Natural Image Statistics for Human and Computer Vision

Thesis submitted for the degree of

"Doctor of Philosophy"

by

Daniel Zoran

Submitted to the Senate of the Hebrew University of Jerusalem

7/2013

This work was carried out under the supervision of
Prof. Yair Weiss

Acknowledgments

I would first like to thank my advisor Prof. Yair Weiss for his wonderful guidance, support and generosity. It has been a privilege to work under his supervision, always showing me the right direction, but allowing me to walk the path myself.

I would also like to thank my lab and office mates over the years who have kept me company, gave me support and were a constant source of encouragement.

Lastly, I would like to thank my family - my parents Rachel and Gabi and brother Yuval for inspiring me to pursue this route, my wife Maya who is always there for me and my son Yonatan who is a constant source of inspiration to me.

Abstract

The statistics of natural images are of great interest to anyone interested in problems of computer vision, image processing and biological vision. As natural images form the basic stimuli in most vision oriented tasks it is of great importance to understand their unique properties and statistical structure. In this thesis I present a series of works all attempting to better understand the statistical properties of natural images, ranging from marginal filter response statistics to a complete analytical model for whole natural image patches. In all of these works we compare the proposed models to current, cutting edge models and methods.

Natural images have several important statistical properties which manifest themselves in different contexts, many of them relevant to the works presented. Two major properties I will review here is the scale invariance of natural images and their non-Gaussian structure. Natural images have scale invariant statistics, meaning that their statistical properties remain the same when the images are scaled up or down. One of the hallmarks of this property is the power-law spectrum of natural images, the energy of the spectrum decays with frequency such that $p(f) \propto 1/f^2$. Another important property which will play a significant role in this work is the non-Gaussianity of natural images. The filter response histograms of zero mean filters applied to natural images always portray highly non-Gaussian shapes with strong peaks and heavy tails. This is a result of several different phenomena related to the structure of natural images and will be discussed in many of the works presented here.

Several works are the basis for this thesis. The first work presented here deals with marginal filter responses in natural images. We show that the kurtosis of marginal filter histograms is affected by noise present in the image. We suggest that in clean natural images this kurtosis tends to be constant with changes in the scale of the filter, and that noise causes this to constancy to change. We show that this can be used to estimate noise present images and obtain state-of-the-art performance in a variety of noise estimation tasks.

In the second work we propose a model which approximates the joint distribution of natural image patches using a tree structured graphical model learned from these patches. We show that such a model learns a set of oriented, localized, band-pass filters, resembling "simple cells" of the V1 cortex. These filters are joined together by the tree graph to form structures which resemble "complex cells" in the V1 cortex - cells which share location and orientation but differ in phase are grouped together in the tree, creating a phase invariant structure akin to complex cells. This model is compared with several models and is shown to be a strong performer in modeling natural image patches.

The last two works of this thesis present a new model for natural images which is based on a simple and common model - the Gaussian Mixture Model (GMM). In the first work in this series, we present a framework which allows patch priors to be used in whole image restoration. Using this framework and the GMM model we show that we can achieve state-of-the-art performance in a variety of image restoration tasks, competing with the most sophisticated engineered methods. The GMM prior is further analyzed in the second work in this series. In this work we attempt to explain the surprising success of the GMM. We compare the GMM performance to current cutting edge models of natural images and we show what are the properties of natural images that are captured by the model. To conclude, we propose an analytical model for natural image patches which we call "Mini Dead Leaves" which models some of the salient properties learned by our model - occlusion, texture and contrast.

Contents

Chapter 1

Introduction

1.1 Natural images

The greater part of this thesis deals with *natural images*. The term natural images is used in many different contexts in the relevant literature [1, 2, 3, 4], but there is no agreed upon definition of natural images. In the context of this work, a *natural image* would be the result of taking an ordinary digital camera, pointing it somewhere in the world and pressing the shutter button. While this is still not a very good definition, it definitely coincides with the majority of images taken in the world today. This definition also ignores many different aspects of acquiring digital images, the effects of the lens, sensor, pre- and post-processing of the image etc [5, 6]. Some of these effects are important as we will see later on in this work, and some are not, at least not to the subject at hand.

Natural images serve as the main stimuli of the human visual system, and are the main input of many computer vision systems. As such, knowing more about the structure (and statistics) of this extremely complex and diverse stimuli is important if we ever hope to understand our own visual system or to build better computer vision systems.

1.2 Natural images and statistics

Natural images are an extremely diverse and complex entity. Because they are projections of the vast world around us, they embody many of the physical properties of this world. An explicit representation for such a complex thing may be very hard to find, and as such, natural images lend them selves to a *statistical* description.

In the context of this thesis, a digital image $\mathbf{x}$ will be a matrix of $N \times M$ pixels. Each pixel may be a grayscale value (a scalar) or a color value (usually an RGB triplet). To understand the sheer size of the possible space of images, consider a small 32×32 grayscale image where each pixel can be assigned one of 256 grayscale values. This space of all possible image for this small image comprises of 256^{1024} different images. This is, of course, a huge number of possible images with a much larger number of images than atoms in this universe. A majority of these images, however, will be complete garbage - most of them will not contain anything similar to objects we see in this world, just noise. Natural images comprise only a small subset of this huge space, and hopefully they have distinct and informative properties which allow us to model them using statistics.

Given that, it seems that the main challenge in modeling the statistics of natural images, would be to find a good model that will tell us given an image $\mathbf{x}$ how *likely* it is that this image is natural. We can think of this "naturalness" score as a probability density function in image space $p(\mathbf{x})$, where we would like it to have high probability density around images that resemble natural

(a) A likely image (b) A less likely image (c) An unlikely image

Figure 1.1: An ideal statistical model of natural images would give high likelihood values to images of a natural source, and low likelihood values to other images.

images, and low probability density in other areas. See Figure 1.1 for an example of this.

This function $p(\mathbf{x})$ is the "holy grail" of natural image statistics and is the main subject of this thesis.

1.3 Statistical properties of natural images

The study of natural image statistics dates back to the 1950's where early studies on television signals revealed that their power spectrum behaves like a power-law [7, 8] (though these television signals were not called "natural images" at the time, their content inherently coincides with natural images). This power-law spectrum property will be discussed in detail in the next section. The majority of earlier works all used classical image analysis tools such as Fourier decomposition and Gabor filters [9]. These classical tools helped reveal many of the important statistical properties of natural images we know today.

Starting from the late 1980's there has been a surge of works all dealing with natural images directly [10, 11, 2, 12, 13, 14, 15, 16, 17]. While there are many works on the subject matter covering many different aspects of natural images statistics, there are several notable properties of these statistics which deserve special attention. These properties will play an important role in the first published paper in this thesis (Section 2.1), and to some extent are important to all the works presented here.

1.3.1 Scale invariance

One of the more robust phenomenon observed in natural images is the *scale invariance* of their statistics. While there are several scale invariant properties in natural images [11, 2, 3] the most notable one and by far the most studied one is the *power law* spectrum. Taking the Fourier transform of natural images reveals that the *power* of different frequency magnitudes f (that is, spatial frequencies averaged over orientations) has the following form:

$$P(f) \propto \frac{1}{f^{2-\eta}} \tag{1.1}$$

where η is usually a small constant. Consider a natural image I_1 for which the power spectrum is $\mathcal{F}_1^2(f) = \frac{c}{f^2}$ (where $c > 0$), and its down-scaled version I_2 such that:

$$I_1(a\mathbf{x}) = I_2(\mathbf{x}) \quad a > 0 \tag{1.2}$$

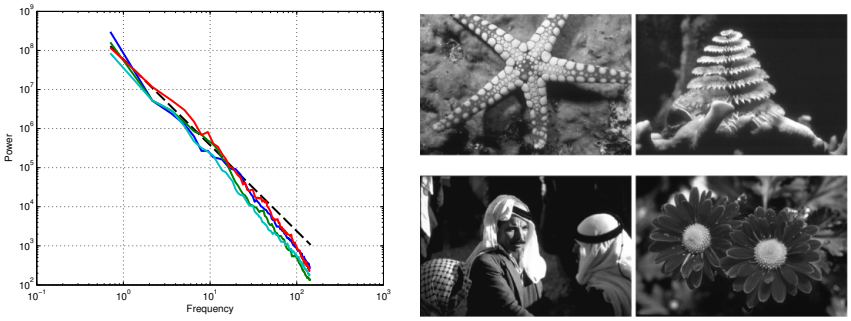


Figure 1.2: Scale invariant spectrum of natural images. The power spectrum of the four images to the right of the graph. Note that even though the images here are very different from one another, their power spectrum is very similar. The dashed line in the left plot is proportional to $1/f^{2.3}$.

Then the spectrum of I_2 will be:

$$\begin{aligned} \mathcal{F}_2^2(I_2) &= a^2 \mathcal{F}_1^2\left(\frac{f}{a}\right) \\ &= a^2 \frac{c}{\left(\frac{f}{a}\right)^2} \\ &= \frac{c}{f^2} \end{aligned} \tag{1.3}$$

where the first line is due to the Fourier scaling theorem. This means that spectrum of I_2 also follows Equation 1.1, is the same as I_1 and is scale invariant. Figure 1.2 depicts the power spectrum of several, quite different natural images. One important thing to note here is that this is an *empirical* result, there is no a-priori reason to assume natural images should have this kind of spectrum.

There are many work which attempt to explain this and related phenomena [2, 3, 18] and among those the "Dead Leaves" model is especially important [19, 18, 20, 2]. The dead leaves model is a model for generating synthetic images. With a careful selection of parameters [2, 18], the images generated by this model have statistical properties which are closely related to those of natural images and among these scale invariance and heavy tailed statistics (discussed in the Section 1.3.2). The basic process to generate an image with the dead leaves model is to place random "shapes" (say, circles or ellipses) with sizes chosen from a distribution $p(s)$ and random intensity on an image plane, in a random location. Each new shape is placed on the image plane in a random location, occluding everything beneath it (like dead leaves falling on the ground off a tree). The process stops when all pixels in the image plane have been covered by at least one shape. Figure 1.3 depicts sample images from such a model (using circles as the shape element).

While this model seems simplistic it is in fact a very rich and interesting model and is able to reproduce a lot of the statistics of natural images, depending on the distribution of sizes $p(s)$ and other model parameters [18]. Many works have been devoted to the analysis of this model, and it is out scope to survey them here.

1.3.2 Non-Gaussianity and heavy tails

Another notable property of natural images is their *non-Gaussianity*. This property comes to play in many different aspects of natural image statistics, but the simplest and most common way to observe it is to look at marginal filter histograms. Convolving an image with almost any zero mean linear filter results in a histogram similar to the one visible in Figure 1.4 (depicted here is the horizontal derivative filter response histogram). Compared to a Gaussian (also displayed on the same plot) the resulting histogram is much more heavy tailed — the probability of getting

extreme values (much larger than the standard deviation) is magnitudes larger than in a Gaussian distribution. Correspondingly, the peak around zero is much narrower (a direct result of the requirement to be normalized density function). If images were Gaussian, we would see Gaussian marginal histograms. There are many explanations to why this non-Gaussian distribution arises in natural images [21, 22, 18]. The dead leaves model which we encountered in Section 1.3.1 also reproduces this effect (as well as some related ones).

This property comes into play in almost any model we will discuss in this thesis, from modeling the statistics to image restoration, the non-Gaussianity of natural images plays an important role in the challenges for the field. In the published paper presented in Section 2.1 we will see how we can use this property together with scale invariance (see Section 1.3.1) to estimate noise content in natural images. The last two papers in this thesis (Section 2.3 and 2.4) present a model which is able to account for much of the non-Gaussianity of images and also show that conditioned on some local properties, this non-Gaussianity disappears (see the discussion in Section 3 for more details).

1.4 Learning models of natural image patches

In the previous section we encountered some important properties of natural image statistics. These properties, however, are not *learned* from data - that is, they are empirical observations made by researching images with classical signal processing tools. With the advance of computing power, storage capacities and the ever growing amount of data available, there has been a surge of works which attempt to model natural images via techniques and methods from *machine learning.*

Machine learning opens up a world of possibilities when it comes to natural image statistics. As natural images are easy to obtain - many of the images on the internet, are, in fact, natural images - the amounts of available data to feed into learning algorithms is practically infinite. Since most learning in natural image statistics is unsupervised no tagging is necessary, making the data cheap and widely available. This privilege really opens up doors for richer, more sophisticated models of natural images, and as we will see in the published papers in Section 2.3 and 2.4 of this thesis, allows us to advance the state-of-the-art in modeling natural images and image restoration.

Modern digital images have a great number of pixels — usually millions per image. This usually prohibits learning models on whole images, and constraints us to focus on learning models for image *patches*. Indeed, the majority of the works presented in this thesis focus on modeling local patch statistics, usually of tens or hundreds of pixels. Working with smaller image patches does not make the problem of modeling their statistics easier, it just makes the problem feasible. As we have seen in Section 1.2 the number of possible patches is amazingly large, and finding a model which allows us to model the natural image patches is a daunting task. The remainder of this section will be

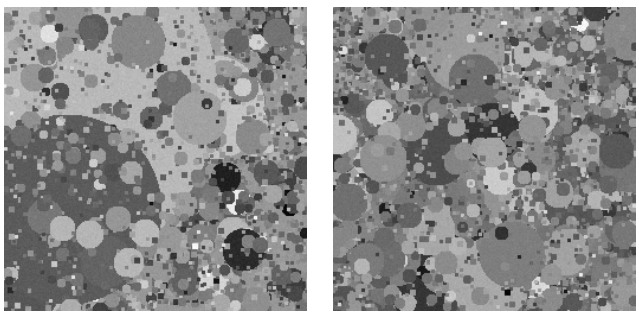


Figure 1.3: Dead Leaves model. Depicted here is a sample from a dead leaves model. While the image does not resemble a natural image, these images share many of the statistical properties of natural images and have been the focus of much research in the natural image statistics community.

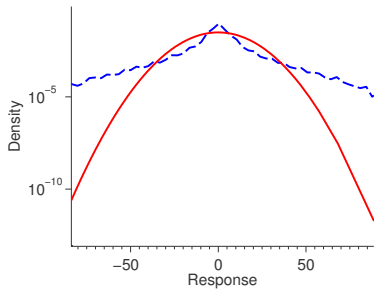


Figure 1.4: Non-Gaussianity of natural images. Depicted is the histogram of horizontal derivative filter responses over a natural image, in log scale. Note that the response histogram has much heavier tails than a Gaussian (solid red line, presented here for comparison)

devoted to presenting some of the more popular models of natural image patches of recent years.

1.4.1 PCA

Principal Component Analysis (PCA) [23] is a popular method for data analysis, dimensionality reduction and density modeling. In PCA we look for a set of linear, orthogonal projections of the data $\mathbf{W}$ for which the variance in the corresponding direction is maximized. The k-th principal component is given by:

$$\mathbf{w}_k = \arg\max_{\mathbf{w}} \sigma^2(\mathbf{w}^T\mathbf{X}) \quad s.t \quad \mathbf{w}^T\mathbf{w} = 1, \mathbf{w}_k \perp \mathbf{w}_{k-1} \tag{1.4}$$

Where $\mathbf{X}$ is the $N \times P$ data matrix having P samples in columns, each having N dimensions. Figure 1.5a depicts this basic idea. Actually finding the principal components of the data amounts to estimating the eigenvectors of the data covariance matrix, a relatively simple operation. PCA is also closely related to data *whitening* — an operation which decorrelates the data, discarding all second order statistics. Whitening is a common preprocessing step for many of the models presented in this thesis.

Interpreted as a probabilistic model [24] PCA can be regarded as a Gaussian model, taking into account only the first and second moments of the data. Under this model, the likelihood of an image $\mathbf{x}$ is simply:

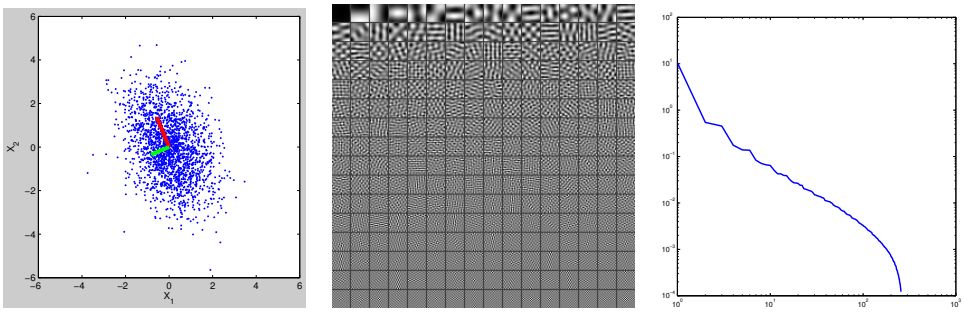
$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, \mathbf{\Sigma}) \tag{1.5}$$

Where Σ is the data covariance matrix and $\mathcal{N}$ is the multivariate Gaussian density. Figure 1.5b depicts the leading principal components of natural image patches (see also [25]). Due to the stationary nature of natural image statistics, the covariance matrix of natural image patches is approximately circulant, and as such, its eigenvectors are approximately the 2D Fourier basis vectors. Figure 1.5c also depicts the eigenvalues of the covariance matrix — it can be seen that the eigenvalues decay as a power-law much like the spectrum we have seen in Section 1.3.1.

As we have seen in Section 1.3.2, natural images have very non-Gaussian statistics and while PCA tells us an interesting story about natural images, it is definitely not the whole story. In the upcoming sections we will address some of the non-Gaussian elements of natural images, though in many cases, the first step would be discarding the second order statistics, those which are indeed captured by PCA.

1.4.2 ICA

We have seen that PCA allows us to *decorrelate* the data, that is, discard all second order statistics. This is an example of *redundancy reduction*, obtaining a more compact representation of data. This has been suggested as one of the goals of the sensory system [1, 11, 15, 26, 27]. When dealing



(a) (b) (c)

Figure 1.5: Principal Component Analysis (PCA). **(a)** The basic idea of PCA is to find the directions in which the projected data variance is maximized. In the example here the red line is the first principal component and the green is the second. **(b)** The principal components of natural image patches — note how the eigenvectors resemble the Fourier basis, a result of the stationarity of image statistics. **(c)** The eigenvalues associated with the principal components, note the power-law like decay.

with Gaussian data PCA makes the resulting projections *independent* — the ultimate form of redundancy reduction. For N dimensional data we move from a probability density in N dimension to N univariate densities, a significant reduction indeed!

Making non Gaussian data such as natural images independent is more challenging. Since there are higher order correlations PCA is not able to achieve this goal and for the purpose Independent Component Analysis (ICA) has been developed. ICA is a family of algorithms which share a common goal: finding a linear projection of the data $\mathbf{W}$ which makes the resulting data projections as *statistically independent* as possible [17, 28].

There are many different ways to achieve this goal [29, 28], but one of the simpler ones is gradient ascent on a factorial objective likelihood function. The first step of this method is *whitening* the data. As we have seen above this gets rid of all second order correlations and can be done easily using PCA:

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T\mathbf{X} \tag{1.6}$$

where $\mathbf{D}$ and $\mathbf{V}$ are the diagonal eigenvalue matrix and the eigenvector matrix correspondingly.

After whitening we search for an *orthonormal* filter matrix which maximizes the following log likelihood:

$$\log L(\mathbf{X}) = \sum_{p=1}^{P}\sum_{i=1}^{N} \log f(y_i^p) \quad \mathbf{Y} = \mathbf{W}\mathbf{Z} \tag{1.7}$$

where $f(x)$ is a leptokurtotic density function - usually the Laplace density, and $\mathbf{Z}$ is defined in Equation 1.6. The reason we look for an orthonormal filter matrix $\mathbf{W}$ is that *every* rotation of a whitening matrix is also a whitening matrix - restricting the search space considerably. Using gradient ascent on the log likelihood in Equation 1.7 and constraining the matrix $\mathbf{W}$ to be orthonormal after each step we can find the ICA filter matrix for $\mathbf{X}$.

Figure 1.6 shows the ICA filter matrix obtained by training on natural image patches. This seminal result, closely related to the results we will see in the next section, is one of the most celebrated results of natural image statistics. The resulting filters are localized in space and frequency, and are selective for orientation and phase [17, 30]. This resembles the receptive field of simple cells neurons in the primary visual cortex [17, 31] as well as some popular image transforms such as wavelets and Gabor filters [17].

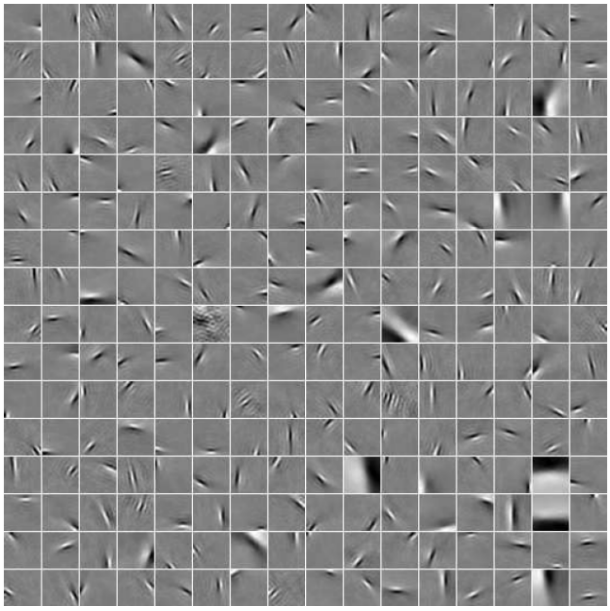


Figure 1.6: Independent Component Analysis (ICA) filter matrix trained on natural image patches. Note how the resulting filters are localized in space and frequency and selective for orientation and phase. These filters resemble the receptive field of "simple cell" in the primary visual cortex as well as image transforms such as the Gabor filter and wavelets. (replotted from [25])

1.4.3 Sparse coding

Sparse coding is a the name given for a large variety of models, all trying to find a *sparse* representation of data [32, 15, 33, 34, 35]. The exact details differ from discipline to discipline but in its essence in sparse coding we look for a representation of the data in which most of the variables will be zero, and only a small subset will be assigned non-zero values. This is usually obtained by using a dictionary — a set of linear basis functions which are linearly mixed by the sparse coefficients to form the image. This dictionary may be hand designed or adapted to the data. Sparse coding models have become extremely popular in recent years, both in the neuroscience community [15, 36, 37, 38] and the image processing and computer vision community [39, 40, 41, 42],

From a generative perspective sparse coding can be modeled as:

$$\mathbf{x} = \mathbf{D}\alpha + \eta \tag{1.8}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the image patch, $\mathbf{D} \in \mathbb{R}^{N \times M}$ is the sparse coding dictionary, $\alpha \in \mathbb{R}^M$ is the sparse representation of coefficients and η is zero mean Gaussian noise. In general we require $M \geq N$ where in cases where the equality holds (the *complete* case) the problem is closely related to ICA from Section 1.4.2 [25] and learning of the dictionary is much simpler.

The learning problem in sparse coding is two fold, given a set of patches we would like to find both the sparse representation vector α for each patch as well as the dictionary $\mathbf{D}$ which would make it easier to find sparse representations of the data. Both problems have many different formulations [34, 35]. A common formulation for the former problem is:

$$\hat{\alpha} = \arg\min_{\alpha} \|\mathbf{x} - \mathbf{D}\alpha\|^2 + \lambda \|\alpha\|_0 \tag{1.9}$$

where $\|\cdot\|_0$ denotes the L_0 norm and λ is the regularization parameter. The first term is the reconstruction error, constraining the patch to be close to the observed patch while the second term The problem with this formulation is that it makes this problem non-convex. While there are several algorithms which attempt to solve this problem directly, the common solution to this is

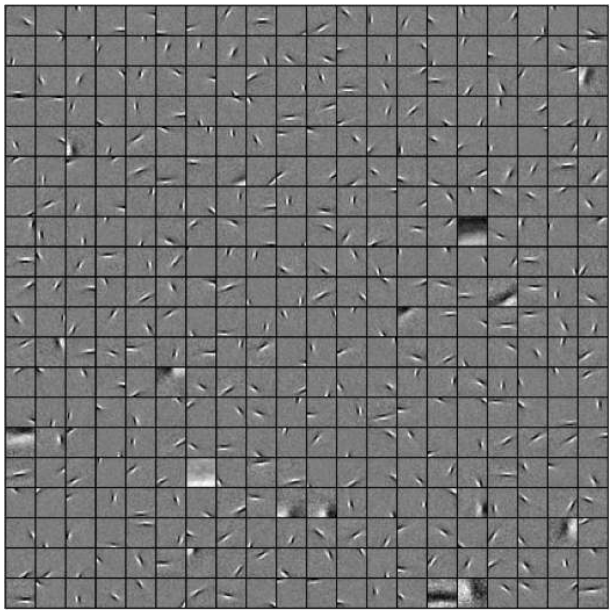


Figure 1.7: Sparse coding dictionary trained on natural image patches. The resulting basis function resemble Gabor filters, compare to Figure 1.6. (Image by David Field)

to use the L_1 norm [15, 35] (or any other sparsity inducing norm [35] for α). This relaxed version is easier to solve and there is a plethora of methods that find the sparse coefficients. In cases where we want to learn the dictionary together with the (relaxed) sparse coefficients the problem becomes finding:

$$\hat{\alpha}, \hat{\mathbf{D}} = \arg\min_{\alpha, \mathbf{D}} \|\mathbf{x} - \mathbf{D}\alpha\|^2 + \lambda \|\alpha\|_1 \tag{1.10}$$

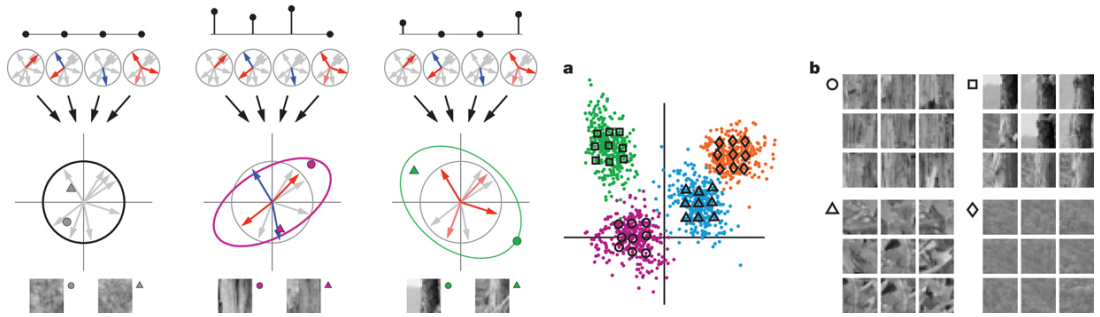
This is usually solved in an iterative manner — first solving for α given the dictionary $\mathbf{D}$ and then solving for $\mathbf{D}$ given the sparse coefficients α.

When applied to natural images the resulting dictionary elements (basis functions) resemble Gabor filters [15] much like the ones we have in Section 1.4.2. This celebrated result once more ties links between natural image statistics and neuroscience — one can argue that if our brain is looking for a sparse representation of natural images then this can explain the observed properties of simple cells in the primary visual cortex [15, 37].

1.4.4 Hierarchical models

All the models we have seen thus far are single layer models, each patch is a linear superposition of basis functions weighted by the corresponding coefficients which may be sparse, independent etc. depending on the model. These kind of models have limited a expressive power because it is hard to model the higher order interactions explicitly [43]. Several *hierarchical* models have been suggested in recent years [14, 44]. In all of these models there is an additional layer of interaction which accounts for the dependencies between the different coefficients in each model. The most notable of these models is the model by Karklin and Lewicki [44].

This model is composed of two layers, depicted in Figure 1.8. From a generative perspective, the model consists of a layer of n hidden "neurons" y_j which are assumed to be sparse and independent. These neurons modulate a set of basis functions (a dictionary) $\mathbf{B}$ weighted by weighting matrix $\mathbf{W}$ which connects each neuron to all the dictionary elements in $\mathbf{B}$. The modulated set of dictionary



(a) (b)

Figure 1.8: **(a)** The model by Karklin and Lewicki [44], a prime example of a hierarchical model. Sparse "neurons" (top bars) are used to linearly combine basis vectors which form covariance structures (middle rows). Different combinations of basis vectors form different "families" of covariance structures and image patches. **(b)** Generalizations learned by the model - depicted are non-linear projections resulting from the model, together with characteristic patches from each cluster. Note that that clusters generalize different families of patches - edges, textures and more. (Image replotted from [44])

elements is used to create a covariance matrix $\mathbf{C}$ such that:

$$\log \mathbf{C} = \sum_{jk} y_j w_{jk} \mathbf{b}_k \mathbf{b}_k^T \tag{1.11}$$

Finally, the image patch $\mathbf{x}$ conditioned on the activations of the hidden neuron $\mathbf{y}$ is multivariate Gaussian with covariance $\mathbf{C}$:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; 0, \mathbf{C}) \tag{1.12}$$

Training and inference procedures for this model are out of scope for this thesis, but when trained on natural image patches, the model parameters $\mathbf{B}$ and $\mathbf{W}$ learn to capture several interesting properties of natural images. The dictionary elements in $\mathbf{B}$ learn basis functions which again resemble Gabor filters, localized in space and frequency and selective for orientation and phase. Here, however, these basis functions are combined together to create intricate receptive fields which code for edges, boundaries and textures. See Figure 1.8b for examples of this.

1.5 Image restoration using natural image statistics

A basic problem in low-level vision research is the problem of image restoration [45, 46, 41, 42]. Given an image which has been corrupted by some corruption process we wish to recover the clean, original image. In almost all cases this is a highly under determined problem having many more unknowns than constraints and as such requires regularization of some sort. This regularization term is where natural image statistics may help us. If we have a "good" model for clean images, that is, a model which knows how clean images should look like, then we might be able to recover such a clean image given a corrupted measurement. When using a *prior* to help us in image restoration we are in fact taking the Bayesian approach to image restoration.

1.5.1 The corruption model

There are many ways to corrupt an image, starting from the acquisition process: the lens, the sensor, through the analog to digital conversion and quantization processes and ending with compression and representation artifacts [5]. Since it is out of scope for this thesis to review this vast field of research we will focus on a simple problem formulation which covers a variety of corruption process in a simple model. Given a clean image $\mathbf{x}$ we generate the corrupted image $\mathbf{y}$ using the

following model:

$$\mathbf{y} = \mathbf{DHx} + \eta \tag{1.13}$$

Where η is Gaussian noise with known statistics, $\mathbf{H}$ is a convolution matrix with known blurring kernel and $\mathbf{D}$ is a down-sampling matrix (also known). This formulation covers the problem of *image denoising* and *image inpainting* when $\mathbf{D}$ and $\mathbf{H}$ are the identity matrix, *non-blind image deconvolution* when $\mathbf{D}$ is the identity, and *single image super-resolution* when $\mathbf{D}$ is a down-sampling matrix. In all cases we are given $\mathbf{y}$ and we wish to recover $\mathbf{x}$ knowing only the noise statistics and the matrices $\mathbf{D}$ and $\mathbf{H}$ (the related *blind* case, where we need to also estimate the corruption parameters will not be discussed here, though the published paper in Section 2.1 deals directly with the problem of estimating the statistics of the noise η in case they are unknown).

As can be seen, since we don't know the noise realization, only its statistics, at best we have two times the unknowns than equations — a hard problem indeed! We will see in the next section how natural image statistics can help us with solving this problem.

1.5.2 Using a prior for image restoration

Suppose we have a prior for natural images $p(\mathbf{x})$ and that we are given a corrupted image $\mathbf{y}$ from which we wish to restore the unknown clean image $\mathbf{x}$. We assume we have the complete corruption model $p(\mathbf{y}|\mathbf{x})$ the process is similar for other restoration tasks. There are two sensible solutions for this problem, one is to look for the *maximum a-posteriori* (MAP) solution for the problem:

$$\begin{aligned}
\hat{\mathbf{x}}_{MAP} &= \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \\
&= \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \\
&= \arg\max_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x})
\end{aligned} \tag{1.14}$$

This looks for most likely estimate given the corrupted observation and prior. Actually computing the result for Equation 1.14 may be very easy or very hard, depending on the exact problem setting. The alternative to finding the MAP estimate is to find the Bayesian Least Squares (BLS) or the conditional mean:

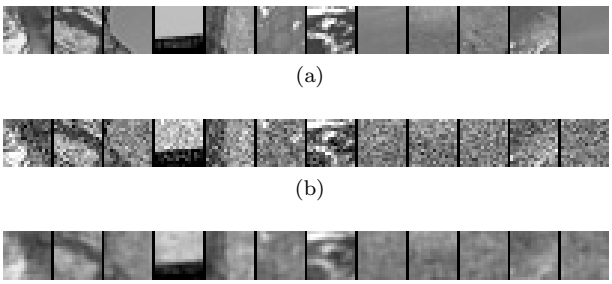
$$\begin{aligned}
\hat{\mathbf{x}}_{BLS} &= \mathbb{E}_{p(\mathbf{x}|\mathbf{y})}(\mathbf{x}) \\
&= \int_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})\mathbf{x}\, d\mathbf{x} \\
&= \frac{1}{p(\mathbf{y})} \int_{\mathbf{x}} p(\mathbf{x})p(\mathbf{y}|\mathbf{x})\mathbf{x}\, d\mathbf{x}
\end{aligned} \tag{1.15}$$

Equation 1.15 can again be quite easy or quite hard to solve, depending on the exact problem setting. Theoretically, the BLS estimate is the optimal estimate if one wants to minimize the Mean Squares Error (MSE). However, this is true only if the prior $p(\mathbf{x})$ is the "true" prior for the observed data. Since we have no guarantee for this in the case of images (if we would have known the true $p(\mathbf{x})$ there wouldn't be much point in writing this thesis), it is sometimes preferable to use the MAP estimate and sometimes the BLS estimate, depending on the exact setting, computational constraints etc.

As an example to this, let's consider the case for denoising an image corrupted with zero mean Gaussian white noise with variance σ^2, and a Gaussian prior on images with covariance $\mathbf{\Sigma}$ and zero mean. This means that:

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I}) \tag{1.16}$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|0, \mathbf{\Sigma}) \tag{1.17}$$



(a)

(b)

(c)

Figure 1.9: Image denoising using natural image statistics. **(a)** Clean image patches **(b)** Images corrupted by white Gaussian noise, $\sigma = 25$ **(c)** Restored images using a Gaussian prior trained on natural images

Solving for the MAP, Plugging Equations 1.16 and 1.17 into Equation 1.14 we get:

$$\hat{\mathbf{x}}_{MAP} = \arg\max_{\mathbf{x}} -\frac{1}{2}\left(\mathbf{y}-\mathbf{x}\right)^T\left(\sigma^2\mathbf{I}\right)^{-1}\left(\mathbf{y}-\mathbf{x}\right) - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} \tag{1.18}$$

taking the derivative w.r.t to $\mathbf{x}$ and setting to 0 we obtain:

$$\hat{\mathbf{x}}_{MAP} = \left(\sigma^2\mathbf{I}+\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\Sigma}\mathbf{y}$$

Unsurprisingly, this is just the the Wiener filter solution, which is exactly the solution for this example. This case is one rare case where the MAP solution has a closed form solution. In many in the cases we will encounter below there is no closed form solution and an approximate MAP estimation procedure is required. Coincidentally, the MAP solution for this case is also the BLS solution but this is certainly not true in the general case, of course. Figure 1.9 depicts clean images, noisy images and denoised images using this Gaussian model.

1.6 Interim summary

In this section, I have tried to give a broad overview of the relevant works from the natural image statistics literature. We have seen some of the more important properties of natural images, scale invariance and non-Gaussianity and we have encountered some recent models for natural images, as well as how to apply these models to image restoration tasks. The upcoming sections are the main part of this thesis, and includes four published works. Each of these works stems from at least some of the works we have seen here, and hopefully expands our understanding of the implications of these important works.

Chapter 2

Results

2.1 Noise and scale invariance in natural images

Scale Invariance and Noise in Natural Images

Daniel Zoran
Interdisciplinary Center for Neural Computation
The Hebrew University of Jerusalem, Israel
daniez@cs.huji.ac.il

Yair Weiss
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel
http://www.cs.huji.ac.il/~yweiss/

Abstract

Natural images are known to have scale invariant statistics. While some eariler studies have reported the kurtosis of marginal bandpass filter response distributions to be constant throughout scales, other studies have reported that the kurtosis values are lower for high frequency filters than for lower frequency ones. In this work we propose a resolution for this discrepancy and suggest that this change in kurtosis values is due to noise present in the image. We suggest that this effect is consistent with a clean, natural image corrupted by white noise. We propose a model for this effect, and use it to estimate noise standard deviation in corrupted natural images. In particular, our results suggest that classical benchmark images used in low-level vision are actually noisy and can be cleaned up. Our results on noise estimation on two sets of 50 and a 100 natural images are significantly better than the state-of-the-art.

1. Introduction and Related Work

1.1. Scale Invariance in Natural Images

One of the most striking properties of natural image statistics is their scale invariance [14]. The most notable scale invariant property is the power-law spectrum. When decomposing an image to its local bandpass filter components, the power, or the variance of coefficient distributions decays as a power-law of the form: $P(\omega) = \frac{A}{|\omega|^{2-\eta}}$ where η is usually a small number and ω is the magnitude of the spatial frequency [15]. This property is very robust and holds across different images and scenes. Various other properties of natural images have been shown to be scale invariant as described in [13, 14].

Natural images, in addition to having scale invariant statistics, are also extremely non-gaussian. The distributions of the different wavelet coefficients, for example, have very large peaks, heavy tails and are highly kurtotic. These distributions can be generally well fitted with a generalized gaussian distribution, which captures this distinctive shape [1]. The kurtosis of a generalized Gaussian distribution is directly dependent on its shape parameter α. Assuming x is generalized Gaussian distributed such that $x \sim GG(\mu, \sigma^2, \alpha)$ where μ is the mean, σ^2 the variance and α is the shape parameter, the kurtosis of x is:

$$\kappa_x(\alpha) = \frac{\Gamma(\frac{1}{\alpha})\Gamma(\frac{5}{\alpha})}{\Gamma(\frac{3}{\alpha})^2} \quad (1)$$

Where Γ is the standard gamma function [4]. As can be seen, the kurtosis is inversely related to the shape parameter α. For natural images, α is usually rather small, having values of between 0.5 and 1 [15].

In light of this, one would expect to see some sort of scale invariance in the kurtosis of marginal coefficient distributions, specifically, a reasonable assumption would be that the kurtosis should be constant throughout scales. This, however, is not always the case. There is inconclusive evidence to whether the kurtosis values change with the scale of the measured filter response distribution. In [6] it has been reported that the kurtosis is constant throughout scales for DCT filters marginal distributions, whereas in [1] it has been reported that the kurtosis changes with scale. Specifically, in [1] it is reported that for higher frequencies, the kurtosis values are lower than for the low frequency ones.

Figure 1 shows two different natural images, one is a natural scene captured in good light conditions and the other is the ubiqitous "Lena" image, which also depicts a natural scene. As can be seen in the figure, while the kurtosis values for the natural image are more or less constant throughout scales, the Lena image displays changes in kurtosis values in different scales. Higher frequency filter responses have less kurtotic distributions than lower frequency ones.

In this work we propose a model which explains this discrepancy. We suggest that the kurtosis of marginal distributions in clean, natural images should be constant throughout scales, and that noise added to the image at various stages of production causes the kurtosis values to change, and violates the scale invariance principle. Figure 2 shows the result of adding noise at different standard deviations to a

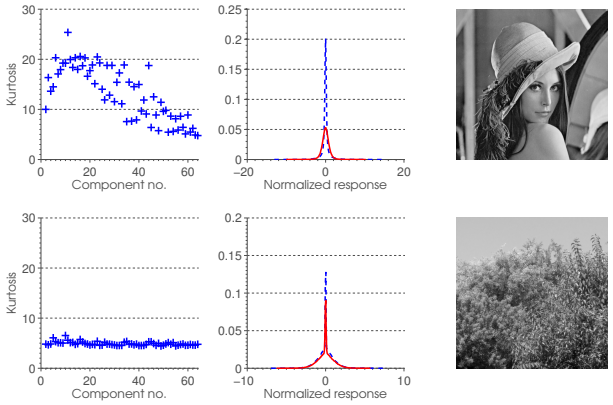


Figure 1: Kurtosis values for two different natural images. Top row on the left is the kurtosis profile - the kurtosis as a function of component number, or frequency, for the Lena image. Kurtosis values for higher frequencies have lower values than for low frequencies. In the middle of the row are the response histograms of two components for this image - the 12th (Low Frequency) and 55th (High Frequency) normalized by their variance. As can be seen, the lower frequency have a higher peak than the high frequency, making it more kurtotic. Bottom row shows the same plot for a clean natural image, taken at good light and downscaled. Kurtosis values are constant throughout scales. This can be seen in the response histogram as well. All results are for 8×8 DCT filters, both images are 512×512 pixels.

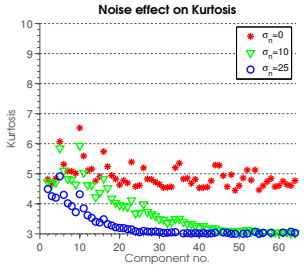


Figure 2: Result of adding noise at different standard deviation values to a clean image. The figure depicts the kurtosis of marginal filter response distribution for the DCT filter basis (8×8 pixels) as a function of spatial frequency. As the noise standard deviation rises, the kurtosis values drop, and the shape of the graph distorts with more change in the higher frequencies.

clean, natural image. As the noise standard deviation rises, kurtosis values drop, and more so for higher frequencies.

1.2. Estimating Noise Standard Deviation in Images

Many low-level computer vision algorithms combine the image evidence with a prior or regularization term. The relative weight of these two terms depends crucially on the observation noise and many computer vision algorithms assume this noise level is given as input to the algorithm [12, 11, 16, 18]. Different noise models are used in different algorithms but by far the most common model for noise is an additive, white Gaussian noise (sometimes referred to as AWGN). There has, however, been much effort to estimate the observation noise automatically. For the case of color images, Liu et al. [8] showed how an assumption of piecewise constant color allows estimating noise from a single image.

For gray level images, the MAD framework [19] uses the deviation from a smooth image model to estimate the noise. Specifically, two state-of-the-art methods [2, 10] take a similar approach. A Laplacian filter is convolved with the image, removing most second order dependencies between neighboring pixels. Since pixels in natural images have very high correlations between neighbours, this effectively removes most of the information in the original image. The only information that remains is at edges in the original image and the noise itself. Estimating the noise variance (or standard deviation) from the Laplacian image usually results in overestimation. This overestimation is due to edges contributing to the overall variance. To compensate for this, [2] apply a non-linear decay function over higher values of the block variance histogram in an iterative manner. [10] take a different approach - a Sobel edge detector is applied to the image, and using an adaptive threshold, edge pixels are marked and removed from the statistics. Both methods work quite well in general, but in images with prevelant edges, they overestimate the noise variance considerably.

The most similar approach to ours is that of Stefano et al. [3]. However, they assume a Laplacian distribution for the marginals, and do not assume anything about scale invariance or the relation between different wavelet coefficients. They do note, however, that their method works best for higher frequencies in which the SNR is lower - where the change in kurtosis is more pronounced.

We propose a method for noise estimation the relies on the assumption that kurtosis values of different scale filter distributions should not change with scale, and that any systematic change in these values is due to added noise. Our method performs much better on low-noise corrupted, elaborate natural image than current methods, and is comparable to other methods when the noise is higher, or the image simpler.

2. Model

2.1. Noise and the Generalized Gaussian Distribution

We start by modeling the change in kurtosis of a generalized Gaussian distributed random variable due to added

Gaussian noise. Denote x a generalized Gaussian random variable such that $x \sim GG(\mu, \sigma_x^2, \alpha)$ and denote η an independent Gaussian random variable with zero mean and variance σ_n^2. Denote y a random variable such that:

$$y = x + \eta$$

We wish to calculate the kurtosis κ_y of y. Going back to images, x would represent the original distribution for a local coefficient (for example) in the clean image, η the noise added and y the measured coefficient distribution from the noise corrupted image.

In this work, we refer to kurtosis as the fourth central moment normalized by the variance squared, or:

$$\kappa = \frac{\mu_4}{\sigma^4}$$

This is not the same as *excess kurtosis* which also includes a -3 term that makes the kurtosis of the Gaussian distribution zero. Due to the independence of noise, the variance of y is simply the sum of σ_{GG}^2 and σ_n^2 or:

$$\sigma_y^2 = \sigma_x^2 \left(1 + \frac{\sigma_n^2}{\sigma_x^2}\right)$$

The fourth central moment of y is easy to calculate using the cumulants and independence of noise:

$$\mu_4(y) = 3\sigma_x^4 \left(1 + \frac{\sigma_n^2}{\sigma_x^2}\right)^2 + \sigma_x^4 \left(\kappa_x(\alpha) - 3\right)$$

Where $\kappa_x(\alpha)$ is defined in Eq. 1. Finally, by normalizing with the squared variance calculated above we get:

$$\kappa_y = \frac{\kappa_x(\alpha) - 3}{\left(1 + \frac{\sigma_n^2}{\sigma_x^2}\right)^2} + 3 \tag{2}$$

Using this result we can predict the kurtosis of a marginal filter response distribution taken from a noise corrupted image, given the original image. This, however, is not usually the case as the original image is typically not available. In the next section we describe a method to estimate σ_n^2 using measurements of the noisy variable y and the assumption that κ_x is unknown, but constant throughout measurements of x for different scales.

2.2. Noise Estimation using Scale Invariance

At the base of our noise estimation procedure is the assumption that the original, uncorrupted image had scale invariant statistics. Specifically, we assume that the kurtosis of marginal filter response distributions for the original image is an unknown constant, and that adding noise to the image resulted in changes to kurtosis values throughout scales for the corrupted image.

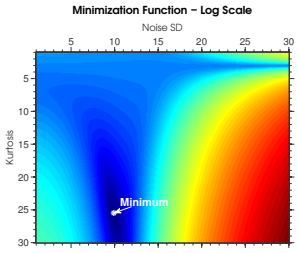


Figure 3: Function space for Eq. 3. The minimum is the actual point found numerically. Noise added to the image had a standard deviation of 10.

The first step of the algorithm is gathering statistics over the noise corrupted image I_n. We convolve the image with each filter from the $N \times N$ DCT basis, to produce a response image, y_i for the i-th filter. We estimate the variance and kurtosis for this response image to obtain $\hat{\sigma}_{y_i}^2$ and $\hat{\kappa}_{y_i}$. We do this procedure for every component i in the range $2..N^2$, hence ignoring the DC component.

Given these variance and kurtosis measures, we wish to estimate the variance of the added noise. This is done by finding the pair $\hat{\kappa}_x$, the kurtosis of the original uncorrupted image distribution and $\hat{\sigma}_n^2$ the variance of the noise, which minimizes:

$$\hat{\kappa}_x, \hat{\sigma}_n^2 = \underset{\kappa_x, \sigma_n^2}{\arg\min} \sum_{i=2}^{N^2} \left| \frac{\kappa_x - 3}{\left(1 + \frac{\sigma_n^2}{\hat{\sigma}_{y_i}^2 - \sigma_n^2}\right)^2} + 3 - \hat{\kappa}_{y_i} \right| \tag{3}$$

Figure 3 shows an example of what the function space look like, when the noise added to the image has a standard deviation of 10. As can be seen, there is a rather pronounced valley at the minimum point (which the was numerically found).

2.3. Non Gaussian Noise

Although the above model assumes white Gaussian noise, it assumes that it is white and Gaussian in the filter domain only. Since many types of independent noise in the pixel domain will mix in to Gaussian noise in the filter domain, this method work with other types of noise. The summation over the noise in pixels while calculating the response image for the DCT basis causes the distribution of the sum to be Gaussian - due to the central limit theorem and noise independence. An example can be shown in Figure 4, even though the noise is very non Gaussian in the pixel domain, it becomes Gaussian in the coefficient domain. In Section 3 we show that adding uniform noise to images, for example, does not change the performance of our method.

Other image corruption methods, such as JPEG compression artifacts, quantization noise and sensor noise were

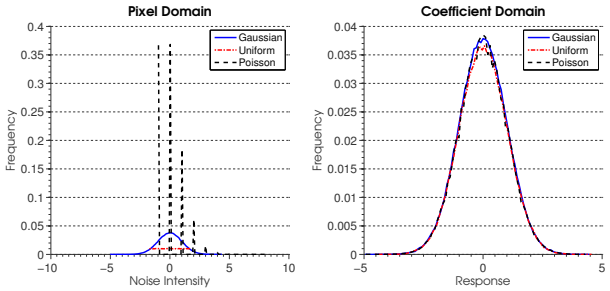


Figure 4: Response histograms of independent Gaussian, Uniform and Poisson pixel noise. Noise images were 512×512 pixels, having variance 1 and mean 0. Coefficient domain histograms made using an 8×8 DCT filter. It can be clearly seen that while the three look very different at the pixel domain, all three are mixed to be Gaussian with the same variance and mean in the coefficient domain.

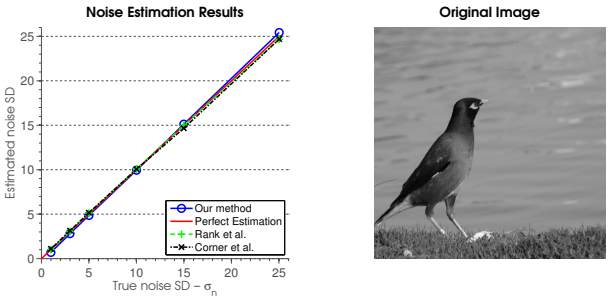


Figure 5: Noise estimation on a simple image, with no prevalent edges (BIRD). All methods estimate the noise correctly for a range of values.

tested. Results are shown in the next section.

3. Results

3.1. Methods

We first compared the proposed method with existing methods by synthetically adding noise to clean images. We minimized equation 3 using MATLAB's fminsearch function.

3.2. Noise Estimation Results

When the original image is relatively simple, such as the one depicted in Figure 5 algorithms perform similarly, estimating the noise variance well for a range of values. However, when using a more complex image, with a lot of textured areas such as the one depicted in Figure 6, the picture changes. All results are means over 3 noise realizations - standard deviations were not included in the graphs because they were too low for all methods to be visible in the graph.

Due to the large amount of edges in the image in Figure 6, the methods in [2, 10] over estimate the noise greatly. Our method performs better on the low to medium noise regime (σ_n between 1 to 15). As the noise levels rise, it is harder for our method to accurately estimate the noise. The reason for overestimation of the noise SD in other methods is that the textured areas have a lot of edges, causing the Laplacian filtering be less effective at separating the noise from the original image. Since the textured areas are from a natural image and have scale invariant statistics, our methods handles this quite well, and enables us to recover the noise SD more precisely. When the noise is sufficiently large, the variance contributed from the edges is small relatively to the noise variance, hence the similar performance at this regime. Table 1 summarizes results for all experiments. In the results table the estimation results for uniform noise can be observed. It seems that all methods handle this type of noise relatively well. Using different patch sizes for our method resulted in little difference in performance, it seems that the central limit theorem comes in to play even on small patch sizes (8×8 in this case, 64 pixels are more than enough).

Finally, we estimate noise in 50 images from the Van Hateren natural image database [17] as well as for a 100 images from the Berkeley database [9]. White Gaussian noise was added to each of the images and noise was estimated from them. It is obvious our method is on par or better than other methods for all noise levels, for both databases. Calculating the relative error for all noise levels over the van Hateren database we obtain a mean error rate of 3%, while other methods obtain 22% for Rank et al. and 28% for Corner et al. For the Berkeley our method obtains an average error that is 9%, while other methods obtain 49% (Rank et al.) and 65% (Corner et al.). Differences were even larger when neglecting the lowest noise level estimation (in which even a small error in estimation leads to a large relative error, results are then 1% for our method and 20% (Rank et al.) and 22% (Corner et al.)). Results are shown in Table 1.

3.3. Other types of noise and corruption scenarios

We tested the proposed method under several other image corruption scenarios. Under all scenarios under which there's no clear parameter for the noise standard deviation σ_n, we calculate the standard deviation of the difference between the original image I and the corrupted image I_n such that:

$$\sigma_n = \langle |I - I_n| \rangle \tag{4}$$

Where the mean is over all the pixels in the images.

The first scenario we tested is first motion blurring an image, and then adding Gaussian noise to it. This scenario is important as correct estimation of noise prior to image deblurring is important to minimize common artifacts [7].

	BIRD			FIELD			Van Hateren			Berkeley		
σ_n	$\hat{\sigma}_n$	$\hat{\sigma}_n^R$	$\hat{\sigma}_n^C$	$\hat{\sigma}_n$	$\hat{\sigma}_n^R$	$\hat{\sigma}_n^C$	$\hat{\sigma}_n$	$\hat{\sigma}_n^R$	$\hat{\sigma}_n^C$	$\hat{\sigma}_n$	$\hat{\sigma}_n^R$	$\hat{\sigma}_n^C$
1	0.68	1.01	1.1	2.06	7.16	8.3	0.89 ± 0.71	2.02 ± 1.19	2.25 ± 1.1	1.5 ± 1.8	2.9 ± 2.8	3.7 ± 2.8
3	2.8	3.05	3.13	3.67	8.04	8.76	2.92 ± 0.62	3.64 ± 0.89	3.75 ± 0.72	2.9 ± 1.8	4.5 ± 2.3	4.9 ± 2.3
5	4.83	5.11	5.14	5.52	9.43	9.56	4.96 ± 0.59	5.47 ± 0.72	5.55 ± 0.52	4.9 ± 1.9	6.4 ± 2	6.6 ± 1.9
10	9.98	10.11	10.08	10.35	13.56	12.89	10.13 ± 0.89	10.2 ± 0.49	10.2 ± 0.48	9.7 ± 1.8	11.1 ± 1.6	11.1 ± 1.3
15	15.15	14.97	14.98	15.34	17.9	17.05	15.32 ± 1.13	15 ± 0.38	14.87 ± 0.62	14.7 ± 1.8	15.9 ± 1.3	15.7 ± 1.1
25	25.47	24.75	23.73	25.57	26.8	26.27	26.33 ± 1.81	24.65 ± 0.28	23.9 ± 1.47	24.8 ± 2.0	25 ± 1.0	25 ± 1.2
$\langle \varepsilon \rangle$	7%	**1**%	4%	**24**%	155%	176%	**3**%	22%	28%	**9.8**%	49%	65%

Table 1: Results summary for images and methods presented, for each image the first column is our method estimation $\hat{\sigma}_n$, the second Rank et al. $\hat{\sigma}_n^R$ and finally Corner et al. $\hat{\sigma}_n^C$. Last row is the average relative estimation error. On 50 images from the Van Hateren natural image database we obtain an average 3% error rate, while current state-of-the art method obtain 22% and 28%. Results for the Berkeley database (100 images) are similar, our method out-performs current state-of-the-art methods.

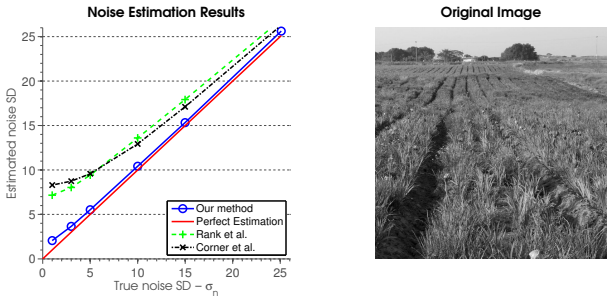


Figure 6: Noise estimation on a more complex image having a lot edges and texture data (FIELD). With low noise standard deviation other methods perform poorly, overestimating the noise by a large percentage. Out method performs much better, estimating the noise standard deviation much better.

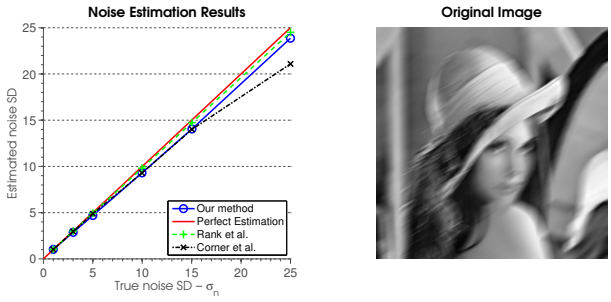


Figure 7: Noise estimation under motion blur. The image was first blurred with a motion blur mask and then corrupted by noise. Noise was then estimated by several methods. It seems that all methods estimate the noise well for a range of values.

Figure 7 shows the result of such a scenario - it seems that none of the methods is affected in any way by the blurring operation, and all methods perform well in estimating the noise standard deviation.

Second we tested whether corruption due to quantization can be estimated using this method. We quantized 256 gray scale levels images to 128, 64, 32, 16, 8 and 4 levels, noise standard deviation was estimated as in Equation 4. From the quantized image we tried to estimate this standard deviation. The proposed method works quite well, results can be seen in Figure 8.

Third, a more realistic noise scenario was used. As was done in [8], we obtained a Camera Response Function (or CRF) from [5]. With this CRF we inversely mapped an intensity image into a "lightness" image, added noise to the "lightness" image and mapped the lightness image back to a noisy, saturated, intensity image. This effectively simulates the sensor noise of a digital camera for a single channel. Results of noise estimation can be seen in Figure 9. All methods slightly underestimate the noise in the image. The reason for this is the clipping of the noise due to the CRF's saturation reduces variance near the extreme values (near 0 and near 255).

Finally, we tested whether JPEG corruption can be estimated using the proposed method. A clean image was compressed in the JPEG format using different quality levels. The compressed images were then used to estimate the corruption in them and compared to the standard deviation of the difference image. Results were very poor for all meth-

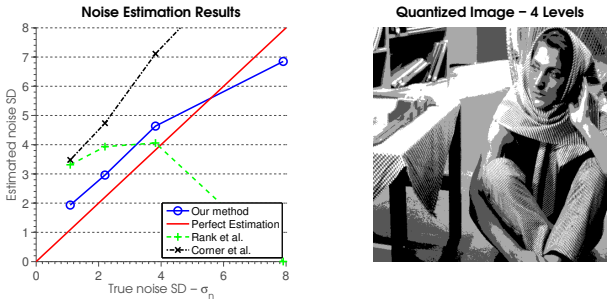


Figure 8: Noise estimation under quantization. The 256 gray scale level image was quantized to 128, 64, 32, 16, 8, and 4 levels. The standard deviation of the difference image between the original and quantized image was used as the true noise standard deviation. Estimation was done directly on the quantized image. Our method out-performs other methods on all quantization levels.

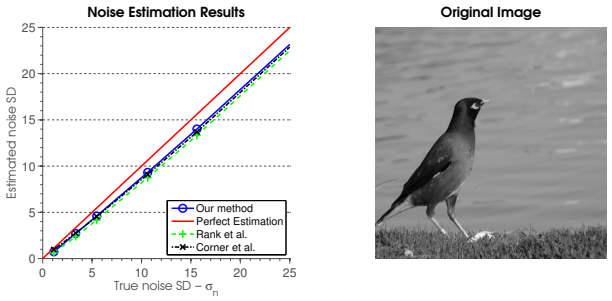


Figure 9: Noise estimation under simulated sensor noise. All methods slightly underestimate the noise in the image, due to the saturation caused by the CRF.

ods. It seems that this kind of artifacts disrupt the kurtosis of marginal filter distributions in such a way that this method can not handle properly. Specifically, since JPEG compression is primarily a frequency domain operation, the distributions change radically, having very extreme kurtosis values which are inconsistent with the above model.

3.4. Uncovering the original Lena

One particularly interesting example is the famous "Lena" image. The Lena image is very old - photographed in 1972 and scanned in 1973. One can only assume that scanning and printing technology of these days were not of the highest grade, and noise has probably corrupted the image to some extent. Looking at the kurtosis values, this is clearly evident as was shown in Figure 1. It would be interesting to see if one can estimate the noise in the original Lena image, and maybe denoise it, uncovering the original, clean image.

Estimating the inherent noise in the original, uncorrupted image by our method yields a standard deviation of $\hat{\sigma}_n = 2.08$. Other methods yield a bit more - this implies that Lena in itself is noisy, as the original kurtosis profile hints at. Taking this into account, it seems that in denoising experiments with Lena, which are very common [11] there is a maximal PSNR value that can be taken into account. When measuring the PSNR between the original image and the denoised image in a denoising experiment, any result above 41.76dB might not be possible without over fitting. It can certainly be the case that a denoising algorithm will discard some of the inherent noise in the original image, and that when measuring the PSNR value with the original image one will get a *lower* value for *cleaner* images. Of course, the inherent noise in Lena is not necessarily additive, white or Gaussian so the statement above should be taken with a grain of salt, but nevertheless, the noise is present.

Assuming that Lena is noisy, can we uncover the original, uncorrupted Lena image? We applied a simple Bayesian denoising scheme which assumes a Generalized Gaussian model for marginal filter distributions. We use our proposed noise estimation method to estimate the noise standard deviation in the image, and also the constant kurtosis value underlying in the original image. Using these two parameters, we estimated the MAP value of local DCT coefficients for all 8×8 patches of the images, and then averaged the inverse DCT results to obtain a denoised image. Results can be seen in Figure 10. Not surprisingly, estimating the noise standard deviation of the denoised image using our method yields a very low value (0.000001). This reason for this is obvious when looking at the kurtosis of the denoised image - as can be seen in Figure 11 it is almost constant throughout scales. Now we can measure the PSNR value between the original (which has noise in it) and the denoised version (which is assumed to be noise free) - this yield a PSNR of 42.1dB, consistent with what we suggest above. Denoising with a second algorithm, BRFOE by Weiss and Freeman [18] yielded similar results (see Figure 10). This is interesting because this denoising model does not assume scale invariance, and yet still, noise estimation by our method yields a very low value, since the kurtosis in the denoised image is rather constant.

4. Discussion and Future Work

In this work we describe and explain a baffling phenomena. When measuring the kurtosis of marginal filter response distributions in natural images, in many (but not all) natural images, values of kurtosis for lower frequency filters are higher than high frequency ones. This is in contrast to the scale-invariant nature of natural images. We argue that clean, natural images should have a constant kurtosis value throughout scales, and propose that deviations from this are due to noise inherent in the image. Using this assumption we show how the noise level in a corrupted image can be

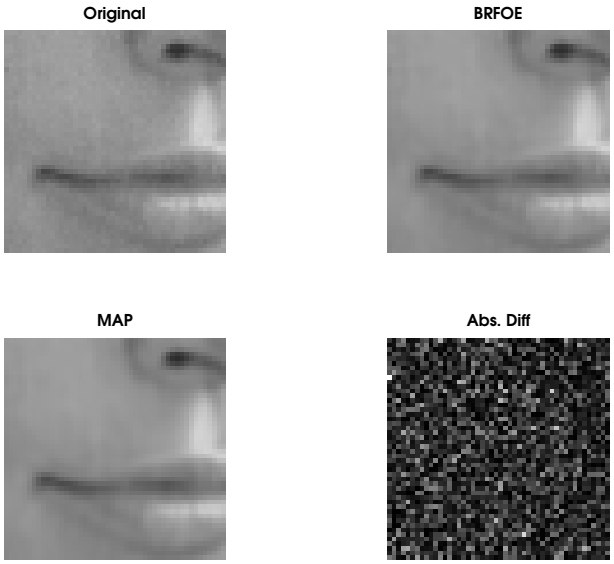


Figure 10: Denoising results for the original Lena image using simple scale-invariant Bayesian and BRFOE denoising. On the top left is the original image (detail), on the top right is the denoised image (BRFOE). Details are fully preserved in the denoised image, but noise is much less apparent. MAP denoising is on bottom left. Difference image is scaled, and with BRFOE image.

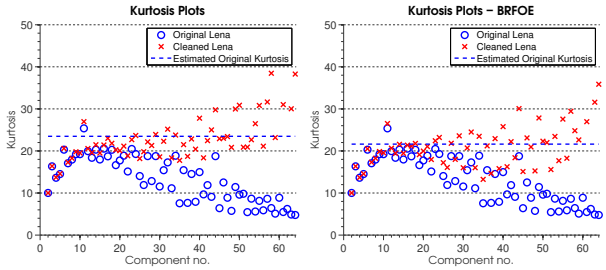


Figure 11: Kurtosis plot for the denoised Lena image and original. An almost constant kurtosis throughout most of the scales is apparent in the denoised image. The dashed line shows the Kurtosis estimated by our noise estimation algorithm. Left is the result from scale-invariant Bayesian denoising, right is the BRFOE result.

accurately estimated, for a range of scene types and noise levels, and under different corruption scenarios.

A particularly intriguing example is the ubiquitous Lena image - a very common benchmark image - which we show here to be noisy. We showed that this image is noisy in its original form, and using it as "ground truth" in low-level vision experiments should be done with caution.

Future work will include several directions. The first is investigating what other types of image corruption come into play when examining the kurtosis of marginal distributions. Second, handling non-white noise should be relatively simple as long as some model for the noise power spectrum is assumed. Finally, there's still the possibility that the kurtosis profile should have another kind of scale invariance, power-law or other, but not necessarily constant (which is a special case of power-law). Extending this method to include power-law is possible, but requires further work as merely adding another parameter to the minimization or model will not work.

References

[1] M. Bethge. Factorial coding of natural images: how effective are linear models in removing higher-order dependencies? 23(6):1253–1268, June 2006.

[2] B. Corner, R. Narayanan, and S. Reichenbach. Noise estimation in remote sensing imagery using data masking. *International Journal of Remote Sensing*, 24(4):689–702, 2003.

[3] A. De Stefano, P. R. White, and W. B. Collis. Training methods for image noise level estimation on wavelet components. *EURASIP J. Appl. Signal Process.*, 2004(1):2400–2407, 2004.

[4] J. Domínguez-Molina, G. González-Farías, R. Rodríguez-Dagnino, and I. Monterrey. A practical procedure to estimate the shape parameter in the generalized Gaussian distribution. *technique report I-01-18_eng. pdf, available through http://www. cimat. mx/reportes/enlinea/I-01-18_eng. pdf.*

[5] M. Grossberg and S. Nayar. What is the Space of Camera Response Functions? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 602–609, Jun 2003.

[6] E. Lam and J. Goodman. A mathematical analysis of the dct coefficient distributions for images. *Image Processing, IEEE Transactions on*, 9(10):1661–1666, Oct 2000.

[7] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 70, New York, NY, USA, 2007. ACM.

[8] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):299–314, 2008.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l*

Conf. Computer Vision, volume 2, pages 416–423, July 2001.

[10] K. Rank, M. Lendl, and R. Unbehauen. Estimation of image noise variance. In *Vision, Image and Signal Processing, IEE Proceedings*-, volume 146, pages 80–84, 1999.

[11] S. Roth and M. Black. Fields of Experts: A Framework for Learning Image Priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 860. IEEE Computer Society; 1999, 2005.

[12] S. Roth and M. Black. On the Spatial Statistics of Optical Flow. *International Journal of Computer Vision*, 74(1):33–50, 2007.

[13] D. L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37:3385–3398, 1997.

[14] D. L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. In *NIPS*, pages 551–558, 1993.

[15] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *J. Math. Imaging Vis.*, 18(1):17–33, 2003.

[16] M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *IEEE International Conference on Computer Vision*, volume 2, pages 900–906, 2003.

[17] J. van Hateren. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences*, 265(1394):359–366, 1998.

[18] Y. Weiss and W. Freeman. What makes a good model of natural images? *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[19] V. Zlokolica, A. Pizurica, and W. Philips. Noise Estimation for Video Processing Based on Spatio–Temporal Gradients. *IEEE SIGNAL PROCESSING LETTERS*, 13(6):337, 2006.

2.2 The "tree dependent" components of natural images are edge filters

The "tree-dependent components" of natural scenes are edge filters

Daniel Zoran
Interdisciplinary Center for Neural Computation
Hebrew University of Jerusalem
daniez@cs.huji.ac.il

Yair Weiss
School of Computer Science
Hebrew University of Jerusalem
yweiss@cs.huji.ac.il

Abstract

We propose a new model for natural image statistics. Instead of minimizing dependency between components of natural images, we maximize a simple form of dependency in the form of tree-dependencies. By learning filters and tree structures which are best suited for natural images we observe that the resulting filters are edge filters, similar to the famous ICA on natural images results. Calculating the likelihood of an image patch using our model requires estimating the squared output of pairs of filters connected in the tree. We observe that after learning, these pairs of filters are predominantly of similar orientations but different phases, so their joint energy resembles models of complex cells.

1 Introduction and related work

Many models of natural image statistics have been proposed in recent years [1, 2, 3, 4]. A common goal of many of these models is finding a representation in which components or sub-components of the image are made as independent or as sparse as possible [5, 6, 2]. This has been found to be a difficult goal, as natural images have a highly intricate structure and removing dependencies between components is hard [7]. In this work we take a different approach, instead of *minimizing dependence* between components we try to *maximize* a simple form of *dependence* - tree dependence.

It would be useful to place this model in context of previous works about natural image statistics. Many earlier models are described by the marginal statistics solely, obtaining a factorial form of the likelihood:

$$p(x) = \prod_i p_i(x_i) \tag{1}$$

The most notable model of this approach is Independent Component Analysis (ICA), where one seeks to find a linear transformation which maximizes independence between components (thus fitting well with the aforementioned factorization). This model has been applied to many scenarios, and proved to be one of the great successes of natural image statistics modeling with the emergence of edge-filters [5]. This approach has two problems. The first is that dependencies between components are still very strong, even with those learned transformation seeking to remove them. Second, it has been shown that ICA achieves, after the learned transformation, only marginal gains when measured quantitatively against simpler method like PCA [7] in terms of redundancy reduction. A different approach was taken recently in the form of radial Gaussianization [8], in which components which are distributed in a radially symmetric manner are made independent by transforming them non-linearly into a radial Gaussian, and thus, independent from one another.

A more elaborate approach, related to ICA, is Independent Subspace Component Analysis or ISA. In this model, one looks for independent subspaces of the data, while allowing the sub-components

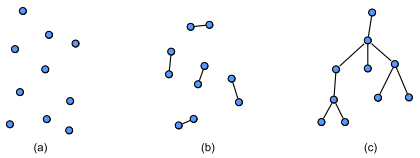


Figure 1: Our model with respect to marginal models such as ICA (a), and ISA like models (b). Our model, being a tree based model (c), allows components to belong to more than one subspace, and the subspaces are not required to be independent.

of each subspace to be dependent:

$$p(x) = \prod_k p_k(x_{i \in K}) \tag{2}$$

This model has been applied to natural images as well and has been shown to produce the emergence of phase invariant edge detectors, akin to complex cells in V1 [2].

Independent models have several shortcoming, but by far the most notable one is the fact that the resulting components are, in fact, highly dependent. First, dependency between the responses of ICA filters has been reported many times [2, 7]. Also, dependencies between ISA components has also been observed [9]. Given these robust *dependencies* between filter outputs, it is somewhat peculiar that in order to get simple cell properties one needs to assume *independence*. In this work we ask whether it is possible to obtain V1 like filters in a model that assumes dependence.

In our model we assume the filter distribution can be described by a tree graphical model [10] (see Figure 1). Degenerate cases of tree graphical models include ICA (in which no edges are present) and ISA (in which edges are only present within a subspace). But in its non-degenerate form, our model assumes any two filter outputs may be dependent. We allow components to belong to more than one subspace, and as a result, do not *require* independence between them.

2 Model and learning

Our model is comprised of three main components. Given a set of patches, we look for the parameters which maximize the likelihood of a whitened natural image patch $\mathbf{z}$:

$$p(\mathbf{z}; \mathbf{W}, \beta, T) = p(y_1) \prod_{i=1}^{N} p(y_i | y_{pa_i}; \beta) \tag{3}$$

Where $\mathbf{y} = \mathbf{W}\mathbf{z}$, T is the tree structure, pa_i denotes the parent of node i and β is a parameter of the density model (see below for the details). The three components we are trying to learn are:

1. The filter matrix $\mathbf{W}$, where every row defines one of the filters. The response of these filters is assumed to be tree-dependent. We assume that W is orthogonal (and is a rotation of a whitening transform).
2. The tree structure T which specifies which components are dependent on each other.
3. The probability density function for connected nodes in the tree, which specify the exact form of dependency between nodes.

All three together describe a complete model for whitened natural image patches, allowing likelihood estimation and exact inference [11].

We perform the learning in an iterative manner: we start by learning the tree structure and density model from the entire data set, then, keeping the structure and density constant, we learn the filters via gradient ascent in mini-batches. Going back to the tree structure we repeat the process many times iteratively. It is important to note that both the filter set and tree structure are learned from the data, and are continuously updated during learning. In the following sections we will provide details on the specifics of each part of the model.

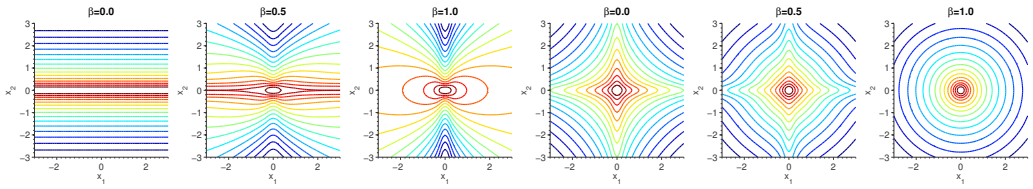


Figure 2: Shape of the conditional (Left three plots) and joint (Right three plots) density model in log scale for several values of β, from dependence to independence.

2.1 Learning tree structure

In their seminal paper, Chow and Liu showed how to learn the optimal tree structure approximation for a multidimensional probability density function [12]. This algorithm is easy to apply to this scenario, and requires just a few simple steps. First, given the current estimate for the filter matrix $\mathbf{W}$, we calculate the response of each of the filters with all the patches in the data set. Using these responses, we calculate the mutual information between each pair of filters (nodes) to obtain a fully connected weighted graph. The final step is to find a maximal spanning tree over this graph. The resulting unrooted tree is the optimal tree approximation of the joint distribution function over all nodes. We will note that the tree is *unrooted*, and the root can be chosen arbitrarily - this means that there is no node, or filter, that is more important than the others - the direction in the tree graph is arbitrary as long as it is chosen in a consistent way.

2.2 Joint probability density functions

Gabor filter responses on natural images exhibit highly kurtotic marginal distributions, with heavy tails and sharp peaks [13, 3, 14]. Joint pair wise distributions also exhibit this same shape with varying degrees of dependency between the components [13, 2]. The density model we use allows us to capture both the highly kurtotic nature of the distributions, while still allowing varying degrees of dependence using a mixing variable. We use a mix of two forms of finite, zero mean Gaussian Scale Mixtures (GSM). In one, the components are assumed to be independent of each other and in the other, they are assumed to be spherically distributed. The mixing variable linearly interpolates between the two, allowing us to capture the whole range of dependencies:

$$p(x_1, x_2; \beta) = \beta p_{dep}(x_1, x_2) + (1 - \beta) p_{ind}(x_1, x_2) \tag{4}$$

When $\beta = 1$ the two components are dependent (unless p is Gaussian), whereas when $\beta = 0$ the two components are independent. For the density functions themselves, we use a finite GSM. The dependent case is a scale mixture of bivariate Gaussians:

$$p_{dep}(x_1, x_2) = \sum_k \pi_k \mathcal{N}(x_1, x_2; \sigma_k^2 \mathbf{I}) \tag{5}$$

While the independent case is a product of two independent univariate Gaussians:

$$p_{ind}(x_1, x_2) = \left(\sum_k \pi_k \mathcal{N}(x_1; \sigma_k^2) \right) \left(\sum_k \pi_k \mathcal{N}(x_2; \sigma_k^2) \right) \tag{6}$$

Estimating parameters π_k and σ_k^2 for the GSM is done directly from the data using Expectation Maximization. These parameters are the same for all edges and are estimated only once on the first iteration. See Figure 2 for a visualization of the conditional distribution functions for varying values of β. We will note that the marginal distributions for the two types of joint distributions above are the same. The mixing parameter β is also estimated using EM, but this is done for each edge in the tree separately, thus allowing our model to theoretically capture the fully independent case (ICA) and other degenerate models such as ISA.

2.3 Learning tree dependent components

Given the current tree structure and density model, we can now learn the matrix $\mathbf{W}$ via gradient ascent on the log likelihood of the model. All learning is performed on whitened, dimensionally

reduced patches. This means that $\mathbf{W}$ is a $N \times N$ rotation (orthonormal) matrix, where N is the number of dimensions after dimensionality reduction (see details below). Given an image patch $\mathbf{z}$ we multiply it by $\mathbf{W}$ to get the response vector $\mathbf{y}$:

$$\mathbf{y} = \mathbf{W}\mathbf{z} \tag{7}$$

Now we can calculate the log likelihood of the given patch using the tree model (which we assume is constant at the moment):

$$\log p(\mathbf{y}) = \log p(y_{root}) + \sum_{i=1}^{N} \log p(y_i | y_{pa_i}) \tag{8}$$

Where pa_i denotes the parent of node i. Now, taking the derivative w.r.t the r-th row of $\mathbf{W}$:

$$\frac{\partial \log p(\mathbf{y})}{\partial \mathbf{W}_r} = \frac{\partial \log p(\mathbf{y})}{\partial y_r} \mathbf{z}^T \tag{9}$$

Where $\mathbf{z}$ is the whitened natural image patch. Finally, we can calculate the derivative of the log likelihood with respect to the r-th element in $\mathbf{y}$:

$$\frac{\partial \log p(\mathbf{y})}{\partial y_r} = \frac{\partial \log p(y_{\mathrm{pa}_r}, y_r)}{\partial y_r} + \sum_{c \in C(r)} \frac{\partial \log p(y_r, y_c)}{\partial y_r} - \frac{\partial \log p(y_r)}{\partial y_r} \tag{10}$$

Where $C(r)$ denote the children of node r. In summary, the gradient ascent rule for updating the rotation matrix $\mathbf{W}$ is given by:

$$\mathbf{W}_r^{t+1} = \mathbf{W}_r^t + \eta \frac{\partial \log p(\mathbf{y})}{\partial y_r} \mathbf{z}^T \tag{11}$$

Where η is the learning rate constant. After update, the rows of $\mathbf{W}$ are orthonormalized.

This gradient ascent rule is applied for several hundreds of patches (see details below), after which the tree structure is learned again as described in Section 2.1, using the new filter matrix $\mathbf{W}$, repeating this process for many iterations.

3 Results and analysis

3.1 Validation

Before running the full algorithm on natural image data, we wanted to validate that it does produce sensible results with simple synthetic data. We generated noise from four different models, one is $1/f$ independent Gaussian noise with 8 Discrete Cosine Transform (DCT) filters, the second is a simple ICA model with 8 DCT filters, and highly kurtotic marginals. The third was a simple ISA model - 4 subspaces, each with two filters from the DCT filter set. Distribution within the subspace was a circular, highly kurtotic GSM, and the subspaces were sampled independently. Finally, we generated data from a simple synthetic tree of DCT filters, using the same joint distributions as for the ISA model. These four synthetic random data sets were given to the algorithm - results can be seen in Figure 3 for the ICA, ISA and tree samples. In all cases the model learned the filters and distribution correctly, reproducing both the filters (up to rotations within the subspace in ISA) and the dependency structure between the different filters. In the case of $1/f$ Gaussian noise, any whitening transformation is equally likely and any value of beta is equally likely. Thus in this case, the algorithm cannot find the tree or the filters.

3.2 Learning from natural image patches

We then ran experiments with a set of natural images [9][1]. These images contain natural scenes such as mountains, fields and lakes. . The data set was 50,000 patches, each 16×16 pixels large. The patches' DC was removed and they were then whitened using PCA. Dimension was reduced from 256 to 128 dimensions. The GSM for the density model had 16 components. Several initial

[1] available at http://www.cis.hut.fi/projects/ica/imageica/

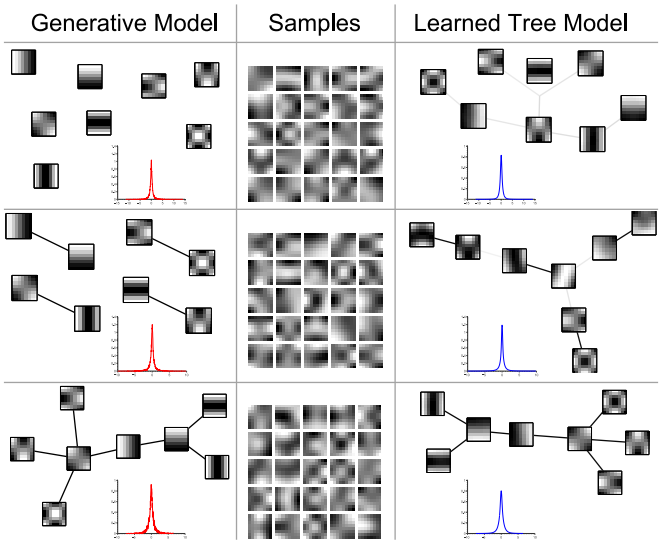


Figure 3: Validation of the algorithm. Noise was generated from three models - top row is ICA, middle row is ISA and bottom row is a tree model. Samples were then given to the algorithm. On the right are the resulting learned tree models. Presented are the learned filters, tree model (with white edges meaning $\beta = 0$, black meaning $\beta = 1$ and grays intermediate values) and an example of a marginal histogram for one of the filters. It can be seen that in all cases all parts of the model were correctly learned. Filters in the ISA case were learned up to rotation within the subspace, and all filters were learned up to sign. β values for the ICA case were always below 0.1, as were the values of β between subspaces in ISA.

conditions for the matrix $\mathbf{W}$ were tried out (random rotations, identity) but this had little effect on results. Mini-batches of 10 patches each were used for the gradient ascent - the gradient of 10 patches was summed, and then normalized to have unit norm. The learning rate constant η was set to 0.1. Tree structure learning and estimation of the mixing variable β were done every 500 mini-batches. All in all, 50 iterations were done over the data set.

3.3 Filters and tree structure

Figures 4 and 5 show the learned filters ($\mathbf{WQ}$ where $\mathbf{Q}$ is the whitening matrix) and tree structure (T) learned from natural images. Unlike the ISA toy data in figure 3, here a full tree was learned and β is approximately one for all edges. The GSM that was learned for the marginals was highly kurtotic.

It can be seen that resulting filters are edge filters at varying scales, positions and orientations. This is similar to the result one gets when applying ICA to natural images [5, 15]. More interesting is

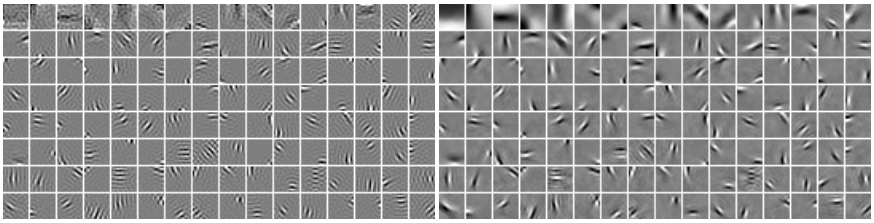


Figure 4: Left: Filter set learned from 16×16 natural image patches. Filters are ordered by PCA eigenvalues, largest to smallest. Resulting filters are edge filters having different orientations, positions, frequencies and phases. Right: The "feature" set learned, that is, columns of the pseudo inverse of the filter set.

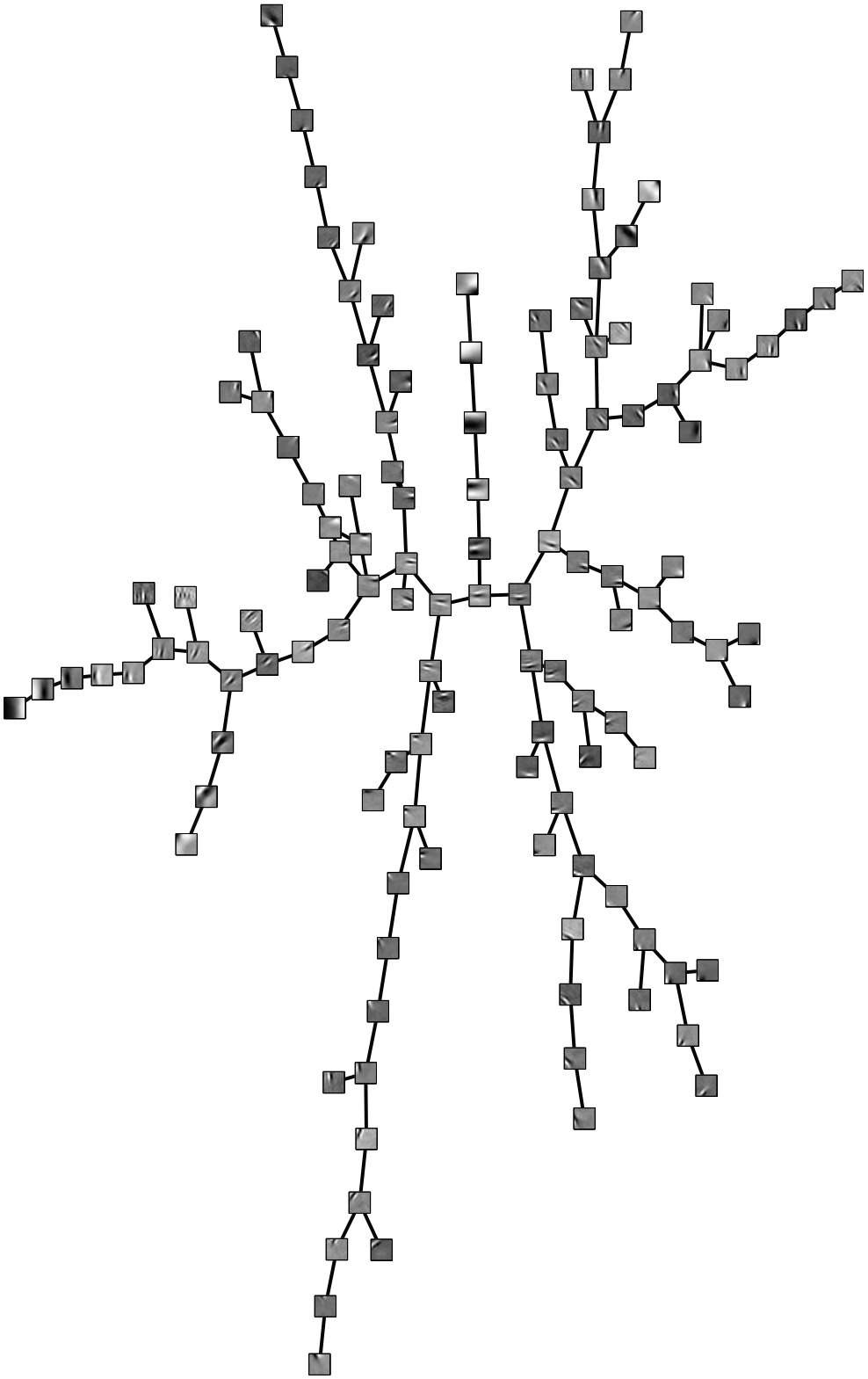


Figure 5: The learned tree graph structure and feature set. It can be seen that neighboring features on the graph have similar orientation, position and frequency. See Figure 4 for a better view of the feature details, and see text for full detail and analysis. Note that the figure is rotated CW.

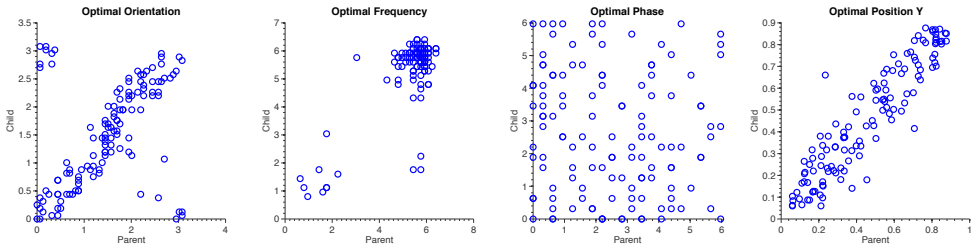


Figure 6: Correlation of optimal parameters in neighboring nodes in the tree graph. Orientation, frequency and position are highly correlated, while phase seems to be entirely uncorrelated. This property of correlation in frequency and orientation, while having no correlation in phase is related to the ubiquitous energy model of complex cells in V1. See text for further details.

Model	Log Likelihood
Marginal PCA	-162.5
Marginal ICA	-157.0
ISA	-159.4
Our Model	**-144.8**

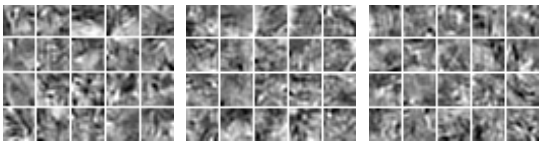


Figure 7: Left: Comparison of log likelihood values of our model with PCA, ICA and ISA. Our model gives the highest likelihood. Right: Samples taken at random from ICA, ISA and our model. Samples from our model appear to contain more long-range structure.

the tree graph structure learned along with the filters which is shown in Figure 5. It can be seen that neighboring filters (nodes) in the tree tend to have similar position, frequency and orientation. Figure 6 shows the correlation of optimal frequency, orientation and position for neighboring filters in the tree - it is obvious that all three are highly correlated. Also apparent in this figure is the fact that the optimal phase for neighboring filters has no significant correlation. It has been suggested that filters which have the same orientation, frequency and position with different phase can be related to complex cells in V1 [2, 16].

3.4 Comparison to other models

Since our model is a generalization of both ICA and ISA we use it to learn both models. In order to learn ICA we used the exact same data set, but the tree had no edges and was not learned from the data (alternatively, we could have just set $\beta = 0$). For ISA we used a forest architecture of 2 node trees, setting $\beta = 1$ for all edges (which means a spherical symmetric distribution), no tree structure was learned. Both models produce edge filters similar to what we learn (and to those in [5, 15, 6]). The ISA model produces neighboring nodes with similar frequency and orientation, but different phase, as was reported in [2]. We also compare to a simple PCA whitening transform, using the same whitening transform and marginals as in the ICA case, but setting $\mathbf{W} = \mathbf{I}$.

We compare the likelihood each model gives for a test set of natural image patches, different from the one that was used in training. There were 50,000 patches in the test set, and we calculate the mean log likelihood over the entire set. The table in Figure 7 shows the result - as can be seen, our model performs better in likelihood terms than both ICA and ISA.

Using a tree model, as opposed to more complex graphical models, allows for easy sampling from the model. Figure 7 shows 20 random samples taken from our tree model along with samples from the ICA and ISA models. Note the elongated structures (e.g. in the bottom left sample) in the samples from the tree model, and compare to patches sampled from the ICA and ISA models.

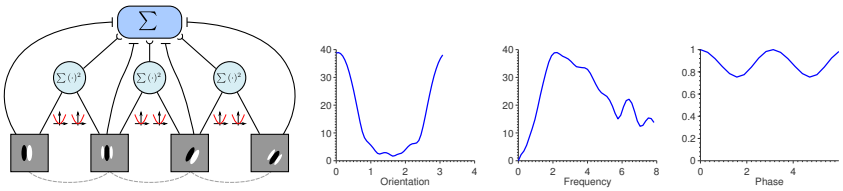


Figure 8: Left: Interpretation of the model. Given a patch, the response of all edge filters is computed ("simple cells"), then at each edge, the corresponding nodes are squared and summed to produce the response of the "complex cell" this edge represents. Both the response of complex cells and simple cells is summed to produce the likelihood of the patch. Right: Response of a "complex cell" in our model to changing phase, frequency and orientation. Response in the y-axis is the sum of squares of the two filters in this "complex cell". Note that while the cell is selective to orientation and frequency, it is rather invariant to phase.

3.5 Tree models and complex cells

One way to interpret the model is looking at the likelihood of a given patch under this model. For the case of $\beta = 1$ substituting Equation 4 into Equation 3 yields:

$$\log L(\mathbf{z}) = \sum_i \rho(\sqrt{y_i^2 + y_{pa_i}^2}) - \rho(|y_{pa_i}|) \tag{12}$$

Where $\rho(x) = \log\left(\sum_k \pi_k \mathcal{N}(x; \sigma_k^2)\right)$. This form of likelihood has an interesting similarity to models of complex cells in V1 [2, 4]. In Figure 8 we draw a simple two-layer network that computes the likelihood. The first layer applies linear filters ("simple cells") to the image patch, while the second layer sums the squared outputs of similarly oriented filters from the first layer, having different phases, which are connected in the tree ("complex cells"). Output is also dependent on the actual response of the "simple cell" layer. The likelihood here is maximized when both the response of the parent filter y_{pa_i} and the child y_i is zero, but, given that one filter has responded with a non-zero value, the likelihood is maximized when the other filter also fires (see the conditional density in Figure 2). Figure 8 also shows an example of the phase invariance which is present in the learned "complex cell" (energy of a pair of learned filters connected in the tree) - it seems that sum squared response of the shown pair of nodes is relatively invariant to the phase of the stimulus, while it is selective to both frequency and orientation - the hallmark of "complex cells". Quantifying this result with the AC/DC ratio, as is common [17] we find that around 60% percent of the edges have an AC/DC ratio which is smaller than one - meaning they would be classified as complex cells using standard methods [17].

4 Discussion

We have proposed a new model for natural image statistics which, instead of minimizing dependency between components, maximizes a simple form of dependency - tree dependency. This model is a generalization of both ICA and ISA. We suggest a method to learn such a model, including the tree structure, filter set and density model. When applied to natural image data, our model learns edge filters similar to those learned with ICA or ISA. The ordering in the tree, however, is interesting - neighboring filters in the tree tend to have similar orientation, position and frequency, but different phase. This decorrelation of phase, in conjunction with correlations in frequency and orientation are the hallmark of energy models for complex cells in V1.

Future work will include applications of the model to several image processing scenarios. We have started experimenting with application of this model to image denoising by using belief propagation for inference, and results are promising.

Acknowledgments

This work has been supported by the AMN foundation and the ISF. The authors wish to thank the anonymous reviewers for their helpful comments.

References

[1] Y. Weiss and W. Freeman, "What makes a good model of natural images?" *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[2] A. Hyvarinen and P. Hoyer, "Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, 2000.

[3] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, "On advances in statistical modeling of natural images," *J. Math. Imaging Vis.*, vol. 18, no. 1, pp. 17–33, 2003.

[4] Y. Karklin and M. Lewicki, "Emergence of complex cell properties by learning to generalize in natural scenes," *Nature*, November 2008.

[5] A. J. Bell and T. J. Sejnowski, "The independent components of natural scenes are edge filters," *Vision Research*, vol. 37, pp. 3327–3338, 1997.

[6] B. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[7] M. Bethge, "Factorial coding of natural images: how effective are linear models in removing higher-order dependencies?" vol. 23, no. 6, pp. 1253–1268, June 2006.

[8] S. Lyu and E. P. Simoncelli, "Nonlinear extraction of 'independent components' of natural images using radial Gaussianization," *Neural Computation*, vol. 21, no. 6, pp. 1485–1519, Jun 2009.

[9] A. Hyvrinen, P. Hoyer, and M. Inki, "Topographic independent component analysis: Visualizing the dependence structure," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000), Espoo, Finland*. Citeseer, 2000, pp. 591–596.

[10] F. Bach and M. Jordan, "Beyond independent components: trees and clusters," *The Journal of Machine Learning Research*, vol. 4, pp. 1205–1233, 2003.

[11] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, pp. 239–236, 2003.

[12] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[13] E. Simoncelli, "Bayesian denoising of visual images in the wavelet domain," *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pp. 291–308, 1999.

[14] A. Levin, A. Zomet, and Y. Weiss, "Learning to perceive transparency from the statistics of natural scenes," *Advances in Neural Information Processing Systems*, pp. 1271–1278, 2003.

[15] J. van Hateren, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society B: Biological Sciences*, vol. 265, no. 1394, pp. 359–366, 1998.

[16] C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," in *Digital images and human vision*. MIT Press, 1993, p. 138.

[17] K. Kording, C. Kayser, W. Einhauser, and P. Konig, "How are complex cell properties adapted to the statistics of natural stimuli?" *Journal of Neurophysiology*, vol. 91, no. 1, pp. 206–212, 2004.

2.3 From learning models of natural image patches to whole image restoration

From Learning Models of Natural Image Patches to Whole Image Restoration

Daniel Zoran
Interdisciplinary Center for Neural Computation
Hebrew University of Jerusalem
daniez@cs.huji.ac.il

Yair Weiss
School of Computer Science and Engineering
Hebrew University of Jerusalem
http://www.cs.huji.ac.il/~yweiss

Abstract

Learning good image priors is of utmost importance for the study of vision, computer vision and image processing applications. Learning priors and optimizing over whole images can lead to tremendous computational challenges. In contrast, when we work with small image patches, it is possible to learn priors and perform patch restoration very efficiently. This raises three questions - do priors that give high likelihood to the data also lead to good performance in restoration? Can we use such patch based priors to restore a full image? Can we learn better patch priors? In this work we answer these questions.

We compare the likelihood of several patch models and show that priors that give high likelihood to data perform better in patch restoration. Motivated by this result, we propose a generic framework which allows for whole image restoration using any *patch based prior for which a MAP (or approximate MAP) estimate can be calculated. We show how to derive an appropriate cost function, how to optimize it and how to use it to restore whole images. Finally, we present a* generic*, surprisingly simple Gaussian Mixture prior, learned from a set of natural images. When used with the proposed framework, this Gaussian Mixture Model outperforms all other generic prior methods for image denoising, deblurring and inpainting.*

1. Introduction

Image priors have become a popular tool for image restoration tasks. Good priors have been applied to different tasks such as image denoising [1, 2, 3, 4, 5, 6], image inpainting [6] and more [7], yielding excellent results. However, learning good priors from natural images is a daunting task - the high dimensionality of images makes learning, inference and optimization with such priors prohibitively hard. As a result, in many works [4, 5, 8] priors are learned over small image *patches*. This has the advantage of making computational tasks such as learning, inference and likelihood estimation much easier than working with whole images directly. In this paper we ask three questions: **(1)** Do patch priors that give high likelihoods yield better *patch* restoration performance? **(2)** Do patch priors that give high likelihoods yield better *image* restoration performance? **(3)** Can we learn better patch priors?

2. From Patch Likelihoods to Patch Restoration

For many patch priors a closed form of log likelihood, Bayesian Least Squares (BLS) and Maximum A-Posteriori (MAP) estimates can be easily calculated. Given that, we start with a simple question: Do priors that give high likelihood for natural image patches also produce good results in a restoration task such as denoising? Note that answering this question for priors of whole images is tremendously difficult - for many popular MRF priors, neither the log likelihood nor the MAP estimate can be calculated exactly [9].

In order to provide an answer for this question we compare several popular priors, trained over 50,000 8×8 patches randomly sampled from the training set of [10] with their DC removed. We compare the log likelihood each model gives on a set of *unseen* natural image patches (sampled from the test set of [10]) and the performance of each model in patch denoising using MAP estimates. The models we use here are: Independent pixels with learned marginals (Ind. Pixel), Multivariate Gaussian over pixels with learned covariance (MVG), Independent PCA with learned (non-Gaussian) marginals and ICA with learned marginals. For a detailed description of these models see the Supplementary Material.

The results for each of the models can be seen in Figure 1. As can be seen, the higher the likelihood a model gives for a set of patches, the better it is in denoising them when they are corrupted.

3. From Patch Likelihoods to Whole Image Restoration

Motivated by the results in Section 2, we now wish to answer the second question of this paper - do patch priors that give high likelihoods perform better in *whole image*

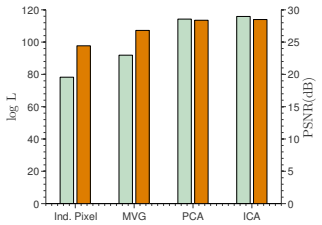


Figure 1: The likelihood of several off-the-shelf patch priors, learned from natural images, along with their *patch* denoising performance. As can be seen, patch priors that give higher likelihood to the data give better *patch* denoising performance (PSNR in dB). In this paper we show how to obtain similar performance in *whole image* restoration.

restoration? To answer this question we first need to consider the problem of how to use patch priors for whole image restoration.

To illustrate the advantages and difficulties of working with patch priors, consider Figure 2. Suppose we learn a simple patch prior from a given image (Figure 2a). To learn this prior we take all overlapping patches from the image, remove their DC component and build a histogram of all patches in the image, counting the times they appear in it. Under this prior, for example, the most likely patch would be flat (because the majority of patches in the original image are flat patches), the second most likely patch would be the tip of a diagonal edge and so on (see Figure 2b for a subset of this histogram). This prior is both easy to learn and easy to do denoising with by finding the MAP estimate given a corrupted patch. Now, suppose we are given a new, noisy *image* we wish to denoise (Figure 2c) - how should we do this using our *patch* prior?

The first, and simplest solution to this problem is to decompose the noisy image into a set of *non-overlapping* patches, denoise each patch independently by finding the MAP estimate from our prior and restore the image by placing each of the cleaned patches into its original position. This simple solution creates notorious artifacts at patch borders (see Figure 2d for an example) - if we now take a *random* patch from our newly constructed image (red patch in Figure 2d), it will be extremely unlikely under our prior (as most of the patches in the reconstructed image do not even exist in our prior, so their likelihood is 0). A more sophisticated solution may be to decompose the image into all *overlapping* patches, denoise each one independently and then average each pixel as it appears in the different patches to obtain the reconstructed image. This yields better results (see Figure 2f) but still has its problems - while we average the pixels together we create new patches in the reconstructed image which are not likely under our prior (red patch in Figure 2f). We can also take the central pixel from each of the overlapping patches but this suffers from the same problems (Figure 2e).

Going back to the motivation from Section 2, the intuition for our method is simple - suppose we take a random patch from our *reconstructed* image, we wish this patch to be *likely* under our prior. If we take another random patch from the reconstructed image, we want it also to be likely under our prior. In other words, we wish to find a reconstructed image in which *every* patch is *likely* under our prior while keeping the reconstructed image still close to the corrupted image — maximizing the *Expected Patch Log Likelihood* (EPLL) of the reconstructed image, subject to constraining it to be close to the corrupted image. Figure 2g shows the result of EPLL for the same noisy image — even though EPLL is using the *exact same* prior as in the previous methods, it produces superior results.

3.1. Framework and Optimization

3.1.1 Expected Patch Log Likelihood - EPLL

The basic idea behind our method is to try to maximize the Expected Patch Log Likelihood (EPLL) while still being close to the corrupted image in a way which is dependent on the corruption model. Given an image $\mathbf{x}$ (in vectorized form) we define the EPLL under prior p as:

$$EPLL_p(\mathbf{x}) = \sum_i \log p(\mathbf{P}_i\mathbf{x}) \tag{1}$$

Where $\mathbf{P}_i$ is a matrix which extracts the i-th patch from the image (in vectorized form) out of all *overlapping* patches, while $\log p(\mathbf{P}_i\mathbf{x})$ is the likelihood of the i-th patch under the prior p. Assuming a patch location in the image is chosen uniformly at random, EPLL is the expected log likelihood of a patch in the image (up to a multiplication by $1/N$).

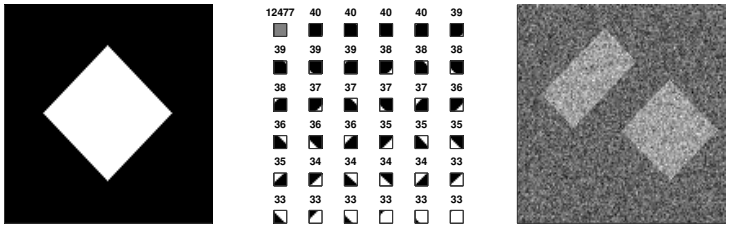
Now, assume we are given a corrupted image $\mathbf{y}$, and a model of image corruption of the form $\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2$ - We note that the corruption model we present here is quite general, as denoising, image inpainting and deblurring [7], among others, are special cases of it. We will discuss this in more detail in Section 3.1.3. The cost we propose to minimize in order to find the reconstructed image using the patch prior p is:

$$f_p(\mathbf{x}|\mathbf{y}) = \frac{\lambda}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2 - EPLL_p(\mathbf{x}) \tag{2}$$

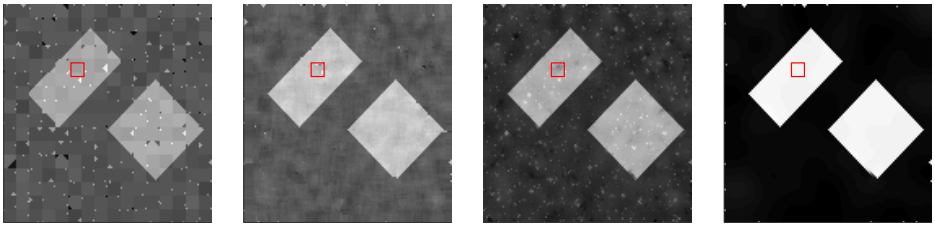
Equation 2 has the familiar form of a likelihood term and a prior term, but note that $EPLL_p(\mathbf{x})$ is *not* the log probability of a full image. Since it sums over the log probabilities of all overlapping patches, it "double counts" the log probability. Rather, it is the expected log likelihood of a randomly chosen patch in the image.

3.1.2 Optimization

Direct optimization of the cost function in Equation 2 may be very hard, depending on the prior used. We present here an



(a) Training Image (b) Prior Learned (c) Noisy Image



(d) Non Overlapping (e) Center Pixel (f) Averaged Overlapping (g) Our Method

Figure 2: The intuition behind our method. 2a A training image. 2b The prior learned from the image, only the 36 most frequent patches are shown with their corresponding count above the patch - flat patches are the most likely ones, followed by edges with 1 pixel etc. 2c A noisy image we wish to restore. 2d Restoring using non-overlapping patches - note the severe artifacts at patch borders and around the image. 2e Taking the center pixels from each patch. 2f Better results are obtained by restoring all overlapping patches, averaging the results - artifacts are still visible, and a lot of the patches in the resulting image are unlikely under the prior. 2g Result using the proposed method - note that there are very few artifacts, and most patches are very likely under our prior.

alternative optimization method called "Half Quadratic Splitting" which has been proposed recently in several relevant contexts [11, 7]. This method allows for efficient optimization of the cost. In "Half Quadratic Splitting" we introduce a set of patches $\left\{\mathbf{z^i}\right\}_1^{\mathbf{N}}$, one for each overlapping patch $\mathbf{P}_i\mathbf{x}$ in the image, yielding the following cost function:

$$c_{p,\beta}(\mathbf{x}, \left\{\mathbf{z^i}\right\}|\mathbf{y}) = \frac{\lambda}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2 + \sum_i \frac{\beta}{2}\left(\|\mathbf{P}_i\mathbf{x}-\mathbf{z}^i\|^2\right) - \log p(\mathbf{z}^i) \quad (3)$$

Note that as $\beta \to \infty$ we restrict the patches $\mathbf{P}_i\mathbf{x}$ to be equal to the auxiliary variables $\left\{\mathbf{z^i}\right\}$ and the solutions of Equation 3 and Equation 2 converge. For a *fixed* value of β, optimizing Equation 3 can be done in an iterative manner, first solving for $\mathbf{x}$ while keeping $\left\{\mathbf{z^i}\right\}$ constant, then solving for $\left\{\mathbf{z^i}\right\}$ given the newly found $\mathbf{x}$ and keeping it constant.

Optimizing Equation 3 for a fixed β value requires two steps:

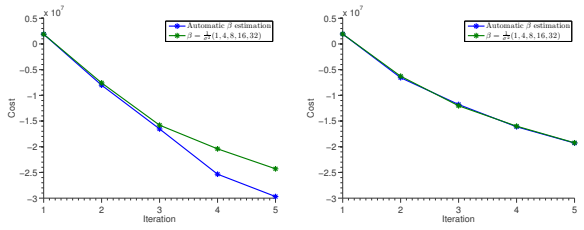
- Solving for $\mathbf{x}$ given $\left\{\mathbf{z^i}\right\}$ — This can be solved in closed form. Taking the derivative of 3 w.r.t to the vector $\mathbf{x}$, setting to 0 and solving the resulting equation yields

$$\hat{\mathbf{x}} = \left(\lambda\mathbf{A}^T\mathbf{A} + \beta\sum_j \mathbf{P}_j^T\mathbf{P}_j\right)^{-1} \left(\lambda\mathbf{A}^T\mathbf{y} + \beta\sum_j \mathbf{P}_j^T\mathbf{z}^j\right) \quad (4)$$

  Where the sum over j is for all overlapping patches in the image and all the corresponding auxiliary variables $\left\{\mathbf{z^i}\right\}$.

- Solving for $\left\{\mathbf{z^i}\right\}$ given $\mathbf{x}$ — The exact solution to this depends on the prior p in use - but for *any* prior it means solving a MAP problem of estimating the most likely patch under the prior, given the corrupted measurement $\mathbf{P}_i\mathbf{x}$ and parameter β.

We repeat the process for several iterations, typically 4 or 5 — at each iteration, solve for $\mathbf{Z}$ given $\mathbf{x}$ and solve for $\mathbf{x}$ given the new $\mathbf{Z}$, both given the current value of β. Then, increase β and continue to the next iteration. These two steps improve the cost $c_{p,\beta}$ from Equation 3, and for large β we also improve the original cost function f_p from Equation 2. We note that it is not necessary to find the *optimum* of each of the above steps, any approximate method (such as an approximate MAP estimation procedure) which still



(a) MAP estimation (b) BLS estimation

Figure 3: Optimization of the cost function from Equation 2, using the proposed optimization method. The results shown are for a simple denoising experiment, once with automatic estimation of β values, and once with a fixed schedule. In Sub-Figure 3a the MAP estimate for the patches was used. In Sub-Figure 3b the BLS estimate of the patches was used - it can be seen that even though we don't use the MAP in this experiment, the cost still decreases.

improves the cost of each sub-problem will still optimize the original cost function (albeit at different rates, depending on the exact setting).

The choice of β values is an open question. We use two approaches - the first is optimizing the values on a set of training images (by hand, or by brute force). The second option, which is relevant in denoising, is to try to estimate β from the current image estimate at every step — this is done by estimating the amount of noise σ present in the image $\hat{\mathbf{x}}$, and setting $\beta = \frac{1}{\sigma^2}$. We use the noise estimation method of [12].

Figure 3 shows a small experiment in which we verify that the original cost function in Equation 2 indeed decreases as a function of iterations as β grows. At the first experiment, β was estimated from the current image estimate and in the second, we used a fixed β schedule. The prior used was the ICA prior for which the likelihood is easily calculated. Even though the half quadratic splitting is only guaranteed to monotonically decrease the cost for infinite β values, we show experimentally that the cost decreases for different schedules of β where the schedule effects mostly the convergence speed. In addition, even when using BLS (instead of MAP), the cost still decreases - this shows that we don't need the find the optimum at each step, just to improve the cost of each sub-problem.

In summary, we note three attractive properties of our general algorithm. First, it can use *any* patch based prior and second, its run time is only 4–5 times the run time of restoring with simple patch averaging (depending on the number of iterations). Finally, perhaps the most important one is that this framework does not require learning a model $P(\mathbf{x})$ where $\mathbf{x}$ is a natural image, rather, learning needs only to concentrate on modeling the probability of image patches.

3.1.3 Denoising, Deblurring and Inpainting

In denoising, we have additive white Gaussian noise corrupting the image, so we set the matrix $\mathbf{A}$ from Equation 4 to be the identity matrix, and set λ to be related to the standard deviation of the noise ($\approx \frac{1}{\sigma^2}$). This means that the solution for $\mathbf{x}$ at each optimization step is just a weighted average between the noisy image $\mathbf{y}$ and the average of pixels as they appear in the auxiliary overlapping patches. The solution for $\mathbf{Z}$ is just a MAP estimate with prior p and noise level $\sqrt{\frac{1}{\beta}}$. If we initialize $\mathbf{x}$ with the noisy image $\mathbf{y}$, setting $\lambda = 0$ and $\beta = \frac{1}{\sigma^2}$ results in simple patch averaging when iterating a single step. The big difference, however, is that in our method, because we iterate the solution and $\lambda \neq 0$, at each iteration we use the current estimated image, averaging it with the noisy one and obtaining a *new* set of $\mathbf{Z}$ patches, solving for them and then obtaining a new estimate for $\mathbf{x}$, repeating the process, while increasing β. For image deblurring (non-blind) $\mathbf{A}$ is a convolution matrix with a known kernel.

Image inpainting is similar, $\mathbf{A}$ is a diagonal matrix with zeros for all the missing pixels. Basically, this can be thought of as "denoising" with a per pixel noise level - infinite noise for missing pixels and zero noise for all other pixels. See Supplementary Material for some examples of inpainting.

3.2. Related Methods

Several existing methods are closely related, but are fundamentally different from the proposed framework. The first related method is the Fields of Experts (FoE) framework by Roth and Black [6]. In FoE, a Markov Random Field (MRF) whose filters are trained by approximately maximizing the likelihood of the training *images* is learned. Due to the intractability of the partition function, learning with this model is extremely hard and is performed using contrastive divergence. Inference in FoE is actually a special case of our proposed method - while the learning is vastly different, in FoE the inference procedure is equivalent to optimizing Equation 2 with an independent prior (such as ICA), whose filters were learned before hand. A common approximation to learning MRFs is to approximate the log probability of an image as a sum of local marginal or conditional probabilities as in the method of composite likelihood [13] or directed models of images [14]. In contrast, we do not attempt to approximate the global log probability and argue that modeling the local patch marginals is sufficient for image restoration. This points to one of the advantages of our method - learning a patch prior is much easier than learning a MRF. As a result, we can learn a much richer patch prior easily and incorporate it into our framework - as we show later.

Another closely related method is KSVD [3] - in KSVD, one learns a patch based dictionary which attempts to maximize the sparsity of resulting coefficients. This dictionary

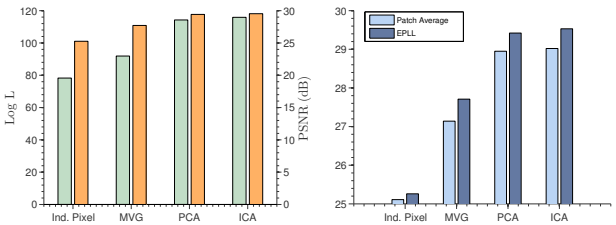
can be learned either from a set of natural image patches (generic, or global as it is sometimes called) or the noisy image itself (image based). Using this dictionary, all overlapping patches of the image are denoised independently and then averaged to obtain a new reconstructed image. This process is repeated for several iterations using this new estimated image. Learning the dictionary in KSVD is different than learning a patch prior because it may be performed as part of the optimization process (unless the dictionary is learned beforehand from natural images), but the optimization in KSVD can be seen as a special case of our method - when the prior is a sparse prior, our cost function and KSVD's are the same. We note again, however, that our framework allows for much richer priors which can be learned beforehand over patches - as we will see later on, this boasts some tremendous benefits.

A whole family of patch based methods [2, 5, 8] use the noisy image itself in order to denoise. These "non-local" methods look for similarities within the noisy image itself and operate on these similar patches together. BM3D [5] groups together similar patches into "blocks", transforms them into wavelet coefficients (in all 3 dimensions), thresholds and transforms backs, using all the estimates together. Mairal et al. [8] which is currently state-of-the-art take a similar approach to this, but instead of transforming the patches via a wavelet transform, sparse coding using a learned dictionary is used, where each block is constrained to use the same dictionary elements. One thing which is common to all of the above non-local methods, and indeed almost all patch based methods, is that they all average the clean patches together to form the final estimate of the image. As we have seen in Section 3, this may not be the optimal thing to do.

3.3. Patch Likelihoods and the EPLL Framework

We have seen that the EPLL cost function (Equation 2) depends on the likelihood of patches. Going back to the priors from Section 2 we now ask - do better priors (in the likelihood sense) also lead to better *whole* image denoising with the proposed EPLL framework? Figure 4 shows the average PSNR obtained with 5 different images from the Berkeley training set, corrupted with Gaussian noise at $\sigma = 25$ and denoised using each of the priors in section 2. We compare the result obtained using simple patch averaging (PA) and our proposed EPLL framework. It can be seen that indeed - better likelihood on patches leads to better denoising both on independent patches (Figure 1) and whole images (Figure 4). Additionally, it can be seen that EPLL improves denoising results significantly when compared to simple patch averaging (Figure 4).

These results motivate the question: Can we find a better prior for image patches?



(a) (b)

Figure 4: **(a)** Whole image denoising with the proposed framework with all the priors discussed in Section 2. It can be seen that better priors (in the likelihood sense) lead to better denoising performance on *whole* images, left bar is log L, right bar is PSNR. **(b)** Note how the EPLL framework improves performance significantly when compared to simple patch averaging (PA)

4. Can We Learn Better Patch Priors?

In addition to the priors discussed in Section 2 we introduce a new, simple yet surprisingly rich prior.

4.1. Learning and Inference with a Gaussian Mixture Prior

We learn a finite Gaussian mixture model over the pixels of natural image patches. Many popular image priors can be seen as special cases of a GMM (e.g. [9, 1, 14]) but they typically constrain the means and covariance matrices during learning. In contrast, we do not constrain the model in any way — we learn the means, full covariance matrices and mixing weights, over all pixels. Learning is easily performed using the Expectation Maximization algorithm (EM). With this model, calculating the log likelihood of a given patch is trivial:

$$\log p(\mathbf{x}) = \log \left(\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k) \right) \tag{5}$$

Where π_k are the mixing weights for each of the mixture component and μ_k and Σ_k are the corresponding mean and covariance matrix.

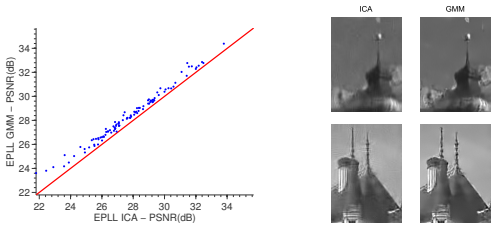
Given a noisy patch $\mathbf{y}$, the BLS estimate can be calculated in closed form (as the posterior is just another Gaussian mixture) [1]. The MAP estimate, however, can not be calculated in closed form. To tackle this we use the following approximate MAP estimation procedure:

1. Given noisy patch $\mathbf{y}$ we calculate the conditional mixing weights $\pi'_k = P(k|\mathbf{y})$.
2. We choose the component which has the highest conditional mixing weight $k_{max} = \max_k \pi'_k$.
3. The MAP estimate $\hat{\mathbf{x}}$ is then a Wiener filter solution for the k_{max}-th component:

$$\hat{\mathbf{x}} = \left(\mathbf{\Sigma}_{k_{max}} + \sigma^2 \mathbf{I}\right)^{-1} \left(\mathbf{\Sigma}_{k_{max}} \mathbf{y} + \sigma^2 \mathbf{I} \mu_{k_{max}}\right)$$

		Patch Restoration		Image Restoration	
Model	Log L	BLS	MAP	PA	EPLL
Ind. Pixel	78.26	25.54	24.43	25.11	25.26
MVG	91.89	26.81	26.81	27.14	27.71
PCA	114.24	28.01	28.38	28.95	29.42
ICA	115.86	28.11	28.49	29.02	29.53
GMM	**164.52**	**30.26**	**30.29**	**29.59**	**29.85**

Table 1: GMM model performance in log likelihood (Log L), patch denoising (BLS and MAP) and image denoising (Patch Average (PA) and EPLL, the proposed framework) - note that the performance is better than all priors in all measures. The patches, noisy patches, images and noisy images are the same as in Figure 1 and Figure 4. All values are in PSNR (dB) apart from the log likelihood.



(a) PSNR Comparison (b) Detail Shots

Figure 5: Comparison of the performance of the ICA prior to the high likelihood GMM prior using EPLL and noise level $\sigma = 25$. 5a depicts a scatter plot of PSNR values obtained when denoising 68 images from[10]. Note the superior performance of the GMM prior when compared to ICA on all images. 5b depicts a detail shot from two of the images - note the high visual quality of the GMM prior result. The details are best seen when zoomed in on a computer screen.

This is actually one iteration of the "hard version" of the EM algorithm for finding the modes of a Gaussian mixture [15].

4.2. Comparison

We learn the proposed GMM model from a set of 2×10^6 patches, sampled from [10] with their DC removed. The model is with learned 200 mixture components with zero means and full covariance matrices. We also trained GMMs with unconstrained means and found that all the means were very close to zero. As mentioned above, learning was performed using EM. Training with the above training set takes around 30h with unoptimized MATLAB code[1]. Denoising a patch with this model is performed using the approximate MAP procedure described in 4.1.

Having learned this GMM prior, we can now compare its performance both in likelihood and denoising with the priors we have discussed thus far in Section 1 on the same dataset of *unseen* patches. Table 1 shows the results obtained with the GMM prior - as can be seen, this prior is superior in likelihood, patch denoising and whole image denoising to all other priors we discussed thus far.

In Figure 5a we show a scatter plot of PSNR values obtained with ICA and the GMM model using EPLL at noise level $\sigma = 25$ on 68 images from the Berkeley test set. Note that the high likelihood GMM model is superior to ICA in denoising, on all tested images. Figure 5b shows details from images in the test-set, note the high visual quality of the GMM model when compared to the ICA result.

Why does this model work so well? One way to understand it is to recall that in a zero-mean Gaussian mixture model, every sample x is well approximated by the top m eigenvectors of the covariance matrix of the mixture component that it belongs to. If we consider the set of all m eigenvectors of all mixtures as a "dictionary" then every sample is approximated by a sparse combination of these dictionary elements. Since there are 200 mixture components, only (1/200) dictionary elements are "active" for each x so this is a very sparse representation. But unlike other models that assume sparsity (e.g. ICA and Sparse Coding), the active set is extremely constrained — only dictionary elements that correspond to the same component are allowed to be jointly active. We have recently learned that this "dual" interpretation of a GMM was independently given by [16] for the case of image-specific GMMs.

What do these dictionary elements model? Figure 6 depicts the eigenvectors of the 5 randomly selected mixture components from the learned model. Note that these have rich structures - while some resembles PCA eigenvectors, some depict forms of occlusions, modeling texture boundaries and edges. These are very different from the Gabor filters usually learned by sparse coding and similar models. It would seem that these structures contribute much to the expressive power of the model.

4.3. Comparison to State-Of-The-Art Methods

We compare the performance of EPLL with the proposed GMM prior with leading image restoration methods - both generic and image based. All the experiments were conducted on 68 images from the test set of the Berkeley Segmentation Database [10]. All experiments were conducted using the same noisy realization of the images. In all experiments we set $\lambda = \frac{N}{\sigma^2}$, where N is the number of pixels in each patch. We used a patch size of 8×8 in all experiments. For the GMM prior, we optimized (by hand) the values for β on the 5 images from the Berkeley training set - these were set to $\beta = \frac{1}{\sigma^2} \cdot [1, 4, 8, 16, 32, 64]$. Running times on a Quad Core Q6600 processor are around 300s per image with unoptimized MATLAB code.

[1] Downloaded from: http://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model

σ	KSVDG	FoE	GMM-EPLL
15	30.67	30.18	**31.21**
25	28.28	27.77	**28.71**
50	25.18	23.29	**25.72**
100	22.39	16.68	**23.19**

(a) Generic Priors

σ	KSVD	BM3D	LLSC	GMM-EPLL
15	30.59	30.87	**31.27**	31.21
25	28.20	28.57	**28.70**	**28.71**
50	25.15	25.63	**25.73**	**25.72**
100	22.40	**23.25**	23.15	23.19

(b) Image Based Methods

Table 2: Summary of denoising experiments results. Our method is clearly state-of-the-art when compared to generic priors, and is competitive with image based method such as BM3D and LLSC which are state-of-the-art in image denoising.

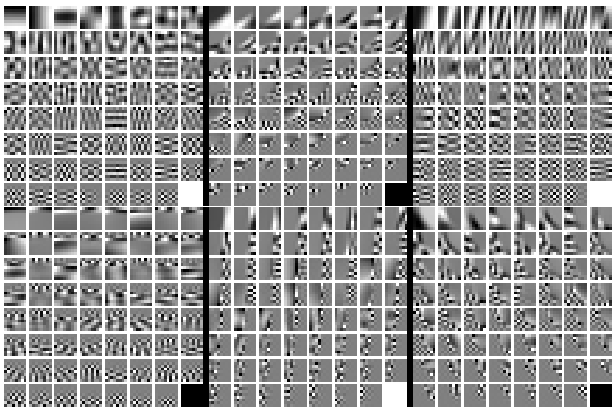
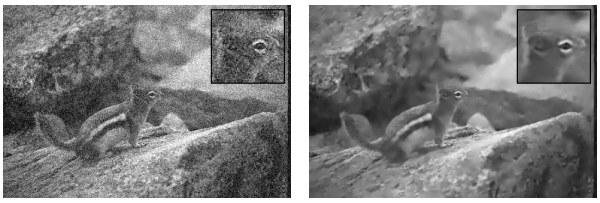
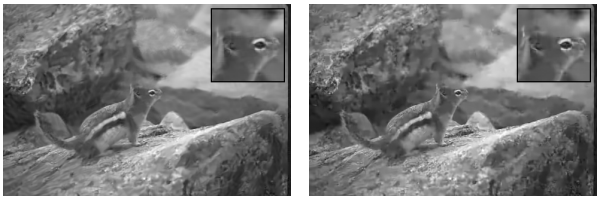


Figure 6: Eigenvectors of 6 randomly selected covariance matrices from the learned GMM model, sorted by eigenvalue from largest to smallest. Note the richness of the structures - some of the eigenvectors look like PCA components, while others model texture boundaries, edges and other structures at different orientations.



(a) Noisy Image - PSNR: 20.17 (b) KSVD - PSNR: 28.72



(c) LLSC - PSNR: 29.30 (d) EPLL GMM - PSNR: 29.39

Figure 7: Examples of denoising using EPLL-GMM compared with state-of-the-art denoising methods - KSVD [3] and LLSC [8]. Note how detail is much better preserved in our method when compared to KSVD. Also note the similarity in performance with our method when compared to LLSC, even though LLSC learn from the noisy image. See supplementary material for more examples.

4.3.1 Generic Priors

We compare the performance of EPLL and the GMM prior in image denoising with leading *generic* methods - Fields of Experts [6] and KSVD [3] trained on natural image patches (KSVDG). The summary of results may be seen in Table 2a - it is clear that our method outperforms the current state-of-the-art generic methods.

4.3.2 Image Based Priors

We now compare the performance of our method (EPLL+GMM) to image specific methods - which learn from the noisy image itself. We compare to KSVD, BM3D [5] and LLSC [8] which are currently the state-of-the-art in image denoising. The summary of results may be seen in Table 2b. As can be seen, our method is highly competitive with these state-of-the-art method, even though it is generic. Some examples of the results may be seen in Figure 7.

4.3.3 Image Deblurring

While image specific priors give excellent performance in denoising, since the degradation of different patches in the same image can be "averaged out", this is certainly not the case for all image restoration tasks, and for such tasks a generic prior is needed. An example of such a task is image deblurring. We convolved 68 images from the Berkeley database (same as above) with the blur kernels supplied with the code of [7]. We then added 1% white Gaussian noise to the images, and attempted reconstruction using the code by [7] and our EPLL framework with GMM prior. Results are superior both in PSNR and quality of the output, as can be seen in Figure 8.

(a) Blurred (b) Krishnan et al. (c) EPLL GMM

	Krishnan et al.	EPLL-GMM
Kernel 1 17×17	25.84	27.17
Kernel 2 19×19	26.38	27.70

Figure 8: Deblurring experiments

5. Discussion

Patch based models are easier to learn and to work with than whole image models. We have shown that patch models which give high likelihood values for patches sampled from natural images perform better in patch and image restoration tasks. Given these results, we have proposed a framework which allows the use of patch models for whole image restoration, motivated by the idea that patches in the restored image should be likely under the prior. We have shown that this framework improves the results of whole image restoration considerably when compared to simple patch averaging, used by most present day methods. Finally, we have proposed a new, simple yet rich Gaussian Mixture prior which performs surprisingly well on image denoising, deblurring and inpainting.

While we have demonstrated our framework using only a few priors, one of its greater strengths is the fact that it can serve as a "plug-in" system - it can work with any existing patch restoration method. Considering the fact that both BM3D and LLSC are patch based methods which use simple patch averaging, it would be interesting to see how would these methods benefit from the proposed framework.

Finally, perhaps the most surprising result of this work, and the direction in which much is left to be explored, is the stellar performance of the GMM model. The GMM model used here is extremely naive - a simple mixture of Gaussians with full covariance matrices. Given the fact that Gaussian Mixtures are an extremely studied area, incorporating more sophisticated machinery into the learning and the representation of this model holds much promise - and this is our current line of research.

Acknowledgments

The authors wish to thank Anat Levin for helpful discussions.

References

[1] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[2] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 60–65, IEEE, 2005.

[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[4] Y. Hel-Or and D. Shaked, "A discriminative approach for wavelet denoising," *IEEE Transactions on Image Processing*, vol. 17, no. 4, p. 443, 2008.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *SPIE Electronic Imaging*, 2008.

[6] S. Roth and M. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[7] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in Neural Information Processing Systems 22*, pp. 1033–1041, 2009.

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2272–2279, IEEE, 2010.

[9] Y. Weiss and W. Freeman, "What makes a good model of natural images?," *CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001.

[11] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *Image Processing, IEEE Transactions on*, vol. 4, no. 7, pp. 932–946, 2002.

[12] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2209–2216, Citeseer, 2009.

[13] B. Lindsay, "Composite likelihood methods," *Contemporary Mathematics*, vol. 80, no. 1, pp. 221–39, 1988.

[14] J. Domke, A. Karapurkar, and Y. Aloimonos, "Who killed the directed model?," in *CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[15] M. Carreira-Perpiñán, "Mode-finding for mixtures of Gaussian distributions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1318–1323, 2002.

[16] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *CoRR*, vol. abs/1006.3056, 2010.

2.4 Natural images, Gaussian mixtures and dead leaves

Natural Images, Gaussian Mixtures and Dead Leaves

Daniel Zoran
Interdisciplinary Center for Neural Computation
Hebrew University of Jerusalem
Israel
http://www.cs.huji.ac.il/ daniez

Yair Weiss
School of Computer Science and Engineering
Hebrew University of Jerusalem
Israel
yweiss@cs.huji.ac.il

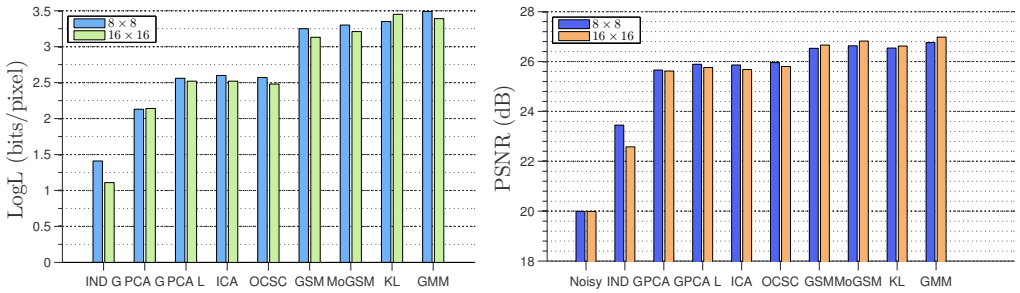
Abstract

Simple Gaussian Mixture Models (GMMs) learned from pixels of natural image patches have been recently shown to be surprisingly strong performers in modeling the statistics of natural images. Here we provide an in depth analysis of this simple yet rich model. We show that such a GMM model is able to compete with even the most successful models of natural images in log likelihood scores, denoising performance and sample quality. We provide an analysis of what such a model learns from natural images as a function of number of mixture components — including covariance structure, contrast variation and intricate structures such as textures, boundaries and more. Finally, we show that the salient properties of the GMM learned from natural images can be derived from a simplified Dead Leaves model which explicitly models occlusion, explaining its surprising success relative to other models.

1 GMMs and natural image statistics models

Many models for the statistics of natural image patches have been suggested in recent years. Finding good models for natural images is important to many different research areas — computer vision, biological vision and neuroscience among others. Recently, there has been a growing interest in comparing different aspects of models for natural images such as log-likelihood and multi-information reduction performance, and much progress has been achieved [1, 2, 3, 4, 5, 6]. Out of these results there is one which is particularly interesting: simple, unconstrained Gaussian Mixture Models (GMMs) with a relatively small number of mixture components learned from image patches are extraordinarily good in modeling image statistics [6, 4]. This is a surprising result due to the simplicity of GMMs and their ubiquity. Another surprising aspect of this result is that many of the current models may be thought of as GMMs with an exponential or infinite number of components, having different constraints on the covariance structure of the mixture components.

In this work we study the nature of GMMs learned from natural image patches. We start with a thorough comparison to some popular and cutting edge image models. We show that indeed, GMMs are excellent performers in modeling natural image patches. We then analyze what properties of natural images these GMMs capture, their dependence on the number of components in the mixture and their relation to the structure of the world around us. Finally, we show that the learned GMM suggests a strong connection between natural image statistics and a simple variant of the dead leaves model [7, 8], explicitly modeling occlusions and explaining some of the success of GMMs in modeling natural images.



(a) Log Likelihood (b) Denoising

Figure 1: **(a)** Log likelihood comparison - note how the GMM is able to outperform (or equal) all other models despite its simplicity. **(b)** Denoising performance comparison - the GMM outperforms all other models here as well, and denoising performance is more or less consistent with likelihood performance. See text for more details.

2 Natural image statistics models - a comparison

As a motivation for this work, we start by rigorously comparing current models for natural images with GMMs. While some comparisons have been reported before with a limited number of components in the GMM [6], we want to compare to state-of-the-art models also varying the number of components systematically.

Each model was trained on 8×8 or 16×16 patches randomly sampled from the Berkeley Segmentation Database *training* images (a data set of millions of patches). The DC component of all patches was removed, and we discard it in all calculations. In all experiments, evaluation was done on the same, *unseen* test set of a 1000 patches sampled from the Berkeley *test* images. We removed patches having standard deviation below 0.002 (intensity values are between 0 and 1) as these are totally flat patches due to saturation and contain no structure (only 8 patches were removed from the test set). We do not perform any further preprocessing. The models we compare are: White Gaussian Noise (**Ind. G**), PCA/Gaussian (**PCA G**), PCA/Laplace (**PCA L**), ICA (**ICA**) [9, 10, 11], $2\times$Overcomplete sparse coding ($2\times$**OCSC**) [9], Gaussian Scale Mixture (**GSM**), Mixture of Gaussian Scale Mixture (**MoGSM**) [6], Karklin and Lewicki (**KL**) [12] and the **GMM** (with 200 components).

We compare the models using three criteria - log likelihood on unseen data, denoising results on unseen data and visual quality of samples from each model. The complete details of training, testing and comparisons may be found in the supplementary material of this paper - we encourage the reader to read these details. All models and code are available online at: www.cs.huji.ac.il/∼daniez

Log likelihood The first experiment we conduct is a log likelihood comparison. For most of the models above, a closed form calculation of the likelihood is possible, but for the $2\times$OCSC and KL models, we resort to Hamiltonian Importance Sampling (HAIS) [13]. HAIS allows us to estimate likelihoods for these models accurately, and we have verified that the approximation given by HAIS is relatively accurate in cases where exact calculations are feasible (see supplementary material for details). The results of the experiment may be seen in Figure 1a. There are several interesting results in this figure. First, the important thing to note here is that GMMs outperforms all of the models and is similar in performance to Karklin and Lewicki. In [6] a GMM with far less components (2-5) has been compared to some other models (notably Restricted Boltzman Machines which the GMM outperforms, and MoGSMs which slightly outperform the GMMs in this work). Second, ICA with its learned Gabor like filters [10] gives a very minor improvement when compared to PCA filters with the same marginals. This has been noted before in [1]. Finally, overcomplete sparse coding is actually a bit *worse* than complete sparse coding - while this is counter intuitive, this result has been reported before as well [14, 2].

Denoising We compare the denoising performance of the different models. We added independent white Gaussian noise with known standard deviation $\sigma_n = 25/255$ to each of the patches in the test set $\mathbf{x}$. We then calculate the MAP estimate $\hat{\mathbf{x}}$ of each model given the noisy patch. This can

be done in closed form for some of the models, and for those models where the MAP estimate does not have a closed form, we resort to numerical approximation (see supplementary material for more details). The performance of each model was measured using Peak Signal to Noise Ratio (PSNR): $\text{PSNR} = \log_{10}\left(\frac{1}{\|\mathbf{x}-\hat{\mathbf{x}}\|^2}\right)$. Results can be seen in Figure 1b. Again, the GMM performs extraordinarily well, outperforming all other models. As can be seen, results are consistent with the log likelihood experiment - models with better likelihood tend to perform better in denoising [4].

Sample Quality As opposed to log likelihood and denoising, generating samples from all the models compared here is easy. While it is more of a subjective measure, the visual quality of samples may be an indicator to how well interesting structures are captured by a model. Figure 2 depicts 16×16 samples from a subset of the models compared here. Note that the GMM samples capture a lot of the structure of natural images such as edges and textures, visible on the far right of the figure. The Karklin and Lewicki model produces rather structured patches as well. GSM seems to capture the contrast variation of images, but the patches themselves have very little structure (similar results obtained with MoGSM, not shown). PCA lacks any meaningful structure, other than $1/f$ power spectrum.

As can be seen in the results we have just presented, the GMM is a very strong performer in modeling natural image patches. While we are not claiming Gaussian Mixtures are the *best* models for natural images, we do think this is an interesting result, and as we shall see later, it relates intimately to the structure of natural images.

3 Analysis of results

So far we have seen that despite their simplicity, GMMs are very capable models for natural images. We now ask - what do these models learn about natural images, and how does this affect their performance?

3.1 How many mixture components do we need?

While we try to learn our GMMs with as few a priori assumptions as possible, we do need to set one important parameter - the number of components in the mixture. As noted above, many of the current models of natural images can be written in the form of GMMs with an exponential or infinite number of components and different kinds of constraints on the covariance structure. Given this, it is quite surprising that a GMM with a relatively small number of component (as above) is able to compete with these models. Here we again evaluate the GMM as in the previous section but now systematically vary the number of components and the size of the image patch. Results for the 16×16 model are shown in figure 3, see supplementary material for other patch sizes.

As can be seen, moving from one component to two already gives a tremendous boost in performance, already outperforming ICA but still not enough to outperform GSM, which is outperformed at around 16 components. As we add more and more components to the mixture performance increases, but seems to be converging to some upper bound (which is not reached here, see supplementary material for smaller patch sizes where it is reached). This shows that a small number of components is indeed

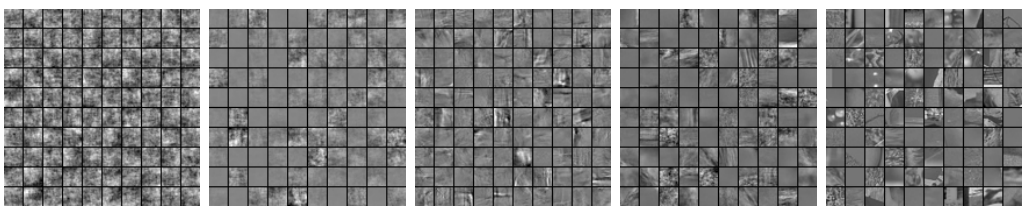
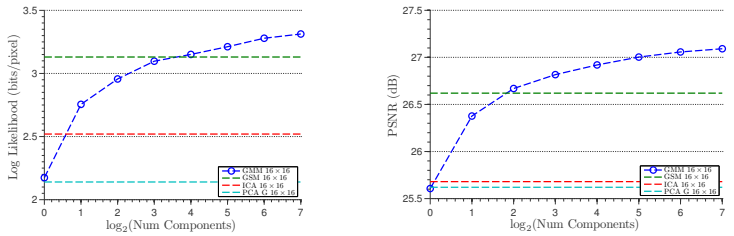


Figure 2: Samples generated from some of the models compared in this work. PCA G produces no structure other than $1/f$ power spectrum. GSM capture the contrast variation of image patches nicely, but the patches themselves have no structure. The GMM and KL models produce quite structured patches - compare with the natural image samples on the right.



(a) Log Likelihood (b) Denoising

Figure 3: **(a)** Log likelihood of GMMs trained on natural image patches, as a function of the number of components in the mixture. Models of 16×16 were trained on a training set. Likelihood was calculated on an unseen test set of patches. Already at 2 components the GMM outperforms ICA and at 16 components it outperforms the 16 component GSM model. Likelihood continues to improve as we add more components. See supplementary material for other patch sizes. **(b)** Denoising performance as a function of number of components - performance behave qualitatively the same as likelihood.

sufficient to achieve good performance and begs the questions - what do the first few components learn that gives this boost in performance? what happens when we add more components to the mixture, further improving performance? Before we answer these questions, we will shortly discuss what are the properties of GMMs which we need to examine to gain this understanding.

3.2 GMMs as generative models

In order to gain a better understanding of GMMs it will be useful to think of them from a generative perspective. The process of generating a sample from a GMM is a two step procedure; a non-linear one, and a linear one. We pick one of the mixture components - the chances for the k-th component to be picked are its mixing weight π_k. Having picked the k-th component, we now sample N *independent* Gaussian variables with zero mean and unit variance, where N is the number of pixels in a patch (minus one for the DC component). We arrange these coefficients into a vector $\mathbf{z}$. From the covariance matrix of the k-th component we calculate the eigenvector matrix $\mathbf{V}_k$ and eigenvalue matrix $\mathbf{D}_k$. Then, the new sample $\mathbf{x}$ is:

$$\mathbf{x} = \mathbf{V_k} \mathbf{D}_k^{0.5} \mathbf{z}$$

This tells us that we can think of each covariance matrix in the mixture as a dictionary with N elements. The dictionary elements are the "directions" each eigenvector in patch space points to, and each of those is scaled by the corresponding eigenvalue. These are linearly mixed to form our patch. In other words, to gain a better understanding of what each mixture component is capturing, we need to look at the *eigenvectors* and *eigenvalues* of its corresponding covariance matrix.

3.3 Contrast

Figure 4 shows the eigenvectors and eigenvalues of the covariance matrices of a 2 component mixture - as can be seen, the eigenvectors of both mixture components are very similar and they differ only in their eigenvalue spectrum. The eigenvalue spectrum, on the other hand, is very similar in *shape* but differs by a multiplicative constant (note the log scale). This behavior remains the same as we add more and more components to the mixture — up to around 8-10 components (depending on the patch size, not shown here) we get more components with similar eigenvector structure but different eigenvalue distributions.

Modeling a patch as a mixture with the same eigenvectors but eigenvalues differing by a scalar multiplier is in fact equivalent to saying that each patch is the product of a scalar z and a multivariate Gaussian. This is exactly the Gaussian Scale Mixture model we compared to earlier! As can be seen, 8–10 components are already enough to equal the performance of the 16 component GSM. This means that what the first few components of the mixture capture is the contrast variability of natural image patches. This also means that factorial models like ICA have no hope in capturing this as contrast is a global scaling of all coefficients together (something which is highly unlikely under factorial models).

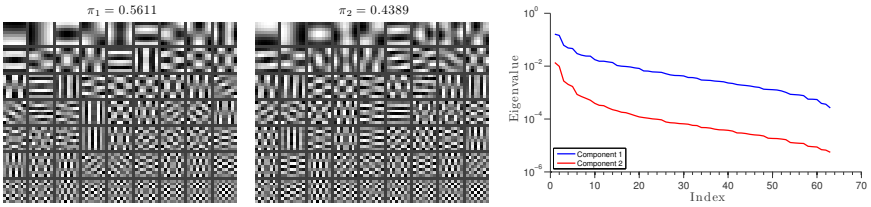


Figure 4: Eigenvectors and eigenvalues of covariance matrices in a 2 component GMM trained on natural images. Eigenvectors are sorted according to decreasing eigenvalue order, top left is the largest eigenvalue. Note that the two components have approximately the same eigenvectors (up to sign, and both resembling the Fourier basis) but different eigenvalue spectra. The eigenvalues mostly differ by a scalar multiplication (note the log scale), hinting that this is, in fact, approximately a GSM (see text for details).

3.4 Textures and boundaries

We have seen that the first components in the GMM capture the contrast variation of natural images, but as we saw in Figure 3, likelihood continues to improve as we add more components, so we ask: what do these extra components capture?

As we add more components to the mixture, we start revealing more specialized components which capture different properties of natural images. Sorting the components by their mixing weights (where the most likely ones are first), we observe that the first few tens of components are predominantly Fourier like components, similar to what we have seen thus far, with varying eigenvalue spectra. These capture *textures* at different scales and orientations. Figure 5 depicts two of these texture components - note how their eigenvector structure is similar, but samples sampled from each of them reveal that they capture different textures due to different eigenvalue spectra.

A more interesting family of components can be found in the mixture as we look into more rare components. These components model *boundaries* of objects or textures — their eigenvectors are structured such that most of the variability is on one side of an edge crossing the patch. These edges come at different orientations, shifts and contrasts. Figure 5 depicts some of these components at different orientations, along with two flat texture components for comparison. As can be seen, we obtain a Fourier like structure which is concentrated on one side of the patch. Sampling from the Gaussian associated with each mixture component (bottom row) reveals what each component actually captures - patches with different textures on each side of an edge.

To see how these components relate to actual structure in natural images we perform the following experiment. We take an unseen natural image, and for each patch in the image we calculate the most likely component from the learned mixture. Figure 6 depicts those patches assigned to each of the five components in Figure 5, where we show only non-overlapping patches for clarity (there are many more patches assigned to each component in the image). The colors correspond to each of the components in Figure 5. Note how the boundary components capture different orientations, and prefer mostly borders with a specific ordering (top to bottom edge, and not vice versa for example), while texture components tend to stay within object boundaries. The sources for these phenomena will be discussed in the next section.

4 The "mini" dead leaves model

4.1 Dead leaves models

We now show that many of the properties of natural scenes that were captured by the GMM model can be derived from a variant of the dead leaves model [15]. In the original dead leaves model, two dimensional textured surfaces (which are sometimes called "objects" or "leaves") are sampled from a shape and size distribution and then placed on the image plane at random positions, *occluding* one another to produce an image. With a good choice of parameters, such a model creates images which

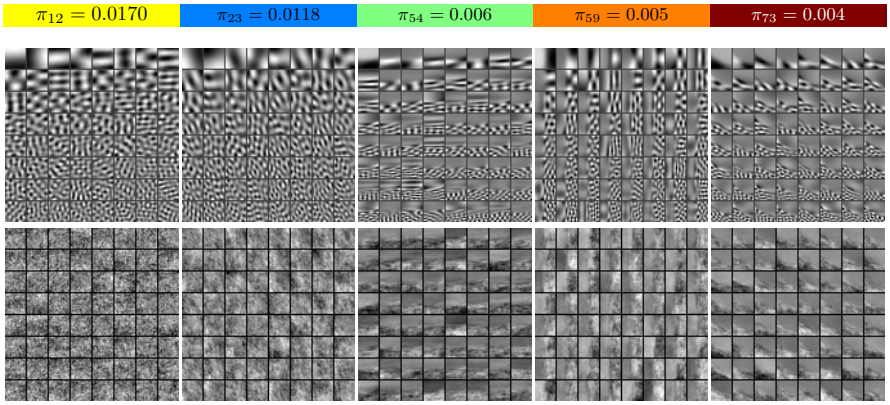


Figure 5: Leading eigenvectors (top row) and samples (bottom row) from 5 different components from a 16×16 GMM. From left to right: components 12 and 23, having a similar Fourier like eigenvector structure, but different eigenvalue spectra, notable by different texture generated from each component. Three different "boundary" like component: note how the eigenvector structure has a Fourier like structure which is concentrated only on side of the patch, depicting an edge structure. These come in different orientations, shifts and contrasts in the mixture. The color markings are in reference to Figure 6.

share many properties with natural images such as scale invariance, heavy tailed filter responses and bow-tie distributions for conditional pair-wise filter responses [16, 17, 8]. A recent work by Pitkow [8] provides an interesting review and analysis of these properties.

4.2 Mini dead leaves

We propose here a simple model derived from the dead leaves model which we call the "Mini Dead Leaves" model. This is a patch based version of the dead leaves model, and can be seen as an approximation of what happens when sampling small patches from an image produced by the dead leaves model.

In mini dead leaves we generate an image patch in the following manner: for each patch we randomly decide if this patch would be a "flat" patch or an "edge" patch. This is done by flipping a coin with probability p. **Flat** patches are then produced by sampling a texture from a given texture process. In this case we use a multidimensional Gaussian with some stationary texture covariance matrix which is multiplied by a scalar contrast variable. We then add to the texture a random scalar mean value, such that the final patch $\mathbf{x}$ is of the form: $\mathbf{x} = \mu + z\mathbf{t}$ where $\mu \sim \mathcal{N}(0, 1)$ is a scalar, $\mathbf{t} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is a

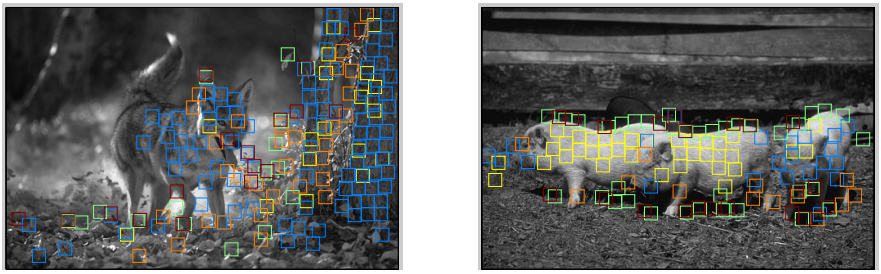


Figure 6: Components assignment on natural images taken from the Berkeley *test* images. For each patch in the image the most likely component from the mixture was calculated - presented here are patches which were assigned to one of the components in Figure 5. Assignment are much more dense than presented here, but we show only non-overlapping patches for clarity. Color codes correspond to the colors in Figure 5. Note how different components capture different structures in the image. See text and Figure 5 for more details.

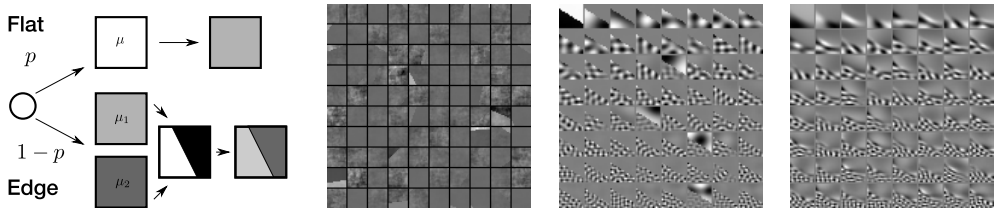


Figure 7: **(a)** The mini dead leaves models. Patches are either "flat" or "edge" patches. **Flat** patches are sampled from a multivariate Gaussian texture which is scaled by a contrast scalar and a mean value is added to it to form the patch. **Edge** patches are created by sampling two flat patches, an occlusion mask and setting the pixels of each side of the mask to come from a different flat patch. See text for full details. **(b)** Samples generated from the mini dead leaves model with their DC removed. **(c)** Leading eigenvectors of an edge component from a mini dead leaves model. **(d)** Leading eigenvectors of a component from the GMM trained on natural images - note how similar the structure is to the mini dead leaves model analytical result. See text for details.

vector and the scalar z is sampled from a discrete set of variables z_k with a corresponding probability π_k. This results in a GSM texture to which we add a random mean (DC) value. In all experiments here, we use a GSM trained on natural images.

Edge patches are generated by sampling two independent **Flat** patches from the texture process, $\mathbf{f}$ and $\mathbf{g}$, and then generating an *occlusion* mask to combine the two. We use a simple occlusion mask generation process here: we choose a random angle θ and a random distance r measured from the center of the patch, where both θ and r may be quantized — this defines the location of a straight edge on the patch. Every pixel on one side of the edge is assumed to come from the same object, and pixels from different sides of the patch come from different objects. We label all pixels belonging to one object by L_1 and to the other object by L_2. We then generate the patch by taking all pixels $i \in L_1$ to $x_i = f_i$ and similarly $x_{i \in L_2} = g_i$. This results in a patch with two textured areas, one with a mean value μ_1 and the other with μ_2. Figure 7a depicts the generative process for both kind of patches and Figure 7b depicts samples from the model.

4.3 Gaussian mixtures and dead leaves

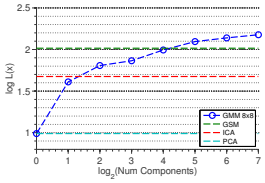
It can be easily seen that the mini dead leaves model is, in fact, a GMM. For each configuration of hidden variables (denoting whether the patch is "flat" or "edge", the scalar multiplier z and if it is an edge patch the second scalar multiplier z_2, r and θ) we have a Gaussian for which we *know* the covariance matrix *exactly*. Together, all configurations form a GMM - the interesting thing here is how the *structure* of the covariance matrix given the hidden variable relates to natural images.

For **Flat** patches, the covariance is trivial - it is merely the texture of the stationary texture process Σ multiplied by the corresponding contrast scalar z. Since we require the texture to be stationary its eigenvectors are the Fourier basis vectors [18] (up to boundary effects), much like the ones visible in the first two components in Figure 5.

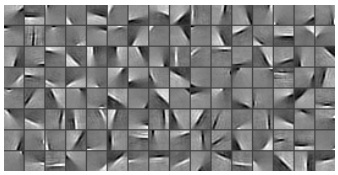
For **Edge** patches, given the hidden variable we know which pixel belongs to which "object" in the patch, that is, we know the shape of the occlusion mask exactly. If i and j are two pixels in different objects, we know they will be independent, and as such uncorrelated, resulting in zero entries in the covariance matrix. Thus, if we arrange the pixels by their object assignment, the eigenvectors of such a covariance matrix would be of the form:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{v} \end{bmatrix} \text{ or } \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}$$

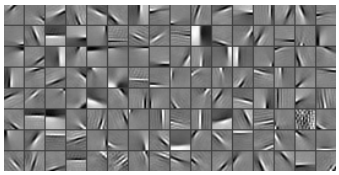
where $\mathbf{v}$ is an eigenvector of the stationary (within-object) covariance and the rest of the entries are zeros, thus eigenvectors of the covariance will be zero on one side of the occlusion mask and Fourier-like on the other side. Figure 7c depicts the eigenvector of such an edge component covariance - note the similar structure to Figure 7d and 5. This block structure is a common structure in the GMM learned from natural images, showing that indeed such a dead leaves model is consistent with what we find in GMMs learned on natural images.



(a) Log Likelihood Comparison



(b) Mini Dead Leaves - ICA



(c) Natural Images - ICA

Figure 8: **(a)** Log likelihood comparison with mini dead leaves data. We train a GMM with a varying number of components from mini dead leaves samples, and test its likelihood on a test set. We compare to a PCA, ICA and a GSM model, all trained on mini dead leaves samples - as can be seen, the GMM outperforms these considerably. Both PCA and ICA seek *linear* transformations, but since the underlying generative process is *non-linear* (see Figure 7a), they fail. The GSM captures the contrast variation of the data, but does not capture occlusions, which are an important part of this model. **(b)** and **(c)** ICA filters learned on mini dead leaves and natural image patches respectively, note the high similarity.

4.4 From mini dead leaves to natural images

We repeat the log likelihood experiment from sections 2 and 3, comparing to PCA, ICA and GSM models to GMMs. This time, however, both the training set and test set are generated from the mini dead leaves model. Results can be seen in Figure 8a. Both ICA and PCA do the best job that they can in terms of finding *linear* projections that decorrelate the data (or make it as sparse as possible). But because the *true* generative process for the mini dead leaves is *not* a linear transformation of IID variables, neither of these does a very good job in terms of log likelihood. Interestingly - ICA filters learned on mini dead leaves samples are astonishingly similar to those obtain when trained on natural images - see Figure 8b and 8c. The GSM model can capture the contrast variation of the data easily, but not the structure due to occlusion. A GMM with enough components, on the other hand, is capable of explicitly modeling contrast and occlusion using covariance functions such as in Figure 7c, and thus gives much better log likelihood to the dead leaves data. This exact same pattern of results can be seen in natural image patches (Figure 2), suggesting that the main reason for the excellent performance of GMMs on natural image patches is its ability to model both contrast and occlusions.

5 Discussion

In this paper we have provided some additional evidence for the surprising success of GMMs in modeling natural images. We have investigated the causes for this success and the different properties of natural images which are captured by the model. We have also presented an analytical generative model for image patches which explains many of the features learned by the GMM from natural images, as well as the shortcomings of other models.

One may ask - is the mini dead leaves model a good model for natural images? Does it explain everything learned by the GMM? While the mini dead leaves model definitely explains some of the properties learned by the GMM, at its current simple form presented here, it is not a much better model than a simple GSM model. When adding the occlusion process into the model, the mini dead leaves gains ~0.1 bit/pixel when compared to the GSM texture process it uses on its own. This makes it as good as a 32 component GMM, but significantly worse than the 200 components model (for 8×8 patches). There are two possible explanations for this. One is that the GSM texture process is just not enough, and a richer texture process is needed (much like the one learned by the GMM). The second is that the simple occlusion model we use here is too simplistic, and does not allow for capturing the variable structures of occlusion present in natural images. Both of these may serve as a starting point for a more efficient and explicit model for natural images, handling occlusions and different texture processes explicitly. There have been several works in this direction already [19, 20, 21], and we feel this may hold promise for creating links to higher level visual tasks such as segmentation, recognition and more.

References

[1] M. Bethge, "Factorial coding of natural images: how effective are linear models in removing higher-order dependencies?" vol. 23, no. 6, pp. 1253–1268, June 2006.

[2] P. Berkes, R. Turner, and M. Sahani, "On sparsity and overcompleteness in image models," in *NIPS*, 2007.

[3] S. Lyu and E. P. Simoncelli, "Nonlinear extraction of 'independent components' of natural images using radial Gaussianization," *Neural Computation*, vol. 21, no. 6, pp. 1485–1519, Jun 2009.

[4] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 479–486.

[5] B. Culpepper, J. Sohl-Dickstein, and B. Olshausen, "Building a better probabilistic model of images by factorization," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[6] L. Theis, S. Gerwinn, F. Sinz, and M. Bethge, "In all likelihood, deep belief is not enough," *The Journal of Machine Learning Research*, vol. 999888, pp. 3071–3096, 2011.

[7] G. Matheron, *Random sets and integral geometry*. Wiley New York, 1975, vol. 1.

[8] X. Pitkow, "Exact feature probabilities in images with occlusion," *Journal of Vision*, vol. 10, no. 14, 2010.

[9] B. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[10] A. J. Bell and T. J. Sejnowski, "The independent components of natural scenes are edge filters," *Vision Research*, vol. 37, pp. 3327–3338, 1997.

[11] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[12] Y. Karklin and M. Lewicki, "Emergence of complex cell properties by learning to generalize in natural scenes," *Nature*, November 2008.

[13] J. Sohl-Dickstein and B. Culpepper, "Hamiltonian annealed importance sampling for partition function estimation," 2011.

[14] M. Lewicki and B. Olshausen, "Probabilistic framework for the adaptation and comparison of image codes," *JOSA A*, vol. 16, no. 7, pp. 1587–1601, 1999.

[15] A. Lee, D. Mumford, and J. Huang, "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model," *International Journal of Computer Vision*, vol. 41, no. 1, pp. 35–59, 2001.

[16] C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," in *Digital images and human vision*. MIT Press, 1993, p. 138.

[17] E. Simoncelli, "Bayesian denoising of visual images in the wavelet domain," *Lecture Notes in Statistics - New York-Springer Verlag*, pp. 291–308, 1999.

[18] D. Field, "What is the goal of sensory coding?" *Neural computation*, vol. 6, no. 4, pp. 559–601, 1994.

[19] J. Lücke, R. Turner, M. Sahani, and M. Henniges, "Occlusive components analysis," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1069–1077, 2009.

[20] G. Puertas, J. Bornschein, and J. Lücke, "The maximal causes of natural scenes are edge filters," in *NIPS*, vol. 23, 2010, pp. 1939–1947.

[21] N. Le Roux, N. Heess, J. Shotton, and J. Winn, "Learning a generative model of images by factoring appearance and shape," *Neural Computation*, vol. 23, no. 3, pp. 593–650, 2011.

Chapter 3

Discussion

3.1 Summary of contributions of this thesis

In the previous sections I have presented four papers which form the main body of this thesis. We will now survey the main contribution of each paper and how each one relates to current literature as it was presented in Section 1. In the following sections we will give a broader perspective of the works here and point to some future directions.

The first paper of this thesis presented a novel hypothesis for the marginal statistics of filter responses in natural images. We proposed that in clean images, the *kurtosis* of marginal filter responses is constant when computed for various band-pass filters, and does not depend on frequency. This is in contrast to $1/f^2$ scaling law for variance we have seen in section 1.3.1. We have shown that when noise is added to the image the kurtosis of the marginals change as a result and we show how we can use this to estimate noise in natural images, achieving state-of-the-art performance in noise estimation.

In the second of this thesis we presented statistical model for natural images patches based on tree structured graphical models. We proposed that because there are strong dependencies between filter responses in natural images it would make sense to learn a model which attempts to capture these dependencies explicitly. We have presented a method to learn this tree structured graphical model together with the pair-wise potentials and filters which maximize the likelihood of the model. The method learned filters which are similar to the ones learned by ICA (as we have seen in Section 1.4.2). Additionally we saw that filters with similar selective properties are grouped to be close on the tree graph, such that the branches of the tree form long chains of related filters. The model was demonstrated to achieve higher log-likelihood scores than competing models.

The third and fourth papers may be regarded as one continuous work. We began these pair of works by asking the question "how can we use *patch* priors for *whole image* restoration?". In Section 1.5 we have discussed how priors in general may be used in image restoration, but there we assumed that the prior is defined over the same space we wish to restore i.e. images or patches. In the paper in Section 2.3 we suggest that what needs to be done is to maximize the *expected patch log-likelihood* (EPLL) of the output image — this creates the necessary link between our patch prior and the whole image. We continued this work by presenting the Gaussian Mixture Model (GMM) prior for natural image patches. This model turned out to be a strong performer in log-likelihood and denoising performance. We showed that we can achieve state-of-the-art denoising performance by using the EPLL framework together with the GMM prior. We further analyze the reasons of the GMM's success in the fourth paper. We show that the GMM learns basic properties of natural images: edges, textures, boundaries and so on. We link these results to the "dead leaves" model (which we will come back to in Section 3.3).

3.2 Dependencies and redundancy reduction in natural images

Redundancy reduction is often regarded as one of the primary goals of the visual system [1, 26]. Since visual input has an intricate high dimensional structure it is very hard to process it in its raw form. Any vision system, be it biological vision or machine vision, has to somehow make the problem of vision simpler. Redundancy reduction is one way of making a complex signal simpler, moving from a complex high dimensional description of the data to a simpler low dimensional one.

Many of the models we have discussed in this thesis do some kind of redundancy reduction - either explicitly or implicitly. Out of those models ICA (Section 1.4.2) is probably the most explicit model as it directly attempts to find a representation in which the data is factorial. Though explicit, ICA really fails in achieving its goal when it comes to natural images [43, 47]. As there are many high order dependencies in natural images which a linear factorial model such as ICA can't capture, the residual information remaining after performing ICA is substantial. This rather surprising result has served as a starting point for the paper in Section 2.2. In that paper we attempt to explicitly model the dependencies ICA and related models can't discard of. There is, however, a limitation to such an explicit modeling because the dependencies in natural images are complex and mostly unknown, thus finding such a model without knowing what you are looking for is hard.

In the papers in Section 2.3 and 2.4 we have presented the GMM prior and we have shown it is a very strong model for modeling natural image statistics. As such, one may ask if the GMM is also good at redundancy reduction, and here we show that there is strong evidence that it is quite good at this as well. One way to quantify redundancy reduction is calculating the Mutual Information (MI) between pairs of coefficients in each model. For linear models such as PCA and ICA this is trivial because the coefficients are direct projections of the data. In the case of the GMM this is more challenging because there are multiple basis choices (coming from each of the GMM's components). The way we handle this is to choose for each data sample the *most likely* component, $\hat{k}$ that is, the component which maximizes the likelihood for the given patch $\mathbf{x}$:

$$\hat{k} = \arg\max_k \pi_k \mathcal{N}(\mathbf{x}; 0, \mathbf{\Sigma}_k) \tag{3.1}$$

where π_k and $\mathbf{\Sigma}_k$ are the k-th mixing weight and covariance matrix in the mixture, correspondingly. Conditioned on this $\hat{k}$ component we have the "right" covariance matrix for the patch $\mathbf{x}$ and we can use its eigenvectors to project the patch into a basis in which we will hopefully reduce the pair-wise dependencies (this is much like PCA, only there is a model selection step which is non-linear and allows for different "families" of covariances to be used). Figure 3.1 depicts pair-wise MI histograms for a large test set of natural image patches, one for PCA, one for ICA and one for the GMM using the aforementioned method. Note that while ICA definitely reduces the mean pair-wise MI, the GMM is much better at this task. This, again, hints that the GMM indeed finds meaningful representations for natural images which not only allows for better denoising and likelihood performance but also serve as excellent redundancy reduction tool.

3.3 Richer dead leaves models?

At the end of the paper in Section 2.4 we have suggested that a simple "Mini Dead Leaves" (MDL) model inspired by the dead leaves model (see Section 1.3.1) can explain some of the salient properties learned by the GMM model. While it definitely explains *some* of the properties captured by the GMM, when we compare the performance of the MDL model to the GMM we see that it

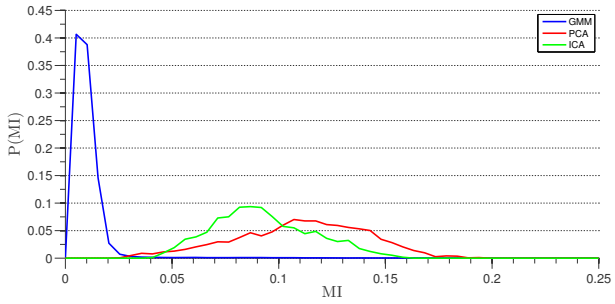


Figure 3.1: Redundancy reduction for three natural image models. Depicted are three histograms for pair-wise MI for all pairs of coefficients in each model. ICA definitely improves upon PCA, as the mean MI is lower but as can be seen, the GMM overshadows them both, allowing for a much more efficient representation via its inherent non-linear properties. See text for details.

falls behind. Figure 3.2a depicts the performance of the MDL model together with the PCA, ICA, GSM and GMMs with a varying number of components. The texture process used for this MDL is the GSM model, with 16 components trained on natural images. It seems that while moving from the GSM to the MDL by adding the occlusion masks improves performance when comparing to the GSM, it is not as good as the GMM. This means that there are some natural image properties which the GMM captures which the MDL model does not.

In order to find what kind of patches does the GMM model better than the MDL model we can perform the following experiment: calculate the log likelihood each model gives to each patch in the test set and then calculate the difference between the two models, GMM - MDL. Sorting these log likelihood differences reveals which patches does the GMM model better, and which patches does the MDL model better. Figure 3.2b depicts the patches for which the log likelihood difference is greatest for the GMM and the MDL. As expected patches which the MDL models well are mostly edge patches with very clear edges. The GMM, however, seems to be able to capture patches which have a "bar" like structure — most of these patches have two edges or borders which are parallel to each other with different texture or mean value at each area.

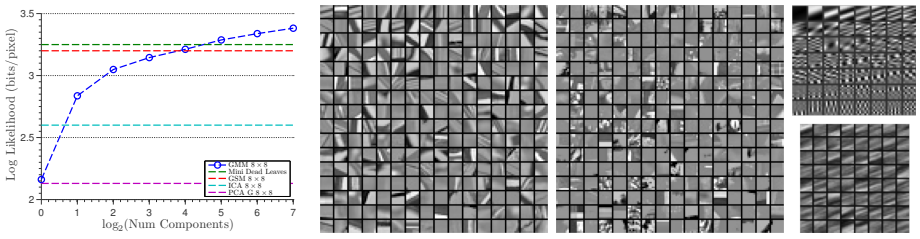
Because the occlusion structure in MDL is very simple and allows for only a single edge in each patch this model can't properly handle these kind of patches. But how does the GMM handle these kind of structures? Figure 3.2d depicts the eigenvectors of the component assigned to these "bar" patches. It can be seen that the structure of correlation is mostly oriented along the bar direction, allowing for relatively flat textured bars to reside all along the patch (see the samples from the component below in Figure 3.2d). This simple analysis shows us that in order to build a better explicit model for natural images a richer mask model is required, probably together with a richer texture model.

3.4 Future work

To conclude this thesis I will now survey some possible directions for further research the works presented here may lead to.

In the first presented work here (Section 2.1) we have conjectured that the kurtosis of marginal filter response histograms in natural images is constant with scale. This simple form of scale invariance is not predicted in current works on scale invariance in natural images. Considering this scale invariant property is an empirical observation it would be interesting to look for an analytical model which may be able to predict this property, together with other properties we have encountered throughout this thesis.

The second work here (Section 2.2) presents a tree structured graphical model for modeling the joint density of natural image patches. There several possible directions for future work on



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

Figure 3.2: Mini Dead Leaves vs. GMM. **(a)** Log likelihood of several models, including MDL on natural image patches. Note that the introduction of simple occlusion does not improve performance in a significant way. **(b)** Patches which are the GMM models better than the MDL. **(c)** Patches which the MDL models better than the GMM. As can be seen, the GMM is able to capture bar like structures which the MDL can not, having more than one edge in the patch. The MDL captures sharp, simple edge like structures well. **(d)** The eigenvectors and samples from the GMM component assigned to the "bar" patches in **(b)**. See text for details.

this front, but the most promising one is going beyond trees. The field of probabilistic graphical models has become a popular research area in recent years [48, 49] and there have been much progress in estimating, learning and inference of higher order structures. Going beyond trees will be of great importance for natural image modeling — because the interactions in natural images are sometimes patch global (as we have seen, for example in Section 2.4), the order required for a proper model is much higher than the pair-wise, tree structured interactions the proposed model captures.

Finally, the future directions for the works in Sections 2.3 and 2.4 are probably the most varied and promising. Starting with the EPLL framework presented in Section 2.3, and considering, again, the scale invariance properties we have encountered throughout this thesis, it would be interesting to extend it into the multiscale domain. This may be done either by extending the framework to work at multiple scales explicitly by moving from patches to a multiscale representation, to train priors which take into account multiple scale and/or combinations of the two. The GMM prior which was thoroughly discussed here also holds some promise in making our understanding of natural images better. Creating a multiscale version of it would be an obvious direction to go to, as this property is largely ignored in its analysis and, as we have seen, is an important property that should probably be accounted for. Additionally, the analysis we saw in Section 3.3 really begs us to find a better, richer parameterization for the MDL model — this will allow us to have an explicit model for natural images which may be useful not only for low-level vision tasks as denoising, but also for mid-level tasks such as boundary detection, segmentation and more.

Bibliography

[1] H. Barlow, "Possible principles underlying the transformation of sensory messages Sensory Communication ed WA Rosenblith," 1961.

[2] D. L. Ruderman, "Origins of scaling in natural images," *Vision Research*, vol. 37, pp. 3385–3398, 1997.

[3] S. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1236–1250, 1997.

[4] Y. Weiss and W. Freeman, "What makes a good model of natural images?" *CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[5] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Gatesmark Publishing Knoxville, 2009, vol. 2.

[6] C. Liu, R. Szeliski, S. Bing Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, 2008.

[7] E. Kretzmer, "Statistics of television signals," *Bell Syst. Tech. J*, vol. 31, no. 751-763, p. 2, 1952.

[8] N. Deriugin, "The power spectrum and the correlation function of the television signal," *Telecommunications*, vol. 1, no. 7, pp. 1–12, 1956.

[9] D. Gabor, "Theory of communication. part 1: The analysis of information," *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, vol. 93, no. 26, pp. 429–441, 1946.

[10] D. J. Field *et al.*, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Am. A*, vol. 4, no. 12, pp. 2379–2394, 1987.

[11] D. Field, "What is the goal of sensory coding?" *Neural computation*, vol. 6, no. 4, pp. 559–601, 1994.

[12] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, "On advances in statistical modeling of natural images," *J. Math. Imaging Vis.*, vol. 18, no. 1, pp. 17–33, 2003.

[13] S. Roth and M. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[14] Y. Karklin and M. Lewicki, "A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals," *Neural Computation*, vol. 17, no. 2, pp. 397–423, 2005.

[15] B. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[16] A. Hyvrinen, P. Hoyer, and M. Inki, "Topographic independent component analysis: Visualizing the dependence structure," in *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000), Espoo, Finland*. Citeseer, 2000, pp. 591–596.

[17] A. J. Bell and T. J. Sejnowski, "The independent components of natural scenes are edge filters," *Vision Research*, vol. 37, pp. 3327–3338, 1997.

[18] X. Pitkow, "Exact feature probabilities in images with occlusion," *Journal of Vision*, vol. 10, no. 14, 2010.

[19] A. Lee, D. Mumford, and J. Huang, "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model," *International Journal of Computer Vision*, vol. 41, no. 1, pp. 35–59, 2001.

[20] D. L. Ruderman and W. Bialek, "Statistics of natural images: Scaling in the woods," in *NIPS*, 1993, pp. 551–558.

[21] M. J. Wainwright and E. P. Simoncelli, "Scale mixtures of gaussians and the statistics of natural images." in *NIPS*. Citeseer, 1999, pp. 855–861.

[22] E. Lam and J. Goodman, "A mathematical analysis of the dct coefficient distributions for images," *Image Processing, IEEE Transactions on*, vol. 9, no. 10, pp. 1661–1666, Oct 2000.

[23] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[24] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[25] A. Hyvèarinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics*. Springer, 2009, vol. 39.

[26] H. Barlow, "Redundancy reduction revisited," *Network: computation in neural systems*, vol. 12, no. 3, pp. 241–253, 2001.

[27] J. Bastian, M. J. Chacron, and L. Maler, "Plastic and nonplastic pyramidal cells perform unique roles in a network capable of adaptive redundancy reduction," *Neuron*, vol. 41, no. 5, pp. 767–779, 2004.

[28] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[29] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, 1999.

[30] A. Hyvarinen and P. Hoyer, "Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, 2000.

[31] J. van Hateren, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society B: Biological Sciences*, vol. 265, no. 1394, pp. 359–366, 1998.

[32] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 40–44.

[33] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[34] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: Design of dictionaries for sparse representation," in *IN: PROCEEDINGS OF SPARS05*, 2005, pp. 9–12.

[35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 689–696.

[36] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, no. 5456, pp. 1273–1276, 2000.

[37] B. Olshausen and D. Fteld, "Sparse coding with an overcomplete basis set: a strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[38] B. A. Olshausen and D. J. Field, "How close are we to understanding v1?" *Neural computation*, vol. 17, no. 8, pp. 1665–1699, 2005.

[39] A. Hyvarinen, P. Hoyer, and E. Oja, "Image denoising by sparse code shrinkage," in *Intelligent Signal Processing*. IEEE Press, 1999.

[40] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[41] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *SPIE Electronic Imaging*, 2008.

[42] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2010, pp. 2272–2279.

[43] M. Bethge, "Factorial coding of natural images: how effective are linear models in removing higher-order dependencies?" vol. 23, no. 6, pp. 1253–1268, June 2006.

[44] Y. Karklin and M. Lewicki, "Emergence of complex cell properties by learning to generalize in natural scenes," *Nature*, November 2008.

[45] D. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[46] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[47] D. Zoran and Y. Weiss, "The "tree-dependent components" of natural scenes are edge filters," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 2340–2348.

[48] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, pp. 239–236, 2003.

[49] D. Kollar and N. Friedman, *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

טבעיות ומראים כי במונחי נראות אנו מציעים שיפור משמעותי.

שתי העבודות האחרונות בתזה זו מציגות מודל שמתבסס על כלי פשוט ונפוץ - תערובות גאוסיניים. בעבודה הראשונה בסדרת עבודות זו אנו מציעים מסגרת עבודה שמאפשרת לעבוד עם מודלים המוגדרים עבור "טלאים" של תמונות כאשר אנו רוצים לעבוד עם תמונות מלאות. באמצעות מסגרת עבודה זו ומודל תערובות הגאוסיינים אנו מראים כי ניתן להשיג תוצאות מעולות בשחזור וניקוי תמונות, אשר מתחרות אפילו עם השיטות המובילות בעולם בתחום. בהמשך לכך, בעבודה האחרונה בתזה והשנייה בסדרה זו, אנו ממשיכים בניתוח הסיבות להצלחה של מודל תערובת הגאוסיינים והצלחתו המפתיעה. בעבודה זו אנו מבצעים השוואה מדוקדקת של מודל זה עם מגוון מודלים מובילים לתמונות טבעיות ומנתחים את התכונות אותן לומד מודל תערובת הגאוסיינים אשר מאפשרות את הצלחתו. לסיום, אנו מציעים מודל אנליטי ל"טלאים" של תמונות טבעיות אשר אנחנו מכנים מודל "העלים המתים הקטנים" ואשר מתאר בצורה מפורשת כמה מהתכונות הבולטות אותן לומד מודל תערובת הגאוסיינים - ניגודיות, טקסטורה והסתרות.

תקציר

עבור כל מי שעוסק בעיבוד תמונה, ראייה חישובית או ראייה ביולוגית, הסטטיסטיקה של תמונות טבעיות היא נושא רלוונטי ומרכזי. בהיותן הקלט המרכזי לכל מערכת ראייה באשר היא, יש עניין רב בלהבין את המבנה המיוחד של התמונות הטבעיות והתכונות הסטטיסטיות אשר מייחדות אותן. בתזה זו, אציג סידרת עבודות אשר כל אחת מהן מנסה לקדם את ההבנה שלנו של תמונות טבעיות ותכונותיהן הסטטיסטיות - מהתפלגויות שוליות של תגובות פילטרים לינארים ועד למודלים אנליטיים שלמים עבור "טלאים" אשר נגזרים מתמונות שלמות. בכל העבודות כאן נשווה את המודלים המוצעים למודלים והשיטות אשר נמצאים בחזית המחקר כיום.

לתמונות טבעיות יש מגוון תכונות סטטיסטיות שבאות לידי ביטוי בהקשרים שונים, רבים מהם רלוונטים לעבודות המוצגות פה. שתי תכונות מרכזיות בהן נדון בעבודה זו הן האינבריאנטיות לשינויים בגודל, והסטטיסטיקה הלא גאוסיינית של תמונות טבעיות. הסטטיסטקה של תמונות טבעיות אינה תלויה בסקאלה בהן בוחנים אותן, משמעות הדבר שהגדלת תמונות, או הקטנתן, לא משפיעות על הסטטיסטיקה שלהן. אחת התכונות הבולטות והנחקרות ביותר אשר נובעות מתכונה זו הינה הספקטרום שלהן, אשר מתנהג כמו חוק חזקה עם התדר: האנרגיה בספקטרום של תמונות טבעיות דועכת עם התדר כמו התדר בריבוע. תכונה חשובה נוספת של תמונות טבעיות אשר תשחק תפקיד מרכזי בעבודות אשר מוצגות בתזה זו היא האי-גאוסייניות של התמונות הטבעיות. אם נערוך היסטוגרמה של תגובות פילטרים לינארי בעל ממוצע אפס על פני תמונות טבעיות, נגלה כי אנו מקבלים התפלגויות מאוד לא גאוסייניות, עם זנבות עבים, ופסגות גבוהות מאוד סביב האפס. תכונה זו היא תוצאה של מגוון תופעות שונות אשר בהן נדון בעבודות השונות אשר מרכיבות את תזה זו.

תזה זו מורכבת ממספר עבודות שונות. העבודה הראשונה דנה בהתפלגויות השוליות של תגובות פילטרים ליניארים כאשר מפעילים אותם על תמונות טבעיות. אנו מראים כי הקורטוזיס (המומנט הרביעי המרכזי) של התפלגויות אלה מושפע מרעש אשר נוכח בתמונות. אנו מציעים כי בתמונות טבעיות נקיות, הקורטוזיס אינו משתנה עם תדר הפילטר וכי רעש אשר נוכח בתמונה גורם לשינויים בערכיו. אנו מראים כי ניתן להשתמש בתכונה זו על מנת לשערך את עוצמת הרעש בתמונה נתונה ברמת דיוק המתחרה בשיטות המובילות בעולם כיום בשערוך רעש. בעבודה השניה אנו מציעים מודל אשר מקרב את פונקציית ההתפלגות המשותפת של "טלאים" מתמונות טבעיות על ידי מודל גרפי הסתברותי בצורת עץ, אשר נלמד מ"טלאים" אלה. אנו מראים כי מודל כזה לומד סט של פילטרים אשר ממוקמים במרחב ובתדר וסלקטיביים לאוריינטציה ופאזה, בדומה למודלים של תאים "פשוטים" בקליפת המוח הראייתית. פילטרים אלה מקובצים יחד על ידי מבנה העץ כל שהם יוצרים מבנים אשר מזכירים מודלים של תאים "מורכבים" בקליפת המוח הראייתית - תאים אשר סלקטיביים למיקומים, תדרים ומיקומים דומים אך פאזות שונות נמצאים בשכנות ויוצרים מבנים המזכירים פונקציונלית תאים אלו. אנו משווים מודל זה למספר מודלים מובילים לתמונות

עבודה זו נעשתה בהדרכתו של
פרופסור יאיר וויס

סטטיסטיקה של תמונות טבעיות עבור ראיית אדם ומכונה

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת

דניאל צורן

הוגש לסנט האוניברסיטה העברית בירושלים

יולי 2013