Statistical Methods for Image and Signal Processing

by

PHILIP ANDREW SALLEE

B.S. (Biola University) 1993
M.S. (University of California, Davis) 2002

DISSERTATION
Submitted in partial satisfaction of the requirements for the degree of
DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES
of the
UNIVERSITY OF CALIFORNIA
DAVIS

Approved:

Professor Bruno A. Olshausen (Chair)

Professor Zhaojun Bai

Professor Naoki Saito

Committee in Charge

2004

Statistical Methods for Image and Signal Processing

Copyright © 2004

by

Philip Andrew Sallee

To Trisha, my love, who always encourages me to follow my dreams. Thanks for listening to my wild ideas for countless hours. Without your constant love and support, this work could not have happened. You are my true companion.

And to my daughters Charisse and Karina. May you dream big and never let go, and never stop wondering.

Acknowledgements

So many people have impacted this work, that I cannot begin to thank them all. But I want to particularly thank my advisor, Bruno Olshausen, for countless hours spent mentoring me, listening to my ideas, guiding me and reviewing my work. This work is a result of his vision which I caught and hope to pass on to others. I also want to thank the rest of my thesis committee, Naoki Saito and Zhaojun Bai, for their patience, support and encouragement until this was completed. Thanks also, to those in the Computer Science department who have been so supportive of my efforts. A special thank you is due to Tye Stallard for that email which inspired my steganography work, and to Eero Simoncelli for providing source code, ideas and discussions.

Many thanks to those at the Center for Neuroscience who helped out on many occasions, reviewed my work, contributed ideas, and sometimes just kept things interesting: Issac Trotts, Maysha Mohamedi, Scott Murray, and Jeff Colombe. Special thanks go to Jeff Johnson for his help with the EEG data, Matthew Godwin for investigating matching pursuit algorithms and Surya De who helped to write JPEG tools. Kevin O'Conner, thank you for many discussions and for providing the natural sound database.

Many thanks are due to my family without whose support I would not have finished. To my wife and children, thank you for your sacrifices and your encouragement. Mom and Dad, thank you for your loving support and advice through all these years. And my brother, Greg, thanks for reading my papers and for lots of interesting discussions. Finally, thanks be to God, author and designer of all things. You have given us more to explore than we could begin to understand after many lifetimes, and I am richly blessed to have such wonderful people in my life to explore it with.

Abstract

Statistical methods provide a principled means for solving many types of problems which require the estimation of missing or uncertain information. This dissertation discusses methods for adapting statistical models to images, sounds and other types of signals for applications in image and signal processing. Wavelets provide a multi-scale representation which has been shown to be well suited for describing many naturally occurring signals. These are typically designed by hand based on certain mathematical properties and may not achieve the best match to the data. We describe an approach for using an overcomplete wavelet framework as part of a generative statistical model with a sparse prior placed on the wavelet coefficients. The wavelet functions are adapted to a given dataset by maximizing the average log likelihood of the model. This is demonstrated for natural images, sounds, and EEG data. The learned representations are shown to have a higher degree of sparsity than other wavelet bases. This statistical framework also provides a principled approach for performing certain types of signal estimation, such as denoising, in terms of a statistical inference process. We explore two inference methods for the overcomplete wavelet models presented: A Gibbs sampling method, and a greedy optimization procedure known as matching pursuit. We also demonstrate how a statistical model may be applied to a form of secure information hiding, known as steganography, in which the objective is to hide information in an image or some other media so that it cannot be detected without a cryptographic "key". This model-based approach provides a means for maximizing the capacity of stored information while obtaining provably secure steganography insofar as the model is accurate. Using this methodology, a steganography method is proposed for JPEG images which achieves higher embedding efficiency and message capacity than previous methods, while remaining secure against first order statistical attacks. Methods for applying statistical models for steganalysis, the art of detecting steganographic messages, are also presented.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We are continually immersed in a sea of naturally occurring signals produced by our environment. Every moment, our brain processes these "natural" signals in order to make sense of the events taking place around us. Images, sounds, and other sensations are continuously fed to our brain through our nervous system. In order to interact with our world, these signals must be processed and related to the ongoing sequence of events in our environment. The speed, accuracy, and apparent ease by which the brain interprets this sensory data may lead us to assume that this is a trivial task. In fact, we rarely need to consider that this type of signal processing is going on at all. Because most of this processing occurs below the level of conscious thought, it is natural to assume that information comes to us already explained in terms of things in our world, and that relating the sensory input to objects in our environment is a trivial task.

Decades of scientific research into computer vision and signal processing have taught us that this is not at all the case, however. The cohesive picture we maintain of our environment does not come without much complex analysis. Noisy and ambiguous data streams must first be interpreted and glued together to

produce a cohesive picture of our world. Missing or noisy data must be inferred, and the brain must choose from many possible explanations which events are responsible for giving rise to these raw signals. How this computation takes place in the brain is still largely a mystery, pieces of which we have only started to understand. We are even further from being able to recreate this process in order to match the signal processing capabilities of the brain.

Were we able to interpret natural signals to this degree, the effects would be staggering. Human-machine interaction would truly be revolutionized. People could be automatically and accurately identified from their appearance and speech. Speech recognition would no longer be restricted to single words, or be speaker dependent. Machines could understand gestures and facial expressions. Faces could be accurately picked out of a crowd by computer without error. Machines could allow the blind or deaf to "see" or "hear" what is going on around them by intepreting these events into speech or images. Biomedical neural interfaces could possibly be designed to allow them to experience these sensations directly. Multimedia streams would be compressible far beyond what is currently possible, supporting the constant demand for information from today's high-tech mobile world. Images, video and text could be searched based on content and meaning, rather than only by keywords and filenames. Robots would be able to navigate and manipulate complex environments, revolutionizing industry, modern convenience and possibly even transportation. These are only a few of the many possibilities.

While some of these technologies exist today in a rudimentary form, they are generally quite limited and error prone and not able to handle the complexity of real-world data or to interpret signals to the level of abstraction necessary to accurately identify meaningful events. For example, to recognize objects it is

necessary to first estimate and compensate for the specific viewpoint and lighting conditions, and variations of the same type of object such as in color, size or form. Current object recognition methods are usually restricted to locating two dimensional patterns rather than reconstructing and matching three dimensional shapes. For those that estimate shape, they are subject to significant limitations in lighting conditions, surface properties or viewpoints. For speech, it is necessary to take into account different accoustical properties of the room, background noise, separate multiple speakers, and distinguish speaker specific characteristics such as gender, age, regional accents or whether a person has a cold, from the phonetic information being uttered. Current methods for speech recognition are still far from achieving this level of accuracy.

One reason these tasks have proven to be so difficult is because interpreting the causes of real-world signals is an ill-posed problem. For any given signal there can be many explanations, some of which are more likely than others. For images, for example, the true causes of the data are surfaces, objects, and light sources in three spatial dimensions which are projected onto a 2D plane. For any image, there are an infinite number of 3D explanations. Traditional approaches for obtaining more meaningful representations of signals often take the form of feedforward computations, performing a fixed series of operations on a signal in order to produce a new output signal. Linear transforms and thresholding operations are examples of such feedforward approaches. While transforms account for certain types of statistical dependencies present in the signal, this direct approach has clear limitations. While it is possible in theory to accomplish any arbitrary computation on a signal using combinations of transforms and nonlinear operations, it is not clear how to design such a system or select the transforms and operations to use. Additionally, this approach is fundamentally unsuited to han-

dling the types of ambiguities that are present in natural signals. In many cases, multiple descriptions may provide an equally good fit to the data. Deterministic filtering approaches which accept an input signal and produce a single output cannot simultaneously entertain multiple possibilities which need to be disambiguated. Also, the complexity of solving perceptual tasks in such a feedforward manner may be too prohibitive, as it requires a mapping between every signal that could be encountered and its most probable explanation.

On the other hand, describing the process by which many signals are generated, up to some approximation, appears more managable. While computing this generative function is not necessarily any less difficult, there is reason to believe that efficient means of computing such a function will exist for naturally occurring signals in terms of the actual processes by which they were generated, or by a simplified approximation of these processes. The cost of computing an explanation for a signal can then be spread out over an iterative search process where candidate explanations for a signal are identified and compared before reaching a conclusion regarding the causes of the signal. For a biological system, in which signals are temporally related and decisions often must be made on incomplete information, this process provides a natural way to use initial, possibly less accurate, hypotheses until later information is obtained which can determine which hypothesis is correct. At each point in time the current hypothesis needs only to be refined, rather than having to compute a complete analysis at every step. In this manner, a working model of the world can be generated and updated over time, and current events can be related to and predicted by past events. The primary difficulty with using generative methods for signal analysis is that while a generative model provides a straightforward prescription for generating the data, it may be intractable, depending on the model, to accurately estimate

the modeled "causes" of a particular signal. The challenge, therefore, is to select a model for which known algorithms allow these "causes" to be inferred.

In this thesis, I describe statistical methods based on this generative modeling approach. While recreating the brain's perceptual abilities in order to solve the problems previously mentioned is beyond from the scope of this paper, the intent here is to further develop the generative modeling approach using models which are simple enough to explore these ideas, yet still accurate enough to be of some use for image and signal processing applications. It is hoped that this exploration will aid in the future development of inference methods for more powerful models for these types of signals. The next sections provide a more mathematical basis and background for these concepts and introduces concepts necessary to describe the specific approaches taken in this thesis.

1.1 Generative models

According to one theory, the brain accomplishes perceptual tasks through an iterative process, using a generative statistical model of its environment to infer the most probable causes of each signal. A generative model assigns probabilities to data in terms of how the data may have been generated from an initial set of parameters referred to as "causes" of the data. Specifically, each data instance $\mathbf{D}$ is assumed to be generated by a function f of unknown causes $\mathbf{C}$ plus noise $\mathbf{n}$:

$$\mathbf{D} = f(\mathbf{C}) + \mathbf{n} \tag{1.1}$$

The causes $\mathbf{C}$ are assumed to be distributed according to some specified prior distribution $P(\mathbf{C})$. An interpretation, or explanation, of a given signal $\mathbf{D}$ thus amounts to a particular assignment of the causes $\mathbf{C}$. The probability for the

signal can be specified by summing over the likelihoods $P(\mathbf{D}|\mathbf{C})$ for all possible causes, weighted by the prior probability of each interpretation:

$$P(\mathbf{D}) = \sum_{\mathbf{C}} P(\mathbf{D}|\mathbf{C})\ P(\mathbf{C}) \tag{1.2}$$

To infer the most probable explanation for a given input signal, competing interpretations can be weighed according to their likelihood under the model:

$$P(\mathbf{C}|\mathbf{D}) \propto P(\mathbf{D}|\mathbf{C})\ P(\mathbf{C}) \tag{1.3}$$

Information collected about the environment is incorporated into the generative model in the form of the function f, and the priors over the causes, $P(\mathbf{C})$, each of which may be parameterized and those parameters adapted to the structure of the data. This statistical framework thus provides a principled approach for reasoning about novel signals by using top-down information previously obtained from the same environment. The primary goal of this dissertation is to demonstrate how to apply this statistical framework to problems in image and signal processing.

Given the complexity of naturally occuring signals, obtaining the correct statistical model for those signals might seem a hopeless task. Unsupervised learning methods, however, provide a promising approach which avoids the necessity of having explicit training information with which to select the correct model. These methods provide a means for adapting a statistical model based on information-theoretic principles, such as minimizing the Kullback-Leibler (KL) divergence between the distribution over signals generated by the model and the distribution over signals taken from the environment. One such method which provides a primary focus for this dissertation is known as sparse coding [24].

1.2 Sparse coding

Sparse coding refers to the process of modeling data as generated by a linear superposition of basis functions having "sparse" distributions characterized as being peaked at zero with heavy tails. Thus, it is assumed that any given signal may be usually described in terms of only a small number of basis functions, which are considered to be the statistically independent causes of the signal. The basis functions, as well as the exact shape of the sparse distributions, are fit to a set of signals by maximizing the model likelihood for the data set. Sparse coding is well suited to situations where there may be noise added to the signals, or where the dimensionality of the model parameters is greater than the dimensionality of the signals. This generative modeling approach provides a principled framework for dealing with a variety of inverse problems commonly faced in signal analysis, such as deconvolution or denoising, by inserting into the generative model the appropriate distortions or transformations which are assumed to have occurred to the signal.

Each observed instance of a signal $\mathbf{x}$ is assumed to be generated by a linear superposition of basis functions which are columns of an N by M weight matrix $\mathbf{A}$, with the addition of Gaussian noise $\mathbf{n}$:

$$\mathbf{x} = \mathbf{A}\ \mathbf{s} + \mathbf{n} \tag{1.4}$$

where $\mathbf{x}$ is an N-element signal vector, and $\mathbf{s}$ is an M-element vector representing the assumed statistically independent and sparse sources of the signal. The probability of generating a signal $\mathbf{x}$, given a source vector $\mathbf{s}$ and assuming *i.i.d.* Gaussian noise $\mathbf{n}$ (with variance $1/\lambda_{\mathbf{n}}$), is

$$P(\mathbf{x}|\mathbf{s},\theta) = \frac{1}{Z_{\lambda_{\mathbf{n}}}} e^{\frac{\lambda_{\mathbf{n}}}{2}|\mathbf{x}-\mathbf{A}\mathbf{s}|^2} \tag{1.5}$$

where θ denotes the parameters of the model and includes $\lambda_{\mathbf{n}}$ and $\mathbf{A}$, as well as parameters for the prior $P(\mathbf{s})$.

While currently limited to linear generative models, sparse coding provides a first step towards more advanced models in the form of a generative framework for modeling data in terms of statistically independent components. Prior work in sparse coding and ICA has been limited to working with small blocks of data, however, as the existing algorithms are not easily scaled to larger dimensionality. This limits their usefulness for image coding applications, as processing images in managably sized blocks introduces artifacts at block boundaries and fails to capture statistical dependencies between blocks. Chapters 2 and 3 demonstrate how the sparse coding framework can be applied to larger images, sounds and other signals efficiently by means of a wavelet filter bank parameterization of the basis functions. It is hoped that extending the sparse coding framework in this manner will pave the way for the development of more advanced generative models for these types of data, such models of images which explicitly represent contours or surfaces.

1.2.1 Priors

Each element s_i of the source vector $\mathbf{s}$ is assumed to be drawn from a prior distribution which is sparse. There is some latitude regarding the exact form of the prior. Ideally, the chosen prior should match the actual distribution of sources, assuming the model accurately describes the process by which the signals were actually generated. In practice, different priors having a variety of sparse forms have been shown to provide similar results in terms of the learned basis functions for natural images. The intuition behind choosing a sparse distribution

for the sources is that it favors sources which have simple descriptions, in terms of having fewer significant valued components. Thus, it embraces Occam's razor, which states that simpler descriptions are preferable to ones that are unnecessarily complex. Sparse distributions, having lower entropy, can also be compressed into fewer bits. Thus the sparse prior can be thought of as favoring minimal length descriptions of the data.

The form of the prior distribution is assumed to be factorial and sparse:

$$P(\mathbf{s}) = \prod P(s_i) \tag{1.6}$$

$$P(s_i) = \frac{1}{Z_S} e^{-S(s_i)} \tag{1.7}$$

where Z_S is a normalizing constant referred to as the partition function, and S is a function that shapes $P(s_i)$ to have the requisite sparse form. Here a *sparse* distribution is loosely defined to mean one that is peaked at zero with heavy tails, or has positive kurtosis. One possibility is to choose a Cauchy distribution, where

$$S(s_i) = \log(1 + (s_i/\sigma)^2). \tag{1.8}$$

This choice for S, being smooth and non-convex, has a certain advantage in that it allows for gradient descent solutions when seeking to maximize the posterior distribution over the source components $P(\mathbf{s}|\mathbf{x})$. However, it does not correspond well to known sparse distributions over wavelet coefficients which are better characterized by Laplacian or Generalized Laplacian distributions that are more sharply peaked at zero. Additionally, if the representation is highly overcomplete, having many more source components than signal components, a more optimal choice for the prior would assign a higher probability for coefficients having exact zero values. Ideally, one would like to impose a prior which has a fixed cost for nonzero coefficients, and is otherwise relatively ambivalent about the

magnitude of nonzero values. Chapter 2 defines a mixture prior consisting of a combination of a Delta function and a Gaussian, and describes how to adapt the parameters of this prior in the context of a sparse model of natural images. This Delta-plus-Gaussian mixture prior supports the intuitive notion that features are either present or not present in an image.

1.2.2 Inference

Given a signal $\mathbf{x}$, and a model, defined by parameters θ, source coefficients s_i need to be inferred. If the representation is overcomplete, or if noise has been added to the signal, there will be more than one set of source coefficients that could have been used to generate the data under the model. The coefficients s_i should provide good reconstruction of the signal subject to the variance of the noise believed present in the signal, while being as probable as possible given the prior distribution over the coefficients. Bayesian inference provides a principled method for balancing these two constraints in order to select coefficients which are most likely for a given signal.

According to Bayes' Rule, the posterior distribution for a set of coefficients $\mathbf{s}$ is defined as

$$P(\mathbf{s}|\mathbf{x},\theta) = \frac{P(\mathbf{x}|\mathbf{s},\theta)P(\mathbf{s}|\theta)}{P(\mathbf{x})} \tag{1.9}$$

To infer the most likely causes of a signal under the model, we choose a coefficient vector $\hat{\mathbf{s}}$ that maximizes the posterior, otherwise known as the maximum *a posteriori*, or MAP estimate:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x},\theta)$$

$$= \arg\max_{\mathbf{s}} P(\mathbf{x}|\mathbf{s},\theta)P(\mathbf{s}|\theta) \tag{1.10}$$

$$= \arg\min_{\mathbf{s}} \left[\frac{\lambda_{\mathbf{n}}}{2}|\mathbf{x}-\mathbf{A}\mathbf{s}|^2 + \sum_i S(s_i)\right] \tag{1.11}$$

where $S(s_i) = -\log P(s_i)$, as defined in equation 1.7.

Gradient descent

A local minimum may be found via gradient descent, provided the gradient of the log of the prior is continuous and non-convex, which is the case for the Cauchy prior (see eqs. 1.7,1.8). The problem is cast as an energy minimization problem, where the energy E is defined as the negative log posterior (equation 1.11):

$$E = \frac{\lambda_{\mathbf{n}}}{2}|\mathbf{x}-\mathbf{A}\mathbf{s}|^2 + \sum_i S(s_i) \tag{1.12}$$

Partial differentiation with respect to E yields the differential equation

$$\dot{\mathbf{s}} \propto \frac{\partial E}{\partial \mathbf{s}}$$

$$= \lambda_{\mathbf{n}}\mathbf{A}^T\,\mathbf{e} - S'(\mathbf{s}) \tag{1.13}$$

$$\mathbf{e} = \mathbf{x} - \mathbf{A}\,\mathbf{s}\,. \tag{1.14}$$

There are a few problems that result from this approach. Gradient descent methods may not provide a global optimum if the solution space has multiple local minima. Additionally, the form of the prior may be unsuitable for inferring coeffients that are highly sparse due to having an overcomplete representation. In this situation, many coefficients should have exact zero values, due to the redundancy in the representation. However, the Cauchy prior will not appropriately penalize nonzero coefficients, resulting in many coefficients with values near zero but not exactly zero. This can result in reduced efficiency for coding applications.

Gibbs sampling

Gradient descent methods for inference may be unstable if the log of the prior is discontinuous or non-convex, which is generally the case for mixture priors or priors having sharp peaks at zero. Convergence problems may also exist due to local minima in the posterior caused by the overcompeteness of the representation. One alternative is to sample from the posterior using a Monte Carlo approach. A Gibbs sampling method is presented in Chapter 2 which simulates a Markov process in order to effectively sample from the Delta-plus-Gaussian prior previously mentioned. The Markov process is constructed in order to have a stationary distribution which matches the desired posterior distribution. In theory, the process must run indefinitely in order to obtain an unbiased sample. In practice, however, a reasonable approximation can often be obtained after relatively few iterations. If a MAP estimate is preferred, for signal estimation or coding applications, the sampling method can be combined with a simulated annealing approach in which the posterior distribution is raised to a power $1/T$. T is generally referred to as the *temperature* when considering the analogy to convergence brought about by cooling in physical systems. To obtain an estimate located near the maximum of the posterior, T is lowered gradually towards zero during the iterative process. By this means, the final sample can be be biased towards the most likely solution while avoiding local maxima solutions that might be obtained if T were lowered too quickly. In some cases, the mean of the posterior may be preferred. For example, this minimizes the mean squared error of the source estimate: $|s - \hat{s}|^2$, which may be desirable for denoising with a Gaussian noise model. In this case, an estimate of the posterior mean can be obtained by averaging a number of samples take from the posterior using Gibbs sampling.

A Gibbs sampling method for a mixture-of-Gaussians prior was presented in [25] for adapting a basis function matrix for 8x8 image patches. This distribution models each source coefficient as belonging to one of two states, either an inactive state or an active state. The state of a coefficient is determined by a binary state variable. A Gaussian distribution with large variance and zero mean is used to model the values of coefficients in the active state, and a Gaussian with relatively small variance and zero mean is used to model inactive coefficients. Using this model, Gibbs sampling is performed by iterating over the state variables, flipping the state variables stochastically according to probabilities derived from the posterior. For each flip, new values for the coefficients are computed very quickly by use of the an inverse Hessian matrix. While this method has been shown to be successful for adapting basis functions for small image patches, the method is not scalable to larger images due to the complexity of storing a large Hessian matrix which cannot be compactly represented. In Chapter 2, a method is presented that overcomes these restrictions in order to perform Gibbs sampling with a mixture model in a way that is easily scaled to images of any size, without requiring storage for a Hessian matrix.

Matching pursuit

One problem with Gibbs sampling methods is that they require many iterations in order to exact a single sample, and thus may be computationally prohibitive. As with gradient descent methods, they are also subject to some problems with local minima, so that in some cases an unbiased sample from the desired distribution may be unobtainable within a reasonable amount of time. Chapter 3 presents an alternative method of inference in the context of 1D audio signals known as *matching pursuit*. Matching pursuit is a greedy method

introduced by Mallat and Zhang for decomposing a signal into a linear set of waveforms selected from a overcomplete dictionary of functions [22]. Initially, the coefficients s_i are set to zero, and the optimization repeats this procedure by selecting the basis function at each iteration whose absolute inner product with the remaining residual $(\mathbf{x}-\mathbf{A}\,\mathbf{s})$ is maximal. The coefficient corresponding to that basis function is adjusted to maximally reduce the squared residual using that basis function, and the optimization proceeds in this manner until a tolerance value is reached.

While matching pursuit may be faster than Gibbs sampling, the standard method proposed by Mallat still has considerable complexity. For a Gabor dictionary of size $N \log N$, each iteration requires $\mathcal{O}(N \log N)$ computations, since at each iteration the inner products for each function are compared in order to find the one with maximum magnitude. Assuming that the number of coefficients needed to accurately represent the signal is proportional to the dimensionality of the signal, the total cost of the decomposition is thus $\mathcal{O}(N^2 \log N)$. In Chapter 3, an implementation of matching pursuit is presented for a class of dictionaries generated from wavelet filter banks that has a reduced total complexity of only $\mathcal{O}(N \log N)$ by making use of the special structure of the wavelet dictionaries, and by performing separate optimizations for each multi-resolution scale.

Besides matching pursuit, several other non-linear methods for selecting signal representations for overcomplete wavelet dictionaries have been proposed. A similar algorithm was proposed for Gabor dictionaries by Qian and Chen [28]. The *basis pursuit* method of Chen and Donoho selects the representation that minimizes the ℓ^1 vector norm $\sum |s_i|$ of the coefficients [29]. Although the basis pursuit method is not presented in a probabilistic framework, the ℓ^1 objective can be seen as a MAP estimator for Laplacian distributed sources. However,

this assumed model does not account for noise in the signal, and may not always obtain representations that are suitably sparse for highly overcomplete representations. In some situations, minimizing the ℓ^1 norm also provides the optimally sparse solution in terms of maximizing the number of coefficients with absolute zero values. However, this is not guaranteed to be the case in general. Other methods for obtaining decompositions with overcomplete dictionaries include the *method of frames* [12], and the *best orthogonal basis* method [8].

None of these methods, including matching pursuit, make use of an explicit probabilistic model, and thus do not provide a means of adapting the basis, considering the effect of noise in the input signal, or making use of priors. Also, some of these methods can only be used with specific dictionaries, or place restrictions such as orthogonality on the selected representation. The matching pursuit algorithm presented in chapter 3 is based on the one presented by Mallat and Zhang, but corrects these problem by means of an explicit statistical model. In this context, we show that the matching pursuit algorithm can be seen as an approximation to a MAP estimate when the prior over the coefficients assigns a high probability for coefficients with absolute zero values but is relatively ambivalent about the magnitudes of nonzero coefficients, such as a mixture distribution consisting of a delta function and a uniform distribution. Matching pursuit can be applied to any dictionary, making it an ideal choice when the dictionary is to be fit to the data.

1.2.3 Learning

A primary advantage of this statistical framework is that it provides a principled method for adapting bases to be optimal for describing a particular class of

signals. The model parameters θ, including the basis functions $\mathbf{A}$ and parameters which determine the form of the prior, can be adapted by maximizing the average log likelihood of the model for a given set of signals:

$$\mathcal{L} = \langle \log P(\mathbf{x}|\theta) \rangle \tag{1.15}$$

It can be shown that by maximizing $\mathcal{L}$, the KL divergence $D_{KL}(p||q)$ between the true density p and the model density $q = P(\mathbf{x}|\theta)$ is also minimized, thus lowering the bound on the average description length for signals under the model, since

$$\begin{aligned} \hat{H}(q) &= -\sum_x p(x) \log q(x) & (1.16) \\ &= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} & (1.17) \\ &= D_{KL}(p||q) + H(p) & (1.18) \end{aligned}$$

where $\hat{H}(q)$ is a lower bound on the average description length of signals drawn from the true distribution p and encoded using the model q [21].

The model distribution $P(\mathbf{x}|\theta)$ can be obtained by marginalizing over the internal states $\mathbf{s}$:

$$P(\mathbf{x}|\theta) = \int P(\mathbf{x}|\mathbf{s}, \theta)\, P(\mathbf{s}|\theta)\, d\mathbf{s} \tag{1.19}$$

Update rules for the model parameters can be obtained by differentiation with respect to $\mathcal{L}$. For example, the basis functions may adapted by the following update rule, as described in [21]:

$$\begin{aligned} \Delta\mathbf{A} &\propto \frac{\partial \mathcal{L}}{\partial \mathbf{A}} \\ &= \lambda_{\mathbf{n}} \left\langle \lambda_{\mathbf{n}} \int \mathbf{e}^T\, \mathbf{s} P(\mathbf{s}|\mathbf{x}, \theta)\, d\mathbf{s} \right\rangle . & (1.20) \end{aligned}$$

where $\hat{\mathbf{e}} = \mathbf{x} - \mathbf{A}\,\hat{\mathbf{s}}$.

The integral in equation 1.20 may be estimated by averaging the quantity $\mathbf{e}^T \mathbf{s}$ while sampling from the posterior $P(\mathbf{s}|\mathbf{x}, \theta)$, or approximated with a MAP estimate as described above. The MAP estimation has been shown to be suitable in some cases for adapting the basis functions [24], but requires some corrective steps to normalize the basis functions. This normalization is necessary due to a consistent bias towards trivially sparse representations (all $s_i = 0$) when using the MAP estimate [24].

1.2.4 Relation to other methods

The relationship of sparse coding to other methods of analysis can be seen by comparing them in terms of their assumed generative models. Principal Components Analysis (PCA), Independent Components Analysis (PCA), Factor Analysis (FA), and Sparse Coding (SC) all model data in terms of linear components. Figure 1.1 illustrates the relationships between the various methods. Both PCA and ICA make the assumption that there is no noise added to the signal ($\mathbf{n} = 0$), and that the mixing matrix $\mathbf{A}$ is square ($N = M$). PCA models the sources as Gaussian, thus fitting a multidimensional Gaussian model to the data. The standard ICA approach, on the other hand, can be viewed as using a generative model with sparse sources, which are assumed to be statistically independent. Unlike PCA, therefore, ICA can be used to find source components which are non-orthogonal, provided that the marginal distributions over the actual sources are sufficiently sparse.

Sparse Coding (SC) and Factor Analysis (FA) both consider the case where noise has been added to the signal, and generally model this noise using as Gaussian distributed. Factor Analysis, like PCA, models the sources as Gaussian,

while Sparse Coding, like ICA, models the sources as sparse and statistically independent. Unlike ICA, however, the sparse coding method can be used to model more sources than signal dimensions, $(M > N)$, and noise which may have been added to the signal.

1.3 Wavelets

Current image and signal processing methods often employ wavelet decompositions in order to obtain multi-scale representations which are amenable to further processing. A wavelet decomposition, illustrated here in one dimension, represents a function f in terms of a set of waveforms that are generated from a *mother wavelet* function $\psi(t)$ by changes in scale and translation:

$$f(t) = \int_0^\infty \int_{-\infty}^\infty g(u,s) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2} \tag{1.21}$$

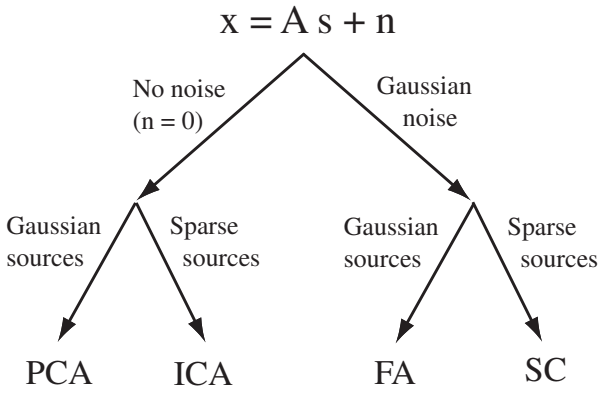


Figure 1.1. Relationship of Principal Components Analysis (PCA), Independent Components Analysis (ICA), Factor Analysis (FA), and Sparse Coding (SC) in terms of their assumed generative models.

where $g(u, s)$ is the *wavelet transform* of a signal f at the scale s and position u, and can be computed by correlating f with the generated wavelets:

$$g(u, s) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi \left(\frac{t - u}{s} \right) dt. \tag{1.22}$$

The term *wavelet* is generally taken to mean that the generator ψ is well localized and oscillating, hence a small wave. This may be more rigorously defined in terms of some number n of *vanishing moments* for which:

$$\int_{-\infty}^{\infty} t^k \, \psi(t) dt = 0 \quad \text{for} \quad 0 \leq k < n. \tag{1.23}$$

For practical purposes, wavelet transforms are similarly defined over discrete sequences. Invertible filter banks provide an efficient means of computing these transforms by recursively filtering and subsampling the signal.

When applied to images and many other types of signals, wavelet representations have been shown to have sparse distributions which are well-suited for data compression. When the wavelet functions are well matched to the structure of the signal, thresholding the wavelet transform and inverting is an effective way to reduce noise present in the signal, a technique known as "coring" or "wavelet shrinkage". Because the wavelet functions smoothly overlap, thresholding the wavelet coefficients does not produce "blocking" artifacts common to methods which segment the data and process each block independently. Wavelets provide a powerful and elegant way to describe data which is well characterized by a linear combination of events which are well localized in both frequency and time (or space) and which are self-similar across position and scale.

Despite these advantages to wavelets, certain questions regarding the use and design of wavelets remain unanswered. For example, it is not clear how to choose the optimal wavelet for a particular application. Countless families of wavelet

decompositions have been constructed to satisfy certain mathematical properties, but their effectiveness for specific image classes has been left to empirical testing. Most of these have been developed for 1D signals and are not best suited for images or signals of higher dimensionality. Of the directional wavelets that have been designed for 2D, which are most suited for describing the general class of natural scenes, or to specific sub-classes of natural scenes? Can optimality be determined? Overcomplete representations, which have more wavelet coefficients in the representation than image pixels, have been shown to be more effective at denoising in terms of mean squared error and introduce fewer noticable artifacts. However, how can one threshold optimally in a non-orthogonal representation? How can one apply wavelets to solve inverse problems such as deconvolution?

In this dissertation, I show that these questions can be addressed within the statistical framework described above, which is described in detail in chapters 2 and 3 and applied to images and audio signals. Instead of considering a wavelet as a feedforward transform, the wavelet bases are taken to be part of a linear generative model. By assuming a sparse prior over the wavelet coefficients, the wavelet generator functions (filter banks) may be adapted by matching the image model to the statistical structure of a given dataset. Thus, an optimal discrete wavelet basis or overcomplete dictionary can be determined for a given class of data, within the assumed wavelet self-similarity and sampling lattice constraints.

This statistical framework also provides an intuitive understanding for why thresholding a wavelet representation is effective for denoising, since with a Gaussian noise model and an orthogonal basis the mean posterior estimate is equivalent to a soft-threshold or coring operation[30]. Thus, thresholding may be viewed as a type of inference. Similarly, even if the wavelet functions are non-orthogonal, the generative modeling framework still allows us to pose denoising and other

problems as ones of inference. While exact solutions may not always be efficiently obtained, this provides a principled approach for solving them which may lead to reasonable approximations.

1.4 Information Hiding

For some problems, it is less clear how to apply statistical models for optimal solutions. One interesting area involves the hiding or detection of information within certain types of media. Information hiding generally falls under two main categories: *digital watermarking*, and *steganography*. Digital watermarks are generally used for authentication or data rights management in order to protect copyrighted material. Steganography is used for hidden communication, and has the goal of communication without detection. *Steganalysis* is the art or science of detecting the presence of hidden messages. The data in which information is hidden is referred to as the *cover*.

Digital watermarking and steganography have different objectives. Figure 1.2 shows a pictoral representation of the relationship between four competing objectives common to information hiding problems: *undetectability*, *capacity*, *robustness*, and *quality*. *Capacity* refers to the amount of information that can be hidden in a given sized cover, *robustness* to how difficult it is to remove the information without rendering the cover unusable, *undetectability* to how hard it is to detect the presence of the hidden information, and *quality* to how little distortion is incurred in the cover data. These objectives form the corners of a tetrahedron, with possible trade-off points existing at all points within its volume. In steganography, the goal is to maximize capacity while avoiding detection. For digital watermarking applications, the goal is usually to hide information so that

it is as robust as possible while avoiding adding noticable distortion to the cover.

If the distribution of *stego objects*, signals containing embedded messages, is measurably different than the distribution over non-stego objects, detection is possible. This forms the basis for an information theoretic definition of steganographic security proposed by Cachin [5] which defines the uses the KL divergence between the two distributions as a security measure. If the KL divergence between the distributions over stego and non-stego objects is zero, the method is perfectly secure. Steganography methods usually embed messages in the least significant bits of the coefficients used to represent the cover message, or by incrementing or decrementing the coefficients in small amounts. Arbitrary embedding methods are likely to alter the statistical properties of the cover, making it possible for an attacker to detect the hidden message. Only by carefully modeling the statistical properties of the cover media, and by making sure these statistics are maintained

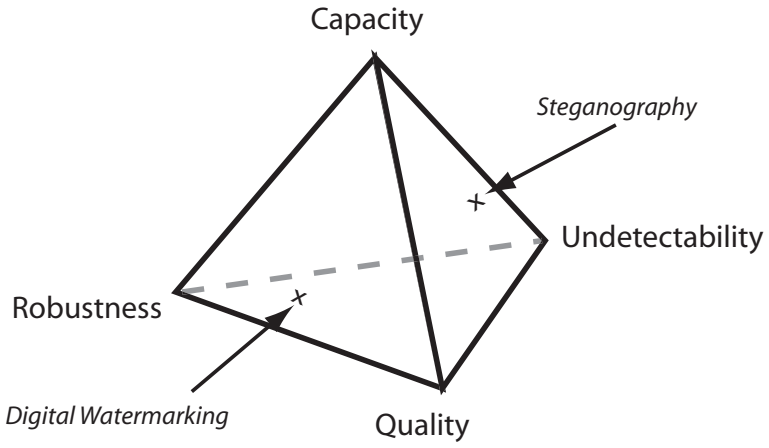


Figure 1.2. A pictoral representation of information hiding problems. Points of the tetrahedron represent basic competing objectives, forming a volume of possible trade-off points in which steganography and digital watermarking exist as points on different faces of the tetrahedron.

by the embedding method, can one ensure that certain features cannot be used to detect the hidden messages.

1.5 Outline of Dissertation

This dissertation is organized as follows. Chapter 2 describes an overcomplete wavelet image model using a Delta-plus-Gaussian prior, and demonstrates a Gibbs sampling method for sampling from the resulting posterior. The wavelet model is adapted to a set of natural images, and the model is tested to verify how it performs in terms of sparsity of the representations, and in terms of mean square error for denoising applications. Chapter 3 describes a wavelet framework which can be applied to audio and other 1D signals, and introduces a fast matching pursuit algorithm within the context of an overcomplete wavelet dictionary. The resulting wavelet basis adapted using matching pursuit on a database of nature recordings is shown. Wavelet functions are learned for several resolutions, showing that the sounds are not entirely self-similar across scale. A set of shiftable functions are learned for multichannel EEG (electroencephalograph) data, using matching pursuit with a non-negative constraint applied to the coefficients. These functions are shifted, but not rescaled, to comprise the basis set. The resulting functions capture dependencies across time as well as across channels, unlike previous approaches for applying ICA to EEG which only capture dependencies across channels.

The inference methods described in Chapters 2 and 3 can be applied to a wide variety of inverse problems with only minor alteration to the generative model. Not all data analysis problems can be posed in terms of inference with a statistical model, however. Chapter 4 describes a methodology for applying statistical mod-

els to a different class of problems in the area of information hiding. A statistical approach is presented for performing steganography and steganalysis, showing how a statistical model of the cover media can be used to hide information, or detect whether information is hidden. The proposed steganography method ensures perfect security in the Cachin sense, to the degree that the assumed model is correct, while maximizing the amount of information that can be hidden. Due to the related nature of the problems, the framework may also have important implications for digital watermarking. Methods are demonstrated for performing steganography and steganalysis in JPEG images using this model-based approach and are shown to outperform existing methods. In Chapter 5, the contributions presented are briefly summarized and implications and conclusions based on these results are discussed.

Chapter 2

Adapting Wavelets to Natural Images

This chapter is focused on characterizing statistical patterns found in so-called "natural" images, i.e. photographic scenes taken from the real world. Here, the term "natural" is used only to distinguish between photographic images and those images which are generated by human or machine, such as images of text or of graphical designs. We do not make the distinction here between photographs of man-made versus natural objects, since it is expected that the statistical properties measured here will be common to both. That said, the experiments in this chapter use images of trees, landscapes, and wildlife and contain very few man-made structures.

Considering the enormous variety of natural scenes, it may seem surprising that they have any predictable structure whatsoever. Photographic images can contain practically anything, it seems, so how could one predict anything about them? If one examines a number of these images closely, however, it is evident

that there are indeed predictable qualities, or features, to be found even within such a large class of data. For example, one may notice that adjacent pixels in an image often have very similar values. This can be quantified mathematically by measuring the correlation between pairs of pixels across an image, for a given relative distance, revealing a high degree of correlation for pairs of pixels positioned close to each other. Additional inspection may reveal that changes in image intensity frequently occur along oriented edges.

These initial observations represent only a small fraction of the structure that is present in images. If one generates artificial images containing only these statistical qualities, they still appear much different from recognizable scenes. Not only do natural images contain predictable features, they are laden with these predictable properties. In fact, out of all of the possible images one could create, only a very few have the many special properties that would cause it to be recognized as a real-world image. Imagine creating images at random by choosing random pixel values one at a time. Even for very small images, one could spend a lifetime repeating the process without ever generating a recognizable scene. This demonstrates that there is far less that is unpredictable about an image than one might at first realize.

The challenge is to characterize, or model, these predictable qualities. By modeling the statistical regularities present in images, we can apply this knowledge to practically any image processing application. For example, in order to compress an image into as few bits as possible without significant perceptible loss of quality, it is necessary to reduce the amount of redundant information being stored. Information that can be predicted by the model does not need to be transmitted, since it can be regenerated using the same model when the image is decoded. Statistical models can also be used to improve the quality of an image,

by identifying irregularities that are likely due to measurement errors, or noise. The ability to separate noise, or various forms of distortion, from the original image is a key step in many types of image analysis. Identifying components in the data, such as edges and textures, can allow for an image to be segmented into parts and represents a first step in some object recognition schemes. Even the problem of hiding or detecting hidden information in images, such as watermarks or secret communication, can be solved using an appropriate image model. This problem is discussed in detail in chapter 4.

One can model certain types of statistical regularities by selecting an appropriate representation for the data. While the standard pixel representation is a necessary format for viewing an image properly, it is poorly suited for compression and most image processing applications. This is because it does not make explicit the features, or frequently occurring patterns, which are present in the image. Thus, if pixels are changed independently from each other, they will produce noisy artifacts, or if they are encoded independently from each other, dependencies between coefficients will not be taken into account for maximum compression. The ideal representation for a class of data describes the data in terms of a statistically independent set of features, rather than pixels. Complete statistical independence is not practically achievable, however. If it were, all of the statistical structure contained in images would be accounted for in the representation, making perfect compression and image recognition possible. In practice, the goal is to select a representation which makes explicit certain components of the data which are as independent as possible.

This thesis is concerned primarily with image representations which are linear, meaning that they represent images in terms of a linear superposition of functions which we will refer to as *basis functions*. Describing images in terms

of linear components is a vast over-simplification and can model only a small set of the statistical properties of images. However, linear models form a starting point which is just simple enough for exploring the generative modeling framework, and have already been shown to be of use in characterizing some kinds of image structure. For instance, linear representations such as the discrete cosine transform (DCT) and discrete wavelet transforms (DWT) are used for most image compression standards today, including the JPEG (Joint Picture Experts Group) and the newer JPEG2000 standards. The set of basis functions which form the representation is referred to as the *basis*. For sake of simplicity and lack of more accepted terminology, the terms *basis* and *basis function* are used whether or not the representation is orthogonal, and thus may not be strictly considered a *basis*.

In the case of the Fourier or DCT basis, for example, the basis functions are sine waves of varying frequency. While Fourier components are decorrelated for natural scenes, the basis functions (continuous sine waves) do not account well for the localized and oriented structure present in images. In order to describe a feature which is localized in an image, such as an edge, many Fourier coefficients having large amplitudes are required. For this reason, wavelets, having basis functions which are localized in position and frequency, have been shown to be very useful for describing images using only a few "active" basis functions (having large amplitude coefficients). Additionally, wavelet functions are self-similar, and scaled so that there are the same number of each type of function at each spatial scale. This type of *multi-scale* representation assumes a form of scale invariance of image structure which is well suited for images in which objects may be arbitrary distances from the camera. In reality, however, the statistics of natural images are only approximately scale invariant. Systematic differences have been shown

to exist across scales for most images, which can be used to differentiate between broad classes of images such as indoor and outdoor scenes. Such differences may be due to biases imposed by the type of environment in which we live and the limited viewpoints of human observers [34].

There has been increasing interest in the use of overcomplete image representations, or *dictionaries*, where the number of basis functions exceeds the number of image pixels. Overcomplete dictionaries, composed of many different kinds of time-frequency atoms, have been found to provide useful descriptions which can be more closely related to the original causes of many signals [22]. For images, overcompleteness has been shown to allow for more stable representations, where small shifts or rotations to an image do not affect drastic changes in the coefficients of the representation. In critically sampled multi-scale representations, this type of instability is unavoidable due to subsampling of the high-frequency components [32, 15].

Overcomplete image representations may thus provide more meaningful representations in the sense that changes made to an image that seem small conceptually (as perceived by a human) are linked to similar changes to the coefficients. Ideally, image features (such as edges) should be well described by only a few coefficients, regardless of where they are located in the image, how they are rotated, or how large they are. Such a representation may translate into gains in coding efficiency for image compression, and improved accuracy for tasks such as denoising. For example, overcomplete representations have been shown to reduce Gibbs-like ringing artifacts common to thresholding methods employing critically sampled wavelets [9, 6, 30].

Previously, overcomplete dictionaries for images have been constructed by hand by combining multiple orthogonal bases, as in the wavepacket dictionary

of Coifman *et al.* [8], or by other mathematical construction, as in the infinite and finite Gabor dictionaries used by Mallat *et. al* [22]. Rather than composing wavelet dictionaries for audio signals through an arbitrary process, however, we consider in this chapter how to apply the sparse coding framework to learn overcomplete wavelet dictionaries which are adapted to a collection of images. The goal of sparse coding is to identify components which are as statistically independent as possible, characterized by coefficients having *sparse* distributions (peaked at zero with heavy tails). The sparse coding algorithm provides a means for identifying features present in the data in an unsupervised fashion, thus optimizing the basis description for a given class of data. Previously, sparse coding algorithms have been applied only to small image patches (around 16x16 pixels) because the algorithms do not scale well to larger images [24, 25]. By imposing wavelet constraints of self-similarity across position and scale, the algorithm can be applied to larger images requiring only a relatively small set of parameters to be learned.

Common wavelet denoising approaches generally apply either a hard or soft-thresholding function to coefficients which have been obtained by filtering an image with a the basis functions. One can view these thresholding methods as a means of selecting coefficients for an image based on an assumed sparse prior on the coefficients [2, 7]. This statistical framework provides a principled means of selecting an appropriate thresholding function. When such thresholding methods are applied to overcomplete representations, however, problems arise due to the dependencies between coefficients. Choosing optimal thresholds for a non-orthogonal basis is still an unsolved problem. In one approach, orthogonal subgroups of an overcomplete shift-invariant expansion are thresholded separately and then the results are combined by averaging [9, 6]. In addition, if the coeffi-

cients are obtained by filtering the noisy image, there will be correlations in the noise that should be taken into account.

Here I address two major issues regarding the design and use of overcomplete representations for images. First, what is the optimal basis to use for a specific class of data? To help answer this question, a method is presented for adapting an overcomplete wavelet basis, or dictionary, to the statistics of natural images. Second, given an overcomplete set of basis fundtions, how should the coefficients for representing an image be computed? Problems associated with thresholding are avoided by using the wavelet basis as part of a generative model, rather than a simple filtering mechanism. Coefficients are sampled from the resulting posterior distribution by simulating a Markov process known as a Gibbs-sampler.

To obtain image representations which are sparse, our model imposes a prior distribution over the wavelet coefficients which is composed of a mixture of a Gaussian and a Dirac delta function, so that inactive coefficients are encouraged to have exact zero values. Similar models employing a mixture of two Gaussians have been used for classifying wavelet coefficients into active (high variance) and inactive (low variance) states [7, 11]. Such a classification should be even more advantageous if the basis is overcomplete. A method for performing Gibbs-sampling for the Delta-plus-Gaussian prior in the context of an image pyramid is derived, and demonstrated to be effective at obtaining very sparse representations which match the form of the imposed prior. Biases in the learning are overcome by sampling instead of using a MAP estimate.

2.1 Wavelet image model

Each observed image $\mathbf{I}$ is assumed to be generated by a linear superposition of basis functions which are columns of an N by M weight matrix $\mathbf{A}$, with the addition of Gaussian noise $\mathbf{n}$:

$$\mathbf{I} = \mathbf{A}\,\mathbf{s} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{I}$ is an N-element vector of image pixels and $\mathbf{s}$ is an M-element vector of basis coefficients. The probability of generating an image $\mathbf{I}$, given coefficients $\mathbf{s}$, parameters θ, assuming Gaussian i.i.d. noise $\mathbf{n}$ (with variance $1/\lambda_{\mathbf{n}}$), is

$$P(\mathbf{I}|\mathbf{s},\theta) \;=\; \frac{1}{Z_{\lambda_{\mathbf{n}}}} e^{-\frac{\lambda_{\mathbf{n}}}{2}|\mathbf{I}-\mathbf{A}\,\mathbf{s}|^2}. \tag{2.2}$$

In order to achieve a practical implementation which can be seamlessly scaled to any size image, it is assumed that the basis function matrix $\mathbf{A}$ is composed of a small set of spatially localized *mother wavelet* functions $\psi_i(x,y)$, $i = 1..B$, which are shifted in position (x,y) in the image and rescaled by factors of two. Unlike typical wavelet transforms which use a single 1-D mother wavelet function to generate 2-D functions by inner product, the functions $\psi_i(x,y)$ are not constrained to be 1-D separable. Moreover, any one $\psi_i(x,y)$ does not give rise to a complete basis, but the set as a whole does.

The functions $\psi_i(x,y)$ provide an efficient way to perform computations involving $\mathbf{A}$ by means of convolutions. Basis functions of coarser scales are produced by upsampling the $\psi_i(x,y)$ functions and blurring with a low-pass filter $\phi(x,y)$, also known as the *scaling function*. The image model in equation 2.1 may be re-expressed to make these parameters explicit:

$$I(x,y) \;=\; g^0(x,y) + n(x,y) \tag{2.3}$$

$$g^l(x,y) \;=\; \begin{cases} \left[g^{l+1}(x,y) \uparrow 2\right] * \phi(x,y) + \sum_i s_i^l(x,y) * \psi_i(x,y) & l < L-1 \\ s^l(x,y) & l = L-1 \end{cases} \tag{2.4}$$

where the coefficients $s_i^l(x,y)$ are indexed by their position (x,y), band (i) and level of resolution (l) within the pyramid ($l = 0$ is the highest resolution level). The symbol $*$ denotes convolution, and $\uparrow 2$ denotes upsampling by two and is defined as

$$f(x,y) \uparrow 2 \;\equiv\; \begin{cases} f(\frac{x}{2}, \frac{y}{2}) & x \,\text{even} \;\;\&\;\; y \,\text{even} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The wavelet pyramid model is schematically illustrated in figure 2.1. Traditional wavelet bases for images typically utilize three bands ($B = 3$), in which case the representation is *critically sampled* (same number of coefficients as image pixels) provided that the coefficients are subsampled by an additional factor of 2 starting at the lowest level ($l = 0$) of the pyramid. Here, the lowest level is not subsampled so as to avoid aliasing, so that even with only one band the representation is overcomplete. We examine the cases of $B = 2, 4$ and 6 in order to explore varying degrees of overcompleteness.

No restrictions are imposed on the form of the mother wavelet functions ψ_i, except that it is assumed the functions can be accurately represented by a specified number of sample points. The term *wavelet* is used here to refer only to constraints of self-similarity across position and scale (a dyadic sampling lattice is imposed), and the functions are also normalized to have zero mean. Because these functions are adapted to efficiently describe natural images (as defined by maximizing the sparsity of the representation), any further constraints should arise naturally from the data to the degree the constraints are justified for obtaining an efficient descriptions of the chosen dataset. We also do not impose the restriction that the learned basis functions are self-inverting. As the purpose

here is to adapt an overcomplete dictionary of functions, the self-inverting solution would not necessarily be the sparsest, or presumably the most meaningful, representation. If desired, however, the learning rules can be adjusted to add a self-inverting constraint.

Figure 2.2 shows a system diagram for the wavelet pyramid decomposition. For analysis, an image $I(x, y)$ is filtered into low-pass and high-pass subbands by correlation, denoted by $\star$, with the scaling function ϕ, and a high-pass filter ϕ_C. To ensure proper reconstruction, ϕ_C is designed in the frequency domain to be the complement of ϕ (when ϕ is applied twice). This is accomplished by selecting ϕ_C to have an amplitude spectrum equal to 1 minus the power spectrum of the scaling function ϕ, and is equivalent to subtracting the eventual contribution of the low-pass subband (after upsampling and convolving again with ϕ) from the

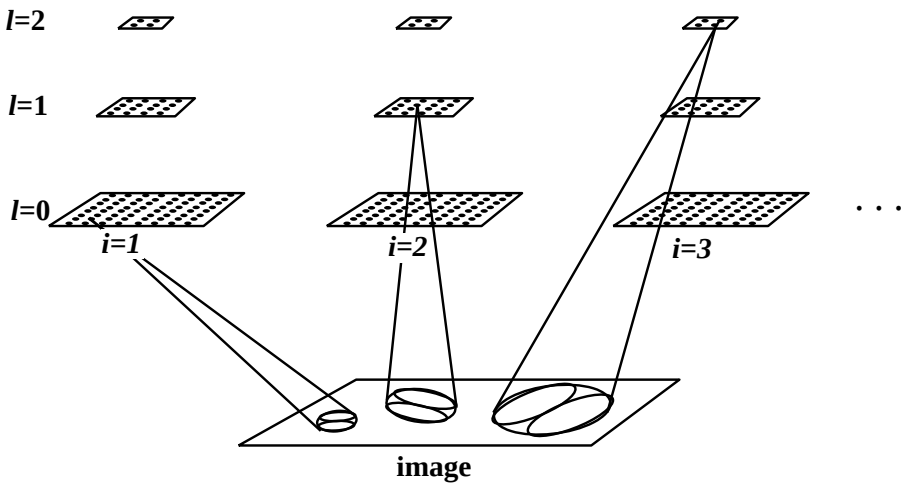


Figure 2.1. Wavelet image model. Shown are the coefficients of the first three levels of a pyramid ($l = 0, 1, 2$), with each level split into a number of different bands ($i = 1...B$). The highest level ($l = 3$) is not shown and contains only one low-pass band.

image. Gibbs sampling (G.S.) is performed on the high-pass subband to select the coefficients $s_i(x, y)$ for each wavelet function ψ_i. The low-pass band is subsampled by two in both x and y, denoted by $(\downarrow 2)$, and the decomposition is recursively applied at the next higher scale (except at the highest scale) by inserting the diagram into the location marked by a filled circle. A reconstructed image $\hat{I}(x, y)$ is obtained by convolving the coefficients with the wavelet functions and adding the result to the the reconstructed low-pass image from the next highest scale after upsampling by two and convolving with ϕ.

2.1.1 Delta-plus-Gaussian prior

The prior probability over each coefficient s_i is modeled as a mixture of a Gaussian distribution and a Dirac delta function $\delta(a_i)$. A binary state variable u_i for each coefficient indicates whether the coefficient s_i is *active* (any real value), or *inactive* (zero). The probability of a coefficient vector $\mathbf{s}$ given a binary state

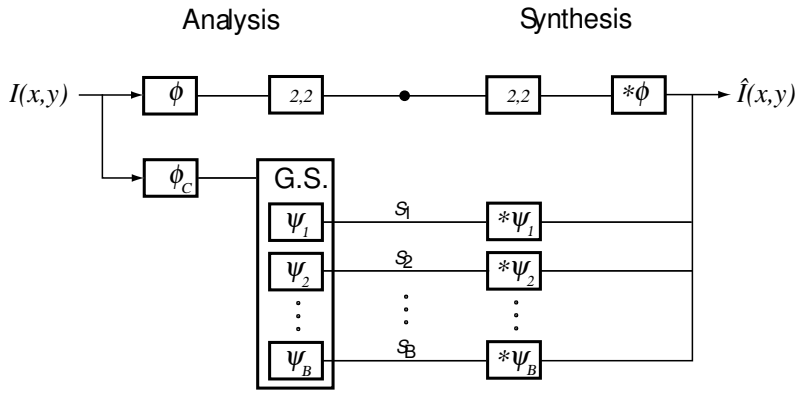


Figure 2.2. System diagram for wavelet pyramid decomposition.

vector $\mathbf{u}$ and model parameters $\theta = \{\mathbf{A}, \lambda_{\mathbf{n}}, \lambda_{\mathbf{s}}, \mathbf{\Lambda_u}\}$ is defined as

$$P(\mathbf{s}|\mathbf{u},\theta) = \prod_i P(s_i|u_i,\theta) \tag{2.6}$$

$$P(s_i|u_i,\theta) = \begin{cases} \delta(s_i) & \text{if} \quad u_i = 0, \\ \frac{1}{Z_{\lambda_{s_i}}} e^{-\frac{\lambda_{s_i}}{2} s_i^2} & \text{if} \quad u_i = 1 \end{cases} \tag{2.7}$$

where $\lambda_{\mathbf{s}}$ is a vector with elements λ_{s_i}. The probability of a binary state $\mathbf{u}$ is

$$P(\mathbf{u}|\theta) = \frac{1}{Z_{\mathbf{\Lambda_u}}} e^{-\frac{1}{2}\mathbf{u}^T \mathbf{\Lambda_u}\, \mathbf{u}}. \tag{2.8}$$

Matrix $\mathbf{\Lambda_u}$ is assumed to be diagonal (for now), with nonzero elements λ_{u_i}. The form of the prior is shown graphically in figure 2.3. Note that the parameters $\mathbf{A}$, $\lambda_{\mathbf{s}}$, and $\mathbf{\Lambda_u}$ are actually defined by a much smaller set of parameters. Since translation and scale invariance is assumed, to define these parameters only the mother wavelet function $\psi_i(x,y)$, and a single λ_{u_i} and λ_{s_i} parameter need to be specified for each wavelet band, along with the scaling function $\phi(x,y)$.

The total image probability is obtained by marginalizing over the possible coefficient and state values:

$$P(\mathbf{I}|\theta) = \sum_{\mathbf{u}} P(\mathbf{u}|\theta) \int P(\mathbf{I}|\mathbf{s},\theta) P(\mathbf{s}|\mathbf{u},\theta)\, d\mathbf{s} \tag{2.9}$$

2.2 Sampling and Inference

In order to select the coefficients for an image, a method is presented for sampling from the posterior distribution, $P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)$, for an image $\mathbf{I}$ using a Gibbs sampler. For each coefficient and state variable pair (s_i,u_i), we sample from the posterior distribution conditioned on the image and the remaining coefficients $s_{\bar{i}}$: $P(s_i, u_i|\mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta)$. After all coefficients (and state variables) have been updated, this process is repeated until the system has reached equilibrium. To

infer an optimal representation $\mathbf{s}$ for an image $\mathbf{I}$ (for coding or denoising purposes), one can either average a number of samples to estimate the posterior mean, or with minor adjustment locate a posterior maximum by raising the posterior distribution to a power $(1/T)$ and annealing T to zero. To sample from $P(s_i, u_i|I, s_{\bar{i}}, u_{\bar{i}}, \theta)$, we first draw a value for u_i from $P(u_i|\mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta)$, and then draw s_i from $P(s_i|u_i, \mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta)$.

For $P(u_i|\mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta)$ we have:

$$P(u_i|\mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta) \propto P(u_i|u_{\bar{i}}, \theta) \int P(\mathbf{I}|s_i, s_{\bar{i}}, \theta) P(s_i|u_i, \theta) ds_i \tag{2.10}$$

where

$$P(u_i|u_{\bar{i}}, \theta) = \frac{1}{Z_{u_i|u_{\bar{i}}}} e^{-\frac{\lambda_{u_i}}{2} u_i}, \tag{2.11}$$

$$P(\mathbf{I}|s_i, s_{\bar{i}}, \theta) = \frac{1}{Z_{\lambda_{n_i}}} e^{-\frac{\lambda_{n_i}}{2}(s_i - b_i)^2}, \tag{2.12}$$

and

$$\lambda_{n_i} = \lambda_{\mathbf{n}} |\mathbf{A}_i|^2, \qquad b_i = \frac{\mathbf{A}_i \cdot (\mathbf{I} - \mathbf{A}\, \mathbf{s}_{i=0})}{|\mathbf{A}_i|^2}. \tag{2.13}$$

The notation $\mathbf{A}_i$ denotes column i of matrix $\mathbf{A}$, $|\mathbf{A}_i|$ is the length of vector $\mathbf{A}_i$, and $\mathbf{s}_{i=0}$ denotes the current coefficient vector $\mathbf{s}$ except with s_i set to zero. Thus, b_i denotes the value for s_i which minimizes the reconstruction error (while holding $s_{\bar{i}}$ constant). Since u_i can only take on two values, one can compute equation 2.10 for $u_i = 0$ and $u_i = 1$, integrating over the possible coefficient values. This yields the following sigmoidal activation rule as a function of b_i:

$$P(u_i = 1|\mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta) = \frac{1}{1 + e^{-\beta_i(b_i^2 - t_i)}} \tag{2.14}$$

where

$$\beta_i = \frac{1}{2} \frac{\lambda_{n_i}^2}{\lambda_{n_i} + \lambda_{s_i}}, \tag{2.15}$$

$$t_i = \frac{\lambda_{n_i} + \lambda_{s_i}}{\lambda_{n_i}^2} \left[\lambda_{u_i} - \log \frac{\lambda_{s_i}}{\lambda_{n_i} + \lambda_{s_i}} \right]. \tag{2.16}$$

For $P(s_i|u_i, \mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta)$ we have:

$$P(s_i|u_i, \mathbf{I}, s_{\bar{i}}, u_{\bar{i}}, \theta) = \begin{cases} \delta(s_i), & u_i = 0, \\ \mathcal{N}(\frac{\lambda_{n_i} b_i}{\lambda_{n_i} + \lambda_{s_i}}, \sqrt{\frac{1}{\lambda_{n_i} + \lambda_{s_i}}}), & u_i = 1. \end{cases} \tag{2.17}$$

Figure 2.3 shows the form of the prior (dashed line), and a histogram of the coefficient values obtained by sampling from the posterior (solid line) for a single coefficient type for a set of natural images. Note that the histogram closely matches the prior, indicating that the model appears to be a reasonable fit to the data. Since the histogram was obtained by sampling from the posterior, instead of the prior, the data could have influenced the statistics away from the imposed prior.

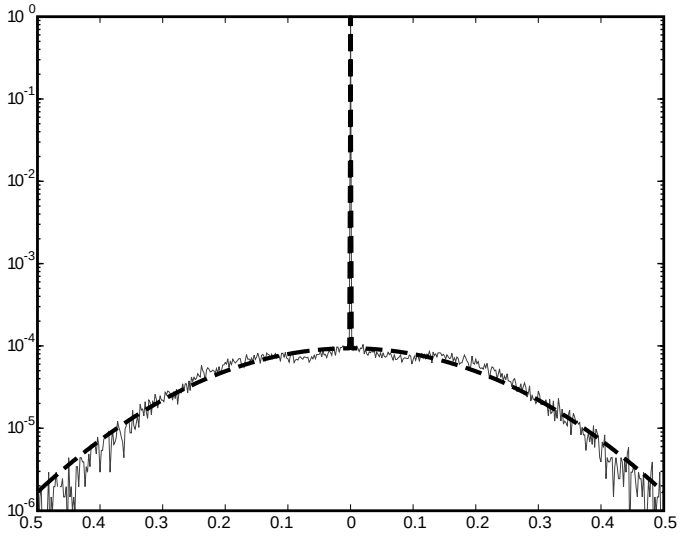


Figure 2.3. Prior distribution (dashed), and histogram of samples taken from the posterior (solid) for a single coefficient. The y-axis is plotted on a log scale.

2.3 Adapting the model to images

The objective for adapting the model is to adjust the parameters, θ, to maximize the average log-likelihood of the model for a set of images:

$$\hat{\theta} = \arg\max_{\theta} \langle \log P(\mathbf{I}|\theta) \rangle \tag{2.18}$$

The parameters are updated by gradient ascent on this objective, which results in the following update rules:

$$\Delta\lambda_{u_i} \propto \frac{1}{2}\left\langle\left\langle\left[\frac{1}{1+e^{\frac{1}{2}\lambda_{u_i}}} - u_i\right]\right\rangle_{P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)}\right\rangle \tag{2.19}$$

$$\Delta\lambda_{s_i} \propto \frac{1}{2}\left\langle\left\langle u_i\left[\frac{1}{\lambda_{s_i}} - s_i^2\right]\right\rangle_{P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)}\right\rangle \tag{2.20}$$

$$\Delta\lambda_{\mathbf{n}} \propto \frac{1}{2}\left\langle\left[\frac{N}{\lambda_{\mathbf{n}}} - |\mathbf{I} - \mathbf{A}\,\mathbf{s}|^2\right]\right\rangle_{P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)} \tag{2.21}$$

$$\Delta\psi_i(x,y) \propto \sum_{l=0}^{L-2} \lambda_{\mathbf{n}}\left\langle\left\langle e^l(x,y) \star s_i^l(x,y)\right\rangle_{P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)}\right\rangle, \tag{2.22}$$

where $\star$ denotes 2D cross correlation and

$$e^l(x,y) = \begin{cases} I(x,y) - \hat{I}(x,y) & l = 0 \\ \left[e^{l-1}(x,y) \star \phi\right] \downarrow 2 & 0 < l < L-1 \end{cases} \tag{2.23}$$

is the reconstruction error $\hat{\mathbf{I}} = \mathbf{I} - \mathbf{A}\,\mathbf{s}$ filtered up through the pyramid. It is only necessary to compute a center portion of each cross correlation having the same extent of the $\psi_i(x,y)$ functions. The outer brackets denote averaging over many images. The notation $\langle\rangle_{P()}$ denotes averaging the quantity in brackets while sampling from the specified distribution.

Similarly, the scaling function ϕ may also be learned, using the following

update rule:

$$\Delta\phi(x,y) \quad \propto \quad \sum_{l=0}^{L-2} \lambda_{\mathbf{n}} \left\langle \left\langle e^l(x,y) \star [g_i^{l+1}(x,y) \uparrow 2] \right\rangle_{P(\mathbf{s},\mathbf{u}|\mathbf{I},\theta)} \right\rangle. \tag{2.24}$$

In the following experiments, however, ϕ was not learned, but was designed by hand in the frequency domain to have a flat frequency response up to $\frac{1}{4}$ of the Nyquist rate, and smoothly taper off to zero (using a cosine function in log frequency) by $\frac{1}{2}$ Nyquist in order to prevent aliasing. Refer to [31, 32] for issues regarding the design and use of filters for pyramid subband transforms.

2.4 Results

2.4.1 One octave scaling

The image model was trained on 22 512x512 grayscale natural images (not whitened). These images were generated from color images taken from a larger database of photographic images [19]. Smaller images (64x64 pixels) were selected randomly for sampling during training. Assuming scale invariance, the wavelet functions ψ_i were adapted to fit a single spatial frequency band. Each image was initially bandpass filtered for an octave range using the scaling function ϕ, so that the learned functions would correctly fit within the pyramid framework and could be applied to any scale. The functions ψ_i were represented using 17 x 17 pixel masks, and were initialized to random values. The λ_{s_i} and λ_{u_i} parameters were constrained to be the same for all orientation bands and were adapted over many images with $\lambda_{\mathbf{n}}$ fixed at .05 (not learned), corresponding to a noise variance less than 1% of the image variance ($\sigma_{\mathbf{I}}^2 = 2.855e{+}03$).

Shown in figure 2.4 are the $\psi_i(x,y)$, with their corresponding 2D spectra when 2, 4, and 6 orientation bands were learned. The learned functions are well

localized in position and orientation, and appear to be rotated versions of each other, even though no self-similarity constraints between the functions were imposed. Increasing the number of bands B produces narrower orientation tuning. The learned functions appear very similar to the equivalent basis functions of the *steerable pyramid* [31]. The functions for a steerable pyramid constructed for 6 orientation bands and 1 octave scaling are shown in figure 2.5, alongside the learned 6-band functions. The rotational averages of their 2D spectra, showing power as a function of spatial frequency, are also shown for comparison. The similarity between the learned functions and the steerable functions is surprising, since the steerable basis functions were not designed with sparsity as an objective. Instead they were designed to be shiftable and steerable, meaning that a linear combination of the functions can be used to produce the same functions shifted and rotated to any position and angle [15]. By optimizing for sparsity, we obtain nearly the same result.

There are some differences between the steerable functions and the learned functions, however. Note that the learned functions have a power spectrum with a steeper descent than the steerable functions, indicating a possible deficiency in the steerable filters for efficiently representing natural scenes. The steerable basis functions are designed to have a flat power spectrum in order to be self-inverting. The self-inverting property may thus be a restriction that reduces the efficiency of the representation. In section 2.4.3, the efficiency of the learned basis functions are compared to those of the steerable pyramid in terms of the sparsity of the representations, showing that the learned basis functions allow for a slightly higher degree of sparsity than the steerable functions.

2.4.2 Two octave scaling

We wished to test the assumption that that an octave bandwidth is the appropriate scaling for natural images. To do this, we trained the image model on images that were bandpass filtered to two octaves. If a single octave scaling is optimal for our natural image dataset, the learned functions should be localized to an octave width, with different bands specialized for high and low spatial frequencies. For this experiment, the λ_{s_i} and λ_{u_i} parameters were learned independently for each band, so as not to bias the result towards self-similarity between bands.

The resulting functions are shown in figure 2.6, with their 2D spectra, and a line plot depicting the rotational average of the 2D spatial frequency plots. With the exception of the first function, which is localized to low spatial frequencies,

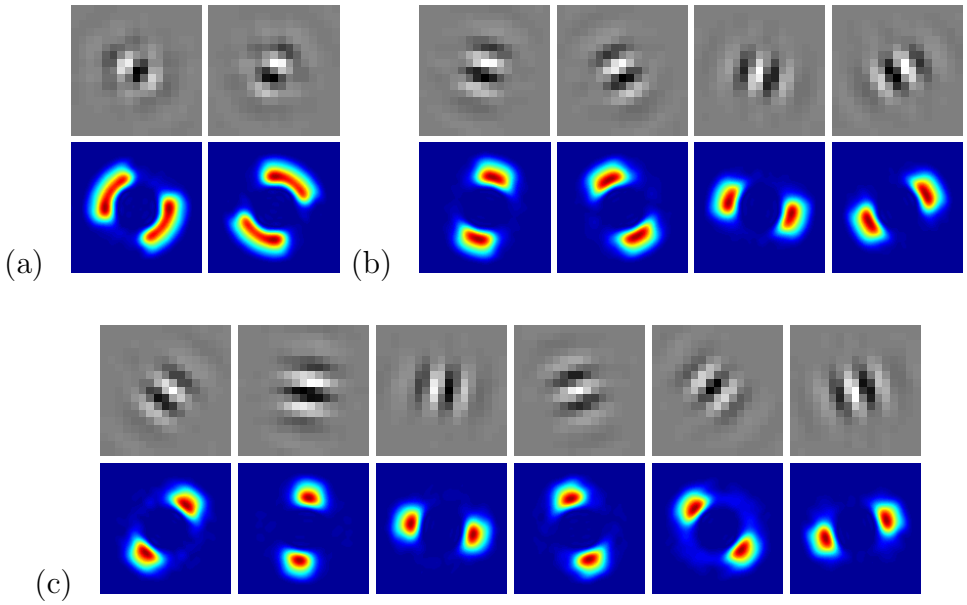
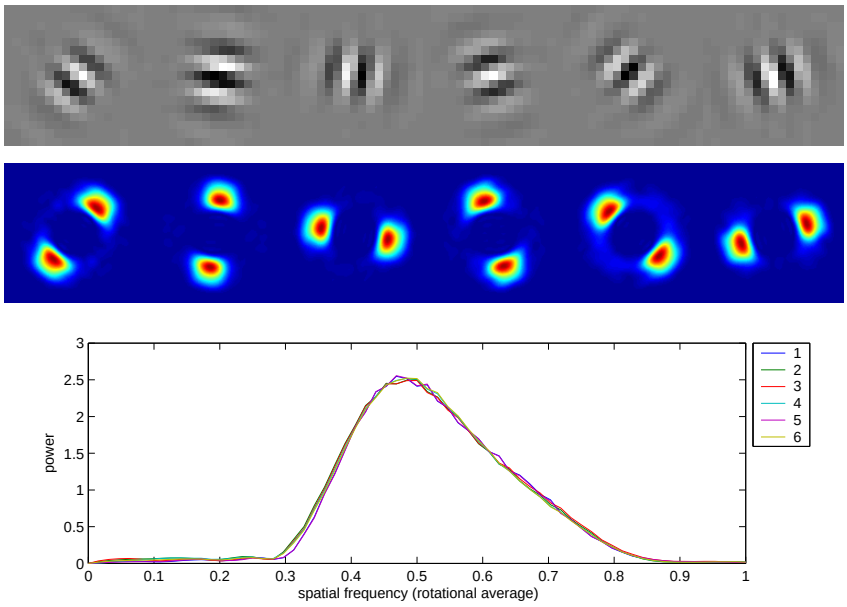
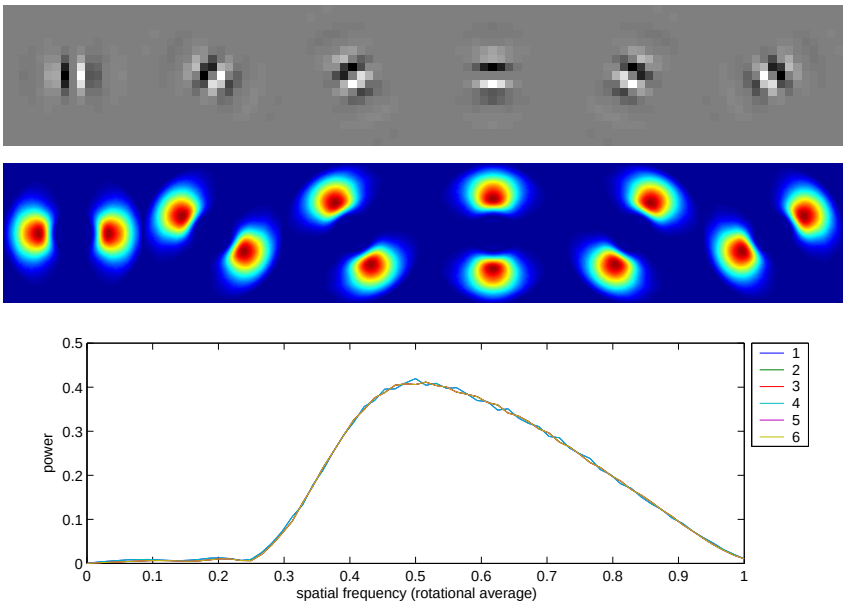


Figure 2.4. Wavelet functions $\psi_i(x, y)$ for varying degrees of overcompleteness, and corresponding spectra showing power as a function of spatial frequency in the 2D Fourier plane. (a) B = 2, (b) B = 4, (c) B = 6.



(a)



(b)

Figure 2.5. (a) Wavelet functions $\psi_i(x, y)$ for 6 bands (B=6) with corresponding 2-D spectra. Line plot depicts the rotational average of the spectra for each filter. (b) Equivalent basis functions for the Steerable Pyramid, when constructed for a single octave scaling and 6 bands, their spectra, and rotational averages.

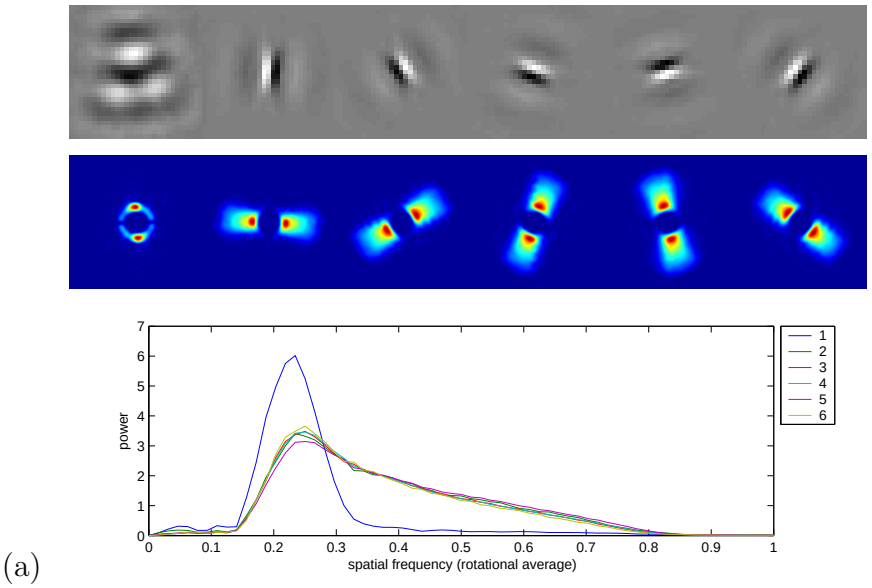
the functions use the full two octave bandwidth rather than separating into more localized high and low frequency subbands. The functions appear to be rotated versions of each other. The learned λ_{s_i} and λ_{u_i} parameters are given in Table 1. From this table, we see that the first wavelet function ψ_1 was rarely used compared to the other functions.

Also shown in figure 2.6 is the Steerable pyramid basis adjusted to span 2 octaves. Differences between the resulting learned functions and the Steerable pyramid basis are now more apparent. We see that the learned basis functions have a power spectrum that tapers off with higher spatial frequencies, unlike the steerable filters, which have a flat power spectrum except for the low-pass and high-pass transitions. The combined power spectrum of the oriented learned functions shown in figure 2.7 (a), more closely matches the power spectrum of natural images (within their bandpass region). The power spectrum of natural images is known to approximate $1/f^2$, where f represents the spatial frequency.

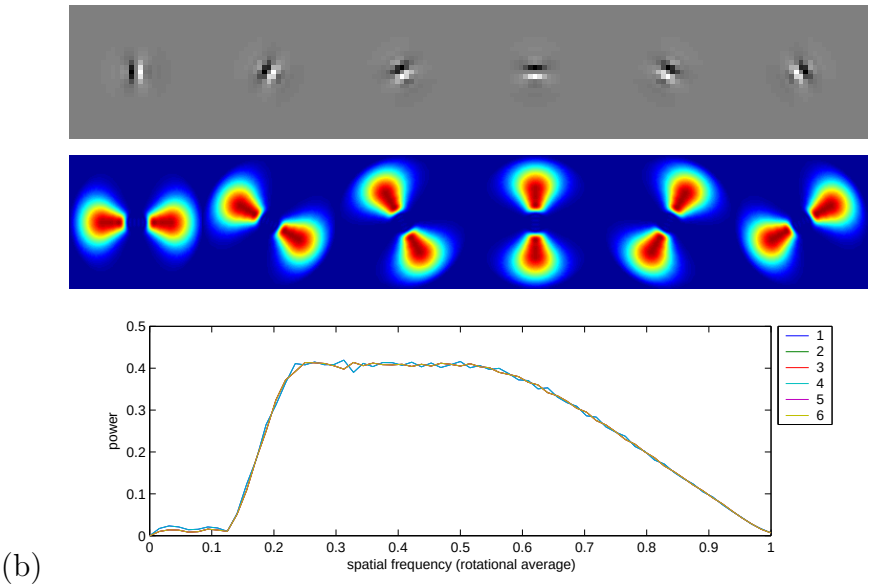
The steerable functions also spread out in orientation with increased spatial frequency, while the learned functions appear to maintain a more constant width in orientation as the frequency increases. This is similar to the ridglet and curvelet functions proposed by Donoho et. al [13]. This raises the question as to how well such functions can be used to tile the entire spatial frequency spectrum, since it would seem that gaps would be produced at the higher spatial frequencies where the spectra do not spread out to meet each other. In figure 2.7 (a), the combined spectra is shown for the five oriented higher frequency functions shown from figure

	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6
λ_{s_i}	.0205	.0020	.0016	.0016	.0021	.0014
λ_{u_i}	8.5109	6.7313	6.8729	7.1202	6.8316	7.2646

Table 2.1. λ_{s_i} and λ_{u_i} for the learned ψ_i function in figure 2.6



(a)



(b)

Figure 2.6. (a) Wavelet functions $\psi_i(x, y)$ for 6 bands (B=6) trained on 2 octave bandpassed images with corresponding 2-D spectra. Line plot depicts the rotational average of the spectrum for each filter. (b) Equivalent steerable pyramid basis functions when constructed for 6 bands and 2 octave scaling, their spectra, and rotational averages.

2.6. Note that their combined spectra are approximately orientation invariant and there are no noticable gaps in the high frequency portions of the spectra. The apparent lack of spreading towards higher spatial frequencies noticed in the spectra of the learned filters may simply be due to the $1/f^2$ power spectrum. To demonstrate this, an idealized version of their combined spectrum, shown in figure 2.7 (b), was generated by taking a rotational average of (a) and interpolating in all directions to produce a rotationally symmetric version of (a). Next, a 1/5th wedge was taken from (b) using a raised cosine function, in the same manner as the Steerable pyramid functions are produced (except with flat power spectrum), to generate the spectrum shown in (c), for a single idealized oriented function. The basis function corresponding to (c) is shown in (d). Note that the spectrum in (c) does not appear to spread out for higher spatial frequencies as much as the steerable pyramid functions, and is more similar to the learned functions.

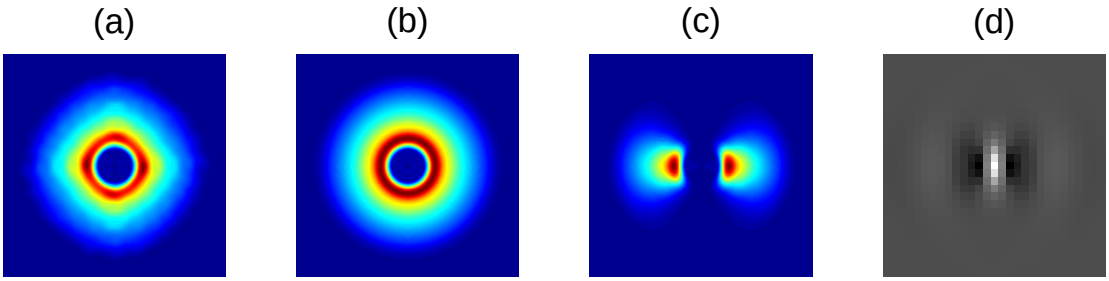


Figure 2.7. Results demonstrating the tiling properties of the 2 octave learned functions from figure 2.6: a) Combined spectra for the five oriented functions showing no obvious gaps. b) Idealized spectra formed by rotational average of (a). c) A 1/5th wedge created by multiplying spectra (b) with a raised cosine function in angular frequency. d) Corresponding basis function for (c).

2.4.3 Sparsity

We evaluated the sparsity of the representations obtained with a set of learned functions with 4-orientation bands using the sampling method and compared these results to the four band Steerable Pyramid filters [31] using the same sampling method. In order to explore the SNR curves for each basis, a variety of values for λ_u were used so as to obtain different levels of sparsity. The same images were used for both bases. The results are given in figure 2.8. Each dot on the line represents a different value of λ_u. The results were similar, with the learned basis yielding slightly higher SNR (about 0.5 dB) for the same number of active coefficients.

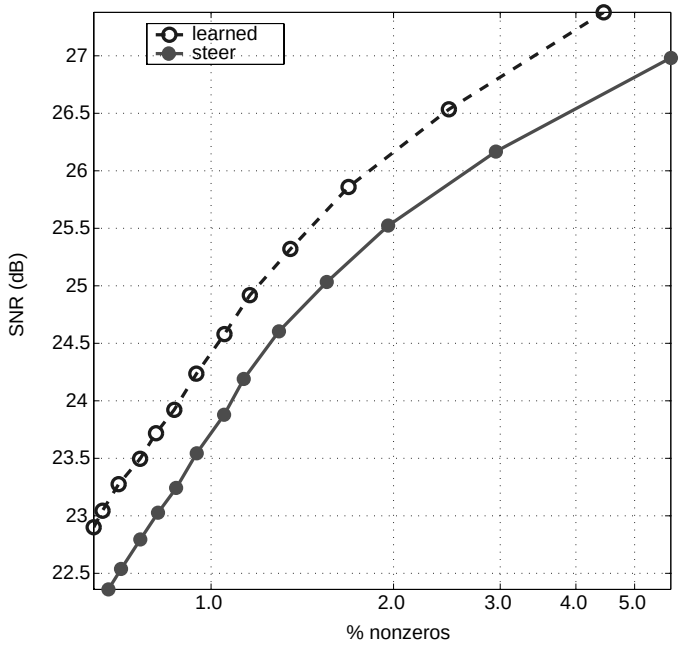


Figure 2.8. Sparsity comparison between the learned basis (top) and the steerable basis (bottom). The y axis represents the signal-to-noise ratio (SNR) in dB achieved for each method for a given percentage of nonzeros.

2.4.4 Denoising

We evaluated our inference method and learned basis functions by denoising images containing known amounts of additive i.i.d. Gaussian noise. Denoising was accomplished by averaging samples taken from the posterior distribution for each image via Gibbs sampling to approximate the posterior mean. Gibbs sampling was performed on a four level pyramid using the 6 band learned wavelet basis, and also using the 6 band Steerable basis. The $\lambda_{\mathbf{n}}, \lambda_{u_i}$ and λ_{s_i} parameters were adapted to each noisy image during sampling using the learning rules in section 2.3 for blind denoising in which the noise variance was assumed to be unknown. We compared these results to the wiener2 function in MATLAB, and also to BayesCore [30], a Bayesian method for computing an optimal soft thresholding, or coring, function for a generalized Laplacian prior. Wiener2 is a version of wiener filtering that accounts for changes in image variance within a specified neighborhood size. For wiener2, the best neighborhood size was used for each image. Table 2 gives the SNR results for each method when applied to some standard test images for three different levels of i.i.d. Gaussian noise with standard deviation σ. Figure 4 shows a cropped subregion of the results for the standard "Einstein" image (not in our training set) with $\sigma = 10$.

Denoising using Gibbs sampling with the Delta-plus-Gaussian prior produce improved results, in terms of mean square error (MSE), over both wiener2 and the Bayes coring method. The oriented basis captures higher order statistical properties which are not captured by the wiener2 method, which only accounts for pairwise statistics captured by the local power spectrum. Thus, it is not surprising that the Bayes coring and Gibbs sampling methods show improved results over wiener2. The MSE between the original and denoised images for

Gibbs sampling are consistently around .5 db or more lower than with the Bayes coring, method. This is true when using either the learned or the Steerable pyramid basis, and even though the Bayes core method used is not completely blind (the noise variance was given to the algorithm). There are two factors that can account for this improvement. First, the Gibbs sampling method takes into account dependencies between coefficients caused by the non-orthogonality of the basis. Second, the form of the prior may be more suitable for natural images. Further experimentation is needed to determine which of these factors is more significant.

Images denoised with the Delta-plus-Gaussian prior also appear less grainy than those of the wiener2 method or the coring method, as the prior encourages

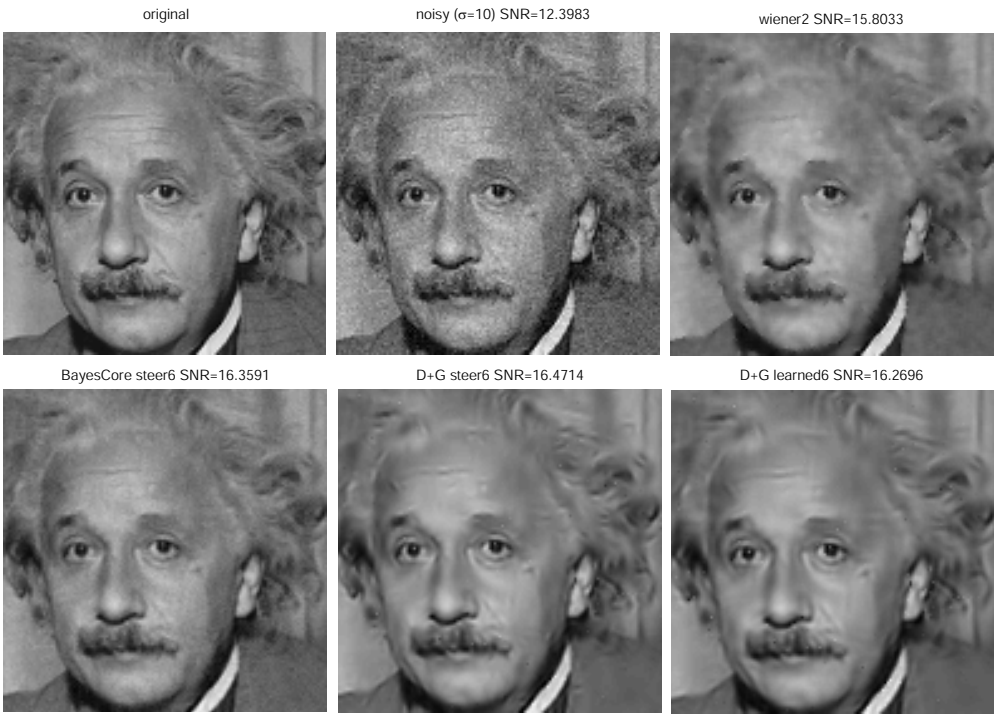


Figure 2.9. Denoising example. A cropped subregion of the Einstein image and denoised images for each noise reduction method for noise std.dev. σ=10.

more coefficients to have exact zero values. However, more wavelet artifacts are visible in areas where a limited number of coefficients became active. Better results might be obtained using a model which captured joint statistics between the coefficients. Such a model could encourage edges to be extended or filled in where there was less information available, if the complex dependencies between neighboring coefficients could be accurately described. It should be noted that better methods than the Bayes coring method used here have been published. The coring method was used because it provides a more direct comparison to the learned model. Models which take into account some joint statistics between coefficients have been shown to achieve better results, although these models are not generative and do not take into account the non-orthogonality of the basis[33].

Image	noise level	noisy	wiener2	BayesCore S6	D+G S6	D+G L6
Einstein	$\sigma = 10$	12.40	15.80	16.36	16.47	16.27
	$\sigma = 20$	6.40	12.61	13.44	13.80	13.85
	$\sigma = 30$	2.89	10.95	11.81	12.28	12.39
Lena	$\sigma = 10$	13.61	19.05	19.91	20.37	20.21
	$\sigma = 20$	7.59	15.51	16.88	17.46	17.54
	$\sigma = 30$	4.07	13.25	14.99	15.48	15.55
Goldhill	$\sigma = 10$	13.86	17.56	18.14	18.10	17.90
	$\sigma = 20$	7.83	14.32	15.18	15.41	15.41
	$\sigma = 30$	4.28	12.64	13.61	13.92	13.95
Fruit	$\sigma = 10$	16.25	21.87	22.09	22.78	22.38
	$\sigma = 20$	10.24	18.15	18.97	19.61	19.42
	$\sigma = 30$	6.70	15.97	17.21	17.72	17.66

Table 2.2. SNR values (in dB) for noisy and denoised images with additive i.i.d. Gaussian noise of std.dev. σ. "D+G" = Gibbs sampling with Delta-plus-Gaussian prior, "S6" = 6-Band Steerable basis, "L6" = 6-Band Learned basis.

2.5 Discussion

We have shown that a wavelet basis and a mixture prior composed of a Dirac delta function and a Gaussian can be adapted to natural images resulting in very sparse image representations. The resulting learned basis is similar to a Steerable basis both in appearance and sparsity of the resulting image representations. This may indicate that the Steerable basis is nearly optimal for producing sparse representations of natural scenes. However, results with fewer constraints show that there are distinct differences between learned basis functions and the Steerable filters. Specifically, the learned functions will span more than one octave, and have a power spectrum which is approximately $1/f^2$. Denoising results indicate that using a sparse prior and an inference method to properly account for the non-orthogonality of the representation may yield an improvement over wavelet coring methods that use filtered coefficients. Denoising results were similar for the learned basis functions and the Steerable pyramid basis. Future work should be done to determine whether the coding gains achieved are due to the choice of prior or the inference/estimation method used.

Chapter 3

Adapting Wavelet Dictionaries to 1D Signals

An orthogonal basis is often not well suited for describing all of the events in a given class of signals. For instance, many audio signals contain a wide variety of features ranging from brief transients to more stationary events. For example, sounds in an animal's natural environment generally contain a wide assortment of harmonic and percussive animal vocalizations, as well as environmental sounds such as wind, rain, rushing water or crunching leaves. When ICA is performed on these types of sounds, very different bases are obtained depending on whether the signals contain speech, animal vocalizations, or ambient environmental sounds [20]. Using a single orthogonal basis for all sounds an animal might encounter is clearly sub-optimal. This will have the unfortunate effect of diluting information across the entire representation about certain events which are not well described by the basis functions. This makes it difficult to extract meaning from the representation regarding which event occurred and when. The most meaningful description of such signals will be one that maintains an explicit

description of the features present in the signal, without obscuring their presence with awkward and overly complex descriptions.

The analogy of language, used by Mallat, appropriately describes the coding strategy needed [22]. In a language, there are many ways to describe the same set of events. We find utility in maintaining such redundancy for language because it allows for descriptions which use only a few words to convey a large amount of information, and because such descriptions provide a tool for reasoning in which important events are made explicit and easy to manipulate. One can imagine how cumbersome it would be to rely solely on a language in which there were barely enough words to describe any event and each event only had one possible description. Describing a single event would require a precise combination of many simple terms, and would usually require the use of every word in the vocabulary. This sounds absurd, yet it is an accurate description of many "languages" used for signal processing, including Fourier and orthogonal wavelet transforms.

This chapter presents methods for adapting a wavelet dictionary to 1D signals, using the same basic wavelet pyramid framework established in the previous chapter for images. The application of the framework to 1D signals is straightforward, as the 2D case is more general. In a similar manner as shown in the previous chapter, we adapt an overcomplete set of wavelet functions which are applied at all positions and scales to generate the full set of basis functions, or dictionary elements. This overcomplete representation provides a framework for the unsupervised learning of basis functions which are well matched to the features present in a given class of 1D signals.

Given such an overcomplete descriptive "language", one can no longer compute the best description of a signal as a linear function of the input signal, as is possible with an orthogonal basis description. Such is the price that must be

paid for having a more meaningful description of the signals. Meaning must be established through a certain amount of disambiguation, requiring a process of inference or pattern recognition. The computational resources required to perform this inference process under certains statistical models, if done exactly and according to theoretical prescription, may be prohibitively expensive. For example, the method of Gibbs sampling presented in the last chapter requires too many computations to be of use for many practical applications. If an equilibrium state cannot be attained in a practical number of iterations, this lack of convergence leads to biases in the sampling and results in instabilities in the learning process. A more cost effective method of inference is needed.

In this chapter we consider a greedy approximation known as *matching pursuit*, introduced by Mallat and Zhang [22]. Matching pursuit is a method for obtaining sparse decompositions of signals with an overcomplete dictionary by selecting basis vectors to represent the signal one at a time, choosing at each step the basis vector that best correlates with the residual. Matching pursuit can be seen as an approximation to a MAP estimate with a certain sparse prior, and provides lower cost alternative for obtaining sparse representations of signals with an overcomplete dictionary than the Gibbs sampling method used in the previous chapter. For wavelet packet or discrete Gabor dictionaries and a signal with N samples, matching pursuit has a total asymptotic complexity of $\mathcal{O}(N^2 \log N)$. Although matching pursuit has a higher computational complexity than other basis selection methods such as the *best orthogonal basis* decomposition decomposition algorithm of Coifman and Wickerhauser [8], which has a total complexity of $\mathcal{O}(N \log N)$, it is more generally suited to describe signals that do not necessarily have stationary properties. Unlike the best basis algorithm, matching pursuit is not limited to an orthogonal set of descriptors for a given signal and is

not restricted to use with signal dictionaries composed of orthogonal bases.

We introduce our own implementation of matching pursuit that has significantly lower computational complexity than the standard matching pursuit for our overcomplete wavelet framework. Because the basis functions generated by the overcomplete wavelet filter bank are self-similar and well localized in position, we can significantly reduce the cost of computing inner product updates when basis functions are added to the representation for a signal. Additionally, we make the simplifying assumption that the optimization can be performed independently within each scale (ignoring dependencies across scale), or as a coarse-to-fine procedure starting with the highest scales of the pyramid and working down, without significant loss to the quality (*ie.* sparsity) of the representation. Thus, each basis vector has a limited number of neighboring vectors with significant overlap. This allows a fast implementation of matching pursuit with a reduced total asymptotic complexity of $\mathcal{O}(N \log N)$, the cost of a single iteration of the standard matching pursuit algorithm, and comparable with the total complexity of more restrictive methods such as the best basis algorithm.

The algorithm used here may also be widely applicable to *vector quantization* methods, popular for data compression [18]. Similarities and differences between matching pursuit and vector quantization are discussed in [22]. A principal difference is that vector quantization methods are typically performed on dictionaries composed without self-similarity constraints and due to the resulting complexity can only be performed directly on signals with low dimensionality (generally smaller than 16) and applied to larger signals by blocking.

In this chapter, results are shown for wavelets adapted to natural sound data, and also to multi-channel electroencephalogram recordings (EEG). The algorithms used here may also be directly applied to other signals besides audio

and EEG. It is assumed that there exists a large number of applications for which multi-scale representations are sometimes appropriate, and for others a convolution model composed of shiftable functions is more appropriate. Additionally, many signals are have multiple channels. Between audio and EEG, we demonstrate a variety of constraints which can be thought of as a toolkit for application to different classes signals. Additionally, the matching pursuit algorithm can be easily extended to any number of dimensions, allowing its application to images or signals of even higher dimensionality.

3.1 Overcomplete wavelet model for 1D signals

We model a 1D signal, $x(t)$, with length N, as a linear superposition of M basis functions $a_i(t)$, with amplitudes s_i, with additive Gaussian i.i.d. noise $n(t)$:

$$x(t) = \sum_{i=1}^{M} s_i \; a_i(t) + n(t) \tag{3.1}$$

where $t = 1..N$ is a position index. A sparse, factorial prior is imposed upon the coefficients s_i. Additionally, we assume there may be more feature descriptors than there are samples in the signal, in which case $M > N$.

In the manner previously described in chapter 2 for images, it is assumed that the basis functions $a_i(t)$ are composed of a small set of temporally localized *mother wavelet* functions $\psi_b(t)$, $b = 1..B$, which are shifted in position and rescaled by factors of two. Basis functions of coarser scales are produced by upsampling the $\psi_b(t)$ functions and blurring with a low-pass filter $\phi(t)$, also known as the *scaling function*. The model in equation 2.1 may be re-expressed to make these parameters explicit:

$$x(t) \;\; = \;\; g^0(t) + n(t) \tag{3.2}$$

$$g^l(t) \;=\; \begin{cases} \left[g^{l+1}(t) \uparrow 2\right] * \phi(t) + \sum_b s_b^l(t) * \psi_b(t) & l < L-1 \\ s^l(t) & l = L-1 \end{cases} \tag{3.3}$$

where the coefficients $s_b^l(t)$ are indexed here by their position (t), band (b) and level of resolution (l) within the pyramid ($l = 0$ is the highest resolution level). The symbol $*$ denotes convolution, and $\uparrow 2$ denotes upsampling by two and is defined as

$$f(t) \uparrow 2 \;\equiv\; \begin{cases} f(\frac{x}{2}, \frac{y}{2}) & x \,\text{even} \;\;\&\;\; y \,\text{even} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

In order to avoid aliasing, the lowest level is not subsampled, so that even with only one band the representation is overcomplete. No restrictions are imposed on the form of the mother wavelet functions ψ_b, except it is assumed that each wavelet functions can be accurately represented by a specified number of sample points T. The term *wavelet* is used here to refer only to constraints of self-similarity across position and scale (a dyadic sampling lattice is imposed), and the functions are also normalized to have zero mean.

Figure 3.1 shows a system diagram for the 1D wavelet pyramid decomposition. For analysis, a signal $x(t)$ is filtered into low-pass and high-pass subbands by correlation, denoted by $\star$, with the scaling function ϕ, and a high-pass filter ϕ_C. To ensure proper reconstruction, ϕ_C is designed in the frequency domain to be the complement of ϕ (when ϕ is applied twice). This is accomplished by selecting ϕ_C to have an amplitude spectrum equal to 1 minus the power spectrum of the scaling function ϕ, and is equivalent to subtracting the eventual contribution of the low-pass subband (after upsampling and convolving again with ϕ) from the signal. Matching pursuit (M.P.) is performed on the high-pass subband to select the coefficients $s_i(t)$ for each wavelet function ψ_b. The low-pass band is subsampled by two, denoted by $(\downarrow 2)$, and the decomposition is recursively applied

at the next higher scale (except at the highest scale) by inserting the diagram into the location marked by a filled circle. A reconstructed signal $\hat{x}(t)$ is obtained by convolving the coefficients with the wavelet functions and adding the result to the the reconstructed low-pass signal from the next highest scale after upsampling by two and convolving with ϕ.

3.2 Inference via matching pursuit

In chapter 2, we specified a sparse prior $P(\mathbf{s})$ over our source coefficients $\mathbf{s}$, and obtained the coefficients to represent a given signal by sampling from the posterior distribution $P(\mathbf{s}|\mathbf{x}) \propto P(\mathbf{x}|\mathbf{s})P(\mathbf{s})$ using a Gibbs sampler. In this chapter, we use a matching pursuit approximation to obtain the coefficients for a signal.

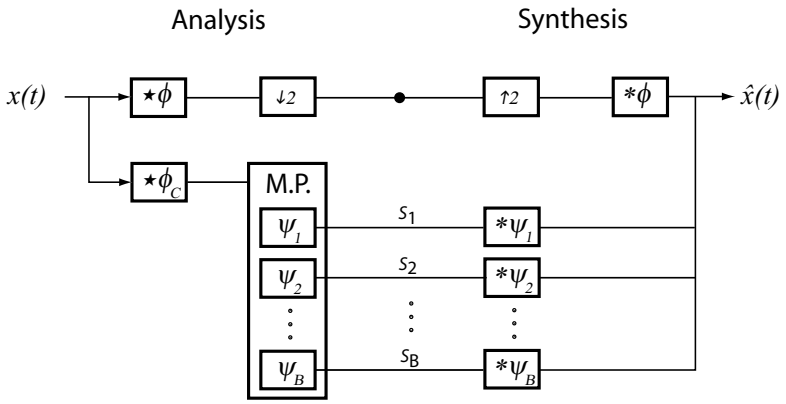


Figure 3.1. System diagram for wavelet pyramid decomposition.

3.2.1 Standard Matching pursuit algorithm

Given a signal $x(t)$, and an overcomplete dictionary of waveforms $a_i(t)$, we seek to decompose $x(t)$ in terms of sparse coefficients $s_i(t)$. Without loss of generality, let $|a_i(t)| = 1 \quad \forall i$, where $|f|$ denotes the vector norm computed by $\sqrt{\sum_t f(t)^2}$.

Let $\hat{x}^m(t) = \sum_i s_i^m \, a_i(t)$ represent the reconstructed approximation to the signal after iteration m, and let $r^m(t) = x(t) - \hat{x}^m(t)$ represent the residual vector obtained from subtracting the reconstructed signal from the original. Thus, $r^m(t)$ is an approximation to the noise term in equation 3.1.

Let the initial source vector $s_i^0 = 0 \quad \forall i$. Thus, $r^0 = x$. At each iteration $m = 1..m'$, we seek to update a single coefficient which maximally reduces the vector norm of the residual, $|r^m|$. This is done by choosing the coefficient k^m whose inner product with the residual has maximum magnitude:

$$k^m = \arg\max_k | < r^{m-1}(t), a_k(t) > | \tag{3.5}$$

where $< f, g >$ denotes the inner product of (f, g) and is defined by

$$< f, g >= \sum_t f(t) \, g(t). \tag{3.6}$$

Coefficient s_k^m is made "active", by assigning it the value that most reduces the length of the current residual, while all other coefficients remain the same:

$$s_i^m = \begin{cases} < r^{m-1}, a_i > & i = k^m \\ s_i^{m-1} & \text{otherwise.} \end{cases} \tag{3.7}$$

In our implementation, the algorithm terminates after iteration m' when the maximum squared inner product reaches a predefined tolerance τ:

$$\tau > \max_k [< r^{m'}, a_k >^2]. \tag{3.8}$$

One can view the matching pursuit algorithm as imposing a type of sparse prior equivalent to a mixture of a delta function at zero and a uniform distribution. Thus, the sparse cost term $S(s)$ of the prior is proportional to the total number of non-zero coefficients. We cast the problem as an energy minimization problem where the energy E is the negative log of the resulting posterior:

$$E = \frac{\lambda_{\mathbf{n}}}{2} \sum_t [x(t) - \sum_i s_i \, a_i(t)]^2 + \lambda_s \sum_i \delta(s_i), \tag{3.9}$$

where $\lambda_{\mathbf{n}}$ is the reciprocal of the noise variance, λ_s is a sparse cost parameter which determines the penalty for an active coefficient, and

$$\delta(x) = \begin{cases} 1 & \text{if} \quad x = 0, \\ 0 & \text{if} \quad x \neq 0. \end{cases} \tag{3.10}$$

Matching pursuit can be seen, then, as a greedy (non-optimal) minimization of this energy function with a tolerance $\tau = 2\lambda_s/\lambda_{\mathbf{n}}$, since the change in energy for making a coefficient k active is:

$$\Delta E = \lambda_s - \frac{\lambda_{\mathbf{n}}}{2} < r, a_k >^2 \tag{3.11}$$

3.2.2 Fast matching pursuit algorithm

The complexity of the the matching pursuit algorithm may be a significant deterrent to more widespread use. At each iteration, inner products with the the new residual must be computed for each basis function, and the basis function must be selected which best correlates with the residual. For an arbitrary dictionary, this may require $\mathcal{O}(M)$ computations per iteration, where M is the number of basis vectors in the dictionary. The matching pursuit implementation presented by Mallat and Zhang for use with a Gabor dictionary has a computational complexity of $\mathcal{O}(N \log N)$ per iteration for a signal of N samples. In order

to achieve this complexity with the Gabor basis set, they imposed certain simplifying restrictions. First, the dictionary element chosen at each iteration is not required to be the one that maximally correlates with the residual. Rather, an element is chosen which has an absolute inner product with the residual that is within a constant factor of the maximum. An updating formula is used which alleviates the need to recompute all of the inner products at each iteration. Rather, only the inner products for elements that have non-negligible overlap with the last chosen element need to be updated. To reduce the number of inner products that must be computed, the inner product computations are limited in precision.

Here we consider how to reduce this complexity for the overcomplete wavelet framework described in this chapter. To do this, we limit the computation to a single scale at a time, thus describing each bandpass portion of the signal separately. This ensures that of the dictionary elements being considered at any given time, each element has non-zero inner-product with only a few neighboring functions. Since the discrete wavelet functions are already well localized, we do not need to impose additional restrictions on the precision of the computations to limit the number of inner products that must be computed. At each iteration, the inner products are computed efficiently using the updating formula described in [22]. By themselves, these steps only ensure a complexity similar to that achieved by Mallat and Zhang for the Gabor basis set. For either basis set, locating the next element to be updated at each iteration represents the greatest bottleneck. Straightforward search requires computations proportional to the number of basis elements. By making use of a special data structure to store the inner product computations, however, we can alleviate this bottleneck and achieve a complexity of $\mathcal{O}(\log N)$ for each iteration resulting in a *total* complexity of $\mathcal{O}(N \log N)$ for the full matching pursuit.

The proposed Fast MP algorithm follows the same basic steps as the standard algorithm introduced by Mallat and Zhang. Once a basis vector a_k is added to the representation, we compute the inner product of the new residual with any basis vector a_i using the formula

$$< r^{m+1}, a_i > = < r^m, a_i > \; - \; s_k^m < a_k, a_i > . \tag{3.12}$$

Since the inner products between the basis vectors $< a_k, a_i >$ can be precomputed, this update takes only constant time at each iteration for each basis vector having non-zero inner product with the new vector a_k. Each basis function has a relatively small number $K = B(2T - 1)$ "neighboring" basis functions for which $< a_i, a_j >$ is non-zero. Here B represents the number of bands, or different ψ_b functions, and T is the extent of each function.

To use the update formula, it is necessary to first precompute the inner products between each basis function a_i and the signal x. For an arbitrary dictionary, the asymptotic cost to initially precompute the inner products between M dictionary elements would be $\mathcal{O}(M^2 N)$. For the wavelet dictionaries used here, however, the inner products between two wavelet functions are the same for a given relative position of the two functions regardless of where in the signal they occur. Thus, the inner products between all pairs of basis functions for a given scale can be obtained by cross-correlating a small set of mother wavelet functions, greatly reducing this cost. The cost for computing the cross-correlation of these functions is $\mathcal{O}(BK)$. This is a one time cost, as it depends only on the dictionary, and does not need to be repeated for each signal decomposition. The cost to precompute the initial inner products between all of the basis functions and the signal is $\mathcal{O}(MT)$, and must be computed only once for a given signal decomposition.

In order to efficiently locate the coefficient to be updated at each iteration, the inner products between the basis functions and the current residual are stored in a heap. A *heap* is a tree, having the property that the value stored at each node the parent index is larger than each of its children. Each node contains the absolute value of the inner product of a basis function a_i with the current residual at iteration m: $|<r^m, a_i>|$, and the pointer i which references the basis function a_i. This permits the next basis function to update to be located in constant time, as it is located at the top of the heap, with an additional cost of $\mathcal{O}(K \log M)$ each iteration to update the heap. If we assume that B and L, the number of levels in the wavelet pyramid, are both fixed relative to the length of the signal N, then the number of basis function elements is proportional to N. Assuming K is a relatively small fixed constant, this is sufficient to achieve the $\mathcal{O}(N \log N)$ total asymptotic complexity with regard to the size of the input.

However, if B or T are not small, the neighborhood size K may be signifi-

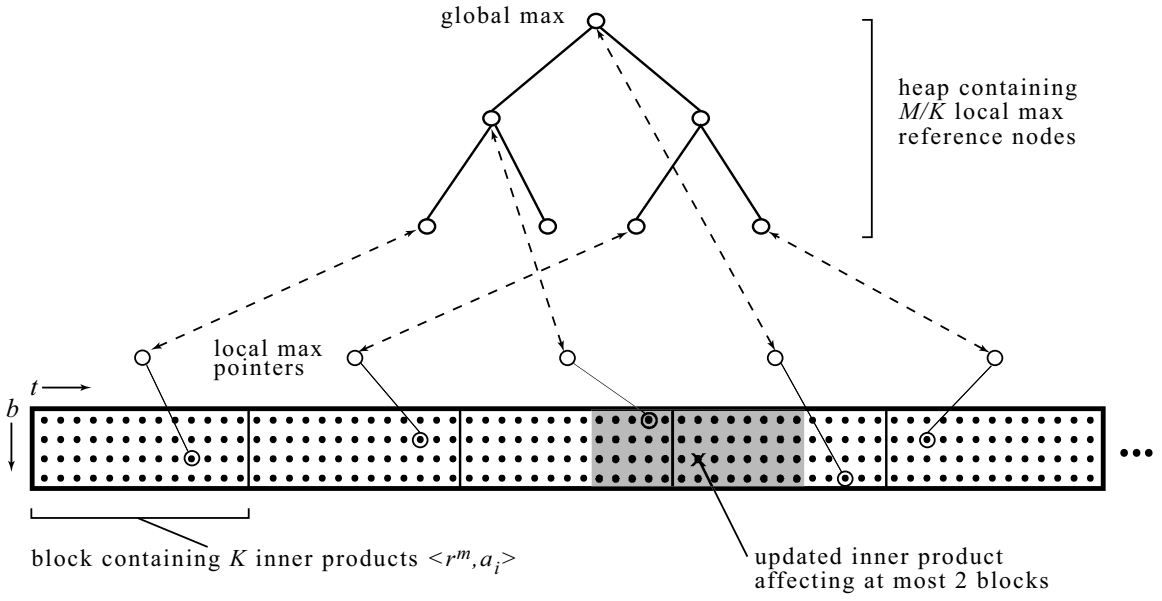


Figure 3.2. Data structure used for storing inner product values. See text.

cant. The cost per iteration relative to K and M is $\mathcal{O}(K \log M)$, which accounts for selecting the next basis element and updating K inner products in the heap. However, we can reduce this cost by a slight modification to the algorithm. By altering the data structure used to store the inner products, an improved complexity of $\mathcal{O}(K + \log(M/K))$ can be achieved. Rather than storing all of the inner products in the heap, most of the inner products are stored in an BxN array in the order of their position in the signal. This array is broken into M/K logical blocks of size K. Refer to diagram in figure 3.2. The small dots in the diagram represent the inner products between the residual and each basis vector. For each block, a "local max" pointer is maintained which references the element having the maximum absolute inner product value within that block. A node for each local max pointer is stored in the heap which contains $\lceil M/K \rceil$ nodes, one for each block of size K. Each heap node is indexed by the absolute inner product value which is maximal for one of the blocks, and contains a pointer to the block.

An iteration of the matching pursuit proceeds as follows. The next basis element to be updated can be found by following the pointer for the "global max" node at the top of the heap. The selected coefficient is updated according to 3.7. Neighboring inner product values are updated using the update formula 3.12. This affects at most two blocks, requiring no more than $2K$ comparisons to update the two local max pointers for a cost of $\mathcal{O}(K)$. The heap node for each local max pointer that has been changed is then updated in the heap. The cost for each heap update is proportional to the height of the heap, $\log(M/K)$. Since there are at most two heap nodes updated, the complete cost for the iteration is $\mathcal{O}(K + \log(M/K))$.

3.2.3 Gibbs sampling versus matching pursuit

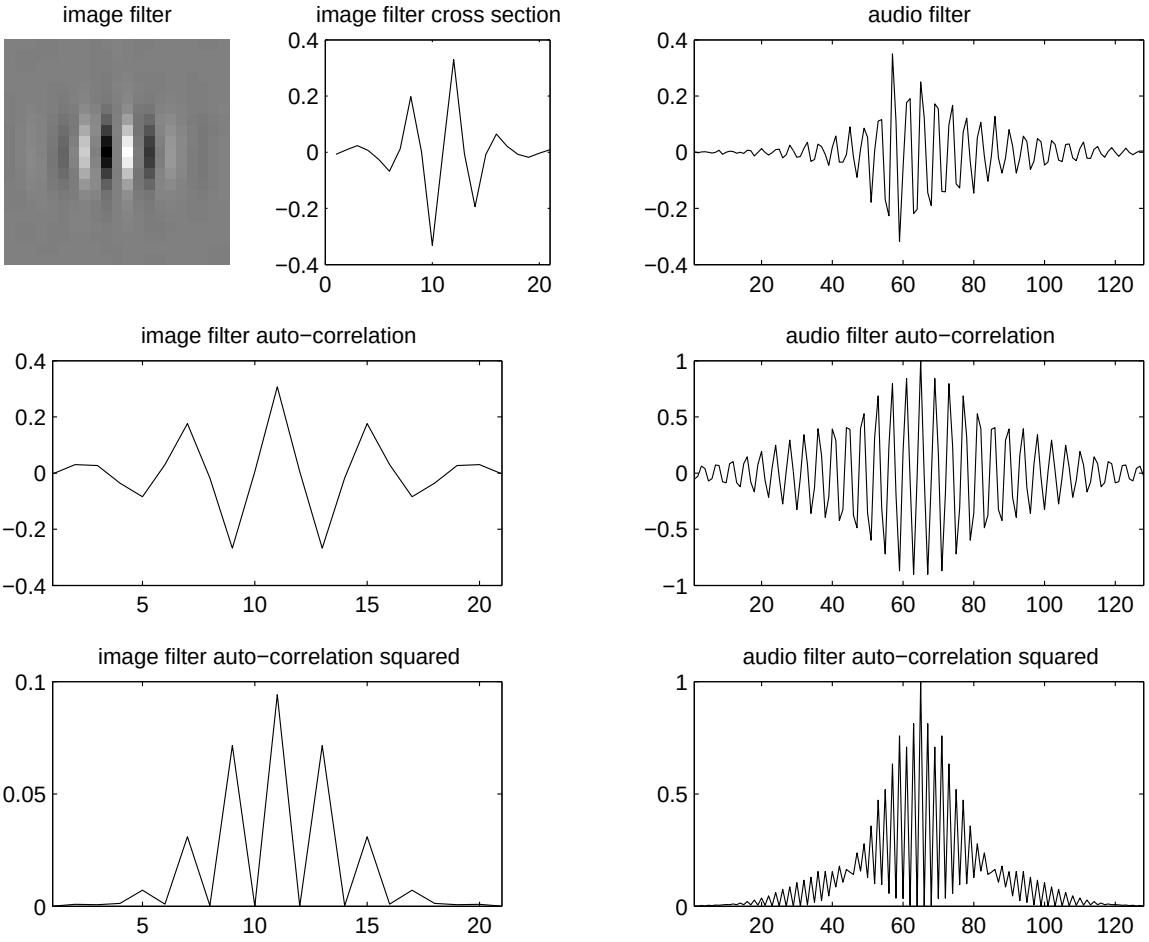


Figure 3.3. A learned audio filter (right) and a cross section of a learned image filter (left) are depicted. Below each function is shown its auto-correlation which is then squared. Peaks reveal neighboring sub-optimal positions for the filter when describing a signal which can lead to local minima in the solution space. The highly oscillatory nature of the audio filter produces many more local minima than the image filter.

Besides having reduced complexity, matching pursuit presents an additional

advantage over Gibbs sampling in avoiding local minima. When first attempting to adapt wavelet functions to natural sounds, we applied the Gibbs sampling method with a Delta-plus-Gaussian prior described in the previous chapter. However, significant problems were encountered. The learning procedure failed to converge properly despite running the sampling procedure for a significant number of iterations. Gibbs sampling is only guaranteed to converge after an infinite number of iterations, and in practice there is not assurance of obtaining a true sample from the posterior. The convergence problems experienced are believed to be a result of problems with extreme local minima due to the size and oscillatory nature of the optimal basis functions.

Gibbs sampling proceeds by considering each position for a function in turn, and either turning on (selecting a non-zero coefficient) or turning off a given basis function at that point. As shown in figure 3.3 Gibbs sampling is likely to select a sub-optimal position for such a function as it will first reach many positions which reduce the residual almost as much as the optimal position. Once set, it is unlikely to later reverse that decision. These nearby locations are indicated by peaks in the squared auto-correlation of the basis functions which are of similar height to the center peak. While these peaks also occur in the wavelet functions adapted to natural images, there are fewer peaks that may cause problems. If the basis function is used inconsistently, the learning procedure will not give good results.

Matching pursuit, on the other hand, initially selects the position which most reduces the residual error of the description, so that it easily overcomes this problem. If the signal can be described accurately by a single function, for instance, matching pursuit will choose the correct interpretation of the signal in a single step. While it may not select the optimal description for some signals, as combi-

nations of features may be more optimal than the features the greedy approach selects, matching pursuit avoids trivial forms of sub-optimality which result from simply shifting a function from its best position.

3.3 Adapting the model to natural sounds

The basis functions a_i are adapted so as to maximize the average log-probability of a set of signals under the model. When trained on natural sounds, the basis functions become localized in time and frequency. This has been previously demonstrated in the context of a critically sampled and noiseless ICA model on small segments of audio by Lewicki [20], who showed that the time-frequency tiling of basis functions trained on environmental sounds was similar to a wavelet representation, while animal vocalizations more closely resembled the Fourier basis being more localized in frequency than in time. Time-frequency analysis of speech sounds resembles a compromise between environmental sounds and animal vocalizations.

Here, instead of blocking sounds into segments we adapt a set of functions which can be shifted and scaled within the pyramid framework to describe signals of any length. Gradient descent on the maximum likelihood objective yields the following update rule:

$$\Delta\psi_b(t) \quad \propto \quad \sum_{l=0}^{L-2} \lambda_{\mathbf{n}} \left\langle \left\langle e^l(t) \star s_b^l(t) \right\rangle_{P(\mathbf{s}|x(t),\mathbf{A})} \right\rangle, \tag{3.13}$$

where $\mathbf{A}$ is the matrix of basis functions, $\star$ denotes cross correlation, and

$$e^l(t) \quad = \quad \begin{cases} x(t) - \hat{x}(t) & l = 0 \\ \left[e^{l-1}(t) \star \phi \right] \downarrow 2 & 0 < l < L-1 \end{cases} \tag{3.14}$$

is the reconstruction error $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{A}\,\mathbf{s}$ filtered up through the pyramid. It is

only necessary to compute a center portion of each cross correlation having the same extent of the $\psi_b(t)$ functions. The outer brackets denote averaging over many sounds. The notation $\langle\rangle_{P()}$ denotes averaging the quantity in brackets while sampling from the specified distribution.

Although equation 3.13 calls for the coefficients $\mathbf{s}$ to be sampled from the posterior $P(\mathbf{s}|x(t), \mathbf{A})$, we approximate a sample from the posterior with the sparse decomposition obtained from the matching pursuit algorithm. Matching pursuit provides a solution which is close to a maximum *a posteriori* (MAP) estimate. The MAP estimate has previously been used for learning sparse components of natural image patches [24]. Although the MAP estimate is more efficient to compute, there is a small price to pay for this simplification with regard to the learning procedure. Biases caused by using the MAP estimate instead of sampling from the posterior prevent other parameters, such as the noise variance, from being fit to the data and also necessitates renormalization of the basis functions after each update.

Ideally, all coefficients would be inferred simultaneously in the matching pursuit algorithm. However, this involves many additional computations since basis functions at the higher scales (lower resolution), overlap with many other basis functions from the higher resolution scales. Instead, the signal is first bandpass filtered for each scale and the matching pursuit algorithm is applied separately to the bandpass filtered signal for each scale. The bandpass signal for level l is computed as:

$$x^l(t) = x^l_\phi(t) \star \phi_C \tag{3.15}$$

where

$$x^l_\phi(t) \quad = \quad \begin{cases} x(t) & l = 0 \\ \left[x^{l-1}_\phi(t) \star \phi\right] \downarrow 2 & 1 < l < L-1 \end{cases} \tag{3.16}$$

ϕ and ϕ_C are complementary lowpass and highpass filters designed in the frequency domain. This is depicted graphically in figure 3.1.

For a level l, $x^l(t)$ is then used as the initial residual r^0 for the fast matching pursuit algorithm. The following simplified learning rule is used in place of equation 3.13:

$$\Delta\psi_b(t) \quad \propto \quad \sum_{l=0}^{L-2} \lambda_{\mathbf{n}} \left\langle \hat{e}^{\,l}(t) \star \hat{s}_b^l(t) \right\rangle, \tag{3.17}$$

where $\hat{s}_b^l(t)$ are the sparse coefficients obtained using matching pursuit for band b and level l, and $\hat{e}^{\,l}(t)$ is the remaining residual error for level l after the matching pursuit:

$$\hat{e}^{\,l}(t) = x^l(t) - \sum_b \hat{s}_b^l(t) * \psi_b(t). \tag{3.18}$$

By performing the matching pursuit algorithm separately at each scale, we are treating the noise at each level as i.i.d. Gaussian. This is actually not the case, since correlations are introduced into the bandpass images when filtering with the scaling functions. It is possible to overcome this problem by computing the inner products between the basis functions $< a_k, a_i >$ used in the matching pursuit update formula 3.12 as the inner products between the effective basis functions (upsampled for each level and cross correlated), rather than as the inner products between the $\psi_b(t)$ functions. Preliminary tests did not show a significant difference between the two methods. The results shown here were obtained using the simpler method in which the precomputed inner products between the basis functions were generated by cross correlation of the $\psi_b(t)$ functions, which were used for all levels.

3.4 Results

3.4.1 Combined scales

Using equation 3.17, we adapted a set of wavelet functions $\psi_b(t)$ which are shiftable and scalable to a database containing a broad assortment of nature recordings[1]. The wavelet functions were initialized to a random state before the learning procedure started. A single set of mother wavelet filters were adapted to all scales (or levels) of the pyramid. The nature sounds included a wide variety of animal vocalizations, as well as environmental sounds such as rain and ocean waves. The resulting mother wavelet functions are shown in figure 3.4 for the case of 4 wavelet bands (B=4), and figure 3.5 for 6 wavelet bands (B=6). Each wavelet function contains 128 sample points (T=128). Each mother wavelet function is drawn in an arbitrary color for use as reference in the frequency plot below the functions. The learned functions are similar to Gabor functions, localized in both time and frequency, and evenly tile the allocated bandwidth.

3.4.2 Separate scales

In order to test whether the statistics of our natural sounds are scale invariant, we also adapted a separate separate wavelet functions ψ_b^l to each scale (or pyramid level) for the natural image data used in the previous experiment. The results are shown in figure 3.6 for 4 separate levels. Levels are shown from left (being the highest frequency scale) to right (lowest frequency scale). Wavelet functions for lower frequency scales are applied by upsampling by powers of two and smoothing to generate the effective basis function. Note that the statistics of our data do not appear to be completely scale invariant, as the wavelet functions for each level

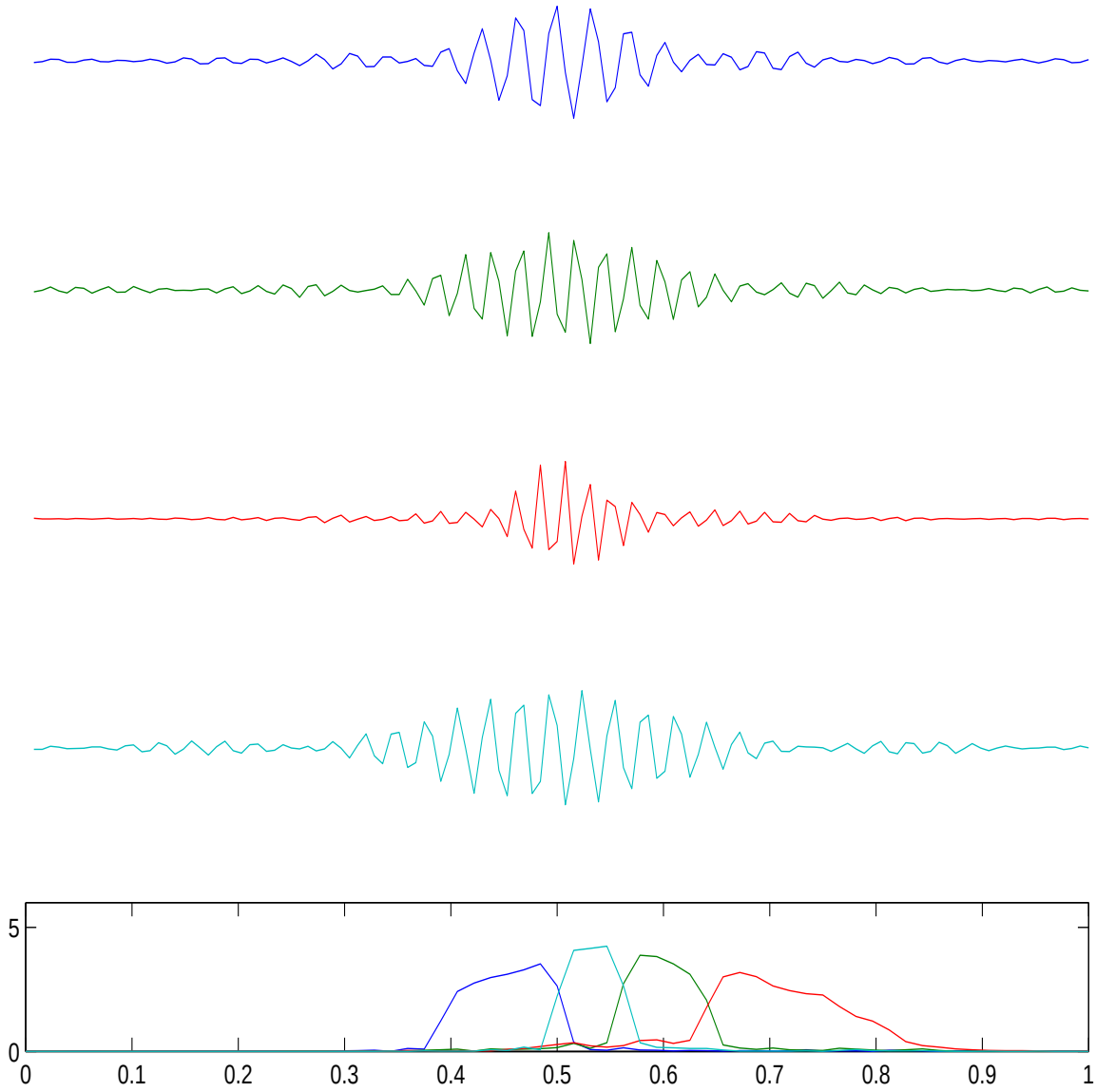


Figure 3.4. Wavelets ψ_b adapted to audio for 4 bands (B=4). Wavelets (above) are depicted in color to provide a reference to their spectral plot (below). Plot depicts amplitude as a function of temporal frequency with 1 as Nyquist.

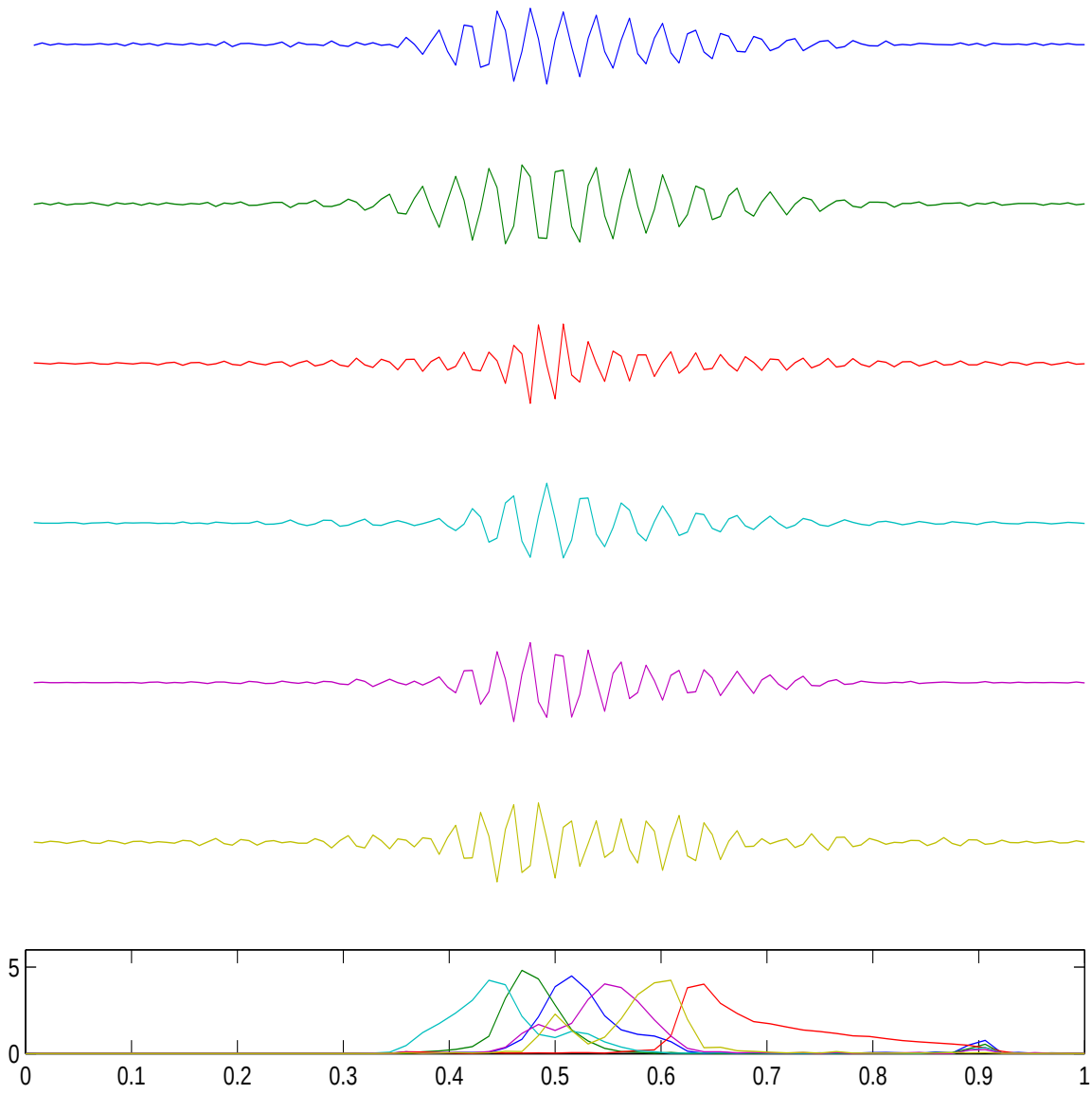


Figure 3.5. Wavelets ψ_b adapted to audio for 6 bands (B=6). Wavelets (above) are depicted in color to provide a reference to their spectral plot (below). Plot depicts amplitude as a function of temporal frequency with 1 as Nyquist.

are different in both appearance and in the manner in which they tile frequency. These differences may likely be due to a limit in the variety of the training data, however. Although hours of training data was used, only a limited number of environments were represented in the training data.

3.4.3 Sound textures

In this section, results are given for training data that is limited to a single environment, or "sound texture". Figure 3.7 shows a set of mother wavelet functions adapted to separate scales for an ensemble of sounds taken from a river environment. The river sounds contain only sounds of rushing and trickling water. Each wavelet function contains 256 sample points. Separate functions were learned for 4 different scales, or resolutions of the pyramid. Levels are shown from left (highest resolution scale) to right (lowest resolution scale). The learned wavelets for these sounds appear remarkably scale invariant. This suggests the intriguing notion that sounds made by rushing water are extremely similar across scales, while other sounds are not. As an example, figure 3.8 shows the resulting wavelet functions after adapting to only frog vocalizations. For these sounds, only a small number of functions are similar across scale. Also, the wavelet functions are more localized in frequency and less localized in time as one might expect given the harmonic nature of the sounds.

To learn more about the statistics that were being captured by the model, synthetic sounds were generated from the learned wavelet functions for the river and frog "sound textures". To do this, the marginal statistics of the wavelet coefficients were measured after performing matching pursuit to obtain a pyramid decomposition of each sound. Coefficients were then sampled independently from

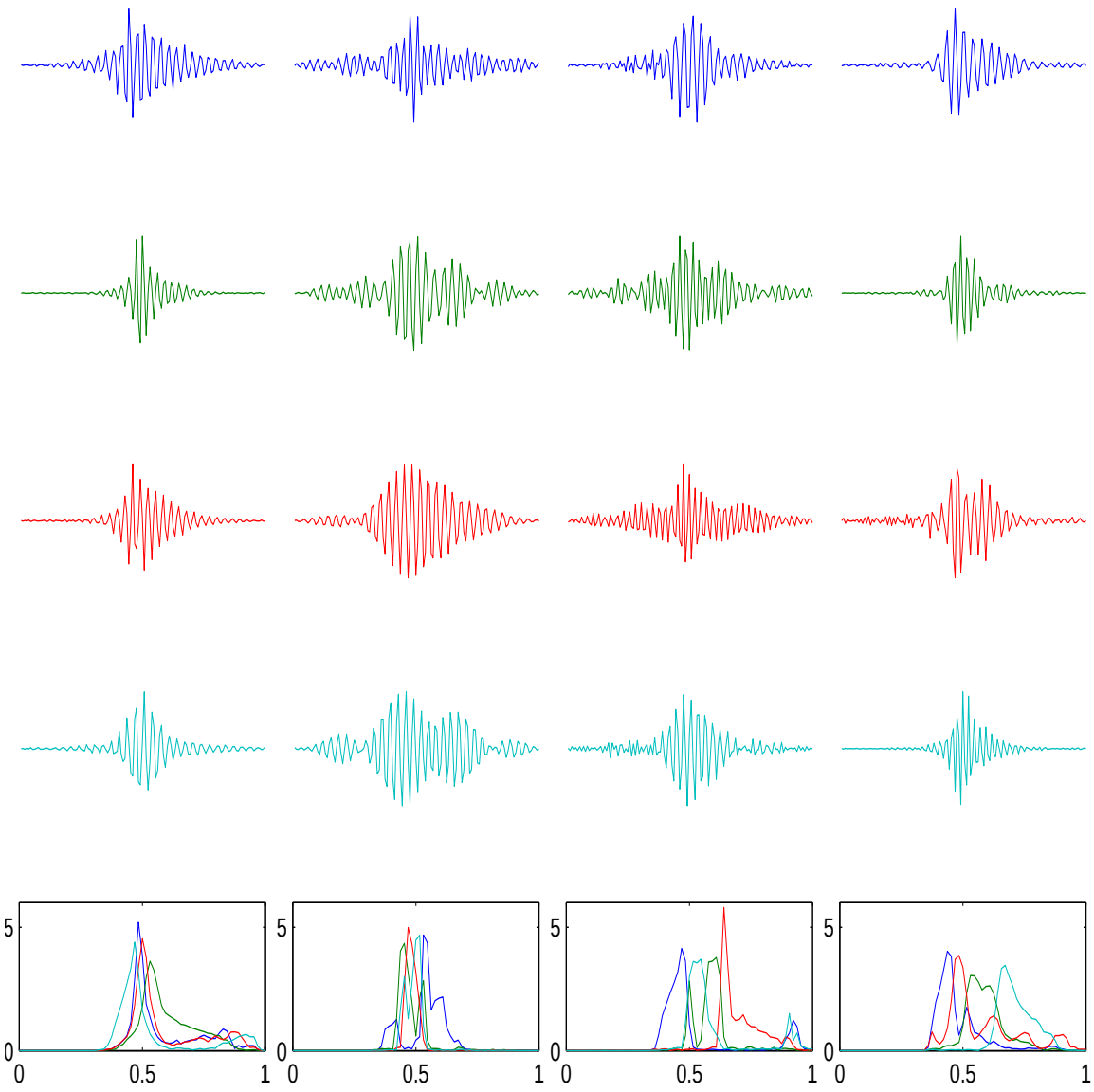


Figure 3.6. Wavelets ψ_b^l adapted separate scales/levels. Results are for 6 bands (B=6) and 4 levels (shown left to right in order of decreasing frequency). Wavelets (above) are depicted in color to provide a reference to their spectral plot (below). Plot depicts amplitude as a function of temporal frequency with 1 as Nyquist.

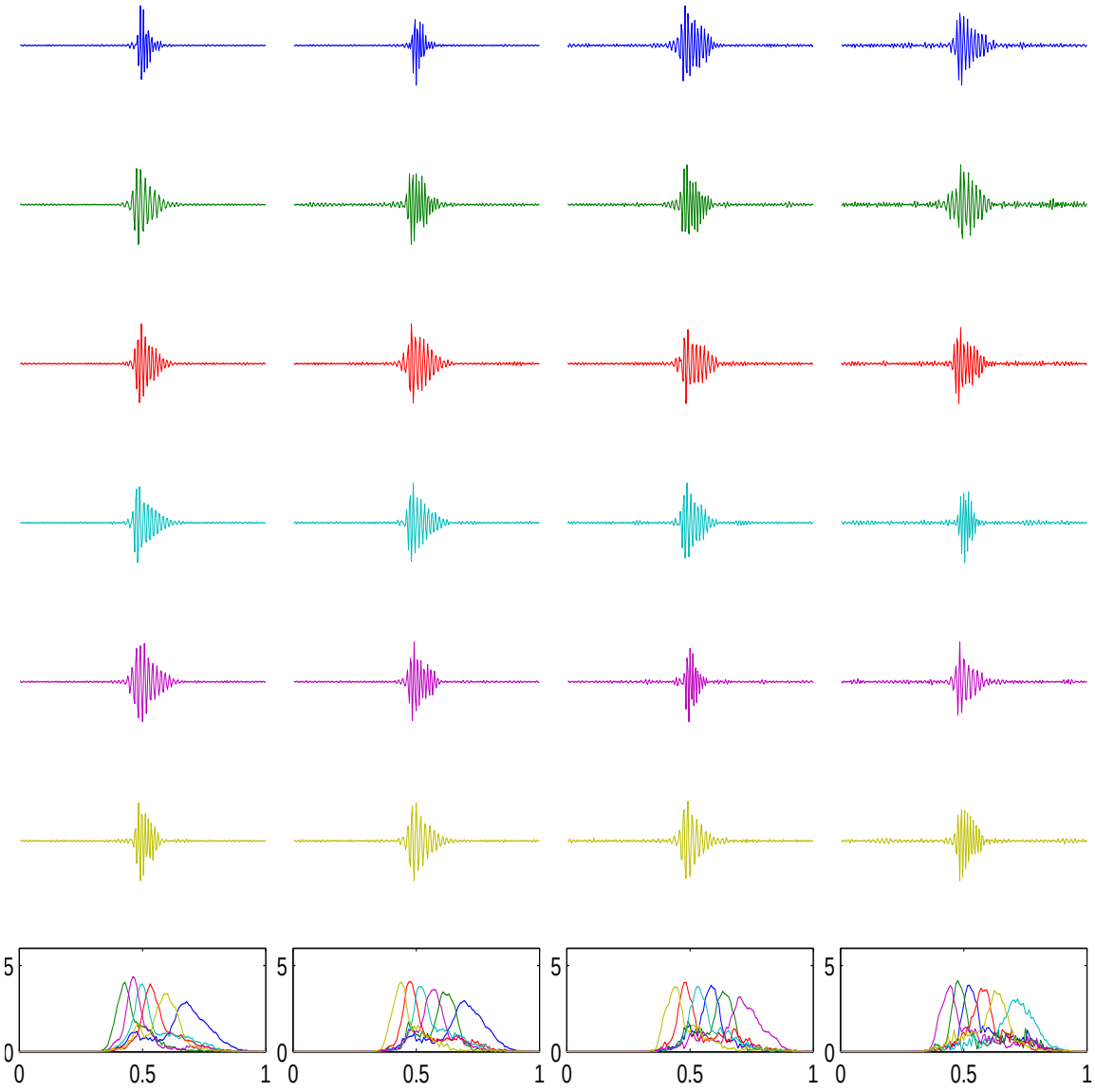


Figure 3.7. Wavelets ψ_b^l adapted to river and stream sounds for 6 bands (B=6)and 4 levels (shown left to right in order of decreasing resolution). Wavelets (above) are depicted in color to provide a reference to their spectral plot (below). Plots depicts amplitude as a function of temporal frequency with 1 as Nyquist.

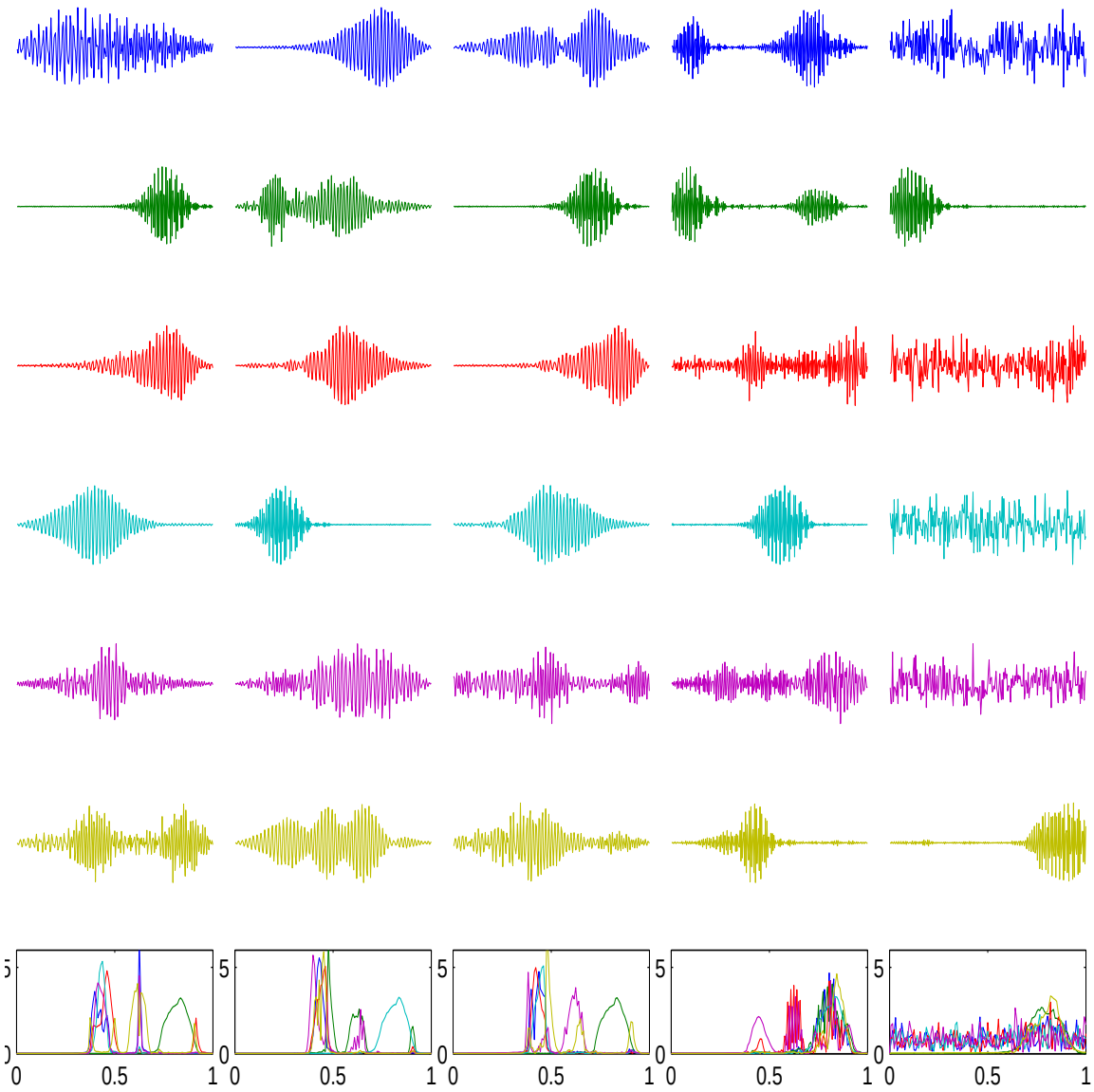


Figure 3.8. Wavelets ψ_b^l adapted to a chorus of frog sounds for 6 bands (B=6)and 5 levels (shown left to right in order of decreasing resolution). Wavelets (above) are depicted in color to provide a reference to their spectral plot (below). Plots depicts amplitude as a function of temporal frequency with 1 as Nyquist.

the measured histograms, and the resulting pyramids inverted to synthesize the sounds.

For wavelets adapted to river sounds, the synthetic sounds are remarkably similar to natural water. Although it is possible to discern differences after hearing both the real and synthetic versions, the synthetic water can easily pass for actual recordings. On the other hand, the synthetic frog sounds are quite different from actual frog recordings. While there is some qualitative similarity between the synthetic and actual frog sounds, much of the temporal structure of the frog calls are lost in the synthetic reconstructions. This is understandable, since the statistics of the frog vocalizations are not as homogeneous as the water sounds.

3.5 Extending the model to EEG

In addition to natural sounds, we also consider a different class of 1D signals obtained by multi-electrode electroencephalogram (EEG) recordings of human brain activity. By applying the matching pursuit and sparse coding methods to EEG, we demonstrate that overcomplete dictionaries can be learned for a wide variety of signals having very different types of features. In addition, this method has direct applications for neuroscience research which seeks to isolate the timing of certain information processing events occurring in the brain, or to relate EEG activity to events occurring in the real world.

The most significant drawback to using EEG for detecting brain activity is the large amount of "noise" in the signal caused by field generating events which are unrelated to the experiment at hand. ICA is commonly used to identify some of these events, such as eye-blinks, in order to filter them from the signal. ICA models signals as a linear mixture of sparse and statistically independent source

components and is applied to multi-electrode EEG in the form of a single N by N "demixing" matrix $\mathbf{W}$ which is multiplied to the N dimensional vectors $\mathbf{x}_t$ obtained for N channels at each time sample t. Therefore, these ICA methods only capture some statistical dependencies occuring across channels, and not across time, and are limited to descriptions which are *critically sampled*, having the same number of sources as channels. The sparse coding method described here not only captures these dependencies across channels, but models temporal dependencies present in the signals as well. Additionally, the overcomplete framework provides flexibility for a more accurate description of EEG features. This may allow a much more accurate identification of event related signals in EEG. The goal, for now, is to assess the feasibility of this type of analysis as a proof of concept by adapting a wavelet basis to the EEG data for a single subject. This also serves to demonstrate a different approach which may serve useful in adapting wavelets to other types of data.

3.5.1 Method

This study demonstrates how to apply sparse coding to multi-channel data, as a collection of 1D signals related in time, by adapting a set of multi-channel mother wavelet functions $\psi_b(t, c)$ which are shifted in time to generate the full basis set. A multi-channel EEG signal $x(t, c)$, indexed by time t and channel c, is modeled as a linear superposition of M basis functions $a_i(t, c)$ plus i.i.d. Gaussian noise:

$$x(t, c) = \sum_b s_b(t) * \psi_b(t, c) + n(t, c) \tag{3.19}$$

where the coefficients $s_b(t)$ are indexed here by their position (t), and band (b). The symbol $*$ denotes convolution in time (not across channels). As it is not

clear whether a multi-scale representation is appropriate for EEG data, we have chosen to modify the model from the previous section so that the basis functions are generated by wavelet functions that are shifted in time, but are not rescaled. Figure 3.9 depicts the model graphically. Here, each datapoint and coefficient is shown as a dot. Each coefficient in band b is multiplied by the mother wavelet $\psi_b(t, c)$ which is placed into the image at the coefficient's location.

For EEG, it seems unlikely that each of the feature vectors in the data has a negative counterpart. Because of this, we also apply a non-negativity constraint to the coefficients in the representation. This non-negativity constraint can be implemented with a trivial change to the matching pursuit algorithm, and makes the model more general. The only change necessary is to select at each iteration the coefficient having maximal inner product with the residual, rather than the

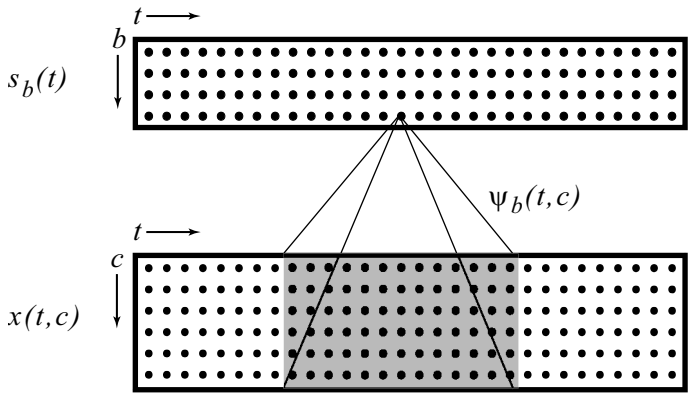


Figure 3.9. Diagram depicting the model for a multi-channel signal $x(t, c)$ indexed by time t and channel c. Each coefficient $s_b(t)$ and datapoint $x(t, c)$ is indicated by a dot. To generate the signal, coefficients are multiplied by a multi-channel mother wavelet function $\psi_b(t, c)$ indicated by the shaded region, which is added to the signal centered at the coefficient's position.

maximal *absolute* inner product with the residual. That is, we simply ignore the negative valued inner products.

3.5.2 Results

The model was adapted to 22 channel EEG data collected from a single subject. This data was taken from a study in which subjects viewed images on a computer screen and responded by pressing a button after each image presentation as to whether a specified object was present in the image. 44 mother wavelet functions were adapted to the EEG data using equation 3.17. Each function is of size 22x128. The functions were initialized to a completely random state before training. The learned wavelet functions are shown in figure 3.10. The first wavelet function is consistently used to describe EEG features caused by eyeblinks. Other functions may be correlated to specific events in the experiment protocol, or to general brain activity of the subject. Note that many of the basis functions have frequency components, and are clearly non random. The next step, not taken here, would be to reverse correlate the wavelet coefficients with events recorded in the protocal, such as button presses or image presentations.

3.6 Discussion

Wavelets have become an increasingly popular tool for signal processing applications. Overcomplete representations, in particular, such as wavelet dictionaries, are well suited for describing many types of natural signals, as the overcompleteness allows for feature descriptors which are better matched to features present in the signal. We have shown how matching pursuit can be used as an efficient

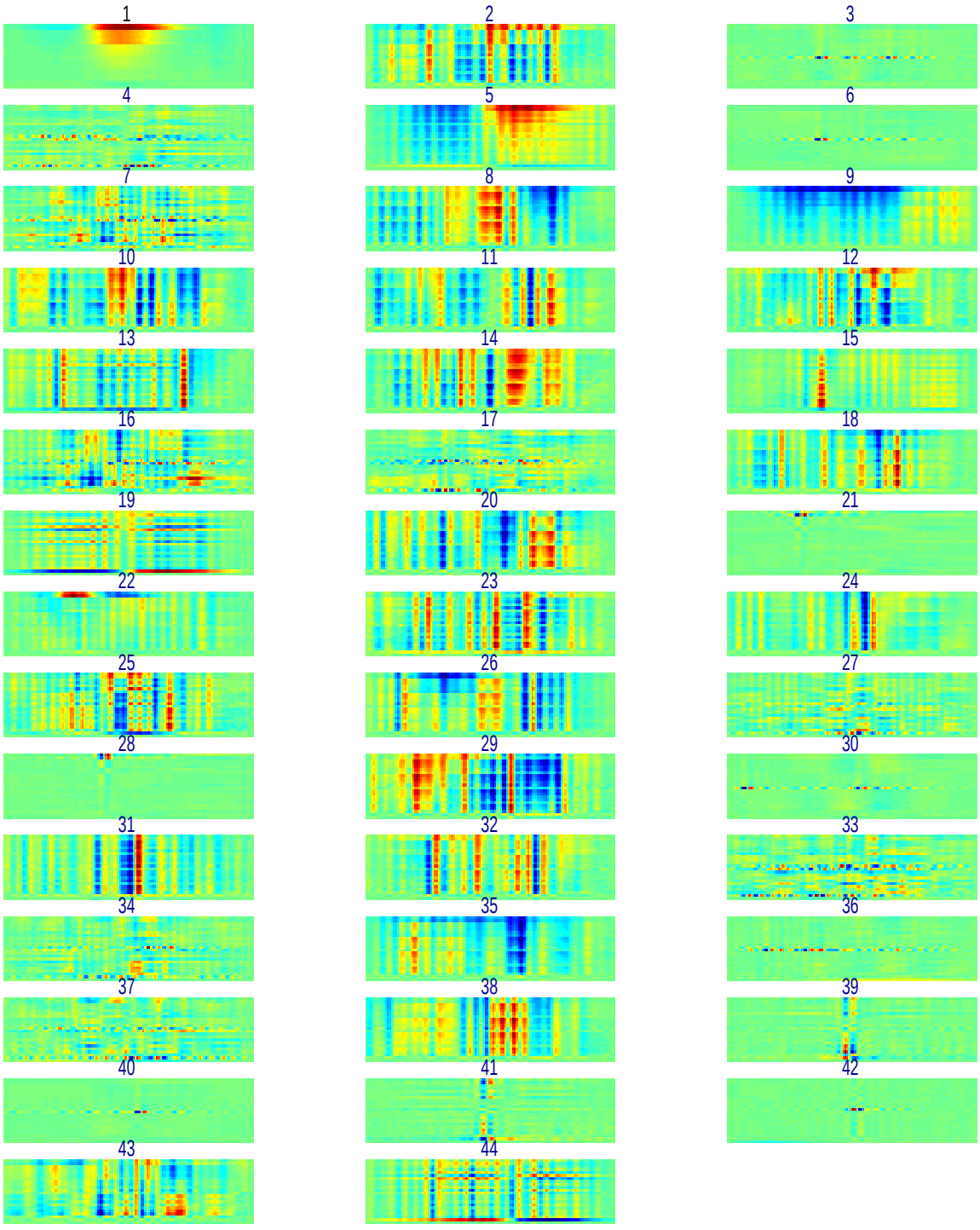


Figure 3.10. Learned EEG mother wavelet functions. Each function is 44x128 and depicted as a colormap (positive values are red, negative values are blue, green is 0)

method for inferring sparse decompositions in an overcomplete representation, and how to adapt a wavelet model to 1D signals using matching pursuit. By adapting the wavelet functions to a given class of data, the feature descriptors in the dictionary will be better suited for describing the data.

We have shown the flexibility of this approach for modeling different types of data. Wavelet models were adapted for natural audio signals, as well as for multi-channel EEG recordings. This demonstrates one of the advantages of this approach. The wavelets, and even the framework, can be customized to the data. For instance, a multi-resolution analysis may not be suitable for all types of data. For EEG, we adapted a convolution model, where wavelet functions were shifted but not rescaled, to represent the signal. Many signals have multiple channels, such as stereo encoded audio or hyper-spectral data, with complex dependencies between the channels. It may be unclear in such a situation what type of wavelet would be appropriate, or how best to describe these cross channel dependencies. The approach taken here, adapting wavelet functions which are multi-channel, provides a way to model these dependencies without pre-assumed knowledge of the structure present in the data.

The matching pursuit algorithm used here is a faster implementation, with complexity of $\mathcal{O}(\log N)$ for each iteration, than the standard matching pursuit algorithm introduced by Mallat and Zhang [22] which has complexity of $\mathcal{O}(N \log N)$ per iteration. To achieve this computational efficiency in a multi-resolution framework, the resolutions, or levels, were optimized independently. This simplification is not expected to greatly reduce the sparsity of the resulting representation. However, this assumption has not been rigorously tested. Additionally, a more detailed comparison between the solutions obtained using matching pursuit and those obtained by other methods, such as simulated annealing with Gibbs sampling, would be beneficial.

One topic for further research is to investigate stochastic versions of the matching pursuit algorithm. Matching pursuit is much more efficient at locating highly probable solution vectors, yet its deterministic behavior may not be ideal for some applications. For example, to adapt all of the parameters of the model it would be advantageous to sample from the posterior while avoiding the inefficient approach taken by a Gibbs sampler. It may also be possible to modify the matching pursuit algorithm to take into account dependencies between coefficients in order to move beyond the linear generative models shown here.

It is hoped that the methods shown here will provide a starting point for adapting wavelet models to many types of signals. The fast matching pursuit algorithm is easily extendable to higher dimensions, and the approaches taken here may provide a template for other data models. Each class of data presents its own special challenges. Having tools which can adapt to the structure of the data can provide an important advantage when selecting the best descriptive language to use for a given task.

Chapter 4

Statistical Methods for Information Hiding

This chapter demonstrates the application of statistical models to a form of information hiding known as steganography. Steganography, derived from the Greek words for 'covered writing', is the science of hiding information so that it remains undetected except by its intended receiver. It is necessary when one wishes to communicate privately without arousing the suspicion that would be caused by sending an encrypted message in plain view. The secret communication is hidden inside a larger message, referred to as the *cover message*, which can be transmitted without arousing any suspicion. The resulting message containing the hidden content is referred to as the *stego message* or *steganogram*. Here, we will assume any adversary is *passive*, meaning that we are concerned only whether an adversary can detect the presence of a hidden message by observing the stego message, and not whether he can disrupt communications by modifying the transmission before it reaches the intended recipient. Steganography and cryptography are complementary, and are generally used together to

achieve maximal security. While cryptography is used to conceal the meaning of a message, the goal of steganography is to conceal the message itself.

Many methods have been proposed for hiding messages in digital media including JPEG images, and MP3 audio files.[35, 26, 27, 36] Current methods generally encode the messages in the least significant bits (LSBs) of the cover media coefficients. While LSB encoding is often not detectable by visual inspection, it can alter the statistical properties of the coefficients in easily detectable ways.[36, 37] By altering the LSBs indiscriminately, the marginal statistics (histograms) of the coefficient values will be changed in ways that make steganographic tampering evident. By reducing the size of the message, these kinds of statistical signatures can be reduced. However, one would obviously prefer to use a steganography method that is secure despite having a large capacity, where capacity is defined as the ratio between the size of the message and the size of the cover data in which it is hidden.[36]

Recently, some methods have been devised which offer reasonably high capacity steganography while attempting to preserve the marginal statistics of the cover coefficients. One such method for encoding messages inside JPEG images is F5.[36] Rather than simply flipping LSBs to encode the message bits, F5 increments and decrements coefficient values, among other tricks, in order to maintain coefficient histograms that appear unaltered. However, it has been shown that F5 still changes the histograms of the coefficients in a detectable way. By estimating the original histograms of the coefficients from a cropped and re-JPEG'd version of the image, differences between the steganogram's histograms and the estimated original histograms become evident.[17] Another method that preserves marginal statistics more successfully is the OutGuess algorithm.[27] OutGuess reserves around half of the available coefficients for the purpose of correcting the

statistical deviations in the global coefficient histogram caused by changing LSBs in the other half. For example, if a coefficient's value was moved from histogram bin A to bin B during the encoding process, another coefficient has to be moved from bin B to bin A to correct this change. While this is effective at maintaining the global histogram of the coefficients, it reduces the capacity by about half.

This raises the following questions: Is it possible to avoid detection by attacks that rely on marginal statistics of the coefficients without sacrificing half of the message capacity? What is the maximum message size that can be embedded in a given cover medium without risking detection? How can we achieve this maximum capacity? For answers, we turn to a new methodology based on statistical modeling and information theory. Section 2 presents a general framework for performing steganography and steganalysis using a statistical model of the cover media. To demonstrate the value of the model-based approach, an example steganography method is proposed for JPEG images in section 3 that achieves a higher message capacity than previous methods while remaining secure against first order statistical attacks. We also present a method for defending against "blockiness" attacks, which have been used to successfully break the Outguess algorithm.[16] To illustrate how a model-based approach can be used for steganalysis, section 4 describes a method for estimating the length of messages hidden with the JPEG Jsteg steganography algorithm.

4.1 General methodology

4.1.1 Compression and steganography

Before describing the details of the model-based approach, it is helpful to first discuss the relationship between compression and steganography. This relationship has been previously discussed but it is useful to review it here.[3] Suppose we had a method for perfect compression of some cover media, such as images taken from the real world. Thus, we could feed our compressor random scenes from our world and it would convert them to perfectly compressed, truly random bit sequences (containing no statistical regularities) for each image. This is only possible if our compressor has access to a complete and perfect model of the statistical properties found in plausible cover images. Every statistical redundancy, every predictable quality, must be taken into account in order to accomplish this task - edges, contours, surfaces, lighting, common objects, even the likelihood of finding objects in certain locations.

We could, of course, place these compressed bit sequences in the corresponding decompressor to get back our original images. Suppose instead we had the idea to put our own random bit sequences into the decompressor. Out would come sensible images of the real world, sampled from the machine's perfect statistical model. Nothing would prevent us from also putting compressed and encrypted messages of our own choosing through the decompressor and obtaining for each message an image that should arouse no suspicion whatsoever were we to send it to someone. Assuming that our encryption method produces messages that appear random without the proper key, and that our intended receiver has the same image compressor we do, we will have perfectly secure steganography. Steganog-

raphy is considered perfectly secure if there is no statistical difference between the class of cover messages and the class of stego messages[5].

Admittedly, this is unhelpful in that we do not know how to make such a compression machine. However, let us now consider a different approach that uses the same concept of decompression for practical steganography without the necessity of having a perfect model of the cover media. Assume instead that we have a model that captures some, but not all, of the statistical properties of the cover media. We can use a similar paradigm to provide steganography that is undetectable by all except those possessing a superior model of the cover media, or more specifically, a model capturing statistical properties of the cover media that are not captured by our model. This is accomplished by applying this decompression paradigm with a parametric model to replace only a least significant portion of cover media that has been sampled from the real world.

The security of this steganography system will depend on the ability of the assumed model to accurately represent the distribution over cover messages. Specifically, such steganography will be ϵ-*secure against passive adversaries*, as defined by Cachin, where ϵ is the relative entropy between the assumed model and the true distribution over cover messages[5]. Thus, this model-based approach provides a principled means for obtaining steganography that is provably secure in the information theoretic sense, insofar as the model upon which it is based accurately captures the statistical properties of the cover media. As long as it remains possible that someone possesses a better model of the cover media, we cannot be sure that such steganography is completely undetectable. But if we consider a steganographic algorithm to be reasonably secure if it is not detectable by a specific statistical model, we can start to make some definitive statements regarding the maximum message length that can be securely hidden with this model and

give a working strategy for obtaining this capacity. This approach will hopefully shift the emphasis which has up to now been placed on embedding methods towards more principled steganography methods based on statistical models. That is, we can start asking how to best model our cover data rather than trying to anticipate specific attacks or invent clever ways to flip least significant bits. And we can ask whether a steganographic method embeds messages optimally given its assumed statistical model. This provides us with a unifying framework with which to view and improve steganography and steganalysis methods.

4.1.2 Method

Let x denote an instance of a class of potential cover media, such as JPEG compressed images transmitted via the internet. If we treat x as an instance of a random variable X, we can consider the probability distribution $P_X(x)$ over transmissions of this class of media. Thus, if we transmit signals drawn from P_X, we can be assured that they are indistinguishable from similar transmissions of the same class regardless of how many such signals we transmit. Since P_X represents data taken from the real world, we can draw a valid instance from P_X using a digital recording device. Given such a sample, x, we separate it into two distinct parts, x_α which remains unperturbed, and x_β which will be replaced with x'_β, our encoded message. For LSB encoding, x_α represents the most significant bits of the cover coefficients as well as any coefficients not selected to send the message, and x_β represents the least significant bits of the selected coefficients. We can consider these parts as instances of two dependent random variables X_α and X_β. Using our model distribution $\hat{P}_X$, we can then estimate the distribution over possible values for X_β conditioned on the current value for X_α: $\hat{P}_{X_\beta|X_\alpha}(X_\beta|X_\alpha = x_\alpha)$. Provided that we select x'_β so as to obey this conditional

distribution, the resulting $x' = (x_\alpha, x'_\beta)$ will be correctly distributed according to our model $\hat{P}_X$.

Now, in truth, it would appear that we haven't gained anything from this since we cannot model $P_{X_\beta|X_\alpha}$ perfectly any more than we could perfectly model P_X. However, we have gained something quite important. If we make a careful choice as to how X_α and X_β are separated, we can ensure that our changes to x_β are difficult or impossible to detect using the most sophisticated model of P_X on the planet: the human perceptual system. For instance, if we generate random samples from current image models, the result at best looks like 1/f noise or texture. But while the human visual system is fantastic at modeling images, it lacks a certain degree of precision. This lack of precision is what LSB encoding methods exploit. However, even the simplest models, such as those that capture the marginal statistics of X_β, do not lack this precision, and thus can be used to detect when LSBs are modified by some other distribution.

The solution proposed here is to use a parametric model of P_X to estimate $P_{X_\beta|X_\alpha}$, and then use this conditional distribution to select x'_β so that it conveys our intended message and is also distributed according to our estimate of $P_{X_\beta|X_\alpha}$. We can accomplish this task using the decompression paradigm previously discussed. Given a message M that is assumed to be compressed and encrypted so that it appears random, decompress M according to the model distribution $\hat{P}_{X_\beta|X_\alpha}$ using an entropy decoder, where x_α is part of an instance x drawn from the true distribution P_X via a digital recording device. While this cannot guarantee perfect security unless our model of P_X is perfect, it prevents all attacks except for those that use a better model of $P_{X_\beta|X_\alpha}$ than ours. Unless an attacker models statistical properties of X that we do not, or models them more accurately, our steganogram x' will contain the same measured statistical properties

as others drawn from the true distribution P_X.

Figure 4.1 illustrates the proposed model-based method for encoding steganography. First, an instance x of our class of cover media X is separated into x_α and x_β. x_α is fed to our model estimate of P_X which is used to compute the conditional probability distribution $P_{X_\beta|X_\alpha}$. The compressed and encrypted message M is given to an entropy decoder which uses $P_{X_\beta|X_\alpha}$ to decompress M resulting in a sample x'_β drawn from this distribution. The parts x_α and x'_β are then combined to form the steganogram x', distributed according to our model P_X which is transmitted to our receiver. Figure 4.2 illustrates the method used to recover the original message. Our steganogram x' is divided into x_α and x'_β. The x_α portion is fed into the model P_X which is again used to compute the condition

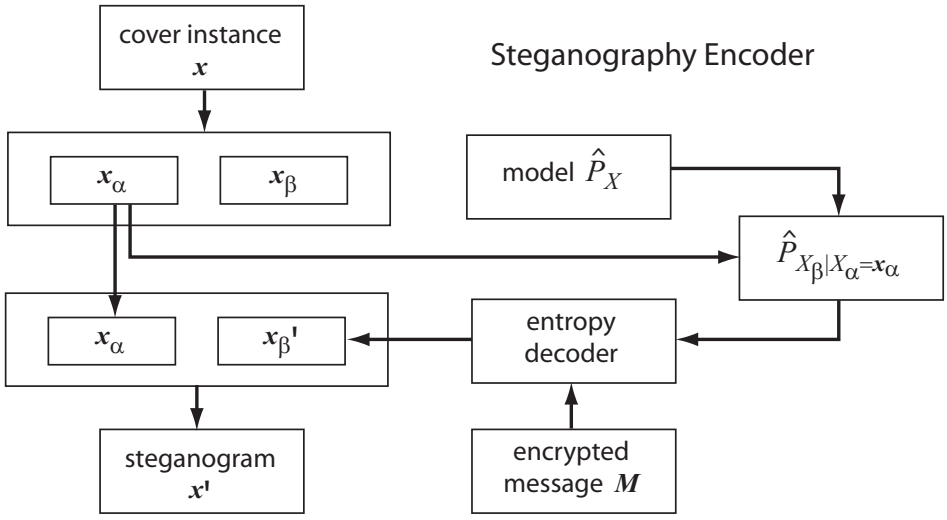


Figure 4.1. Model-based steganography encoder: A cover x, such as an image, is split into two parts x_α (e.g. MSBs) and x_β (e.g. LSBs). A parametric model $\hat{P}_X$ over possible instances X is used to calculate the distribution over possible x_β instances given x_α. These probabilities are passed to an entropy decoder to which decompresses the encrypted message M, creating x'_β which is combined with x_α to create the steganogram.

distribution $P_{X_\beta|X_\alpha}$. Thus, the same model is given to the entropy encoder that was fed into the entropy decoder during the encoding stage. The entropy decoder returns the encrypted message. Assuming we have a key, we can decrypt the message and verify its contents. If on the other hand, we do not have the key, the encrypted message will appear random, which is the same result we would get from decoding an instance of X that does not contain steganography. Note that the encryption key does not necessarily need to be a private key. If we use a public key encryption method, we can just as easily obtain a method for public key steganography as suggested by Anderson *et al.*[3]

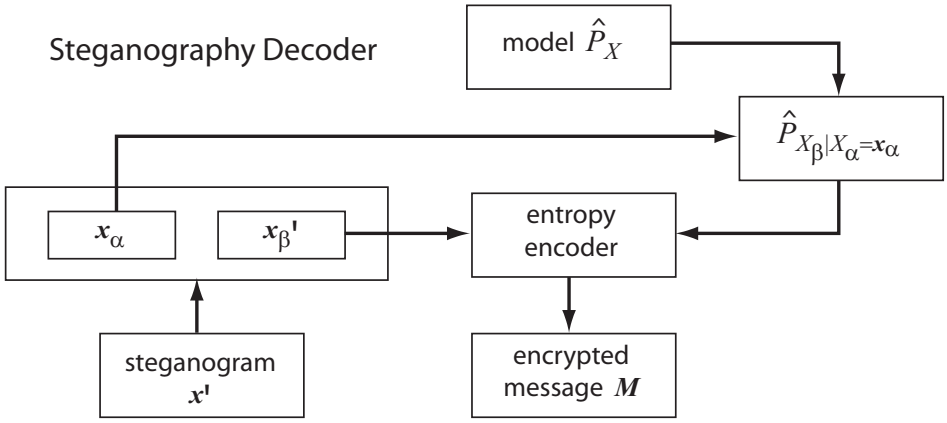


Figure 4.2. Model-based steganography decoder: A steganogram x' is split into parts x_α and x'_β. A parametric model $\hat{P}_X$ is used to calculate the same probability distribution over possible x_β sequences that was used in the encoding process. x'_β is then fed into the entropy encoder which uses these probabilities to return the original message M.

4.1.3 Capacity

Determining how large a message can be hidden inside a cover message without becoming detectable has been a long unanswered question. If what we mean by detectable is detectable by any method, this question remains unanswered as we would need to model P_X perfectly to ensure total security. However, we can estimate the average maximum message length that can be hidden without becoming detectable by our measured statistics of P_X. If we consider that X_β is being used as an information channel, we know from information theory that the maximum amount of information on average that can be transmitted through such a channel is equal to the entropy of the conditional distribution $\hat{P}_{X_\beta|X_\alpha}$:

$$H(X_\beta|X_\alpha = x_\alpha) = -\sum_{x_\beta} \hat{P}_{X_\beta|X_\alpha}(x_\beta|x_\alpha) \log_2 \hat{P}_{X_\beta|X_\alpha}(x_\beta|x_\alpha) \tag{4.1}$$

Using our model, this capacity limit can be measured for a given x_α. We can also see that our encoding method will be able to encode messages with this length on average, since an entropy encoder is designed to achieve this limit. Note that this limit is a function of x_α, and thus may vary depending on the content of our particular cover instance x.

4.1.4 Implicit models used by current methods

With this framework, we can gain a deeper understanding of current steganography methods. For instance, we can view certain methods as performing optimal model-based embedding using some implicitly assumed model over the cover media. Here we consider a class of generative models over cover messages, in which cover instances are assumed to be generated by a set of coefficients sampled from

some prior distribution and passed through an inverse transform (such as the blocked DCT basis). Methods which encode messages in the coefficient LSBs at a rate of one bit per coefficient can be viewed as equivalent to the model-based approach if one uses a model in which the coefficients of the cover media are statistically independent and have least significant bits which are uniformly distributed. Thus, these methods assume a model histogram over the coefficients in which pairs of histogram bins have equal probability. In this case x_β represents the LSBs of the coefficients, and the entropy decoder will simply copy bits of the message into the LSBs of the coefficients, equivalent to simple LSB embedding. Since this model is obviously incorrect, one can easily see that encoding at such a rate must result in a significant change to the marginal statistics of the coefficients, and we can see that they are changed to appear more likely to be drawn from the assumed model. While methods such as OutGuess attempt to compensate for this change by making compensatory changes to extra coefficients, we can see that this approach is not guaranteed to obtain maximum capacity, and is likely to reduce the capacity we could achieve if we incorporate the true marginal distributions of the coefficients into our model.

4.1.5 Steganalysis

We can also apply a model-based approach to the problem of steganalysis. The goal of steganalysis is to detect the presence of a hidden message within a suspected stego object. Current steganalysis methods already make use of statistical tests. However, many current methods for steganalysis do not employ an explicit statistical model of the cover data. Rather, they often measure a single statistic that serves as an indicator that the cover message has been altered. Given a good statistical model of the cover data, one should be able to apply it to

steganalysis as well. Provided that the model captures statistical properties of the data that are not preserved by the steganography method used, the likelihood of the stego messages under the model will serve as an indicator that they have been altered. The accuracy with which stego and non-stego images may be separated will depend on the similarity between the distributions over stego and non-stego objects, limited by the statistical properties included in the model.

In the case when there is a known separation between x_β and x_α for a target steganography method we wish to attack, and we have a model which can be conditioned on x_α, we can apply the framework described here directly by computing the negative log likelihood of x_β given x_α under our model for a target instance x: $-\log_2 \hat{P}_{X_\beta|X_\alpha}(X_\beta = x_\beta|X_\alpha = x_\alpha)$, which has an expected value equal to the entropy $H(X_\beta|X_\alpha = x_\alpha)$ (see eq. 4.1), our expected message length. Measuring the negative log likelihood of x_β is equivalent to running the steganographic decoding process already described on x and measuring the length of the resulting "message". The value should be larger than the expected message length, which can also be easily computed, to the degree that x_β violates the statistics of our model. Of course, this will only detect changes made by a steganography method that did not preserve statistical properties of the data that are captured by our model. It will be blind to optimally steganography embedded using the same model, or one more complete.

If we target a specific steganography method, we can go a step further by modeling the steganographic changes made by that method. From our model P_X, we may be able to generate a model of the stego objects P_{X_s} by altering our the model P_X to take into account changes introduced by the steganography method. Determining which class a particular instance x belongs in then amounts to measuring the *log-likelihood ratio*: $\log \frac{P_X(x)}{P_{X_s}(x)}$ and comparing this ratio with

some threshold. This is known as hypothesis testing.[5] In some cases, the hidden message length m can be added as an additional parameter to the model $P_{X_s|m}$ over stego objects. Thus, normal cover images are included as a special case where $m = 0$. The message length m may be estimated by maximizing the average log likelihood:

$$\hat{m} = \arg\max_{m} \left[\log P_{X_s|m}(x|m) \right] \tag{4.2}$$

An example of this approach is illustrated for detecting Jsteg steganography in section 4.3.

4.2 Application to JPEG steganography

In order to demonstrate how the model-based methodology works in practice, we will now describe an example steganography system that is applied to compressed images stored in the file format defined by the Joint Photographic Experts Group (JPEG). Although JPEG is undoubtedly not the best compression format available, it is chosen for this demonstration because of its abundant use in email transmissions and on public internet sites. While the discussions from this point on will be aimed specifically at this JPEG implementation, the method used here can be easily applied to other file formats. In the JPEG compression standard, images are broken into 8x8 blocks. Each pixel block is passed through a 2-dimensional DCT (Discrete Cosine Transform) to produce 64 DCT coefficients for each block. Compression is accomplished by quantizing these DCT coefficients and then encoding them using a Huffman (or other entropy) encoder. The amount of compression is determined by the quantizer step size used before the entropy encoding, which is lossless.

The method described here is not intended to be secure against any known at-

tack, but rather is primarily intended to demonstrate the methodology described in the previous section. We will use a fairly simple model which captures only the marginal statistics of the quantized DCT coefficients. Our total image model, then, assumes that images are generated by statistically independent DCT coefficients. While this takes into account some correlations between image pixels, it is still a very limited image model as it does not describe higher order dependencies or even correlations across 8x8 blocks. It is expected that more complete image models which take into account joint statistics of the DCT coefficients would provide better steganographic security, and could also be used to attack this method.

An example of such an attack is described by Farid and Lyu,[14] who detect steganographic content by examining the marginal statistics of wavelet coefficients. Since the wavelet basis is much better than the DCT basis at describing the structure found in images, this would describe certain dependencies present between DCT coefficients. Taking into account joint statistics while encoding a message into DCT coefficients appears difficult, however, and so to some degree we are limited in our steganographic security by the image model imposed by our choice of cover media. If these methods were applied to a wavelet compression format such as JPEG 2000 instead of JPEG, however, it would provide resistance to attacks which use marginal statistics of wavelet coefficients.

4.2.1 Model

As with many steganographic methods, we will modify the least significant portions of the coefficients to encode our hidden information. Our model will consist of a parametric description of the marginal DCT coefficient densities. Be-

cause the DC coefficients (which represent the mean luminance within a block) are not well characterized by a parametric model, and because modifications to these coefficients are more likely to result in perceptible blocking artifacts, we will use only the AC (Alternating Current) coefficients during the encoding. Zero valued coefficients are also skipped for the encoding, because these often occur in featureless areas of the image where changes are most likely create visible artifacts. The remaining AC coefficients are modeled using the following parametric density function, which is a specific form of a Generalized Cauchy distribution.

$$P(u) = \frac{p-1}{2s}(|u/s| + 1)^{-p} \tag{4.3}$$

where u is the coefficient value and $p > 1$, $s > 0$. The more general form for Generalized Cauchy distribution also raises $|u/s|$ to an exponent. Here we assume this inner exponent is 1. The corresponding cumulative density function is

$$D(u) \;=\; \begin{cases} \frac{1}{2}(1 + |u/s|)^{1-p} & \text{if} \quad u \leq 0, \\ 1 - \frac{1}{2}(1 + |u/s|)^{1-p} & \text{if} \quad u \geq 0 \end{cases} \tag{4.4}$$

Other probability distributions, such as the generalized Laplacian distribution,[30] have also been used to describe coefficient histograms that are peaked at zero. The distribution used here was chosen because it appeared to provide a better fit to the AC coefficient histograms, particularly in the tails of the distribution, and also because there is a closed form solution for its cumulative density function. This allows more precise fitting of the distribution to coefficient histograms and provides an efficient means of computing the probabilities for each histogram bin.

The first step in the embedding algorithm is to compute low precision histograms (with bin size > 1) of *each type* of AC coefficient for a cover image x. We will call the bin size of the low precision histogram the embedding step size. Each coefficient value is represented by a histogram bin index and a symbol which indi-

cates its offset within the bin. If the embedding step size is 2, for instance, there will be two possible offsets within each nonzero bin. The zero bin is restricted to a width of 1 because we are skipping zero valued coefficients. The bin indices for all the coefficients comprise x_α, which will remain unchanged, and the bin offsets will comprise x_β which will be changed to encode our message.

For each image and each type of coefficient, model parameters s and p are fit to the low precision histograms we have computed. The distributions are fit to only the most significant information in the coefficients because it is critical that both the encoder and the decoder compute the same estimated probabilities. The steganography decoder cannot know the least significant portions of the original coefficients as these may have been altered by the encoder. We fit the model parameters s and p to a histogram $\mathbf{h}$ of coefficients $\mathbf{x}$ by maximizing the log likelihood for the coefficients under the model:

$$\begin{aligned} \hat{p}, \hat{s} &= \arg\max_{p,s} \left[\log P(\mathbf{x}|\mathbf{p}, \mathbf{s})\right] && (4.5) \\ &= \arg\max_{p,s} \left[\log \prod_i P(x_i|p, s)\right] && (4.6) \\ &= \arg\max_{p,s} \left[\sum_j h_j \log \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} P(u|p, s)\ du\right] && (4.7) \end{aligned}$$

where h_j is the number of coefficients in $\mathbf{x}$ with values between $(j - \frac{1}{2})\Delta x$ and $(j + \frac{1}{2})\Delta x$.

During embedding, the coefficients are altered only within the low precision histogram bins (only the bin offsets are changed) so that the same estimates for p and s for each coefficient type may be obtained by the decoder. Figure 4.3 shows the histogram of the (2,2) DCT coefficients for a sample image measured in log probability and the model density after being fit to the histogram using the maximum likelihood approach.

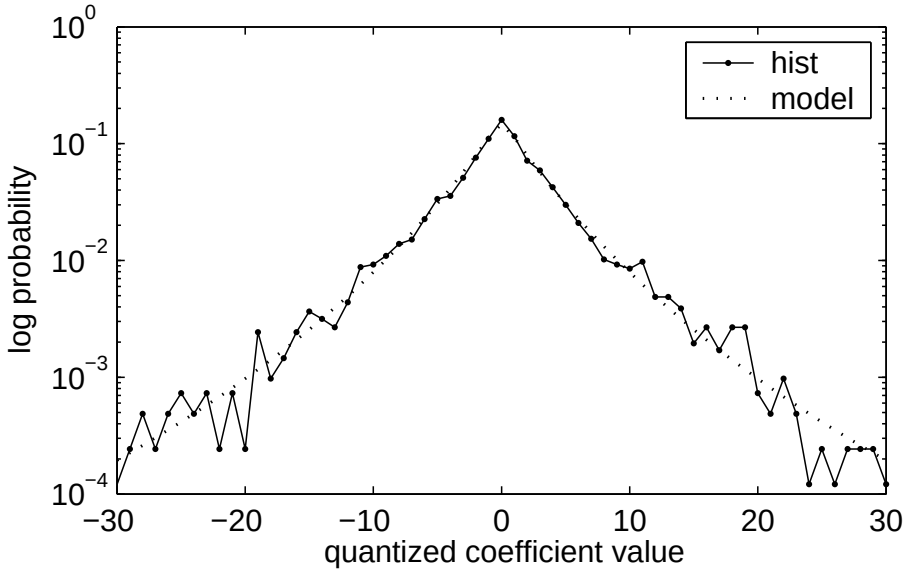


Figure 4.3. Measured histogram (in log probability) of DCT coefficient (2,2) for the goldhill image, and the model pdf with parameters $s = 18.28$, $p = 6.92$.

4.2.2 Embedding method MB1

Once the model is fit to the histograms for an image, it is used to compute the probability of each possible offset symbol for a coefficient given its bin index. These offset symbols, and their respective probabilities are passed to a non-adaptive arithmetic entropy decoder[10] along with the message we wish to embed in the cover image. The offset symbols returned by the entropy decoder comprise x'_β which are combined with the bin indices to compute the coefficient values of the steganogram x'. To avoid visual attacks caused by changing coefficients only in part of the image, the order in which coefficients are used for encoding the message is determined by computing a pseudo-random permutation seeded by a key. This technique is known as permutative straddling.[36] If during the process of embedding, we use all of the available coefficients before we have exhausted our message bits, we have exceeded our maximum message length

for this image. In order to anticipate when this will happen, we can obtain the average maximum message length by computing the entropy of our symbol frequencies. If the message is shorter than the maximum message length, any remaining symbols are assigned according to the original coefficient offsets so that these coefficients remain unchanged. We will refer to this algorithm as MB1, which stands for Model-Based embedding.

A similar process is used to decode the message from the steganogram, except that the bin offset symbols x'_β in the steganogram are passed along with the symbol probabilities to an arithmetic encoder. Assuming the message length is encoded into the message, we can stop once the end of the message is reached. The algorithms for embedding and retrieving the message are outlined below:

Outline of embedding algorithm MB1

1. Given a cover image in JPEG format, and an encrypted message, generate low precision (bin size > 1) histograms of coefficient values. This information comprises x_α.

2. Fit the p and s parameters of our parametric model to each histogram by maximum likelihood.

3. Assign symbols to represent the offset of each coefficient within its respective histogram bin. These symbols comprise x_β. Compute the probability of each possible symbol for each coefficient using the model cdf.

4. Choose a pseudo-random permutation to determine the ordering of the coefficients.

5. Pass the message, and the symbol probabilities computed in step 3 in the

order specified by step 4 to a non-adaptive arithmetic decoder in order to obtain symbols specifying the new bin offsets for each coefficient. The resulting symbols comprise x'_β.

6. Compute the new coefficients from the histogram bin indices (x_α) of the symbol offsets (x'_β).

Outline of the decoding algorithm

1-4. Same as embedding algorithm steps 1-4.

5. Pass the symbols and symbol frequencies obtained in steps 1-4 to the non-adaptive arithmetic encoder to obtain the original message.

Embedding step sizes

An embedding step size of 2 roughly corresponds to LSB encoding since each nonzero AC coefficient can take on one of two new values. Larger embedding step sizes will increase the message capacity (still without altering the marginal statistics) by sacrificing some image quality. During embedding, coefficients will be modified by a value of at most one less than the embedding step size. These modifications will result in artifacts with the appearance of JPEG quantization noise. If the cover media is not already highly quantized, a higher step size can be used before image quality becomes noticeably diminished. This provides a convenient means of increasing message capacity. However, transmitting images that are not very highly compressed may arouse suspicion in some situations.

Arithmetic encoding

The model-based approach described here requires an entropy codec. We modified a non-adaptive arithmetic encoding method published by Witten *et al.*[38] to accept frequencies passed for each symbol rather than estimating them adaptively. For another example of non-adaptive entropy coding see [4], or refer to [10, 38] for details on arithmetic encoding methods.

Embedding efficiency

One way to demonstrate the effectiveness of the model-based approach is to calculate the embedding efficiency. Embedding efficiency is the average number of message bits embedded per change to the coefficients .[36] It is generally assumed that the more changes that are made to the coefficients, the easier on average it will be to detect the steganography. Thus, we would like to minimize the number of these changes for a particular message length. In the model-based approach, the embedding efficiency will be determined by the entropy of the symbol distributions. Let us assume for now that we are using an embedding step size of 2, since that is most comparable to other steganography methods. We can show that the embedding efficiency of MB1 will always achieve an embedding efficiency greater than or equal to 2, regardless of the message length.

Let k represent the probability of one of the two offset symbols for a given coefficient. The average number of bits we will encode, or the embedding rate, is equal to the entropy of the channel: $H = -(k \log_2 k + (1-k)\log_2(1-k))$. The probability that the value of the coefficient will be changed by encoding a different symbol than the original one, the rate of change, is $k(1-k)+(1-k)k = 2k(1-k)$.

The expected embedding efficiency is the ratio of these two rates,

$$E[\texttt{efficiency}] = \frac{-(k \log_2 k + (1-k)\log_2(1-k))}{2k(1-k)} \tag{4.8}$$

which is never smaller than 2 for $0 < k < 1$. This function is plotted for values of k between 0 and 1 in figure 4.4. If $k = \frac{1}{2}$, the embedding efficiency will be exactly 2 because we will encode our message at a rate of 1 bit per coefficient and will be changing a coefficient from its original state half of the time. Otherwise, the embedding efficiency will always be greater than 2 and will be the largest (and the capacity the smallest) when the symbol probabilities are furthest apart.

The F5 algorithm uses a technique known as matrix encoding to obtain an arbitrarily high embedding efficiency by reducing the message capacity. However, for its maximum message capacity which is around 13%, the embedding efficiency is only 1.5.[36] The OutGuess algorithm, since it must change about two coefficients on average for every other bit it embeds, has an embedding efficiency close

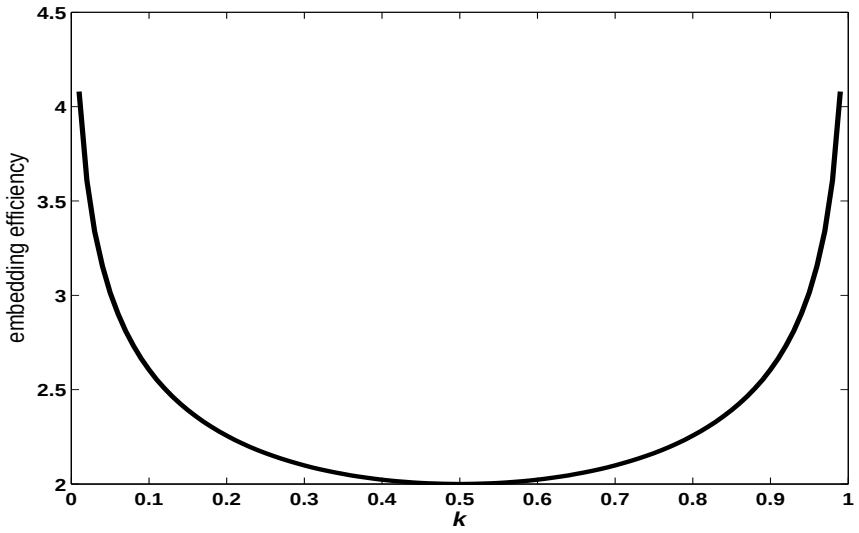


Figure 4.4. Efficiency of MB1 plotted as a function of k, where k is the probability for a given x_β symbol. Efficiency is always larger than 2 and is the largest when the symbol probabilities are the furthest apart.

to 1. In practice, we found that OutGuess provided an embedding efficiency of about 0.96 and a maximum message capacity of about 6.5%.

4.2.3 Defending against blockiness attacks: Method MB2

Recently Fridrich *et al.* showed that JPEG steganography methods are vulnerable to attacks which measure "blockiness".[16] By using a simple measure of the discontinuities between adjacent 8x8 JPEG blocks, they showed how to estimate the number of coefficients that had been altered. They used the following blockiness measure:

$$B = \sum_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum_{j=1}^{N} |g_{8i,j} - g_{8i+1,j}| + \sum_{j=1}^{\lfloor (N-1)/8 \rfloor} \sum_{i=1}^{M} |g_{i,8j} - g_{i,8j+1}| \tag{4.9}$$

where g_{ij} are pixel values in an MxN grayscale image and $\lfloor x \rfloor$ denotes the integer part of x.

They found that blockiness B appears to increase linearly with the number of DCT coefficients that have been changed. By comparing blockiness measurements before and after adding steganography to a suspected image, and using similar measurements made to a cropped and recompressed version of the same image, a reasonable message length estimate can be obtained. This method was used to successfully break the OutGuess algorithm,[16] but we found it to be just as effective at attacking our method. This is not unexpected, since the blockiness measure represents a form of joint dependencies between neighboring coefficients that is not described by our model. This is a shortcoming of the blocked DCT basis used by JPEG, which uses hard edged functions to describe images, and would not be an issue if applying the model-based approach to other types of carrier media.

Ideally, our statistical model should be improved to capture blockiness statistics in addition to the marginal statistics of the coefficients. This may be possible by means of a joint model which takes into account coefficient values from neighboring blocks. This is a problem we will explore in future research. For now, we present a less optimal but more direct approach of countering the blockiness attack which we will refer to as MB2. For method MB2, a message is embedded in the same manner as MB1, but at least half of the coefficients are reserved for the purpose of reducing blockiness artifacts. Coefficients not used to encode the message are then adjusted within the limits imposed by the embedding step size in order to reduce the blockiness to the amount present in the original image. Note that this will significantly decrease the message capacity and also greatly reduce the embedding efficiency. To make this adjustment, we first compute a change Δg_{ij} to each image pixel that will decrease the blockiness measure:

$$\Delta g_{ij} = \Delta r_{ij} + \Delta c_{ij} \tag{4.10}$$

where

$$\Delta r_{ij} = \begin{cases} \frac{1}{2}(g_{i+1,j} - g_{i,j}) & \text{if } \; i = 8\lfloor i/8 \rfloor, \; 1 < i < 8M \\ \frac{1}{2}(g_{i-1,j} - g_{i,j}) & \text{if } \; i-1 = 8\lfloor i/8 \rfloor, \; 1 < i < 8M \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

and

$$\Delta c_{ij} = \begin{cases} \frac{1}{2}(g_{i,j+1} - g_{i,j}) & \text{if } \; j = 8\lfloor j/8 \rfloor, \; 1 < j < 8N \\ \frac{1}{2}(g_{i,j-1} - g_{i,j}) & \text{if } \; j-1 = 8\lfloor j/8 \rfloor, \; 1 < j < 8N \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

This change to the pixels is then passed through the block DCT transform to compute an adjustment to the coefficients which were not used during the embedding process. The coefficients are only adjusted within their low precision histogram bins (limited by the embedding step size) so that the parameter

estimates and conditional probabilities computed during decoding will not be affected. After calculating the blockiness that would result if this maximal adjustment were applied to all non-message coefficients, we compute a linear estimate of the number of coefficients that should be changed to achieve the desired blockiness and change a random subset of the coefficients accordingly. This procedure is repeated a few times until the desired blockiness is obtained. This procedure was tested on several grayscale images. When the blockiness attack was applied to the deblocked steganograms, the returned message length estimates were always zero. The marginal statistics of the coefficients do not appear to be significantly altered by the "deblocking" process. This was confirmed by decoding the deblocked steganograms with random keys, which returned messages having lengths that remain close to the entropy limit expected by the model. Additional analysis is necessary in order to verify the security of this method, but it is believed that MB2 is significantly harder to detect than all currently published JPEG steganography methods including F5 and Outguess.

Image name	File size (bytes)	Message size (bytes)	Capacity	Embedding Efficiency
barb	48,459	6,573	13.56%	2.06
boat	41,192	5,185	12.59%	2.03
bridge	55,698	7,022	12.61%	2.07
goldhill	48,169	6,607	13.72%	2.11
lena	37,678	4,707	12.49%	2.16
mandrill	78,316	10,902	13.92%	2.07

Table 4.1. Results from embedding maximal length messages with MB1 into several 512x512 grayscale JPEG images with an embedding step size of 2. Files were compressed using a JPEG quality factor of 80 and optimized Huffman tables.

4.2.4 Results

Table 1 gives the results obtained from encoding maximal length messages in several grayscale test images using the proposed model-based method MB1. While we tested our method on grayscale images, nothing prevents its application to color images. The images were first compressed to a JPEG quality factor of 80. This method does not double compress, which would leave a detectable signature on the coefficient histograms.[16] Instead the least significant bits of the coefficients are simply replaced, so the result steganogram maintains the same quantization tables as the original. The steganogram file size, message length and embedding efficiency for an embedding step size of 2 are shown for each image. Figure 4.5 shows the coefficient histograms of the DCT coefficient (2,2) before and after different steganography methods have been applied to the goldhill image. As can be seen, F5 (version 11+) greatly increases the number of zeros while the model-based methods MB1 and MB2 retain the shape of the original histogram. Note that they aren't expected to be identical to the original, since we are sampling from a model to select our coefficient values. The maximum allowable message length was used for each method during the comparison.

4.3 Application to JPEG steganalysis

This section demonstrates an application of the model-based approach for the detection of Jsteg steganography for JPEG images, a method proposed by Derek Upham.[35] Jsteg embeds messages in the least significant bits of DCT coefficients, skipping coefficients with values of 0 or 1. In this section, we focus on estimating the hidden message length m. We will consider only the case where

the DCT coefficients are selected for embedding in a random sequence. Attacks for Jsteg have been previously described in the literature.[37, 39] We approach the problem here not for the sake of improving upon those results, but rather in order to describe a model-based approach to steganalysis. This approach may be of more general importance for detecting other types of steganography, or may be improved by using a better statistical model. Here, we apply the same model used in the previous section for steganographic embedding, but now for the purpose of detecting Jsteg steganography and estimating the hidden message length. Note that this method cannot be used to detect the steganographic method described in the last section without using an improved model, since the statistics captured by this model are preserved by the model-based steganography method.

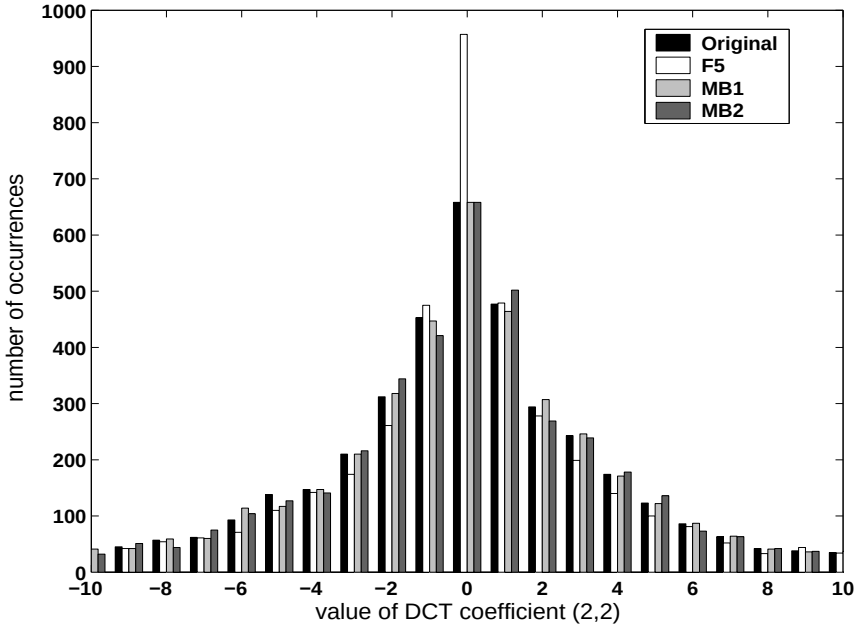


Figure 4.5. Histogram of DCT coefficient (2,2) for goldhill image, and after embedding with F5 (4984 bytes), MB1 (6544 bytes) and MB2 (3250 bytes).

4.3.1 Method

We fit the two parameter form of the Generalized Cauchy distribution to the histogram of the AC coefficients in the JPEG DCT coefficients (eq 4.3). We make two changes to the model, however. First we apply the model to the global coefficient histogram which includes all AC coefficient types rather than the individual histograms. Secondly, we model stego images by augmenting the model with a parameter β which is equal to the relative number of modified coefficients not equal to 0 or 1. We know $0 \leq \beta \leq .5$, since at most only half of the coefficients will be changed by Jsteg in order to embed a maximal length message. From β, and the coefficient histogram of the stego image, an estimate of the hidden message length may be easily computed.

Let $q(j|p,s)$ denote the model probability for an AC DCT coefficient (after quantization) with absolute value equal to j in the original JPEG image before Jsteg was applied, which can be computed from equations 4.3 and 4.4:

$$q(j|p,s) = \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} P(u \mid p,s)\, du \tag{4.13}$$

$$= \frac{1}{2}\left(1+\frac{j-\frac{1}{2}}{s}\right)^{1-p} - \frac{1}{2}\left(1+\frac{j+\frac{1}{2}}{s}\right)^{1-p} \tag{4.14}$$

where p, s are model parameters which will be fit to the global coefficient histogram for each image.

Jsteg sets the least significant bit of each coefficient to be equal to the next bit of the message to be embedded. If the coefficient is a 0 or a 1, the coefficient is skipped and embedding resumes with the next available coefficient. In general, a coefficient with value $j \notin \{0,1\}$ may be changed to a value $\bar{j} = j - 2(j \bmod 2) + 1$, depending on the next message bit to encode.

Let $\tilde{q}(j|p,s,\beta)$ denote the model probabilty for a coefficient with absolute

value equal to j after Jsteg is applied:

$$\tilde{q}(j|p,s,\beta) = \begin{cases} q(j|p,s) & \text{if } j \in \{0,1\} \\ (1-\beta)q(j|p,s) + \beta q(\bar{j}|p,s) & \text{otherwise} \end{cases} \tag{4.15}$$

Given a histogram $\mathbf{h}$ for a suspected stego image, where h_j denotes the number of occurrences of an AC DCT coefficient with value equal to j in the suspected stego image, model parameters estimates $\hat{p}, \hat{s}$, and $\hat{\beta}$ are chosen to maximize the log likelihood for the measured histogram under our Jsteg model. Assuming independence for the coefficients,

$$\hat{p}, \hat{s}, \hat{\beta} = \arg\max_{p,s,\beta} \left[\log P(\mathbf{h}|\mathbf{p}, \mathbf{s}, \beta)\right] \tag{4.16}$$

$$= \arg\max_{p,s,\beta} \left[\sum_j h_j \log \tilde{q}(j|p,s,\beta)\right] \tag{4.17}$$

A local maximum for this objective may be obtained using a standard optimization procedure. In the following tests, the downhill simplex method of Nelder and Mead was used.[23] While this method is not guaranteed to provide the global optimum, this did not appear to be a problem in our tests.

An estimate $\hat{m}$ for the hidden message length may be computed from $\hat{\beta}$ as:

$$\hat{m} = 2\ \hat{\beta} N - h_0 - h_1 \tag{4.18}$$

where N is the total number of AC coefficients available to Jsteg and h_0 and h_1 represent the number of AC coefficients with values of 0 and 1, respectively. This assumes that the message has been compressed or encrypted before embedding with Jsteg so that message bits are distributed uniformly between 0 and 1.

While this procedure produced results with similar accuracy to those of Zhang and Ping[39], we found there to be a significant amount of variability from one

image to the next in the $\hat{\beta}$ obtained for an image before steganography was added. When different message lengths were added to the same image, however, the resulting $\hat{\beta}$ estimates varied as a linear function of the true β with very little additional variability. Thus, it seemed beneficial to add an additional calibration step. Using a technique described by Fridrich *et. al* for estimating the statistics of the pre-stego image[16, 17], stego images were cropped by 4 pixels in each dimension and recompressed with the same JPEG quality factor. Using the original stego image, the recompressed stego image, and an image generated by re-embedding the stego image with Jsteg, three β estimates were obtained as follows:

1 Estimate β for the stego image using the maximum likelihood procedure described above. Call this estimate β_0

2 Estimate β after cropping and recompressing. Call this estimate β_1.

3 Re-embed the stego image with a maximum length Jsteg message. Estimate β for this image, and call the estimate β_2.

4 Compute the calibrated estimate $\hat{\beta} = (\beta_0 - \beta_1)/(\beta_2 - \beta_1)$

4.3.2 Results

Steps 1-4 outlined above were tested on a database of 22 grayscale 512x512 JPEG images with a quality factor of 80. For each image, Jsteg was applied with the relative number of coefficient modifications β selected at random between 0 and .5. Each image was independently tested 20 times, with a random true β value selected each time, for a total of 440 stego images. Table 4.3 shows the proportion of images for which the error in the message length estimation is less

than specific tolerance values for all of the stego images tested. The actual and estimated message lengths and relative number of coefficients changed for some of the stego images are shown in Table 4.2. Calibration steps 1-4 were used for these results.

Filename	β	$\hat{\beta}$	m	$\hat{m}$
alex	0.328	0.336	27234	27551
bear	0.046	0.052	4148	4597
butterflies	0.451	0.464	38776	39715
cactus	0.183	0.201	12112	13257
cedars	0.249	0.245	31324	30651
chilis	0.482	0.488	53628	54137
corn	0.194	0.190	18844	18305
cows	0.115	0.117	7248	7199
elephant	0.235	0.232	28178	27756
fish	0.113	0.114	9406	9247
flowers	0.343	0.342	40156	40124
flowers2	0.297	0.295	27202	27201
house	0.070	0.077	11640	12559
lava	0.307	0.310	49476	50215
leaning	0.309	0.303	37962	37214
maples	0.470	0.473	62846	63082
muir	0.032	0.033	4482	4598
pond	0.432	0.423	37384	37119
shells	0.257	0.255	29334	29015
sitting	0.136	0.142	21162	21880
squash	0.095	0.088	7380	6845
visby	0.329	0.331	41160	41426

Table 4.2. Results for 22 of the 440 tested stego images. Shown are the relative number of modifications β, the corresponding message length m, and their estimated values: $\hat{\beta}$ and $\hat{m}$.

Figure 4.7 plots the estimated $\hat{\beta}$ parameter against the actual β parameter for all stego images as a scatter plot for estimations obtained with and without calibrating using the recompressed and reembedded images. When no calibration was used, more accurate results were obtained using only a subset of the coeffi-

cient types having a mean coefficient value greater than .1. This improved the baseline estimate for each image, but increased the variability of the estimates within each image. When calibrating using steps 1-4, best results were obtained using all of the AC coefficient types. Figure 4.6 shows the histogram of the errors $\beta - \hat{\beta}$ of all stego images when using calibration. The variance of the errors was 5.934e-005 (stdev = .0077).

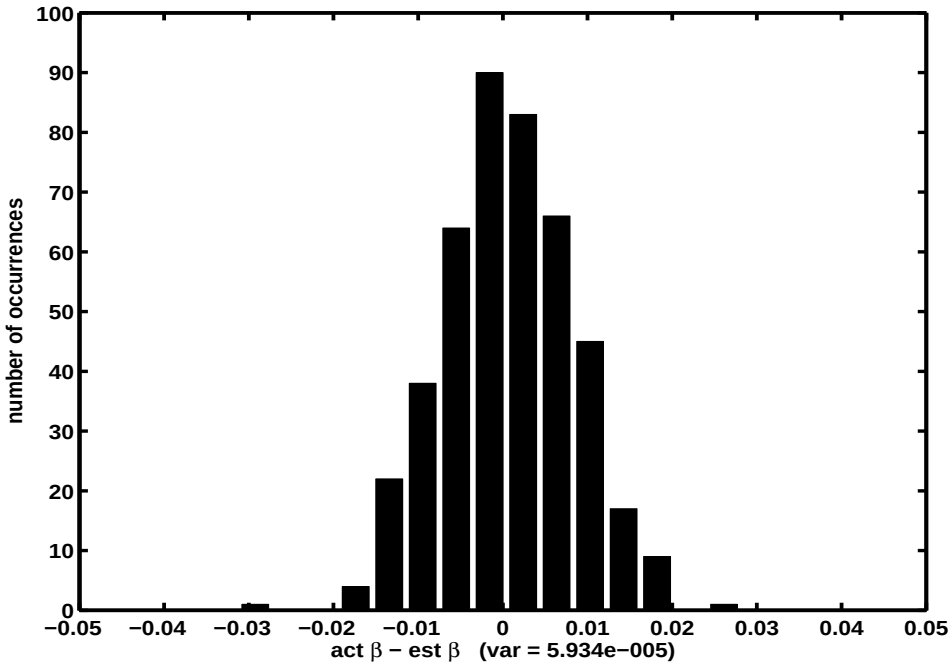


Figure 4.6. Histogram of estimation errors $\beta - \hat{\beta}$ (using calibration).

4.4 Discussion

We have presented a new model-based approach to steganography which uses a parametric model of the cover messages in order to embed maximum length messages while avoiding detection by a given set of statistics. This methodology provides a unified framework for understanding current steganography methods

Tol	% of capacity	% of images
.020	4.0%	99.6%
.018	3.6%	98.0%
.016	3.2%	96.4%
.014	2.8%	94.1%
.012	2.4%	88.4%
.010	2.0%	80.5%

Table 4.3. Percent of images with message length estimation errors $\beta - \hat{\beta}$ <Tol, for Tol = .02, .018, .016, .014, .012, and .010. Equivalent percentage of total hiding capacity is also shown.

and may also be applied to steganalysis. We have demonstrated how to apply the model-based methodology to JPEG images using a model which captures marginal statistics of the DCT coefficients. The resulting steganography method MB1 achieves higher embedding efficiency than current methods while maximizing message capacity, and is resistant to first order statistical attacks. For example, MB1 can embed twice as long a message as OutGuess while changing fewer coefficients, and unlike OutGuess maintains not only global coefficient histograms but individual coefficient histograms as well. Recently, it was shown that steganography methods which use JPEG as a cover media are subject to attack because they predictably increase the amount of "blockiness" artifacts as a function of the number of coefficients that are changed to encode the message.[16] We described a method called MB2 for "deblocking" stego images which successfully defeats this attack. A model-based approach was also described and evaluated for estimating the hidden message length for JPEG stego images embedded with Jsteg. This method demonstrates an approach for using a parametric model of cover objects for steganalysis.

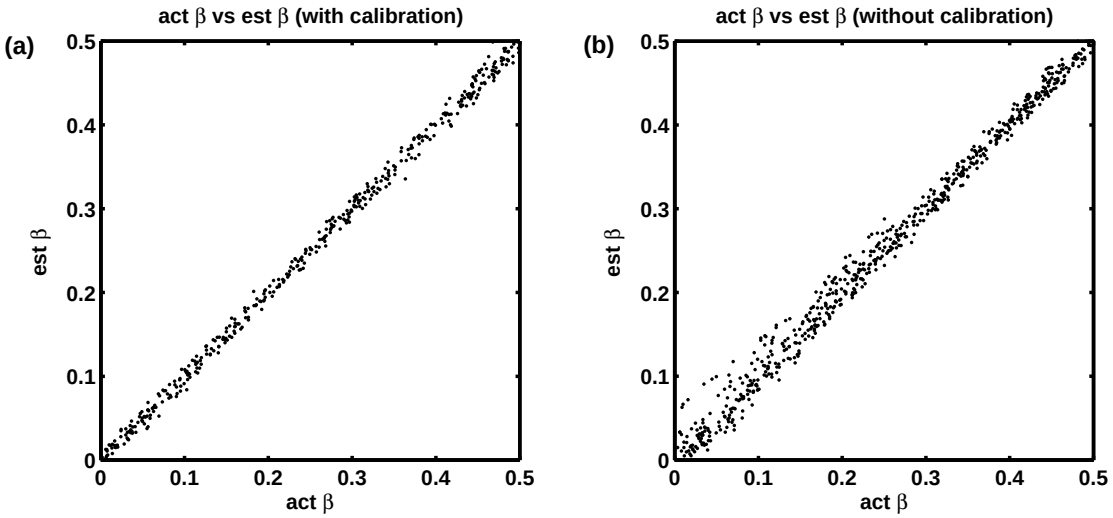


Figure 4.7. Actual relative number of changes β plotted against the estimated relative number of changes $\hat{\beta}$. Separate plots are shown for (a) with calibration using a cropped and recompressed image and reembedding (b) without calibration, using a single maximum likeihood estimation for each stego image.

Chapter 5

Conclusions

This dissertation discussed methods for adapting statistical models to images, sounds and other types of signals and how such statistical models may be applied to various applications. In this chapter, we summarize the work presented and discuss its implications for future work.

5.1 Summary

We presented methods for adapting overcomplete wavelet representations to natural signals by incorporating the wavelet basis into a generative statistical model with a sparse prior over the wavelet coefficients. First, we considered a mixture prior over the coefficients consisting of a Gaussian and a delta function, in order to model the coefficients for representations which are highly overcomplete. The wavelet basis functions are parameterized by a set of "mother wavelet" functions. These functions are adapted to maximize the average log-likelihood of the model for a large database of natural images. When adapted to natural images, these functions become selective to different spatial orientations, and

they achieve a superior degree of sparsity on natural images as compared with traditional wavelet bases. The learned basis is similar to the Steerable Pyramid basis, and yields slightly higher SNR for the same number of active coefficients. To infer the coefficients in such a representation, a Gibbs sampler was presented. The Gibbs sampler provides a means of sampling from the posterior distribution, or to obtain a MAP estimate by simulated annealing. A denoising method was demonstrated by averaging a number of samples drawn from the posterior, and the results compare favorably with wavelet coring methods for denoising.

In chapter 3, the statistical modeling approach was applied to 1-D signals including natural sounds and electroencephalograph (EEG) recordings. The methods presented demonstrate how to apply statistical learning methods for adapting overcomplete representations to 1-D signals, including signals which have multiple channels and signals which are not well suited to a multi-scale representation. In doing so, we demonstrate that the ideas presented here are applicable to a wide variety of data. In particular, the ability to adapt a model in an unsupervised fashion in the form of an overcomplete dictionary of feature vectors, is ideal for situations in which the structure of the data is not known in advance. We also described a faster approach for performing inference using a greedy algorithm known as matching pursuit. Our implementation runs in $\mathcal{O}(N \log N)$ time for the entire decomposition for a signal of length N, compared to the $\mathcal{O}(N^2 \log N)$ complexity of previous implementations.

In chapter 4, we discussed how to apply statistical modeling approaches for hiding and detecting the presence of hidden information in media. Methods were presented for performing steganography and steganalysis using a statistical model. Steganography is the science of hiding information in such a way that it cannot be detected without a cryptographic "key". Steganalysis is the art of

detecting hidden messages. The model-based approach we presented provides a means for obtaining steganography that is provably secure in the information theoretic sense, insofar as the statistical model upon which it is based accurately captures the statistical properties of the cover message. Using the model-based methodology, an example steganography method was proposed for JPEG images which achieves a higher embedding efficiency and message capacity than previous methods, while remaining secure against first order statistical attacks. Some general methods for applying statistical models to steganalysis were discussed and a method was demonstrated for detecting Jsteg steganography in JPEG images with a statistical model.

5.2 Implications for future work

There are a number of directions in which this work can be expanded. First, the models demonstrated here fall into the class of linear generative models. It is clear that natural signals are not generated by linear processes. While linear models provide a first step for exploring statistical inference methods, improved models are needed to accurately describe the statistical dependencies of the data. Describing these dependencies presents an ongoing challenge for future work. A major difficulty with generative models is inferring the representation from the data. The Gibbs sampling method presented in this dissertation provides a means of sampling from the posterior distribution for highly overcomplete representations, which is necessary for properly adapting all of the parameters of the model without special renormalization methods. However, Gibbs sampling is not practical for many applications due to its computational complexity. The manner in which the solution space is explored is highly inefficient and is often susceptible to

local minima. These problems were more significant when applying the learning method to natural sounds. The matching pursuit algorithm, provides a much more efficient way to explore the solution space. However, due to its greedy nature, it may also be unsuitable in some situations. We demonstrated that it can be used to adapt basis functions within the sparse coding framework, yet biases in the posterior estimates make it difficult to adapt all of the parameters of the model. One area of future research is to consider a form of matching pursuit which is stochastic. That is, rather than seeking the "best" feature vector to update at each step, a feature vector could be chosen from an appropriate distribution resulting in an improved markov sampler which explores the solution space more effectively than the standard Gibbs sampler.

This work has implications for future research in models of natural scenes and other types of signals. The learned basis functions reveal features which are localized in position and spatial frequency, and are spatially oriented. The learned functions appear similar to those of the Steerable pyramid, yet there were differences shown. When certain constraints were released, it became evident that the learned functions tend to span more than one octave in bandwidth, and have a amplitude spectrums that are similar to the $1/f$ spectrum of natural images. One avenue of investigation is to modify the Steerable pyramid basis in a way that mimics the learned functions in order to see how this may contribute to improved coding efficiency of the representation. Also, by releasing additional constraints the tiling properties of the learned basis functions can shed more light on the statistics of natural images.

The preliminary work shown for EEG data indicate that there are many statistical dependencies across time as well as channels that can be captured with a linear generative model. Future work is needed in order to determine which

of these dependencies are limited to events related to the experimental protocol used to collect the data, and which are due to general brain activity. Also, it would be interesting to determine how well the learned features generalize to multiple subjects. The shiftable functions learned for EEG present an interesting challenge to identify the components and correlate them with real-world events. This type of analysis may provide a tool which is useful for future neuroscience research in order to better understand some of the underlying mechanisms of the brain. The method shown here may also be used to denoise EEG signals, or to ore accurately remove unwanted artifacts such as eye-blinks.

Finally, the work presented for steganography and steganalysis has significant implications for the field of information hiding. In order to hide information in such a way that it cannot be detected by statistical tests, it is necessary to first model those statistics and then to properly match the statistics while embedding the message. The method presented here allows this to be done to maximum capacity, while perfectly preserving a measured set of statistics. There are some limitations, however. While the general prescription is clear, the entropy encoding method demonstrated here works only for representations in which the components are assumed to be independent. More work is needed to show how this may be accomplished when the components being modified have known inter-dependencies. One problem in applying advanced statistical models to the problem of information hiding, is that one is limited to some degree by the compression format used to transmit the data. For example, for JPEG images one is limited by the quantization performed in the DCT domain. If one wishes to apply a better image model than the DCT transform implicitly provides, one must consider the effects of the DCT quantization which will may destroy any information hidden in another domain. Another area for future work

is to investigate other methods for applying statistical models to detecting hidden information. For example, it may be possible to use advanced statistical models, and inference methods, to steganalysis by attempting to infer the original image through a "denoising" process, thus treating the hidden content as a form of noise. This is a slightly different approach than the statistical methods discussed in chapter 4 for steganalysis.

Dealing with real-world data is a difficult problem. While wavelets are a useful tool for representing such data, we have only begun to scratch the surface of describing the complex statistical dependencies that exist in images, sounds and other real-world signals. We know that it is possible to infer the causes of these signals, because our brains serve as working examples. Understanding how this process works, however, is a difficult challenge. Statistical models provide a framework for addressing this problem, and we can gain valuable insight from considering the problem in terms of statistical inference. New inference methods and new types of statistical models will need to be developed. It is hoped that the work done here will provide a starting point for developing improved models of natural signals and new inference methods for use with these models.

Bibliography

[1] Echoes of Nature CD (boxed set). Laserlight Music, 1993.

[2] F. Abramovich, T. Sapatinas, and B.W. Silverman. Wavelet thresholding via a Bayesian approach. *J of the Royal Statistical Society: Ser B*, 60(4):725–749, 1998.

[3] R. J. Anderson and F. A. P. Petitcolas. On the limits of steganography. *IEEE Journal of Selected Areas in Communications*, 16(4):474–481, May 1998. Special issue on copyright & privacy protection.

[4] R. W. Buccigrossi and E. P. Simoncelli. Progressive wavelet image coding based on a conditional probability model. In *Proceedings ICASSP-97 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2597–2600, Munich, Germany, 1997.

[5] C. Cachin. An information-theoretic model for steganography. *Lecture Notes in Computer Science*, 1525:306–318, 1998.

[6] S. Grace Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Transactions on Image Processing*, 9(9):1522–1531, 2000.

[7] H. Chipman, E. Kolaczyk, and R. McCulloch. Adaptive bayesian wavelet shrinkage. *J. Am. Stat. Assoc.*, 92(440):1413–1421, 1997.

[8] R. Coifman and V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38, March 1992.

[9] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*. Springer-Verlag, Berlin, Germany, 1995.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[11] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Signal Proc.*, 46(4):886–902, 1998.

[12] I. Daubechies. Time frequency localization operators: a gemotric phase space approach. *IEEE Transactions on Information Theory*, 34:605–612, 1998.

[13] D. L. Donoho and A. G. Flesia. Can recent innovations in harmonic analysis 'explain' key findings in natural image statistics? *Network: Comput. Neural Syst.*, 12(3):371–393, 2001.

[14] H. Farid and S. Lyu. Detecting hidden messages using higher-order statistics and support vector machines. In F.A.P. Petitcolas, editor, *Information Hiding: Fifth International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 340–354, Berlin Heidelberg, Germany, 2003. Springer-Verlag.

[15] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[16] J. Fridrich, M. Goljan, and D. Hogea. Attacking the outguess. In *ACM Special Session on Multimedia Security and Watermarking*, Juan-les-Pins, France, 2002.

[17] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In F.A.P. Petitcolas, editor, *Information Hiding: Fifth International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 310–323, Berlin Heidelberg, Germany, 2003. Springer-Verlag.

[18] R. Gray. Vector quantization. *IEEE ASSP Magazine*, April 1984.

[19] P. Greenspun. Online image database, used with permission. `http://www.photo.net`.

[20] M. S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002.

[21] M. S. Lewicki and B. A. Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America*, 16(7), 1999.

[22] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[23] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[24] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[25] B. A. Olshausen and K. J. Millman. Learning sparse codes with a mixture-of-gaussians prior. In K.R. Muller S.A. Solla, T.K. Leen, editor, *Advances in Neural Information Processing Systems*, volume 12, pages 841–847. MIT Press, 2000.

[26] F. A. P. Petitcolas. MP3Stego, 1998. `http://www.cl.cam.ac.uk/~fapp2/steganography/mp3stego`.

[27] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–336, Washington, DC, 2001.

[28] D. Chen S. Qian. Signal representation using adaptive normalized gaussian functions. *Signal Processing*, 36:1–11, 1994.

[29] M. A. Saunders S.D. Chen, D.L. Donoho. Atomic decomposition by basis pursuit. *Siam J. Sci. Comput.*, 20(1):33–61, 1998.

[30] E. P. Simoncelli and E. H. Adelson. Noise removal via Bayesian wavelet coring. In *3rd Int'l Conf on Image Processing*, volume I, pages 379–382, Lausanne, 1996. IEEE Sig Proc Society.

[31] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *2nd Int'l Conf on Image Proc*. IEEE Sig Proc Society, 1995.

[32] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, 1992.

[33] V. Strela, J. Portilla, and E. Simoncelli. Image denoising using a local Gaussian scale mixture model in the wavelet domain. In *Proc. SPIE 45th Annual Meeting*, 2000.

[34] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Comput. Neural Syst.*, 14:391–412, 2003.

[35] D. Upham. Jpeg-jsteg-v4. `http://www.funet.fi/pub/crypt/steganography/`.

[36] A. Westfeld. High capacity despite better steganalysis (f5 - a steganographic algorithm). In I.S. Moskowitz, editor, *Information Hiding: Fourth International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302, Berlin Heidelberg, Germany, 2001. Springer-Verlag.

[37] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding: Third International Workshop*, volume 1768 of *Lecture Notes in Computer Science*, pages 61–75, Berlin Heidelberg, Germany, 2000. Springer-Verlag.

[38] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

[39] T. Zhang and X. Ping. A fast and effective steganalytic technique against Jsteg-like algorithms. In *ACM Symposium on Applied Computing*, Melbourne, Florida, USA, March 2003.