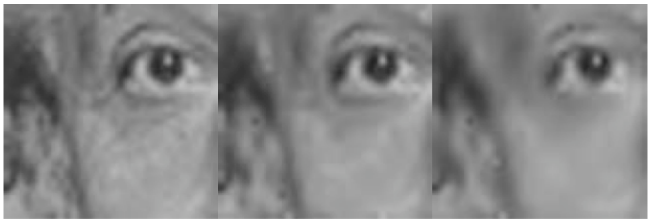


Fig. 6. Comparison of Methods on einstein32.bmp



(a) (b) (c)

Fig. 7. Einstein32: (a) Original, (b) SBL, (c) Basis Pursuit

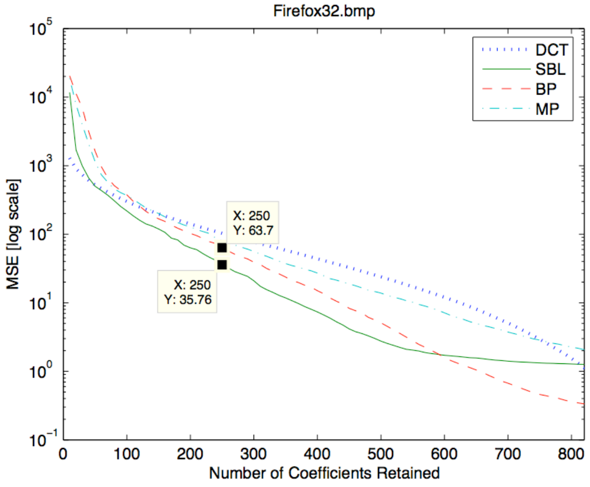
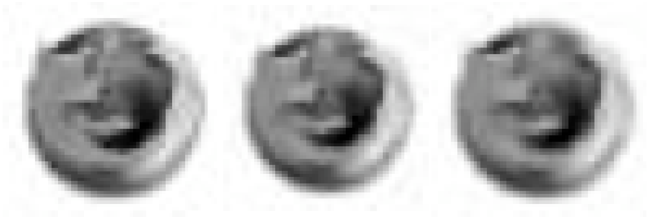


Fig. 8. Comparison of Methods on firefox32.bmp



(a) (b) (c)

Fig. 9. Firefox32: (a) Original, (b) SBL, (c) Basis Pursuit

reconstructed with the steerable pyramid is clearer and less speckled than the one reconstructed with the overcomplete DCT. While these results are not conclusive evidence that the steerable pyramid is the optimal set of basis functions to use, they do validate our dictionary choice.

We then performed three separate basis selection methods (SBL, Matching Pursuits, and Basis Pursuit) using the steerable pyramid dictionary on the Lena image. We performed 50 iterations of SBL, 5000 iterations of Matching Pursuit, and approximately 15 iterations of Basis Pursuit. As in the experiment mentioned before, we then found the mean squared error corresponding to a certain number of coefficients selected by each method. As a baseline, the performance of the DCT is shown in Figure 4 as well. It is clear that SBL peformed better than all methods when using a sufficiently sparse coefficient vector. Similarly to the previous experiment we show the reconstructed Lena image in Figure 5 using 250 coefficients selected by SBL and Basis Pursuit. One can see that with Basis Pursuit, the image is more blurred and Lena's left eye, nose, and lips are less defined.

The same experiments were performed on the Einstein image and on the Firefox image. As shown in Figures 6 and 8, SBL outperformed the other methods in the areas where a sparse solution would be obtained. Although the reconstructed images using 250 coefficients for SBL and Basis Pursuit are similar for the Einstein image in Figure 7, BP had an MSE of 39.26 while the MSE using SBL was 26.32. It is interesting to note that SBL outperformed the other methods for a synthetic image, the Firefox logo, when approximately 100 to 600 coefficients were retained. We can conclude from these results that SBL gives a more accurate representation of an image when a sparse solution is required.

V. Application II: MEG Reconstruction

Magnetoencephalography (MEG) and electroencephalography (EEG) are non-invase brain image techniques that attempt to measure brain activation in the cerebral cortex. Both methods have very high temporal resolution compared to other brain imaging techniques but suffer greatly in spatial resolution as a result of limited spatial sampling; activation at 10,000 locations in the brain must be recovered from only 275 sensor placed on the surface of the scalp [10]. Although MEG is technically more challenging to implement than EEG, it has shown a much greater robustness to noise. As a consequence much research has focused on recovering the locations of brain activation from MEG measurments. In this section we present a novel application of SBL to the MEG inverse problem.

The unconstrained nature of MEG reconstruction necessitates some form of regularization. Currently studied inverse methods attempt to use prior knowledge of brain activation characteristics to create a convex solution space. According to [1], brain activation appears to consist of localized energy sources, i.e. the activity is "often limited in spatial extent, but otherwise is distributed over arbitrarily shaped areas." Thus much research has looked for solutions that are somehow sparse in nature.

A. Forward Model

A primary assumption of neuroimaging is that electric currents correspond directly to brain activation. The first step in using MEG is to create a forward model of how these currents in the brain are mapped to the magnetic fields recorded at the sensors. The physics of electromagnetic waves as summarized by Maxwell's equations dictate the magnetic field $b(\boldsymbol{r})$ anywhere in space and a function of a primary current $i(\boldsymbol{r})$ where $\boldsymbol{r}$ is a location vector. Assuming that the conductivity in the head is known, a linear representation of the mapping can be found using a quasi-static approximation of Maxwell's equations and superposition [9]. A great deal of work has gone into developing elaborate multilevel head conductivity models that account for differences in skin, skull, CSF, and brain tissue [10].

We will denote samples of $b(\boldsymbol{r})$ as $\boldsymbol{t} \in \mathcal{R}^N$ where N is the number of sensors, and samples of $i(\boldsymbol{r})$ as $\boldsymbol{w} \in \mathcal{R}^M$ where M is the number of location in our model of the brian. We assume that $M >> N$; in this study $M = 9981$ and $N = 73$. The matrix $\Phi \in \mathcal{R}^{N \times M}$ is the approximate linear mapping from $\boldsymbol{w}$ to $\boldsymbol{t}$. We represent the forward model as equation (1).

B. Inverse Calculation

If we accept the validity of our forward model, calculating the brain currents $\boldsymbol{w}$ is an under-constrained linear inverse as described in section II.A general overview of methods for MEG is given in [10]. The authors present the linearly constrained minimum variance (LCMV) beamformer and the multiple signal classification (MUSIC) algorithms as well as the Tikhonov regularized version of the pseudo inverse (TR). In [1], a method called *Focal Underdetermined System Solution*, (FOCUSS) is used with very low resolution brain models, and in [9], a method called Best Orthogonal Basis is applied to more sophisticated brain models with limited success. In this paper we present novel application SBL to the MEG inverse problem using the same models as [10],[9].

C. Results

We tested SBL following the general evaluation procedure described in [10]. We constructed a synthetic test current w consisting of two patches of current, one positve and one negative. Both had magnitude one and consisted of eight contiguous locations. We created the corresponding measurement t using the forward model with zero noise, and tried to recover w from t using MOF, MP, and SBL. All methods were implemented using the programs and platforms described in section IV-B and we found that the initialization parameters which led to fastest convergence were ($\sigma \approx 10^{-5}$, $\gamma \approx 10^{-2}$).

The results, denoted w_{MOF}, w_{MP}, and w_{SBL} are shown in Figure 10 on a smoothed cortical surface. We see that w_{MOF} is highly distributed over the brain and does not correspond to w. Although w_{MP} is a very sparse signal it too fails to resemble w. Only w_{SBL} shows the two patches along with the addition of a third negative spot near the original positive patch.

Although SBL is clearly the most faithful reconstruction of the original current, the mean squared errors (MSE) of the

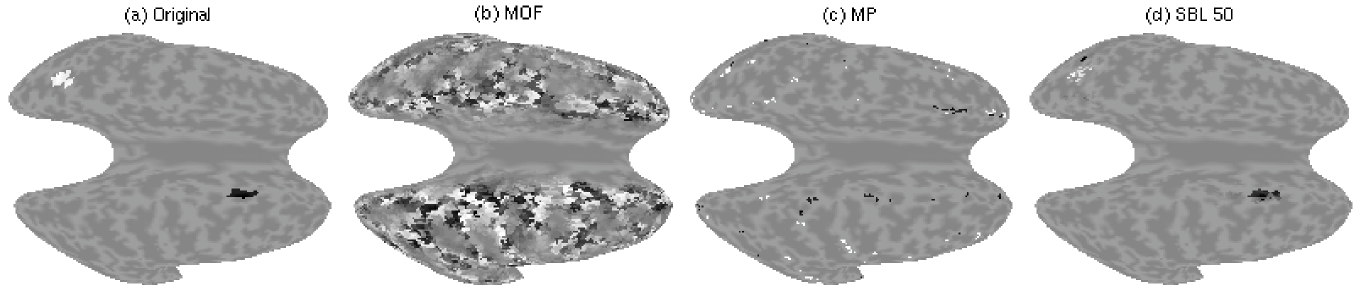


Fig. 10. Cortical brain activation for original current (a) and reconstructed currents using MOF (b), MP (c), and SBL with 50 iterations (d). White indicates positive and black negative. The amplitudes for MOF and MP have been greatly amplified to allow visualization; SBL 50 is shown on the same scale as the original signal.

reconstructed signals does not show significant differences between the rmethods. To analyze the signals we treat the patches of current as targets. For each location in the brain we want to determine, using some decision rule and our reconstructed brain current, whether or not that location corresponds to a non-zero current in the original brain. We can an achieve this by thresholding the magnitude of the reconstructed currents using a given threshold T. We evaluate our target detection using the receiver operating characteristic (ROC) [10]. The ROC presents the the ratio of true positives (TP) versus false positives (FP) over all decisions rules (In our case over all thresholds T). A high ratio indicates good detection and the ROC allows us to compare methods over the full range of FP penalties.

In Figure 11 we present the ROC for SBL with different numbers of iterations. Performance is low after only two iterations but is near optimal after only ten. We see that stopping at 50 iterations gives the best performance in the high FP region, and continuing to 100 iterations gives the best performance in the low FP regions. This occurs because as SBL continues to converge, the results become more sparse, and very few positives will be identified.

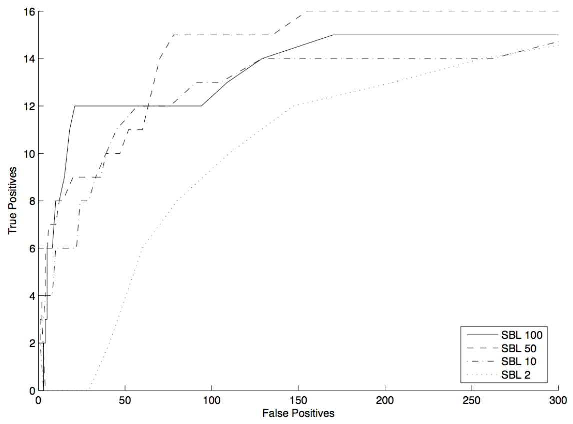


Fig. 11. FROC curves for different iterations of SBL with 16 total targets.

In Figure 12 we compare the two best SBL implementations against MOF. As we suspected SBL greatly outperforms MOF over the entire curve. We do not show MP because it fails to locate any of the 16 TPs at any threshold.

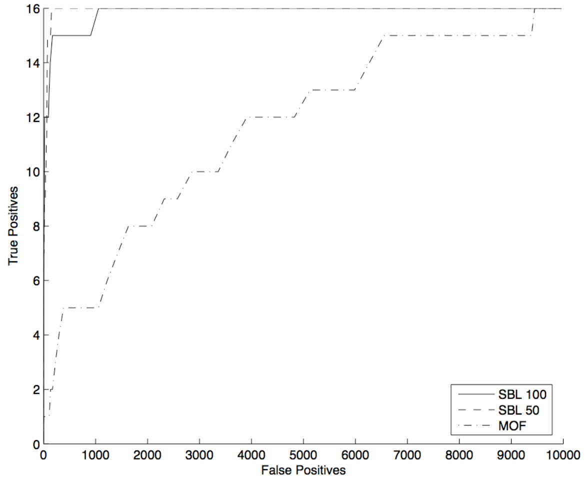


Fig. 12. FROC curves for SBL and MOF with 16 total targets.

Our results show that SBL can be applied to the MEG inverse problem. Future analysis of SBL should study the affects of additive noise in the inverse model, analyze the results over a large number of test signals, and compare its performance against other state-of-the art localization techniques such as TR, MUSIC and LCMV. As real data becomes available more substantial claims can be made as to the appropriateness of any of these methods.

Additionally, it would be beneficial to apply SBL to multiple images over time. Although this is initially a daunting undertaking due the computational complexity, it may be feasible with a recursive implementation. For each time SBL provides both the reconstructed current as well as model parameters. It seems logical to exploit the correlation in time of the model to jump-start the implementation at a subsequent time. Additionally, a multi-scale approach could could allow faster implementation by isolating reconstructed brain currents to some subset of locations.

VI. Concluding Remarks

Indeed, the Sparse Bayesian Learning algorithm arrived at very sparse representations of our test signals. Coupled with the Steerable Pyramid dictionary, it performed better than all other basis selection algorithms in our image compression trials, for both natural and synthetic images. SBL was also highly successful at reconstructing source locations from MEG measurements. The main drawback of this algorithm is its computational complexity, an issue which has kept us from testing our implementation on images of significant size. Since the effectiveness of the algorithm has already been demonstrated, the next step is to improve the implementation of SBL by incorporating fast dictionaries, the pruning of unnecessary basis functions between iterations, and other tricks to make its widespread use more feasible.

ACKNOWLEDGEMENTS

The authors wish to thank Professor Avideh Zakhor for an enjoyable course, Pierre Garrigues for guidance, Dr. Mike Eklund for providing data, and Parvez Ahammed for his helpful comments.

REFERENCES

[1] I. F. Gorodnitsky, J. S. George, and B. D. Rao. Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm. *Electroencephalography and clinical Neurophysiology*, 95(4):231–251, 1995.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[3] D. P Wipf and B. D. Rao. Spare bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004.

[4] S. Mallat and Z. Zhong. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, 41:617–643, 1993.

[5] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning*, 1:211–244, 2001.

[6] E. P. Simoncelli, W. T. Freeman, E. H. Adelson and D. J. Heeger Shiftable Multi-scale Transforms. *IEEE Trans. in Information Theory*, 2004.

[7] E. P. Simoncell, W. T. Freeman The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation *IEEE 2nd International Conference on Image Processing*, 1995.

[8] http://www.cns.nyu.edu/ eero/steerpyr/

[9] J. M. Eklund, R. Bajcsy, J. Sprinkle, and G. V. Simpson. Computing meg signal sources. In *Proceedings of the 2005 Computational Systems Bioinformatics Conference Workshops*, 2005.

[10] F. Darvas, D. Pantazis, E. Kucukaltun-Yildirim, and R. M. Leahy. Mapping human function with meg and eeg: methods and validation. *NeuroImage*, 23(1):S289–S299, 2004.