Representing and modeling images with multiscale local orientation

by

David K. Hammond

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Mathematics
New York University
September 2007

Eero P. Simoncelli – Advisor

Dedicated to the memory of

my grandfather
George S. Hammond

Acknowledgements

Firstly, I would like to thank my adviser, Eero Simoncelli, whose imagination and passion for science is inspiring, and who has taught me a lot. Working with Eero over the past few years has been a very good experience.

I would like to thank some friends I was fortunate to get to know during my time at the Courant Institute: Chris König, Arjun Raj, Andrea Barreiro, Rob Meyers, Barney Bramham, Tyler Neylon, Fred Laliberté. I would like to thank my friends and colleagues from the Laboratory for Computational Vision : Umesh Rajashekar, Alan Stocker, Zhou Wang, Cynthia Rudin, Rosa Figueras, Jonathan Pillow, James Hedges, Peggy Series and Martin Raphan.

I would like to thank David Hansel for advising me as a summer student in both France and Israel, and Jon Rinzel for putting me in contact with him.

I would like to thank Dan Tranchina for making the time for some non technical academic career and general life discussions.

I would like to thank Mehryar Mohri for many valuable discussions about the machine learning topics that ended up in this thesis.

I would like to thank my parents and my sister.

And finally I would like to thank Jeong-Yin Janet Choi, for all her love and support, and for taking most of the photographs that appear in this thesis.

Abstract

Digital photographs are not random collections of pixels, but have strong structural and statistical regularity. Understanding the properties of natural image signals allows the development of better algorithms for image processing applications. One important property of natural images is the presence of strongly oriented features. In this thesis, I explore using measures of the local image orientation for representing and modeling images.

I develop a novel nonlinear image representation based on multiscale local orientation measurements. Specifically, an image is first decomposed using a two-orientation Steerable Pyramid, a multiscale wavelet type transform where the basis functions are derivative operators. Transforming these multiscale image gradients into polar coordinates partitions the image data into local magnitudes and local orientations. I show that it is possible to reconstruct the original image from only the orientation measurements. An algorithm for reconstructing the original image is developed based on projection onto convex sets. Additionally, I demonstrate the robustness of the representation to quantization of the orientation measurements.

Following, I describe a pair of statistical models for images that explicitly capture variations in local orientation and contrast. The first model describes patches of image coefficients as samples of a fixed Gaussian process that are rotated and scaled by hidden variables controlling the local contrast and orientation. The second introduces an additional hidden variable that mediates adaptation to the orientedness of the local signal. I develop optimal Bayesian least squares error estimators for these models that function by conditioning upon and integrating over the hidden variables. The resulting denoising procedures give results that are visually superior to those obtained with a Gaussian scale mixture model that does not explicitly incorporate local orientation.

An alternate method for constructing a spatially adaptive denoising method by combining two distinct local denoising methods is explored using machine learning methodology. Interpolation between the two methods is controlled by a spatially varying decision function that may be learned from example data. I use weighted kernel ridge regression to solve this learning problem for the Gaussian scale mixture and the orientation adapted Gaussain scale mixture methods described above, yielding an improved performance "hybrid" denoiser.

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

The term “Digital Image Processing” encompasses a large number of techniques for compressing, restoring and otherwise manipulating images that are stored numerically. Images may arise from familiar photographic sources such as a digital camera, or from a multitude of other sensor modalities such as infrared or radar images, medical MRI images or astronomical data. Images from a particular type of source have strong underlying structural and statistical regularity, and understanding these properties is important for designing effective image processing methods.

Images must be represented numerically before they can be stored or processed on a digital computer. Many image processing tasks can be viewed as mathematical operations on the numerical coefficients that describe an image in a certain fixed representation. Both the ease of expression and the performance of image processing algorithms may depend strongly on the particular representation used. The design of appropriate and effective image representations is often motivated by understanding of underlying image structure. Conversely, however, descriptions of image statistical properties are often expressed in terms of distributions of coefficients in some fixed representation. There are thus intimate connections between the design of image processing algorithms, image representation and underlying image structural and statistical properties.

The space of all possible digital images of fixed resolution is astronomically large. For example, the set of 300 x 500 greyscale images with 8 bit pixel values between 0 and 255 has 256^{150000} elements. Only a fraction of these possible images would look anything like a photographic image. For algorithms that operate on photographic images, good performance is desired for the types of images that one will encounter in practice, not for completely arbitrary patterns of pixel intensities. In order to take advantage of the special structure of the space of photographic images, one must be able to describe what this space is. It is common to think of the set of "natural images", images that could arise as pictures of scenes in the world, as a distinct subset of the set of total possible images. This concept is somewhat vague; though there have been attempts to more precisely define what "natural" means in this context, it nonetheless is widely used in the literature [60, 23]. Loosely speaking, any method for describing this subset of images may be called a "natural image model". Two common types of models are function space models and statistical models. Function space models typically describe images as sets of functions from R^2 to R that satisfy certain regularity constraints such as piecewise continuity, piecewise C^α or bounded variation [13]. Such models are often employed by the harmonic analysis and function approximation community.

Statistical models seek to place a probability distribution on the set of possible images such that more "natural" images are assigned higher probability.

Within such a framework, many image processing tasks may be interpreted as statistical estimation problems. For example, restoration of an image with missing data can be thought of as estimating the "most likely" image consistent with the known information. In the image denoising problem, one seeks to recover a clean image that has been corrupted by noise. Intuitively, this problem may be solved with an image probability model where one would take the corrupted image and attempt to make it "more natural". If one can assume or estimate a statistical description of the noise process, one may use Bayesian inference methods to approach this in a more principled way.

The earliest models of image statistics, originally developed by television engineers in the 1950's, were Gaussian power spectral models. These models treat images as samples from a multivariate Gaussian density where the form of the covariance matrix is constrained by assuming structural properties of natural images. The two key assumptions are translation invariance, often called stationarity, and scale-invariance. Translation invariance asserts that the statistical dependencies between pixel values in the image depend only on their relative displacements, and not on their absolute positions in the image. This is essentially equivalent to asserting that two images that are simply translated copies of each other occur with the same probability. Scale invariance may be stated analogously, that two images related to each other via scaling by a single parameter occur with equal probability. These assumptions are reasonable to expect of a set of images taken by a camera at all possible viewpoints of a three dimensional scene : all possible translations and scalings will be generated as the camera moves throughout the scene. These two structural assumptions are sufficient to determine the form of the model covariance. Translation invariance implies that the covariance matrix will be diagonalized by the Fourier transform, so that the model may be specified by the variance of each Fourier coefficient. The magnitudes of the Fourier transform of a random process are commonly referred to as the power spectrum of the process, which gives these models their name. Scale invariance imposes a strong constraint on the Fourier coefficient variances. It can be shown (see [52]) that the only power spectral densities which are scale invariant are those where the variance of each Fourier coefficient follows a power law, i.e.

$$|\hat{f}(\vec{w})| \propto \frac{1}{|\vec{w}|^p}$$

where $\hat{f}(\vec{w})$ is the variance of the Fourier coefficient corresponding to the spatial frequency $\vec{w}$. This power-law decay of Fourier coefficient magnitudes of natural images has been examined empirically; values for p between 1.8 and 2 have been reported in the literature [62, 46].

Understanding of the properties of natural images is helpful for designing useful image representations. The most straightforward way of representing an image is as a rectangular array of pixel values. In this way one can think of an $m \times n$ resolution greyscale image as a point in the high dimensional linear space $R^{m\times n}$. Any invertible linear transformation of this linear space affords a different representation of the original image data. For many image processing tasks it is more convenient to manipulate the image data in the space of transform coefficients. The Discrete Fourier Transform, introduced above in the context of power spectral models, is a classic example of such a “linear image representation”. The basis functions of the Discrete Fourier Transform are global, with their support covering the entire image plane. However, a salient feature of natural images is the presence of highly localized image features such as edges and corners. This is problematic for applications such as image coding where one typically seeks to represent an image with as few nonzero coefficients as possible. The Fourier basis is able to represent local features with basis functions of global extent only by careful superposition and cancellation, which requires a large number of nonzero coefficients. Thresholding in such a basis leads to undesirable artifacts such as Gibbs oscillations.

One method of avoiding basis functions with large support is by dividing the image into smaller blocks and decomposing each block with the Fourier transform. This method is employed by the JPEG compression standard which uses the related Discrete Cosine Transformation applied to 8x8 blocks [39]. Image processing using block-based transforms often suffers from artifacts introduced by the block boundaries, as may be easily seen in a highly compressed JPEG image. Another problem is that the introduction of homogenous blocks fixes a single spatial scale for describing image content. Images evince statistical and structural regularities across multiple scales, however, and block based methods are unable to capture or take advantage of such regularity.

Representing image content uniformly across multiple scales requires the use of basis functions with varying sizes. An early example of such a “multiscale representation” is the Laplacian pyramid, formed by recursively encoding the difference between the image and a lowpass filtered copy [7]. If the lowpass filter is designed such that the smoothed copy may be subsampled without loss of information, repeated cascading of this process results in an exact pyramid representation where each level of the pyramid corresponds to image detail at a particular scale. Similar ideas are involved in the design of orthogonal wavelet decompositions developed in the 1980's and 1990's, which represent the signal as a sum of basis functions that are scaled and translated copies of a small number of "mother wavelet" functions [12, 32, 34]. The ability to represent image content simultaneously at multiple scales permitted the design of algorithms which take advantage of the cross-scale statistical regularities present in natural images.

Wavelet transforms may be viewed as linear maps that operate by taking inner products of the original image against the wavelet basis functions. By grouping the coefficients that are arise from translates of the same basis function, one can partition them into what are known as wavelet subbands. These subbands are essentially filtered and downsampled copies of the original image, and thus preserve some of the original spatial structure of the image. One can plot the coefficients of a wavelet subband and clearly see the presence of features from the original image. This may be contrasted with looking at a map of the Fourier coefficients of an image, which will never show any spatial structure. This property of wavelet coefficient subbands is important to recognize, as it indicates that there are statistical interdependencies between nearby coefficients that may be important to model.

The wavelet coefficients of natural images display a number of striking statistical features that are distinct from coefficients of random signals such as white noise. As the wavelet basis functions are localized, the coefficient magnitudes form a measure of local signal power. Natural images typically have sparsely distributed wavelet coefficients, where most of the coefficients are very close to zero and much of the signal information is contained in a small number of large magnitude coefficients. This is certainly not the case in the original pixel representation, where the signal information is spread evenly through all of the pixel values. The Fourier representation does perform some signal compaction, as more of the energy is present in the lower spatial frequencies. However, the distribution of magnitudes of the Fourier coefficients are not especially sparse. The sparsity of image wavelet coefficients is very important for image compression applications. A large volume of recent research in image coding is based on using wavelet expansions, including the JPEG 2000 standard intended as a successor to JPEG [58].

The sparsity property of image wavelet coefficients can also be seen by examining their marginal statistics. The marginal distributions of these coefficients for natural images are symmetric around zero, as the wavelet filters are zero mean. They have large peaks near zero and "heavy tails", when in compared Gaussian distributions. These wavelet marginal distributions are well fit by so-called generalized Gaussian distributions of the form

$$p(x) \propto e^{-|\frac{x}{s}|^{\alpha}} \tag{1.1}$$

Values for α near 0.7 are typically observed. The tails of these distributions decay slower than Gaussian, which is given by the expression when $\alpha = 2$. Note that these distributions are really characteristic of natural image signals. The marginal statistics of wavelet coefficients of white noise images, for example, are necessarily Gaussian.

While the marginal description of wavelet coefficients is informative, it completely disregards the spatial interactions of neighboring coefficients. These spatial regularities are important, and can be taken advantage of for modeling and processing images. One very striking property is that large coefficients tend to occur in clusters, i.e. nearby in scale and space to other large coefficients. This behavior arises as images contain localized features. The clusters of large magnitude coefficients occur as the support of several nearby coefficients will overlap with each local image feature, yielding large coefficient values. This regularity is exploited in zero-tree coding developed by Shapiro for image compression. [51].

A key property of natural images that distinguishes them from random noise is their spatial inhomogeneity. Intuitively speaking, natural images may have very different local content in different spatial regions. A single image may consist of smooth regions such as open sky or blank walls, strongly oriented edge regions formed by object boundaries, and textured regions that may or may not be oriented. One explicit measure of the local inhomogeneity in images is the local signal power. This may be measured by taking the average of the magnitudes of wavelet coefficients over a small neighborhood. The clustering of high magnitude coefficients in localized regions implies that the local signal power varies greatly across the image. Smooth regions will have low local power, where edge and texture regions have higher power.

Another crucial feature of natural images is the presence of strongly oriented local features. Often these arise from edges formed by the boundaries between objects. Some types of texture regions, such the pattern formed by stems of standing grass, may also be strongly oriented. Oriented features can occur at all different angles throughout the image. It is possible to measure the orientation at each location in an image by measuring the image gradient. The angle of the gradient vector defines the local orientation of the image at each point in space. For most images, this local orientation is not constant but changes throughout the image. This variation in local orientation is another source of local signal inhomogeneity.

The assumption that image features are equally likely to occur at all orientations is equivalent to stating that the ensemble of natural images are rotation invariant. This rotation invariance assumption has important implications for image representation. Orthogonal wavelets, while consistent with scale invariance, do not respect this rotation invariance as their basis functions are not formed from rotated copies of a single function. Simoncelli and Freeman developed an overcomplete, multiscale representation called the Steerable Pyramid which is consistent with both scale, translation and rotation invariance [54, 53]. The basis functions used in the Steerable Pyramid are directional derivative operators. The basis functions at each scale are not only rotated copies of each, but generate rotationally invariant subspaces. Any filter rotated by an arbitrary angle may be written as a linear combination of the K derivative filters at the same scale and location in space. The Steerable Pyramid thus provides a good fundamental set of tools for measuring and manipulating local image orientation.

Knowing the local orientation at a particular location in an image provides significant information about the local signal content. In this thesis, I apply this concept to both deterministic representation of images, and to stochastic image modeling. For deterministic representation, the underlying question is how much information is encoded in the local orientation. This work begins with decomposing the image with the Steerable Pyramid representation with

2 orientation bands. The basis functions of this transform are a set of first order derivative operators in the x and y directions at multiple spatial scales. These coefficients thus form a representation of the image gradient. Transforming these multiscale gradient vectors into polar coordinates effectively splits the image information into magnitude and orientation subbands at each scale. Performing this separation raises the natural question of how much information is carried by the orientation bands. I show that even if the gradient information is discarded, it is possible to reconstruct the original image from the orientation bands. This is interesting as it provides a nonlinear image representation in terms of a purely geometrical quantity, the local orientation. The reconstruction algorithm is shown to perform projection onto convex sets, which provides a proof that the algorithm converges. I also study the stability of this representation to quantization of the local orientation measurements, and show that the reconstruction quality decays gracefully with increasingly coarse quantization.

In the second half of the thesis, I use the local orientation as a tool for constructing stochastic image models that are appropriately adapted the local signal content. These models describe small patches of wavelet coefficients, and thus able to model local behavior and capture some of the dependencies between nearby coefficients. Natural images are inhomogeneous, exhibiting significant changes in local signal properties across space. This implies that stochastic image models should be able to adapt their description to the current local structure of the image. The models developed in this thesis are constructed by using a set of spatially varying hidden variables that capture the essential variations in local signal properties. They have the property that conditioned on these local hidden variables, the signal description is a multivariate zero mean Gaussian. The complete model is then realized as a Gaussian mixture, where the covariance of each mixture component is parameterized by the hidden variables.

Two of the most important aspects of this inhomogeneity are variations in local signal power and local orientation. The first model developed uses a pair of hidden variables (z, θ) that model the local power and local orientation. Under this model, each patch is described as a sample from a single Gaussian process that is then multiplied by $\sqrt{z}$ and *rotated* by θ. This model is an extension of the Gaussian Scale Mixture developed by Wainwright and Simoncelli [63] that includes only the z hidden variable. As the novel model includes adaptation by θ, it is called the Orientation Adapted Gaussian Scale Mixture (OAGSM) model.

One shortcoming of the OAGSM is that it describes all signal locations as oriented. Some portions of images, such as textured areas and junction regions or corners, are not well described by the oriented signal model. This issue is addressed by including an additional hidden variable that models whether or not the current image signal is strongly oriented. This yields the OAGSM with non-oriented component (OAGSM/NC) model, that is able to adapt to the local signal power, orientation and orientedness.

As both a practical application of these models and a test of their descriptive power, they are used for image denoising. Using the OAGSM and OAGSM/NC as signal priors, I develop a Bayesian Least Squares optimal estimator for removing additive Gaussian noise from images. The resulting denoising algorithms show improvement in both visual quality and mean squared error over a similar algorithm based on the original Gaussian Scale Mixture model.

Chapter 2

Deterministic Representation of Images

When we look at something, the optics of our cornea and lens gather light from the outside world and focus it onto the retina at the back of the eye. The subjective experience of "seeing" involves the subsequent processing and interpretation by the brain of this spatial pattern of light intensities. A photograph is an object which captures and later recreates the perception of viewing a particular scene. An image may thus be described as a two dimensional pattern of intensity values, that when transformed by the optics of the eye gives a pattern of retinal intensity values that mimic those arising from viewing an actual scene. For an image without color, we can represent the intensity at each location by a single real number. In this way one can think of an image as a real valued function defined on some two dimensional domain D. Typically D will be a rectangle.

Before an image can be stored or manipulated by a digital computer, the continuous intensity values must be sampled at discrete spatial locations and quantized. This spatial sampling is typically done on a regular rectangular lattice. This gives the “pixel representation” of an image, where the image intensity is specified over each cell, or pixel, of a regular rectangular grid. In a digital camera, for example, this discretization of the image is performed by an array of light sensors. In order to display the image at its full resolution, we need to know the intensity values for each pixel. The pixel representation may be considered the canonical way of representing a digital image.

For many image processing or image analysis applications, however, the original pixel representation is not the most natural or convenient representation to work with. For image manipulation, the fundamental image attributes that one wishes to alter may be related in a very complicated way to the original pixel values. Likewise for image analysis, the underlying patterns in the image signal that one is interested in may be difficult to detect directly from the pixel values. However, the underlying patterns or attributes that one wishes to measure or manipulate may become more apparent after some transformation of the original pixel data. Different computations may be easier to perform in different transform domains. For example, the power spectral density of an image is complicated to express directly in terms of the pixel values but is easily computed from the Fourier coefficients of the image. Conversely, the image dynamic range is simple to compute in the pixel domain but difficult to compute directly in the Fourier domain.

Taking an appropriate transform of the pixel data can make it easier to express certain image processing or analysis tasks. For image analysis problems such as object recognition or image classification, it may not matter if the transform is invertible. For image processing applications, however, as the ultimate output of any algorithm will be another image, it is crucial that the transform used be invertible. In this case we say the transform gives a representation of the image. Another way of saying this is that an image representation is a set of measurements from which one may reconstruct the original image or an approximation of the original image.

Given such a representation, one may manipulate images by first transforming them, performing subsequent manipulations in the space of transform coefficients, and then inverting the representation. There is thus a very tight connection between the development of image processing algorithms and image representations. A large amount of research has focused on developing novel representations appropriate for various image processing applications.

Representations should be appropriate for the underlying signal class. For example, one might not want to use the same representation for greyscale photographic images as for images of printed text, even though both are two dimensional real-valued functions. There is thus a strong interplay between developing appropriate signal representations and studying the structural and statistical properties of the underlying signal. The design of an appropriate representation depends strongly upon both the desired task and the properties of the underlying signal. For example, in image compression applications one typically seeks a representation that will give a good approximation of the image signal with only a small number of nonzero transform coefficients.

For other processing tasks, while the sparsity of the representation may be less important, one may still motivate the design of a representation by seeking to more explicitly represent structural properties of images that are important for perception. One of the most noticeable properties of natural images is that they typically contain strongly oriented features such as edges and lines. There has been much recent interest in image representations that are well suited for capturing local geometry. Some authors have proposed sparse overcomplete expansions that approximate the local geometry well e.g, [18]. The approach of Taubman and Zakhor [57], and the more recent bandlet approach of Pennec and Mallat construct a new local basis by resampling the image adaptively according to the local orientation [40]. Mallat and Zhong [31] showed that an image could be reconstructed with reasonable accuracy from knowledge of the locations of multi-scale zero-crossings. The wedgelet scheme [17, 45] represents image as using step edges parameterized by intensity value and orientation. Li has developed explicit representations of the local phase structure around edges [26]. A direct representation of images in terms of edges was proposed by Elder, who extracted the orientation, slope and position of edges and showed that this information was sufficient to reconstruct the original image [19].

Knowing that it is possible to reconstruct an image from a certain type of measurement is also of theoretical interest, as it tells us that those measurements are sufficient for capturing image structure. This can provide new ways of thinking about what an image is.

In this chapter I present a novel nonlinear image representation based on directly representing local image orientation. This orientation representation is based on a linear wavelet type transform known as the Steerable Pyramid. The steerable pyramid allows the representation of the image in terms of its gradient at multiple scales. By transforming these gradient vectors at different locations and scales into polar coordinates, the gradient is partitioned into magnitude and orientation information. I demonstrate that it is possible to discard the magnitude information yet still recover the original image. This yields a representation of the image in terms of the local orientation at different scales, a purely geometric quantity.

The reconstruction algorithm works by employing alternating projections onto two convex sets. Viewing the algorithm in this way gives a simple proof of convergence to a fixed point. Uniqueness of the reconstructed image has not been proven, but is always observed in practice. As a measure of the stability of this representation, I study its behavior under quantization of the local orientations. After quantization, the reconstruction is no longer exact. However, I find that the reconstruction performance decays gracefully with increasingly coarse quantization. Even after extreme quantization to as few as three orientations, reasonable quality images are reconstructed.

2.1 Steerable Pyramid Transform

The Steerable Pyramid (SP) is a multiscale linear image representation originally developed by Simoncelli et al [53, 54]. As the work presented in this thesis uses the Steerable Pyramid extensively, a detailed description is given here.

The SP is a filter bank transform where the filters are derivative operators at multiple spatial scales. The output of the transform is a set of subbands that are produced by convolving the original image with each of the filters. The SP decomposes an $m \times n$ pixel image into distinct orientation subbands at multiple scales. The number of spatial scales J and orientations K may be chosen freely, provided $J < \log_2(\min(m, n)) - 2$. With J and K fixed, the SP decomposes the image into a highpass residual bank, J sets of K oriented bandpass bands, and a lowpass residual band. Throughout this work, the number of “bands” of the SP transform refers to the number of orientations, K, of the specified transform.

The SP filters are chosen in order to possess several important properties. In particular the filters are designed to be polar separable in the Fourier domain, to prevent spatial aliasing in each subband after downsampling, and to form a tight frame. The filters corresponding to different orientation subbands also have the key property that they are rotated copies of each other. In this way, all of the bandpass filters can be formed as translated, scaled and rotated copies of a single “mother” filter.

2.1.1 Filter Design

Denote the highpass filter by $H(x, y)$, the bandpass filters by $B_{s,k}(x, y)$ where s and k indicate the scale and orientation of the filter, and the lowpass filter by $L(x, y)$. The dyadic scaling relationship between the bandpass filters can be expressed as

$$B_{s+1,k}(x, y) = B_{s,k}(x/2, y/2) \tag{2.1}$$

where $s = 1$ corresponds to the finest spatial scale and $s = J$ the coarsest spatial scale.

The filter design is easier to study in the Fourier domain. Let $\mathcal{F}[f]$ denote the discrete Fourier transform of f and $\mathcal{F}^{-1}[g]$ the inverse discrete Fourier transform of g. By the convolution theorem one may write

$$I \star B_{s,k} = \mathcal{F}^{-1}\left[\mathcal{F}[I] \cdot \mathcal{F}[B_{s,k}]\right] \tag{2.2}$$

where I is the original input image and $\cdot$ indicates pointwise multiplication.

The operation of each bandpass filter is equivalent to blurring with a radially symmetric bandpass filter followed by taking the (K-1)$^{\text{th}}$ order derivative along a specified direction. As the image signal lives on a discrete lattice, it is necessary to specify what is meant by differentiation. Note that for a continuous differentiable two dimensional function,

$$\begin{aligned}\mathcal{F}\left[\frac{d^{K-1}f(x,y)}{dx^{K-1}}\right] &= (iw_x)^{K-1}\hat{f}(w_x,w_y)\\ &= (ir\cos(\theta))^{K-1}\hat{f}(w_x,w_y)\end{aligned}$$

where r and θ are polar coordinates in the frequency domain. This property is used to define differentiation for the SP filters.

The Fourier transform of the bandpass SP filter at scale s and orientation band k can then be written in polar coordinates as

$$\begin{aligned}\hat{B}_{s,k}(r,\theta) &= \frac{g_s(r)}{r^{K-1}}\left(ir\cos\left(\theta-\frac{(k-1)\pi}{K}\right)\right)^{K-1}\\ &= i^{K-1}g_s(r)\cos^{K-1}\left(\theta-\frac{(k-1)\pi}{K}\right)\end{aligned} \tag{2.3}$$

where $\frac{g_s(r)}{r^{K-1}}$ is the Fourier transform of the radially symmetric blurring operator at the s$^{\text{th}}$ scale, and the (K-1)$^{\text{th}}$ order derivative is taken in the $\frac{(k-1)\pi}{K}$ direction.

The dyadic scaling of the bandpass SP filters implies that

$$g_s(r) = g\left(2^{(s-1)}r\right) \tag{2.4}$$

where $g(r)$ is a "mother" function corresponding to the filters at the finest spatial scale.

The radial function $g(r)$ may be chosen such that the SP transform is a tight frame. Given two Hilbert spaces X and Y, a linear operator $A : X \to Y$ is a tight frame if there is a constant c so that

$$c\,||Ax||_Y^2 = ||x||_X^2 \tag{2.5}$$

for all $x \in X$.

Letting X be the image space with standard inner product and Y the space of undecimated SP coefficients, the tight frame condition (taking $c = 1$) implies

$$||I||^2 = \left(\sum_{s,k} ||I \star B_{s,k}||^2 + ||I \star H||^2 + ||I \star L||^2 \right) \tag{2.6}$$

By Plancharel's relation, each of the norms in the above sum may be computed in the Fourier domain. This implies that the filters must "tile" the Fourier domain, i.e.

$$\sum_{s,k} |\hat{B}_{s,k}|^2 + |\hat{H}|^2 + |\hat{L}|^2 = 1 \tag{2.7}$$

This tiling condition, combined with the dyadic scaling, imposes strong conditions on the mother radial function g. For values of r out of the domain of support of the lowpass and highpass filters, 2.3, 2.4 and 2.7 imply

$$\sum_{s,k} g\left(2^{s-1}r\right)^2 \left[\cos\left(\theta - \frac{(k-1)\pi}{K}\right)\right]^{2(K-1)} = 1 \tag{2.8}$$

The powers of cosine tile over θ, due to the identity

$$\sum_{k=1}^{K} \cos\left(\theta - \frac{(k-1)\pi}{K}\right)^{2(K-1)} = \frac{K}{2^{2(K-1)}}\binom{2K-2}{K-1} \tag{2.9}$$

(see appendix 5). Denote the r.h.s. of (2.9) by C_K. Setting $g(r) = \bar{g}(r)/\sqrt{C_K}$, $g_{high}(r) = \hat{H}$ and $g_{low}(r) = \hat{L}$, the tiling constraint (2.7) is equivalent to

$$g_{low}(r)^2 + g_{high}(r)^2 + \sum_{s=1}^{J} \bar{g}(2^{s-1}r)^2 = 1 \tag{2.10}$$

This design constraint may be satisfied by setting

$$\bar{g}(r) = \begin{cases} \phi(r), & \text{if } \frac{\pi}{2} \leq r \leq \pi; \\ \sqrt{1-\phi(2r)^2} & \text{if } \frac{\pi}{4} \leq r \leq \frac{\pi}{2}; \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

where $\phi(r)$ is a monotonically decreasing function on $[\frac{\pi}{2}, \pi]$ with $\phi(\frac{\pi}{2}) = 1$ and $\phi(\pi) = 0$. The filters in this work are formed using

$$\phi(r) = \sin\left(\frac{\pi}{2}\left|\log_2\left(\frac{r}{\pi}\right)\right|\right) \tag{2.12}$$

The residual lowpass and highpass filters are then determined by requiring (2.7) to be satisfied, with $g_{low}(r)$ supported in $[0, \frac{\pi}{2^J}]$ and $g_{high}(r)$ supported in $[\frac{\pi}{2}, \pi]$. See figure 2.1.1.

For the work in this thesis, the Steerable Pyramid was implemented by multiplication in the Fourier domain. This implicitly implies circular boundary handling.

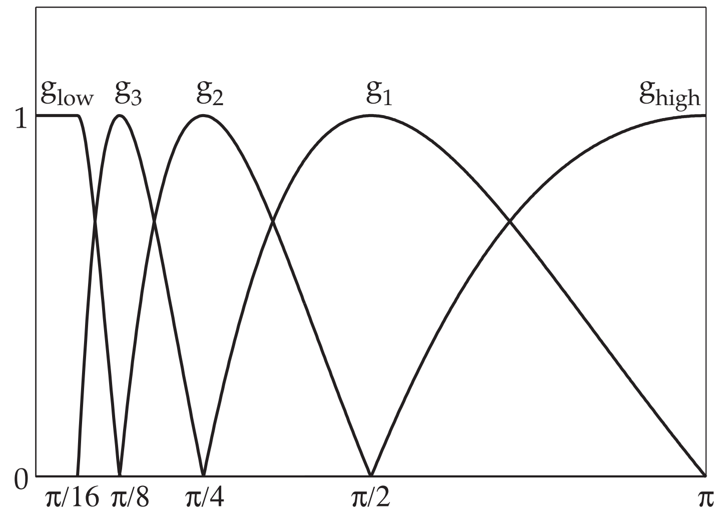


Figure 2.1: Radial functions for SP filters, for J=3 scales

2.1.2 Inverse SP Transform

As the SP transform is overcomplete, it maps a lower dimensional space into a higher dimensional space. Written in matrix form, the SP would not be a square matrix, and thus cannot have both a left and right inverse. It is possible to find a left inverse, although it will not be unique. Again let X denote the space of image pixels and Y the space of SP coefficients. Such a left inverse would be a linear operator $B : Y \to X$ such that

$$BA = I_X$$

where A is the SP transform operator and I_X is the identity operator on the space X.

As the SP is a tight frame, we may take $B = A^{\dagger}$. The adjoint $A^{\dagger}$ corresponds to convolving with the complex conjugate of the original SP filters. The SP

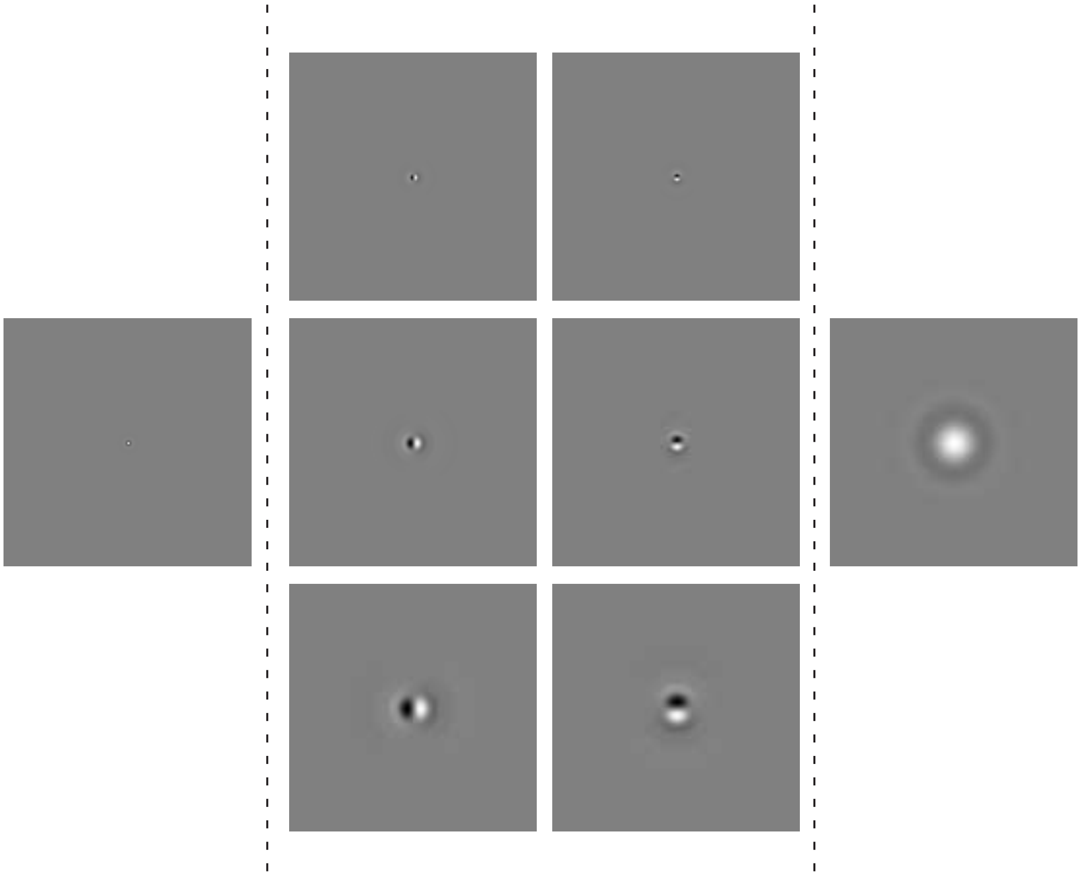


Figure 2.2: Steerable pyramid filters in spatial domain, for 3 scales and 2 orientation bands. **Left:** Residual Highpass filter **Center:** directional filters at three spatial scales, from fine to coarse. **Right:** Residual Lowpass filter

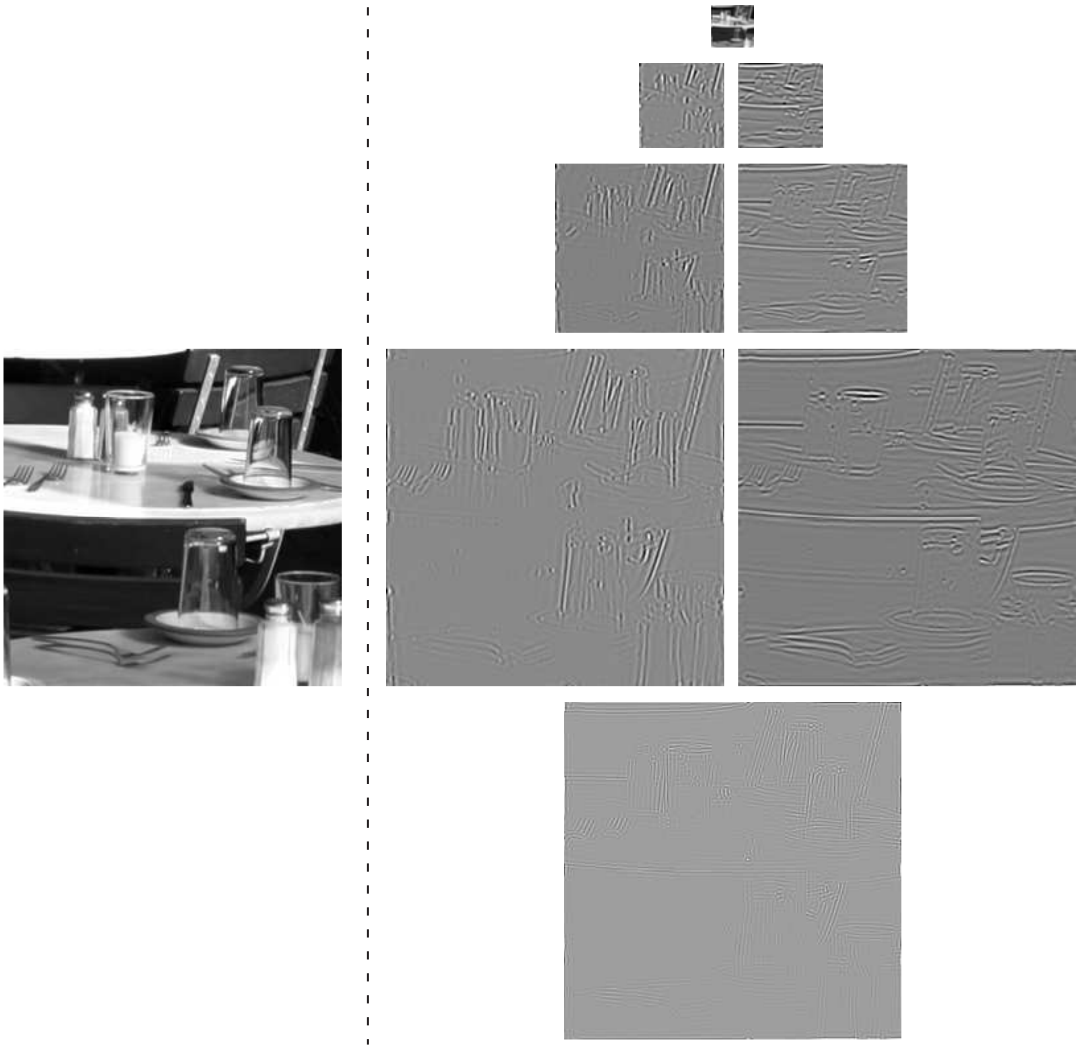


Figure 2.3: Steerable Pyramid bands. **Left:** Original Image. **Right:** SP bands for 3 scales and 2 orientation bands, arranged from coarse to fine from top to bottom. Very top is lowpass residual band, very bottom is highpass residual band.

transform may thus be inverted by convolving each output subband with the complex conjugate of the filter originally used to form the subband, and then adding the results.

2.1.3 Downsampling and overcompleteness

As $g_s(r) = 0$ for $r > \frac{\pi}{2^{s-1}}$, the outputs of all of the filters at scale s have their Fourier transforms supported within $r \leq \frac{\pi}{2^{s-1}}$. This implies that these filter output subbands may be subsampled by a factor of 2^{s-1} in the x and y directions without any loss of information or introduction of any spatial aliasing. While the filter design was explained using the unsampled transform, in practice the SP transform is usually subsampled. The tight frame property still holds, although the inner product on the space of subsampled SP coefficients must be adjusted to account for subsampling.

An important property of the SP transform is that it is overcomplete. Even after the above subsampling, there are more SP coefficients than original image pixels. The amount of overcompleteness may be calculated in the limit of "large J" by considering the sizes of the oriented bands. If the original image is of size MxN, the highpass and finest oriented bands will be the same size. At each coarser scale the size of each oriented band will be reduced by 2 in each direction, so the number of coefficients will be reduced by a factor of 4. For a transformation with K oriented bands, the number of coefficients will then be

$$MN \times (1 + K + K/4 + K/4^2 + K/4^3 + ...) = MN \times (1 + \frac{4K}{3})$$

so the transform will be $1+\frac{4K}{3}$ times overcomplete. The actual overcompleteness of a transform in practice will be slightly different due to using a finite number of spatial scales and that the original image dimensions may not be powers of two.

2.1.4 Steerability

The SP filters are "steerable", meaning that a filter at an arbitrary orientation can be produced by a finite linear combination of filters at fixed orientations. For the K band SP transform, the dimension of this linear combination is K. A filter at a particular location and scale, but arbitrary orientation may be written as a linear combination of the K basis filters at the orientations $\theta_k = \frac{(k-1)\pi}{K}$. This implies that the full set of oriented filters at a particular location and scale span a rotationally invariant subspace.

The steerability property can be analyzed in the Fourier domain. From (2.3), the Fourier transform of the SP filter at orientation ϕ and scale s is

$$i^{K-1} g_s(r) \cos^{K-1}(\theta - \phi)$$

As the radial function $g_s(r)$ for filters at the same scale but different orientations are the same, steerability will hold if there exist coefficients $c_k(\phi)$ such that

$$\cos^{K-1}(\theta - \phi) = \sum_{k=1}^{K} c_k(\phi) \cos^{K-1}\left(\theta - \frac{(k-1)\pi}{K}\right) \tag{2.13}$$

This holds as the translated cosines form a linear space of dimension K. Computation of the $c_k(\phi)$ is discussed in appendix 5.

As the filters act upon the input image data in a linear manner, the steerability of the filters implies the steerability of filter responses. Given an input image I, the convolution of I with a filter at an arbitrary orientation ϕ can be calculated as a linear combination of the convolutions of I with the K filters at orientations $\theta_k = \frac{(k-1)\pi}{K}$.

2.2 Separation of Orientation and Magnitude

The two band (K=2) Steerable Pyramid transform gives a representation of the input image in terms of the filter outputs of oriented filters in the x and y directions. As these filters are first derivative operators, one can take them to be components of the image gradient at multiple scales. In this way, the two-band SP represents the image in terms of its gradient at multiple scales.

The nonlinear image representation described in this chapter is based upon the orientation of these multiscale image gradient vectors. By transforming the gradients into polar coordinates, one can define orientation and magnitude bands for each scale. Specifically, for an input image I of dimension MxN pixels, the output of the two band SP with J consists of the highpass, oriented bandpass and lowpass subbands

$$\begin{aligned} H(m,n) &: m = 1...M, n = 1...N \\ B_{s,x}(m,n) &: m = 1...M/2^{s-1}, n = 1...N/2^{s-1}, s = 1...J \\ B_{s,y}(m,n) &: m = 1...M/2^{s-1}, n = 1...N/2^{s-1}, s = 1...J \\ L(m,n) &: m = 1...M/2^{J}, n = 1...N/2^{J} \end{aligned} \tag{2.14}$$

The orientation and magnitude bands are then computed as

$$M_s(m,n) = \sqrt{B_{s,x}(m,n)^2 + B_{s,y}(m,n)^2}$$
$$\Theta_s(m,n) = \arctan(B_{s,y}(m,n), B_{s,x}(m,n))$$

As the lowpass and highpass bands are scalar quantities and are not oriented, no polar transformation is performed on them.

This transformation divides the information contained in the gradient bands into two parts. The orientation band is a purely geometric quantity measuring the local orientation of the image, while the magnitude band essentially measures the local signal power. One may ask whether the magnitude or orientation bands are more important for representing the signal structure.

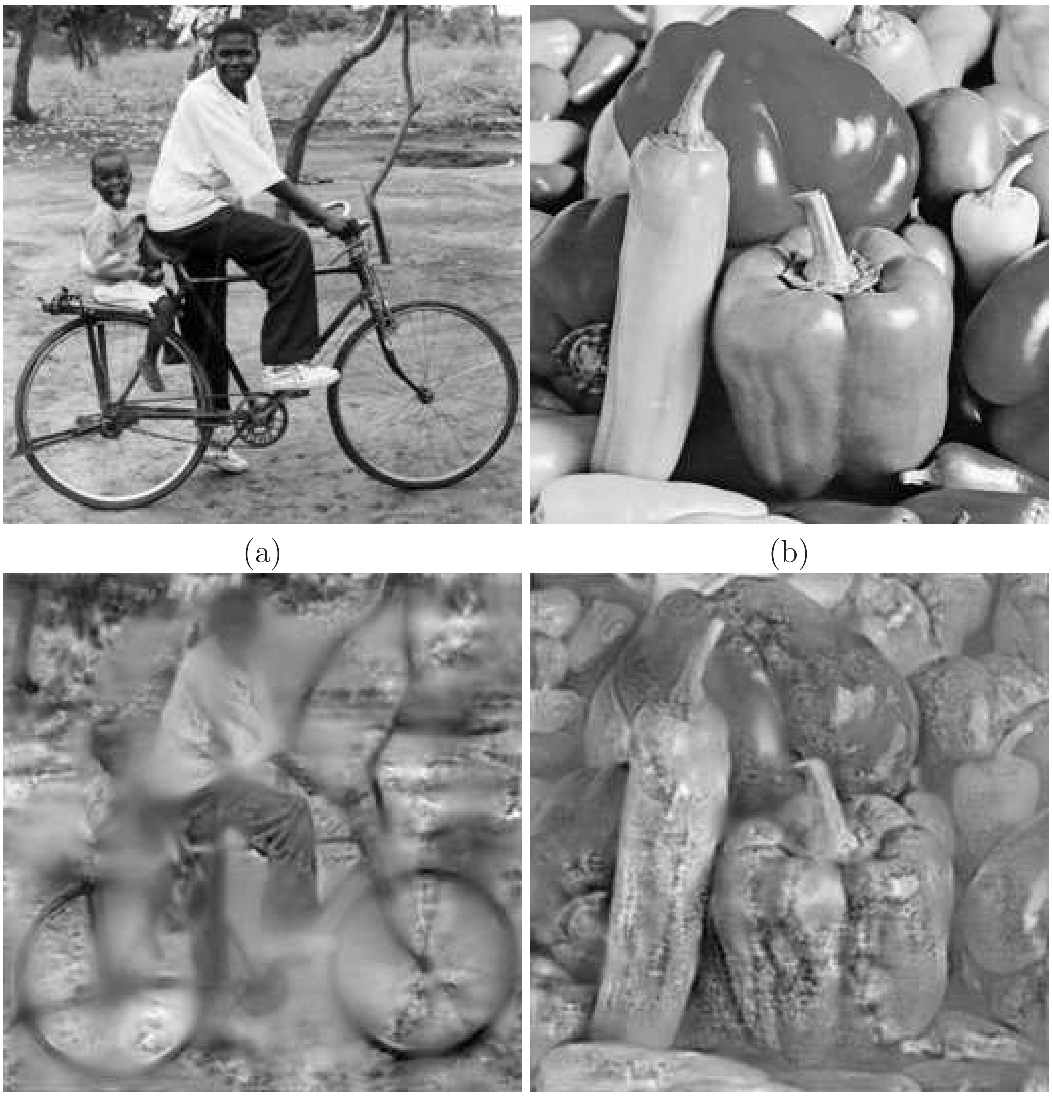
This question is similar in spirit to the classic work of Oppenheim and Lim [38], who investigated the relative importance of the magnitudes and phases of discrete Fourier transform coefficients for representing image structure. They took the information in present in the Fourier transform of an image and partitioned it into magnitude and phase information. By performing numerical experiments involving recombining Fourier magnitudes and phases from different images, as well as with random information, Oppenheim and Lim showed that image structure was more explicitly captured by phase information than by magnitude information.

There are two key differences between the local orientation and magnitude bands studied in this work and the Fourier magnitudes and phases. Firstly, The Fourier coefficients are global quantities, with each coefficient depending on the entire image. In contrast, the steerable pyramid coefficients, and thus the local orientation and magnitude bands, are local measurements. Secondly, the steerable pyramid is an overcomplete transformation while the Fourier transform is not.

Some previous work has focused on recovering image signal from the magnitudes of oriented filter responses. Wundrich et.al. have show that images may be represented keeping the magnitudes (e.g. discarding the phases) of responses to a set of doubly overcomplete Gabor filters [67]. Shams and Von der Malsburg likewise studied reconstructing image features from magnitudes of Gabor filters used as a model of cortical complex cells [50]. However, for multiscale gradients calculated using the steerable pyramid, I find the local orientation to be more important than the magnitudes for representing image structure.

The relative importance of the local orientation information versus the magnitudes is illustrated by three sets of numerical experiments involving hybrid images. In each of these experiments, "true" orientation and magnitudes were taken from actual images, and combined with "other" data to form hybrid images. For one experiment the "other" data were orientations and magnitudes taken from a different image, while for the other two the "other" data were either sampled randomly or chosen to be constant. For all of these experiments, the original images were size 256x256 pixels. The steerable pyramid transforms were computed using the maximum possible number of spatial scales, which is 6 for 256x256 size images.

The first experiment involved "swapping" the orientation and magnitudes of two actual images. The magnitude and orientation bands were extracted from two images I_1 and I_2 as described above. Hybrid images were formed by swapping the two magnitude and orientation bands, and then inverting the SP



(a) (b)

(c) (d)

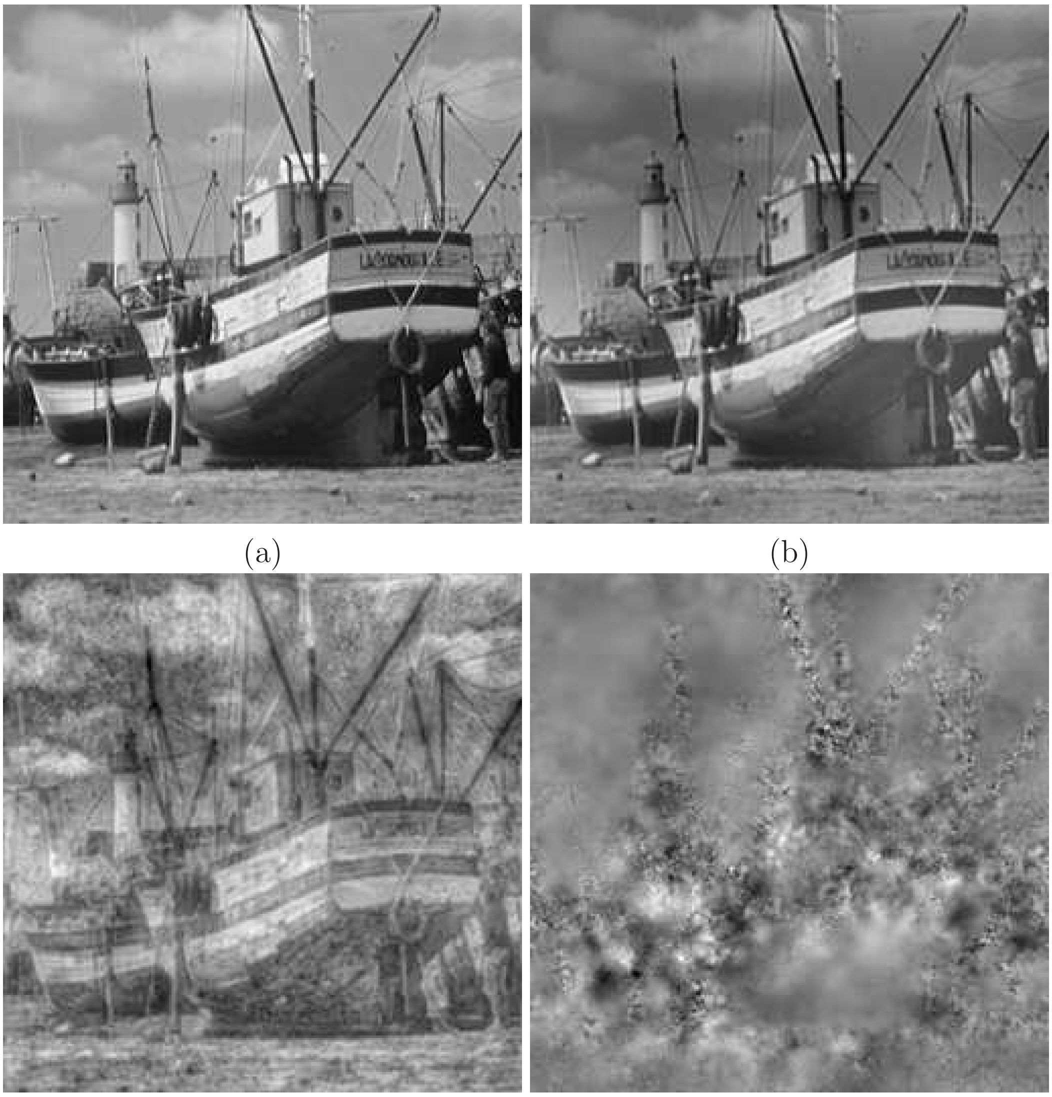
Figure 2.4: Swapping orientation and magnitudes. Magnitudes and orientations were extracted from original images (a) and (b). Hybrid image (c) was formed using orientations from (a) and magnitudes from (b). Hybrid image (d) was formed using orientations from (b) and magnitudes from (a).

transform. As the scalar lowpass and highpass bands could not be treated in this way, they were simply set to zero for the hybrid images. The results of this hybridization are shown in figure 2.4.

Setting the lowpass bands to zero implies that the pixel values of hybrid images will sum to zero, so these will necessarily have many negative pixel values. To view these hybrid images it is necessary to rescale their dynamic ranges. Setting the lowpass and highpass bands to zero is equivalent to bandpass filtering the image. It may seem disturbing that this information is discarded in this series of experiments. However if enough spatial scales are used in the transform, nearly all of the image structure is contained in the bandpass bands. For comparison purposes, the effects of removing the highpass and lowpass bands are displayed in figure (2.5 b).

As can be seen, the hybrid images are much more similar to the image from which the orientation was taken. Some residual effect of the magnitude band can be seen as modulating the local contrast, such as the faint outline of the bicycle in the hybrid image (d). However the image structure is clearly better captured by the orientation information than by the magnitude information.

The second set of hybrid images demonstrating the relative importance of orientation information was generated by combining authentic orientations and magnitudes with randomly generated data. Given an original image I, the magnitude and orientation bands were computed. A set of random orientations were generated by sampled from the uniform distribution on $[0, 2\pi]$. The randomly generated magnitudes at each scale were constrained to have the same marginal statistics as the original magnitudes. This was done by first computing a sample histogram of the original magnitudes, and then drawing synthetic



(a)

(b)

(c)

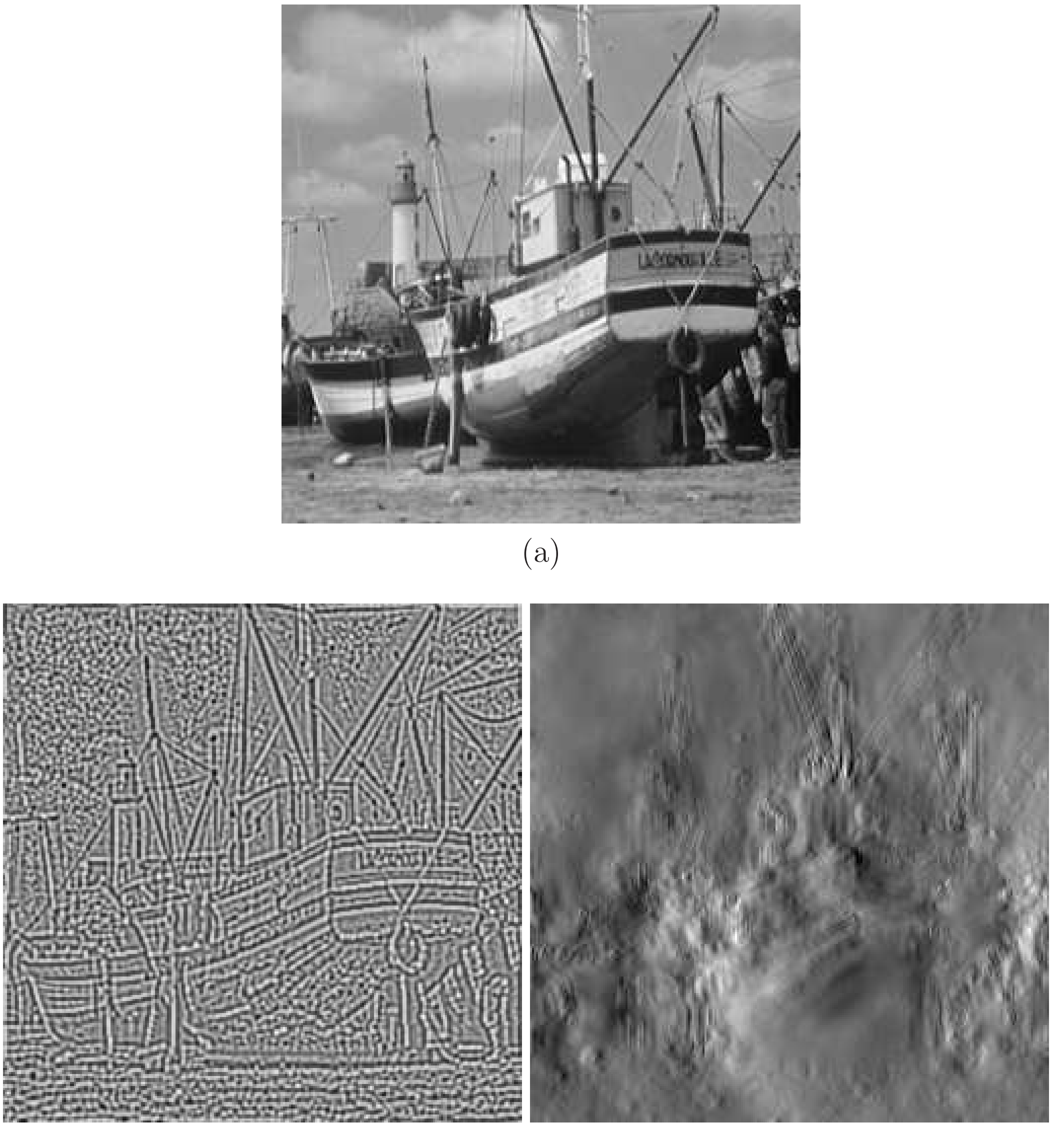
(d)

Figure 2.5: Random orientation and magnitudes. (a) Original image. (b) is original image, with highpass and lowpass bands set to zero. (c) Image formed with orientations extracted from (a) and random magnitudes. (d) Image formed from magnitudes extracted from (a) and random orientations.

magnitudes according to this distribution. Two hybrid images, one containing original magnitudes and random orientations, the second containing original orientations and random magnitudes, were formed. As before, the lowpass and highpass bands were set to zero for the hybrid images. Results are displayed in figure 2.5. While the hybrid image containing the true orientations appears noisy and with poor contrast, it nonetheless manages to preserve much of the original image structure. The hybrid containing random orientations, however, is completely devoid of local structures such as lines and edges. While some residual of the original boat image is visible, it is present as a modulation of the contrast of what looks somewhat like 1/f spectrum noise. These images again provide evidence that the image structure information is carried more explicitly by the orientation information than by the magnitudes.

The third set of images was formed by combining orientation and magnitudes from a real image with constant magnitude and orientation data. Setting the magnitude data to be constant yields a similar result as "whitening" the image, where one attempts to flatten the image power spectrum [24]. The results are shown in figure 2.6. As would be expected, the image formed from original orientations and constant magnitudes has nearly constant contrast. This results in a "noisy" appearance in regions where the original image had low signal power, such as the sky. However, much of the original image structure is preserved.

For the hybrid image with the original magnitudes, the constant synthetic orientation was chosen to be $\theta = 0$. Imposing this constant orientation has the effect of rotating every image gradient vector to be horizontal. As can be seen in the resulting image, only vertically oriented structure such as the mast of the boat and the lighthouse can be represented. Of the two hybrid images, again



(a)

(b) (c)

Figure 2.6: Constant orientation and magnitudes

the one with the correct local orientations is more perceptually more similar to the original image.

2.3 Reconstruction from Local Orientations

The three sets of hybrid images discussed earlier indicate that the structural content of images is more saliently captured by the local orientation data than by the magnitude data. While each of the hybrid images formed from authentic orientations [figures 2.4 (c,d) 2.5 (c), 2.6 (b)] were somewhat similar to the original image providing the orientation data, thus demonstrating that much of the image structure is captured by the orientation information, they were not good quality reproductions of the original. One can ask exactly how much information about the image is encoded in the local multiscale orientations. A natural way to answer this question is by image synthesis. If it is possible to reconstruct a good quality image from a given set of measurements, this is a demonstration that those measurements are sufficient to capture the image information.

The method in which these hybrid images were synthesized was very simple. This raises the natural question of whether it is possible to make better use of the local multiscale orientation data, and through some possibly more sophisticated procedure extract a more faithful version of the original image. This section explains that this is possible and in fact the entire image may be completely recovered after discarding the magnitude information, if the highpass and lowpass residual bands are kept.

The image produced in figure (2.5 c) was synthesized by combining the original orientation bands and random magnitude bands to yield a set of Steerable Pyramid coefficients, and then inverting the SP transform. If this image is then re-analyzed using the SP filters, its local orientations will generally not be the same as the original orientation, because of the overcompleteness of the SP. This discrepancy gives one the freedom to re-impose the original local orientation data, yielding a distinct set of SP coefficients, and reconstruct the image again. The residual highpass and lowpass bands are also imposed. This process is illustrated schematically in figure 2.7. The main result of this chapter is that iterative application of this simple algorithm converges to the original image.

The magnitudes used to synthesize the image from figure (2.5) were sampled to have the same marginal statistics as the original image magnitudes, which relied on the availability of the original magnitudes. In order to demonstrate the feasibility of reconstructing an image after the magnitude information is discarded, the initial iterate should be chosen without any knowledge of the original magnitudes. I chose to initialize the reconstruction algorithm using a random Gaussian noise image with power spectrum proportional to $1/|f|$, which roughly mimics the power spectrum of natural images [46]. Using such an initial

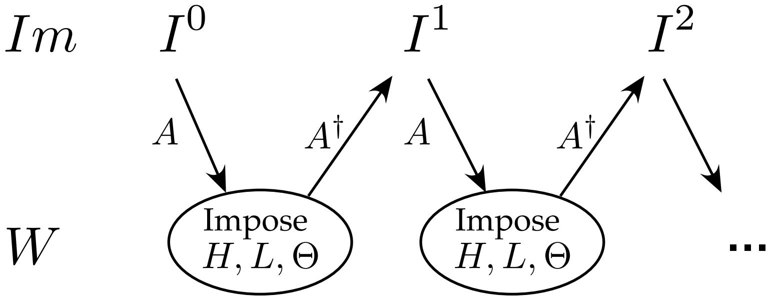


Figure 2.7: Schematic of reconstruction algorithm

image, the first several steps of the reconstruction algorithm are shown in 2.8.

The reconstruction algorithm is easier to understand when viewed entirely in the space of pyramid coefficients. Before proceeding, it is helpful to introduce the following notation. Let Im denote the space of image pixels, and W be the space of steerable pyramid coefficients. Introduce the two-dimensional spaces $W_i = (x_i, y_i)$ of the x and y filter responses at the scale/space location i, where i indexes different locations in space and scale. It will be convenient to write W as the Cartesian product

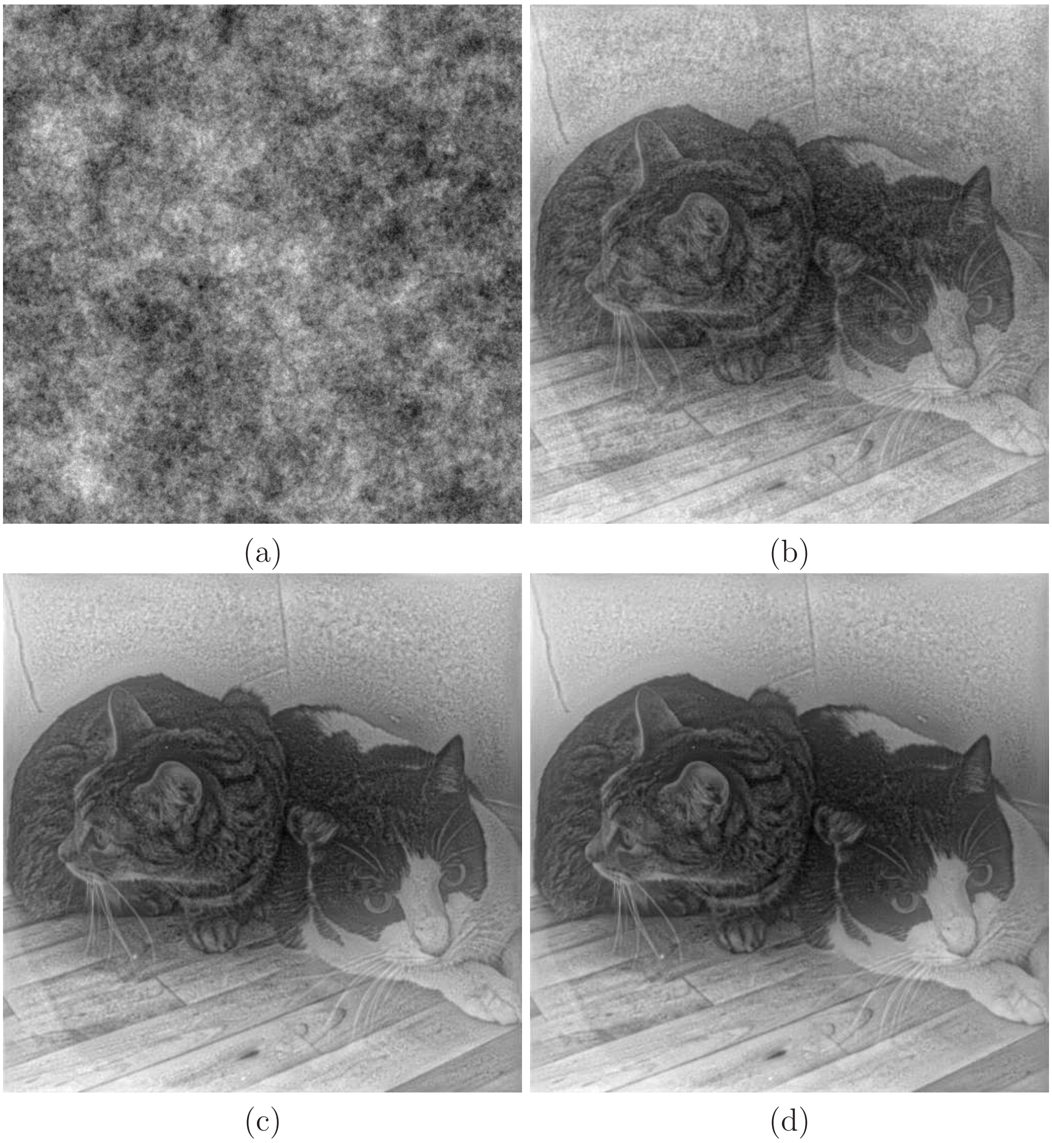
$$W = H \bigotimes \left(\bigotimes_{i=1}^{N_i} W_i \right) \bigotimes L \tag{2.15}$$

where the spaces H and L consist of the coefficients of the highpass and lowpass residual bands.

As the steerable pyramid is overcomplete, $\dim(Im) < \dim(W)$. Let $A : Im \rightarrow W$ be the steerable pyramid transform (the "analysis" operator). As A is a tight frame, $A^\dagger : W \rightarrow Im$ satisfies $A^\dagger \circ A = I_{Im}$, where I_{Im} is the identity operator on Im (see appendix 5). Applying $A^\dagger$ is equivalent to inverting the steerable pyramid transform.

Let $U = A(Im)$ be the "image under A of the image space". This is the set of steerable pyramid coefficients that can be achieved as the transform of an actual image. The operator $A \circ A^\dagger : W \rightarrow W$, which corresponds to inverting and rebuilding the steerable pyramid, is not the identity operator on W. It is in fact an orthogonal projection onto U, as shown in appendix 5.

Let $O_\Theta : W \rightarrow W$ be the function that imposes the specified orientation and highpass and lowpass bands. If $W_\Theta \subset W$ is the set of pyramid coefficients



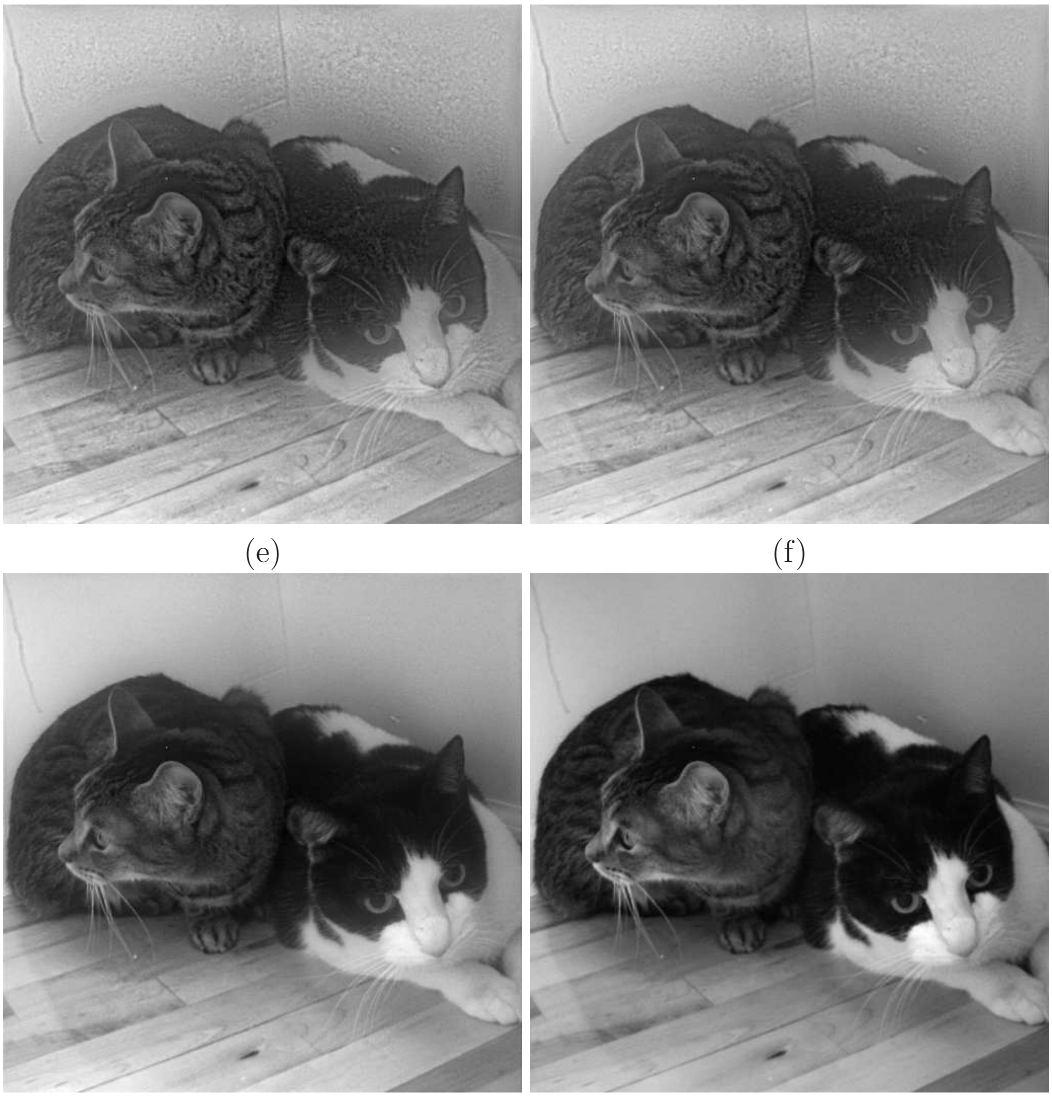
(a)

(b)

(c)

(d)

Figure 2.8: Reconstruction from orientation data. For this series of iterates, highpass and lowpass bands were imposed at each scale, average power was not imposed and extrapolation was not used. (a) image used as starting point (b) 1 iteration (c) 5 iterations (d) 10 iterations



(e) (f)

(g) (h)

Figure 2.8: Reconstruction from orientation data (cont). (e) 15 iterations (f) 20 iterations (g) 100 iterations (h) original image

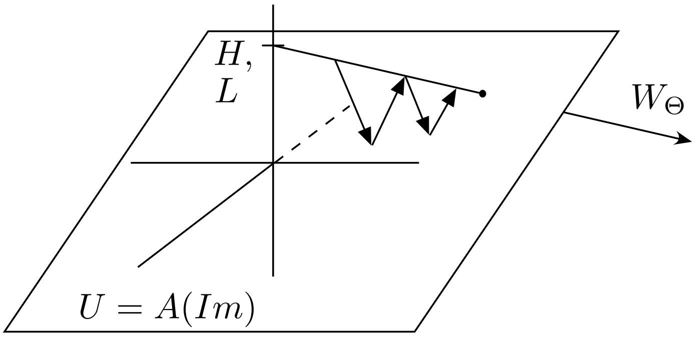


Figure 2.9: Geometry of reconstruction algorithm functioning by projection onto the convex sets U and W_Θ

consistent with the specified orientation, lowpass and highpass information, then $O_\Theta : W \rightarrow W_\Theta$ can be viewed as a projection. I will defer briefly the exact specification of this function as it is possible to impose the orientations in more than one way. Given this notation, a single iteration of the reconstruction algorithm, starting and ending in W, is equivalent to applying $AA^\dagger \circ O_\Theta$. This has the form of alternate projections, first onto the set W_Θ, then onto U. This is sketched in figure 2.9.

Both of these sets are convex. U is trivially convex as it is a linear subspace. W_Θ can be characterized as the positive linear combination of a set of vectors corresponding to single orientation measurements, offset by a vector containing the highpass and lowpass bands. Set $v_{res} \in W$ to have the specified highpass and lowpass components, and be zero for all of the bandpass components. Let the index i refer to different locations in space and scale and θ_i be the the local multiscale orientation at location i. Set $v_i(\theta_i) \in W$ to be zero in every component except for the two corresponding to the x and y filter responses for the i^{th} particular scale and spatial location, which are $\cos(\theta_i), \sin(\theta_i)$. Vectors

$x \in W_\Theta$ are then exactly those with the form

$$x = v_{res} + \sum M_i v_i(\theta_i) \tag{2.16}$$

where M_i are the positive magnitudes at space/scale location i. It is now straightforward to show that W_Θ is convex. Given two points $x_1 = v_{res} + \sum M_i^1 v_i(\theta_i)$ and $x_2 = v_{res} + \sum M_i^2 v_i(\theta_i)$, for any scalar $\lambda \in [0,1]$, one has

$$\lambda x_1 + (1-\lambda)x_2 = v_{res} + \sum \left(\lambda M_i^1 + (1-\lambda)M_i^2\right) v_i(\theta_i)$$

which is clearly in W_Θ as $\lambda M_i^1 + (1-\lambda)M_i^2 \geq 0$.

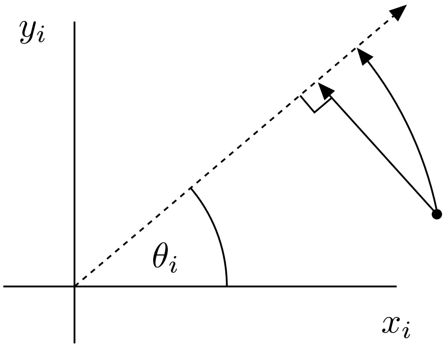


Figure 2.10: Imposition of orientation (operator O_Θ) by rotation or projection. The dotted line represents a slice of W_Θ, the set of coefficients having the correct orientation.

The operator O_Θ acts separately on the component spaces H,L and W_i. On H and L, O_Θ simply imposes the specified highpass and lowpass information. On each two dimensional space $W_i = (x_i, y_i)$, O_Θ can impose the specified orientation either by rotating the two-vector (x_i, y_i) to have orientation θ_i, or by orthogonally projecting onto the line with orientation θ_i. These two possibilities are shown in figure 2.10.

In the rotation case, the behavior on W_i is defined by

$$O_{\Theta}^{ROT}(x_i, y_i) = (\rho_i \cos(\theta_i), \rho_i \sin(\theta_i)) \tag{2.17}$$

where $\rho_i = \sqrt{x_i^2 + y_i^2}$. For the projection case, O_{Θ}^{PROJ} projects the point (x_i, y_i) onto the ray spanned by the unit vector $(\cos(\theta_i), \sin(\theta_i)$. Care must be taken to set points to zero that would project onto the diametrically opposing ray. This can be done by setting

$$O_{\Theta}^{PROJ}(x_i, y_i) = \nu\left[(\cos(\theta_i), \sin(\theta_i)) \bullet (x_i, y_i)\right](\cos(\theta_i), \sin(\theta_i))$$

where ν is the hinge function

$$\nu(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x > 0 \end{cases}$$

If O_{Θ} is chosen to act by projection onto each component subspace W_i, it is an orthogonal projection from the full space W onto W_{Θ}. The reconstruction algorithm then performs alternating orthogonal projection onto convex sets U and W_{Θ}. Such an algorithm is guaranteed to converge provided the convex sets have nonzero intersection, by a basic result due to Cheney and Goldstein [10, 3]. By construction, $W_{\Theta} \cap U$ must contain at least one point, namely the steerable pyramid transform of the original image. This proves

Theorem 1 *The reconstruction algorithm defined by iterating $AA^{\dagger}O_{\Theta}^{PROJ}$ converges to a fixed point in $p \in W$. There is an image $y \in Im$ such that $A(y) = p$,* *and p has the imposed local multiscale orientations θ_i, as well as the residual lowpass and highpass bands.*

As the proof of this relied on the use of orthogonal projections, it does not directly apply when the orientations are imposed by rotation. In practice, however, convergence is observed when using rotation and convergence is even faster than for reconstruction based on projections. This behavior may be partly explained by noting that projection reduces the magnitudes of the coefficients while these are preserved by rotation. After the first few steps of the algorithm with projections, the coefficient magnitudes are greatly reduced in the bandpass bands. Loosely speaking, these magnitudes are only slowly restored by information "leaking" in from the imposed lowpass band. The reconstruction algorithm based on imposing orientations by rotation suffers less from this effect. It should also be noted that the rotation and projection methods converge in the limit of small angle corrections, which occurs as the algorithm approaches a fixed point.

The method of alternating projections has been used before for a number of image processing applications. Thao and Vetterli studied using alternating projections to reconstruct bandlimited signals from quantized sample values [59]. Goyal et. al have studied the more general problem of reconstructing signals from quantized coefficients in an overcomplete linear representation [21]. Hirani and Totsuka employed alternating projections for the inpainting problem of removing user-selected image objects [22].

The result presented in this thesis does not guarantee the uniqueness of the reconstructed image. In practice, exact reconstruction to machine precision has been observed for all images with normal power spectral properties. It is, however, relatively easy to generate images for which the given reconstruction algorithm is not unique. Images which have been bandpass filtered such that their SP transforms have exactly zero values for the lowpass and highpass bands will not be uniquely constrained by the orientation information. This result is easy to understand, as the orientation measurements are invariant under multiplying the entire original image by a single global scalar. Typically, this "free scalar" is determined by imposing the lowpass residual band. If the lowpass band is exactly zero, however, imposing it does not constrain the overall scalar multiplier. In fact, for such bandpass filtered images it was observed that the algorithm would converge to a result that was simply a scalar multiple of the original image, where the exact value of the scalar depended on the initial starting point. It is unclear if this is the only type of example where uniqueness of the reconstruction can fail. Later in this chapter, some conditions on the orientation measurements are described which, if met, are sufficient to imply uniqueness of the representation.

2.4 Acceleration of convergence

The iterative reconstruction algorithms described above converge, but quite slowly. For a typical 512x512 image, beginning at a random noise starting point, about 200 iterations of the rotation-based method are needed before the result is visually indistinguishable from the original image. As alluded to above, the amplitudes are encoded only implicitly in the representation, and their recovery results from interactions between orientations at different positions and scales as well as the lowpass band.

A simple thought experiment reveals how surprising it is that the magnitudes may be recovered. If the original image were multiplied by an overall scalar multiple, none of the orientation measurements would be affected. This global "free scalar" is only pinned down by imposing the scalar lowpass and highpass bands, which are clearly not invariant under such scaling. This suggests that the reconstruction may be accelerated by explicitly imposing this overall scale factor that is only weakly implied by cross-scale interactions during reconstruction. One reasonable way to do this is to include as part of the representation the average power (sum of squares of magnitudes) of the pyramid coefficients at each scale. This is then imposed at each step of the reconstruction algorithm by rescaling by the magnitudes at each spatial scale.

Imposing the average power of each spatial scale greatly speeds the conver-

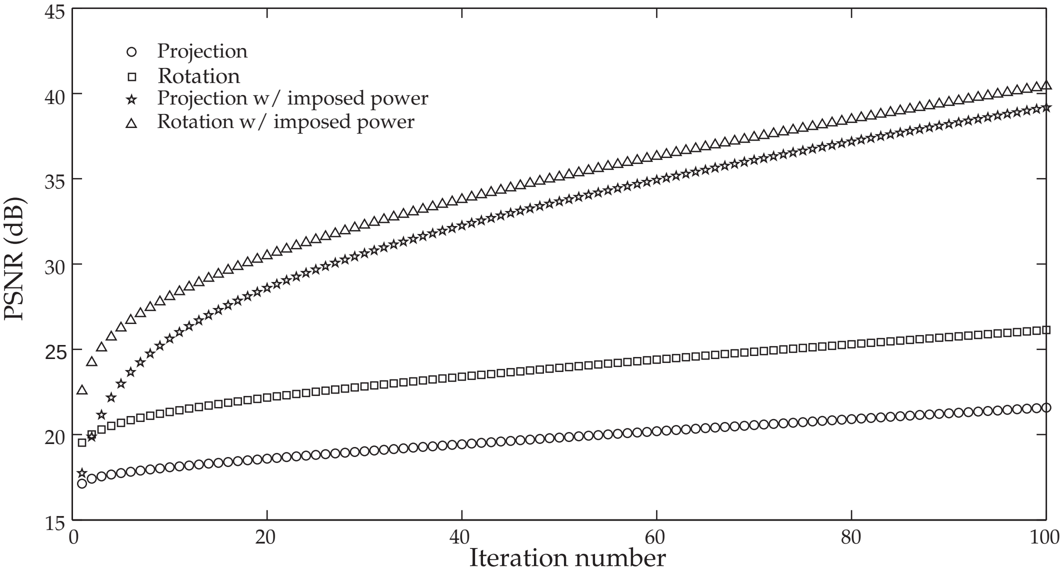


Figure 2.11: Effects of imposing average power to accelerate convergence. Curves are PSNR vs iteration for rotation and projection methods, with and without imposing average power at each scale.

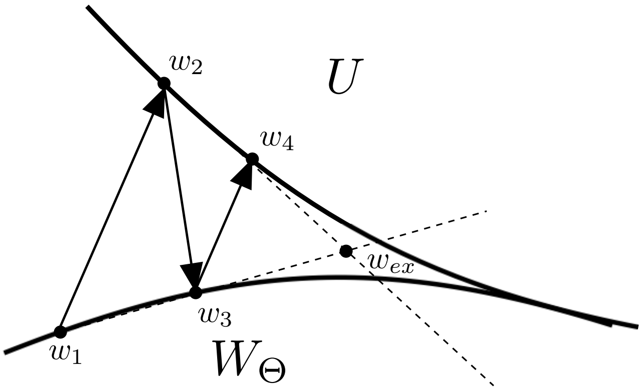


Figure 2.12: Acceleration of convergence by extrapolation in pyramid domain

gence of the algorithm, as shown in figure 2.11. Note that imposing the average power greatly reduces the difference in performance between the projection and rotation methods. This is understandable, as the reason the projection method performed worse was due to it reducing the magnitudes for large angle corrections. Restoring the overall power at each step by rescaling mitigates this effect.

A second method for accelerating the convergence by extrapolation is suggested in figure 2.12. Given four successive points $w_i \in W$, $i = 1...4$ generated by alternating projection, one may form the line l_1 passing through $w_1, w_3 \in W_\Theta$ and l_2 passing through $w_2, w_4 \in U$. One can find the two points where these lines are closest to each other (they will not intersect in general, in a high dimension space), and average them to define w_{ex}. The alternating projection may then be started again with the point w_{ex}, which should be closer to the intersection of U and W_Θ.

These lines may be parameterized by

$$l_1(t) = w_1 + t(w_3 - w_1) \tag{2.18}$$

$$l_2(s) = w_2 + s(w_4 - w_2) \tag{2.19}$$

Finding the points of closest intersection is equivalent to finding s and t minimizing

$$\begin{aligned} E(s,t) &= ||l_1(t) - l_2(s)||_W^2 \\ &= ||w_1 - w_2 + s(w_3 - w_1) + t(w_2 - w_4)||_W^2 \quad = \left|\left|\vec{a} + s\vec{b} + t\vec{c}\right|\right|_W^2 \end{aligned} \tag{2.20}$$

where $\vec{a} = w_1 - w_2$, $\vec{b} = w_3 - w_1$ and $\vec{c} = w_2 - w_4$. E is a quadratic polynomial in s and t and so has a unique minimum. Calculating partial derivatives gives

$$\frac{\partial E}{\partial s} = 2\left\langle \vec{a} + s\vec{b} + t\vec{c}, \vec{b}\right\rangle_W \tag{2.21}$$

$$\frac{\partial E}{\partial t} = 2\left\langle \vec{a} + s\vec{b} + t\vec{c}, \vec{c}\right\rangle_W \tag{2.22}$$

Setting these partial derivatives to zero yields a set of linear equations, solving them gives

$$\begin{pmatrix} s \\ t \end{pmatrix} = -\begin{pmatrix} <\vec{b},\vec{b}>_W & <\vec{b},\vec{c}>_W \\ <\vec{b},\vec{c}>_W & <\vec{c},\vec{c}>_W \end{pmatrix}^{-1} \begin{pmatrix} <\vec{a},\vec{b}>_W \\ <\vec{a},\vec{c}>_W \end{pmatrix} \tag{2.23}$$

The extrapolated point w_{ex} is then given by

$$w_{ex} = \frac{1}{2}\left(w_1 + s(w_3 - w_1) + w_2 + t(w_4 - w_2)\right) \tag{2.24}$$

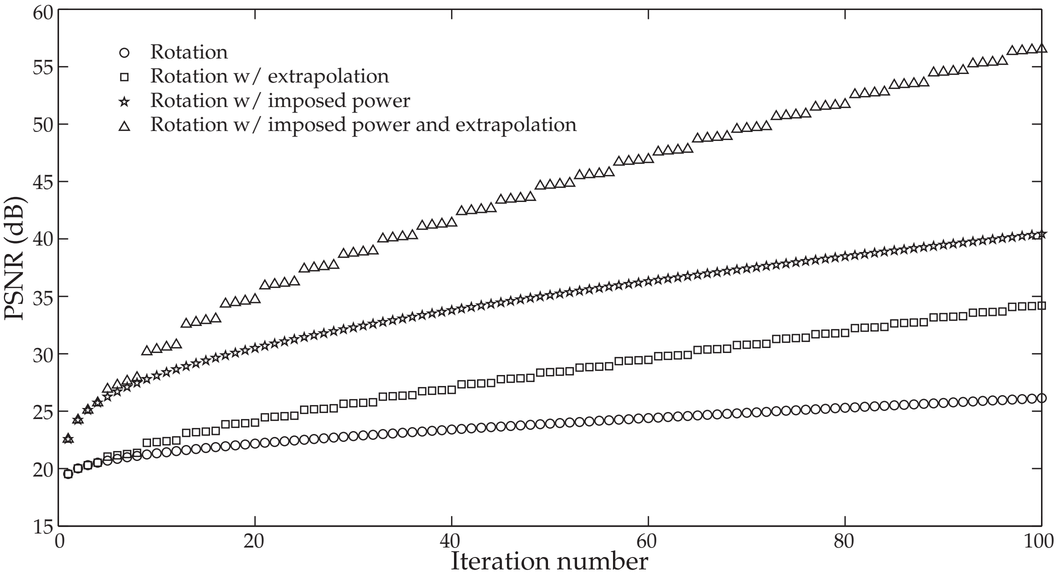
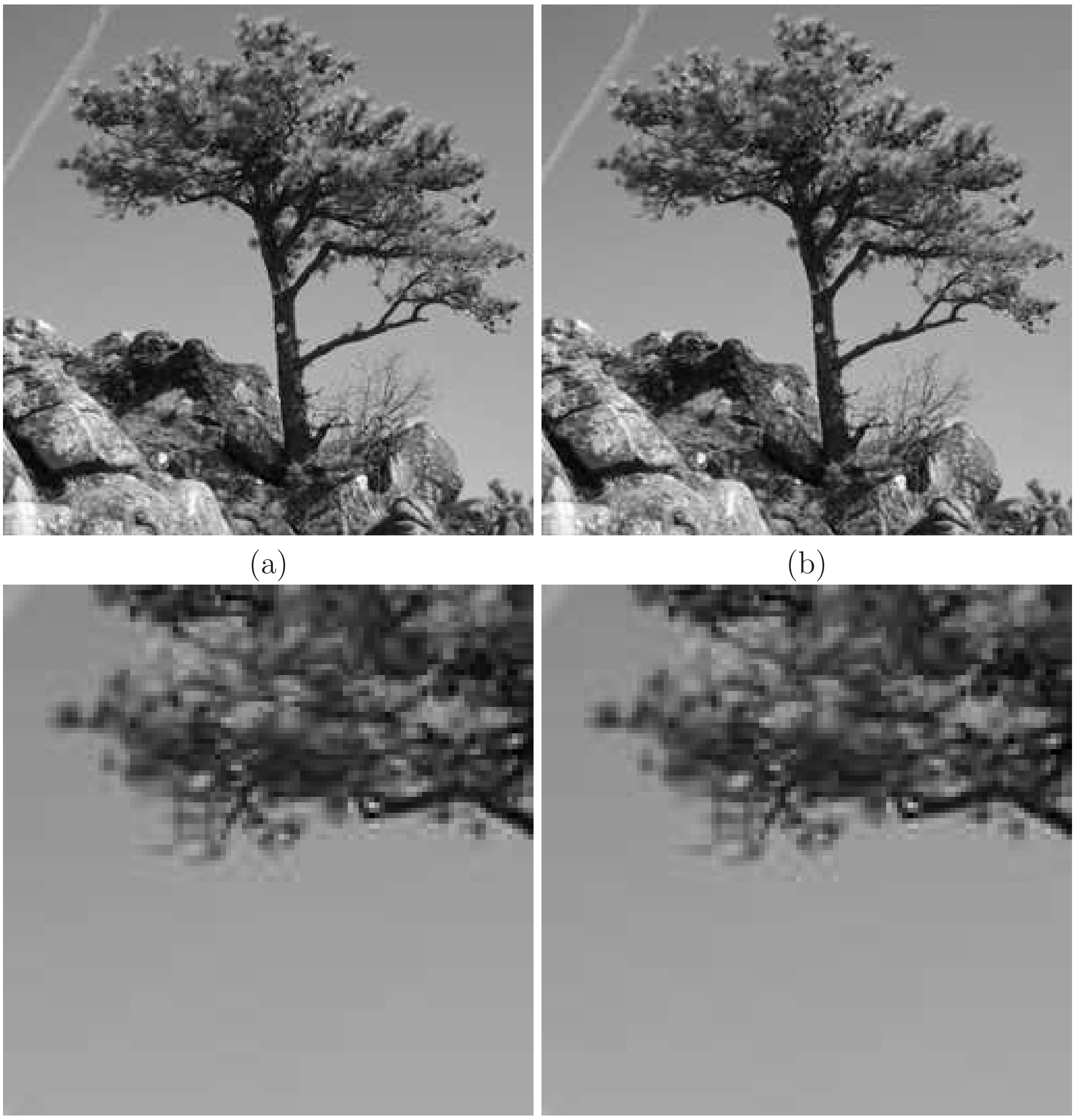


Figure 2.13: Acceleration by extrapolation

Results of extrapolation are shown in figure 2.13. The extrapolation procedure operates using four successive points in the pyramid coefficient domain which correspond to two complete iterations of the reconstruction algorithm. Extrapolation could be performed every after two iterations, however it is advantageous to allow the reconstruction algorithm to "relax" for a few iterations before extrapolating again. Performing extrapolation after every four complete iterations (corresponding to 8 alternating projections) was empirically found to give good results.

The benefits of imposing average power at each scale and performing extrapolation are cumulative, so the fastest reconstruction algorithm uses both techniques. Using this method, approximately 30 iterations are necessary before the resulting image is visually indistinguishable from the original.



(a) (b) (c) (d)

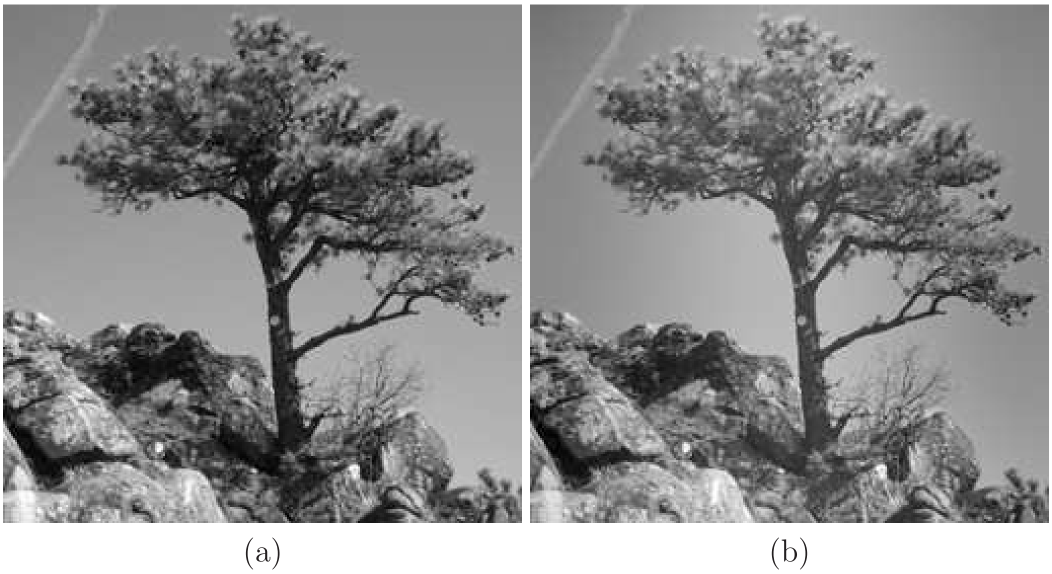
Figure 2.14: Reconstruction without highpass residual. (a) Original 256x256 pixel image. (b) Reconstruction from orientations without highpass band. (c) Detail of original image (a). (d) Detail of image (b)

2.5 Importance of residual scalar bands

While this chapter claims to present an image representation based on orientation measurements, exact reconstruction depends on imposing the highpass and lowpass residual bands which are not measurements of this type. It would undermine my claim that this representation is based on purely geometric measurements if a significant amount of the visual structure was represented simply by imposing the residual bands. As the highpass band has the same number of coefficients as original image pixels, imposition of this band is especially troublesome.

However, for most natural images relatively little information is carried by the highpass band. Natural images typically have power spectra that decay like $\frac{1}{|\omega|^p}$ with p near 2, and often have little power at the spatial frequencies captured by the highpass residual filters. Good quality images can be reconstructed from the orientation representation without imposing the highpass residual. Instead of imposing the highpass at each step, the operator O_Θ leaves the highpass unchanged, except for the very first iteration when the highpass is set to zero. Letting highpass "float" in this way is equivalent to enlarging the set W_Θ to include the entire space H. The algorithm may still be understood as alternating projection and still converges, although uniqueness is lost and the reconstructed image will depend slightly on the starting point. Results are shown in figure 2.14. As can be seen, the image reconstructed without imposing the highpass residual is almost indistinguishable from the original.

Imposing the lowpass band, however, is necessary for fixing the overall image range. If the lowpass bands are allowed to "float" as described above, directly

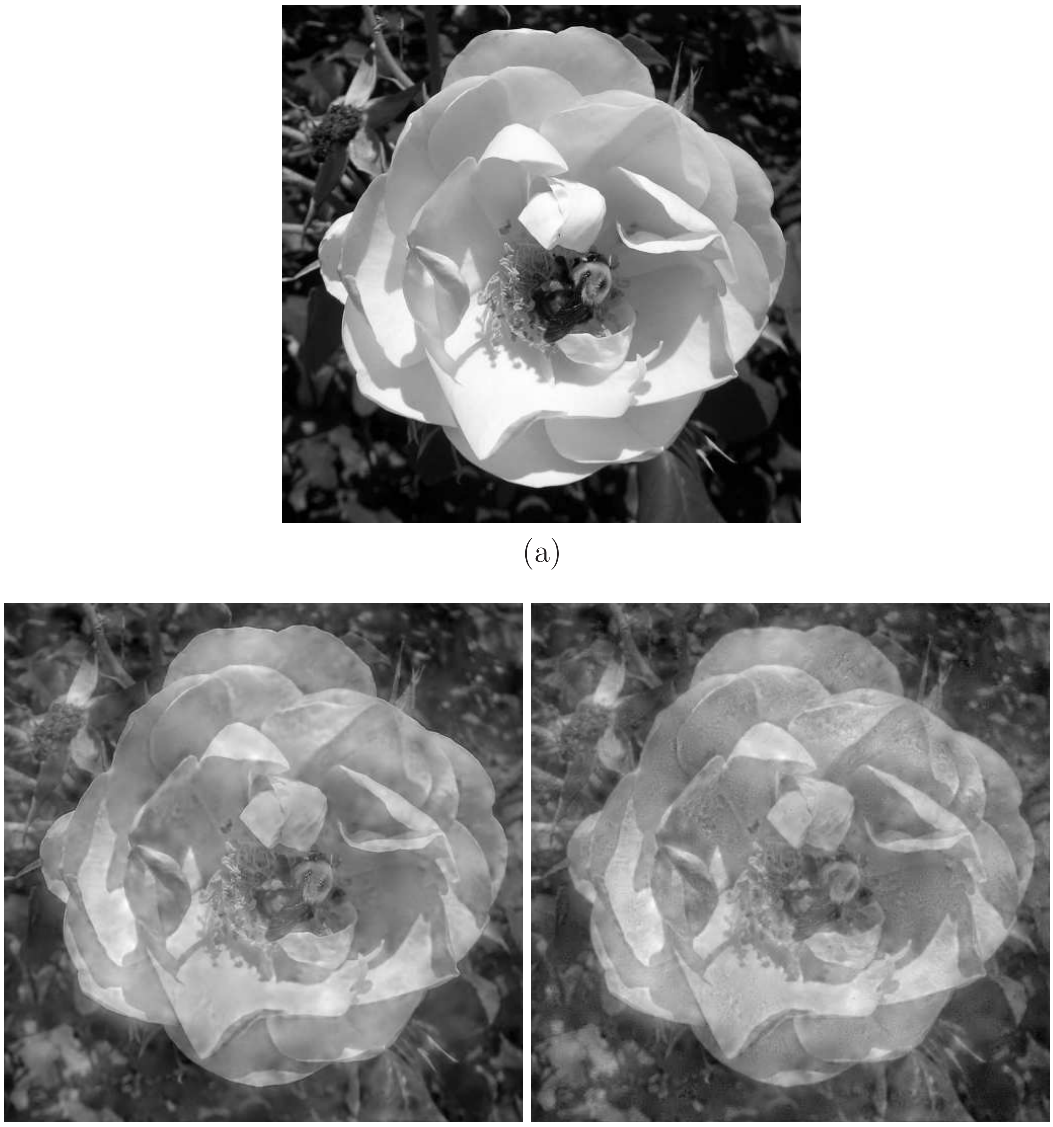


(a) (b)

Figure 2.15: Reconstruction without lowpass residual (a) Original (b) Reconstruction from orientations without lowpass band - range rescaled

applying the reconstruction algorithm without imposing average power at each scale will converge to an image whose dynamic range depends strongly on the starting point. This dynamic range ambiguity can be removed by imposing the average power at each spatial scale, however in this case the lowpass information in the reconstructed image will still depend on the starting point. The visual effect of having the lowpass band incorrect is much more noticeable, as can be seen in figure 2.15.

The lowpass residual thus cannot be discarded for reconstructing good quality images. The number of scalar coefficients in the lowpass residual is small, as few as 16, if the steerable pyramid is constructed to the maximum number of possible spatial scales. The need to impose the lowpass band is thus less troubling for the overall representation.



(a)

(b)

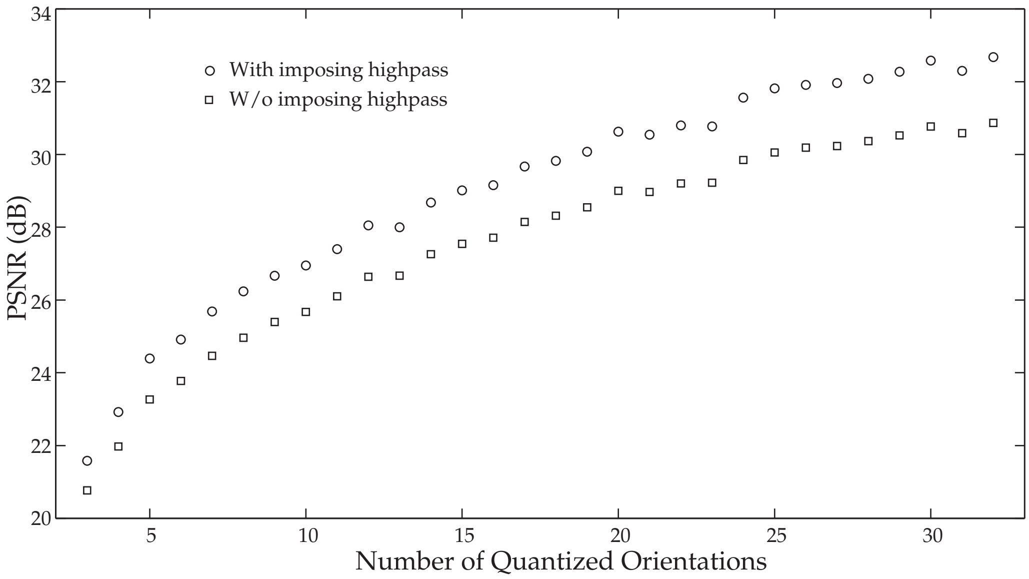
(c)

Figure 2.16: Reconstruction from quantized orientations. (a) Original Image (b) Image reconstructed from orientations quantized to 3 orientations, with imposing highpass (c) Same as (b), without imposing highpass

2.6 Quantization of Orientations

The success of the alternating projection algorithm as described above relies on the existence of a point of intersection of U and W_Θ. While this is ensured for sets of orientations that arise from actual images, it may not be true for sets of orientations that have been altered in some way. This could be troubling for using this representation for image processing tasks in which one would manipulate the orientations and then recover the processed image. If the representation were not stable to perturbations in the orientations, this would undermine its potential utility. As a method of exploring this, I have studied the effects of quantization of the orientations, and found the representation to be well behaved.

For quantization to Q values, the unit circle was divided into Q equal bins of width $\frac{2\pi}{Q}$ with bin centers at $\frac{2\pi q}{Q}$ for $q = 0...Q-1$. The quantized orientation bands $\Theta_s^Q(m,n)$ were formed by replacing the value of $\Theta_s(m,n)$ by the angle of the center of its corresponding bin. When reconstructing from quantized orientations, the orientation imposition step was modified to impose the quantization bin, rather than the bin center value. At each step, orientations that lie within $\frac{2\pi}{Q}$ of the quantized value are left unchanged, those that are outside are pushed to the closest edge of the bin, either $\Theta_s^Q(m,n) + \frac{\pi}{Q}$ or $\Theta_s^Q(m,n) - \frac{\pi}{Q}$. This may still be interpreted as projection onto a convex subset $W_{\Theta^Q} \subset W$, but where W_{Θ^Q} is a Cartesian product of "wedges" rather than rays in W. With this modification to the reconstruction algorithm, as the orientations are quantized more and more coarsely down to three orientations, image quality degrades gracefully, as illustrated in Figure 2.17. Even at extremely course quantization down to



;

Figure 2.17: Reconstruction quality (measured by PSNR) as a function of number of quantization levels

only three orientations, visually reasonable images can be reconstructed (Figure 2.16). Direct imposition of the quantization bin centers was also attempted, but gave poor results at coarse quantization.

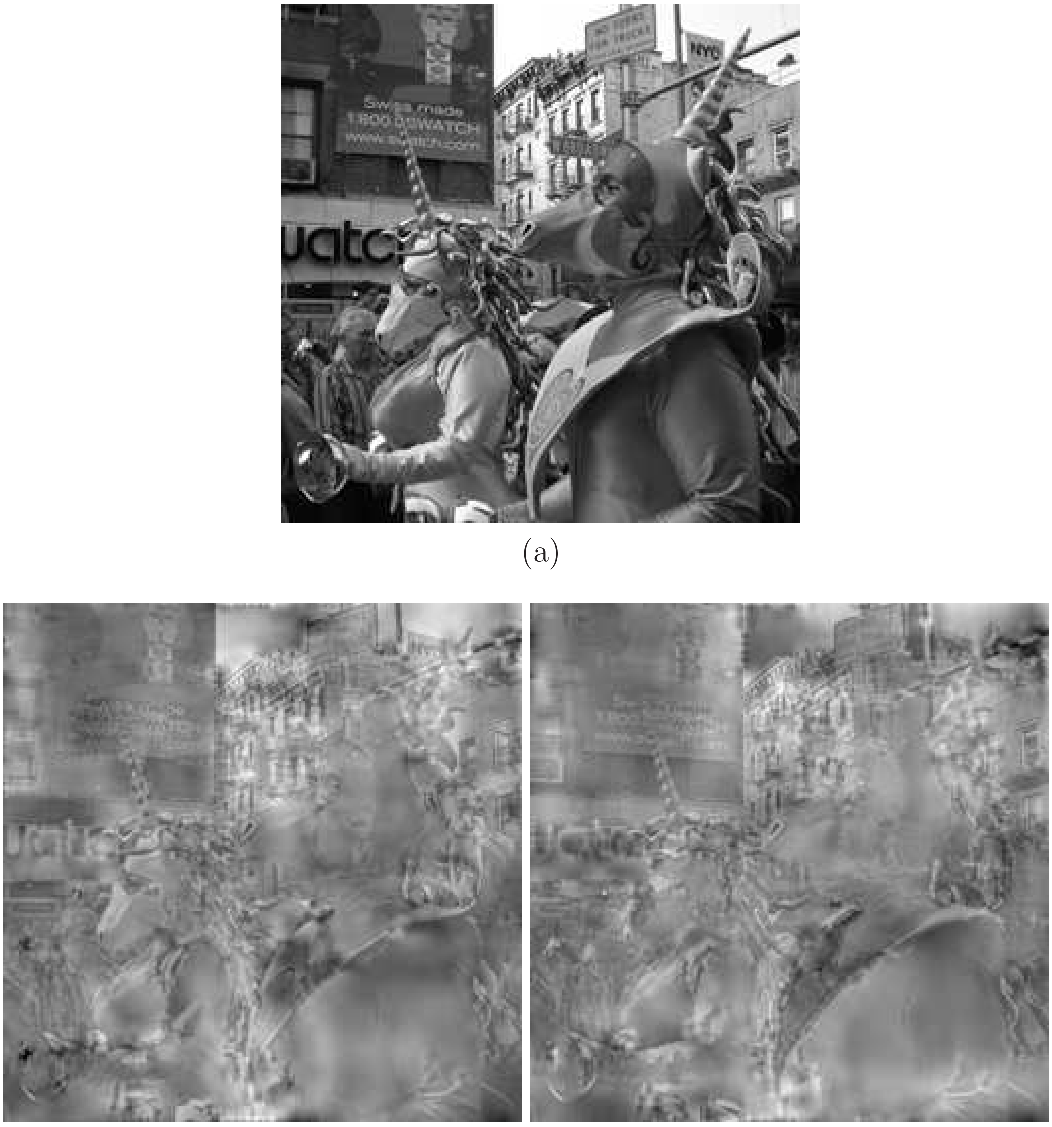
2.7 Imposition of Magnitudes

The previous sections have demonstrated the feasibility of reconstructing images from local multiscale orientation measurements, and described several variations on a practical algorithm for reconstruction. This work was motivated by several numerical experiments that suggested that image structure was more explicitly captured by orientations than by the magnitude information. These examples do not conclusively show that image structure cannot be recovered from the magnitudes, however. It is a natural question to ask what happens if one attempts to iterate a similar algorithm that instead imposes magnitude data at each step. This is related to work of Shams and Von der Malsburg, who studied reconstructing images from the magnitudes of an overcomplete set of Gabor filters [50].

Defining the operator $O_M : W \to W$ that imposes the specified highpass, lowpass and magnitude information, the analogous "magnitude reconstruction" algorithm is defined by iterating $AA^\dagger O_M$. On each two dimensional space W_i, the set of coefficients consistent with the specified magnitude M_i form a circle. O_M can be viewed as projecting onto a subset of M which is the direct product of these circles, which is definitely not a convex set. The magnitude reconstruction algorithm is thus not guaranteed to converge, and this failure to converge is typically observed in practice.

When initialized with a random starting point, the iterates typically converge in the image domain but not in the pyramid coefficient domain. After a large number of iterations the algorithm produces an image $x \in Im$ with $A^\dagger O_M Ax = x$, but $Ax \neq O_M Ax$. Intuitively speaking, the algorithm gets "stuck" and in the pyramid domain bounces between the two unequal points $O_M Ax$ and Ax. The resulting image produced is not unique and is highly dependent on the initial starting point. Some results (in the image domain) are shown in figure 2.18. While much of the visual structure such as edges are represented, the overall images are highly distorted. These images appear solarized, where the contrast is inverted in a patchwork fashion across the image.

The implications of this behavior must be interpreted carefully. As the specified magnitude constraints are not satisfied by the images typically produced by the magnitude reconstruction algorithm, one should not conclude that it is



(a)

(b)

(c)

Figure 2.18: Attempted reconstruction from magnitudes (a) Original Image (b),(c) Reconstructed images, starting from different random initial points

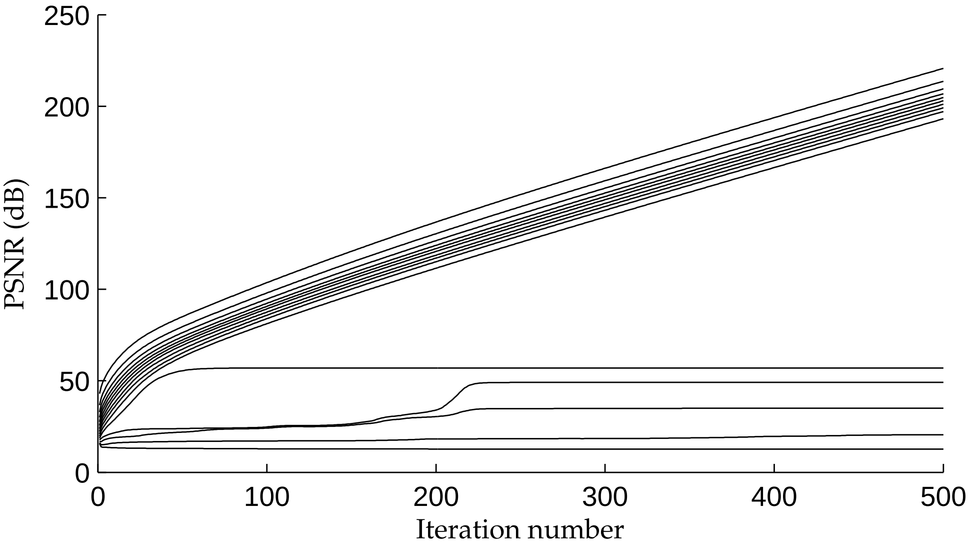


Figure 2.19: Reconstruction by imposing magnitudes with initial points successively further away in orientation domain, as described by equation 2.25. Each curve corresponds to a different value of δ, evenly spaced from $\frac{\pi}{16}$ to π in increments of $\frac{\pi}{16}$. The shift from convergence to failure to converge happens for δ^* between $\frac{11\pi}{16}$ and $\frac{12\pi}{16}$.

impossible to reconstruct or represent images from the coefficient magnitudes. The failure to reconstruct the original image by imposing magnitudes in this way may be viewed simply as a shortcoming of this particular reconstruction algorithm. It may still be that there is only one unique image consistent with the given magnitude data, but a more sophisticated algorithm that avoids getting "stuck" is needed to find it.

Some partial evidence that the magnitude constraints may be sufficient to constrain the image is provided by the following numerical experiment. If the starting point $x_{init} \in W$ of the magnitude imposition algorithm is chosen to be "close" to p, the transform of the original image, then iterative imposition of magnitudes was found to converge successfully to p. This implies the existence of a local “basin of attraction” in the pyramid domain where magnitude imposition is successful. The initial point for these simulations were generated in the following way. Let M_i and θ_i be the magnitudes and orientations for the original image. The initial point was generated by keeping the original magnitudes and perturbing the orientations by

$$\theta_i^0 = \theta_i + \delta n_i \tag{2.25}$$

where $n_i \in [-1, 1]$ was a uniformly distributed random variable. δ controls the width of the “orientation noise”. x_{init} was then generated using the M_i's and θ_i^0's. By increasing δ the distance to p was changed. For small values of δ, iterative imposition of magnitudes resulted in convergence to p. For a fixed noise pattern n_i, convergence is observed for δ below a critical value δ^*, above which the algorithm did not converge to p. Typical values of δ^* were around 2 radians.

The magnitude data would fail to represent an image uniquely if there existed a set of pyramid coefficients distinct from those of the original image that had the same magnitudes and were consistent with the constraint of having coming directly from an image, i.e. were in the subspace U. As the magnitudes would be the same for such a set of coefficients, they could be described by the parameterization implicit in 2.25 for some orientation noise sample. The fact that a basin of attraction was observed thus indicates that the magnitudes may uniquely specify the image. As the number of magnitudes is greater than the number of degrees of freedom in the original image (by a factor or 4/3), this is certainly possible. Nonetheless, the difficulty of this reconstruction from magnitudes should be contrasted with the relative ease of reconstructing from orientation information. It is thus a fair claim to say the image structure is more explicitly captured by the local orientations than by the magnitudes.

2.8 Analysis of convergence rate

The orientation imposition algorithm exhibits exponential convergence to its fixed point. Close inspection of the PSNR vs iteration number curves in figure 2.11 reveal asymptotically linear behavior after an initial superlinear transient. In the asymptotic region, linear increase in PSNR corresponds to exponential decay of the distance between the current iterate and the ultimate fixed point. As the SP transform is a tight frame and thus preserves distances, this decay in error is identical in both the image domain and the coefficient domain.

For the algorithm based on projection without any additional acceleration, a lower bound on this asymptotic convergence rate may be calculated directly from the image data. This calculation relies on looking at the dynamics of the alternating projection algorithm near the fixed point. The dynamics may first be "homogenized" by translating the fixed point to the origin, after which the orthogonal projections are linear operators. The decay in error at each step is then bounded by a term depending on the largest eigenvalue of an operator corresponding to two successive projections of the homogenized system. This term also has a geometric interpretation, related to the minimum angle between the two convex sets at the fixed point.

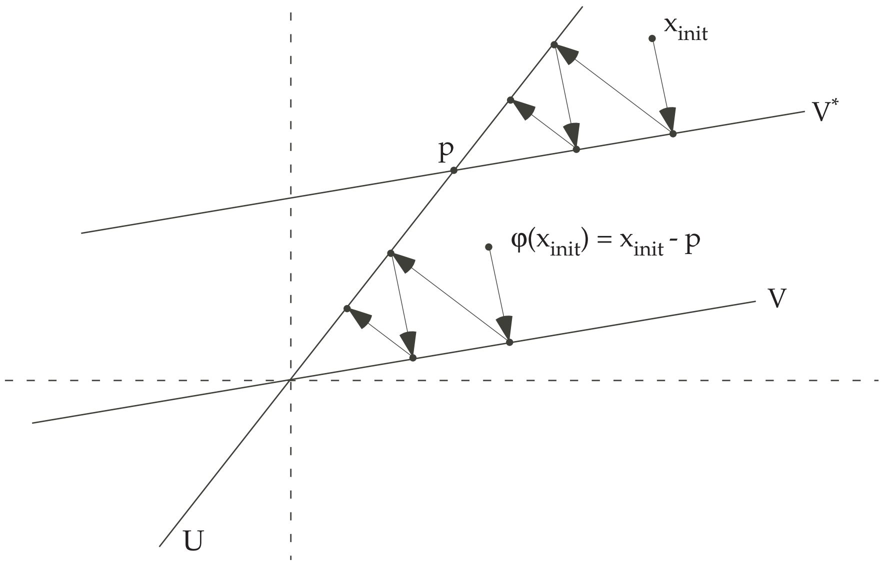


Figure 2.20: Homogenized dynamics - orbits of alternating projection onto U and V^* can be mapped onto orbits of alternating projection onto U and V

2.8.1 Homogenized Dynamics

Let the sets U and W_Θ be as defined in section 2.3. The set U is a linear space. As defined, W_Θ is not a complete hyperplane as the magnitudes defining it in equation 2.16 are non-negative. Define the set V^* to be the complete hyperplane containing W_Θ, i.e. V^* consists of x such that

$$x = v_{res} + \sum_{j=1}^{m} b_j v_j(\theta_j) \tag{2.26}$$

where $b_i \in \mathbb{R}$.

I will focus on the dynamics of alternating projection onto U and V^*. Let $p \in W$ be the steerable pyramid coefficients corresponding to the transform of the original image. By definition then $p \in U \cap W_\Theta$. If none of the magnitudes corresponding to the pyramid coefficients of p are exactly zero, then V^* and W_Θ

are identical in some neighborhood of p. The asymptotic rate of convergence for alternating projections onto U and V^* will then be the same as for alternating projections onto U and W_Θ.

Let $V = \text{Span}\{v_i(\theta_i)\}$ be the "homogenized" version of V^*. V is a linear space parallel to V^*. As illustrated in figure 2.20, there is a one to one correspondence between the orbits of alternating projections onto U and V^* and the orbits of alternating projections onto U and V.

To see this, first introduce the following abuse of notation. Let $\{u_i\}_{i=1}^n$ be a set of spanning vectors for the space U. Assume that the sets $\{u_i\}$ and $\{v_j\}$ are orthonormal in W. Denote also by U and V the matrices

$$U = (u_1, u_2, ..., u_n) \tag{2.27}$$

$$V = (v_1, v_2, ..., v_m) \tag{2.28}$$

formed by taking the spanning vector sets as columns, so $U : R^n \to W$ and $V : R^m \to W$.

Claim 1 : With this notation, orthogonal projection onto V^* may be computed by

$$P_{V^*}x = VV^\dagger(x-p) + p \tag{2.29}$$

Proof: As $V^* = \{Vb+p : b \in R^m\}$, the orthogonal projection of x onto V^* is given by $Vw + p$ where $w \in R^m$ minimizes $||x - (Vw+p)||_W$. This is the same as w minimizing $||(x-p) - Vw||_W$, which is the problem of projecting $x-p$ onto V. Orthogonal projection of x-p onto V is given by $VV^\dagger(x-p)$ (see Appendix 5). So then $Vw = VV^\dagger(x-p)$ and so $P_{V^*}x = Vw + p = VV^\dagger(x-p) + p$. ■

Orthogonal projection onto U is given by $P_U x = UU^\dagger x$. Let $\phi(x) = x - p$. This map establishes the desired correspondence between orbits. Precisely stated, this is

Claim 2: $\phi((P_U P_{V^*})x) = (P_U P_V)\phi(x)$

Proof : As $p \in U$, $UU^\dagger p = p$. Then compute

$$\begin{aligned} \phi(P_U P_{V^*} x) &= UU^\dagger(VV^\dagger(x-p)+p) - p \\ &= UU^\dagger VV^\dagger(x-p) + UU^\dagger p - p \\ &= UU^\dagger VV^\dagger(x-p) \\ &= P_U P_V \phi(x) \blacksquare \end{aligned} \tag{2.30}$$

After k iterations of the alternating projection algorithm, the residual error is given by $E_k = \left|\left|(P_U P_{V^*})^k x - p\right|\right|_W$. According to the above calculations, this is the same as $\left|\left|(P_U P_V)^k (x-p)\right|\right|_W$. Thus the decay in error is determined by the dynamics of iterating the operator $M = UU^\dagger VV^\dagger$.

These dynamics are determined by the eigenvalue spectrum of the operator M. As it is composed of projection operations, M cannot have eigenvalues greater than 1. Let λ^* be the largest magnitude eigenvalue of M, so that the operator norm $|M| = \lambda^*$. Then

$$E_{k+1} = \left|\left|MM^k(x-p)\right|\right|_W \leq |M| \left|\left|M^k(x-p)\right|\right|_W = \lambda^* E_k \tag{2.31}$$

from which it follows that $E_k \leq (\lambda^*)^k E_0$. The signal to noise ratio for the k$^{\text{th}}$

iterate is given by

$$\begin{aligned} SNR(k) &= 10\log_{10}\left(\frac{||p||_W^2}{E_k^2}\right) \\ &\geq 20\log_{10}\left(||p||_W\right) - 20\log_{10}\left((\lambda^*)^k E_0\right) \\ &\geq C - 20\log_{10}(\lambda^*)k \end{aligned} \tag{2.32}$$

where $C = 20\log_{10}\left(\frac{||p||_W}{E_0}\right)$. Thus a lower bound on the asymptotic slope of the SNR versus iteration curve is given by $-20\log_{10}(\lambda^*)$.

This bound is meaningless if $\lambda^* = 1$. In fact, the presence of a unit eigenvalue implies that convergence is not unique, i.e. that the homogenized sets U and V intersect at more than one point.

Claim 3 : M has a unit eigenvalue if and only if $U \cap V$ is nontrivial.

Proof : Assume there exists such an eigenvector y with $P_U P_V y = y$. I wish to show $y \in U \cap V$. Clearly $y \in U$, so it suffices to show $y \in V$ which will be the case iff $P_V y = y$. Writing $P_V y = y + w$, I must show $w = 0$. Applying P_U to both sides of this shows $P_U P_V y = P_U y + P_U w$ which reduces to $y = y + P_U w$, yielding $P_U w = 0$. Applying P_V to both sides of the same expression gives $P_V P_V y = P_V y + P_V w$ which reduces to $P_V y = P_V y + P_V w$, yielding $P_V w = 0$. So w is perpendicular to U and V. But as $w = P_V y - y$ is a difference of two vectors, one in V and one in U, W is in the span of U and V. These two statements imply that $w = 0$, and so $y \in U \cap V$.

Conversely, let $y \in U \cap V$ be a nonzero vector. Then $P_U y = y$ and $P_V y = y$, so clearly $P_U P_V y = y$ and so $M = P_U P_V$ has an eigenvalue equal to one. ■

If U and the homogenized hyperplane V intersect only at the origin, then U

and V^* can intersect only at a single point, namely p. This follows as the map $x \to \phi(x)$ is a one to one map between $U \cap V^*$ and $U \cap V$. Now, as $W_\Theta \subset V^*$, if $U \cap V^*$ is a single point then $W_\Theta \cap U$ can consist of at most one point. But $W_\Theta \cap U$ contains p, so it consists of exactly one point. All together, this proves the following

Theorem 2 *For a given image, if the operator M defined above has no eigenvalues equal to 1, then the representation by local multiscale orientations is unique. If the largest eigenvalue has magnitude $\lambda < 1$, then the reconstruction algorithm based on projections will have an asymptotic rate of convergence bounded below by $-20\log_{10}(\lambda)$ dB / iteration.*

2.8.2 Computation of eigenvalues

The problem of computing the eigenvalues of $M = UU^\dagger VV^\dagger$ may be first reduced to computing the eigenvalues of the smaller matrix $T = U^\dagger VV^\dagger U$. This follows from

Claim 4 : Any nonzero eigenvalue λ of M is also an eigenvalue of T.

Proof : Let y satisfying $UU^\dagger VV^\dagger y = \lambda y$ be an eigenvector of M. Then $x = U^\dagger VV^\dagger y$ satisfies

$$\begin{aligned} Tx = U^\dagger VV^\dagger U(U^\dagger VV^\dagger y) &= U^\dagger VV^\dagger(UU^\dagger VV^\dagger y) \qquad (2.33) \\ = U^\dagger VV^\dagger \lambda y &= \lambda x \end{aligned}$$

so x is an eigenvector of T with eigenvalue λ, provided that $x \neq 0$. But if $x = 0$

then $U^\dagger V V^\dagger y = 0$ and so applying U gives $U U^\dagger V V^\dagger y = \lambda y = 0$, so $\lambda = 0$. But by assumption λ is nonzero. ■

The operator T is still an extremely large matrix. It is a square matrix with dimensions equal to the number of image pixels in the original image. For reasonable sized images, it is impossible to form T explicitly in computer memory. Fortunately, it is straightforward to use the structure of T to compute its action on any vector x without forming the full matrix in memory. The largest magnitude eigenvalue can then be computed numerically by iterative methods.

We have $T = U^\dagger V V^\dagger U$. The vectors $v_i(\theta_i)$, the columns of V , are extremely sparse and have only two nonzero entries corresponding to the x and y filter locations for the i$^{\text{th}}$ space/scale location. Thus both V and $V^\dagger$ can be applied to vectors without allocating memory for their full size. The columns of U may be taken to be the steerable pyramid basis functions for the bandpass bands, so that applying U is equivalent to taking the partial SP transform (without highpass and lowpass bands). As these columns are then orthonormal in W by construction, $U^\dagger$ corresponds to the partial inverse SP transform, again dropping the highpass and lowpass contributions. These can be computed without explicitly forming the matrices in memory, as described in section 2.1.

As an example, these eigenvalues were computed for the T matrices for a set of 32 x 32 image patches extracted randomly from a set of natural test images. The lower bound asymptotic convergence rates from theorem 2 were computed, and compared against the convergence rates calculated by performing the orientation reconstruction using the projection method. The measured asymptotic convergence rates were calculated by a least squares linear fit to the SNR vs

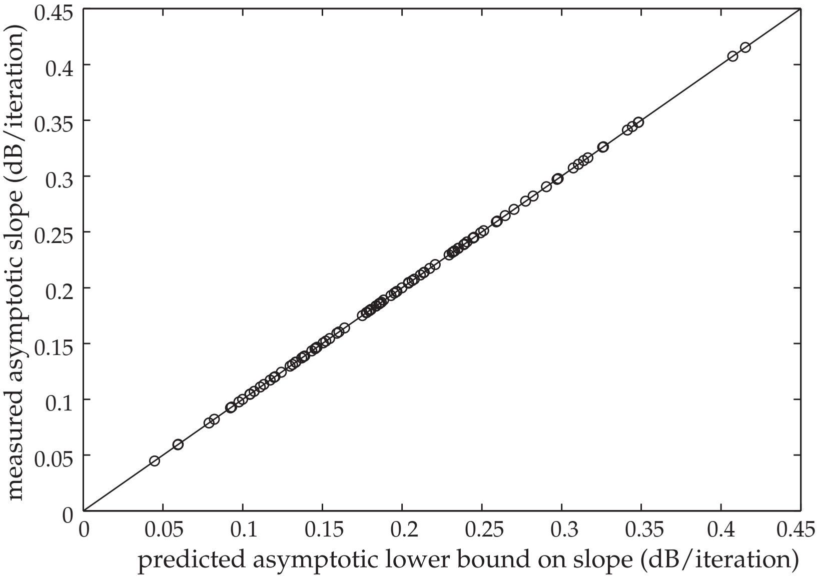


Figure 2.21: Measured asymptotic convergence rates for orientation reconstruction for 100 32x32 image patches versus predicted lower bound

iteration curves excluding the first 100 iterations to allow for decay of the initial superlinear transient. Results are shown in figure 2.21. As can be seen, the predicted asymptotic rates fit the measured convergence rates extremely well. Eigenvalues were computed using the MATLAB eigs routine, which is based on the ARPACK library and implements an iterative method based on Arnoldi iteration [25].

Chapter 3

Stochastic Modeling of Images

Understanding the structural content of natural images is important both for developing effective image processing methods and for the scientific study of vision. Very generally speaking, one can view both biological visual systems and image processing algorithms as systems that accept image signals as their inputs. A key idea is that the input images presented to these systems are typically not random signals, but some restricted subset of images with definable statistical properties. Both man-made and natural systems can benefit from adapting their design to fit the statistical and structural properties of the input signals they receive. In designing image processing algorithms, this adaptation is explicitly engineered. For biological systems, the appropriate adaptation has been performed by evolution over millions of years. Better understanding of the properties of natural images can thus provide insight into the organization of biological visual systems, and guide the development of better image processing algorithms.

When implementing a typical image processing task, such as compression or denoising, good performance is desired for the images that one is likely to encounter in practice. If the input images are not completely arbitrary but have some well defined structural and statistical properties, it is possible to adapt the algorithm to take advantage of these signal properties. By concentrating on performing well on a smaller, more relevant part of the input space of all possible images, one can develop more effective processing algorithms. A simple example of this idea is provided by coding and information theory. Imagine encoding 8 bit greyscale images that were formed by drawing each pixel value randomly from a uniform distribution. Results from information theory show that the number of bits needed to encode such a signal exactly is equal to the entropy, which in this case is maximal and equal to 8 bits per image pixel. However, lossless compression algorithms operating on typical photographic images are able to encode images with fewer than a third as many bits [65]. This is only possible as the entropy of natural images are far lower than of completely random signals. This statement is equivalent to saying that the signals of interest have specific structural regularity that can be exploited.

In order to exploit the properties of natural images for image processing tasks, one must have a quantitative description of what they are. Loosely speaking, an image model is a way of answering the question "what is a natural image?". Many image models can be described as either deterministic or stochastic. Deterministic models essentially answer the question of whether a given two dimensional function is a natural image with a yes or no answer. Many deterministic models are based on function spaces, classifying images as functions that have finite or small values of a particular norm. Common examples of these models include the space of Bounded Variation (BV) functions and spaces based on the Mumford-Shah energy functional [47, 37]. Related deterministic ideas include piecewise smooth models with differentiable boundary segments, such as functions that are C^2 except along twice-differentiable boundaries, and "cartoon plus texture decomposition" models. [8, 35].

In contrast, stochastic image models are based on the idea that an image may be treated as a random variable. In this framework, the act of taking a photograph may be viewed as drawing a particular sample of a random process. If one is considering discretized digital images of a fixed size with n image pixels, the most general form of a stochastic model is a complete probability distribution $p(x)$ for $x \in \mathbb{R}^n$. Here $p(x)$ gives the probability that x will be observed if one takes a photograph. Depending on the methods used, however, building such a global probability distribution over the entire image space may prove to be extremely difficult.

An alternative approach is to build stochastic models that focus on the local statistical properties of images. A very common approach in signal modeling is to describe the distribution not of the original pixel values, but rather the statistics of the coefficients of some transform of the image. One can think of splitting the construction of probability model $p(x)$ for images into separately answering the questions "what is x" and "what is p". Choosing the space for x typically implies picking a particular image transform to measure the statistics of, and deciding the dimension of sets of coefficients to model. Once this space is fixed, the functional form of the distribution p should be parameterized in a manner that balances the ability to faithfully capture the behavior of the data, but is also tractable.

Many of the models used in image processing may be described by the selection of these two components. The classical image power spectrum models pick the space for x to be the coefficients of the Fourier transform of the image, and p to be Gaussian. The Fourier basis is highly nonlocal, however, which limits the power of processing algorithms based on these models. The discrete cosine transformation (DCT), which may be viewed as performing the Fourier transform on discrete image blocks, is a commonly used transform in image processing. Image models based on describing the one-dimensional marginal statistics of DCT coefficients using the generalized Gaussian distributions have been studied since the early 1980's [5, 44]. These generalized Gaussian distributions take the form

$$p(x) \propto e^{-|\frac{x}{s}|^{\alpha}} \tag{3.1}$$

where s and α are free model parameters. This form gives the Gaussian distribution when $\alpha = 2$ and the Laplacian distribution when $\alpha = 1$. For these models, DCT coefficients at distinct locations are treated independently and the joint statistics are not captured.

While the models based on the DCT transform capture local information, they describe the statistics of an image at a fixed spatial scale determined by the image block sizes. However, natural images display strong scale-invariance properties. As objects in the natural environment are just as likely to be photographed at any distance from the camera, to a good approximation individual image features are likely to occur at a wide range of scales with equal probability.

A natural way of respecting the scale invariant statistical properties of images is by building the model in the space of coefficients of a multiscale transform. Over the past 20 years, a variety of wavelet and related multiscale transforms such as steerable pyramids, curvelets, contourlets and bandlets have seen widespread use for image processing. The basic idea underlying all of these methods is that the image signal can be decomposed as a sum of individual "atoms" or "basis functions" that are all scaled and translated versions of a small number of "mother wavelet" functions. Typically, the scaling parameter is sampled at powers of two, so that the basis functions at each spatial scale may be grouped together into discrete subbands. Given such a multiscale transform as a front end, modeling the statistics of each of the subbands in a uniform way generates a scale-invariant image model.

A significant volume of research has studied image models based on the marginal statistics of wavelet coefficients. It was observed early in the development of wavelet theory that natural images display distinctively "sparse" marginal statistics when decomposed with orthogonal wavelet transforms. In this context, sparse marginal responses mean that most of the filter coefficients are zero or very small, but occasionally very large coefficient values occur. Distributions characteristic of these responses have strong peaks at the origin but have heavier tails as compared to the Gaussian distribution. Mallat described these marginal histograms with generalized Gaussian functions and discussed implications for image coding [32]. Similar models were discussed in the context of image coding with biorthogonal wavelets by Antonini et al [1]. Moulin and Liu studied the connection between denoising shrinkage functions implied by using Maximum a Priori estimation with such generalized Gaussian priors and those based on the minimum description length principle [36]. Marginal statistical models have also been used with overcomplete linear transformations. Simoncelli and Adelson studied image denoising based on a generalized Gaussian description of marginal statistics of Steerable Pyramid coefficients [55].

While image models based on marginal statistics are appealing due to their relative simplicity, they are inherently limited by their inability to capture statistical relationships between nearby coefficients in space and scale. Marginal statistical models make the implicit assumption that different transform coefficients are independent. However, wavelet coefficients from nearby image regions have strong statistical interdependencies that should not be ignored. A number of authors have studied the joint statistics of pairs of wavelet coefficients and found significant deviations from independence [23, 6, 2]. These effects arise due to the localized features present in images. Image features leading to strong filter responses, such as edges and lines, tend to be localized and lie along oriented contours. Intuitively speaking, the presence of a large amplitude coefficient in a particular region indicates the presence of such a local image feature, which will likely lead to large coefficient responses for filters nearby in space, scale and orientation. This "clustering" of large magnitude coefficients near image signal features cannot be described or exploited by models that assume independence of different coefficient responses.

These dependencies are driven by the structure of the underlying image data and not in general by the correlation of the wavelet filters themselves. Indeed, orthogonal wavelet filters are uncorrelated and would therefore give completely independent responses to white Gaussian noise inputs. It has also been observed that for natural images, the responses to orthogonal wavelet filters are empirically uncorrelated [20]. Zero correlation does not imply independence, however, for random variables with non-Gaussian statistics. For models based on overcomplete transformations such as the steerable pyramid, the filters will no longer be uncorrelated. In this case there is a mixture of statistical dependence between filter responses arising from both the filter correlation and the underlying signal structure.

Many modeling applications in image coding rely on making predictions of the image signal based on previously observed information. Wavelet filter responses are zero mean, assuming both positive and negative values. While the lack of correlation implies that models based on linearly predicting coefficients based on their neighbors will fail, this is essentially due to the inability of predicting the sign of the coefficients. Taking the absolute values of the coefficients removes this ambiguity. Attempting only to predict the absolute value of coefficients based on the absolute values of neighboring coefficients in space and scale is feasible. Training such a predictive model is one implicit method of representing the dependencies between nearby wavelet coefficients. This type of predictive model of absolute values of coefficients has been used for image coding, denoising and digital forensics applications [6, 9, 28].

Constructing image models that take advantage of the statistical dependencies between neighboring transform coefficients in a principled way is a challenging problem. A major issue is the massive interconnectedness of neighboring coefficients across the entire wavelet domain. One may attempt at first to construct a model capturing the dependencies only between immediately neighboring coefficients. However, the neighbors of one coefficient themselves have neighbors. Following these connections, the statistical dependencies propagate throughout the wavelet coefficient domain and one is led down the path of constructing a global model that may prove intractable to work with. This is one example of the so-called “loopy graph” problem.

Following these ideas naturally leads to the concept Markov random field models [66]. These models are based on the notion of conditional independence - the idea that a single field value is independent of other coefficients when conditioned on some local neighborhood of data. Markov random fields may be used to directly model image pixel intensities or transform coefficient values. However, for many applications the random field consists not of the image data directly, but some quantity representing the local image context. Malfait and Roose develop a Markov random field model of this type where the random field values indicate whether each wavelet coefficient is dominated by noise or by the desired signal [30]. Markov random field models are commonly used for image modeling, but can lead to very computationally intensive estimation methods.

Describing images stochastically offers several advantages for developing certain image processing techniques. Given a stochastic image model, a number of image processing tasks can be tackled using the theoretical apparatus of statistical estimation. One important example of this is the image denoising problem, where one seeks to recover a clean version of an image that has been corrupted with a noise signal. In this problem, it truly makes sense given the physics of noise generation to view the noise as a random process. If the signal is also modeled as random variable, then the whole problem can be treated in a unified theoretical framework using probability theory. While denoising methods based on deterministic models have certainly been studied, as noise is not a deterministic process they do not describe the two components of the corrupted signal on the same footing. Other image processing tasks that can be based on statistical estimation methods include image inpainting, where one seeks to estimate missing image data, and image super-resolution where one attempts to estimate a high resolution version of an image given lower resolution information. Stochastic image models are also highly useful for image coding applications. For this problem, the well developed field of information theory provides a tight link between statistical signal models and coding performance. Strong theoretical results on coding performance are based on the signal entropy, which depends on the distribution of the input signal.

Another advantage of stochastic image models is their flexibility for describing different subtypes of image signals. There are many classes of images in addition to natural photographic images that are relevant for specialized image processing tasks. A large number of physical sensing systems produce data in image form. Examples of different imaging modalities include long range systems such as satellite data or synthetic aperture radar imaging to medical technologies such as magnetic resonance imaging, x-ray computerized tomography or ultrasound. Each of these types of images will have distinct properties that one may wish to describe. Many signal models contain adjustable parameters that may be fit from image data. The probabilistic framework provides well defined methods of fitting these parameters from data, such as by maximum likelihood estimation. This allows one to adapt such models to different signal classes, or even to the statistics of an individual image, in a principled way. This ability to "listen" to the image data itself is an important feature of many stochastic signal models.

An important issue for stochastic signal models is the balance between imposing structural assumptions on the model and fitting from data. As mentioned earlier, adapting the model to the particular signal or subclass of signals of interest is valuable. However, for most problems, some parametric structure must be imposed on the relevant probability density. For local stochastic image models that consist of more than a very small number of dimensions, fitting the probability density non-parametrically from data is infeasible due to the so-called "curse of dimensionality". This occurs as the number of data samples required to fit the density function to a reasonable accuracy grows exponentially in the number of data dimensions.

Such behavior is illustrated by straightforward histogram binning, perhaps the simplest form of non-parametric density estimation. Given N samples of a scalar random variable x taking values bounded on the bounded interval $[a, b]$, one may estimate the density of x by dividing the interval into a certain number of bins, counting the number of times x falls within each bin, and computing the sample histogram. As x is a random variable, repeating this process for N different samples gives a different answer. For the calculation to provide a useful answer, the variance of the sample histogram must be reduced to an acceptable amount. In general the variance for each bin will scale inversely with the expected number of data points falling into that particular bin. While this simple discussion ignores the important issues involved in picking the bin sizes and the bias/variance tradeoff involved, the general result is that the number of data points required to achieve a fixed level of variance will scale linearly with the number of histogram bins. For higher dimensional data, this scaling is disastrous as the number of bins required to maintain a fixed maximum bin diameter scales exponentially in the number of dimensions.

For these reasons, when modeling multidimensional image data some form of parametric assumptions are necessary. Even in the scalar case, parametric models are usually employed. The generalized Gaussian marginal models dis-

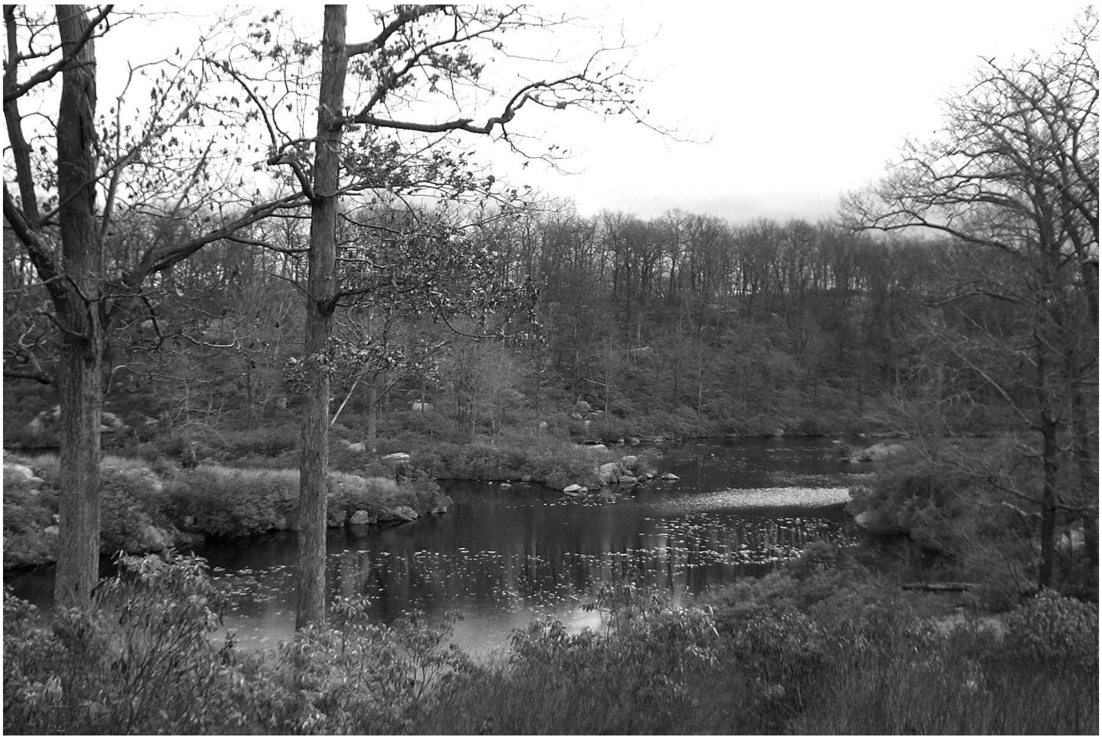


Figure 3.1: A prototypical "natural scene"

cussed previously are a common example of this. Introducing a parametric form for the density imposes constraints. While this is necessary to yield a model that may actually be fit from the data, one should take care to avoid imposing inappropriate constraints on the model. As a loosely stated general principle, one should impose only the constraints that are believed to be present in natural images, and allow the remaining model specification to be done by fitting to the available data. There should be a clear connection between each aspect of the functional form of the parameterized model and structural assumptions about the properties of natural images.

Understanding of the structural properties of natural images should motivate the development of image probability models. As an illustrative introduction to the image properties most relevant for the models developed in this chapter, consider the lake image shown in figure 3.1. One of the most important properties that distinguishes natural images from random noise is spatial inhomogeneity. The local properties of the image signal in smooth regions such as the sky or the still water on the lake are very different from strongly oriented edge regions, like those along the horizon or the tree branches, or textured regions such as the foreground bushes. One quantifiable aspect of this local inhomogeneity is variation in local signal power. Local signal power is measured by the magnitudes of responses of localized zero mean filters, and is thus related to local contrast not simply local pixel intensity. Smooth regions such as the sky and water have low local signal power, while edge and texture regions have higher local power. When images are expanded in a wavelet basis, these variations in local signal power often appear through clusters of large magnitude coefficients near salient image features. A second characteristic feature of natural images is that the majority of these local features are strongly oriented. These can be seen in the lake image as the edges formed between the sky and the treetops, between the water and shore, and the borders of the tree trunks themselves. There are also some texture regions that are strongly oriented, such as the patterns formed by the tree branches and trunks near the horizon. Oriented features occur at a variety of different orientations. If the image is analyzed using filters that measure the image gradient, one may compute the local orientation as a spatially varying function. This variation in orientation across different spatial locations is another significant aspect of local signal inhomogeneity.

Recognizing and accounting for the inhomogeneity in image signals is very important for a wide variety of image processing algorithms. As the image signal is different in different spatial regions, behavior of an algorithm that is appropriate in one region may be inappropriate in another. A large amount of recent research in image processing has focused on developing spatially adaptive algorithms. In the image coding literature, a number of algorithms based on explicitly measuring the "image context" have been described [61, 58]. In these works the "image context" refers to a set of spatially varying set of measurements that attempt to describe the current image signal properties. The coder will then modulate its behavior based on the current value of the image context. The key point which enables this approach to be successful for coding is that the local signal is simpler to describe when conditioned on the image context.

This idea may be extended to stochastic image modeling by parameterizing a local probability function in terms of image context variables. Variables used for the local image context should be able to describe the local image inhomogeneity. As mentioned above, two of the most important manifestations of image variability are the changing local signal power and local orientation. In this chapter, I use these ideas to develop a set of stochastic image models that can describe explicit adaptation to the local signal power and orientation. For the first such model, I explicitly parameterize the image context by a pair of hidden variables that correspond to the local contrast and local orientation. When conditioned on these local hidden variables, the signal description is Gaussian. In this way a complete model is constructed as a mixture of Gaussian components that each have a direct interpretation as representing specific local structural properties of the image.

Another significant property of natural images is scale invariance. The models developed in this chapter describe patches of Steerable Pyramid coefficients.

As the Steerable Pyramid is a multiscale transformation, the resulting model naturally treats different image scales in a uniform manner, respecting the image scale-invariance properties.

The work presented in this chapter is an extension of the Gaussian Scale Mixture (GSM) model originally developed by Wainwright and Simoncelli [63]. The development of the GSM model was motivated by the desire to capture the observed clustering behavior of large magnitude wavelet coefficients. This model describes patches of wavelet coefficients as samples from a single multivariate Gaussian that are then multiplied by a spatially varying hidden scalar variable that controls the local signal power. As nearby wavelet coefficients are controlled by the same hidden scalar variable, large values for the hidden variable in a given location will yield a cluster of large coefficients. In this way the GSM model can account for the observed correlations of coefficient magnitudes.

While the GSM model effectively describes variations in local signal power, it does not explicitly model the local image orientation. The models described in this chapter address this limitation of the original GSM. In this chapter I develop the Orientation Adaptive Gaussian Scale Mixture Model (OAGSM), which is similar in structure to the GSM but includes a hidden variable modeling local orientation. The OAGSM may be viewed as a generative model where patches of coefficients are first drawn from a single oriented multivariate Gaussian process. The patches are multiplied by a scalar hidden variable and then *rotated* by a hidden orientation variable. Much of the power of this model arises from the fact that oriented signals at different orientations in different image regions may be described by the same original oriented process. The GSM model without the orientation hidden variable tends to mix the description of oriented features at different orientations, which is avoided with the OAGSM model.

Although the OAGSM model is a good model for wavelet patches from oriented image regions, images also contain significant non-oriented regions. Low power non-oriented regions such as constant sky regions could still be modeled with an oriented process by simply setting the local magnitude close to zero. However many images do contain regions with significant local power but without a well defined local orientation, such as T-junctions or non-oriented texture areas. The OAGSM as described is an inappropriate model for wavelet patches coming from such regions. This problem is addressed by developing a related model that includes a non-oriented signal component. By introducing a third hidden variable that models the "orientedness" of the local patch, the OAGSM with non-oriented component (OAGSM/NC) model can be treated in a very similar framework as the original OAGSM. The general form of these models and issues involved in fitting parameters from data are described in this chapter. The primary application of the OAGSM model is for image denoising, which is discussed in detail in chapter 4.

3.1 OAGSM model

The models presented in this chapter are all models for local patches of wavelet coefficients. All of the work has been done using the Steerable Pyramid representation, which was described in detail in section 2.1. While the basic structure of the models could be used with any multiscale representation, many of the solutions to issues arising in fitting the model parameters made extensive use of the Steerability properties of the Steerable Pyramid. Throughout the rest of this chapter, the term "wavelet coefficients" will refer to the Steerable Pyramid unless otherwise specified.

The exact geometry of the local wavelet patches may be chosen in a number of different configurations. The Steerable Pyramid with K orientations produces K orientation subbands at each spatial scale. The wavelet patches considered in this chapter consist of coefficients that are "near" to a given central coefficient. The "nearby" coefficients need not be restricted to a single orientation subband, however. Patches may include "parent" coefficients from a coarser scale, "cousin" coefficients from different orientation bands at the same scale, as well as "sibling" spatial neighbors from the same subband. These options are illustrated in figure 3.2. As can be seen, there is a significant amount of freedom available in choosing the so-called generalized wavelet neighborhood of a single central coefficient. Including parent and cousin coefficients can allow the OAGSM to capture some of the cross-scale and cross-subband dependencies present in image data.

Selecting the appropriate size of the generalized wavelet patches is a model selection problem which can only be resolved empirically. If the neighborhood is too small, the model will fail to take advantage of the known statistical dependencies between nearby coefficients. In the extreme case of shrinking the neighborhood to a single coefficient, the OAGSM reduces to a wavelet marginal statistical model, similar to [55]. On the other hand, making the neighborhood too large will also lead to problems. The OAGSM is a local model based on the assumption that the hidden variables capture the current signal properties, so that when conditioned on these the signal may be described successfully with a single multivariate Gaussian. If the neighborhood size is too large, it may begin

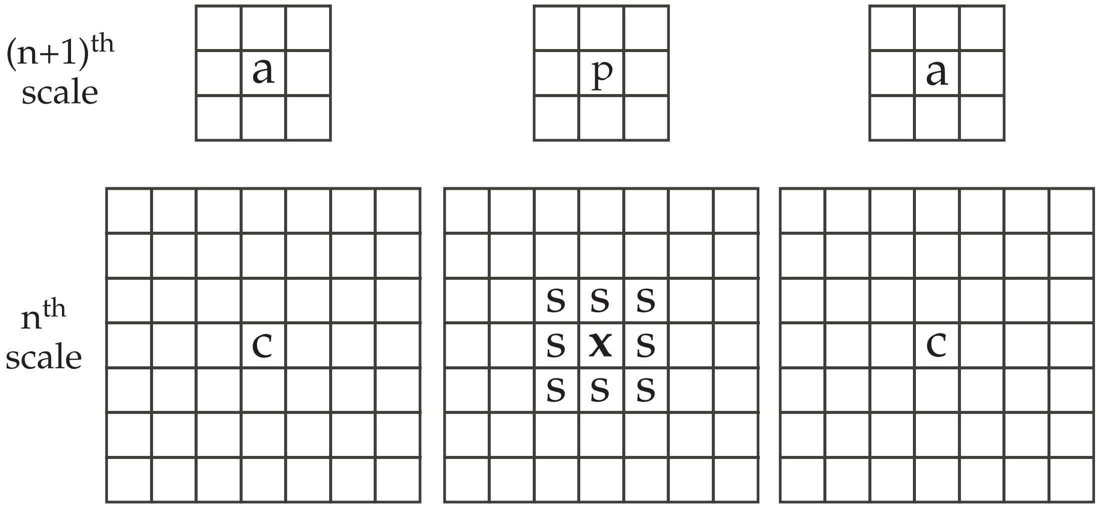


Figure 3.2: Generalized wavelet neighborhood family. The diagram indicates a possible generalized neighborhood for a coefficient at position (x) consisting of sibling (s), cousin (c), parent (p) and auntie (a) coefficients, for a Steerable Pyramid transform with three orientation subbands.

to contain signal with more than a single orientation, or with variations in power within the patch, such that this assumption breaks down. Increasing the patch size also increases the number of parameters that must be fit from the image data. Choosing the generalized neighborhood geometry thus involves a tradeoff between capturing the dependencies of neighboring coefficients, and between the correctness and complexity of the model. In general, this should be resolved empirically by choosing the model size that gives the best performance for the current application. Much of the basic theory of the model and techniques for fitting the parameters from data discussed in this chapter do not depend on the exact neighborhood geometry. A more complete investigation of the effects of neighborhood size will be investigated numerically in the context of image denoising in Chapter 4.

The OAGSM model may be described by the following generative process.

For each patch location, a pair of hidden variables z and θ are chosen according to some fixed prior density. Each patch v is formed by drawing a sample from a fixed multivariate Gaussian process, then rotating v by θ around the center of the patch and scaling by $\sqrt{z}$. This may be written as

$$v = \sqrt{z} R(\theta) u \tag{3.2}$$

where $R(\theta)$ is an operator performing rotation about the center of the patch, and u is a zero mean multivariate Gaussian with fixed covariance C_0.

Both the patch rotation operator $R(\theta)$ and scaling by $\sqrt{z}$ are linear operations. This implies that when conditioned on fixed values for the hidden variables, v is simply a linearly transformed Gaussian, and is thus itself Gaussian. In this case v will have covariance

$$zC(\theta) = zR(\theta) C_0 R(\theta)^T \tag{3.3}$$

which defines the "oriented covariances" $C(\theta)$ in terms of R and C_0.

The hidden variables are assumed to be independent of u. Given their prior $p(z, \theta)$, the density for v may be expanded as

$$\begin{aligned} p(v) &= \iint p(v|z,\theta) p(z,\theta) dz d\theta \\ &= \iint g(v; zC(\theta)) p(z,\theta) dz d\theta \end{aligned} \tag{3.4}$$

where I have introduced the notation

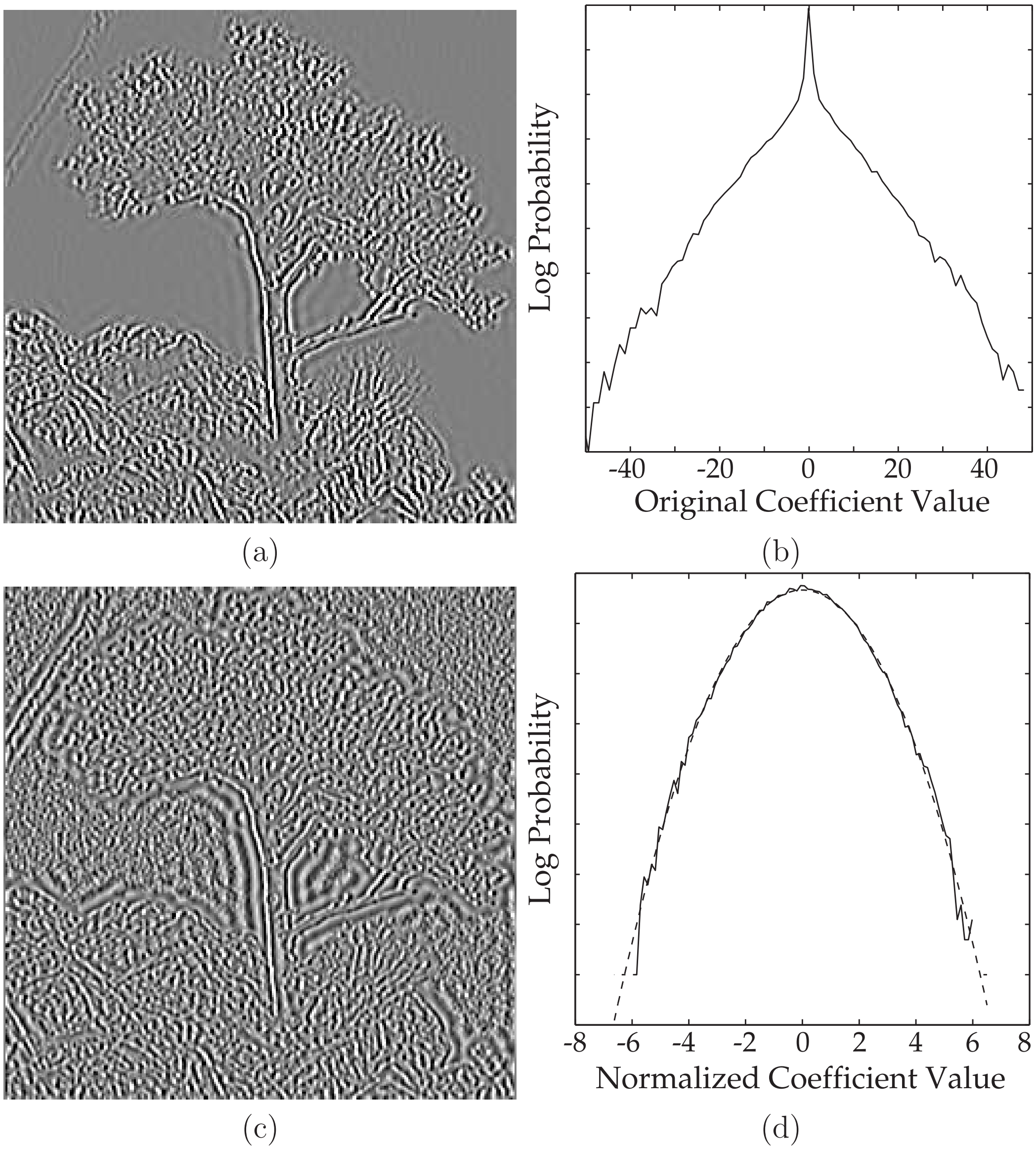
$$g(v; C) = \frac{1}{\sqrt{(2\pi)^n |C|}} \exp\left(-\frac{1}{2} v^T C^{-1} v\right)$$

is the zero multivariate Gaussian density with covariance C.

In general, the overall probability is an infinite Gaussian mixture. If a discrete density is chosen for the hidden variables z and θ, then it may reduce to a finite Gaussian mixture. This reduction to a finite mixture will typically be done when the model is used in practice. The form of the prior may be somewhat constrained by assuming rotation invariance for the model. It is reasonable to assume that orientations are equally likely to occur at any angle, and that there is no systematic dependence between local orientation and local power. While this rotation-invariance property may fail for certain classes of images, such as pictures of buildings which may have a bias for perfectly horizontal and vertical structures, it will be assumed here. This implies the use of a separable prior $p(z, \theta) = p(z)p(\theta)$ for the hidden variables, with $p(\theta)$ equal to the constant density on $[0, 2\pi]$. The density $p(z)$ remains to be specified. The exact choice for $p(z)$ for denoising applications will be discussed in section 4.1.2.

Once the hidden variable priors are fixed, equation 3.3 shows that the remaining model parameters may be specified either by giving C_0 and the exact form of the patch rotation operator $R(\theta)$, or by specifying the oriented covariances $C(\theta)$. While these two are formally equivalent, it is often more convenient in practice to specify $C(\theta)$.

The OAGSM model is based on the idea that the spatial inhomogeneity observed in natural image data can be explained by the action of the two scalar



(a) (b)

(c) (d)

Figure 3.3: Effects of divisive normalization. (a) Original subband (b) log-histogram of marginal statistics for original subband (c) Subband normalized by estimated $\hat{z}$ at each location (d) log-histogram for normalized coefficients, showing Gaussian behavior. Dashed line is parabolic curve fit to histogram, for comparison.

and rotator hidden variables. These spatially varying hidden variables are employed to modulate a single, homogenous Gaussian process, thereby producing the observed local variations. It follows that one prediction of the model is that if the action of these hidden variables could be "undone", the subsequent ensemble of transformed local image patches would appear to be more homogenous and closer to Gaussian. This behavior is observed in the following pair of numerical experiments which provide evidence for the validity of the OAGSM model.

As the z hidden variable acts by multiplication to control local contrast, undoing its action is equivalent to divisively normalizing local coefficients by an estimate of local power. It has been observed by several authors that transforming image data by divisive normalization yields marginal statistics that are closer to Gaussian [63, 46]. This is illustrated in figure 3.3 for one Steerable Pyramid subband. The original subband marginal statistics are far from Gaussian, as can be seen from examining the log histogram. A true Gaussian distribution would have an inverted parabola for its log histogram. The local estimate of the hidden variable $\hat{z}$ may be computed as follows. Temporarily ignoring the θ hidden variable, each patch x may be viewed as a sample of the Gaussian $g(x; zC)$ with covariance zC. As will be explained later in the chapter, C may be estimated for the entire band simply by taking the sample covariance of all of the extracted overlapping coefficient patches. For each individual patch x_i, the maximum likelihood estimate of z is given by

$$\hat{z}_i = \operatorname*{argmax}_z g(x_i; zC) = \frac{1}{d} x_i^T C^{-1} x_i \tag{3.5}$$

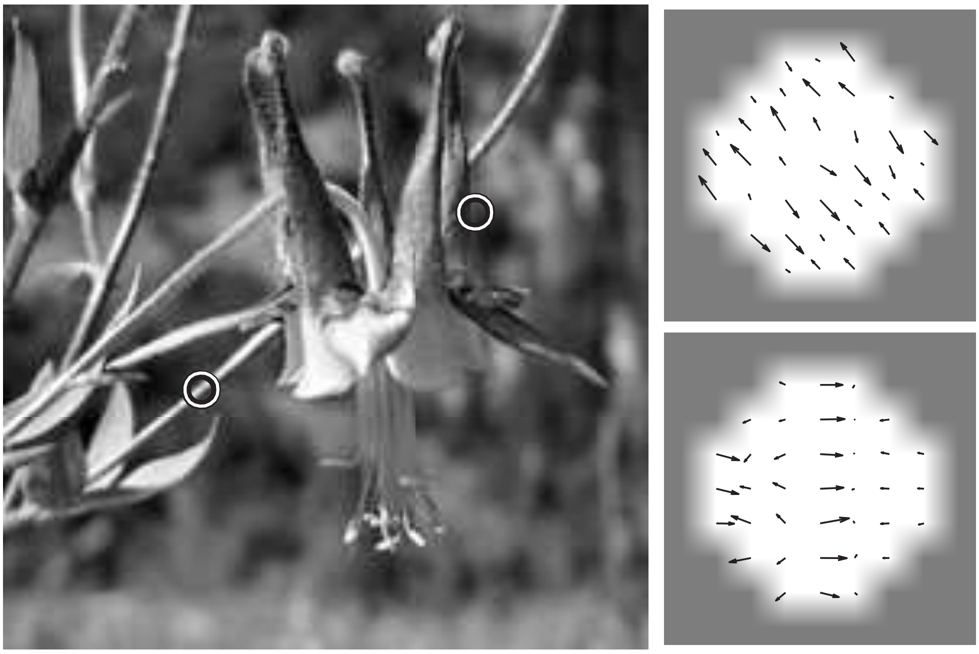


Figure 3.4: **Left**: Image with two strongly oriented patches indicated. **Right**: coefficients for each patch at one scale of a two-band steerable pyramid, displayed as vector fields. Patches are similar up to rotation.

where d is the dimension of the patches.

Dividing each coefficient by the value of $\sqrt{\hat{z}}$ computed using a neighborhood centered around the coefficient gives the transformed subband shown in figure 3.3 (c). As can be seen visually, the power of this normalized subband is much more spatially homogenous than for the original subband. The marginal statistics are also much closer to Gaussian, as may be seen by examining the log-histogram, which is very close to an inverted parabola. Normalizing by local contrast thus gives statistics that are closer to Gaussian.

Similar analysis may be performed for the rotator hidden variable θ. The OAGSM model is based on the idea that differences in structure between coefficient patches in different oriented regions may be explained by the action of θ. Visual evidence for this can be seen in figure 3.4, where two patches of two-band Steerable Pyramid coefficients for two different oriented regions are shown. The coefficient patches are displayed as patches of gradient vectors, as was discussed at length in Chapter 2. As can be seen from examining the two gradient patches, the essential structure of the two patches are similar up to rotation.

Attempting to describe the statistics of an ensemble of such patches without accounting for the rotational relationship between them will result in mixing structures at different orientations together. Such inappropriate data pooling will result in a less powerful signal description as some of the structure will have been averaged out. Conversely, taking advantage of this rotational relationship between coefficient patches leads to a more homogenous, easier to describe model. This statement can be made more precise by analyzing the second-order covariance statistics for ensembles of coefficient patches. One can "undo" the action of the rotator hidden variable by estimating the dominant local orientation of each patch, and then rotating each patch around its center by the estimated orientation. Performing this "orientation normalization" on every patch of coefficient from a particular spatial scale gives an ensemble of transformed patches with the same dominant orientation. The claim is that this set of rotated patches are more homogenous, and therefore easier to describe compactly, than the ensemble of raw original patches.

One simple measure of this is to perform Principle Components Analysis (PCA) on both sets of patches. Let v_i^{raw} and v_i^{rot} denote the raw and rotated patches extracted from one spatial scale of an image. The coefficients have been "vectorized" so that we may consider both v_i^{rot} and v_i^{rot} as vectors in R^d where

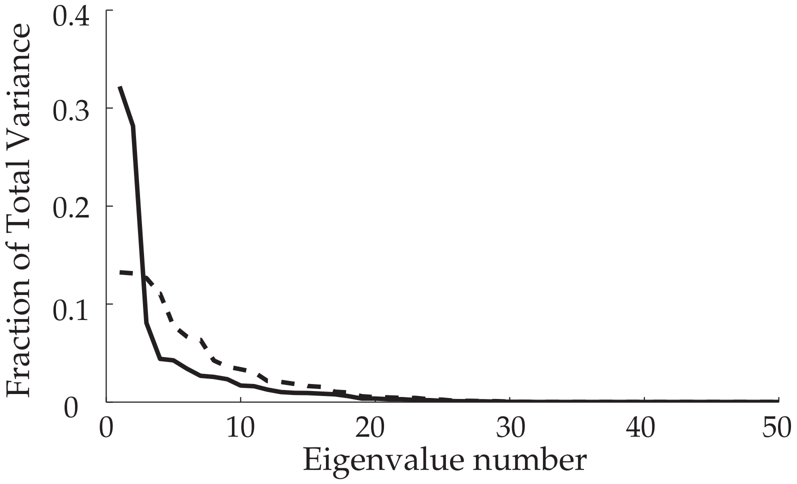


Figure 3.5: Normalized eigenvalues of covariance matrix estimated from coefficient patches drawn from single scale of the pyramid representation of an example image ("peppers"). Dashed curve corresponds to raw patches, and solid curve to patches rotated according to dominant orientation.

d is the dimension of the patch. The current calculation used 5x5 patches of two-band coefficients not including any parent or cousin coefficients, so that $d = 50$. PCA proceeds by first forming the sample covariance matrices

$$C^{\text{raw}} = \sum_i v_i^{\text{raw}} \left(v_i^{\text{raw}}\right)^T$$
$$C^{\text{rot}} = \sum_i v_i^{\text{rot}} \left(v_i^{\text{rot}}\right)^T$$

and then examining the eigenvalues and eigenvectors of these two covariance matrices. If the eigenvalues are normalized by the trace of the covariance, then they may be interpreted as giving the fraction of total signal variance that lies along the direction of the corresponding eigenvector. Normalized eigenvalues for C^{rot} and C^{raw} are plotted in decreasing order in Figure 3.5. For the rotated patches a greater portion of the total variance is accounted for by a smaller number of dimensions as compared to the raw patches. This "energy compaction" is a concrete measure of how the ensemble of rotated patches is more homogenous and therefore easier to describe.

3.2 Estimating Model Parameters

As mentioned earlier, the OAGSM model parameters can be specified by determining the oriented covariance matrices $C(\theta)$. If a continuous mixture of θ variables is used, then one must be able to specify the covariances for arbitrary θ. However, if θ is sampled discretely it is only necessary to determine $C(\theta)$ for a finite set of θ values. While the details of how the hidden variables are sampled will depend on the application of the model, this point that $C(\theta)$ may only be needed for a set of finite samples should be kept in mind. In this chapter I will detail three distinct methods of obtaining the oriented covariances. The first of these will be based on explicitly rotating patches of image data, similar to the numerical experiment shown in Figure 3.5. While this "explicit patch rotation" can give a very good signal description, the method is quite computationally intensive to compute, and does not give an closed form expression for $C(\theta)$. Realistically, it can only be computed for relatively coarsely sampled values for θ.

The second two methods for obtaining the oriented covariances are based on the idea that oriented image regions can be described as locally one-dimensional functions, that are constant in directions transverse to the gradient and have a particular profile moving parallel to the gradient. By assuming a particular form for the "edge profile" and using knowledge of the Steerable Pyramid filters, it is possible to compute a closed form expression for the oriented covariances. While this "explicit edge model" result is appealing, the resulting covariances are not adapted to the image data. The structure of this explicit edge model calculation shows that the elements of the oriented covariances matrices arise from a samples of a set of one dimensional correlation functions that depend on the power spectra of both the edge profile and the SP filters. For the original explicit edge model, these correlation functions can be computed analytically. However, this method can be made empirically adaptive by estimating these underlying one dimensional correlation functions directly from image data. This results in what I call the "1-d empirical" or "implicit rotation" method for obtaining oriented covariances.

Both the explicit edge model and the 1-d empirical methods of obtaining oriented covariances impose the constraint that the oriented signal is purely one dimensional locally. The patch rotation method oriented covariances, on the other hand, are formed from actual coefficient patches that are not purely one dimensional. As a result, the explicit edge model and 1-d empirical covariances perform very poorly when used on their own to model the signal content of the entire image, as they are only appropriate for very strongly oriented regions. However, they can be useful when used with the OAGSM model with non-oriented component, which is discussed later.

3.2.1 Calculation of Neighborhood Orientation

The explicit patch rotation method for estimating oriented covariances relies on estimating the dominant orientation for each neighborhood. This can be viewed as estimating the hidden variable θ at each spatial location. The two band Steerable Pyramid transform provides a representation of the image gradient which may be used to estimate the dominant orientation at each location in space and scale. In this work distinct neighborhood orientations are computed at each location in space and scale. Although there is a strong relationship between the neighborhood orientation at different scales and at the same spatial location, they are not constrained to be identical.

The dominant neighborhood orientation should be a robust measure of the average orientation of the gradient vectors within a patch. Some care must be taken in defining what exactly this measurement should be. One important point is that the dominant neighborhood orientation used in this work is only defined modulo π. Thus reversing the sign of the image intensity values will not change the neighborhood orientation. Taking this equivalence modulo π is important as the two band SP filters are bandpass filters and thus tend to have oscillating responses to a fixed signal feature. The typical behavior of the two band SP gradient vectors in oriented regions can be observed in the patches displayed in Figure 3.4. As can be seen, the vectors tend to be parallel, are all equal along the direction perpendicular to the gradient, and oscillate along the direction parallel to the gradient. As this oscillation results in the gradient vectors constantly changing their direction by π, it is unreasonable to expect to define and robustly measure neighborhood orientation that does not have the property of being defined only modulo π.

The following measure of neighborhood orientation is used. An $m \times m$ neighborhood (without including parent) of two band SP coefficients may be considered as a collection of m^2 gradient vectors v_i for $i = 1...m^2$. Let $k(\phi) =$

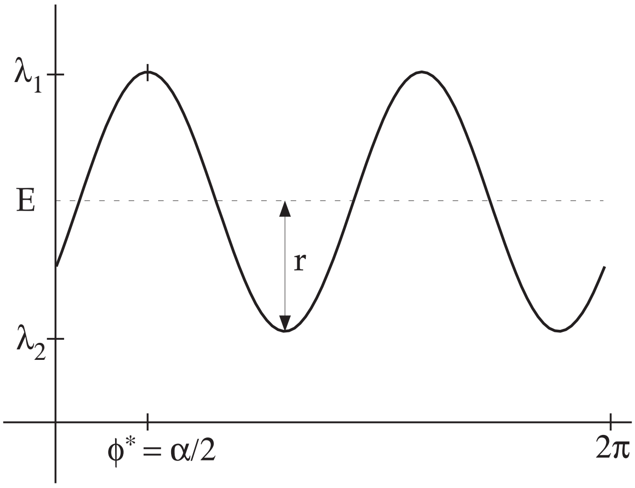


Figure 3.6: orientation response curve $S(\phi)$

$(\cos(\phi), \sin(\phi))^T$ be a unit vector, and let the "response" of the patch to this unit vector

$$S(\phi) = \sum_{i=1}^{m^2} \left(k(\theta)^T v_i\right)^2. \tag{3.6}$$

be the sum of squares of inner products of $k(\phi)$ and the patch vectors. I now define the neighborhood orientation ϕ^* to be

$$\phi^* = \underset{\phi}{\operatorname{argmax}}\, S(\phi), \tag{3.7}$$

the angle of the unit vector producing maximum response to the patch. Note that as $S(\phi) = S(\phi + \pi)$, the neighborhood orientation is clearly only defined modulo π.

It is instructive to rewrite the patch response as a quadratic form. As

$k(\phi)^T v_i = v_i^T k(\phi)$, one may write

$$\begin{aligned} S(\phi) &= \sum_{i=1}^{m^2} k(\phi)^T v_i v_i^T k(\phi) \\ &= k(\phi)^T \left(\sum_{i=1}^{m^2} v_i v_i^T \right) k(\phi) \\ &= k(\phi)^T M k(\phi) = \frac{k(\phi)^T M k(\phi)}{k(\phi)^T k(\phi)} \end{aligned} \tag{3.8}$$

where $M = \sum_{i=1}^{m^2} v_i v_i^T$ defines the 2 × 2 "orientation response matrix" for the patch. M is a symmetric positive-semidefinite matrix. The last equality above follows as $k(\phi)$ is a unit vector. The expression 3.8 for $S(\phi)$ is a Rayleigh quotient, from which it follows that $k(\phi^*)$ will be the eigenvector of M corresponding to the largest eigenvalue.

Some manipulation with trigonometric identities will allow calculation of ϕ^*. Introduce the notation

$$M = \begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix} \tag{3.9}$$

We may then expand

$$\begin{aligned} S(\phi) &= M_{xx} \cos^2(\phi) + 2M_{xy} \cos(\phi) \sin(\phi) + M_{yy} \sin^2(\phi) \\ &= M_{xx} \frac{1 + \cos(2\phi)}{2} + M_{xy} \sin(2\phi) + M_{yy} \frac{1 - \cos(2\phi)}{2} \\ &= \frac{M_{xx} + M_{yy}}{2} + \frac{M_{xx} - M_{yy}}{2} \cos(2\phi) + M_{xy} \sin(2\phi) \end{aligned} \tag{3.10}$$

To proceed, transform $(\frac{M_{xx}-M_{yy}}{2}, M_{xy})$ into polar coordinates (r, α) with

$$r = \sqrt{(\frac{M_{xx} - M_{yy}}{2})^2 + (M_{xy})^2} \tag{3.11}$$

$$\alpha = \angle(\frac{M_{xx} - M_{yy}}{2}, M_{xy}) \tag{3.12}$$

where $\angle(\cdot, \cdot)$ indicates the angle of the vector whose components are specified by the two arguments. In addition, set $E = \frac{M_{xx}+M_{yy}}{2}$.

Substituting these expressions for E, r and α into 3.10 gives

$$\begin{aligned} S(\phi) &= E + r(\cos(\alpha)\cos(2\phi) + \sin(\alpha)\sin(2\phi)) \\ &= E + r\cos(2\phi - \alpha) \end{aligned} \tag{3.13}$$

A representative plot of the orientation response is shown in figure 3.6. From this expression, it is clear that the maximum of $S(\phi)$ is obtained at

$$\phi* = \frac{\alpha}{2} = \frac{1}{2}\angle\left(\frac{M_{xx} - M_{yy}}{2}, M_{xy}\right) \tag{3.14}$$

The three constants E, r and α are a reparameterization of the three degrees of freedom present in the orientation response matrix M. The eigenvalues of M are given by $\lambda_1 = E + r$ and $\lambda_2 = E - r$. In addition to the neighborhood orientation, these constants capture some additional local properties that may be interpreted as measuring the neighborhood power and the degree to which the patch is purely oriented. For perfectly oriented patches, all of the gradient vectors would be parallel. Picking a unit vector orthogonal to the gradient vectors will give exactly zero response for $S(\phi)$. In this case the minimum eigenvalue $\lambda_2 = 0$. Conversely, for completely non-oriented patches the orientation response $S(\phi)$ will be constant. In this case $\lambda_1 = \lambda_2$. The eigenvalues of the orientation response matrix M can thus provide an ad-hoc measurement of the "orientedness" of the patch. I define here the orientedness measure

$$d_{ori} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{r}{E} \tag{3.15}$$

which takes values between 0 (perfectly non-oriented) and 1 (perfectly oriented). Lastly, note that the sum of the eigenvalues, equal to the trace of M, is also equal to the sum of squares of the magnitudes of the patch gradient vectors v_i. In this way, the three degrees of freedom in the matrix M measure the power, orientation, and orientedness of the coefficient patch.

The geometry of the patches of two band SP coefficients used to compute the dominant local orientation at each space and scale location does not need to be exactly the same as the geometry of the generalized coefficient neighborhoods that are modeled by the OAGSM model. As will be seen, the OAGSM model can be constructed for patches of SP coefficients using any number of orientation bands. However, the dominant neighborhood orientations are always computed using the two band SP transform. There is a tradeoff in the size of the patches used to compute dominant orientation. Using larger patches will provide a more robust measurement, especially in the presence of noise. Making the patch size too large, however, will result in more of the patches mixing signal content at different orientations. Throughout this work, neighborhood orientations were measured using 5 × 5 patches without including parent coefficients.

3.2.2 Rotation of Coefficient Patches

The OAGSM model is based on the idea of rotating a single uniform Gaussian Scale Mixture process to produce signal content with spatially varying dominant orientations. The model may be specified without the need for literally implementing this forward rotation process, if the oriented covariance matrices $C(\theta)$ have been determined. However, as the patch rotation method for estimating $C(\theta)$ will rely on rotating coefficient patches, the details of the patch rotation operator $R(\theta)$ must be specified.

Rotating coefficient patches will involve both resampling coefficients off of the original sample grid, and then steering the coefficients as vector components. The ability to do both of these relies on the translation-invariance and steerability properties of the Steerable Pyramid representation. The patch rotation procedure described in this work thus cannot be easily applied to traditional orthogonal wavelet patches which suffer from spatial aliasing which prevents interpolation off of the original sample lattice.

Rotating a patch of wavelet coefficients is equivalent to finding the coefficients that would arise if the underlying image signal was rotated around the center of the patch. Ignoring the pixel sampling, the original image may be viewed as a continuous function from $\mathbb{R}^2$ to $\mathbb{R}$. As this is a scalar function, it is simple to define rotation around a single point in the image domain. It should be noted that this defines patch rotation for Steerable Pyramid coefficients of any order. While it is simpler to visualize the effect of patch rotation for the two-band case when the coefficients can be interpreted as gradient vectors, as in figures 3.4 and 3.7 , rotation of patches of SP coefficients of any order is well defined.

To explain these ideas more precisely, introduce the following notation. Denote the space of functions from $\mathbb{R}^2$ to $\mathbb{R}$ by Im, the model of the image domain. Define the "patch measurement operator" $T : Im \to \mathbb{R}^d$, where d is the dimension of the coefficient patch, which calculates the a wavelet coefficient patch centered at the origin in the image domain. Let $i = 1...d$ index the different coefficients of the patch. Each coefficient corresponds to a SP filter at a particular location, orientation and scale. If the generalized patch includes parents or cousins, the orientation and scale of the filters in the same patch may be different. It is also possible that distinct patch coefficients may be at the same location in space but correspond to different filter orientations. To keep track of these, let $\vec{p_i} = (x_i, y_i)$ be the position of the filter for the i^{th} coefficient, offset from the center of the patch. Similarly, let ϕ_i be the orientation and s_i the scale of the filter for the i^{th} coefficient. Assume that the number of orientation bands K for the SP has been fixed. Let $B^s_\phi(x, y)$ be the SP filter for this transform centered at the origin with orientation ϕ at scale s.

Now, for any $h \in Im$, the i^{th} coefficient of the measured patch is

$$(Th)_i = \iint h(x, y) B^{s_i}_{\phi_i}(x - x_i, y - y_i) dxdy \tag{3.16}$$

Let $r_\theta : Im \to Im$ be the operator that rotates the image domain about the origin by θ. This is given by

$$(r_\theta h)\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = h\left(M_{-\theta}\begin{pmatrix} x \\ y \end{pmatrix}\right) \tag{3.17}$$

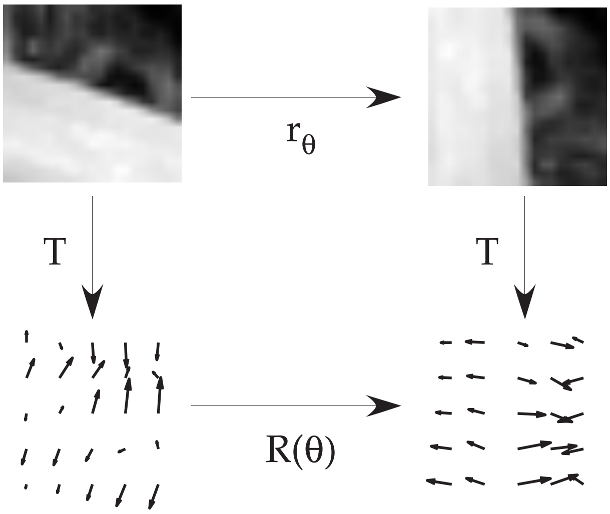


Figure 3.7: Definition of patch rotation. The above diagram commutes.

where

$$M_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{3.18}$$

The patch rotation operator $R(\theta)$ is then formally defined by requiring

$$R(\theta)Th = T(r(\theta)h) \tag{3.19}$$

This is illustrated in figure 3.7.

The i$^{\text{th}}$ coefficient of the rotated patch is thus given by

$$(R(\theta)Th)_i = \iint h\left(M_{-\theta}\begin{pmatrix} x \\ y \end{pmatrix}\right) B^{s_i}_{\phi_i}\left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix}\right) \tag{3.20}$$

This integral is invariant under rotating the (x, y) coordinate system by θ. This yields

$$
\begin{aligned}
(R(\theta)Th)_i &= \iint h\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) B_{\phi_i}^{s_i}\left(M_\theta \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix}\right) && (3.21) \\
&= \iint h\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) B_{\phi_i}^{s_i}\left(M_\theta \left(\begin{pmatrix} x \\ y \end{pmatrix} - M_{-\theta}\begin{pmatrix} x_i \\ y_i \end{pmatrix}\right)\right) && (3.22) \\
&= \iint h\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) B_{\phi_i-\theta}^{s_i}\left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x'_i \\ y'_i \end{pmatrix}\right) && (3.23)
\end{aligned}
$$

where $\binom{x'_i}{y'_i} = M_{-\theta}\binom{x}{y}$.

Thus the i^{th} coefficient of the transformed patch is given by the response to the orignal, unrotated image of the filter with orientation $\phi_i - \theta$ at location (x'_i, y'_i). Using the steerability properties of the SP (see section 2.1.4), this can be computed as a linear combination of the responses of the K filters with the standard orientations $\frac{(k-1)\pi}{K}$ for $k = 1...K$, at the location (x'_i, y'_i). So the problem is reduced to finding the responses of the K standard filters at the location (x'_i, y'_i). In general this point will not lie on the original sample lattice. However, as the SP filters are bandlimited, it is possible to interpolate their values off of the original sample lattice. This is performed in practice by first upsampling each of the K subbands by a factor of 2^{Uf} in each direction by padding the Fourier transform of the subband with zeros and taking the inverse Fourier transform. I then perform bilinear interpolation from the four nearest upsampled lattice points to find the filter response at the location off of the sample grid. It was found by experiment that using $Uf = 4$ gave reasonable quality results.

It should be noted that this method of rotating image patches actually uses information outside of the dimensions of the original image patch. This occurs for two reasons. First of all, it is possible for the transformed sample point (x_i, y_i) to lie outside of the original patch area due to the fact that the patches cannot be perfectly circular, as they are formed by selecting square lattice locations. The "corners" of a rotated square patch may land outside of the original patch, and thus rotating the square patch requires access to information outside of the original patch. Secondly, as the method of interpolating off of the sample lattice is done by padding with zeros in the Fourier domain, it implicitly uses all of the coefficients in the entire subband. As a result of this, coefficient patches can only be rotated by the given method if one has access to some larger surrounding area of coefficients. While this may seem to be a burdensome restriction, in practice one has access to the entire set of coefficients for the image and thus may rotate any patch by using this method.

3.2.3 Estimating Oriented Covariances by Patch Rotation

The patch rotation process described above may be used for estimating the oriented covariance matrices for the OAGSM model. As one of the primary applications the model was developed for is image denoising, I will discuss estimating the oriented covariances from noisy image data. If the original image is corrupted by additive Gaussian white noise, each subband will be corrupted by filtered noise. The noise component in each wavelet patch will be a multivariate Gaussian whose covariance may be calculated using knowledge of the SP filters. Details of this will be deferred until Chapter 4.

The OAGSM describes each noisy patch w as

$$w = \sqrt{z}R(\phi)u + n \tag{3.24}$$

where n is a sample from the zero mean Gaussian filtered noise process with known covariance C_n.

The primary impediment to computing the oriented covariances

$$C(\theta) = E\left[R(\theta)uu^T R(\theta)^T\right]$$

is that the hidden rotator variables are different for every patch. If one had access to an ensemble of noisy patches w that were formed from a single fixed value θ^* of the hidden variable, then assuming the noise and signal are independent, taking the sample outer product of w gives

$$\begin{aligned} E\left[ww^T\right] &= E\left[zR(\theta^*)uu^T R(\theta^*)\right] + E\left[nn^T\right] \\ &= E[z]R(\theta^*)E[uu^T]R(\theta^*) + C_n = E[z]C(\theta^*) + C_n \end{aligned} \tag{3.25}$$

from which computing $C(\theta)$ is straightforward assuming C_n and the distribution for z are known.

While such an ensemble of patches drawn from a fixed value θ of the rotator hidden variable is not immediately accessible, the patch rotation method proceeds by producing such an ensemble by manipulating the patches present in the given noisy image. The true values of the rotator hidden variables ϕ are unknown, but are estimated by computing the dominant neighborhood orientation ϕ^* of the noisy patches, as described in 3.2.1. Given a set of noisy patches w_i with measured neighborhood orientations ϕ_i^*, rotating each patch by $\theta - \phi^*$ produces an ensemble of patches that is equivalent to one produced by the OAGSM process with a fixed value θ for the rotator variable. This then yields, following 3.25

$$E\left[(R(\theta - \phi_i^*)w_i)\,(R(\theta - \phi_i^*)w_i)^T\right] = E[z]C(\theta) + C_n \tag{3.26}$$

Note that in the above expression, θ is a fixed constant while ϕ_i^* is different for every patch. The oriented covariances are thus computed by forming the average outer product of patches rotated to θ beyond their dominant orientation. The noise covariance is subtracted off and the result normalized by $E[z]$. This computation must be repeated for every different value of θ for which $C(\theta)$ is required. As mentioned earlier, this implies that the patch rotation method is only practical if the OAGSM model will be used in a fashion such that θ may be sampled at a relatively small number of values. As the oriented covariance estimates are computed from data, it is possible that they lose positive definiteness after subtracting C_n. Positive definiteness is imposed by diagonalizing the estimated covariances and replacing all negative eigenvalues by a small positive constant.

As mentioned in section 3.2.2, patch rotation may be defined for SP coefficients of any order. This implies that the patch rotation method may be applied to give oriented covariances for SP coefficients of any order. It should be kept in mind that the dominant patch orientations ϕ_i^* are computed using the two-band SP transform, even if the oriented covariances are computed for a SP transform of higher order.

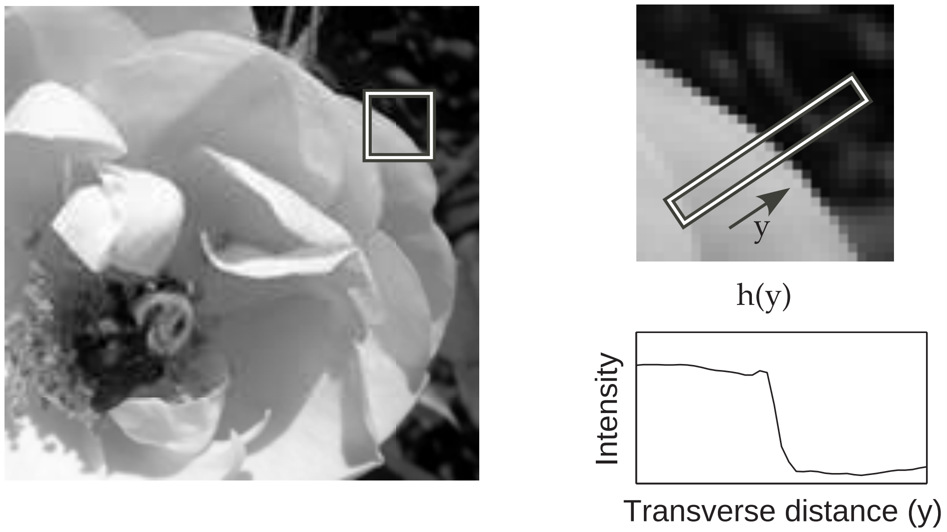


Figure 3.8: In oriented regions, image signal is approximately locally one dimensional, and may be characterized by the transverse profile $h(y)$

3.2.4 Local one-dimensional signal based methods

An alternative framework for obtaining the oriented covariances arises from describing oriented image signal as locally one dimensional. In strongly oriented regions such as occlusion boundaries, the image gradient is often large and locally consistent. Patches of 2-band steerable pyramid coefficients in such regions consist of approximately parallel vector fields with a well defined local orientation. In such image regions the image intensities are approximately constant moving transverse to the local orientation. The image intensities defined by moving parallel to the local gradient orientation give the edge profile $h(y)$ (see fig 3.8).

Given a fixed edge profile $h(y)$, an image with an edge with gradient orientation θ passing through the origin may be written as

$$e_\theta(x, y) = h(x\cos(\theta) + y\sin(\theta)) \tag{3.27}$$

Note that the gradient orientation is perpendicular to the edge. These functions formed from the edge profile form a model for oriented image content. To place this in a consistent framework for the OAGSM, one must describe a model for the same type of generalized wavelet coefficient patches as used above. I describe the ensemble of coefficient patches with orientation θ as the response to the oriented edge e_θ where the center of the patch (x_0, y_0) is random. Taking the sample outer product of this set of oriented patches then defines the oriented covariances $C(\theta)$, which may then be used in the OAGSM model.

Once the edge profile $h(y)$ is fixed, the oriented patches under this model are a deterministic function of x_0, y_0. As the edge model signal e_θ is invariant under translation perpendicular to $(\cos(\theta), \sin(\theta))$, the oriented patches only depend on θ and the perpendicular distance $t = x_0\cos(\theta) + y_0\sin(\theta)$ from the center of the patch to the edge. Let $P(t, \theta) \in \mathbb{R}^d$ denote the vectorized coefficients of a patch perpendicular distance t away from an edge function with orientation θ.

The "edge profile covariances" $C(\theta)$ are given by the average sample outer product of this ensemble of oriented patches. This average is taken over patches generated from all possible values of the perpendicular distance t. Defining this average properly would require taking the average with respect to a well defined probability density on t. However, for edge profiles with compactly supported signal content, the patches $P(t, \theta)$ will decay to zero as $t \to \infty$. Provided each coefficient in the patches decays faster than $t^{-1/2}$, then each term of the outer

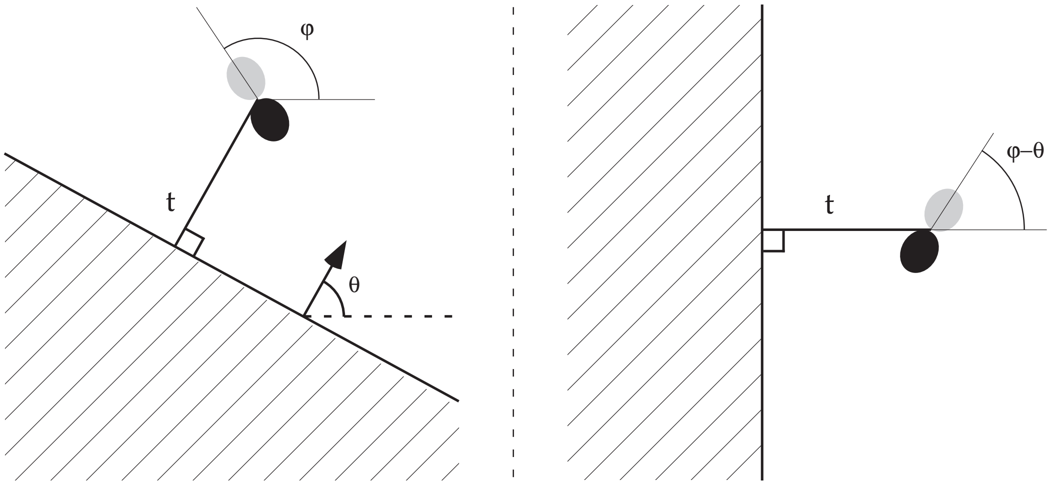


Figure 3.9: Definition of edge response function. Inner products are the same for these two cases, and equal to $f_{er}(t, \phi - \theta)$

product will be an integrable function of t. The "edge profile covariances" $C(\theta)$ may then be defined as

$$C(\theta) = \int_{-\infty}^{\infty} P(t,\theta)P(t,\theta)^T dt \tag{3.28}$$

This expression takes the average over t using a constant density for t, but without normalizing the average. Taking a properly normalized limit of average outer products over a constant density for $t \in [-R, R]$ and sending $R \to \infty$ will result in covariances equal to zero, as the above integral is finite and the normalizing constant $2R$ goes to infinity. However, disregarding the normalization constant will give a finite nonzero expression for the covariances. Accordingly, the covariances given by (3.28) are really only meaningful modulo a global multiplicative constant. However, this is acceptable as they will be used in the OAGSM model where they will be multiplied by the scalar hidden variable z.

Each element of the right hand side of (3.28) may be computed from the Fourier transforms of the edge profile h and of the Steerable Pyramid filters. Each matrix element of $C(\theta)$ will involve an integral of the product of filter responses of two filters spatially shifted by a displacement determined by the patch geometry. The response of a filter with orientation ϕ scale s at location $(x0, y0)$ to the edge model signal through the origin with orientation θ is given by

$$\iint B^s_\phi(x - x_0, y - y_0)e_\theta(x, y)dxdy \tag{3.29}$$

While for a fixed spatial scale s this expression appears to depend on the four variables ϕ, θ, x_0 and y_0, there are in fact only two degrees of freedom. Rotating the coordinate system in the integral (3.29) by $-\theta$ and translating perpendicular to the edge gradient shows that

$$\iint B^s_\phi(x - x_0, y - y_0)e_\theta(x, y)dxdy = \iint B^s_{\phi-\theta}(x - t, y)e_0(x, y)dxdy \tag{3.30}$$

where $t = x_0 \cos(\theta) + y_0 \sin(\theta)$ is the perpendicular distance from the SP filter to the edge. This is diagrammed in figure 3.9. Let

$$f^s_{er}(\varphi, t) = \iint B^s_\varphi(x - t, y)e_0(x, y)dxdy = \iint B^s_\varphi(x - t, y)h(x)dxdy \tag{3.31}$$

be the "edge response function" for the edge profile h.

This function is in fact separable in the two variables φ and t. This follows from the property that the SP filters are polar separable in the Fourier domain. f^s_{er} is an inner product of a SP filter and the oriented edge signal. This inner product may be written as the two dimensional convolution

$$f_{er}^{s}(\varphi, t) = \left(B_{\varphi}^{s} \star \tilde{e}_0\right)[t, 0] \tag{3.32}$$

where $\tilde{e}_0(x, y) = e_0(-x, -y)$. By the convolution theorem, we have

$$B_{\varphi}^{s} \star \tilde{e}_0 = \mathcal{F}^{-1}\left[\mathcal{F}\left[B_{\varphi}^{s}\right]\mathcal{F}\left[\tilde{e}_0\right]\right] \tag{3.33}$$

The Fourier transform of the SP basis functions are separable in the polar domain. Recall from section 2.1.1

$$\hat{B}_{\varphi}^{s}(w_x, w_y) = i^{K-1}\cos^{K-1}(\theta(w_x, w_y) - \varphi)g_s(r(w_x, w_y)) \tag{3.34}$$

where $\theta(w_x, w_y) = \angle(w_x, w_y)$ and $r(w_x, w_y) = \sqrt{w_x^2 + w_y^2}$ are the polar coordinates in the Fourier domain.

The space-flipped oriented edge signal $\tilde{e}_0(x, y) = h(-x)$ may be viewed as a separable product of $h(-x)$ a constant unit function in y. The Fourier transform of this constant function is given by the delta function distribution. Reversing the sign on the x axis results in taking the complex conjugate in the Fourier domain. Accordingly, we have $\hat{e}_0 = \hat{h}^*(w_x)\delta(w_y)$ so that

$$\hat{B}_{\varphi}^{s}\hat{e}_0 = \delta(w_y)\hat{h}^*(w_x)i^{K-1}\cos^{K-1}(\theta(w_x, w_y) - \varphi)g_s(r(w_x, w_y)) \tag{3.35}$$

Because of the presence of $\delta(w_y)$ in this expression, one may replace all other occurrences of w_y by 0 without changing its value. In particular, $r(w_x, 0) = |w_x|$

and

$$\theta(w_x, 0) = \begin{cases} 0 & \text{if } w_x > 0 \\ \pi & \text{if } w_x < 0 \end{cases}$$

which implies $\cos(\theta(w_x, 0) - \varphi) = \cos(\varphi)\frac{w_x}{|w_x|}$.

Substituting these into 3.35 shows that

$$f^s_{er}(\varphi, t) = (\cos(\varphi))^{K-1}\mathcal{F}^{-1}\left[\delta(w_y)\hat{h}(w_x)i^{K-1}g_s(|w_x|)\left(\frac{w_x}{|w_x|}\right)^{K-1}\right][t, 0] \quad (3.36)$$

The terms inside the inverse Fourier transform no longer have any φ dependence. It follows that $f^s_{er}(\varphi, t)$ is separable and we may write

$$f^s_{er}(\varphi, t) = \cos^{K-1}(\varphi)F^s(t) \quad (3.37)$$

where

$$\hat{F}^s(w) = i^{K-1}\hat{h}^*(w)g_s(|w|)\left(\frac{w}{|w|}\right)^{K-1} \quad (3.38)$$

Now reintroduce the notation from section 3.2.2. Let i and j index the coefficients in the generalized patch, and let ϕ_i, s_i and $\vec{p}_i = (x_i, y_i)$ be the orientation, scale and offset from the center for the i$^{\text{th}}$ SP filter in the patch. Set $n(\theta) = (\cos\theta, \sin\theta)$.

Using this notation, the i$^{\text{th}}$ coefficient of a patch that has its center perpendicular distance t from an oriented edge e_θ will be $f^{s_i}_{er}(\phi_i - \theta, t + \vec{p} \cdot n(\theta))$. Accordingly, the $(i, j)^{\text{th}}$ element of the edge model oriented covariance will be given by

$$C(\theta)_{i,j} = \int f^{s_i}_{er}\left(\phi_i - \theta, t + \vec{p}_i \cdot n(\theta)\right) f^{s_j}_{er}\left(\phi_j - \theta, t + \vec{p}_j \cdot n(\theta)\right) dt \quad (3.39)$$

Define $d_{ij}(\theta) = (\vec{p}_i - \vec{p}_j) \cdot n(\theta)$. This gives the component of the displacement between two coefficient locations parallel to the gradient of the edge model signal. Substituting this and 3.37 into 3.39 gives

$$C(\theta)_{i,j} = \cos^{K-1}(\phi_i - \theta) \cos^{K-1}(\phi_j - \theta) \int F^{s_i}(t) F^{s_j}(t - d_{ij}(\theta)) dt \qquad (3.40)$$

This expression splits into the "angle term" $\cos^{K-1}(\phi_i - \theta) \cos^{K-1}(\phi_j - \theta)$ and the correlation of the two functions F^{s_i} and F^{s_j} evaluated at the point $d_{ij}(\theta)$. The number of distinct correlation functions that must be computed will depend on the number of different spatial scales that are present in the generalized patch. If the patch contains only parent and child coefficients (and no "grandparents"), then there will be only three such correlation functions corresponding to the child-child, parent-child and parent-parent interactions. Each entry of the covariance matrix $C(\theta)$ will be formed by the product of an angle term, and the appropriate correlation function sampled at the value $d_{i,j}(\theta)$.

In practice, these correlation functions may be computed in advance, and used as a lookup table to compute the oriented covariances for any value of θ. Set $c^{s_i,s_j}(d) = \int F^{s_i}(t) F^{s_j}(t - d) dt$. Again using the convolution theorem, we have

$$\begin{aligned} \hat{c}^{s_i,s_j}(w) &= F^{s_i}\,(\hat{F^{s_j}})^* \\ &= |h(w)|^2 g_{s_i}(|w|) g_{s_j}(|w|) \qquad (3.41) \end{aligned}$$

The final remaining item to be specified in this calculation is the Fourier transform of the edge profile, $\hat{h}(w)$. In this thesis two simple "edge models" were used, corresponding to a step edge and a line edge. The step edge model sets

$$h_{step}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{3.42}$$

with Fourier transform $\hat{h}_{step}(w) = \frac{i}{w}$. The infinitesimally thin line edge model has $h_{line}(x) = \delta(x)$ with Fourier transform $\hat{h}_{line}(w) = 1$. Numerical computation of the correlation functions c described by equation 3.41 are performed using the 1-d Fast Fourier transform. Once these have been computed, each entry of the oriented covariance matrices are given by

$$C(\theta)_{i,j} = \cos^{K-1}(\phi_i - \theta)\cos^{K-1}(\phi_j - \theta)c^{s_i,s_j}(d_{i,j}(\theta)) \tag{3.43}$$

3.2.5 Empirical 1-d Oriented Covariances

The edge model oriented covariance calculation described above constructs each element of $C(\theta)$ using a set of correlation functions c^{s_i,s_j} that are calculated from a fixed edge profile. However, oriented edges in real images are rarely pure step edges or lines. As the correlation functions c^{s_i,s_j} depend on the underlying edge edge content, it is natural to ask if they can be computed empirically from image data. Such a model is appealing because the structural assumption that oriented image content is locally one dimensional is imposed, but the remaining parameters of the model are fit from actual data.

Equation 3.43 gives each entry of the oriented covariance matrices as a product of an "angle term" and a sample from a particular correlation function. For a Gaussian process, the covariance matrix is given by the average outer product of vectorized patches. Assume one has a set of patches $v_k \in \mathbb{R}^d$, for $k = 1...N$ that are taken from regions with measured neighborhood orientation θ_k. Compute the corresponding angle terms $A_{i,j}(\theta_k) = \cos^{K-1}(\phi_i - \theta_k)\cos^{K-1}(\phi_j - \theta_k)$ and perpendicular displacements $d_{i,j}(\theta_k)$ for every given patch.

The empirical 1-d method proceeds by undoing the effect of the angle term in order to build up a set of correlation function values that are then averaged to give the estimated correlation functions. For each patch k and for each pair of patch coefficients (i, j), compute

$$\begin{aligned} f_{i,j,k} &= \frac{(v_k)_i (v_k)_j}{A_{i,j}(\theta_k)} \\ x_{i,j,k} &= d_{i,j}(\theta_k) \end{aligned} \tag{3.44}$$

As the above expression involves dividing by the angle term $A_{i,j}(\theta_k)$, some care must be taken when this term is close to zero. In practice, terms for when the angle term is below some threshold A^* are simply discarded. A typical value for this threshold is $A^* = .05$. The $f_{i,j,k}$ are then sorted into histogram bins according to the values of the independent variable samples $x_{i,j,k}$, and then averaged over each bin. This procedure must be done separately for each of the correlation functions (typically child-child, child-parent and parent-parent) that need to be computed. Let (S_1, S_2) represent the spatial scales indicating the desired correlation function.

Given a set of histogram bin edges $x_1...x_N$, the estimated correlation function values are

$$\tilde{c}_n^{S_1,S_2} = \frac{1}{M_n} \sum_{x_n \le x_{i,j,k} \le x_{n+1}} f_{i,j,k} \tag{3.45}$$

where $M_n = \#\{x_n \leq x_{i,j,k} \leq x_{n+1}\}$. Only (i,j) indices corresponding to the desired correlation function are used in the sum, i.e. $(i,j) \in \{(i,j)|s_i = S_1, s_j = S_2\}$. It is not necessary that the same bin edges be used for the three different correlation functions. Once these estimated correlation values are obtained, they may be used as samples for the desired correlation c^{S_1,S_2} at the bin centers $d_n = (x_n + x_{n+1})/2$. Intermediate values will be required when forming the oriented covariances $C(\theta)$ from these estimated correlation functions; they may be produced by either linear or cubic interpolation.

3.3 OAGSM with non-oriented component

The OAGSM model provides a good description of patches of wavelet coefficients that arise from oriented image signal regions. Such oriented regions are widespread in natural images, and are often very significant for perception. However, not all image regions are well described by this oriented signal model. Non-oriented regions with significant local power may include texture regions, as well as areas where oriented structure at more than one orientation overlaps, such as at T-junctions.

This shortcoming may be addressed by introducing a non-oriented signal component into the model, yielding the OAGSM with non-oriented component (OAGSM/NC). This non-oriented component consists of a Gaussian Scale Mixture process described by a single covariance C_{nor} that does not depend on the θ hidden variable. I introduce an additional binary hidden variable δ that controls selection of either the oriented or non-oriented component. Given the hidden variables (z, θ, δ), the patch v is drawn from an OAGSM process if $\delta = 1$, or from a GSM process with covariance C_{nor} if $\delta = 0$. The forward process generating the patch v is

$$v = \begin{cases} zR(\theta)u & \text{if } \delta = 1 \\ \\ zw & \text{if } \delta = 0 \end{cases} \tag{3.46}$$

where u is a sample from a Gaussian with covariance C_0 and w is a sample from a Gaussian with covariance C_{nor}.

It follows that the probability distribution for v when conditioned on the hidden variables is

$$p(v|z, \theta, \delta) = g\left(v; \delta z C(\theta) + (1 - \delta) z C_{nor}\right) \tag{3.47}$$

The above expression gives the signal covariance as a function of the hidden variables x, θ and δ. While z and θ are permitted to take on continuous values, the model described here has the "orientedness" variable δ constrained to be binary. This constraint may seem overly restrictive, as patches may show more graded variations in how strongly oriented they are. Additionally, the right hand side of expression (3.47) does give well defined covariance matrices for values of δ between 0 and 1. The restriction to binary values for δ is nonetheless made here as it makes the model easier to interpret as switching between two distinct stochastic processes. The OAGSM is able to describe inhomogeneous signals by transforming a single process by rotation and scalar multiplication. It would be appealing to be able to account for inhomogeneities in orientedness through a similar type of transformation. However, it is unclear how to define a continuous, single parameter family of operations that change the orientedness of image coefficient patches. For this reason, orientedness is modeled by switching between distinct stochastic processes with a binary hidden variable. Studying the construction of models employing continuous hidden variables to model the orientedness would be an interesting possible future extension of this work.

The OAGSM/NC model still has the fundamental structure of a Gaussian mixture, where each of the hidden variables determining the mixture covariances has a direct interpretation in terms of image structure. Note that while the forward generating process makes a binary decision as to sample from the oriented or non-oriented components of the model, these are mixed together for describing a signal patch provided that the prior density on δ does not put all of the weight on either 0 or 1.

A separable hidden variable prior probability $p(z, \theta, \delta) = p(z)p(\theta)p(\delta)$ is assumed. The priors for z and θ will be taken to be the same as in the OAGSM model, and will be specified in detail in chapter 4. As δ is a binary variable, the prior $p(\delta)$ is just a discrete density on two points and is completely specified by its weight for either component. To fix notation, let $\beta = p(\delta = 1)$, the probability that each patch is drawn from the oriented process. The OAGSM/NC distribution is then given by

$$\begin{aligned} p(v) &= \int g\left(v; \delta z C(\theta) + (1-\delta) z C_{nor}\right) p(z)p(\theta)p(\delta) \\ &= \beta \int g\left(v; zC(\theta)\right) p(z)p(\theta) dz d\theta + (1-\beta) \int g\left(v; zC_{nor}\right) p(z) dz \qquad (3.48) \end{aligned}$$

This expression makes it clear that as β varies between 0 and 1, the OAGSM/NC model interpolates between the GSM and the OAGSM models.

3.3.1 Estimating OAGSM/NC model covariances

The model parameters for the OAGSM/NC are the same as for the OAGSM, with the addition of the non-oriented covariance C_{nor} and β. Either of the three OAGSM covariance calculations described in section 3.2 (patch rotation, edge model, or the related empirical 1-d method) may be used to obtain the oriented covariances $C(\theta)$. The non-oriented component covariance C_{nor} may be calculated in the same way as in Portilla's original GSM work, by simply taking the average outer product of the raw, non-rotated coefficient patches. If these must be fit from noisy data, the noise covariance is then subtracted and any negative eigenvalues corrected as described previously.

Using this way of computing C_{nor} in tandem with the patch rotation method for computing $C(\theta)$ may seem strange, as the same coefficient patches are used to compute both the oriented and non-oriented covariances. This is somewhat inconsistent with the forward OAGSM/NC generative model which describes each patch as a sample from either the oriented or the non-oriented process. Intuitively, one should compute the oriented covariances using oriented patches, and the non-oriented covariances using non-oriented patches.

One way of performing this separation is by "winnowing" the patches by a measure of their orientedness before performing the above calculations. As described in section 3.2.1, one can use the eigenvalues of the Orientation Response Matrix measured at each location in space and scale to define the orientedness measure d_{ori}. The same calculation provides a measure of the local power E of the patch. The winnowing method selects patches to use in the calculation of $C(\theta)$ or C_{nor} by simple thresholding according to these orientedness and local power measurements. $C(\theta)$ will be computed using patches with orientedness and local power above certain threshold values, namely $d_{ori} > d_{ori}^{high}$ and $E > E^*$. The patches used for computing C_{nor} will satisfy $d_{ori} < d_{ori}^{low}$, but still have the same minimum power requirement $E > E^*$. This local power condition is included to avoid using very weak signal patches in the covariance estimates. This is especially important when performing these calculations using noisy data, as will be required in Chapter 4.

A simple way of computing appropriate thresholds is by computing the values of d_{ori} and E at every location at a particular scale, and then setting the thresholds d_{ori}^{hign}, d_{ori}^{low} and E^* to be percentile values. Typically d_{ori}^{high} and d_{ori}^{low} may be set at the 85th and 50th percentiles for d_{ori}, respectively, and E^* set at the 30th percentile for E.

3.3.2 EM iteration for estimating oriented component weight

Once the oriented and non-oriented covariances have been calculated, the remaining parameter for the OAGSM/NC model is β. This is done by using the Expectation Maximization (EM) algorithm for a two-component mixture model. Unlike the prior densities for θ and z, this prior on δ will be fit from data for each subband. A seperate value of β will be estimated for each image subband. The estimated β will depend on the image patches for that subband, as well as the oriented and non-oriented component covariances. I consider the problem of estimating β from noisy patches, where the noise process is additive Gaussian of known covariance C_n. In this case, the model for noisy patches may be obtained by simply adding C_n to each of the component covariances.

For estimating β from data, it is helpful to view the OAGSM/NC model as a mixture of two component distributions, namely the oriented and non-oriented distributions. Define the oriented and non-oriented models for noisy patches as

$$\begin{aligned} P_{ori}(w) &= \int g(w; zC(\theta) + Cn)p(\theta)p(z)d\theta dz \\ P_{nor}(w) &= \int g(w; zC_{nor} + Cn)p(z)dz \end{aligned} \tag{3.49}$$

Then the OAGSM/NC model for noisy patches is

$$P(w) = \beta P_{ori}(w) + (1 - \beta)P_{nor}(w) \tag{3.50}$$

Given a collection of noisy patches $\{w_i\}_{i=1}^{m}$, the maximum likelihood estimate of β is given by

$$\beta_{ML} = \underset{\beta}{\operatorname{argmax}}\, L(\beta) = \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{m} (\log(\beta P_{ori}(w_i) + (1 - \beta)P_{nor}(w_i))) \tag{3.51}$$

Direct optimization of the above expression is complicated by the terms that are summed inside of the logarithm. An alternative, iterative approach for optimizing the likelihood function $L(w)$ is to use the EM algorithm [16]. The EM algorithm is an extremely general, widely used iterative method for performing Maximum Likelihood estimation for problems that have so-called “missing data”. When using EM for estimating the weights for a mixture model, such as the problem for β, the “missing data” is really an artificial construct that simply serves to simplify the problem. A very brief description of the EM methodology will be given below, following the treatment found in [33] and [4].

Assume the observed data X has probability distribution $p(X|\Phi)$ where Φ is the set of parameters that must be fit from data. The maximum likelihood estimate may be described as $\Phi^* = \text{argmax}_\Phi \log L(\Phi|X)$ where $L(\Phi|X) = p(X|\phi)$ is the "observed data likelihood" The EM method assumes that the observed data X can be "completed" by adjoining "missing data" Y to give the "complete data" (X, Y). This will only be useful if the "complete data likelihood" $p(X, Y|\Phi)$ is simpler to manipulate than the original observed data likelihood.

The EM algorithm produces a series of estimates Φ^k for the model parameters. As the "missing data" Y is not observed, the complete data likelihood $p(X, Y|\Phi)$ cannot be maximized over Φ, as this expression still refers to Y. The EM algorithm deals with this by taking the expectation over the missing data. This expectation is taken w.r.t $p(Y|X, \Phi^{k-1})$, i.e., conditioned on the observed data and the previously computed estimate of the parameters Φ. This is the "E-step" of the algorithm, namely computing

$$\begin{aligned} Q(\Phi, \Phi^{k-1}) &= E\left[\log(p(X, Y|\Phi)|X, \Phi^{k-1}\right] \\ &= \int \log(p(X, Y|\Phi))p(Y|X, \Phi^{k-1})dY \end{aligned} \tag{3.52}$$

As the X are observed and Y has been integrated out, the above function $Q(\Phi, \Phi^{k-1})$ does not refer to any unknown variables, and may thus be optimized over. This defines the maximization, or M step, to achieve the subsequent iterate

$$\Phi^k = \underset{\Phi}{\text{argmax}}\, Q(\Phi, \Phi^{k-1}) \tag{3.53}$$

It can be shown that the incomplete log-likelihood is guaranteed to increase at each iteration and that the EM algorithm will converge, possibly to a local maximum.

For the component mixture problem at hand, the incomplete data X consists of the m noisy patches w_i. Let W denote the collection of all of the w_i. The model parameters Φ consist of the single parameter β. The key "trick" for using EM in this case is setting up the "missing data" in a way that will simplify the resulting complete data likelihood. Choose the "missing data" Y to consist of a set of binary indicator variables τ_i^δ, where $i = 1...m$ and $\delta = 0, 1$. For each value of i, exactly one of these variables equals 1, indicating which component the i$^{\text{th}}$ sample was drawn from. E.g, if the i$^{\text{th}}$ sample came from the oriented component, then $\tau_i^0 = 0$ and $\tau_i^1 = 1$, otherwise $\tau_i^0 = 1$ and $\tau_i^1 = 0$. Let $\tau_i = (\tau_i^0, \tau_i^1)$ and $\vec{\tau}$ indicate the entire set of indicator variables. The complete data likelihood will be a product of the terms $p(w_i, \tau_i|\beta) = p(w_i|\tau_i, \beta)p(\tau_i|\beta)$. The introduced notation allows these to be written as

$$\begin{aligned} p(w_i|\vec{\tau}, \beta) &= \left(P_{nor}(w_i)\right)^{\tau_i^0} \left(P_{ori}(w_i)\right)^{\tau_i^1} \\ p(\tau_i|\beta) &= (1-\beta)^{\tau_i^0} \beta^{\tau_i^1} \end{aligned} \tag{3.54}$$

The second equation follows as β is the prior likelihood for drawing each sample from the oriented component. The complete data log-likelihood may thus be written as

$$\begin{aligned}\log p(W, \vec{\tau}|\beta) &= \sum_i \log \left((P_{nor}(w_i))^{\tau_i^0} \, (P_{ori}(w_i))^{\tau_i^1} \, (1-\beta)^{\tau_i^0} \beta^{\tau_i^1} \right) \\ &= \sum_i \tau_i^0 \log(P_{nor}(w_i)) + \tau_i^1 \log(P_{ori}(w_i)) + \tau_i^0 \log(1-\beta) + \tau_i^1 \log(\beta)\end{aligned} \tag{3.55}$$

A critical point here is that the "missing data" variables $\vec{\tau}$ appear *linearly* in the above expression. This implies that the expectation operation in the E step may be passed inside the above sum. The E-step may thus be performed by replacing each occurrence of τ_i^δ by $t_i^\delta(\beta) = E[\tau_i^\delta|\beta, w_i]$. Note that

$$\begin{aligned}E[\tau_i^\delta|\beta, w_i] &= 0 \times p(\tau_i^\delta = 0|\beta, w_i) + 1 \times p(\tau_i^\delta = 1|\beta, w_i) \\ &= p(\tau_i^\delta = 1|\beta, w_i)\end{aligned} \tag{3.56}$$

These may be computed using Bayes' rule. In particular

$$p(\tau_i^\delta = 1|\beta, w_i) = \frac{p(w_i|\tau_i^\delta = 1, \beta)p(\tau_i^\delta = 1|\beta)}{p(w_i|\beta)} \tag{3.57}$$

Evaluating these for $\delta = 0, 1$ shows

$$\begin{aligned}t_i^0(\beta) &= \frac{(1-\beta)P_{nor}(w_i)}{\beta P_{ori}(w_i) + (1-\beta)P_{nor}(w_i)} \\ t_i^1(\beta) &= \frac{\beta P_{ori}(w_i)}{\beta P_{ori}(w_i) + (1-\beta)P_{nor}(w_i)}\end{aligned} \tag{3.58}$$

which may easily be computed from the samples w_i. The E step may thus be written as

$$Q(\beta, \beta^{k-1}) = \sum_i t_i^0(\beta^{k-1}) \log(P_{nor}(w_i)) + t_i^1(\beta^{k-1}) \log(P_{ori}(w_i)) + \\ t_i^0(\beta^{k-1}) \log(1-\beta) + t_i^1(\beta^{k-1}) \log(\beta) \tag{3.59}$$

The M step sets $\beta^k = \text{argmax}_\beta = Q(\beta, \beta^{k-1})$. The first two terms for Q do not involve β and thus may be ignored. Setting $\frac{dQ}{d\beta} = 0$ yields

$$\frac{1}{\beta} \sum_i t_i^1(\beta^{k-1}) = \frac{1}{1-\beta} \sum_i t_i^0(\beta^{k-1}) \tag{3.60}$$

which has the solution

$$\beta^k = \frac{\sum_i t_i^1(\beta^{k-1})}{\sum_i t_i^0(\beta^{k-1}) + t_i^1(\beta^{k-1})} = \frac{1}{m} \sum_i t_i^1(\beta^{k-1}) \tag{3.61}$$

This has an appealing form. At every step, β is replaced by the average expected probability that each sample arose from the oriented component, conditioned on the previous iterate of β. Iterating this procedure converges to the ML estimate for β. In practice for the OAGSM/NC model, about 20 iterations are taken.

Chapter 4

Applications to Image Denoising

The OAGSM and OAGSM/NC models provide a description of the signal content of natural images. If they provide a good description of natural image signal, then they should be able to distinguish between natural image signal and other signals, such as noise. A commonly studied problem in image processing is image denoising, where one seeks to estimate the clean version of an image that has been corrupted with noise signal. If the noise process is additive, then any denoising algorithm can be viewed as partitioning a given noisy signal into noise and signal components. In order to perform this separation, an algorithm must have some notion of what typical image signals look like. All denoising methods rely on identifying and exploiting the differences between image and noise signal. There is thus a strong connection between image modeling and image denoising. In this section I derive a set of denoising methods based on the OAGSM and OAGSM/NC models.

Removing noise from images is an important practical engineering problem. All physical sensing devices are subject to some degree of random fluctuations in their outputs. Noise present in electronic sensors can be due to a number of phenomena, from thermal effects to "shot noise" due to counting a discrete number of electrons or photons in a circuit element. Familiar examples of this type of noise include digital photographs taken under low light conditions with high ISO settings. Noise can also be introduced in transmission of images, as is easily verified by watching an analog television set with poor reception. In addition, some deterministic distortions of images, such as quantization of the pixel intensity values, can be viewed as effectively introducing noise to the image. As some level of noise is inevitably introduced into any image produced by any sensing device, it is natural to study the development of post-processing techniques for removing it. For many imaging systems, reducing the intrinsic noise produced by the sensors would require higher quality, and more expensive, sensors. The ability to remove noise after the images have been acquired through processing in software can effectively increase the quality of the imaging device without the cost of using more expensive electronic component. Denoising image signals is clearly a real world problem.

As noise can arise from a number of different sources, there are many different types of noise. These can corrupt the signal in qualitatively different ways. A simple example is additive noise, where each image pixel can be viewed as a sum of the desired signal and the noise process. Another commonly studied case, often used to describe transmission errors, is impulse or "salt-and-pepper" noise, where a subset of randomly selected pixels are replaced by random values. More complicated situations could include multiplicative noise, where each pixel is corrupted by being multiplied by a random quantity, or when the noise process depends on the signal values.

Deriving a detailed description of noise that arises from a particular sensor such as a camera CCD may involve complicated physical modeling of the data acquisition process. If one were designing a noise removal algorithm for a particular device, it may be possible to measure empirical samples of the noise it produces. After such calibration, the statistics of the measured noise could be analyzed and modeled. This sort of highly specialized noise model would be useful for that particular device, but could not be broadly applied to other denoising applications.

An alternative route to follow is to simply assume a fixed distribution for the noise process that is not derived or measured from any particular class of physical sensors. A very commonly studied example of this is the case of signal independent additive Gaussian noise. While this model for the noise process may be inaccurate for certain particular applications, there are several advantages in picking a Gaussian noise model. An enormous number of different stochastic processes have been modeled as Gaussian in the statistics literature. Some theoretical justification for using Gaussian distributions is provided by appealing to the central limit theorem. Random processes that are the result of averaging many independent events will tend toward Gaussian as the number of independent sub-events increases, provided the variances of each of the independent events are finite. While it may be possible to view some noise processes as occurring in this way, the primary reason for using Gaussian noise models is their analytic tractability. Multivariate Gaussian densities are extraordinarily amenable to analytic manipulation, often allowing simple closed form results for calculations that arise in statistical estimation. Deviating from a Gaussian noise model may certainly make sense when studying a particular subset of noise processes for a particular application. However, noise signals encountered in practice are diverse, and may often reasonably approximated by Gaussians. Given their analytic simplicity, assuming a Gaussian noise model is reasonable when studying the denoising problem in general.

In this section I study the problem of denoising natural greyscale images that have been corrupted with additive Gaussian white noise. The noise component for each pixel is drawn independently from a one dimensional Gaussian with zero mean. This process will be homogenous, so that the variance in each pixel will be the same. This type of noise process is widely used in the image processing literature. I assume that the variance of the noise process is known. If this were not the case, then it would need to be estimated from the noisy image. This is the so-called “blind denoising” problem. A number of effective methods for estimating noise power from noisy images have been developed [41, 43]. These function essentially because natural images have spatially varying power, and thus typically contain regions of low signal power. The noise process, however, is homogenous and has the same local power across the entire image. One may thus estimate the properties of the noise by taking measurements from the lowest power regions of the noisy image, which will be dominated by noise.

In addition to the practical engineering concerns, the image denoising problem can provide a good test of the power of the underlying signal model used. The effectiveness of different denoising methods can be tested by artificially generating noise, adding it to clean images, and then comparing the results of the different methods. As one has access to the original clean images, it is possible to compute the residual error left in the different denoised images. This defines a clear numerical experimental methodology for comparing denoising methods. If these methods are based upon well formulated statistical models for image content, then the relative performance of the denoising methods provides a measure of how well the models themselves are capturing the relevant properties of natural image structure. Performing such experiments repeatedly requires the ability to artificially generate noise. Under these conditions the experimenter has exact control over the noise process supplied, and the noise model assumed in the denoising algorithms may be exactly correct. If ones primary interest is knowing the power of the underlying signal model, then for such an experiment it makes sense to use the simplest possible model for noise process. This provides another justification for the use of additive Gaussian white noise models.

The experiment described above relies on using some quantitative measure of the residual distortion present in each of the denoised images. Such a function may be called an "image quality metric", as it is used to measure the quality of the denoised image. One denoising algorithm will be superior to another if it produces "higher quality" denoised images, given the same level of initial noise. As images are ultimately intended to be looked at by human observers, however, this notion of "image quality" depends on how the human visual system perceives the distortion. Human perception of image distortion is complex and not completely understood. The development of more perceptually accurate image quality metrics is currently an area of ongoing research [64].

One of the simplest and most widely used measures of image distortion is mean squared error. Although the mean squared error is not always a good measure of visually perceived distortion, it is highly tractable mathematically. This is important, as one can design algorithms to optimize for the lowest mean squared error. Given a clean image I_c and a distorted image I_n, both defined on an $M \times N$ pixel lattice, the mean squared error (MSE) is

$$MSE(I_c, I_n) = \frac{1}{MN} \sum_{i=1,j=1}^{M,N} |I_c(i,j) - I_n(i,j)|^2 \tag{4.1}$$

Two commonly used related measures of image distortion are the Signal to Noise Ratio (SNR) and the so-called Peak Signal to Noise Ratio (PSNR). The empirical signal variance is calculated by $\sigma^2_{signal} = \frac{1}{MN-1} \sum (I_c(i,j) - \bar{I})^2$ where $\bar{I} = \frac{1}{MN} \sum I_c(i,j)$. Given the distorted image, the noise process in each pixel is $I_c(i,j) - I_n(i,j)$. Taking the sample variance of these gives the noise variance

$$\sigma^2_{noise} = \frac{1}{MN-1} \left[\sum (I_c(i,j) - I_n(i,j))^2 - \left(\sum I_c(i,j) - I_n(i,j) \right)^2 \right] \tag{4.2}$$

The Signal to Noise Ratio is then defined as

$$SNR = 10 \log_{10} \left(\frac{\sigma^2_{signal}}{\sigma^2_{noise}} \right) \tag{4.3}$$

The Peak Signal to Noise Ratio is defined for signals that have their outputs values constrained to a finite range $[I_{min}, I_{max}]$. This is the case for 8-bit greyscale images, which have pixel values between 0 and 255. The PSNR is defined to measure the noise variance relative to the maximum possible signal power for the input range, given by $(\Delta I)^2 = (I_{max} - I_{min})^2$. This gives

$$PSNR = 10 \log_{10} \left(\frac{(\Delta I)^2}{\sigma^2_{noise}} \right) \tag{4.4}$$

All of the images used in this thesis were 8-bit greyscale images, so PSNR values were computed using $\Delta I = 255$.

4.1 Bayesian Framework for Denoising

In this chapter, denoising will be treated as a statistical estimation problem. Given a noisy image I_n, one seeks to produce an estimated clean image $\hat{I}_c$ that satisfies certain optimality conditions. One can view the addition of noise as mapping clean images to noisy images. This mapping is not deterministic, however. As the noise process is a random event, there are many possible clean images I_c that could have been corrupted to give the observed noisy image I_n. One can view a denoising algorithm as selecting one particular estimate from this multitude of possible candidate clean images.

Some criterion is obviously necessary for this selection. The Bayesian approach to this problem works by constructing a probability density on this space of candidate clean images. Once this distribution, known as the "a posteriori" density, is known, calculation of the desired estimate can be chosen according to several different criteria. Common choices include picking the the estimate maximizing the a posterior density, or choosing an estimate that minimizes some cost functional averaged over the posterior. I review here some fundamental concepts for Bayesian signal estimation.

Let $x \in \mathbb{R}^d$ denote the signal of interest. For the image denoising problem, x could consist of the entire clean image, or it could represent some subset of measurements of the image such as Fourier components, or coefficients of a wavelet expansion of the clean image. In this thesis, many of the calculations will be performed where x is a generalized patch of Steerable Pyramid coefficients, as described in section 3.1. However, the basic Bayesian framework can be described independently of exactly what space x lives in. We assume that the original signal x has been transformed by some random process to give the corrupted signal y. In the case of additive Gaussian noise, this transformation is given by

$$y = x + n \tag{4.5}$$

where n is a Gaussian sample of known covariance C_n.

If the statistical properties of the noise are known and one knew the value of the original signal x, then the ensemble of all possible distortions of x defines a distribution on y. This probability, termed $p(y|x)$, serves to encapsulate what is known about the noise process. In the additive Gaussian noise case, given y and x the noise $n = y - x$ and it follows that

$$p(y|x) = g(y - x; C_n) \tag{4.6}$$

In the Bayesian approach, the signal x is itself modeled as a random process with distribution $p(x)$, called the signal prior. When one observes a particular noisy signal y, there are many different possible clean signals x that could have led to the given observation. Intuitively speaking, choosing a particular estimate for x should fuse information from both the given observation y, and from the prior $p(x)$. In the case of additive Gaussian noise, there are two competing desirable properties for the estimate of x. On the one hand x should be "close to" y, as large values of the noise n are increasingly unlikely. On the other hand, the estimate should "look like" signals drawn according to the signal prior, and should thus have large probability according to $p(x)$. If the noise level is very small, then it is highly unlikely that x differs greatly from y, and more emphasis should be placed on the observation. However as the noise level increases and the observation becomes increasingly unreliable, more emphasis should be placed on the prior.

The Bayesian method integrates these two conflicting desirable properties in a consistent probabilistic framework. The key quantity for Bayesian estimation is the posterior density, $p(x|y)$, which gives the probability that the observed signal y was generated by the original signal x. The approach gets its name from the use of Bayes rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{4.7}$$

which is used to compute the posterior density. Note that large values of $p(x|y)$ will occur when both $p(y|x)$ and $p(x)$ are large.

One obvious way of computing an estimator from the posterior density $p(x|y)$ is to chose x giving the greatest posterior probability. This is called the Maximum A Posterior (MAP) estimate

$$\hat{x} = \operatorname*{argmax}_{x} p(x|y) \tag{4.8}$$

Note that as the maximum does not depend on the normalizing probability $p(y) = \int p(y|x)p(x)$, the MAP estimator can be computed by maximizing $p(x)p(y|x)$. The fact that the normalizing constant may be disregarded often simplifies MAP estimation.

For a wide class of signal priors, the MAP estimator has an especially appealing form. If the signal prior $p(x)$ is of exponential type, then it has the form $p(x) \propto \exp(-L(x))$ where the function $L(x)$ can be viewed as a "cost function" penalizing some undesirable signal characteristics. This family includes Gaussian distributions with $L(x) = (1/2)x^T C_x^{-1} x$, and the generalized Gaussian distributions when $L(x) = a\,||x||^p$ for $0 < p < 2$. Functions L that measure norms of gradients of the signal x, and thus penalize signal discontinuities, are also common and often termed "smoothness priors". Given such an exponential prior, under the additive Gaussian white noise case with variance σ, the MAP estimate becomes

$$\begin{aligned} \hat{x} &= \underset{x}{\operatorname{argmax}} \exp(-L(x)) \exp(-\frac{1}{2\sigma^2}\,||y-x||^2) \\ &= \underset{x}{\operatorname{argmin}}\, L(x) + \frac{1}{2\sigma^2}\,||y-x||^2 \qquad (4.9) \end{aligned}$$

where the second equation follows from taking logarithms. This expression clearly shows how MAP estimation picks x that balances between minimizing the "data fidelity" term $||x-y||^2$ and the cost function $L(x)$ contained in the prior.

While taking the estimate for x to be the most probable is quite reasonable, there are other criteria that may be used for Bayesian estimation. In general, one wishes to avoid making errors when estimating the signal x. Errors are unpleasant, and undesirable. It may well be that different errors are not equally undesirable. If a quantitative measure of "displeasure" at making a certain type of error can be written as a cost function $C(\delta x)$, then one may seek to minimize the average cost in estimation. It may be possible to shift the estimation strategy by trading off accepting a larger number of negligible errors in order to avoid a small fraction of large, inordinately painful errors, in a manner that reduces the overall expected error. This sort of cost function is often called the "risk" in the statistics literature.

Once y is observed, one may seek to minimize the expected cost involved in picking the estimate $\hat{x}$. This expectation is taken over the posterior density. Define the expected cost

$$E(w) = \int p(x|y)C(x - w)dx \tag{4.10}$$

One may then pick the estimator $\hat{x} = \operatorname{argmin}_w E(w)$ giving the minimum expected cost.

Different cost functions may be designed for different situations, and may depend intricately on the nature of the underlying problem. As a fanciful example, imagine estimating from noisy sensor measurements the distance between the current location of a moving robot and a nearby brick wall. Underestimating the distance may result in a somewhat wasteful, overcautious motion strategy, while overestimating the distance may result in a crash and subsequent need to buy a new robot. If this estimation were performed by Bayesian methods, the the cost function C should reflect this underlying asymmetry.

For many problems, however, the cost function is simply chosen to be the square of the norm of the error, $C(w) = ||w||^2$. In this case, the estimator minimizing the average squared error is simply the a posterior mean. To see this, note that for this choice of C,

$$E(w) = \int p(y|x)\, ||x - w||^2\, dx \tag{4.11}$$

Taking the gradient of this expression with respect to w gives

$$\nabla E = 2 \int p(x|y)(x - w)dx = 2 \int xp(x|y)dx - 2w \tag{4.12}$$

Setting this to zero implies that the estimator minimizing the expected error is

$$w = \hat{x}(y) = \int xp(x|y)dx \tag{4.13}$$

This estimator is alternately known as the Bayes least squares (BLS) or the the Bayes minimum mean squared error (MMSE) estimator, for obvious reasons. The MAP and the BLS estimators thus calculate $\hat{x}$ by finding the mode and the mean, respectively, of the posterior distribution $p(x|y)$. They will thus be identical when the posterior is symmetric around its mean.

The denoising algorithms derived in this thesis are all based upon Bayesian least squares estimation, where the signal priors $p(x)$ are the OAGSM or OAGSM/NC models described in chapter 3. The BLS estimator is optimized to give the smallest average mean squared error of the resulting estimate. This is conceptually consistent with using mean squared error to evaluate the performance of the resulting denoising algorithms. However, one possible concern is that the BLS estimation described will be performed in the Steerable Pyramid domain, while the ultimate evaluation of the denoised image quality will be done by measuring PSNR in the pixel domain. As the Steerable Pyramid transform is overcomplete, the mapping from the coefficient domain to the pixel domain is not 1-1. There is thus no explicit relation between mean squared error in the coefficient domain and mean squared error in the pixel domain. It is conceivable that the BLS estimator may be optimal for mean squared error in the SP coefficient domain, but suboptimal in the pixel domain.

Fundamentally, both the OAGSM and the OAGSM/NC models can be described as Gaussian mixtures. As a prelude to discussing the full BLS estimators implied by the OAGSM and OAGSM/NC, I discuss the case when the signal model is a single multivariate Gaussian. In this case, the resulting BLS estimator is called the Wiener filter. This result will be used extensively in this thesis, and the calculation is outlined below.

Let x and n be zero mean Gaussians C_x and C_n respectively. Assuming the noise is independent of the signal, then the corrupted signal y will likewise be a zero mean Gaussian with covariance $C_x + C_n$. The posterior distribution is then

$$\begin{aligned} p(x|y) &= \frac{p(x)p(y|x)}{p(y)} \\ &= \frac{1}{p(y)} \frac{1}{(2\pi)^d |C_x|^{1/2} |C_n|^{1/2}} \exp\left(-\frac{1}{2}(x^T C_x^{-1} x)\right) \exp\left(-\frac{1}{2}((y-x)^T C_n^{-1} (y-x))\right) \\ &= \frac{1}{N} \exp\left(-\frac{1}{2}(x^T (C_x^{-1} + C_n^{-1}) x - 2y^T C_n^{-1} x + y^T C_n^{-1} y)\right) \qquad (4.14) \end{aligned}$$

While this expression appears complicated, one can take advantage of the fact that by construction it is guaranteed to be a properly normalized probability density. As the terms inside the exponential are polynomial in x with only up to quadratic terms, the density $p(x|y)$ is Gaussian. We wish to calculate its mean. However, the mean of a properly normalized multivariate Gaussian density will always be given by the minimum of the quadratic form found inside its exponential. Using this argument, one can sidestep some tedious algebra and simply find the minimum of the quadratic expression above. Taking the gradient of

$$Q(x) = x^T(C_x^{-1} + C_n^{-1})x - 2y^T C_n^{-1} x + y^T C_n^{-1} y \tag{4.15}$$

gives $\nabla_x Q = 2C_x^{-1} + C_n^{-1} - 2C_n^{-1}$. Setting this to zero gives the BLS estimate

$$\hat{x}_{BLS}(y) = \left(C_x^{-1} + C_n^{-1}\right)^{-1} C_n^{-1} y \tag{4.16}$$

Using the matrix identity $(A^{-1} + B^{-1})^{-1} = A(A+B)^{-1}B$ (see [49]), this may be written in the form

$$\hat{x}_{BLS}(y) = C_x(C_x + C_n)^{-1} y \tag{4.17}$$

This expression is known as the Wiener filter.

4.1.1 BLS estimator for Gaussian mixture models

Both the OAGSM and the OAGSM/NC models are mixtures of zero-mean multivariate Gaussian components where the covariances of each component are functions of a set of hidden variables. The mixing weights for these components, also referred to as the hidden variable priors, are clearly also functions of this same set of hidden variables. For this general type of model, the Bayes Least Squares estimate can be calculated in a particularly simple form. The resulting estimator is expressed as a weighted combination of the Wiener estimates for each of the underlying Gaussian components, where each weight term is related to the probability that the given noisy sample arose from that component.

Introduce the following general notation. Let $\vec{\tau}$ represent some collection of hidden variables used to construct the mixture model. For the OAGSM model, $\vec{\tau} = (z, \theta)$, while for the OAGSM/NC model we have $\vec{\tau} = (z, \theta, \delta)$. Assume for each value of $\vec{\tau}$ there is a model covariance $C(\vec{\tau})$. For the OAGSM and OAGSM/NC models, the covariances $C(\vec{\tau})$ have very specific functional forms that are related to the interpretation of the hidden variables, i.e. z acts by scalar multiplication and θ by rotation. The calculation of the BLS estimator, however, does not depend on this particular functional form. For this calculation, $C(\vec{\tau})$ could be a completely arbitrary mapping from the space of hidden variables to positive definite matrices.

Let $p(\vec{\tau})$ be the prior density over these hidden variables. The general form of the mixture density is then

$$p(x) = \int p(x|\vec{\tau})p(\vec{\tau})d\vec{\tau} = \int g(x; C(\vec{\tau}))p(\vec{\tau})d\vec{\tau} \tag{4.18}$$

The above expression is written as a continuous integral, however the hidden variables may be sampled discretely. This will in fact be the case for the implementation in this thesis. In this case the integral expressions would reduce to finite sums. As the underlying theory is unchanged, the more general expressions written with integrals will be used in this section.

The BLS estimate $\hat{x}(y)$ will be the a posterior mean $\int xp(x|y)dx$. The posterior $p(x|y)$ can be decomposed by conditioning on, then integrating over, the hidden variables.

$$p(x|y) = \int p(x|y, \vec{\tau})p(\vec{\tau}|y)d\tau \tag{4.19}$$

Substituting this into the a posterior mean and exchanging the order of integration shows

$$\begin{aligned}\hat{x}(y) &= \int x \left(\int p(x|y,\vec{\tau}) p(\tau|y) d\vec{\tau} \right) dx \\ &= \int \left(\int x p(x|y,\vec{\tau}) dx \right) p(\vec{\tau}|y) d\vec{\tau} \qquad (4.20)\end{aligned}$$

The interior integral over x is exactly the form of the BLS estimate of the signal x in the presence of noise, when conditioned on fixed values of the hidden variables $\vec{\tau}$. However, when conditioned on $\vec{\tau}$, x is Gaussian with covariance $C(\vec{\tau})$. As shown before, this type of estimate is given by the Wiener filter. It follows that

$$\int x p(x|y,\vec{\tau}) dx = C(\vec{\tau})(C_n + C(\vec{\tau}))^{-1} y = W_{\vec{\tau}} y \qquad (4.21)$$

where we have set $W_{\vec{\tau}} = C(\vec{\tau})(C_n + C(\vec{\tau}))^{-1}$ to be the Wiener filter corresponding to $\vec{\tau}$. The full BLS estimate for x is then

$$\hat{x}(y) = \int (W_{\vec{\tau}} y) p(\vec{\tau}|y) d\tau \qquad (4.22)$$

This is a weighted average of different Wiener estimates, where the weighting is controlled by $p(\vec{\tau}|y)$. It is really this weighting which allows the denoising algorithm to "adapt" to different local conditions. For example, in the OAGSM model $\vec{\tau}$ consists of z and θ. For noisy signal patches that are best described with power z^* and orientation θ^*, the weights $p(z,\theta|y)$ will be larger for values z and θ closer to z^* and θ^*, and smaller otherwise. The Wiener estimates $W_{z,\theta} y$ will be more accurate when z and θ are close to z^* and θ^* that best describe the signal. As a result, the full estimate $\hat{x}(y)$ will contain more contribution from the Wiener estimates that are more appropriate for the current noisy signal.

Expanding the weighting terms $p(\tau|y)$ by Bayes' theorem yields

$$p(\vec{\tau}|y) = \frac{p(y|\vec{\tau})p(\vec{\tau})}{p(y)} = \frac{p(y|\vec{\tau})p(\vec{\tau})}{\int p(y|\vec{\tau})p(\vec{\tau})d\vec{\tau}} \tag{4.23}$$

In the above expression, $p(\vec{\tau})$ are the hidden variable priors and are specified as part of the model. $p(y|\vec{\tau})$ is the distribution of the noise, assuming that the signal x was drawn from the Gaussian with covariance $C(\vec{\tau})$. As the signal and noise are independent, the covariances add and $p(y|\vec{\tau}) = g(y; C_n + C(\vec{\tau}))$. Substituting these into (4.22) gives the full BLS estimate

$$\hat{x}(y) = \frac{1}{N} \int (W_{\vec{\tau}}y) g(y; C_n + C(\vec{\tau}))p(\vec{\tau})d\vec{\tau} \tag{4.24}$$

where the normalizing constant $N = \int g(y; C_n + C(\vec{\tau}))p(\vec{\tau})d\vec{\tau}$.

4.1.2 Hidden variable prior densities

The above expression for the full BLS estimator involves an integral over the hidden variables, referencing the hidden variable prior density $p(\vec{\tau})$. For the OAGSM model, this set consists of $\vec{\tau} = (z, \theta)$ where for the OAGSM/NC we have $\vec{\tau} = (z, \theta, \delta)$. For the original GSM denoising method of Portilla et, $\vec{\tau}$ consists only of z. The hidden variable priors must be specified in order to complete the description of the estimation procedure. For this work, the hidden variables were sampled at a finite number of discrete points. The integrals in the expression for the BLS estimate thus reduce to finite sums. Let N_θ and N_z denote the number of sample points for z and θ. The δ hidden variable for the OAGSM/NC was originally defined as a binary variable, and so naturally has only two sample points.

I take the priors to be separable. For the OAGSM, the prior is $p(z,\theta) = p(z)p(\theta)$. All of the oriented covariances used in this thesis are π periodic, i.e. $C(\theta) = C(\theta+\pi)$. Accordingly θ may be sampled on the range $[0,\pi]$. I use the sample values $\theta_n = \frac{n-1}{\pi}$ for $n = 1...N_\theta$, and set $p(\theta_n) = \frac{1}{N_\theta}$, the discrete version of the uniform density on $[0,\pi]$.

Following [42], the prior on z is derived from the so-called Jeffrey's non-informative pseudo-prior $p(z) \propto \frac{1}{z}$. This "density" cannot be normalized unless z is truncated within some range $[z_{min}, z_{max}]$. The Jeffrey's pseudo-prior is equivalent to placing a uniform density on $\log z$. As we are sampling z finitely, this is implemented by choosing samples of z_n uniformly logarithmically spaced between z_{min} and z_{max}. Accordingly, I set $p(z_n) = \frac{1}{N_z}$ with

$$z_n = \exp\left(\log(z_{min}) + (n-1)\frac{\log(z_{max}) - \log(z_{min})}{N_z - 1}\right) \tag{4.25}$$

for $n = 1...N_z$. For the denoising results in this thesis I use $\log(z_{min}) = -20.5$ and $\log(z_{max}) = 3.5$.

For the OAGSM/NC, the prior is $p(z,\theta,\delta) = p(z)p(\theta)p(\delta)$. The same priors are used for z and θ as for the OAGSM. For this model, the prior on the binary variable δ is determined by $p(\delta = 1) = \beta$, where the parameter β is the prior probability of drawing each patch from the oriented process. β is estimated for each subband and at each spatial scale, as described in section 3.3

4.2 Denoising Algorithm

Denoising is performed in the wavelet coefficient domain. Noisy images are first decomposed with the Steerable Pyramid. From these noisy coefficients, the parameters for either the OAGSM or OAGSM/NC models are calculated, as described in section 3.2. Several variants of the models are possible, depending on whether the oriented covariances are estimated by patch rotation, using an edge model or using the 1-d empirical covariances.

At each spatial scale, noisy patches y of a fixed generalized patch geometry are extracted. These noisy patches are then denoised according to the BLS estimation procedure described above. Note that the BLS estimator $\hat{x}(y)$ produces an estimate of the entire generalized patch. One possible method for denoising would be to partition the noisy coefficients into non-overlapping square patches, denoise each patch, and then invert the transform. Doing this is likely to introduce block boundary artifacts in each subband. An alternative approach, taken in this thesis, is to take only the center coefficient of each estimate. In this way, each coefficient is estimated using a generalize patch centered on it. These patches are overlapping, as shown in figure 4.1.

The highpass and lowpass bands of the pyramid are treated differently. The highpass band residual band is a scalar quantity, and the highpass filters are not steerable, which makes it difficult to estimate OAGSM covariances for it. Accordingly, the highpass band is denoised with the GSM method, exactly following Portilla [42]. This method may viewed as a degenerate case of the OAGSM, where the covariances $C(\theta)$ do not have any θ dependence. The lowpass band typically has a much higher signal to noise ratio. This follows as

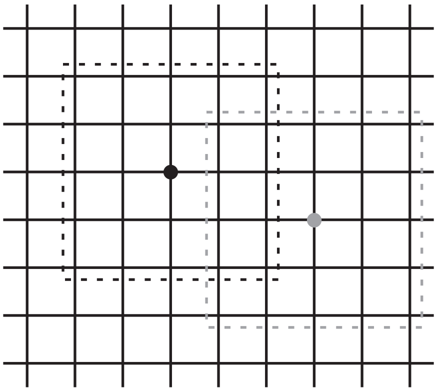


Figure 4.1: Overlapping patches. Each coefficient is denoised using a neighborhood centered on it.

the white noise process has a flat power spectrum, while typical image spectral power is proportional to $\frac{1}{w^p}$ for p close to 2. For lower spatial frequencies, the signal power will increase and dominate the noise process. Another issue is that for coarser spatial scales, estimating the model parameters becomes more difficult as there are fewer available signal patches to fit the parameters from. This suggests that their is an effective limit to the depth of spatial scales it makes sense to try to denoise. For this work, the pyramid representation is built to a depth of three spatial scales, and the lowpass band is simply left unchanged. Once each coefficient has been estimated, the entire transform is inverted to give the resulting denoised image.

4.2.1 Calculating noise covariances in each subband

While the noise process is assumed to be white in the pixel domain, the noise process in each pyramid subband has been shaped by the SP filters. As the transformation from the pixel domain to the generalized patches at each spatial scale is linear, the noise process in each patch will be Gaussian, but no longer white. One simple way of computing the noise covariances would be to take a sample of the noise process in the image domain, compute its SP transform, extract all of the noisy patches and take their sample average. Doing the calculation this way will give an estimate of the noise covariance with some error, as it is sampled empirically. One may avoid this by computing the covariance in the following way, instead. Following the notation in section 3.2.2, let $n \in Im$ denote the noise sample in the image domain, $p \in \mathbb{R}^d$ represent a generalized wavelet patch, and $T : Im \rightarrow \mathbb{R}^d$ be the patch measurement operator, so that $p = Tn$. The desired noise covariance is then

$$C_n = E\left[pp^T\right] = TE[nn^T]T^T \tag{4.26}$$

If the noise process is white with standard deviation σ_n, then $E[nn^T]$ will be σ_n^2 times an $N \times N$ identity matrix, where N is the number of image pixels. The $(i,j)^{\text{th}}$ element of the noise covariance C_n is then $\sigma_n^2 T_i T_j^T$. $T_i \in \mathbb{R}^N$ is given by the response of the filter corresponding to position i to a unit impulse in the pixel domain. Accordingly, all of the products $T_i T_j^T$ may be computed in one step by taking the SP transform of a unit impulse, extracting all of the patches of specified geometry at the desired scale, and taking their sample outer product. Multiplying the resulting matrix by σ_n^2 then gives the noise covariance.

4.2.2 Results

The OAGSM and OAGSM/NC denoising methods were applied to a collection of 10 greyscale images corrupted with 3 different levels of Gaussian white noise.

The images were originally taken in color with a Pentax Optio S4 digital camera, and were later cropped and downsampled to 512x512 pixels and converted to greyscale for use as test images for this thesis. No attempt at camera calibration, or control for different lighting conditions was made. The original greyscale pixel values were numbers between 0 and 255. Noise levels with standard deviation $\sigma = 20, 40$ and 80 were used for the denoising experiments.

Both of the denoising methods have a significant number of parameters, aside from the model covariances, that must be specified. These include the size of the generalized patches, the number of orientation bands used in the Steerable Pyramid decomposition, whether winnowing by orientedness should be used for selecting patches for fitting the model covariances, as well as the number of discrete samplings used for the hidden variables z and θ. 13 points were used for the z hidden variable, and 16 points for the θ hidden variable. It was found that increasing the sampling density of the hidden variables above these values led to minimal improvement in performance.

As described in section 3.1, determining the optimal geometry of the generalized patches used for the OAGSM and OAGSM/NC models should be done empirically. The performance of the OAGSM/NC model was checked for a large number of different patch geometries. The spatial neighborhoods are chosen to be symmetric around a single central coefficient. If square patches are used, then the dimensions must be odd. 6 different patch geometries, consisting of 3x3, 5x5 and 7x7 patches with and without parent bands, were tried. Results of these calculations are tabulated in table 4.1. The best performance was given by either 5x5 or 7x7 patches with parents included. The difference in performance between these two was slight. 5x5 patches with a single parent coefficient were

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
3x3 np	27.246	28.552	34.473	32.404	29.869	25.963	32.284	32.124	32.095	27.855
5x5 np	27.407	28.774	34.639	32.706	30.121	26.083	32.571	32.285	32.425	28.025
7x7 np	27.445	28.756	34.682	32.721	30.200	26.134	32.588	32.311	32.449	28.060
3x3 p	27.326	28.657	34.567	32.467	30.016	26.043	32.397	32.240	32.237	27.961
5x5 p	27.438	**28.824**	34.704	**32.722**	30.207	26.122	**32.641**	32.357	**32.529**	28.068
7x7 p	**27.457**	28.786	**34.735**	32.714	**30.246**	**26.153**	32.630	**32.361**	32.522	**28.076**

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
3x3 np	24.061	25.033	31.556	28.993	26.580	22.829	28.709	29.491	28.495	24.672
5x5 np	24.190	25.246	31.714	29.248	26.771	22.979	28.935	29.566	28.805	24.822
7x7 np	24.252	25.317	31.745	29.295	26.853	23.054	28.953	29.601	28.865	24.893
3x3 p	24.165	25.190	31.643	29.120	26.745	22.941	28.889	29.592	28.678	24.811
5x5 p	24.264	25.361	31.754	29.312	26.887	23.054	**29.052**	29.641	28.925	24.919
7x7 p	**24.298**	**25.396**	**31.773**	**29.337**	**26.937**	**23.094**	29.038	**29.652**	**28.962**	**24.960**

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
3x3 np	21.716	22.145	28.362	25.756	23.696	20.548	25.472	27.072	25.301	22.230
5x5 np	21.796	22.285	28.486	25.992	23.839	20.651	25.620	27.109	25.463	22.273
7x7 np	21.867	22.363	28.477	25.997	23.900	20.741	25.635	27.101	25.465	22.359
3x3 p	21.794	22.274	28.470	25.871	23.819	20.633	25.601	27.117	25.423	22.324
5x5 p	21.856	22.380	**28.532**	**26.036**	23.926	20.721	**25.717**	**27.150**	**25.544**	22.350
7x7 p	**21.911**	**22.433**	28.500	26.021	**23.966**	**20.790**	25.713	27.137	25.540	**22.418**

Table 4.1: Table of denoising results, by PSNR, for different patch geometries, with starting noise levels $\sigma = 20$ (PSNR 22.0977 dB, top), $\sigma = 40$ (PSNR=16.0771, middle), $\sigma = 80$ (PSNR 10.0565, bottom). All results are for the OAGSM/NC model with covariances from patch rotation, without winnowing

selected for use for the remainder of the denoising calculations in this thesis.

The neighborhoods described above, and used for all of the results in this thesis, did not include any "cousin" coefficients from other oriented bands at the same spatial scale. Accordingly, the denoising calculations at each spatial scale must be run K times for each of the K orientation bands used in the SP transform. Inclusion of cousin coefficients was tried, by taking the generalized neighborhood to include all of the coefficients for all orientation bands within a specified patch geometry. For instance, with 5x5 patches including parents for the SP transform with 2 orientation bands, each such neighborhood would include 52 coefficients. In this case the BLS estimation would be performed in a 52 dimensional space, and the two central coefficients would be kept. This is termed "joint" estimation as all of the orientation bands are estimated in a single calculation. It was found that the resulting denoising methods performed much worse than those based on generalized patches without inclusion of cousin coefficients.

Using the selected patch geometry, the OAGSM and OAGSM/NC algorithms were run on the test image set. For both methods, four different ways of calculating the underlying signal covariances were tried. The oriented covariances were calculated by either the patch rotation method, by the step edge model method or by the 1-d empirical covariance method, as described in section 3.2. Denoising was done using the patch rotation method both with and without winnowing. When winnowing was used, the oriented covariances were calculated using patches with d_{ori} above the 85^{th} percentile threshold. The edge model covariances were computed using the step edge profile. It was observed that the step edge profile gave slightly better performance than using the line edge profile. For the 1-d empirical covariance method, the underlying correlation functions were estimated by averaging over the ensemble of the 10 clean images. This was done in an attempt to measure a single, non-adaptive, set of correlation functions in that generate the oriented covariances.

The denoising method based on the Gaussian Scale Mixture (GSM) was used as a baseline for evaluating the performance of the current methods. The GSM results shown here were obtained using exactly the same implementation

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	27.394	28.516	34.463	32.352	29.918	26.064	32.067	32.246	32.081	27.998
O/rot/w	-0.211	0.172	0.158	0.283	0.044	-0.139	0.464	-0.059	0.365	-0.165
O/rot/nw	-0.079	0.241	0.161	0.228	0.249	0.016	0.529	-0.001	0.432	-0.045
O/edge	-1.915	-1.565	-0.706	-1.083	-1.307	-1.789	-1.064	-0.887	-0.943	-1.906
O/1d-emp	-2.046	-2.316	-0.846	-1.802	-1.724	-1.876	-1.548	-1.021	-2.063	-2.079
NC/rot/w	0.060	0.357	0.181	0.400	0.334	0.059	0.603	0.133	0.520	0.093
NC/rot/nw	0.044	0.308	0.241	0.370	0.289	0.057	0.574	0.111	0.448	0.070
NC/edge	0.036	0.268	0.188	0.305	0.224	0.025	0.503	0.102	0.433	0.051
NC/1d-emp	0.024	0.117	0.065	0.158	0.106	0.018	0.251	0.067	0.197	0.024

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	24.251	25.130	31.518	29.031	26.664	23.013	28.668	29.635	28.633	24.871
O/rot/w	-0.192	0.082	0.062	0.163	0.014	-0.153	0.229	-0.181	0.206	-0.172
O/rot/nw	-0.095	0.176	0.177	0.202	0.153	-0.041	0.320	-0.117	0.222	-0.060
O/edge	-1.020	-0.923	-0.113	-0.515	-0.678	-0.991	-0.611	-0.336	-0.517	-1.150
O/1d-emp	-1.169	-1.157	-0.296	-0.741	-0.798	-1.110	-0.709	-0.550	-1.215	-1.160
NC/rot/w	0.046	0.281	0.040	0.222	0.244	0.072	0.355	-0.044	0.285	0.090
NC/rot/nw	0.014	0.231	0.236	0.280	0.223	0.041	0.384	0.006	0.292	0.047
NC/edge	-0.010	0.182	0.218	0.209	0.190	0.015	0.361	0.036	0.283	0.023
NC/1d-emp	-0.017	0.083	0.106	0.053	0.091	0.012	0.168	0.006	0.114	0.005

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	21.913	22.280	28.315	25.790	23.736	20.717	25.406	27.048	25.374	22.383
O/rot/w	-0.215	-0.060	-0.298	0.022	0.011	-0.130	0.104	-0.238	-0.034	-0.230
O/rot/nw	-0.155	0.024	0.176	0.168	0.123	-0.076	0.254	0.044	0.078	-0.139
O/edge	-0.428	-0.399	0.163	-0.018	-0.090	-0.396	0.041	0.066	-0.113	-0.504
O/1d-emp	-0.622	-0.609	-0.010	-0.182	-0.257	-0.554	-0.112	-0.100	-0.614	-0.566
NC/rot/w	-0.060	0.084	-0.363	0.032	0.126	0.015	0.143	-0.256	0.008	-0.047
NC/rot/nw	-0.057	0.099	0.217	0.246	0.190	0.004	0.311	0.102	0.170	-0.033
NC/edge	-0.056	0.083	0.213	0.233	0.192	-0.004	0.294	0.129	0.184	-0.025
NC/1d-emp	-0.066	0.027	0.140	0.110	0.125	-0.006	0.183	0.115	0.091	-0.031

Table 4.2: Comparison of GSM with OAGSM and OAGSM/NC variants, based on using Steerable Pyramid with 2 orientation bands. GSM denoised values are given in PSNR, other methods are relative to GSM baseline. Results presented for noise levels $\sigma = 20$ (PSNR 22.0977 dB, top), $\sigma = 40$ (PSNR=16.0771, middle), $\sigma = 80$ (PSNR 10.0565, bottom).

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	27.426	28.732	34.799	32.565	30.200	26.136	32.447	32.384	32.445	28.068
O/rot/w	-0.112	0.133	0.031	0.234	-0.015	-0.102	0.363	-0.049	0.352	-0.091
O/rot/nw	-0.038	0.133	0.030	0.232	0.087	-0.030	0.363	-0.009	0.302	-0.031
O/edge	-1.540	-1.325	-0.762	-0.890	-1.285	-1.447	-1.066	-0.971	-0.859	-1.461
O/1d-emp	-5.510	-7.311	-5.306	-9.070	-6.825	-5.348	-7.432	-4.247	-7.803	-5.943
NC/rot/w	0.040	0.225	0.026	0.291	0.168	0.015	0.427	0.071	0.409	0.051
NC/rot/nw	0.027	0.161	0.063	0.265	0.113	0.015	0.356	0.050	0.277	0.027
NC/edge	0.029	0.104	-0.020	0.146	0.051	0.009	0.175	0.032	0.192	0.020
NC/1d-emp	0.022	0.025	-0.083	0.074	0.007	0.008	0.049	0.021	0.037	0.011

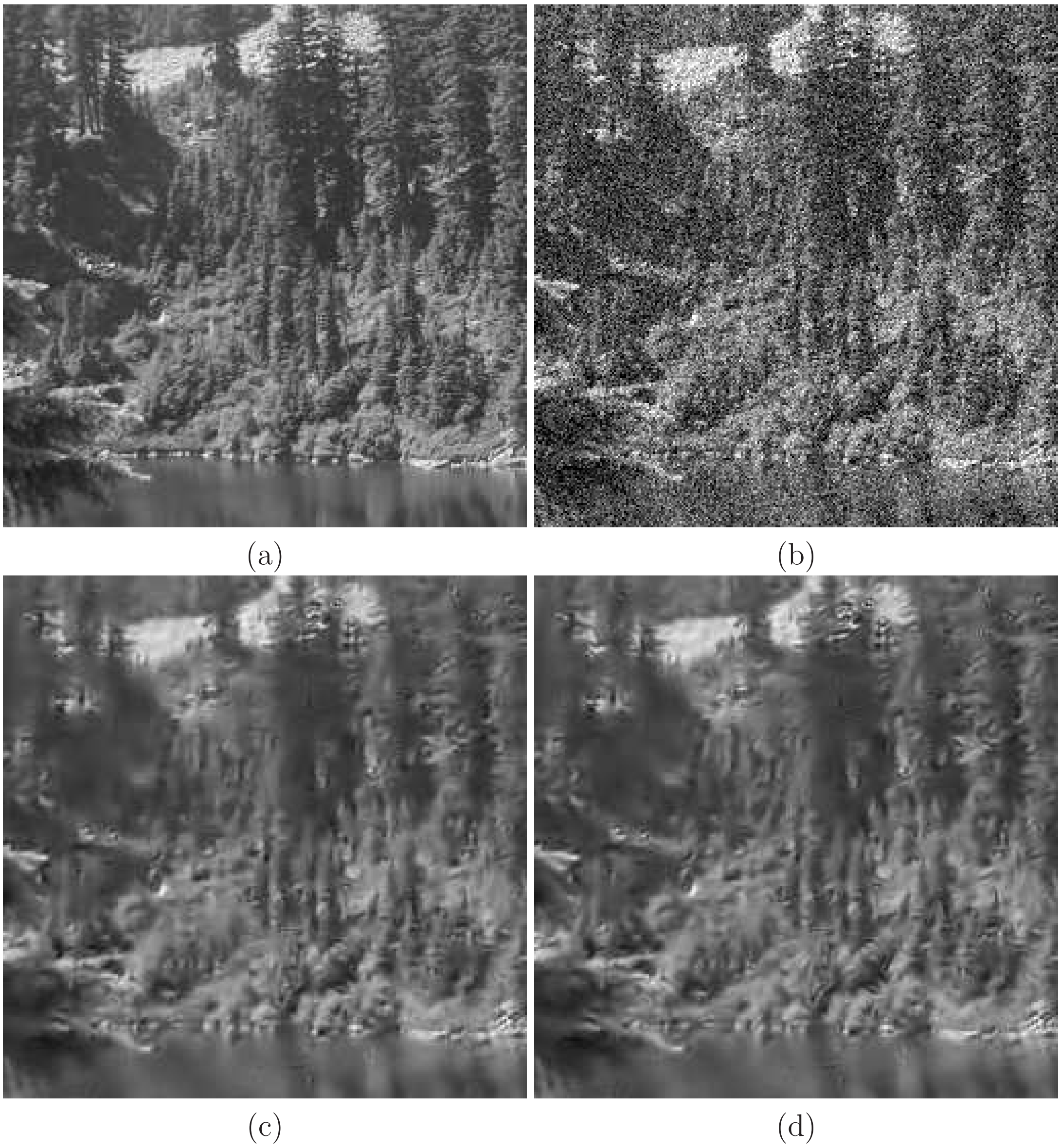
	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	24.301	25.364	31.797	29.322	26.994	23.104	29.056	29.781	28.997	24.971
O/rot/w	-0.134	0.057	0.048	0.096	-0.089	-0.135	0.146	-0.139	0.198	-0.126
O/rot/nw	-0.089	0.057	0.106	0.137	-0.020	-0.084	0.150	-0.103	0.153	-0.073
O/edge	-1.082	-1.015	-0.307	-0.596	-0.899	-1.035	-0.848	-0.538	-0.653	-1.121
O/1d-emp	-3.335	-4.852	-3.197	-6.159	-4.516	-3.296	-4.890	-2.388	-5.145	-3.763
NC/rot/w	0.019	0.166	-0.054	0.142	0.063	0.020	0.166	-0.079	0.200	0.038
NC/rot/nw	-0.007	0.106	0.130	0.161	0.041	-0.001	0.158	-0.031	0.176	0.004
NC/edge	-0.001	0.068	0.068	0.020	-0.001	0.005	0.062	-0.020	0.102	0.007
NC/1d-emp	-0.004	0.020	-0.001	-0.073	-0.035	0.007	-0.043	-0.035	-0.006	0.007

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	21.989	22.504	28.435	26.047	24.041	20.823	25.685	27.135	25.650	22.489
O/rot/w	-0.188	-0.081	-0.157	0.063	-0.048	-0.143	0.097	-0.143	-0.010	-0.202
O/rot/nw	-0.158	-0.059	0.200	0.148	-0.026	-0.118	0.153	0.053	0.012	-0.159
O/edge	-0.590	-0.619	0.053	-0.169	-0.357	-0.569	-0.272	-0.012	-0.341	-0.648
O/1d-emp	-1.668	-2.812	-1.276	-3.843	-2.393	-1.615	-2.535	-0.807	-2.734	-1.940
NC/rot/w	-0.088	0.001	-0.344	-0.010	-0.017	-0.032	0.035	-0.262	-0.034	-0.096
NC/rot/nw	-0.074	0.014	0.193	0.178	0.037	-0.034	0.177	0.080	0.103	-0.069
NC/edge	-0.058	-0.002	0.143	0.098	0.040	-0.024	0.122	0.095	0.077	-0.043
NC/1d-emp	-0.060	-0.025	0.100	0.007	0.009	-0.015	0.065	0.069	0.030	-0.041

Table 4.3: Same as table 4.2, using Steerable Pyramid with 8 orientation bands

as published in [42], which gave state-of-the art performance at the time of its publication in 2003. Comparing to the GSM method is reasonable as the GSM is similar to the current methods, but without adaptation to local orientation. Thus the performance gain over the GSM really measures the specific benefit that is gained by including orientation as a hidden variable. All of the methods, the GSM, OAGSM and OAGSM/NC, can be implemented on top of the Steerable Pyramid with any number of orientation bands. Results are shown using the 2 band pyramid in table 4.2 and using the 8 band pyramid in table 4.3.

For all of these methods, it is observed that denoising performance generally increases as the number of underlying SP orientation bands is increased. Increasing the number of orientation bands for the SP yields SP filters with higher orientation specificity, but also makes the overall SP transform more overcomplete. It has been observed that simply increasing the redundancy of the underlying basis in which one performs denoising often leads to increased denoising performance. This observation is the basis for using so-called "cycle-spinning" with undecimated wavelet bases for denoising [14]. However, increasing the orientation specificity of the underlying SP filters is also likely to improve denoising performance, especially for the GSM model which has no other method for explicitly adapting to the local orientation of the signal. It is thus difficult to separate the effects of both increased redundancy and increased orientation specificity that arise from using a higher order SP transform. It has been observed that for the denoising methods presented here, the improvement in performance obtained from using a SP transform with more than about 8 orientations bands is very minimal. The benefits of adaptation to local orientation, however, are more strongly apparent in the denoising results based on



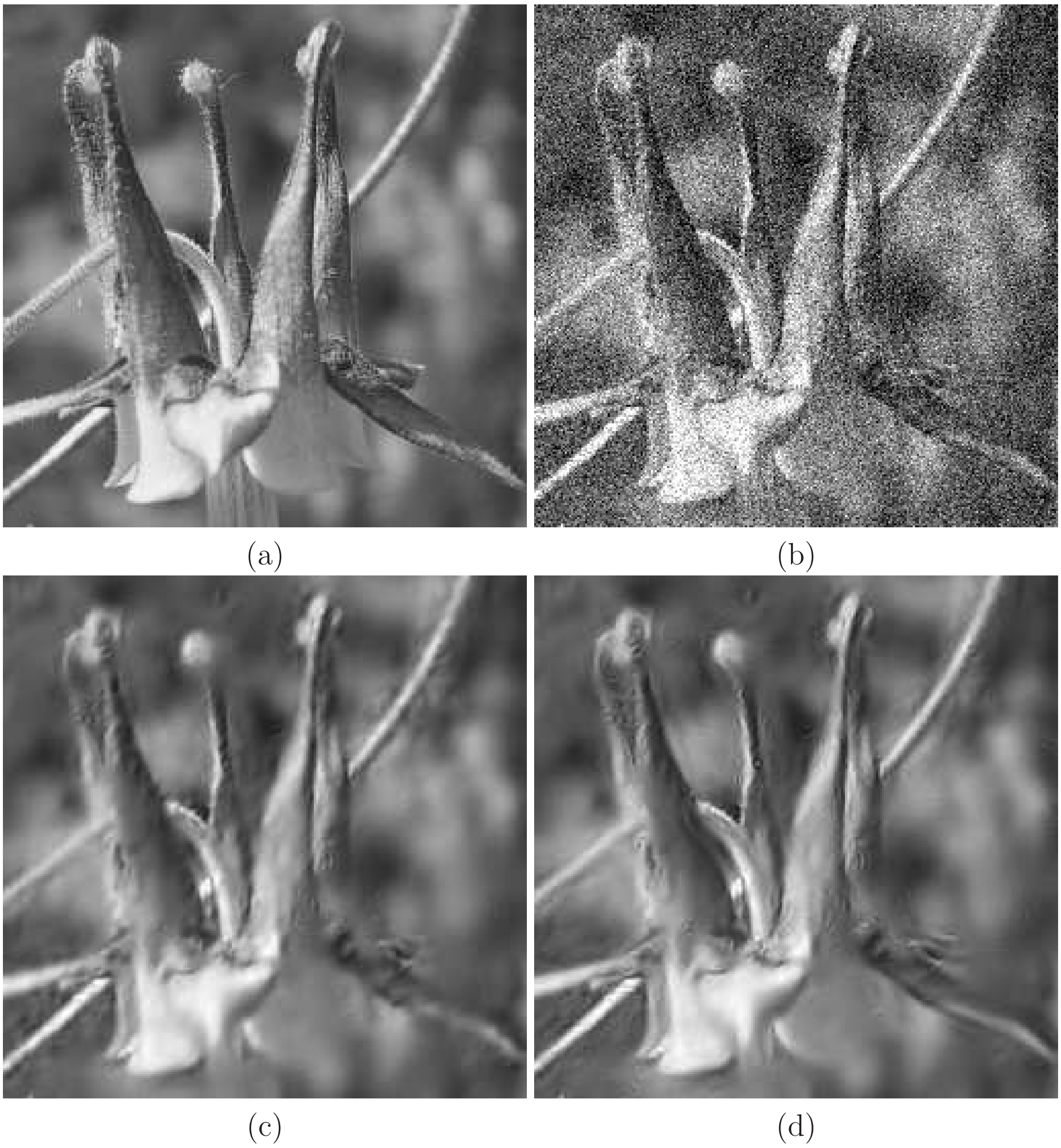
(a) (b)

(c) (d)

Figure 4.2: (a) Image #1 original (b) Noisy $\sigma = 40$ (16.0771) (c) GSM (24.251) (d) OAGSM/NC (24.297)

the 2-band SP. Results are thus presented in this thesis for the algorithms using 2-band SP and for the 8-band SP.

Referring to these tables, it is clear that the OAGSM/NC method consistently outperforms the GSM method. This holds for both the 2-band calcula-



(a) (b)

(c) (d)

Figure 4.3: (a) Image #4 original (b) Noisy $\sigma = 40$ (16.0771) (c) GSM (29.031) (d) OAGSM/NC (29.253)

tions and the 8-band calculations. The performance difference is greater for the 2-band case, with the OAGSM/NC method showing up to 0.6 dB improvement for some images. Performing winnowing by orientedness provides some benefit at low noise levels, but leads to worse performance at higher noise levels. The

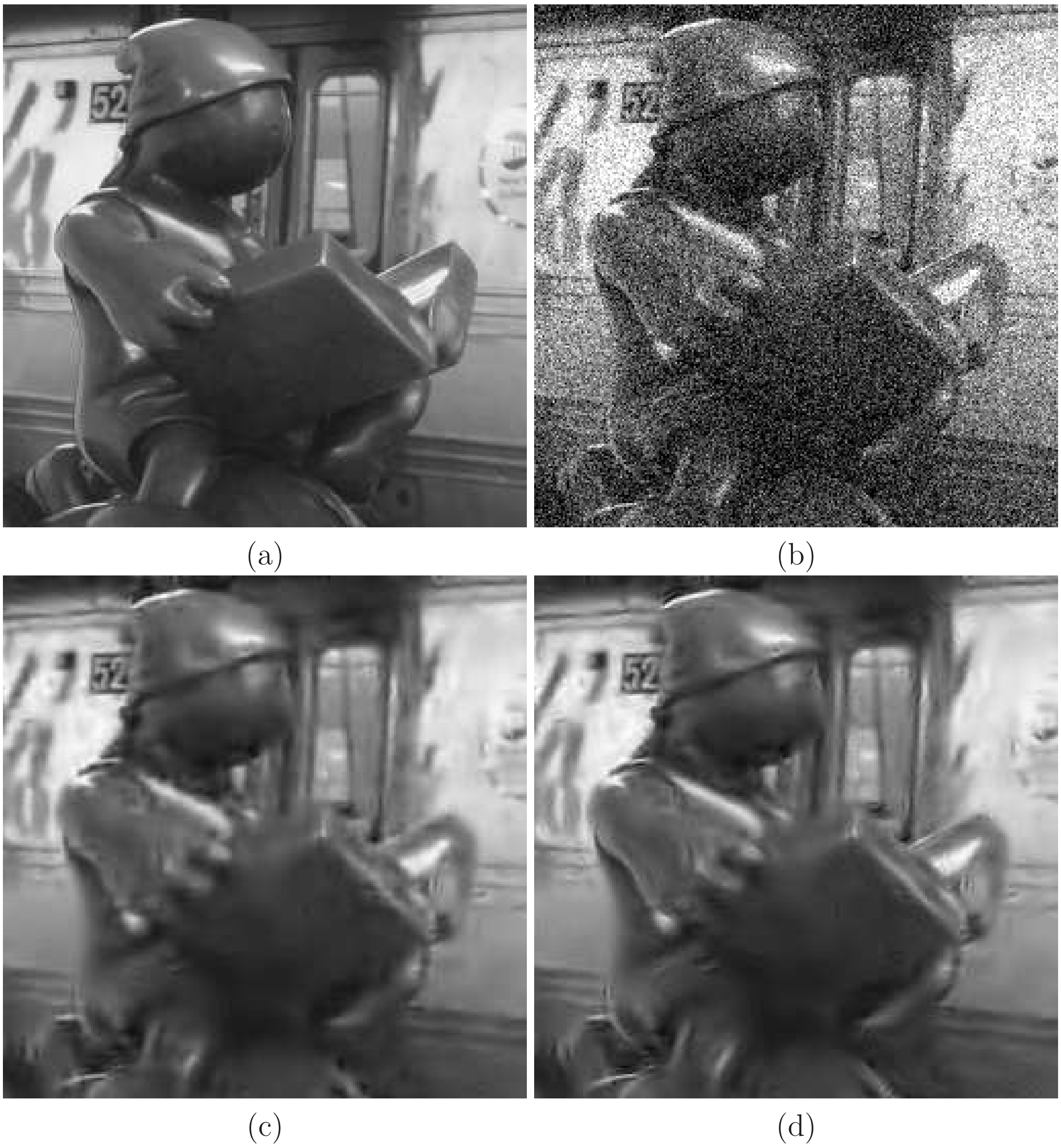

(a) (b) (c) (d)

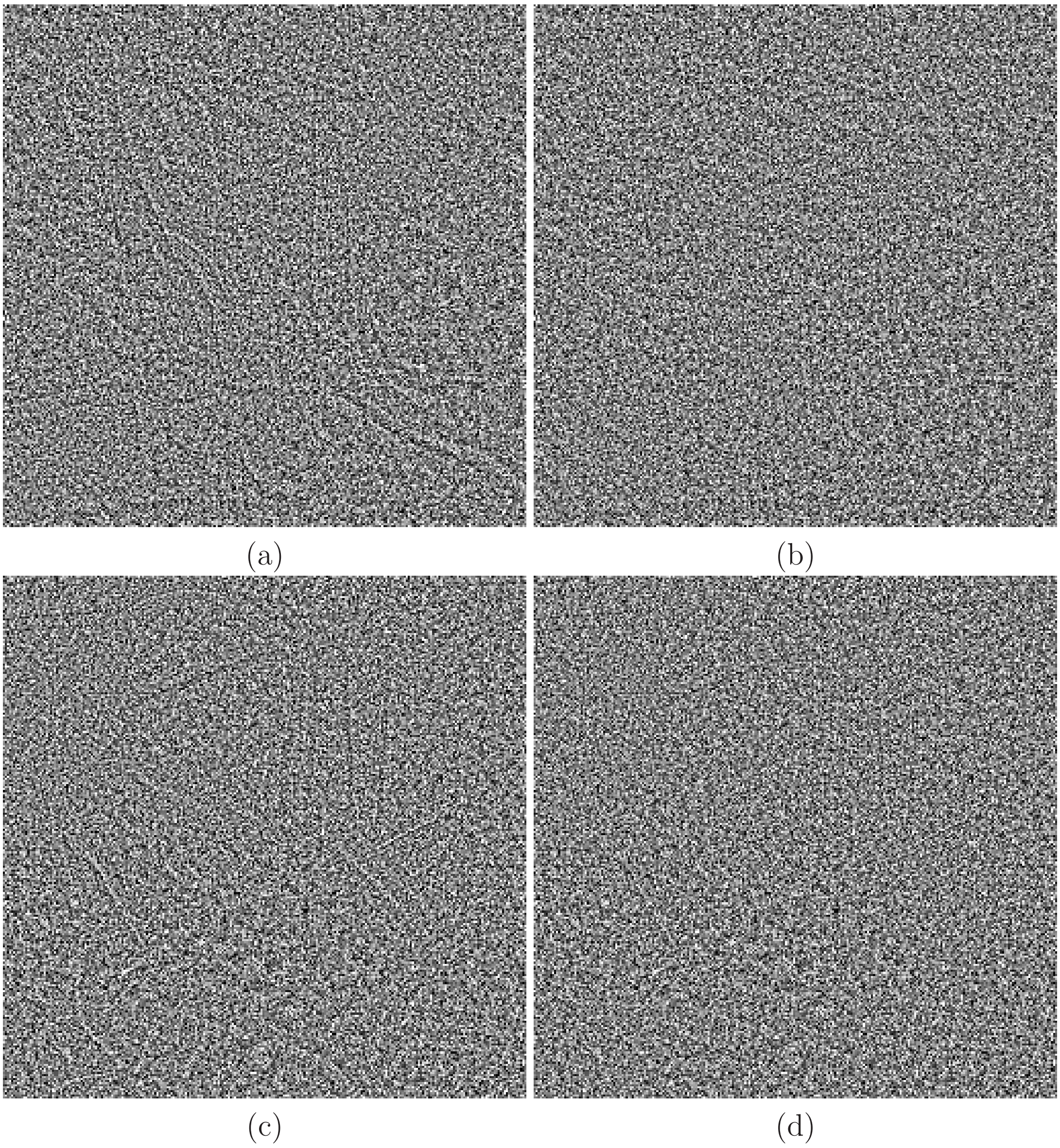
Figure 4.4: (a) Image #9 original (b) Noisy $\sigma = 40$ (16.0771) (c) GSM (28.633) (d) OAGSM/NC (28.918)

performance loss at higher noise levels is due to poor estimation of the oriented covariance. As the winnowing procedure greatly reduces the number of patches that are used to compute the covariances, the resulting estimates are less robust to noise. Thus the failure of winnowing under high noise conditions is not unexpected.

For many images, the OAGSM model performs better than the GSM model. This is highly dependent on the image content, however. Images that are strongly dominated by oriented features will show better performance for the OAGSM than for the GSM. However images that are dominated by non-oriented texture regions may yield better performance for the GSM method. In contrast, as the OAGSM/NC is able to adapt to non-oriented regions, it very rarely performs worse than the GSM method. For texture dominated images, its behavior is in fact very close to that of the GSM. Details of three selected denoised images are shown in figures (4.2 - 4.4). These images are for the noise level with $\sigma = 40$, using two orientation bands. The first image is an example of a texture dominated image. As can be seen, the visual appearance and PSNR performance of the GSM and OAGSM/NC methods are very similar. For this image, the OAGSM method performs worse by about 0.1 dB.

The second two images shown, #4 and #9, have significant oriented content. For the picture #4 of the columbine flower, these oriented structures are due to the object boundary of the flower against the out of focus background. In the image #9 of the public art in the new york city subway, the oriented features are due both to object boundaries and the strongly oriented siding of the train. For both of these images, the OAGSM/NC algorithm gives significantly better performance both in visual appearance and PSNR. The OAGSM also outperforms the GSM for these two, but not by as much as the OAGSM/NC.

Another way of visualizing the differences between different denoising methods is by examining the estimated noise components. Given a noisy image y, and a denoised estimate $\hat{x}$, the estimated noise component image is given by



(a) (b) (c) (d)

Figure 4.5: Estimated noise components $\hat{n}$ corresponding to the denoising results shown in figures 4.3 and 4.4. (a) GSM for Image #4 , (b) OAGSM/NC for Image #4 , (c) GSM for Image #9 , (d) OAGSM/NC for Image #9

$\hat{n} = y - \hat{x}$. If the denoising method were perfect, then the estimated noise component should look exactly like a sample of Gaussian white noise. In practice, however, some residual image structure remains in the estimated noise component. Comparing these for different denoising methods can give some insight into what image structures are "left behind" in the estimated noise component. These estimated noise components are shown for the GSM and OAGSM/NC methods in figure 4.5, for two images that were corrupted with white noise with $\sigma = 40$. While the differences are subtle, some oriented structures can be perceived in the estimated noise component for the GSM method. For the estimated noise components from the OAGSM/NC method, however, these residual structures are less apparent. This is further evidence that the OAGSM/NC is doing a better job of correctly identifying oriented content as belonging to the desired signal.

For both the OAGSM and OAGSM/NC models, the oriented covariances calculated by patch rotation performed better than either the edge model or implicit 1-d covariance methods. It is interesting to note that this difference was much stronger for the OAGSM method than for the OAGSM/NC method. As the OAGSM/NC has the non-oriented component to "fall back on", it will suffer only limited loss in performance from using oriented covariances that fail to appropriately model much of the signal. The OAGSM method, on the other hand, uses the oriented signal process to describe all of the image signal. The edge model and implicit 1-d covariances are quite similar to each other, and are only effective for describing perfectly oriented signal regions. As such perfectly oriented regions are not very common in actual images, the OAGSM method based on either of these oriented covariances performs quite poorly. Such de-

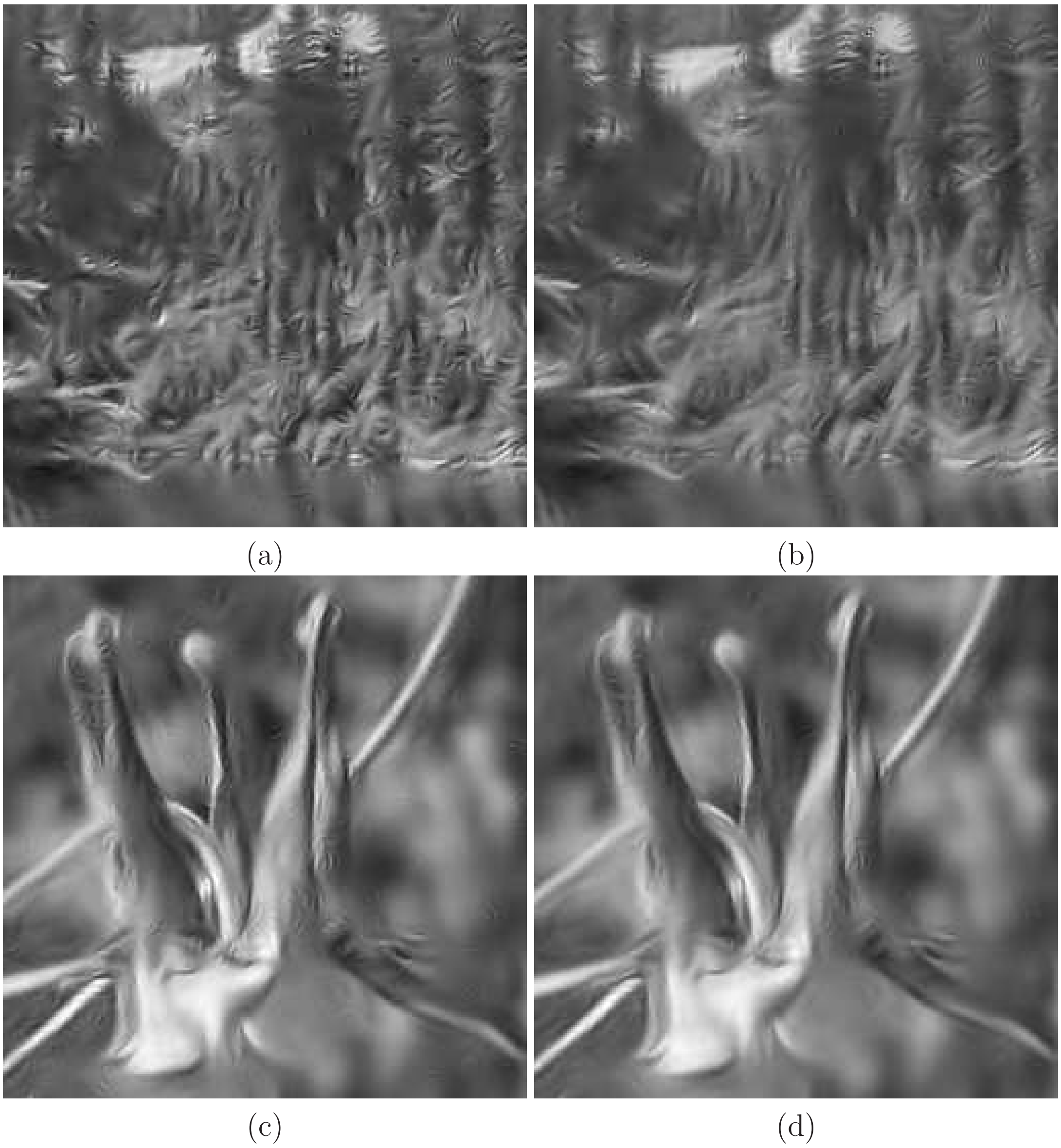


Figure 4.6: OAGSM results with patch rotated covariances (a),(c), and edge model covariances (b),(d).

noised images have interesting visual qualities, as can be seen in figure 4.6. The OAGSM can introduce inappropriate oriented artifacts into textured regions, as can clearly by seen in figure 4.6 (a). Using the edge model covariances produces denoised images that are overly smoothed, but still present clean edges.

The empirical 1-d covariances produce results that are similar but consistently worse than the edge model covariances. This may have to do with residual error involved in discretely sampling and interpolating the underlying correlation functions. There may still be room for improvement for estimating the 1-d empirical covariances.

Unlike the oriented covariance calculated by patch rotation, the edge model covariances are not computed from noisy data. Under very high noise conditions, the OAGSM/NC model using edge model covariances may perform better than using covariances from patch rotation. This is due to the presence of errors in the covariances estimated by patch rotation, which do not occur for the edge model covariances.

4.2.3 Spatially varying interpolation

The OAGSM/NC model incorporates aspects of both the GSM model and the OAGSM model. The OAGSM/NC model probability distribution truly is interpolated between the OAGSM and GSM densities, as pointed out by equation 3.48. The interpolation is controlled by β, the prior probability for the oriented component. As this parameter β is estimated for each subband of the image being denoised, it mediates adaptation of the OAGSM/NC to the statistics of the current signal.

Most images contain both oriented and non-oriented signal content. For such images, the OAGSM and GSM estimators perform better in different, complementary image regions. The OAGSM/NC method is able to obtain better performance than either method by adaptively interpolating between these two estimators at each location at space and scale. This behavior is a general property of BLS estimation performed using a model that is a mixture of component densities. To see this, let the "total" density $p_t(x) = \beta p_1(x) + (1-\beta)p_2(x)$ be such a mixture. This corresponds to the OAGSM/NC where $p_1 = p_{ori}$ and $p_2 = p_{nor}$ as defined in section 3.3.2. Assume y has been corrupted with an arbitrary noise process characterized by $p_n(y|x)$. For this calculation, this need not be additive Gaussian noise. The total BLS estimate is then

$$\hat{x}_t(y) = \int x p_t(x|y) dx = \int x \left(\frac{p_n(y|x)p_t(x)}{p_t(y)} \right) dx \tag{4.27}$$

where $p_t(y) = \int p_n(y|x)p_t(x)dx$. Similarly define $p_1(y) = \int p_n(t|x)p_t(x)dx$ and $p_2(y) = \int p_n(t|x)p_2(x)dx$. Expanding out $p_t(x)$ in terms of its components gives

$$\begin{aligned} \hat{x}_t(y) &= \frac{1}{p_t(y)} \int x p_n(y|x) \left(\beta p_1(x) + (1-\beta)p_2(x) \right) dx \\ &= \frac{\beta}{p_t(y)} \int x p_n(y|x)p_1(x)dx + \frac{1-\beta}{p_t(y)} \int x p_n(y|x)p_2(x)dx \\ &= \frac{\beta p_1(y)}{p_t(y)} \int x \left(\frac{p_n(y|x)p_1(x)}{p_1(y)} \right) dx + \frac{(1-\beta)p_2(y)}{p_t(y)} \int x \left(\frac{p_n(y|x)p_2(x)}{p_2(y)} \right) dx \end{aligned} \tag{4.28}$$

The two integrals in the above expression are the properly normalized BLS signal estimates assuming a signal density of $p_1(x)$ or $p_2(x)$, respectively. The total BLS estimate is thus given by

$$\hat{x}_t(y) = h(y)\hat{x}_1(y) + (1 - h(y))\hat{x}_2(y) \tag{4.29}$$

where $h(y) = \beta p_1(y)/p_t(y)$ functions as a spatially varying interpolation constant for combining the two estimators

4.3 Hybridization of distinct denoising methods via Supervised Learning

The above discussion shows how using BLS estimation with a two component mixture model such as the OAGSM/NC functions by combining two distinct denoising methods via a spatially varying function. The function h used above was derived in a consistent probabilistic framework, as the two component mixture model is a proper probability distribution. While this consistent framework is appealing for denoising applications it is not strictly necessary. In this section I describe an alternative approach for the general problem of combining two distinct "base" denoising methods into a single hybrid method. In this section, no assumptions will be made on the two base denoising functions other than that they operate on some local neighborhood of image coefficient data. In particular, I do not assume that the base denoising methods have the form of BLS estimators, or that they are based on probabilistic signal models.

I introduce a locally adaptive decision function that determines how the two base denoising estimates are to be combined at each location. This decision function is then learned from example data, where I assume access to an "example" clean image whose statistical and structural properties are similar to the image to be denoised. As the statistics of the noise process are assumed to be known, this clean example image can corrupted with a synthetic noise sample. This corrupted example image may be denoised with each of the two underlying base denoising methods. As the clean image is available, it is possible to see which of the two methods performed better in each location. From this data, one may compute the weights of the optimal linear combination at each signal location. These optimal weights, together with some set of features characterizing the signal at each location, form a training data set for learning the decision function.

If the decision function h is constrained to output only binary values, the resulting hybrid denoiser makes a hard decision at each spatial location to chose one denoising method or the other. In this case, learning h is a classification problem. If h is allowed to take continuous values, then the hybrid denoiser can smoothly interpolate between the outputs of the base denoiser at each location. h then maps from a vector space of local image features into the real numbers, and learning h is a regression problem. In this section I describe only the latter case, using a method known as weighted Kernel Ridge Regression. After describing the general theory, I apply the method to calculate the "machine learning" hybrid denoiser using the GSM and OAGSM as the base denoising methods. This provides an alternative way of achieving the spatial adaptation shown by the OAGSM/NC method. The machine learning hybrid shows similar, but consistently better, performance than the OAGSM/NC method.

Machine learning techniques have been applied to image denoising before. Several authors have used support vector regression techniques to directly estimate clean coefficients from noisy coefficients [11, 56]. More closely related to the work in this thesis, Lin and Yu used an SVM classifier to adaptively switch between applying a median filter and the identity filter for removing impulse noise from images [27].

4.3.1 Local Denoising Functions

Considering the noisy signal as a vector $y \in \mathbb{R}^N$, any denoising algorithm may be viewed as a function $f : \mathbb{R}^N \to \mathbb{R}^N$ where $\hat{x} = f(y)$ is the estimate of the clean signal. This space $\mathbb{R}^N$ may represent the original image pixels, or may refer to the representation of the signal in some other domain, such as the space of wavelet coefficients. As before, let a generalized wavelet neighborhood refer to a set of coefficients that are close to each other in space, scale and orientation. Given a generalized wavelet neighborhood, define a local denoising function to be a function $g : \mathbb{R}^d \to \mathbb{R}^n$ taking its input a patch of d wavelet coefficients and returning an estimate of a group of $n < d$ coefficients, typically at the center of the patch. Both the OAGSM and GSM methods are local denoising functions according to this definition. Applying this procedure to overlapping patches and estimating the center coefficients yields a complete estimator for all of the wavelet coefficients, which may be inverted to give the denoised image.

4.3.2 Hybrid Denoiser Form

Given a set of two local denoising functions g_1, g_2 with the same input and output dimensionality, one seeks to combine them into a single hybrid denoising function g_h. Introducing the decision function h, we write the hybrid estimate for a noisy patch $y \in \mathbb{R}^d$ as

$$g_h(y) = h(y)g_1(y) + (1 - h(y))g_2(y) \tag{4.30}$$

The decision function h should determine for each patch which of the two base denoising methods is more reliable. As h is a function of the patch itself, it is spatially adaptive. If the initial base denoisers have been optimized for distinct local signal content, one may view the output of h as classifying each patch into the natural domain for either g_1 or g_2. Allowing h to take arbitrary real values avoids a hard decision for each patch and permits the hybrid denoiser g_h to interpolate smoothly between the outputs of the base denoising functions.

4.3.3 Generation of Training Data

I wish to learn the function h that will yield good performance for the resulting hybrid denoiser. Let $y \in \mathbb{R}^d$ and $x^c \in \mathbb{R}^n$ denote a noisy wavelet patch and corresponding clean center coefficients. Assume that these are drawn from some fixed unknown distribution $\mathbf{D}(y, x^c)$ that is determined by the statistics of the signal and noise processes. I measure the performance of h by the expected squared error for the corresponding hybrid denoiser g_h, given by

$$E_{(y,x^c)}\left[||h(y)g_1(y) + (1 - h(y))g_2(y) - x^c||^2\right] \tag{4.31}$$

In practice one must learn h from a finite set of training examples $\{(y_i, x_i^c)\}_{i=1}^m$. Let h_i represent the value of the decision function for the i$^{\text{th}}$ data point in the training set. The error incurred on the i$^{\text{th}}$ training sample is

$$E(h_i, i) = ||x_i^c - (h_i g_1(y_i) + (1 - h_i)g_2(y_i))||^2 \tag{4.32}$$

which is a quadratic polynomial in h_i. We want to learn h that will lead to low values of this error. Accordingly, define the target value h_i^* to be the minimizer of $E(h_i, i)$. This yields

$$h_i^* = \frac{-(g_1(y_i) - g_2(y_i)) \cdot (g_2(y_i) - x_i^c)}{||g_1(y_i) - g_2(y_i)||^2} \tag{4.33}$$

The pairs $\{(y_i, h_i^*)\}_{i=1}^m$ then form the training data set for learning the decision function h.

One important issue for learning h is that the same amount of error in h for different patches will contribute differently to the error for the hybrid denoiser. For patches where the output of the two base denoisers g_1 and g_2 are either very similar or close to zero, large changes in h will yield only small changes in the output of g_h. Conversely, for image regions where the outputs of the base denoisers are substantially different, small changes in h lead to large changes in g_h and in these regions it is more important for h to be correct.

Appropriate weightings for the training examples can be found by expanding the error of the hybrid denoiser g_h on the training set, the so-called empirical loss, in terms of the target values h_i^*. The empirical loss is

$$\hat{E}_h = \sum_{i=1}^{m} E(h(y_i), i) \tag{4.34}$$

Expanding $E(h_i, i)$ about its minimum gives

$$E(h(y_i), i) - E(h_i^*, i) = ||g_1(y_i) - g_2(y_i)||^2 \, (h(y_i) - h_i^*)^2 \tag{4.35}$$

Summing over i and setting $\rho_i = ||g_1(y_i) - g_2(y_i)||^2$ yields

$$\hat{E}_h = \sum_{i=1}^{m} \rho_i (h(y_i) - h_i^*)^2 + C \tag{4.36}$$

where the constant $C = \sum E(h_i^*, i)$ does not depend on h. The ρ_i define the weights for each training data instance. This expression gives the empirical loss as a weighted sum of the squares of deviations from the target values for h. Intuitively speaking, these weights p_i indicate the relative amount of attention that should be paid to getting the correct value of h for each training data point.

4.3.4 Weighted Kernel Ridge Regression

In the expression above, the empirical loss is written as a weighted sum, where the weights are easily calculated from the training data and the base denoisers g_1 and g_2. This problem differs from standard unweighted regression in that errors on different training data points do not contribute the same amount to the empirical loss. Incorporating these weights into the data-fidelity term for the Kernel Ridge Regression algorithm gives a learning method that respects the relative importance of the different training data points. Standard Kernel Ridge Regression is described in detail in [15], and the weighted version has been used in [48].

Weighted Ridge Regression without the use of Kernels is equivalent to performing linear weighted least squares with a quadratic regularization term. Assuming a linear form for the decision function $h(x) = w^T \cdot x$, this algorithm works by choosing w to minimize the so-called weighted Ridge loss

$$L(w) = \sum_{i=1}^{m} \rho_i (w \cdot y_i - h_i^*)^2 + \alpha \, ||w||^2 \tag{4.37}$$

where α is a learning parameter controlling the regularization.

This optimization problem is soluble in closed form. Introducing the data matrix $\mathbf{Y}$, the vector of target values $\mathbf{H}$, and the diagonal matrix P with $P_{ii} = \rho_i$, we can write

$$L(w) = \alpha w^T w + (\mathbf{H} - \mathbf{Y}w)^T P(\mathbf{H} - \mathbf{Y}w) \tag{4.38}$$

Setting the gradient of L to zero yields the linear weighted Ridge Regression solution for the decision function

$$h(x) = w^T \cdot x = \mathbf{H^T} P\mathbf{Y} \left(\alpha I_d + \mathbf{Y}^T P\mathbf{Y}\right)^{-1} x \tag{4.39}$$

where I_d is an identity matrix of dimension d.

Like many algorithms in machine learning, Ridge regression may be "Kernelized" by examining the form of the solution of the linear version and noting that the training data appear only through their dot products. Replacing these dot products with a Kernel function $K(y_1, y_2)$ yields a nonlinear version of the algorithm that implicitly maps the input data into a higher, possibly infinite dimensional, space before performing weighted Ridge Regression. Applying the matrix identity $(I + AB)^{-1}A = A(I + BA)^{-1}$, one may rewrite

$$h(x) = \mathbf{H}^T(\alpha I_m + P\mathbf{Y}\mathbf{Y}^T)^{-1} P\mathbf{Y}x \tag{4.40}$$

As the i, j entry of $\mathbf{Y}\mathbf{Y}^T$ is $y_i \cdot y_j$, it is replaced by $\mathbf{K}$ where $\mathbf{K}_{i,j} = K(y_i, y_j)$. Similarly, replace $\mathbf{Y}x$ by the mx1 vector $\mathbf{k}(x)$ that has i$^{\text{th}}$ entry $K(y_i, x)$. With this notation, the weighted Kernel Ridge Regression solution is given by

$$h(x) = \mathbf{H}^T(\alpha I_m + P\mathbf{K})^{-1}P\mathbf{k}(\mathbf{x}) \tag{4.41}$$

4.3.5 Results

The hybrid denoising procedure was applied to a collection of ten 256x256 pixel test images that were corrupted with synthetically generated Gaussian white noise. Original image pixel values ranged between 0 and 255. The same three noise levels as in section 4.2.2 were used, with noise deviation $\sigma = 20$, 40 and 80. For these numerical experiments, training and test image pairs were generated by extracting two adjacent non-overlapping 256x256 pixel subregions from the same 512x512 pixel test images that were used before in section 4.2.2.

The noisy training and test images, as well as the clean training image were decomposed using the Steerable Pyramid representation with 3 scales and 2 orientation bands. 5x5 patches including one pair of "parent" coefficients at the coarser scale are used, so each patch may be viewed as a vector in $\mathbb{R}^{52}$. The noisy training image was denoised with both the OAGSM and GSM denoising methods, and at each location in space and scale the decision function target value h_i^* was computed, as described in section 4.3.3. OAGSM covariances were formed using winnowing, with threshold d_{ori}^* set to the 85^{th} percentile value at each scale. Distinct decision functions h were learned for each image and at each scale. To form training features, the noisy patches were extracted at each of the 3 scales. These noisy patches were then rotated by their dominant orientation and the 52 coefficients of these rotated noisy patches were taken as the feature vectors for the learning problems. At each image scale, the rotated patch features were scaled by a divisive constant to lie in the range [-1,1]. This gave 65536 training examples for at the first scale, 16384 training examples at the second scale and 4096 training examples at the third scale. Due to high computational cost, the training examples at the first and second scale were pruned to the 5000 with largest weights.

Gaussian kernels of the form $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}$ were used for the weighted Kernel Ridge Regression. Different values of the learning parameters α and γ were used for different image scales and noise levels, however the same parameters were used across the different images. The learning parameters were selected by four-fold cross-validation on a single training image, using the 3000 training examples with greatest weights at each scale. This was done by dividing the 3000 training examples randomly into four pieces. Using particular values for the learning parameters γ and α, the weighted Kernel Ridge Regression classifier was trained using 3/4 of the training data points. The error was then measured from using this classifier to predict the portion not used for training. Averaging this over all four portions of the training data gave the so-called cross validation error for the values of γ and α used. Cross-validation was done for each point of a logarithmically spaced grid with $\alpha = [2^1, 2^2, ..., 2^{10}]$ and $\gamma = [2^{-5}, 2^{-4}, ..., 2^5]$ and the parameters yielding the lowest cross-validation error were selected.

To obtain the final hybrid denoising results, the OAGSM and GSM denoising estimates were computed for the noisy test images. Each test image patch was then rotated by its dominant orientation and rescaled according to the divisive constants calculated during training. The learned decision function for the

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	26.271	27.741	36.396	31.083	31.710	26.589	31.983	33.079	32.224	27.824
O/rot/w	-0.294	0.005	0.033	-0.021	0.194	-0.188	0.626	0.048	0.217	-0.247
Hybrid	0.126	0.474	0.283	0.181	0.618	0.161	0.826	0.245	0.620	0.115
NC/rot/w	0.009	0.234	0.184	0.123	0.362	0.035	0.617	0.161	0.314	0.019

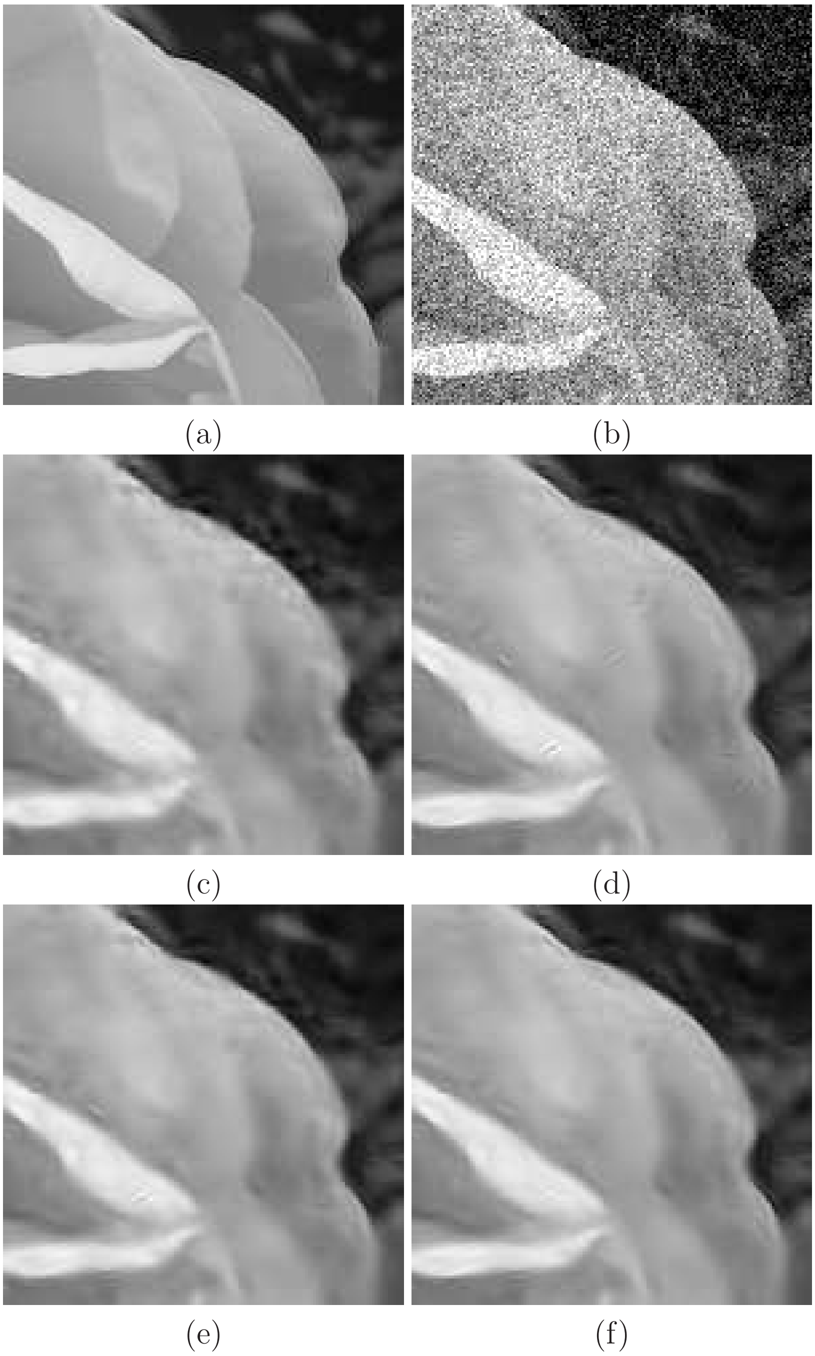
	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	23.194	24.175	33.567	27.848	28.437	23.832	28.525	29.991	28.670	24.787
O/rot/w	-0.185	0.007	-0.275	0.143	0.011	-0.136	0.331	-0.073	0.093	-0.212
Hybrid	0.068	0.356	0.260	0.256	0.451	0.076	0.713	0.186	0.434	0.084
NC/rot/w	0.007	0.173	0.187	0.144	0.251	0.022	0.387	0.119	0.195	0.014

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**	**Im 6**	**Im 7**	**Im 8**	**Im 9**	**Im 10**
GSM	21.107	21.348	29.991	25.219	25.494	21.934	25.504	27.116	25.527	22.434
O/rot/w	-0.080	-0.115	-0.662	0.045	-0.127	-0.153	0.008	-0.330	-0.272	-0.245
Hybrid	0.057	0.175	0.161	0.316	0.275	0.008	0.404	0.086	0.144	0.020
NC/rot/w	0.012	0.074	0.075	0.108	0.165	-0.016	0.201	0.010	0.027	-0.021

Table 4.4: Hybrid denoising results. GSM results given in PSNR, other methods are relative to GSM baseline. Results presented for noise levels $\sigma = 20$ (PSNR 22.0836 dB, top), $\sigma = 40$ (PSNR 16.063 dB, middle), $\sigma = 80$ (PSNR 10.0424, bottom). OAGSM/NC results presented for comparison.

appropriate scale and noise level was then evaluated on these rotated patches, and used to combine the OAGSM and GSM estimates to give a hybrid estimator for each of the 3 scales. As both methods used the GSM estimator for the highpass residual band, no adaptive combination was necessary for the highpass band. Inverting the pyramid transform then gave the resulting denoised images. For comparison, each of these images were also denoised with the OAGSM/NC method.

Denoising results reported by PSNR are given in table 4.4. At all three noise levels, the hybrid method shows significant improvement over both the GSM and OAGSM. The hybrid method also outperforms the OAGSM/NC method by up to 0.3 dB for some images. It should be noted that this is a somewhat unfair comparison as the hybrid method had access to the clean training data image, while the OAGSM/NC was computed entirely from the noisy test image. Visual appearance of the hybrid method and the OAGSM/NC are actually quite similar. Both methods showing clear improvement over the baseline GSM result, as can be seen in figure 4.7.



(a) (b)

(c) (d)

(e) (f)

Figure 4.7: 150x150 detail from hybrid denoising results for image #7. (a) Original (b) Noisy $\sigma = 40$ (16.063) (c) GSM (28.525) (d) OAGSM (28.856) (e) OAGSM/NC (28.912) (f) Hybrid method (29.238)

Chapter 5

Conclusions and Future Directions

Strongly oriented localized features are one of the most distinguishing characteristics of natural photographic images. Accordingly, describing the orientation at a particular location provides a significant amount of information about the local signal content. In this thesis I have pushed this idea in two different, but complementary directions.

On the one hand, I have shown that measuring and imposing a full set of local multiscale orientation measurements completely constrains the resulting image. The local multiscale orientations are measured by first building the two-band Steerable Pyramid transformation of the image. As the filters of this transformation are first order derivative operators, the transform coefficients provide a measurement of the image gradient at multiple scales. Transforming these gradient vectors into polar coordinates defines the magnitude and orientation bands. I have shown that by starting with a random signal and repeatedly imposing the measured orientation bands, as well as the residual highpass and lowpass information, it is possible to reconstruct the input image exactly. This yields a novel deterministic representation of images based on purely geometric quantities. Reconstruction from these orientation measurements can be achieved by starting with a random signal, building the SP transform, imposing the orientations, inverting the transform and repeating. I have shown that this simple algorithm operates by projection onto convex sets, and is thus guaranteed to converge. Some methods for accelerating the convergence of this algorithm were developed.

Although projection onto convex sets proves that the reconstruction algorithm converges, it does not imply that the convergence will be unique. Analyzing the dynamics of the reconstruction algorithm provides a condition for uniqueness of the representation. By translating a fixed point of the algorithm to the origin and analyzing the "homogenized" dynamics, it was shown that the representation will be unique if a certain operator has eigenvalues strictly less than one. This condition may be verified numerically for specific images, thus showing uniqueness of their orientation representations. This is an incomplete result, however, as it does not provide a simple characterization of images for which the orientation representation is unique.

It is simple to come up with certain images for which uniqueness will fail. In particular, images which have been bandpass filtered such that their Steerable Pyramid transforms have zero lowpass bands will fail to be uniquely determined by the orientation measurements. This is not unexpected, however, as the orientation measurements are invariant under multiplication by a scalar constant. This free constant was only pinned down by imposing the lowpass band.

However, if the lowpass band is identically zero as for such bandpass images, imposing it will fail to fix the overall free constant. For such images, however, it was observed that running the reconstruction algorithm would converge to a fixed point that was an exact scalar multiple of the original image. The precise value of the multiple was dependent on the initial starting point. It is unclear if this counterexample is the only type of image for which uniqueness fails. This is an interesting question for further study.

This orientation representation has not yet been used for any image processing applications. It is an open question if useful image manipulations can be made by performing some processing in the "orientation domain" and then applying the reconstruction algorithm. One potential issue for this type of method is that the reconstruction algorithm relied heavily upon the fact that the orientations being imposed were consistent with some image in the image space. If the orientations being imposed were obtained by some other manipulation, this may no longer be the case. For such a situation, it is likely that some form of "soft" imposition of the orientation data would be necessary, similar to that used for the reconstruction from quantized orientations.

The second portion of the thesis focused on describing how knowledge of the local neighborhood orientation can enable local adaptation of a stochastic image model. Natural images are highly inhomogeneous, often showing quite different local signal properties in different image regions. The models developed in this thesis are based on the idea that much of the inhomogeneity present in images may be explained by a few spatially varying hidden variables that parameterize the local signal statistics. Natural images typically show significant spatial variation in both local signal power and in local orientation. Constructing a probability model for the local signal content conditioned on the local power and local orientation enables the development of an adaptive stochastic model. This type of conditional model is an answer to the question, what do I know about the signal given that I know the local neighborhood orientation and local power? The previous orientation representation work shows that if all of the local orientations at every scale are fixed, then the entire image signal is known exactly. However, if only the current dominant neighborhood orientation is specified, then the image signal is not completely constrained.

All of the stochastic models in this work are models for small patches of image wavelet coefficients. By setting the signal model conditioned on the hidden variables to be Gaussian, the complete density becomes a Gaussian mixture. In the first model developed in this thesis, each signal patch is described as a sample from a single, uniform multivariate Gaussian process that is scaled and rotated by the hidden variables controlling local power and orientation. While this OAGSM model provides a good description of oriented signal regions, many images contain significant non-oriented regions that are not well described by the OAGSM. Introducing a third hidden variable that models the "orientedness" of each patch leads to the OAGSM with non-oriented component model. Both of these are Gaussian mixture models, where the covariances of the components are parameterized by hidden variables that describe the local signal properties.

These models are used for denoising images corrupted by additive Gaussian noise, by using them as signal priors for a Bayes least squares estimator. The performance of these methods was compared to denoising based on the Gaussian scale mixture model, which is similar in form but without adaptation to local orientation. For images with significant oriented content, the OAGSM method performs better. However for images dominated by non-oriented textures, the OAGSM method introduces inappropriate oriented artifacts, and can perform worse than the GSM method in some cases. These issues are resolved by the OAGSM/NC, which is able to adapt between oriented and non-oriented local signal, and accordingly shows consistent improvement over the GSM for all of the test images used.

This adaptive selection between two different local denoising methods was studied from another angle, using the formalism of machine learning. The spatial adaptation can be placed into a locally varying decision function that interpolates two specified local denoising methods. Setting up the problem this way, one seeks to learn a decision function that yields low total denoising error. Given access to a clean training image, by corrupting it with noise and denoising with the two given methods, one can learn where the strengths and weaknesses of each method are. This was set up as a supervised learning problem, and solved using the weighted kernel ridge regression algorithm. When applied using the OAGSM and GSM denoising methods, the resulting hybrid denoiser outperformed both of these two base methods, as well as beating the the OAGSM/NC.

The methods developed in this thesis provide a set of interesting and novel tools for using local orientation to model images. There are many opportunities for expanding upon the current work. It is straightforward to sample an isolated image patch from the OAGSM model. This suggests it may be useful for synthesizing image data in areas where one can make a good prediction of the local orientation. As orientations may be predicted across image scales, the OAGSM model may be useful for image super-resolution problems. For this problem, one seeks to estimate a high resolution version of a given image. Super-resolution is equivalent estimating one or more levels of “missing” subbands of wavelet coefficients at the finer spatial scales. It may be possible to first predict the orientation of the fine scale information from the known coarse scales, then sample the fine scale coefficients using the OAGSM model. One issue that will necessarily arise when using sampling from the OAGSM model to estimate an entire image subband is how to address the overlap of neighboring patches. One possible method for addressing this, without the significant complexity of building a entire global probability model, may be to fill in the subband coefficient by coefficient, ordered according to some measure of confidence in the ability to estimate a reasonable value. One such scheme may be to sample each coefficient using a generalized neighborhood that may include previously filled in coefficient values. Sampling may then be done using the OAGSM distribution conditioned on the previously filled in coefficient values.

For all variants of the OAGSM and OAGSM/NC models considered, the prior densities for the z and θ hidden variables were fixed over the entire image. Additionally, no spatial interactions between the hidden variables were introduced. One possible way of introducing some spatial communication between different patch locations would be through modulating the hidden variable prior densities based on nearby regions. For instance, the presence of a single strongly oriented region may indicate that coaligned patches are likely to have similar orientations. This could be gently encouraged through re-weighting of the priors over θ in the adjoining coaligned regions. A more radical modification would be to model the hidden variables z and θ as a random field, perhaps as a Markov random field. Using Markov Chain Monte Carlo techniques, samples may be drawn from the field. In this case, the denoising estimate at each location would be given by the average of Weiner estimates corresponding to each hidden variable, weighted by the relative frequency of their appearance as samples from the random field. This idea is related conceptually to recent work by Lyu and Simoncelli, who achieved very good denoising results using a two stage hierarchical Markov random field model where one of the stages acts as a hidden multiplier, similar in spirit to z in this thesis, modulating a homogeneous Gaussian Markov random field [29].

The OAGSM/NC model hidden variables included the variable δ, which modeled the orientation of the patch. In this model δ was constrained to be a binary variable. An interesting problem would be to estimate signal covariances that are able to smoothly interpolate between oriented and non-oriented. One possible approach to this is suggested by the orientedness measurement d_{ori}. One could estimate a signal covariance $C(d^*)$ adapted to a certain orientedness d^* by using only patches with satisfying $d_{ori} \in [d^* - \epsilon, d^* + \epsilon]$.

Appendix A

Cosine Tiling

I seek to calculate

$$\sum_{k=1}^{K} \cos\left(\theta - \frac{k\pi}{K}\right)^{2(K-1)} \tag{1}$$

Using $\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta})$, this is

$$\sum_{k=1}^{K} \left(\frac{1}{2}\left(e^{i\theta - \frac{k\pi}{K}} + e^{-i(\theta - \frac{n\pi}{K})}\right)\right)^{2K-2} \tag{2}$$

Applying the binomial theorem, this may be expanded as

$$\frac{1}{2^{2(K-1)}} \sum_{k=1}^{K} \sum_{m=0}^{2K-2} \binom{2K-2}{m} e^{i(\theta - \frac{k\pi}{K})m} e^{-i(\theta - \frac{k\pi}{K})(2K-2-m)} \tag{3}$$

Interchanging the order of sums and simplifying yields

$$\frac{1}{2^{2(K-1)}} \sum_{m=0}^{2K-2} \left[\binom{2K-2}{m} e^{i2(m+1-K)\theta} \sum_{k=1}^{K} e^{-ik(m+1)\frac{2\pi}{K}}\right] \tag{4}$$

Now note that the inner sum over k is a geometric series, namely

$$\sum_{k=1}^{K} e^{-ik(m+1)\frac{2\pi}{K}} = \sum_{k=1}^{K} z_m^k \tag{5}$$

for $z_m = e^{-i(m+1)\frac{2\pi}{K}}$. Using the sum formula for a geometric series $\sum_{k=1}^{K} z^k = z\left(\frac{z^K-1}{z-1}\right)$ gives

$$\sum_{k=1}^{K} e^{-ik(m+1)\frac{2\pi}{K}} = e^{-i(m+1)\frac{2\pi}{K}} \frac{(e^{-i(m+1)\frac{2\pi}{K}})^K - 1}{e^{-i(m+1)\frac{2\pi}{K}} - 1} = 0 \tag{6}$$

which is valid for $m \neq K-1$. For $m = K-1$, $e^{-ik(m+1)\frac{2\pi}{K}} = 1$ and the sum in (5) equals K. We thus have

$$\sum_{k=1}^{K} e^{-ik(m+1)\frac{2\pi}{K}} = K\delta_{m,K-1} \tag{7}$$

Substituting this into (4) gives

$$\begin{aligned}\sum_{k=1}^{K} \cos\left(\theta - \frac{k\pi}{K}\right)^{2(K-1)} &= \frac{1}{2^{2(K-1)}} \sum_{m=0}^{2K-2} \binom{2K-2}{m} e^{i2(m+1-K)\theta} K\delta_{m,K-1} \\ &= \frac{K}{2^{2(K-1)}} \binom{2K-2}{K-1} \end{aligned} \tag{8}$$

the desired identity.

Appendix B

Steering

This section explains how to calculate steering coefficients $c_k(\phi)$. These satisfy

$$\cos^{K-1}(\theta-\phi)=\sum_{k=1}^{K} c_k(\phi)\cos^{K-1}(\theta-\theta_k) \tag{9}$$

where $\theta_k=\frac{(k-1)\pi}{K}$. First note that we may decompose

$$\begin{aligned}\cos^{K-1}(\theta-x) &= (\cos(\theta)\cos(x)+\sin(\theta)\sin(x))^{K-1}\\ &= \sum_{j=0}^{K-1}\binom{K-1}{j}(\cos(\theta)\cos(x))^j(\sin(\theta)\sin(x))^{K-1-j}\\ &= \sum_{j=1}^{K} a_j(x) f_j(\theta) \qquad (10)\end{aligned}$$

with $a_j(x)=\binom{K-1}{j-1}\cos^{j-1}(x)\sin^{K-j}(x)$ and $f_j(\theta)=\cos^{j-1}(\theta)\sin^{K-j}(\theta)$. Any translate of $\cos^{K-1}(\theta)$ can thus be written as a linear combination of the K functions $f_j(\theta)$.

The $c_k(\phi)$ can be calculated by expressing all of the cosine powers in terms of this basis. Applying the decomposition (10) to both the left hand and right hand sides of (9) gives

$$\sum_{j=1}^{K} a_j(\phi) f_j(\theta) = \sum_{k=1}^{K} c_k(\phi) \sum_{j=1}^{K} a_j(\theta_k) f_j(\theta) \tag{11}$$

As the $f_j(\theta)$ are linearly independent, this implies that

$$\sum_{k=1}^{K} a_j(\theta_k) c_k(\phi) = a_j(\phi) \tag{12}$$

for $j = 1...K$. This can be interpreted as a linear matrix equation. Defining the matrix A and the vectors $c(\phi)$ and $a(\phi)$ by $(A)_{j,k} = a_j(\theta_k)$, $(c(\phi))_k = c_k(\phi)$ and $(a(\phi))_j = a_j(\phi)$, the solution of (12) may be written as

$$c(\phi) = A^{-1} a(\phi)$$

which gives the steering coefficients $c_k(\phi)$.

Appendix C

Tight Frame Operators

Here I define and prove a few relevant properties of tight frame linear operators.

Let X and Y be Hilbert spaces with associated inner products $<,>_X$ and $<,>_Y$ and resulting norms $||\cdot||_X$ and $||\cdot||_Y$. A linear map $A : X \to Y$ is said to be a frame if there exist positive constants C_1 and C_2 such that

$$C_1||Ax||_Y \leq ||x||_X \leq C_2||Ax||_Y \tag{13}$$

for all $x \in X$. The second of these inequalities implies that a frame is a bounded operator. The first implies that A cannot have a nontrivial null space. In particular, one must have $\dim(Y) \geq \dim(X)$. If the frame bounds C_1 and C_2 are equal, then A is called a tight frame, and then for all $x \in X$

$$||x||_X = C||Ax||_Y \tag{14}$$

The work in this thesis makes extensive use of the following two properties of tight frame operators.

Property 1 : $\frac{1}{C}A^\dagger : Y \to X$ is a left inverse for A, i.e. $A^\dagger A = CI_X$.

Proof : $A^\dagger A$ is self-adjoint and can thus has a full set of eigenvectors. It then suffices to show that all of the eigenvalues are equal to C. Letting v be any such eigenvector, with $A^\dagger Av = \lambda v$. Taking inner products with v gives

$$\lambda < v, v >_X =< A^\dagger Av, v >_X \tag{15}$$

$$=< Av, Av >_Y \tag{16}$$

$$= C < v, v >_X \tag{17}$$

using the properties of the adjoint and that A is a tight frame. This shows $\lambda = C$ for every eigenvector v. ■

Property 2 : Let $U = A(X)$ be the image of X under A. Then $AA^\dagger$ is an orthogonal projection from Y onto U.

Proof : Write $y = Ax + n$, where $x \in X$ (and so $Ax \in U$) and $n \in U^\perp$. $AA^\dagger$ will be an orthogonal projection onto U iff $AA^\dagger y = Ax$. One has $AA^\dagger y = A(A^\dagger A)x + AA^\dagger n = Ax + AA^\dagger n$, as $A^\dagger A = I_{Im}$. It remains to show $AA^\dagger n = 0$. Examine

$$\begin{aligned} \left|\left|AA^\dagger n\right|\right|^2 &= \left\langle AA^\dagger n, AA^\dagger n\right\rangle_Y \\ &= \left\langle n, (AA^\dagger)^\dagger AA^\dagger n\right\rangle_Y = \left\langle n, A(A^\dagger A)A^\dagger n\right\rangle_Y \\ &= C\left\langle n, AA^\dagger n\right\rangle_Y \end{aligned}$$

where I have used $AA^\dagger = CI_X$. As $A^\dagger n \in X$, $AA^\dagger n \in U$. But as $n \in U^\perp$, $\left\langle n, AA^\dagger n\right\rangle_Y = 0$. This shows that $AA^\dagger n = 0$. ■

Bibliography

[1] M Antonini, T Gaidon, P Mathieu, and M Barlaud. *Wavelets in Image Commmunication*, chapter 3, pages 65–188. Elsevier, 1994.

[2] Z Azimifar, P Fieguth, and E Jernigan. Towards random field modeling of wavelet statistics. In *Proceedings of the International Conference on Image Processing*, 2002.

[3] Heinze H Bauschke and Jonathan Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38:367–426, 1996.

[4] Jeff A Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models (TR-97-021). Technical report, U.C. Berkeley Department of Electrical Engineering and Computer Science, 1998.

[5] Keith A Birney and Thomas R Fischer. On the modeling of DCT and subband image data for compression. *IEEE Transactions on Image Processing*, 4:186–193, 1995.

[6] Robert W Buccigrossi and Eero P Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8:1688–1701, 1999.

[7] Peter J Burt and Edward H Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[8] Emmanuel Candes and David Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Communications on Pure and Applied Mathematics*, 57:219–266, 2003.

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

[10] Ward Cheney and Allen A Goldstein. Proximity maps for convex sets. *Proceedings of the American Mathematical Society*, 10:448–450, 1959.

[11] H Cheng, Q Yu, J Tian, and J Liu. Image denoising using wavelet and support vector regression. pages 43–46, 2004.

[12] A Cohen, I Daubechies, and J C Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–560, 1992.

[13] Albert Cohen, Ronald DeVore, Pencho Petrushev, and Hong Xu. Nonlinear approximation and the space $BV(R^2)$. *American Journal of Mathematics*, 121:587–628, 1999.

[14] R R Coifman and D L Donoho. Translation invariant denoising. In *Wavelets and Statistics : Springer Lecture Notes in Statistics 103*, pages 125–150, 1995.

[15] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[16] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[17] D. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics*, 27:859–897, 1999.

[18] P. L. Dragotti, M. Vetterli, and V. Velisavljevic. Directional wavelets and wavelet footprints for compression and denoising. In *Proc. Int'l Wkshp Adv. Methods for Multimedia Sig. Proc.*, Capri, Italy, September 2002.

[19] J. H. Elder. Are edges incomplete? *International Journal of Computer Vision*, 34(2/3):97–122, 1999.

[20] D J Field. Wavelets, vision and the statistics of natural scenes. *Royal Society of London Philosophical Transactions Series A*, 357:2527–2541, 1999.

[21] Vivek K Goyal, Martin Vetterli, and Nguyen T Thao. Quantized overcomplete expansions in R^N : Analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 1:16–31, 1998.

[22] A N Hirani and T Totska. Combining spatial and frequency domain information for fast interactive image noise removal. In *Proceedings SIGGRAPH 96*, pages 269–276, 1996.

[23] Jinggang Huang and David Mumford. Statistics of natural images and models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 1999.

[24] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 2001.

[25] R B Lehoucq, D C Sorensen, and C Yang. *ARPACK User's guide : Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM Publications, 1998.

[26] X. Li. On exploiting geometrical constrants of image wavelet coefficients. *IEEE Trans. Image Processing*, 12(11):1378–1387, Nov 2003.

[27] Tzu-Chao Lin and Pao-Ta Yu. Adaptive two-pass median filter based on support vector machines for image restoration. *Neural Comp.*, 16(2):333–354, 2004.

[28] Siwei Lyu and Hany Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1:111–119, 2006.

[29] Siwei Lyu and Eero Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. *Advances in Neural Information Processing Systems (NIPS) 2006*, 2006.

[30] Maurits Malfait and Dirk Roose. Wavelet-based image denoising using a Markov random field a priori model. *IEEE Transactions on Image Processing*, 6:549–565, 1997.

[31] S Mallat and S Zhong. Characterization of signals from multiscale edges. *IEEE Trans. PAMI*, 14(7):710–732, July 1992.

[32] Stéphane Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Pattern Analysis and Machine Intelligence*, 11(7), 1989.

[33] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley Interscience, 1997.

[34] Yves Meyer. *Wavelets and Operators*. Cambridge University Press, 1993.

[35] Yves Meyer. *Oscillating patterns in image processing and nonlinear evolution equations : the fifteenth Dean Jacqueline B. Lewis memorial lectures*. American Mathematical Society, 2001.

[36] Pierre Moulin and Juan Liu. Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors. *IEEE Transactions on Information Theory*, 45:909–919, 1999.

[37] D Mumford and J Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications of Pure and Applied Mathematics*, 42:577–685, 1989.

[38] A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69:529–541, 1981.

[39] William B Pennebaker and Joan L Mitchell. *JPEG still image data compression standard*. Van Nostrand Reinhold, New York, 1993.

[40] E. Le Pennec and Stéphane Mallat. Sparse geometric image representation with bandelets. *IEEE Trans. Image Processing*, 2005. To appear.

[41] Javier Portilla. Full blind denoising through noise covariance estimation using Gaussian scale mixtures in the wavelet domain. In *Proceedings of the IEEE International Conference on Image Processing*, 2004.

[42] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12:1338–1351, 2003.

[43] Tamer Rabie. Robust estimation approach for blind denoising. *IEEE Transactions on Image Processing*, 14:1755–1765, 2005.

[44] Randall C Reininger and Jerry G Gibson. Distributions of the two-dimensional dct coefficients for images. *IEEE Transactions on Communications*, COM-31:835–839, 1983.

[45] J. Romberg, M. Wakin, and R. Baraniuk. Multiscale Geometric Image Processing. In *SPIE Visual Communications and Image Processing*, Lugano, Switzerland, July 2003.

[46] Daniel L Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–817, 1994.

[47] L Rudin, S Osher, and E Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 30:259–268, 1992.

[48] G Saon. A nonlinear speaker adaptation technique using kernel ridge regression. In *IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[49] Shayle R Searle. *Matrix Algebra Useful for Statistics*. John Wiley and Sons, 1982.

[50] Ladan Shams and Christoph von der Malsburg. The role of complex cells in object recognition. *Vision Research*, 42:2547–2554, 2002.

[51] J M Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions in Signal Processing*, 41:3445–3463, 1993.

[52] E P Simoncelli. Statistical modeling of photographic images. In Alan Bovik, editor, *Handbook of Image and Video Processing*, chapter 4.7, pages 431–441. Academic Press, 2005. 2nd edition.

[53] 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*, volume III, pages 444–447, Washington, DC, October 1995.

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

[55] Eero P Simoncelli and Edward H Adelson. Noise removal via Bayesian wavelet coring. In *Proceedings of the 3rd IEEE International Conference on Image Processing*, 1996.

[56] B Sun, D Huang, and H Fang. Lidar signal denoising using least-squares support vector machine. *IEEE Signal Processing Letters*, 12:101–104, 2005.

[57] D. Taubman and A. Zakhor. Orientation adaptive suband coding of images. *IEEE Trans. Image Processing*, 3:404–420, July 1994.

[58] David Taubman and Michael Marcellin. *JPEG2000 : Image compression fundamentals, standards and practice*. Kluwer Academic Publishers, 2002.

[59] Nguyen T Thao and Martin Vetterli. Deterministic analysis of oversampled a/d conversion and decoding improvement based on consistent estimates. *IEEE Transactions on Signal Processing*, 42:519–531, 1994.

[60] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14:391–412, 2003.

[61] Chengjie Tu and Trac D Tran. Context-based entropy coding of block transform coefficients for image compression. *IEEE Transactions on Image Processing*, 11:1271–1283, 2002.

[62] A van der Schaaf and J H van Hateren. Modelling the power spectra of natural images: Statistics and information. *Vision Research*, 36:2759–2770, 1996.

[63] Martin J Wainwright and Eero P Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems (NIPS) 12*, 2000.

[64] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. Image quality assesment : From error visibility to structural similarity. *IEEE transactions on Image Processing*, 13:600–612, 2004.

[65] Marcelo J Weinberger, Gadiel Seroussi, and Guillermo Sapiro. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Transactions on Image Processing*, 9:1309–1324, 2000.

[66] Gerhard Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, 1995.

[67] Ingo J Wundrich, Christoph von der Malsburg, and Rolf P Würtz. Image representation by complex cell responses. *Neural Computation*, 16:2563–2575, 2004.