Statistics of Images, the TV Algorithm of Rudin-Osher-Fatemi for Image Denoising and an Improved Denoising Algorithm

Mark L. Green*, Dept. of Mathematics and Institute for Pure and Applied Mathematics, UCLA

October 1, 2002

The goal of this paper is to present a new basic model for the joint density function for a broad class of spatial and time-series data. As evidence that this model is indeed useful in practical problems, an application to image denoising in the presence of textures will be explained.

Section 1 discusses the joint density distribution for pixel intensities in naturally occurring images. This reflects an experimental discovery made by examining pixel intensities for a variety of naturally occurring images–that the pixel intensities in many images have a property I call **Differentially Laplacian**. The idea here is to consider not just differences between measurements, but all linear combinations of measurements where the coefficients add up to 0–when the data points are adjacent, these correspond to discretizations of linear differential operators, but it is fruitful to consider such differences for all data points in $k \times k$ sub-blocks of the image for k of modest size. The invertibility of the Radon transform uniquely specifies the density function of a Differentially Laplacian collection of random variables once one knows the autocorrelation function. Further experimental data describes how the autocorrelation between pixel intensities in many images dies off as a function of the distance d between them–this is very often of the form $(1+\gamma d)^{-\alpha}$. Although there are many much more sophisticated statistical models for specific textures, the model given here has a wide range of applicability–specific images of plants, lava, galactic clusters, bacteria, forests, satellite images of craters and cities support it. Moving beyond the world of image processing, time series data on hourly ocean temperatures, digital elevation models of the Rocky Mountains and Australia, and Icelandic daily precipitation also offer a good fit with this relatively simple model.

In section 2, it is shown that the 1-dimensional case of the TV algorithm of Rudin-Osher-Fatemi is equivalent to taking the maximum likelihood estimate for denoising an image with added white noise under the assumption that the differences between adjacent pixel intensities are distributed by a Laplacian density and that these differences are independent of one another. The first assumption is experimentally accurate, while the second of course is not–it is nevertheless an excellent first approximation to reality.

Section 3 uses the information about the joint pixel intensities in section 1 to create a denoising algorithm that improves upon the TV algorithm for images with texture.

The overall approach taken in this paper might be characterized as "empirically Bayesian." I was introduced to Bayesian statistics through my work as an organizer of the program in Functional Genomics at the Institute for Pure and Applied Mathematics. I have also benefited from very useful conversations with my colleagues Stan Osher and Tony Chan, who introduced me to this collection of problems.

* Research partially supported by the National Science Foundation

There is a sizable literature on statistics of natural images, with seminal work of Donoho, Huang, Mumford, Ruderman, Simoncelli, Yuille and Zhu among others. The autocorrelation dieoff function is discussed in [L-M],and its tendency to die off as a power law has been noted by several authors. I have not been able to determine from the literature who first noted that the difference between adjacent pixel intensities frequently has Laplacian density. The statistics of wavelet blocks have been studied by several authors, notably Simoncelli, Mumford and Huang, and the tendency of wavelet coefficients to have Laplacian statistics has been noted; a perspective on these closest to this paper is that of [Hu]. The joint density of pairs of wavelet coefficients is considered by Simoncelli and Huang. Two papers with a good survey of the literature on this are [L-P-M] and [S-L-S-Z]. There is likewise a large literature on image denoising. Soft thresholding goes back to [D-J-K-P]. A paper of Hyvärinen [H] uses maximum likelihood estimates in the context of the Lemma of section 3. However, his overall thrust is in the direction of using supervised learning and independent component analysis. Huang [Hu] also discusses using a maximum likelihood estimate for denoising. Using Laplacian statistics for predicting pixel intensities from those of their neighbors was used, among other places, for lossless image compression in [W-S-S]. The new contributions in this paper–although all have antecedents–are the joint density model proposed in section 1, the maximum likelihood explanation of the TV algorithm, and the application of the density model in section 1 to image denoising.

1. Joint Density Function of Pixel Intensities in Naturally Occurring Images

This section, based on experiment, proposes a simple model for the joint density function of all differences of pixel intensities in a certain class of images. For the applications we have in mind, it is sufficient to get a model which works well for the inter-relationships of pixel intensities in small sub-blocks (e.g. 5 x 5). Thus a practical embodiment of the joint density function we have in mind is to divide an image into $k \times k$ sub-blocks, and then look at the histogram of the k^2 pixel intensities for those blocks within a given image. What we claim is that when we do this, for many images the outcome is modelled very closely by a single very simple model having one new parameter plus a scale factor beyond the variance of the intensity of a single pixel. This model will then be shown to work well for some other types of data.

The class of images we are interested in are naturally occurring images. We want to begin by studying images which are not composites of disparate subimages, each having a different character, but rather images which are uniform over the entire picture and are somehow governed by a single "regime"–the foliage of a tree, a forest, a field of lava, a galactic cluster. If we find a simple model which describes these individual images well, we can then segment a more complex image into pieces that are well-described by the model. A number of things should be ruled out–images with large areas in shadow, or water and sky, and artificial or man-made objects. The model is so highly specific that it is remarkable that any images at all are fitted by it, much less the wide class evidenced in this paper.

The experimental observation is the following:

OBSERVATION. *Break an image into $k \times k$ sub-blocks, and consider a fixed linear combination of the k^2 pixel intensities where the coefficients of the linear combination add* *up to 0. For many naturally occurring images, this has a Laplacian density function for every such linear combination and all k of modest size.*

DEFINITION. *A collection $X_1, \ldots, X_n$ of random variables will be said to be* **Linearly Laplacian** *if every linear combination $X_A = \sum_{i=1}^{n} a_i X_i$ has Laplacian density.*

DEFINITION. *A collection $X_1, \ldots, X_n$ of random variables will be said to be* **Differentially Laplacian** *if every linear combination $X_A = \sum_{i=1}^{n} a_i X_i$ has Laplacian density provided $\sum_{i=1}^{n} a_i = 0$.*

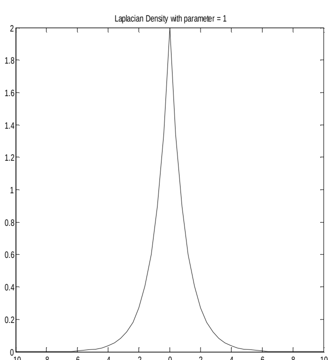
We may rephrase the observation by saying:

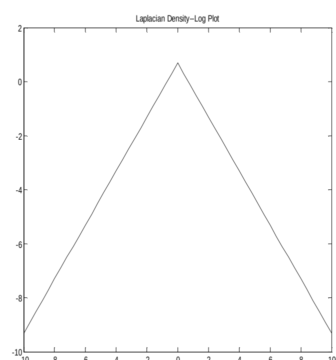
OBSERVATION. *For k of modest size, the pixel intensities for the pixels in $k \times k$ sub-blocks of an image constitute a Differentially Laplacian set of random variables.*

REMARKS:
1. A **Laplacian random variable** is one whose density function is described by the **absolute exponential density**

$$f(x) = \frac{\beta}{2} e^{-\beta|x|}.$$

Such a random variable has mean 0 and variance $2/\beta^2$. A Laplacian density and its log look like:





2. The observation should be interpreted in the sense that we take $k \times k$ sub-blocks of an image, take linear combinations of pixel intensities on the sub-block whose coefficients add up to 0, and then investigate the distribution of the values of this random variable over a decomposition of a large image into $k \times k$ blocks.
3. For the difference in intensities between adjacent pixels, the fact that the density tends to be Laplacian has been noted by other observers (see [H-M]).
4. Unlike Gaussian random variables, for which the sum of independent Gaussian random variables is also Gaussian, a sum of independent Laplacian random variables will not be Laplacian. Thus the property observed is quite unexpected–for example, the joint density function of three adjacent pixel intensities X_1, X_2, X_3 is so constructed that not only the differences $X_1 - X_2$ and $X_2 - X_3$ are Laplacian random variables, but also their sum $X_1 - X_3$ is Laplacian, as is indeed any linear combination of $X_1 - X_2$ and $X_2 - X_3$.

Given an $A \times B$ image, a **rescaling** of the image is obtained by breaking the image into $a \times b$ sub-blocks and averaging the pixel intensities over each sub-block to obtain an $A/a \times B/b$ image. We have the highly suggestive fact that being Differentially Laplacian is inherited under rescaling:

FACT. *If an image is Differentially Laplacian, then any rescaling of the image is also Differentially Laplacian.*

This fact follows automatically from the fact that any linear combination of pixel intensities for the rescaled image is *a fortiori* a linear combination of pixel intensities for the original image, and the property of the coefficients adding up to zero is inherited as well.

Once one has a Differentially Laplacian collection of random variables, the one remaining piece of information is how to predict the parameter β above for all of the X_A. An equivalent piece of information is what I call the **autocorrelation dieoff function**–essentially the same information as what geostatisticians call the variogram–defined by:

DEFINITION. *Let X_{ij} denote the pixel intensity at the (i, j) position of an $m \times m$ sub-block of an image. Assume that the joint density function is invariant under translations and isotropic, so that $\mathrm{Corr}(X_{ij}, X_{kl})$ depends only on $d = \mathrm{distance}((i, j), (k, l))$. Then the autocorrelation dieoff function is*

$$\rho(d) = \mathrm{Corr}(X_{ij}, X_{kl}) \qquad \text{where } d = \mathrm{distance}((i, j), (k, l)).$$

REMARK: The assumption of isotropy is not accurate in images with a "horizon," where height in the picture corresponds to the object tending to be further away. Our model works with non-isotropic autocorrelation dieoff functions, and our denoising algorithm does not assume isotropy.

OBSERVATION. *In many images, the autocorrelation dieoff function tends to behave as*

$$\rho(d) = (1 + \gamma d)^{-\alpha}$$

for some $\alpha, \gamma > 0$.

REMARK: This autocorrelation dieoff function is **rescaleable up to atomic scale**, meaning that for large values of d, it rescales but deviates from this for small values of d. If would rescale perfectly if we used $\gamma d^{-\alpha}$; the 1 is inserted in order to make the correlation come out to be 1 when $d = 0$.

Now consider for a set of random variables $X_1, \ldots, X_n$ the **covariance matrix**

$$B = (b_{ij}); \qquad b_{ij} = \mathrm{Covariance}(X_i, X_j).$$

An easy computation shows:

FORMULA. *A collection of random variables $X_1, X_2, \ldots, X_n$ with density function $f(x_1, \ldots, x_n)$ has a linearly Laplacian density if and only if*

$$\int_{\mathsf{R}^n} f(x_1, \ldots, x_n) e^{i(\omega_1 x_1 + \cdots + \omega_n x_n)} dx_1 dx_2 \cdots dx_n = \frac{2}{\|\omega\|_B^2 + 2},$$

where

$$\|\omega\|_B = \sum_{ij} b_{ij}\omega_i\omega_j.$$

Note that this specifies the Fourier transform of the density function f, and hence determines f. I do not expect that f will have a nice closed-form expression, which perhaps accounts for this type of joint density function not having been discovered before. One should integrate the inverse Fourier transform by integrating first normal to the linear spaces $\sum_i \omega_i x_i$ = constant. Doing this, if one uses the eigenfunctions and renormalizes so that the parameters β are all 1, one sees that the density function is a function of the distance r from the origin, and for $n \geq 2$,

$$f(r) = \text{constant} \int_0^\infty \rho^{n-2} e^{-r\sqrt{1+\rho^2}}\, d\rho,$$

where the constant is chosen so that the total densityis 1. For n odd,this has a simple closed form–for example, for $n = 3$,

$$f(r) = \frac{1}{8\pi}\left(\frac{1}{r^2} + \frac{1}{r}\right)e^{-r}.$$

THEOREM. *There is one and only one probability joint density function for Linearly Laplacian random variables $X_1, \ldots, X_n$ with a given covariance matrix B.*

PROOF: This follows from the Fourier inversion formula.

For a set of variables $X_1, \ldots, X_n$, if $Y_1, \ldots, Y_{n-1}$ is a basis for the linear combinations of $X_1, \ldots X_n$ with coefficients adding up to 0, e.g. $Y_i = X_{i+1} - X_i$, then $X_1, \ldots X_n$ is Differentially Laplacian if and only if $Y_1, \ldots, Y_{n-1}$ is linearly Laplacian.

FACT. *If $X_1, \ldots, X_n$ is Differentially Laplacian, then* $\mathrm{mean}(X_i)$ *is independent of i.*

REMARK: This follows because the mean of the differences is 0 by definition.

The difference between knowing a density function for the X's and for the Y's should be viewed as follows: Just as for Brownian motion, we think of our particle as starting at some fixed point and then moving according the joint density for differences in position, we should think of the pixel intensities as being given by a 2-dimensional Brownian motion "of a different color" starting from some initial point and whose differences in intensity are given by a linearly Laplacian joint density. Under this scenario, there is no preassigned joint density for the pixel intensities X_{ij}–rather, one should start at intensity 0, carry out the process, and then add a constant to all of the pixel intensities. This constant is estimated by the mean of the pixel intensities. For this reason, the mean, variance, and autocorrelation dieoff function give a complete set of invariants of the process.

To complete the discussion of images, we briefly mention what happens for color images. If X_{ij}^R, X_{ij}^G, X_{ij}^B are the pixel intensities at position (i, j) for the red, green and blue channels repsectively, then we have the following variant of being Differentially Laplacian:

OBSERVATION. *The random variables*

$$X_A = \sum_{ij} (a_{ij}^R X_{ij}^R + a_{ij}^G X_{ij}^G + a_{ij}^B X_{ij}^B)$$

have a Laplacian density provided

$$\sum_{ij} a_{ij}^R = 0; \sum_{ij} a_{ij}^G = 0; \sum_{ij} a_{ij}^B = 0.$$

Although we will not develop this further in this paper, this observation is crucial in carrying out a color version of the denoising algorithm of section 3.

A final comment is that there has been a very fruitful study of $1/f$-noise and $1/f$-Brownian motion, arising especially from the work of Mandelbrot [M]. The analogue of Brownian motion discussed here differs from $1/f$-Brownian motion in two ways. First, the differences are Laplacian rather than Gaussian. Secondly, and more importantly, $1/f$-Brownian motion is a class of stochastic processes, rather than a single specific stochastic process. In essence, $1/f$-Brownian motion specifies the density of $X(s) - X(t)$ for all s, t as being Gaussian with variance a power of $|s-t|$. One might complete this by specifying the density of all linear cominations

$$\sum_i a_i X(t_i) \qquad \text{whenever } \sum_i a_i = 0.$$

These linear combinations do not have the same physical meaning as differences do, but as is indicated by the results in this paper, they are useful in describing the joint density function. Some natural phenomena will not have this additional property, but I suspect that many will. For ordinary Brownian motion, consideration of linear combinations is unnecessary, since independence guarantees being Differentially Gaussian, i.e. having Gaussian density for linear combinations summing to 0. The Differentially Laplacian property with a given autocorrelation dieoff function allows us to specify a unique 2-parameter family of motions. The claim of this paper is that pixel intensities for many images, and indeed many other types of spatial and time-series data are described by this Differentially Laplacian variant of Brownian motion.

This completes a description of the theory behind the model being proposed; applications of the theory will appear in sections 2 and 3. The remainder of this section is devoted to empirical evidence supporting the claim that for many images and several other types of data, the model of Differentially Laplacian random variables with autocorrelation dieoff function $(1+\gamma d)^{-\alpha}$.

I will show two types of data in favor of at least some images having the Differentially Laplacian property. One can choose an image and a $k \times k$ matrix A, and then construct the histogram of X_A for that image. A more systematic procedure is the following: Choose a number k and decompose the image into $k \times k$ blocks. Let V be the (k^2-1)-dimensional space of linear combinations of pixel intensities on a $k \times k$ block with coefficients adding up to 0. If $A \in V$, define a quadratic functional by

$$\|A\|^2 = \mathrm{Var}(X_A),$$

or equivalently a positive-definite inner product on V by

$$< A, B >= \mathrm{Cov}(X_A, X_B).$$

We obtain k^2-1 eigenfunctions $A_1, \ldots, A_{k^2-1}$ of the $k \times k$ block for this functional. We can then look at the k^2-1 random variables $X_{A_1}, \ldots, X_{A_{k^2-1}}$ and investigate whether these are in fact all Laplacian random variables. This is a rather efficient way of getting a lot of data at once, and much of our data will be for the 24 eigenfunctions of 5×5 blocks of various images.

We may summarize man of the observations and deductions of this section by:

OVERALL MODEL.
1. Let $A_1, A_2, \ldots, A_{k^2-1}$ be eigenfunctions of the autocorrelation inner product, and let $\beta_1, \beta_2, \ldots, \beta_{k^2-1}$ be the eigenvalues. Then the joint density function is

$$(\text{constant}) \cdot f(\frac{A_1^2}{\beta_1^2} + \cdots + \frac{A_{k^2-1}^2}{\beta_{k^2-1}^2}),$$

where

$$f(r) = \int_0^\infty \rho^{n-2} e^{-r\sqrt{1+\rho^2}} d\rho.$$

2. The most common values of the β_i are the eigenvalues of a symmetric form $<<,>>$ on V of the form

$$<< A, B >>= \sum_{ijkl} A_{ij} B_{kl} (1 + \gamma((i-k)^2 + (j-l)^2)^{.5})^{-\alpha}$$

relative to the symmetric form

$$< A, B >= \sum_{ij} A_{ij} B_{ij},$$

i.e. the solutions of

$$\det(<<>> -\beta <>) = 0.$$

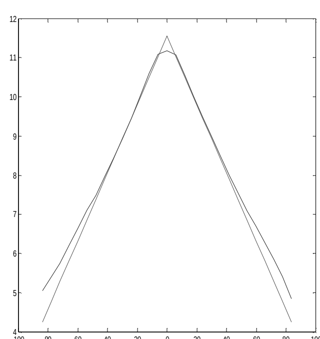
The point here is that any Differentially Laplacian density on $k \times k$ blocks must have the form in the first part of the model for some set of β_i, and that the observed form that the autocorrelation dieoff function takes makes the second part of the model the most likely–however, any choice of autocorrelation dieoff function is compatible with the first part of the model. The example given in this paper show that the first part of the model is applicable to data from a wide range of natural images, and also somedigital elevation, ocean temperature and precipitation data. It is my expectation that this model will be of wide applicability for many types of time-series and spatial data. Part 2 of the model is also widely applicable, although there are cases where 1 holds and 2 does not.

For a time series, one can break it up into blocks of size k and do the same procedure, getting $k-1$ eigenfunctions. We will do several examples for time series data with $k = 25$.

The autocorrelation dieoff function will be computed for horizontal shifts up to 50 pixels wide on a variety of images, and also for the time series data introduced.

The following image of a Mimosa tree, obtained from the web at http://www.forestryimages.org/browse/detail.cfm?imgnum=3694005 is shown here:

This is a good example from our perspective, because it is not a composite of disparate images, each of a different character, but is rather uniform over the entire picture. The histogram of the difference between horizontally adjacent pixel intensities is given by a Laplacian distribution, i.e. $f(x) = \frac{c}{2}e^{-c|x|}$.This fact has already been noted [H-M]. Here is a log histogram of the horizontal pixel differences for the Mimosa tree image, together with the best Laplacian fit:

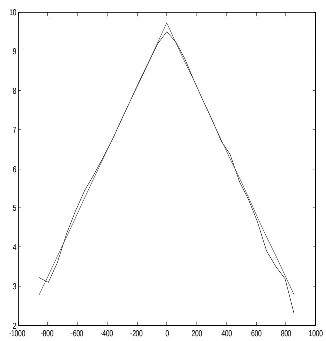


To rephrase the statement that the pixel intensities are Differentially Laplacian, if

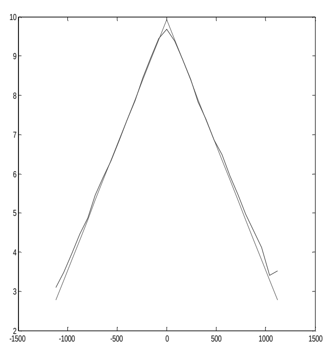
$A = (a_{ij})$ is a matrix, and X_{ij} represents the pixel intensity at a position shifted $i-1$ horizontally and and $j-1$ vertically from a reference position, then the variable $X_A = \sum_{i,j} a_{ij} X_{ij}$ has a Laplacian distribution is $\sum_{ij} a_{ij} = 0$. For example, for the matrix

$$A = \begin{pmatrix} -1 & -6 \\ 3 & 0 \\ 5 & -1 \end{pmatrix},$$

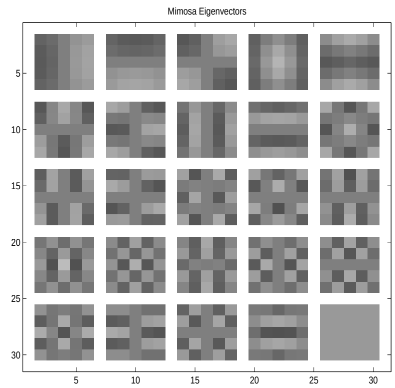
we get:



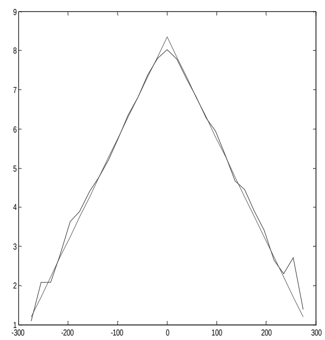
If we use $A = (\,-2 \quad 7 \quad 5 \quad -8 \quad -2\,)^t$, we get



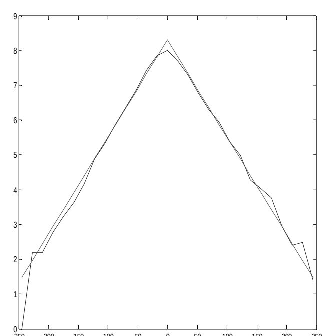
We now investigate all possible shifts over a 5×5 matrix of possibilities. There is a 24-dimensional space of possible linear combinations of pixel intensities over this 5×5 square. We look at the eigenvectors of the covariance matrix for these 24 combinations, and get 24 eigenvectors. A picture of the 24 eigenvectors plus the 5×5 matrix with $1/5$ in every position, which fills out the rest of the space of 5×5 matrices looks like:



If we break up the image into 5×5 blocks and look at the histogram for the coefficients of the 25 eigenvectors, these are once again Laplacian. For example, the first eigenvector has log histogram:

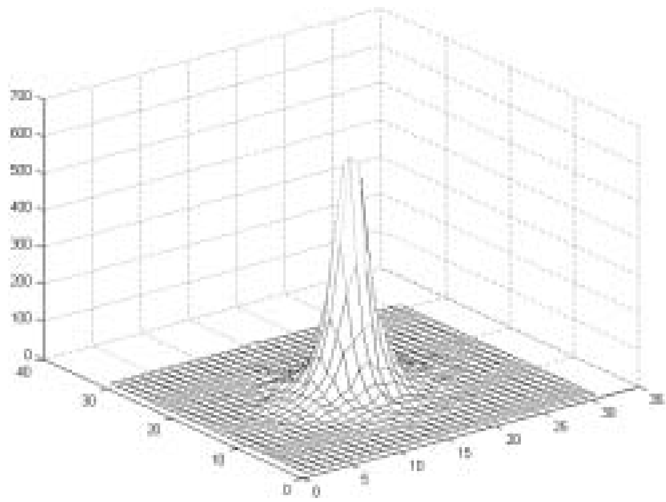


The second one has log histogram

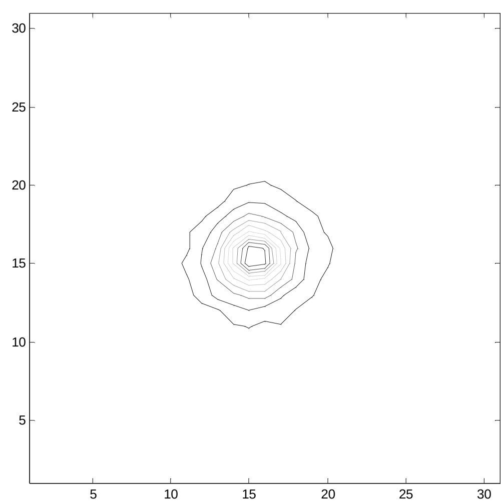


The others are similar. Log histograms for all 24 eigenvectors for several images are in the appendix.

If we graph the log of the joint density function of these two eigenvectors, we get

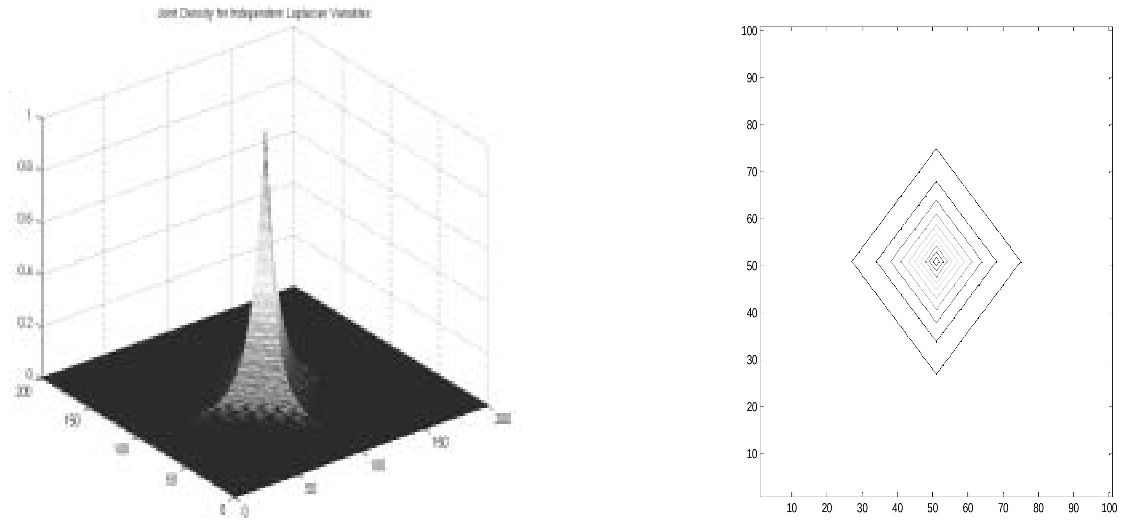


Here is a contour plot:



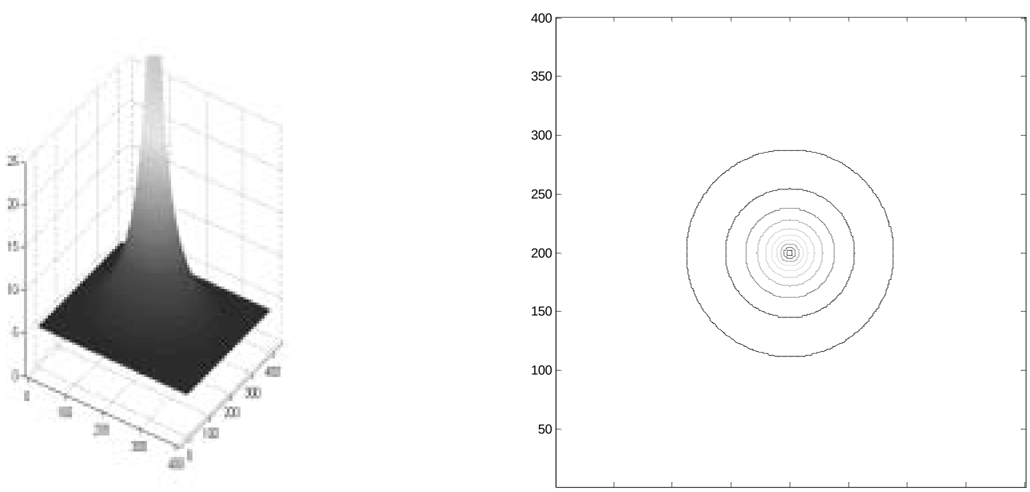
The axes have been adjusted by standard deviation. If these eigenfunctions were indeed independent, we would expect to get the following picture of the bivariate density and contour plot:

Theoretical Bivariate Density for Independent Laplacian Variables



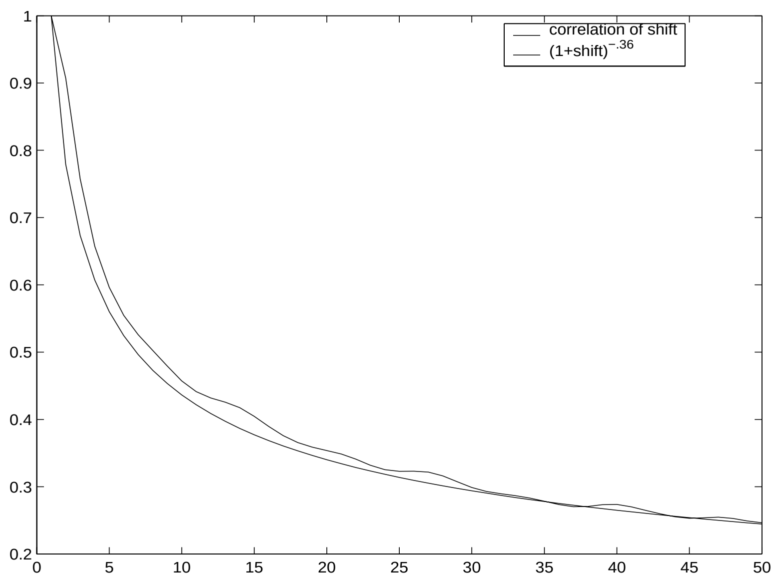
If instead, linear combinations of eigenfunctions remain Laplacian, as the model predicts, we would have:

Theoretical Bivariate Density for Linearly Laplacian Variables

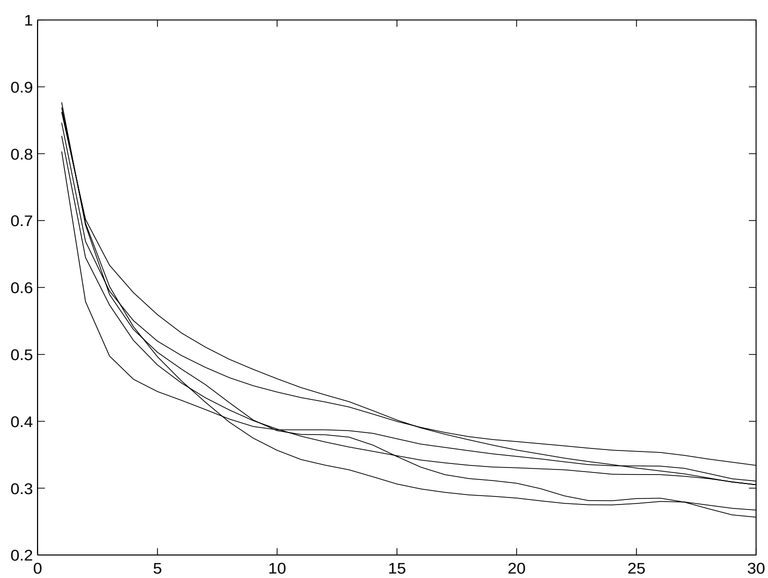


In order to give what we hope is adequate evidence that many images are Differentially Laplacian, in the appendix we give the Log Hist plots for the 24 eigenfunctions for 5×5 sub-blocks of a variety of images and also two digital elevation models, as well as for length 25 sub-blocks of selected time series of temperature and precipitation. We also give the bivariate density and contour plots for all pairs of the first 7 eigenvalues of one image. We have evidence for sub-blocks as large as 30×30 for some images, but this cannot be presented here.

We now turn to the autocorrelation dieoff fucntion. It is useful to look at the correlation of pixel intensities under a variable horizontal shift of from 1 to 49 units–we graph the coorelation against the shift plus 1 for the Mimosa tree image.



We compare it against $(1+\text{shift})^{-.36}$. If we make a similar graph for 6 pictures, it looks like



We may combine all of the above experimental results into the following:

OBSERVATION. *The variable X_A associated to the matrix A whose entries sum to 0 has a Laplacian distribution whose variance is given by*

$$\|A\|_B^2,$$

where B is a symmetric positive definite metric on the space of such matrices given by the formula

$$\|A\|_B^2 = \sum_{iji'j'} c a_{ij} a_{i'j'} (1 + \gamma\sqrt{(i-i')^2 + (j-j')^2})^{-\alpha},$$

where c, α and γ are parameters of the image ($\alpha = .36$, $\gamma = 1$ in the case of the Mimosa image).

I have investigated other natural images, and many of them have roughly similar properties. Of course, none is an absolutely perfect match. I hope that I am very clear that I am not claiming that this is a universal joint density function for pixel intensities in images. This is not my intention. A famous dictum I learned from several statisticians is that "All statistical models are wrong, but only some are useful." I do find it impressive that a model as highly specific as this one is valid, even to a range of 5 pixels, in even one actual image, much less in many. For many images, one gets similar results for sub-blocks ranging from 10×10 to 30×30 pixels. My hope is that this is a useful model for describing at least one type of spatiotemporal variation–that which occurs in certain natural images. In section 2, an extremely simplified version of this model will be seen to form the basis for the celebrated TV denoising algorithm of Rudin-Osher-Fatemi [R-O-F], and in section 3 a more faithful way of simplifying the model will lead to a highly effective denoising algorithm for natural images.

There have been a number of earlier investigations into the statistics of natural images. The fact that adjacent pixel intensities are given by a Laplacian density is well-known, and I am not sure who discovered it first. There was some preliminary investigation by Mumford of the joint density of two pixel diferences. I have not seen the observation that linear combinations of pixel differences always have a Laplacian density, and relating the parameters of the Laplacian densities using the formula for $\|A\|_B$.

2. A Statistical Explanation of the TV Algorithm of Rudin-Osher-Fatemi

The discussion in section 1 was empirical in character, based on observation of a particular set of images. It represents an attempt to extract a comparatively simple model from a complex situation. In this section, we enter the realm of Bayesian statistics, armed with the empirical knowledge gained in the first section.

To get started, let $\mathcal{F}$ be a space of functions with a probability measure $d\phi$ on $\mathcal{F}$. We will assume

$$\mathcal{F} \subseteq \{f : \Lambda \to \mathbf{R}\}$$

is a subset of the real-valued functions on a bounded grid $\Lambda \subset \mathbf{R}^n$.

PROBLEM (Image Restoration Data with Gaussian White Noise–Bayesian Formulation). *Let X be a random variable on $\mathcal{F}$ chosen using the probability measure $d\phi$. Each value $X(x)$ for $x \in \Lambda$ is blurred by an independent Gaussian random variable $G(x)$ using a Gaussian distribution $N_{0,\sigma}$ with mean 0 and standard deviation σ. The image we are given to process is a random variable F on $\mathcal{F}$ with intensity*

$$Y(x) = X(x) + G(x)$$

at each $x \in \Lambda$. The problem is to restore X as closely as possible given that we know Y.

REMARK.

We think of Λ as the set of pixels and $\mathcal{F}$ as the set of possible image intensities we might be called upon to restore. For a given image, $X(x)$ is the **true image intensity** at pixel x and $Y(x)$ is the **measured image intensity** at pixel x. The signal in each pixel is blurred independently by Gaussian noise, as might happen if the image were sent pixel-by-pixel over a noisy line.

The probability density $d\phi$ on $\mathcal{F}$ defines how likely a given collection of pixel intensities is to appear as an image. Such a probability distribution on the set of all possible images is known in the world of Bayesian statistics as a "prior." We are then asking–among all possible original pictures (this is the space $\mathcal{F}$), which is the likeliest one to have given rise to the observed picture? We take into account both the relative likelihood of the original picture and the likelihood of that original picture giving rise to the observed picture when subjected to digital Gaussian white noise with standard deviation σ.

Of course, such a prior is only an idealization, which might apply with reasonable accuracy to some class of naturally occurring images, but which cannot apply in a procrustean fashion to every image in existence. Section 1 gives the lineaments of a reasonable prior to use. In applications, it is enough to have a prior for just part of the information in a picture, e.g. for $k \times k$ sub-blocks of an image

We let $N_{\mu,\sigma}(x)$ denote the normal density function with mean μ and standard deviation σ. We begin by noting that

$$P(Y(x) = y | X = g) = N_{g(x),\sigma}(y) = N_{0,\sigma}(y - g(x)).$$

Thus

$$P(Y(x) = y \text{ and } X = g) \text{ has probability density } N_{0,\sigma}(y - g(x))d\phi(g) \text{ on } \mathcal{F}.$$

Similarly,

$$\begin{aligned} P(Y = f \text{ and } X = g) \text{ has probability density } &\prod_{x\in\Lambda} N_{0,\sigma}(f(x) - g(x))d\phi(g) \\ &= e^{-\sum_{x\in\Lambda} \frac{1}{2\sigma^2}(f(x)-g(x))^2} d\phi(g). \end{aligned}$$

If we enumerate Λ as a set by

$$\Lambda = \{x_1, \ldots, x_M\},$$

then if $g_i = g(x_i)$, we write

$$d\phi = \Phi(g) dg_1 \cdots dg_M.$$

Adopting the notation of statistical mechanics,

$$\Phi(g) = \Psi(\beta) e^{-\beta E(g)},$$

where β is a parameter that would be the inverse of the temperature, and $\Psi(\beta)$ is the partition function, chosen to make the total probability come out to be 1. Then

$P(Y = f$ and $X = g)$ has density function $\Psi(\beta) e^{-\beta E(g) - \sum_{x \in \Lambda} \frac{1}{2\sigma^2}(f(x)-g(x))^2} dg_1 \cdots dg_M$.

We therefore have:

PROPOSITION. *For the problem of image restoration to eliminate Gaussian white noise, the maximum likelihood estimate for the true pixel intensity function g given the measured pixel intensity function f is obtained by choosing g to minimize the functional*

$$\sum_{x \in \Lambda} \frac{1}{2\sigma^2}(f(x) - g(x))^2 + \beta E(g).$$

REMARK. If we are in n dimensions and normalizing the size of the mesh to be 1 for the square lattice Λ, then we may replace this functional by

$$\int_x (f(x) - g(x))^2 dx + 2\sigma^2 \beta E(g).$$

The TV algorithm [R-O-F] consists of minimizing the functional

$$\int_x (f(x) - g(x))^2 dx + \lambda TV(g),$$

where λ is a parameter to be chosen and TV is the total variation

$$TV(g) = \int_x \|\vec{\nabla} g\| dx.$$

We therefore conclude that, up to a constant, we the TV algorithm is using the energy

$$E(g) = 2\sigma^2 TV(g) = 2\sigma^2 \sum_{x \in \Lambda} \|\vec{\nabla} g(x)\|.$$

For simplicity, let us now take Λ to be 1-dimensional. Let

$$\Lambda = \{x_1, \ldots, x_N\}$$

and $g_i = g(x_i)$. We discretize

$$\|\vec{\nabla} g(x_i)\| = (g(x_{i+1}) - g(x_i)).$$

Let $y_i = g_{i+1} - g_i$. Note that

$$dg_1 dg_2 \cdots dg_N = dy_1 \cdots dy_{N-1} d\bar{g},$$

where $\bar{g} = (1/N)(g_1 + \cdots + g_N)$. The variable $\bar{g}$ behaves differently than the others, and we integrate it out to get a measure in $dy_1 \cdots dy_{N-1}$. It follows that the measure on $\mathcal{F}$ with $\bar{g}$ integrated out is

$$d\phi(g) = (\prod_{i=1}^{N-1} \frac{\beta}{2} e^{-\beta |y_i|} dy_i).$$

We therefore have:

PROPOSITION. *For Gaussian white noise in one dimension, the TV algorithm is the maximum likelihood estimate for the density which takes adjacent pixel differences $X_{i+1} - X_i$ to be independent and distributed according to a Laplacian or absolute exponential density*

$$\frac{\beta}{2} e^{-\beta|x|}.$$

The parameter λ is determined by β and σ by the relationship

$$\lambda = 2\beta\sigma^2.$$

REMARK. It is possible to make a reasonable estimate of β and σ from the measured distribution of adjacent pixel differences in the observed image. If D is the random variable representing pixel differences, from $E(D^2)$ and $E(D^4)$ one can solve for β and σ; alternatively, one can fit the convolution of a Laplacian and a Gaussian with unknown parameters β, σ to the measured distribution of D. It is thus possible to determine the correct λ to use. Determining in advance the right λ to use was an unsolved practical issue in applying the TV algorithm.

The purpose of this proposition is not to claim that this prior probability distribution using the total variation is accurate—it is not. The point is rather to bring to light what assumptions underlie the TV algorithm, so that one can make improvements based on a better knowledge of what the correct prior distribution is for the problem one is trying to solve.

3. A Statistically-Motivated Denoising Algorithm

We first note the following

LEMMA. *Let X be a Laplacian random variable with parameter β and G a Gaussian random variable with mean 0 and standard deviation σ. If X, G are independent and $Y = X + G$, then the maximum likelihood estimate for X given the value of Y is*

$$x = \begin{cases} y - \beta\sigma^2 & \text{if } y > \beta\sigma^2; \\ 0 & \text{if } -\beta\sigma^2 \le y \le \beta\sigma^2; \\ y + \beta\sigma^2 & \text{if } y < -\beta\sigma^2. \end{cases}$$

Note that this lemma is of course related to the concept of soft thresholding of Donoho et al [D-J-K-P].
PROOF: We want to maximize

$$P(X = x|Y = y) = (\text{const}) e^{-\frac{(x-y)^2}{2\sigma^2} - \beta|x|}$$

or equivalently to minimize

$$(x - y)^2 + 2\beta\sigma^2|x|.$$

If $x > 0$, the derivative with respect to x is

$$2(x - y) + 2\beta\sigma^2,$$

so

$$x = y - \beta\sigma^2.$$

If $x < 0$, one has $x = y + \beta\sigma^2$. The result follows.
REMARKS:
1. It is worth noting that in the proof, what is needed is that the log density for X is $\beta|x|$. For an algorithm based on this Lemma to work in practice, the log histogram of X needs to be well-approximated by the log histogram of a Laplacian random variable. Inspection of the log histograms in Section 1 and in the appendix confirm that this is indeed a good approximation for many real images.
2. If instead X is a Gaussian random variable with mean 0 and standard deviation τ, and G is as in the Lemma, then the maximum likelihood estimate is

$$x = \frac{\tau^2}{\tau^2 + \sigma^2} y,$$

which is what comes up in a Wiener filter. This is the wrong thing to use when X is a Laplacian random variable.

We now adopt the following assumptions about the joint density function of the pixel intensities of our original image:
1. For k odd, the $k^2 - 1$ eigenvectors of the covariance matrix of the image, taken as linear combinations of the pixel intensities, have Laplacian densities.
2. These $k^2 - 1$ random variables are independent.
3. The values of these eigenvectors are independent of each other on different, non-overlapping $k \times k$ blocks.

Of these assumptions, the first two are supported by the evidence in section 1. The third assumption is false–it is made only to get us down to a workable situation. As k gets large, it becomes more and more reasonable. In practice, we tend to take k to be 5, 7, or 9. The idea is that we make use of the fact that pixels comparatively nearby to each other are strongly correlated in value, and since the noise added to each pixel is independent, the noise tends to throw off the statistics of the $k \times k$ block, and thus can be detected and removed.

We now assume that we are given $Y = X + N$, where X is the true image, N is white Gaussian noise, with mean 0 and known standard deviation σ, independent for each pixel. The object is to restore X from Y as closely as possible.
Step 1: Using the statistics of Y and σ, we obtain good estimates for the coefficients of the $k^2 - 1$ eigenvectors for the image X, and their variances. Under assumptions 1,2 and 3, we now know that they have independent Laplacian densities with known parameters. (If we did not make assumption 3, we could do even better by exploiting the relationships between nearby values, but we can do this anyway by increasing k.)
Step 2: Using these coefficients, we now compute the values of these linear combinations of pixel intensities over each $k \times k$ block, and then apply the Lemma to make a maximum likelihood estimate for what these coefficients were before the noise was introduced.
Step 3: The picture is now reconstructed by averaging the intensity at each pixel obtained from the coefficients in Step 2, averaging over every $k \times k$ block containing that pixel.

One nice feature of this algorithm is that it is a one-pass algorithm and does not require iteration. It scales linearly with the number of pixels.

Some examples of this algorithm using 5×5 blocks are contained in the appendix. Because a certain level of noise can be hidden in the natural variation of an image, a visually optimal denoising is produced by only going part of the way (i.e. using $c\beta$ instead of β, where $.25 \leq c \leq .5$.) This has been done for the denoising examples in the appendix.

Bibliography

[D-J-K-P] Donoho, D. L., Johnstone, I. M., Kerkyacharian, G. and Picard, D., "Wavelet shrinkage: Asymptopia? (with discussion)", Journal of the Royal Statistical Society Series B 57, 301–369. 1995.

[H-M] J. Huang and D. Mumford, "Statistics of Natural Images and Models."

[Hu] J. Huang, "Statistics of Natural Images and Models," Thesis, Brown University (2000).

[H] A. Hyvärinen, "Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation," Neural Computation, 11(7):1739–1768, 1999.

[L-P-M] A. Lee, K. Pedersen, D. Mumford, "The nonlinear statistics of high-contrast patches in natural images," (2001).

[M] Mandelbrot, Benoit B., *Multifractals and 1/f Noise: Wild Self-Affinity in Physics*, Springer-Verlag 1999.

[R-O-F] Rudin, L. I., Osher, S. and Fatemi, E., "Nonlinear total variation based noise removal algorithms," Physica D, Vol. 60, pp. 259-268, 1992.

[S-L-S-Z] A. Srivastava, A. Lee, E. Simoncelli, S.-C. Zhu, "On advances in statistical modeling of natural images," (2002).

[W-S-S] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," IEEE Trans. Image Processing 9, pp 1309-1324 (2000).

Image and Data Sources

University of Georgia Forestry Database
http:www.forestryimages.org/ for Salvina, Mimosa, Wildebeest, White-Tailed Deer, Bamboo, Lava, Scorpion, Forest Scene

2MASS Atlas Image Mosaic
http: www.ipac.caltech.edu/2mass/gallery/ for NGC6992

USGS Digital Elevation Model data
http://edc.usgs.gov/glis/hyper/guide/usgs_dem for Rocky Mountain and Australia DEM's

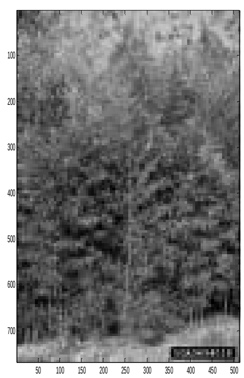
National Ocean Data Center Database
http://www.nodc.noaa.gov/General/getdata.html for hourly water temperature data

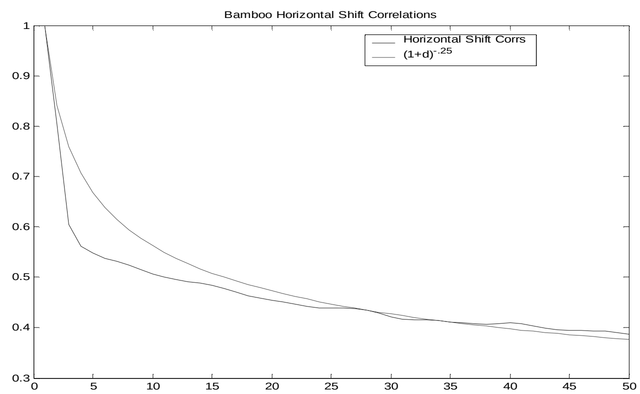
N-Hydrology
http://www-personal.buseco.monash.edu.au/ hyndman/TSDL/htong/precip.dat for Hveravellir daily precipitation data, from Hipel and Mcleod 1994

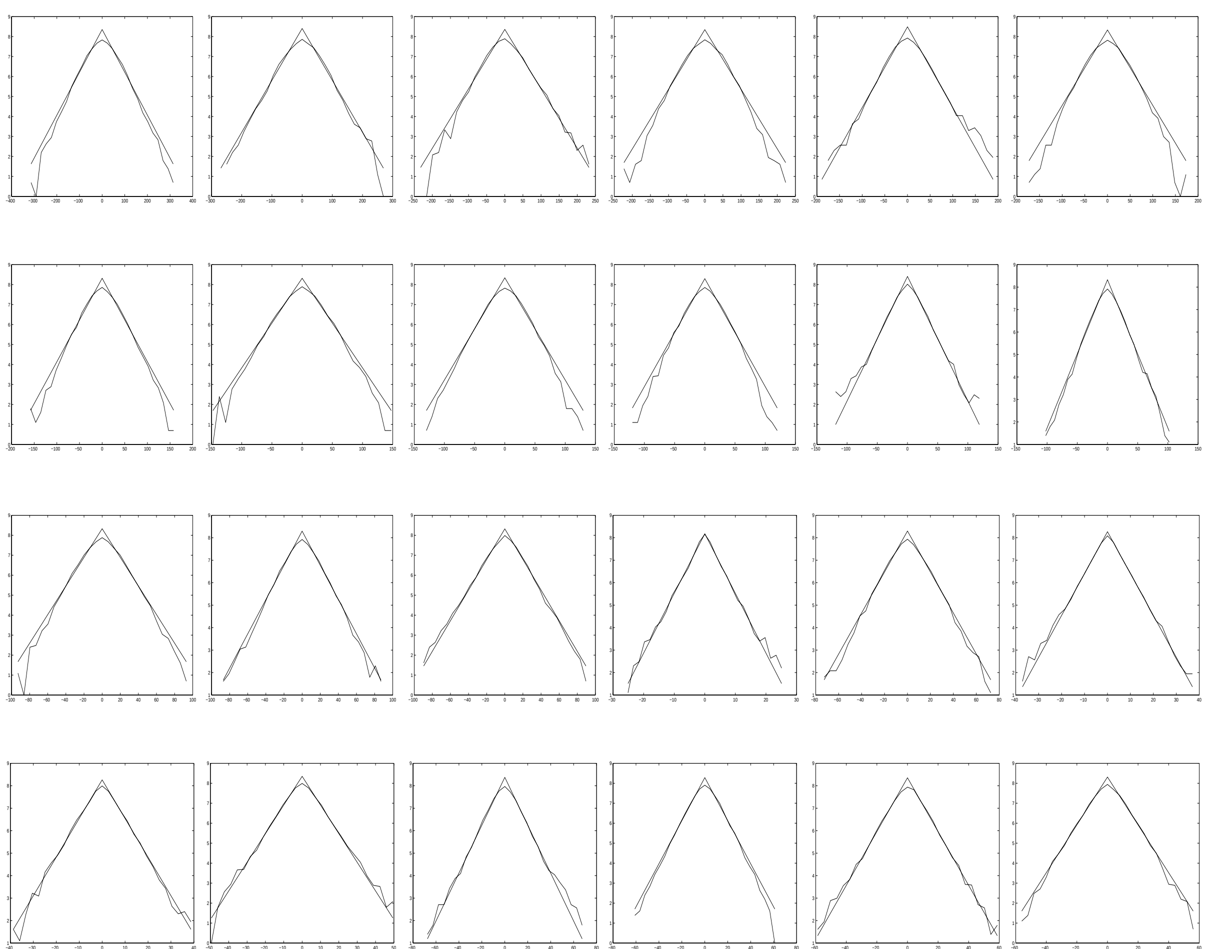
Author's address:
Department of Mathematics, UCLA, Los Angeles, CA 90095
mlg@math.ucla.edu

APPENDIX: SUPPORTING DATA

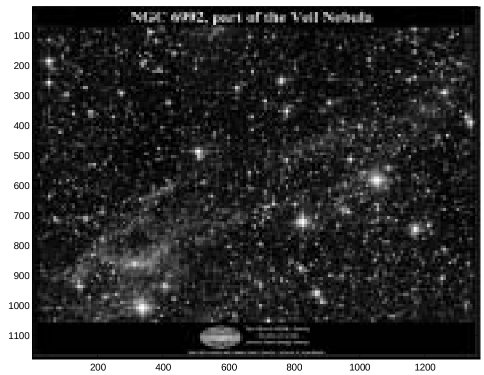
Bamboo Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks

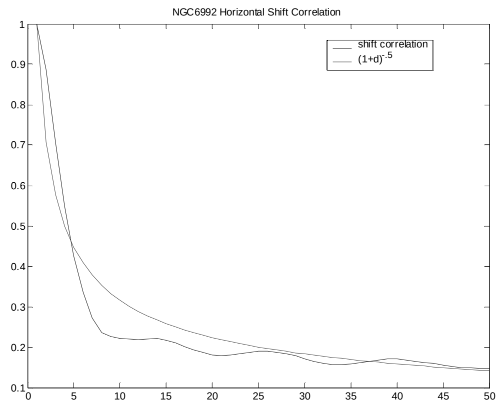


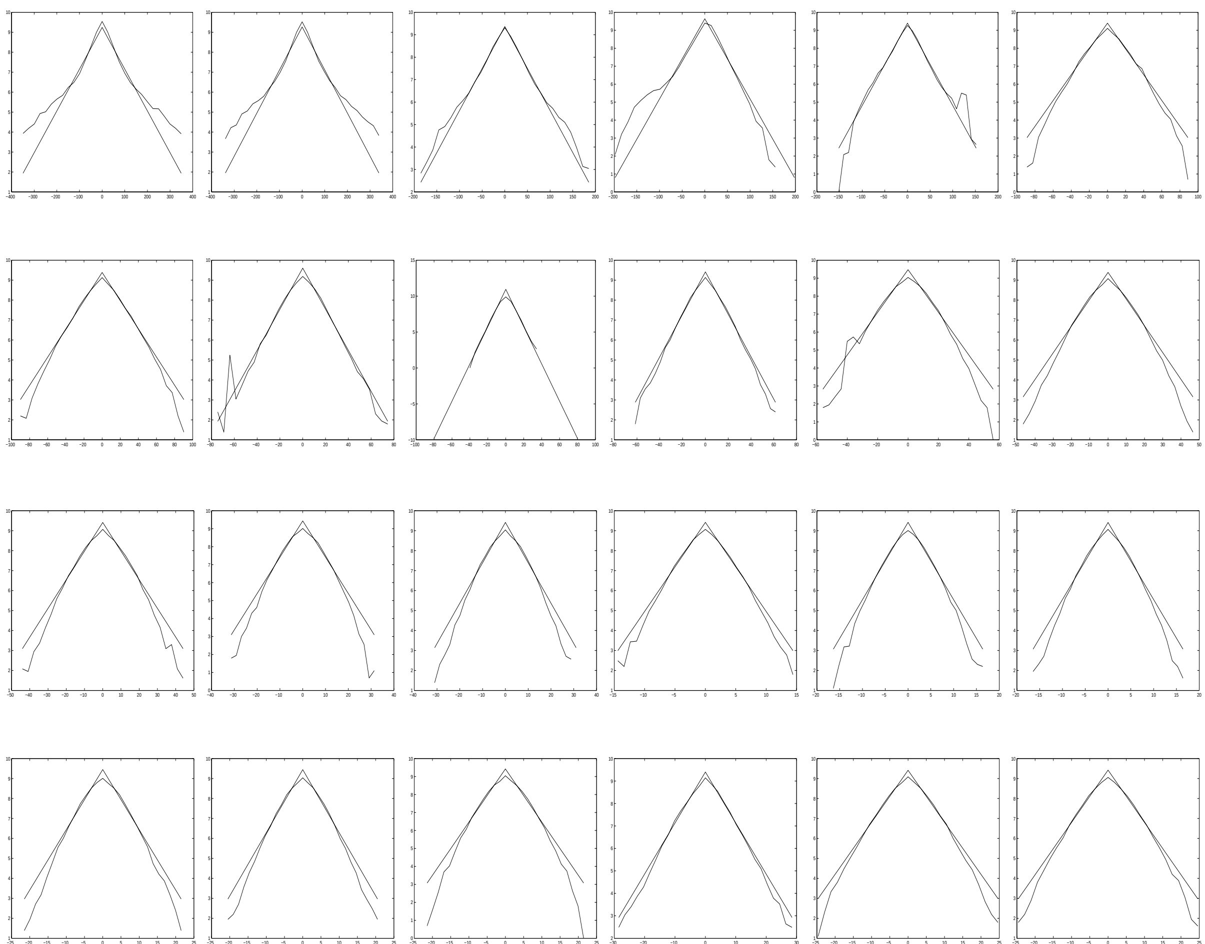




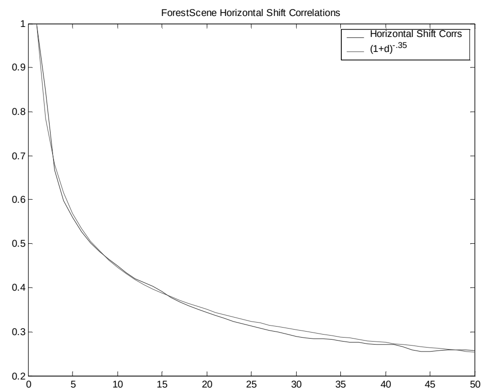
NGC6992 Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks

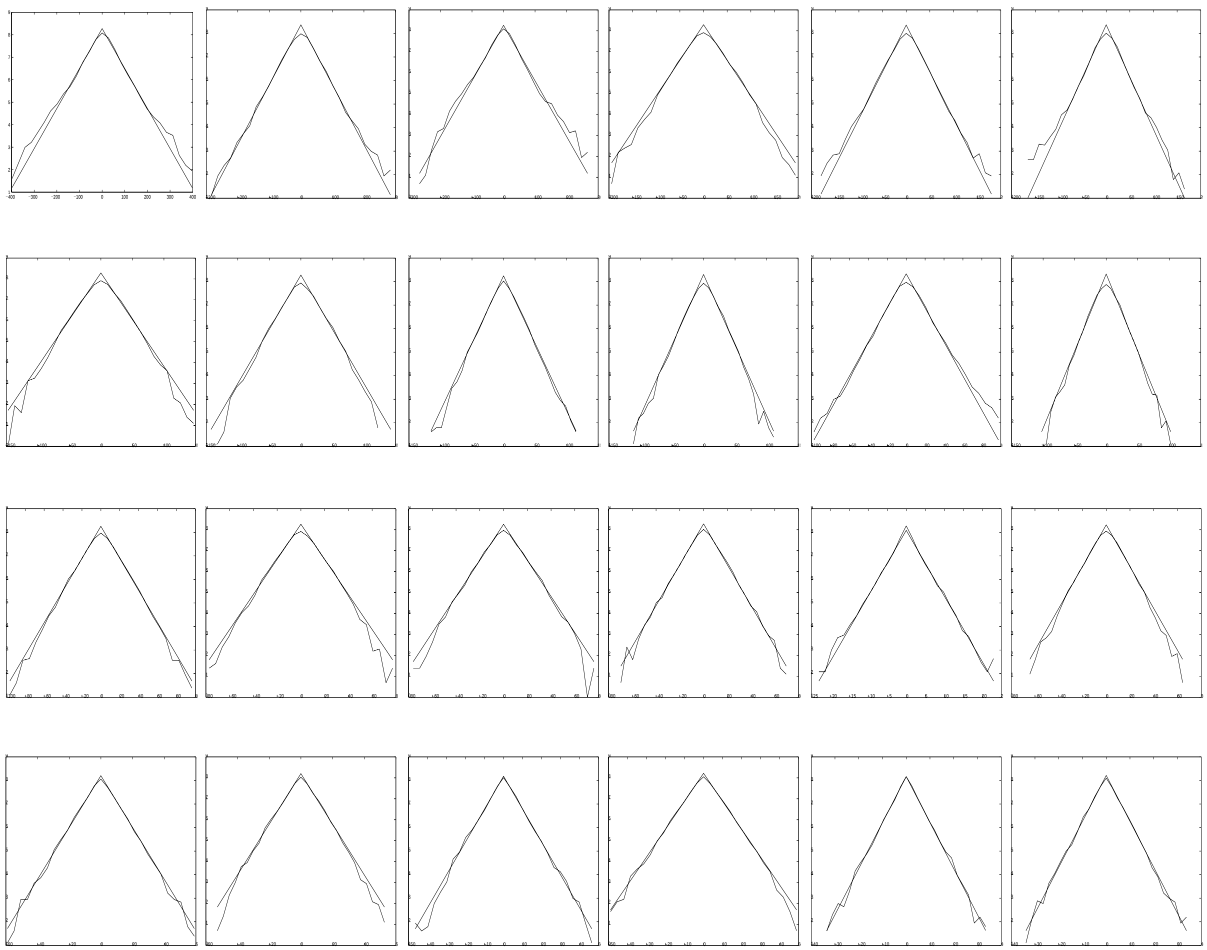




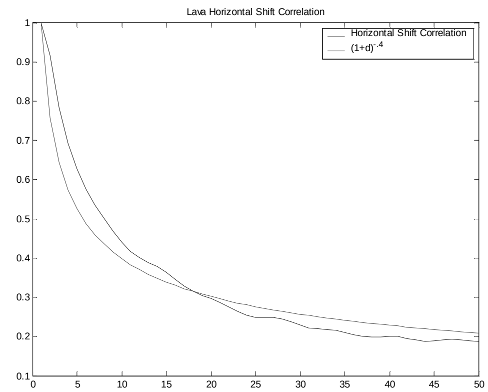


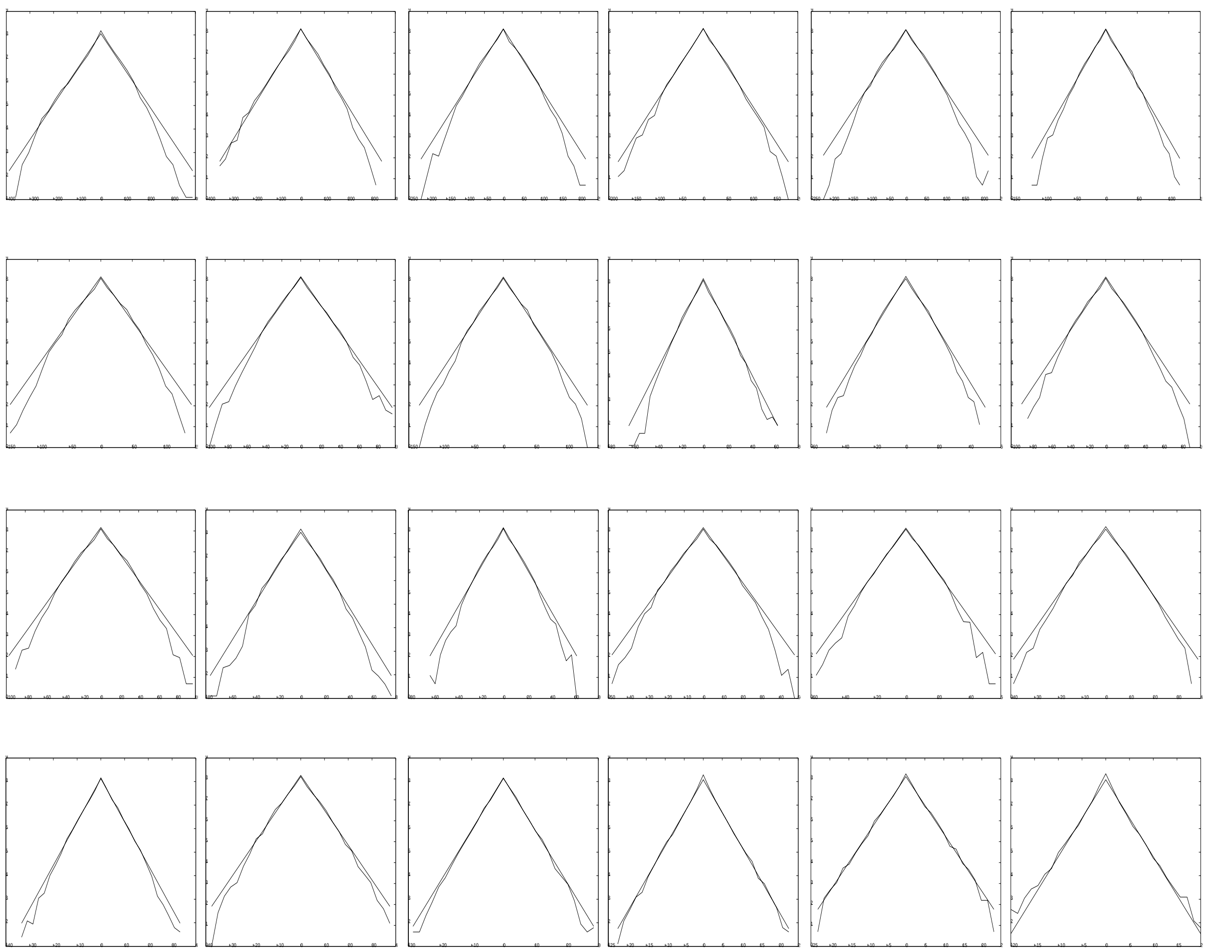
Forest Scene Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks



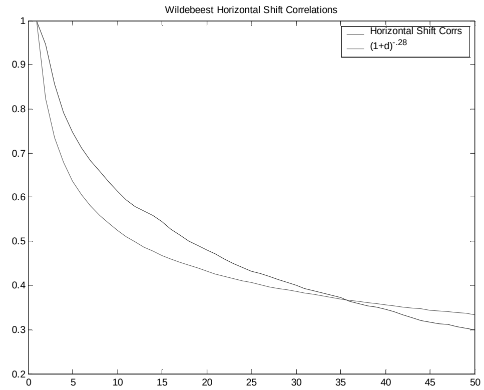


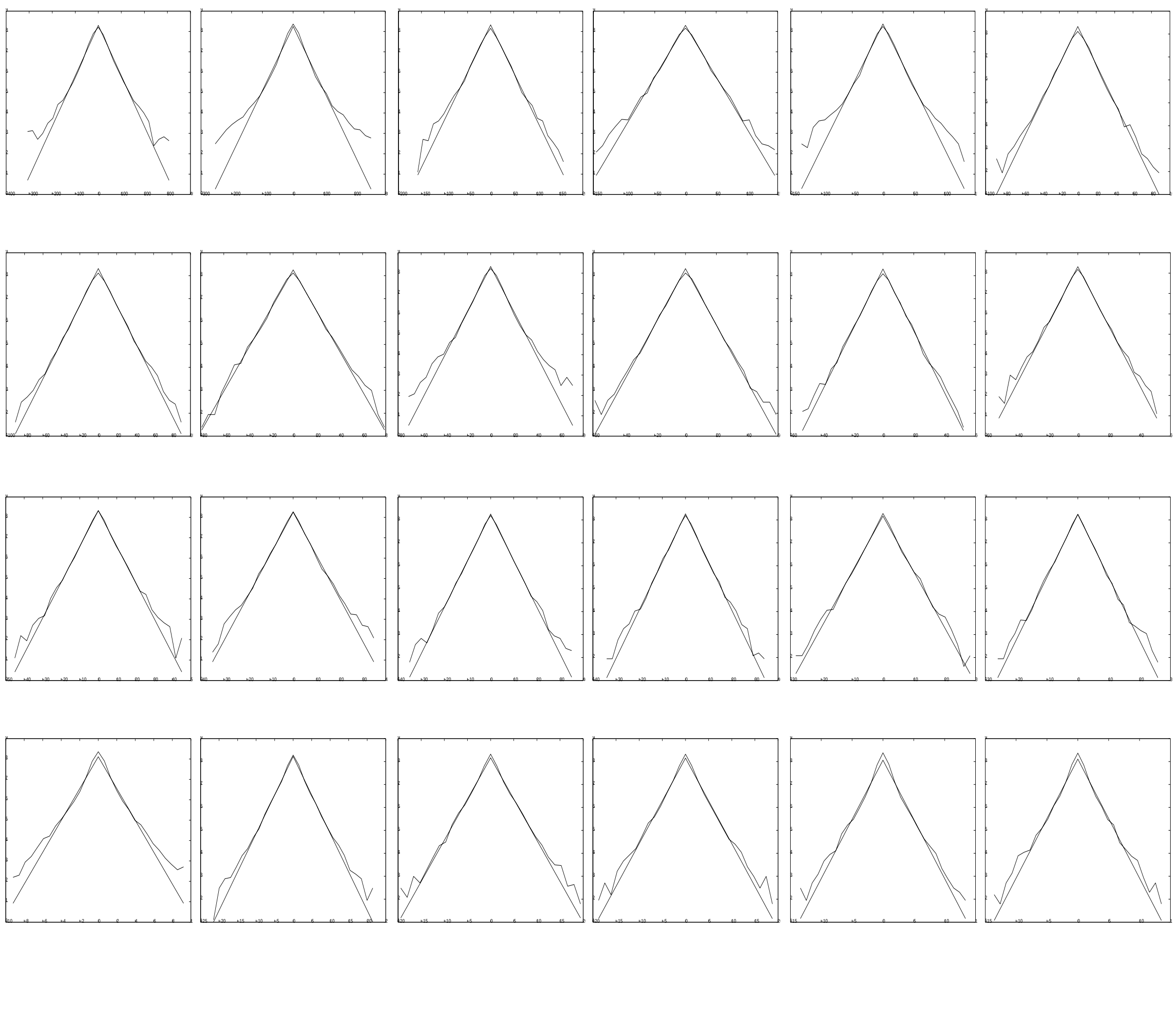
Lava Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks



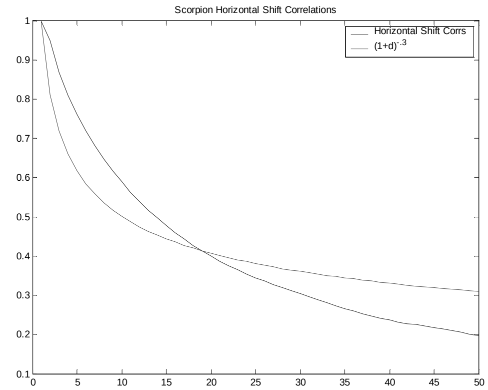


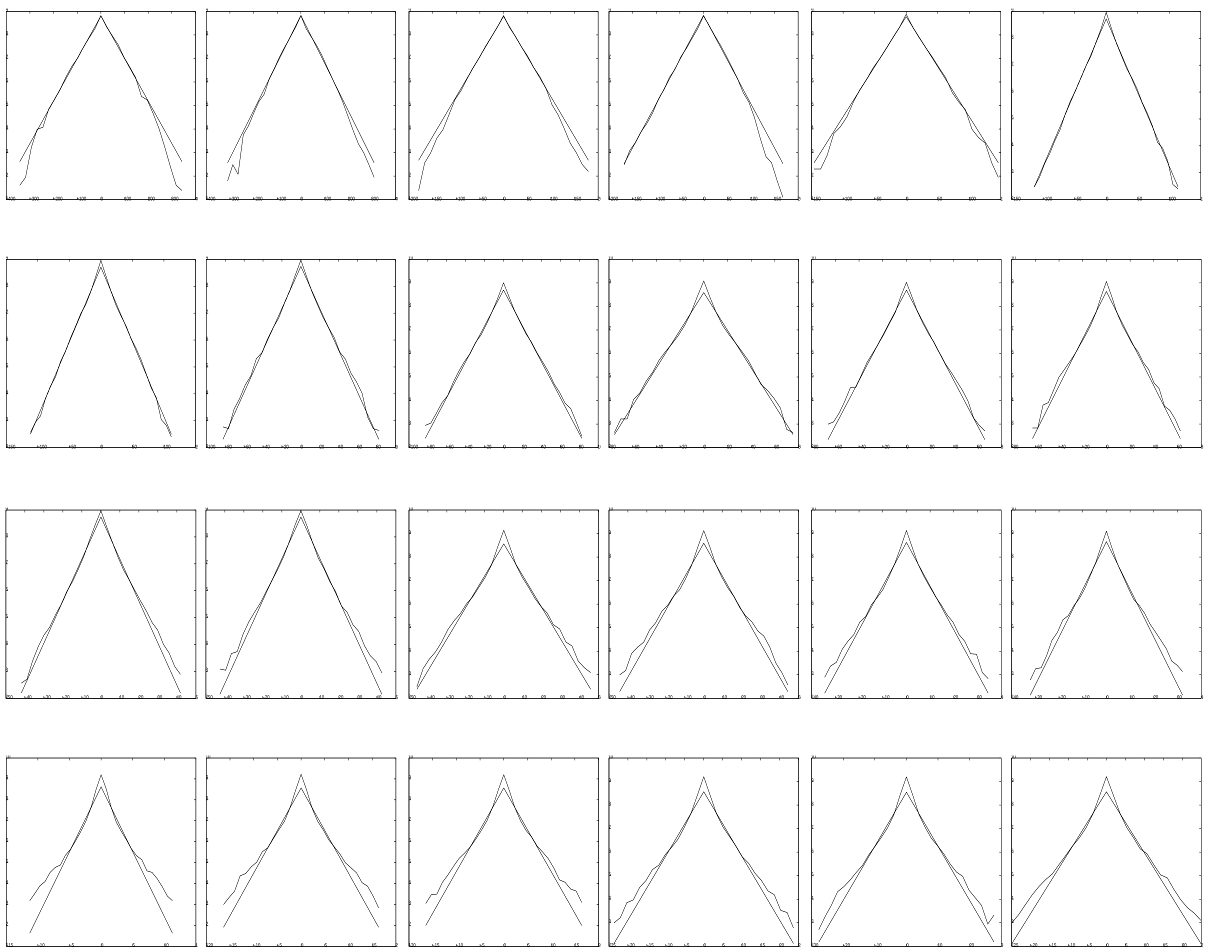
Wildebeest Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks



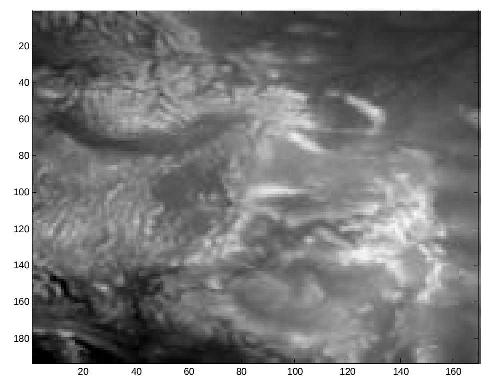


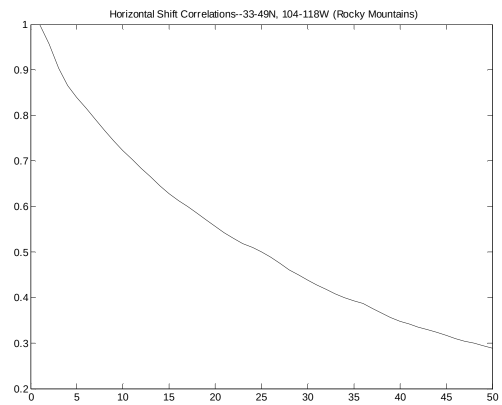
Scorpion Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks

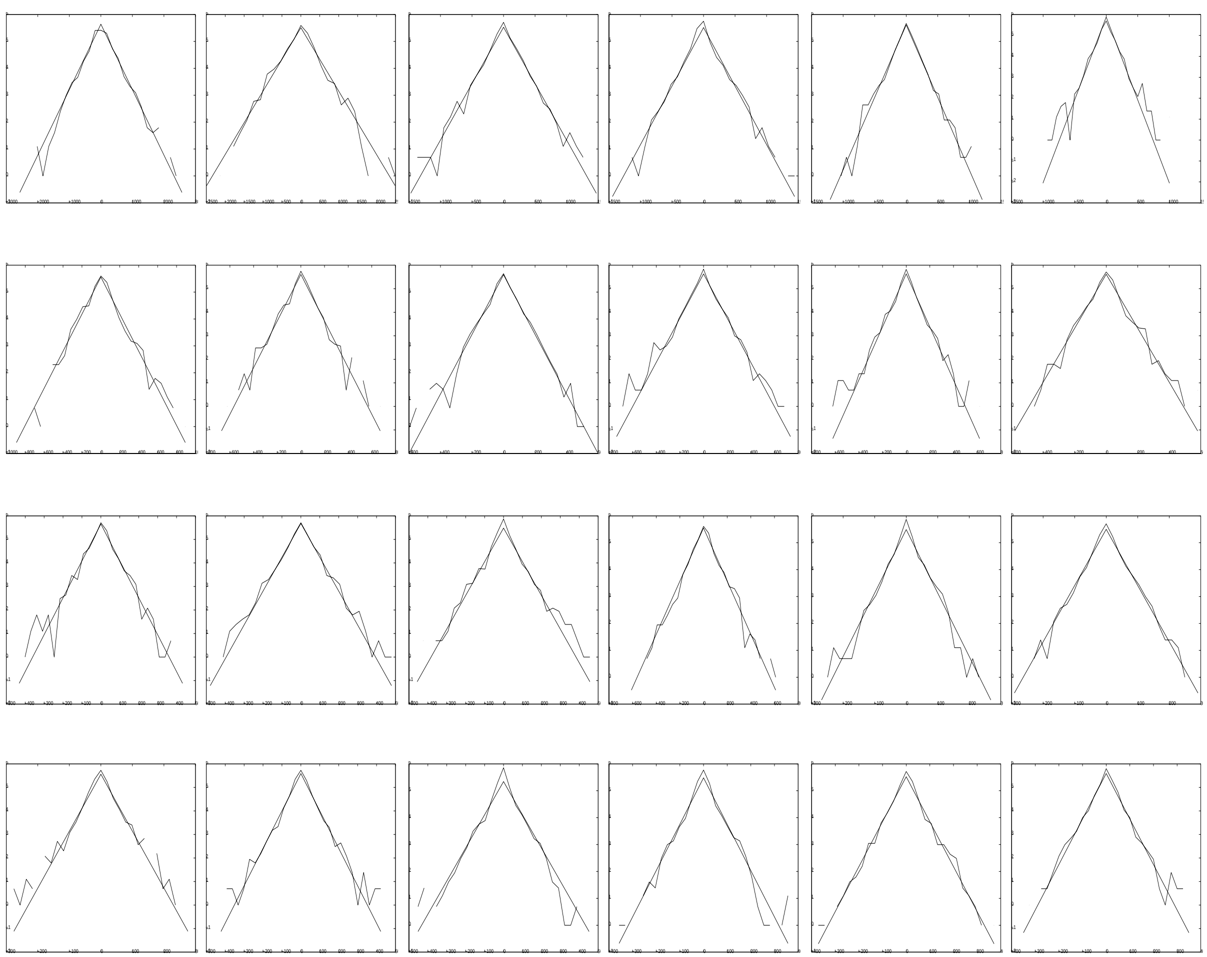




Digital Elevation Map Rocky Mountains 33-49N, 104-118W
5 Minute Grid Values
Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks



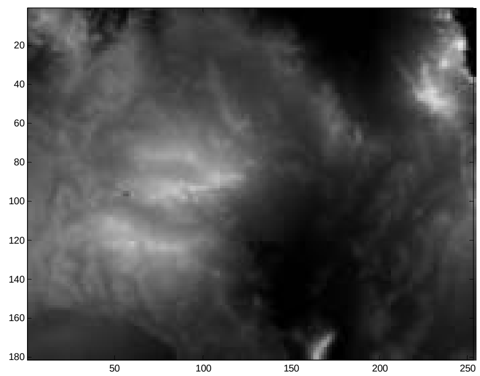


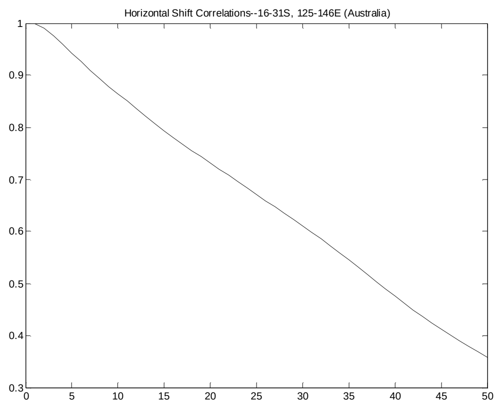


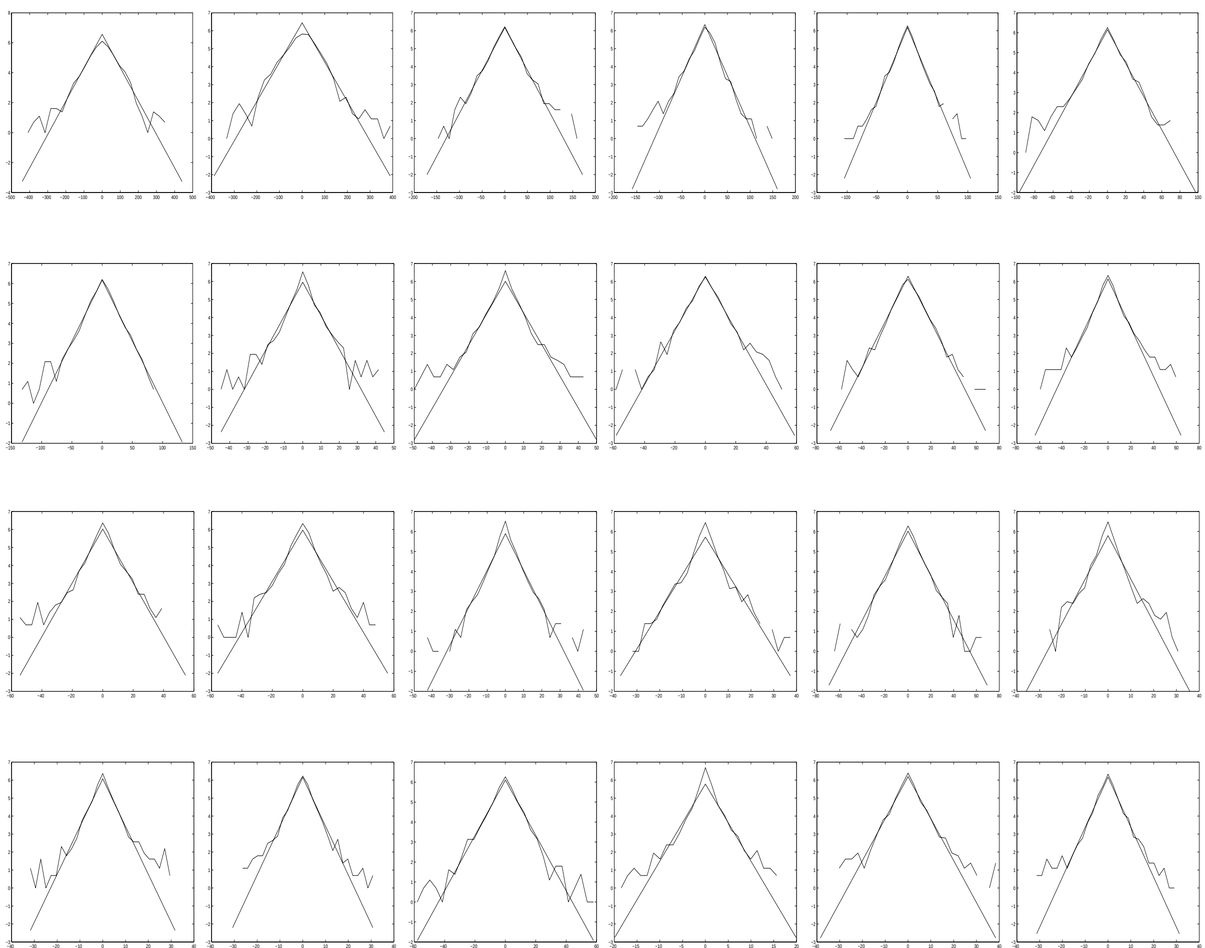
Digital Elevation Map Australia 16-31S, 125-146E

5 Minute Grid Values

Log Histograms for 24 Eigenfunctions of 5×5 Sub-blocks



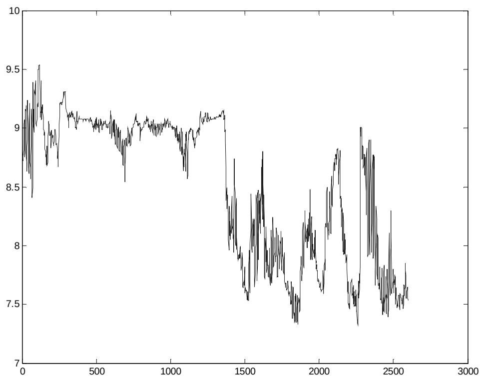


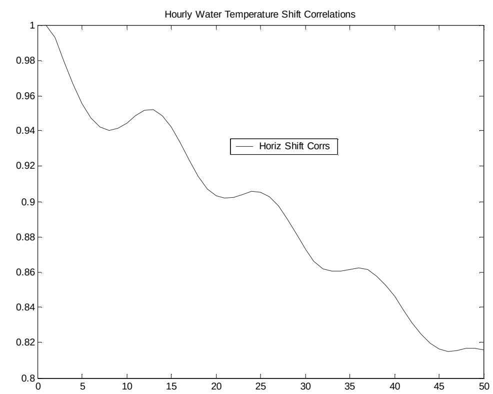


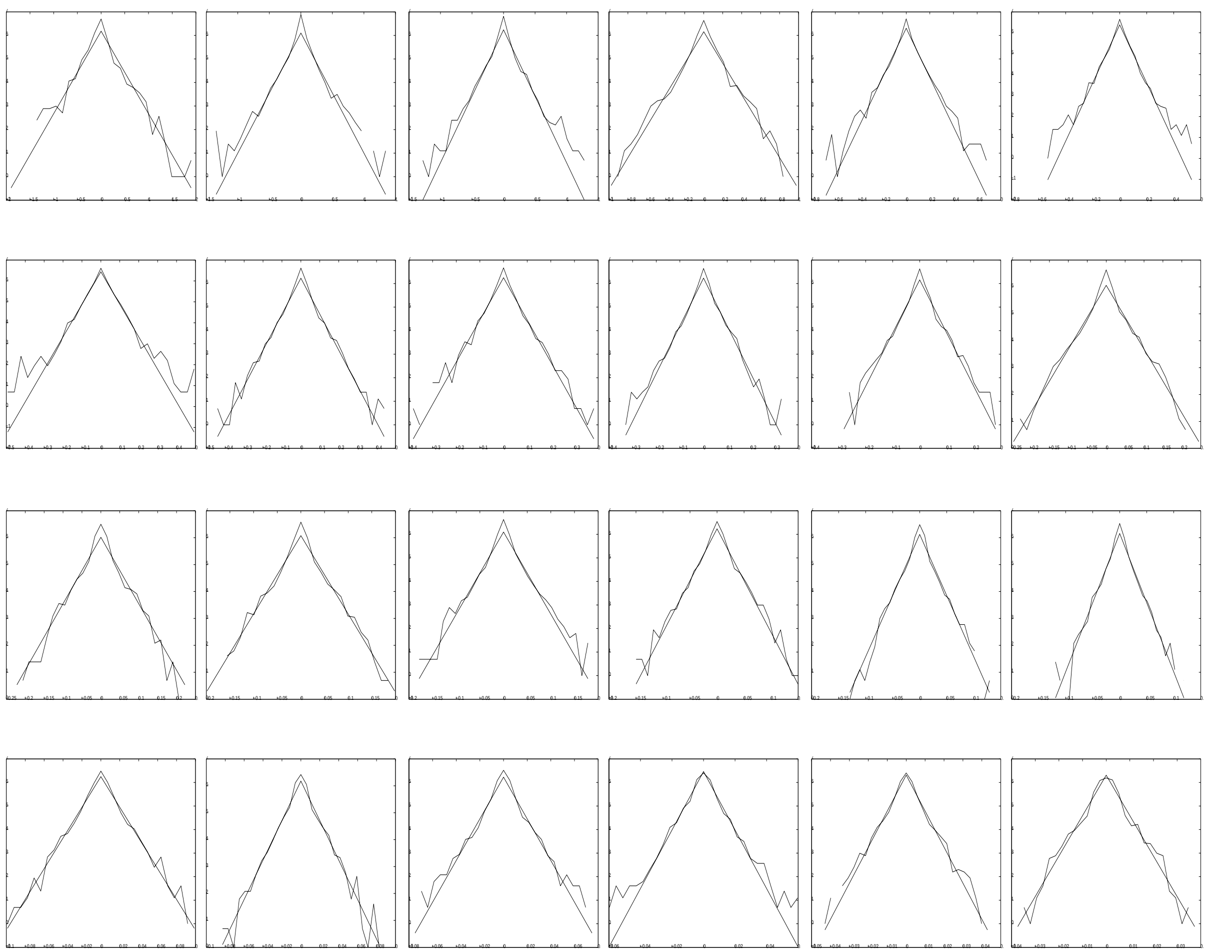
Hourly WaterTemperatures Station Location: 45 00’18”N, 124 05’06”W

Start Date=1975/01/28, Start Time= 09:00GMT
End Date=1975/05/16, End Time= 14:00GMT,Obs. Depth= 53 M

Log Histograms for 24 Eigenfunctions of 25 Hour Sub-blocks



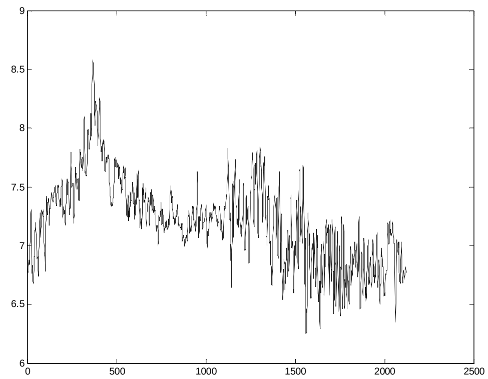


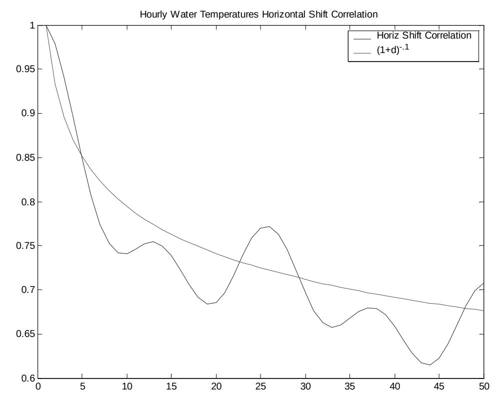


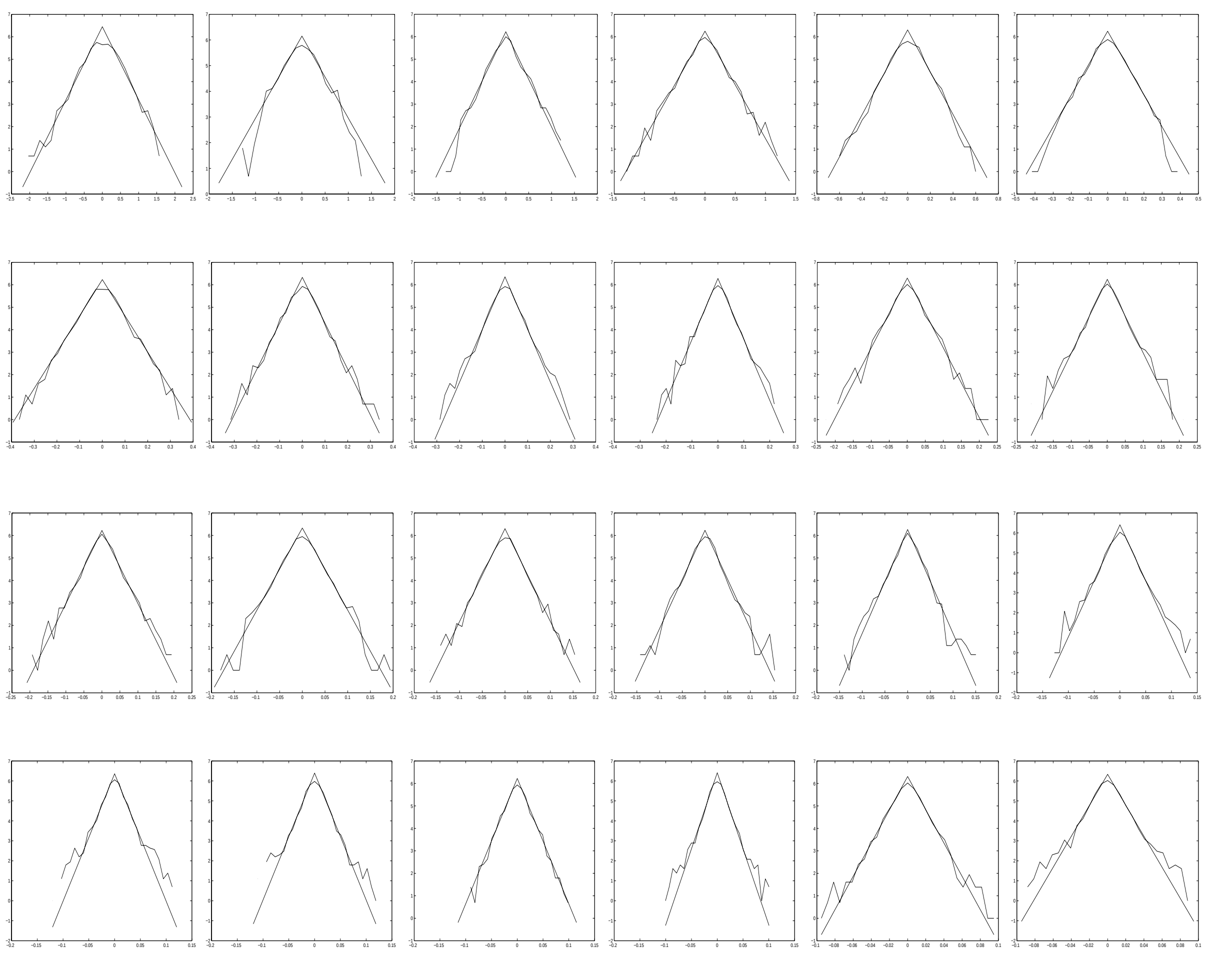
Hourly WaterTemperatures Station Location: Station Location:45 00'12"N, 124 23'00"W

Start Date=1975/01/28, Start Time= 07:00GMT
End Date=1975/04/26, End Time= 16:00GMT, Obs. Depth= 206 M

Log Histograms for 24 Eigenfunctions of 25 Hour Sub-blocks



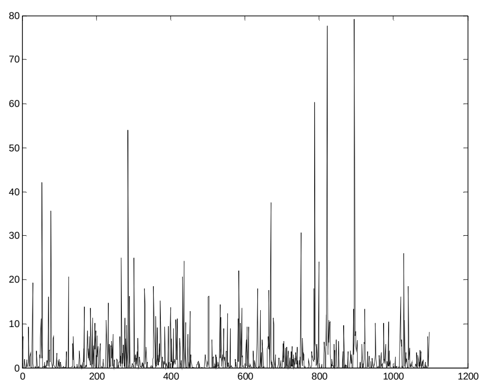


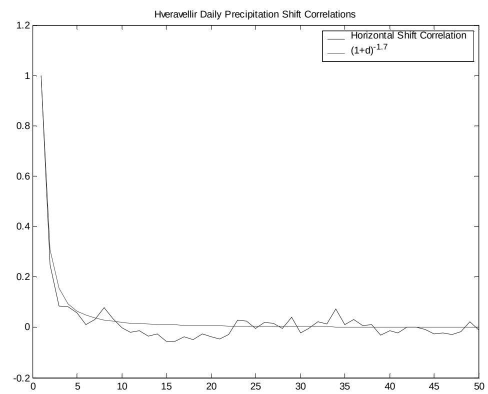


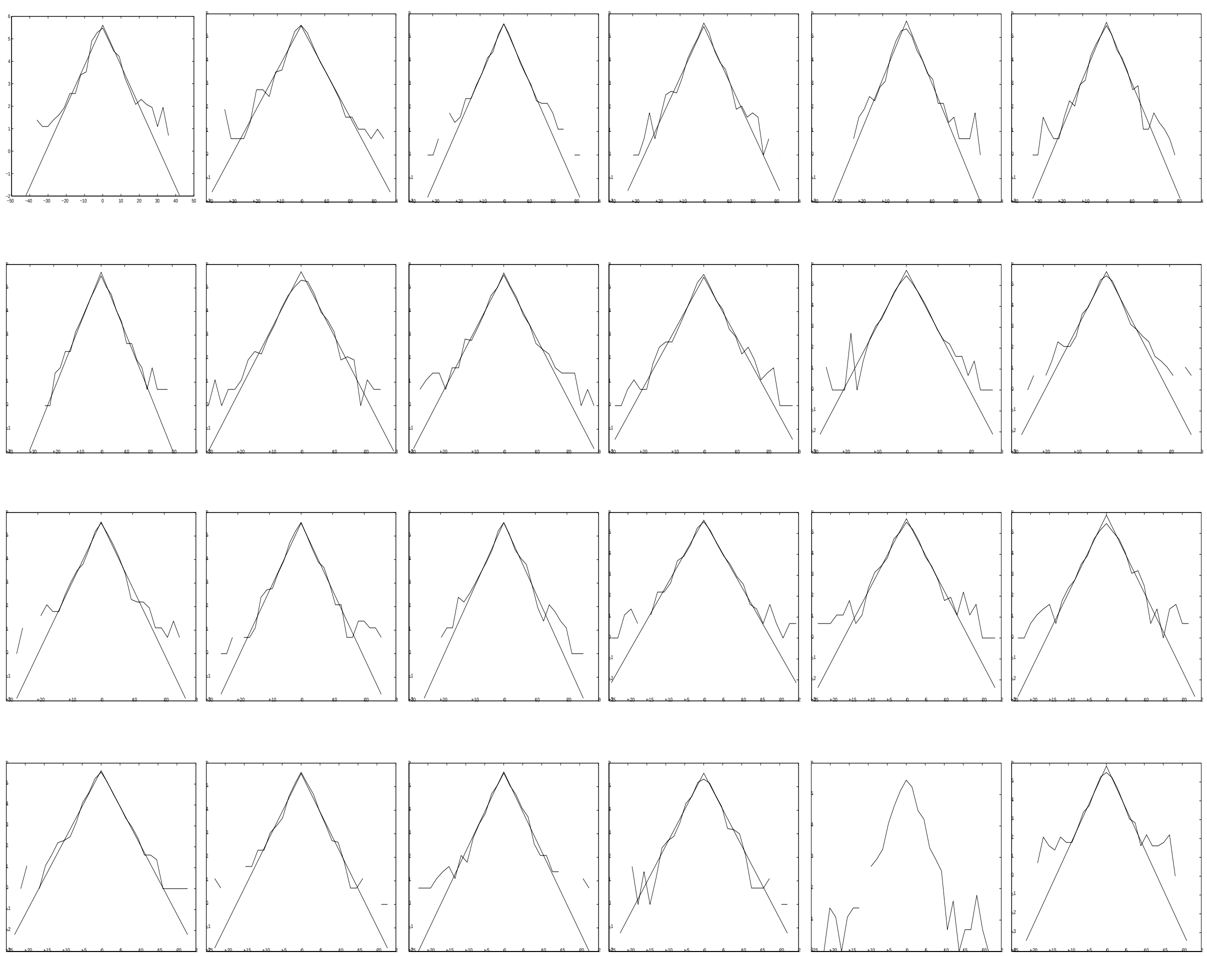
Daily Precipitation, Hveravellir, Jan. 01, 1972 - Dec. 31, 1974

Source: Hipel and Mcleod (1994).

Log Histograms for 24 Eigenfunctions of 25 Hour Sub-blocks

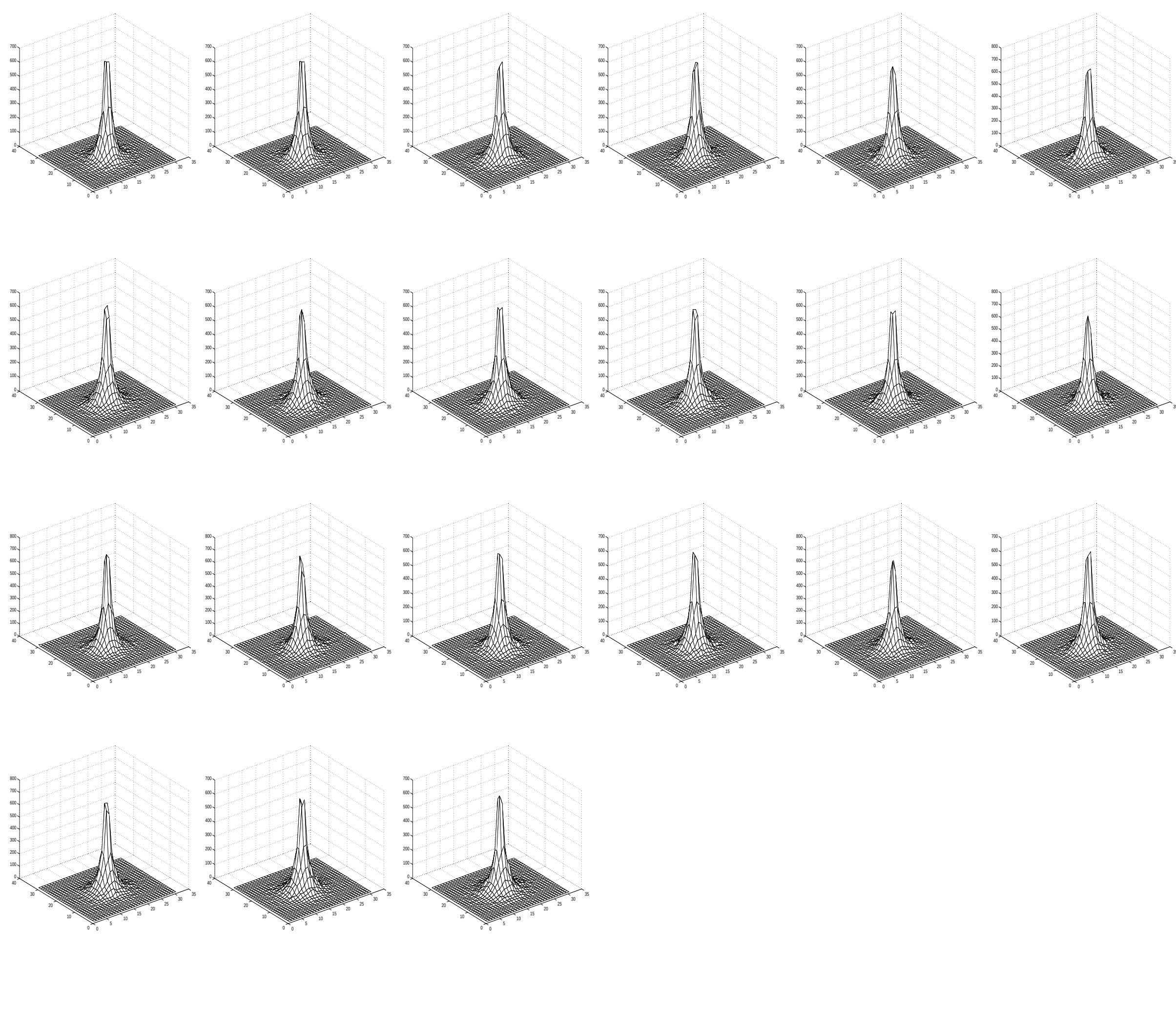






Salvina Bivariate Plots for all pairs of Eigenfunctions 1 to 7

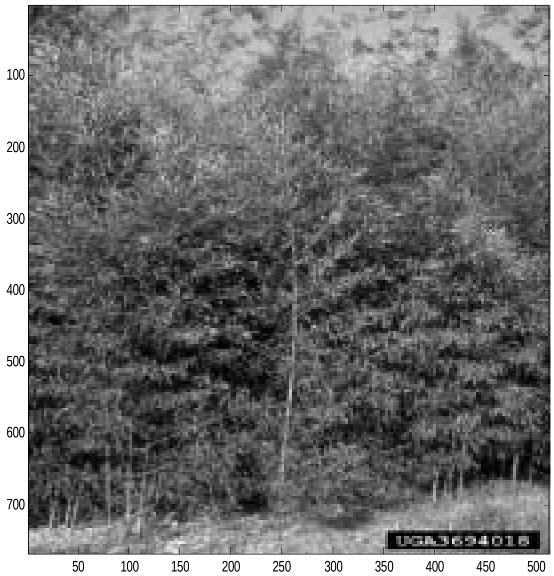
5×5 sub-blocks



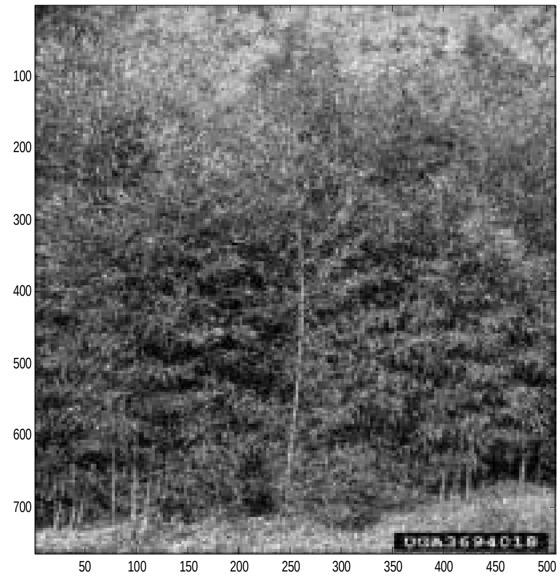
Salvina Bivariate Contour Plots for all pairs of Eigenfunctions 1 to 7

5×5 sub-blocks

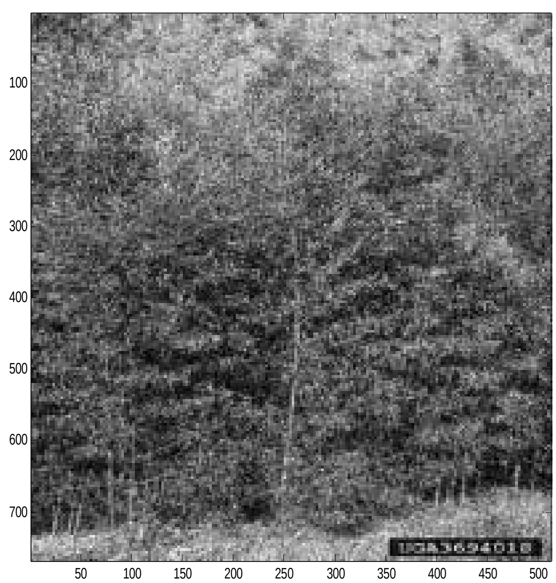
Bamboo Original Image



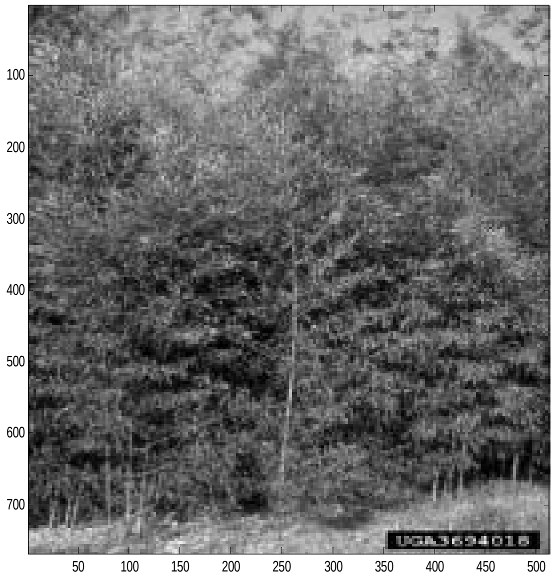
Bamboo Denoised



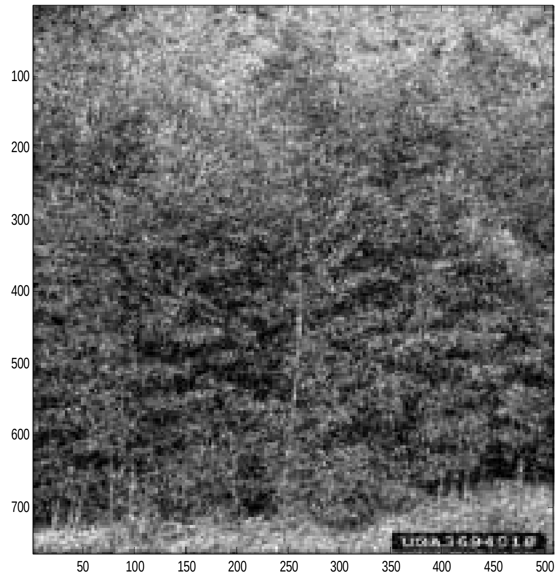
Bamboo with 50 Added Gaussian Noise



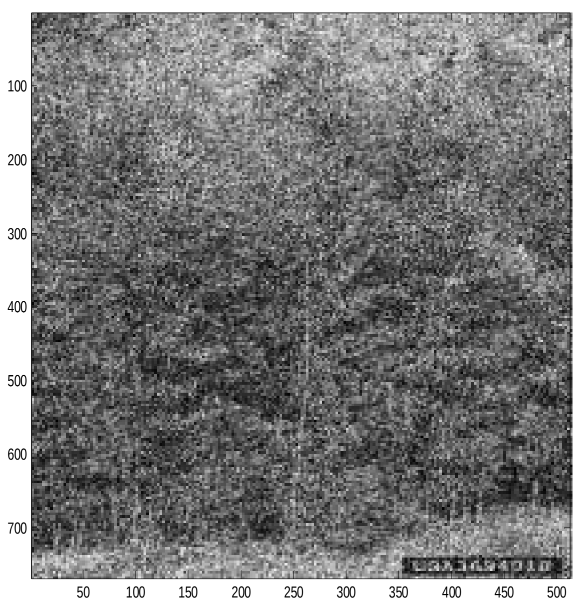
Bamboo Original Image



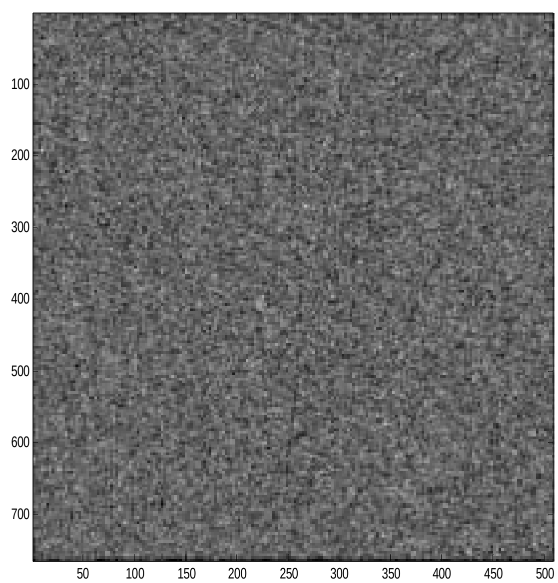
Bamboo Denoised



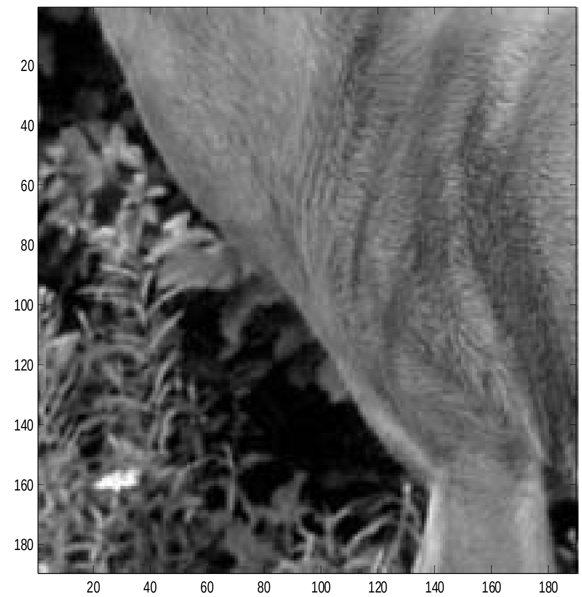
Bamboo + 100 Gaussian Noise



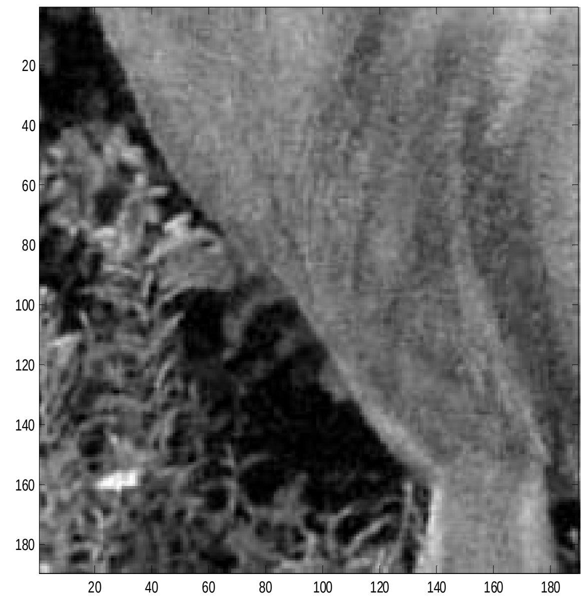
Bamboo Restored Minus Original



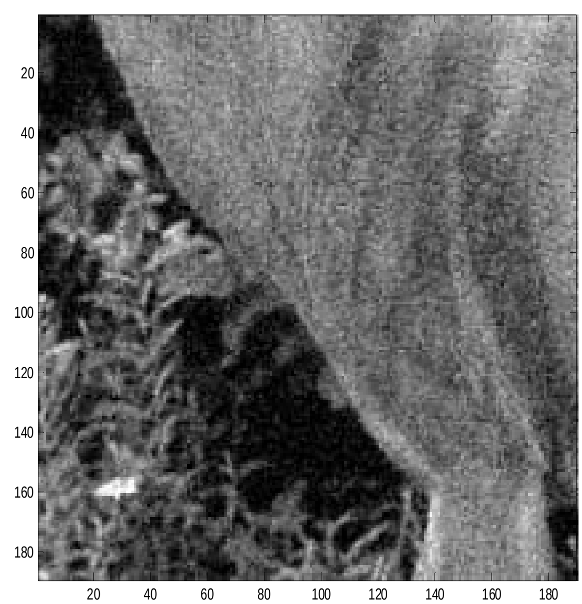
White-Tailed Deer Original Image



White-Tailed Deer Denoised



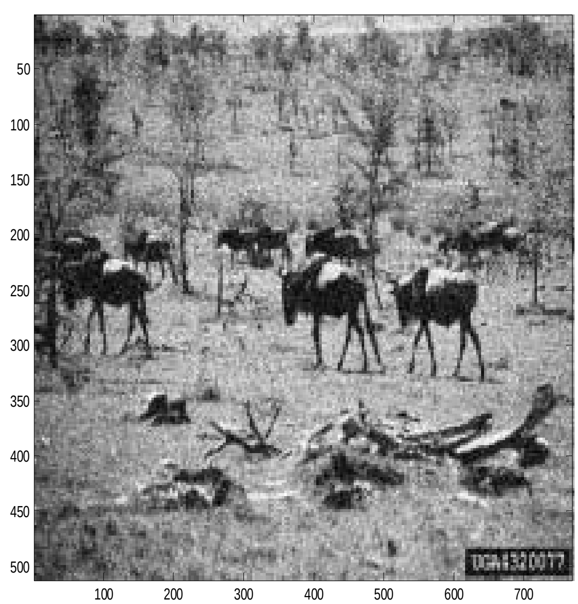
White-Tailed Deer with 20 Added Gaussian Noise



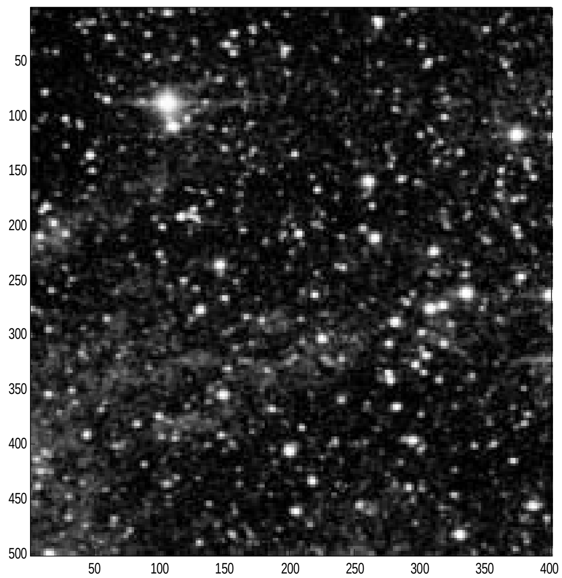
Wildebeest Original Image

Wildebeest Denoised

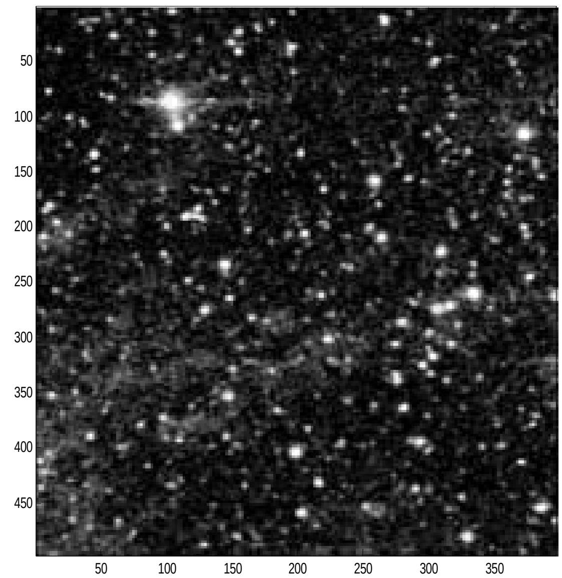
Wildebeest with 50 Added Gaussian Noise



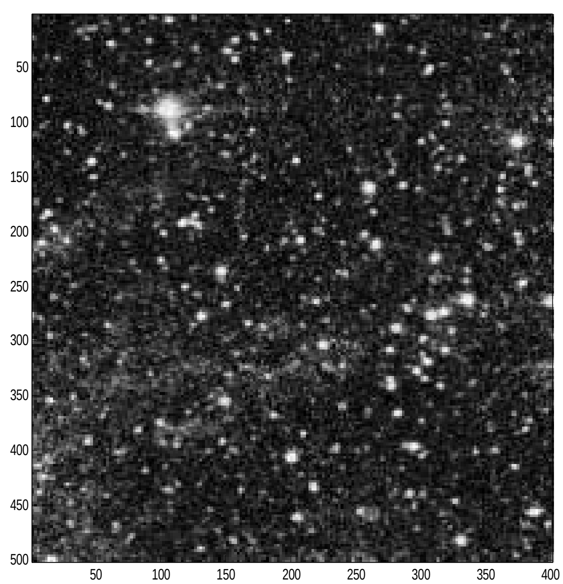
NGC6992 Original Image



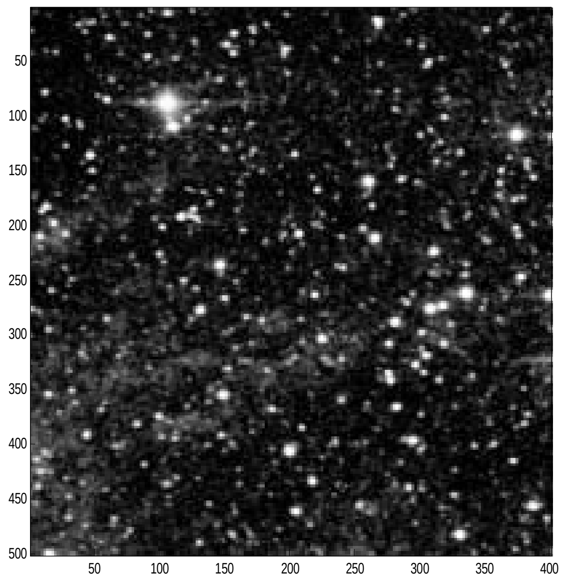
NGC6992 Denoised



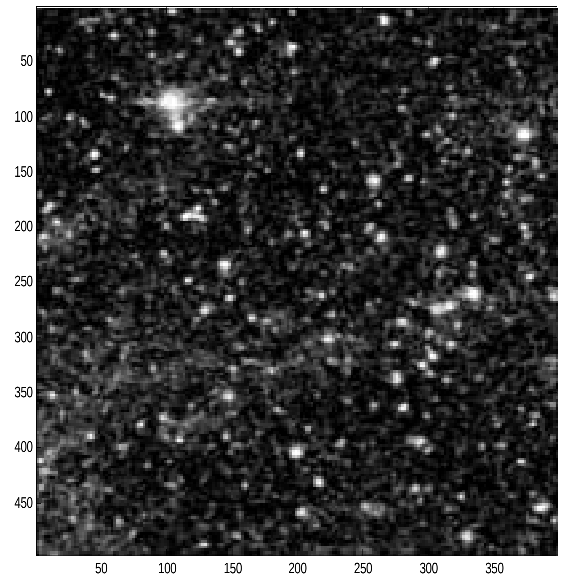
NGC6992 with 50 Added Gaussian Noise



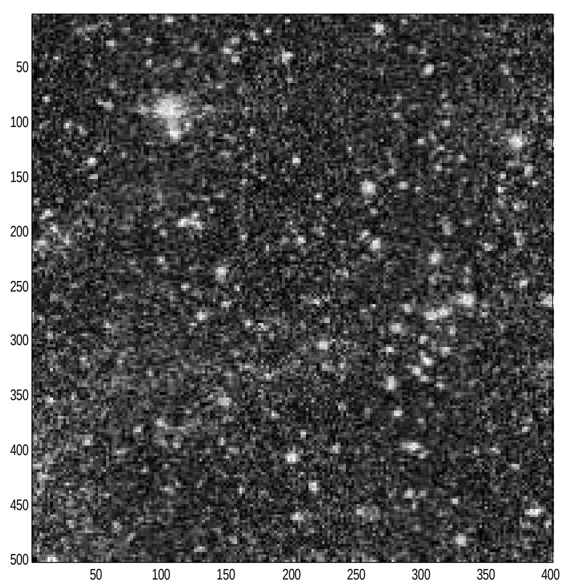
NGC6992 Original Image



NGC6992 Denoised

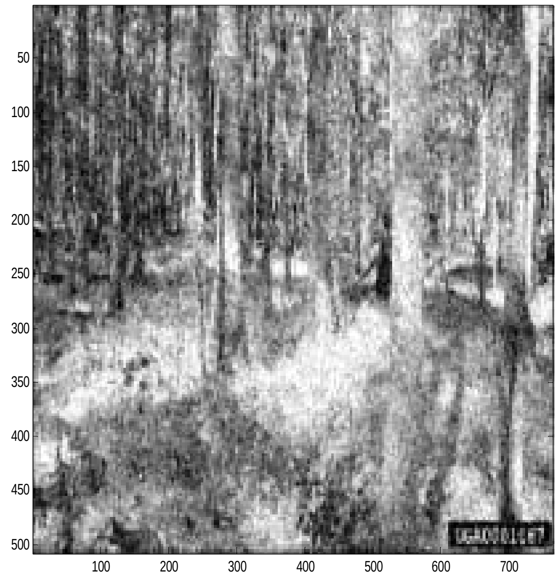


NGC6992 with 100 Added Gaussian Noise



Forest Scene Original Image

Forest Scene Denoised



Forest Scene with 100 Added Gaussian Noise