CAR-TR-742
CS-TR-3366

N00014-93-1-0257
October 1994

Nonlinear Scalespace via Hierarchical Statistical Modeling

David Shulman
Tomas Brodsky

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742-3275

CAR-TR-742 N00014-93-1-0257
CS-TR-3366 October 1994

Nonlinear Scalespace via Hierarchical Statistical Modeling

David Shulman
Tomas Brodsky

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

Nonlinear scalespace should be based on a hierarchical statistical model of the image intensity function. This model should contain an explicit representation of the multiscale structure of edges and corners. Using this model we can have a non-ad-hoc basis for computing the parameters we need to determine how much smoothing we should do at points that appear to be edge points. We also have a basis for computing the apparent error in our scalespace calculations.

Hierarchical statistical modeling is a technique that can be applied to other problems in low-level vision, but in this introductory paper we just present the application of our scalespace theory to image smoothing.

The support of the Office of Naval Research under Contract N00014-93-1-0257 is gratefully acknowledged, as is the help of Sandy German in preparing this paper.

1 Introduction: The Multiresolution Structure of Images

For a long time vision researchers have been aware of the significance of the multiresolution structure of images [4, 7, 12, 13, 18–20, 22] for describing visual data and for solving vision problems via multigrid techniques. The vertebrate retina has a pyramidal multiscale structure [16]. Multiresolution representations are important because images generally contain many features of many different sizes and a coarse-level description of an image can help us make better sense of a fine-level description and vice versa.

The most popular scalespace is the Gaussian linear scalespace. Let $f : R^n \mapsto R$ be the original signal. Then we define

$$f(\vec{x}, \sigma) = G_\sigma * f(\vec{x}) \tag{1}$$

where * means convolution and thus

$$J * L(\vec{x}) = \int J(\vec{x} - \vec{y}) L(\vec{y}) d\vec{y} \tag{2}$$

and G_σ is the standard Gaussian kernel of width σ: $G_\sigma(\vec{x}) = (2\pi\sigma^2)^{-n/2} e^{\frac{-\|\vec{x}\|^2}{2\sigma^2}}$ where $\|\vec{x}\|$ is the length of the vector $\vec{x}$.

This scalespace has many nice mathematical properties [1, 6, 9, 11, 17, 25]. There are theorems characterizing the Gaussian linear scalespace by the fact that it is the unique scalespace such that:

1. It is defined by convolution with a smoothing kernel and thus there is no preferred position as there would be if we did different amounts of smoothing at different points. We also assume isotropy so that the same amount of smoothing is done in all directions. We assume the smoothing operation is linear to simplify calculation. (In the Gaussian case, G is the smoothing kernel.)

2. The smoothing kernel varies as a function of the two variables σ, which we think of as the scale variable, and $\vec{x}$, which we think of as a position variable. This kernel must be continuous with respect to both scale and position.

3. Our scalespace also should not create new structure at coarse scale that cannot be traced back to its origin at fine scale. Structure can mean different things, but it generally means edges and thus every coarse scale edge should be capable of being traced back to a fine scale edge.

(Actually the uniqueness here of the Gaussian kernel is modulo reparameterization of the scale and position variables of the kernel. If we use $\mathcal{G}_\sigma(\vec{x}) = G_{\sigma^4}(2\vec{x})$ in place of G, nothing essential changes.) However, this Gaussian scalespace leads to excessive blurring of discontinuities at coarse scales and consequently to the mislocalization of edges at coarse scales [15]. The basic problem is that nearby edges can interfere with each other and the solution is to blur less at points that appear to be edge points. [3] suggests that if we use a nonlinear multiscale representation rather than a linear one, we need to know the image at fewer scales in order to know the essential structure of the image.

Many nonlinear scalespaces have been defined [5, 14, 15, 23]. They are best described using a diffusion equation. By differentiation one can see that the usual scale space satisfies the diffusion equation

$$\frac{\partial f}{\partial \sigma} = \sigma \ \nabla \cdot \nabla f. \tag{3}$$

Here ∇ represents the gradient and thus $\nabla \cdot \nabla$ is the Laplacian. Usually the Gaussian scalespace is presented with a slightly different parameterization of the scale variable but we choose the parameterization that will make sense of what we will describe in subsequent sections. Lindeberg [8] defines a discrete scalespace for signals defined on a grid by replacing the Laplacian on the right-hand side of (3) with a discrete approximation.

A natural way to obtain a nonlinear scalespace is to change the right-hand side of the diffusion equation to $\nabla \cdot c(\vec{x}, \sigma) \nabla f$. In the linear case, $c(\vec{x}, \sigma) = \sigma$ but in general c can be a function of ∇f, the local measure of how much of a barrier there should be to blurring and thus the amount of blurring varies from position to position, or we might make c a function of a more general measure $g(H * \nabla f)$ of the barrier to blurring. Here H is a blurring kernel. Fairly good results seem to be obtained using these nonlinear scalespaces: edges are much better localized [15, 23]. A problem remains of how best to choose g or H. Usually g will depend on a few parameters and we will need to use some global image statistics to find out a likely good set of values for the parameters. Indeed different g's seem good for different images [15] and the problem remains how to get a good g. If we could derive the g in a non-ad-hoc way from an explicit statistical model of f as a function of scale and position, then maybe we could use that statistical model for other purposes such as obtaining estimates of the error in our scalespace function and the error measure would tell us which zero crossings of the Laplacian or other operator of interest are worth paying attention to.

Before we discuss our nonlinear scalespace and the statistical model underlying it, we will need to present yet another justification of the usual Gaussian scalespace. We use a hierarchical statistical model to justify the preference for the Gaussian kernel. Our scalespace is obtained via a nonlinear generalization of the linear statistical model.

2 Linear Scalespace

We will explain the basic idea of hierarchical statistical modeling and then apply it to linear scalespace. The observed signal can contain features of all sizes and thus we should think of the observed signal as being generated by somehow combining the features generated at each scale. Thus at each scale there is an image that contains only the features of that particular scale. But what we usually mean when we refer to the image at a particular scale, σ, is not just the image of features generated at that particular scale but rather an image from which all features of scale smaller than σ have been removed.

We have to say how the image of features generated at a particular scale is in fact constructed. To say that a feature is of a particular size, σ or particular scale is to say that to understand the feature we need only look in a limited region of size σ; in the multidimensional case if the region of interest is spherical, the size of the region is measured by its radius, but in any case there is some connection between the size of the region and σ.

The image of features of a particular scale is not generated globally in one fell swoop. Each feature is generated locally. This suggests that we should think of this global image of features of a particular scale being generated by combining various locally generated images, one image for each feature. If we are lucky, when we can localize each feature in the global image and say of any such feature that it is of size σ and its generation began at some point $\vec{x}$, we associate the feature with the point at which its generation began. In our situation, the points in question will be centers of the spherical region of interest associated with each feature. Thus associated with each point and each scale is a local image containing the features of the given scale whose generation began at the point in question. If there are no such features, we have a trivial local image at that point and scale.

The local features are the ingredients out of which the global and more complex image is constructed. We want the local features to be simple to ease our computation. In the simple linear case, the various local features have to be combined to form the global image by weighted addition. Later we will have to employ nonlinear combination of features. In the simplest case, the local features should be very simple, simple step functions with at most one nonzero step and that step should have a simple region of support such as a sphere.

Assume we have a model whereby a global image can be constructed from local features. Since many different possible local features can be generated at each point and scale, we can construct many different global images with this model and in general we will not have enough information to recover all the local features given the global image at scale 0. We can say certain global images are such that we are more likely to see them than other global images. This is because certain features are more likely to arise at certain scales and positions than are other features. Thus even if given the global observed image we cannot recover all the local features and moreover cannot recover the global image at a given scale σ from the global image at some scale $\tau < \sigma$, we can at least recover some statistical information about the global image at coarse scale that we wish to recover. We might recover the expected value of the coarse global image. Since there is not enough information in general to recover the actual coarse scale, we know our expected value estimate will usually be in error. We can compute the mean square error of our estimate.

Having presented that intuition that has been guiding us, we are now ready to present formally our statistical model. We will reinterpret the usual Gaussian scalespace using our linear hierarchical model; actually we will derive the diffusion equation.

Let f be the observed signal. We assume f is generated via a linear combination of the signals h_σ generated at each scale. For simplicity, we would like the linear combination to be a sum,

$$f = \int h_\sigma d\sigma. \tag{4}$$

In principle the integration should run from zero to infinity; in reality information about very small and very large scale signals is not obtainable and thus the limits of integration will run from ϵ, which is about pixel size, to some upper limit which is of the order of magnitude of the size of the image. Thus we will write

$$f = \int \frac{1}{k(\sigma)} h_\sigma d\sigma \tag{5}$$

and assume k approximately equal to 1 for the scales of interest. We use the inverse of k in the equation in order to make it easier for us later on to interpret k as a measure of the obstacle to smoothing, the bigger the k the bigger the obstacle.

The signal at scale σ is obtained by summing the components generated at any scale not less than σ:

$$f(\vec{x}, \sigma) = \int_{\sigma}^{\infty} \frac{1}{k(\sigma)} h_{\sigma}(\vec{x}) d\sigma. \tag{6}$$

In reality the observed signal f is not $f(\vec{x}, 0)$ but rather $f(\vec{x}, 1)$ if the pixel size is 1. To say more we have to describe where h comes from.

Since there is no preferred position, at each scale the h must be generated as sums of signals generated at each point:

$$h_{\sigma} = \int h_{\sigma, \vec{x}} d\vec{x}. \tag{7}$$

Each local signal $h_{\sigma,\vec{x}}$ is a step function—a piecewise constant function with at most one nonzero piece and that piece because of isotropy should be a sphere of radius σ centered around $\vec{x}$.

We need only specify how big the signal is on this piece. The size w of the signal on this piece should be a stochastic random variable. Since we will assume that the various local features and hence the w's of the different local features are generated independently, it does not matter very much what the statistics of the w's are; we will be integrating a number of independent random variables and by the central limit theorem if we sum up enough independent random variables, the sum will have a nearly normal distribution. Thus we might as well assume the various w's to have normal distributions to begin with. To simplify computation and analysis, we assume all these normal distributions of the w's have zero mean. All we need to know is the variances of the w's.

Write $v(\sigma)$ for the function that describes how the variance of the w's changes with the scale. There is no position dependence because there is no preferred position. The dependence of variance on scale is also easy to compute. The amount of variance generated between two scales should depend only on the ratio of the scales. The reason for this is that it is the ratio that remains invariant if we change the measure of length. It should not matter whether length is measured in inches, centimeters, or versts. The variance of a sum of independent signals is the sum of the variances. Hence the amount, Q, of variance in f generated between scales σ_1 and σ_2 at point $\vec{x}$ is given by

$$Q = \int_{\sigma_1}^{\sigma_2} \int v(\sigma) d\vec{y} d\sigma. \tag{8}$$

Here the second integral is taken over the sphere of radius σ centered on $\vec{x}$. Q will only depend on the ratio of the scales σ_1, σ_2. To insure that requirement, we will need that v be inversely proportional to σ^{n+1}. In that case Q is really some constant times the integral $\int \frac{\sigma^n}{\sigma^{n+1}} d\sigma$ and depends only on the difference in the logarithms of the two scales and thus depends only on the ratio of scales.

In summary our underlying assumptions are:

1. The global signal OBSERVED at any given scale σ is a combination of the features generated at any scale $\tau \geq \sigma$ and thus is a combination of the global signals GENERATED at scales $\tau \geq \sigma$.

2. The global signal generated at any given scale is a combination of elementary signals generated at that scale and originating at any point.

3. The different elementary signals are generated by independent stochastic processes.

4. Because there is no preferred position, at any given scale, the different stochastic processes, although they may be independent, have identical distributions. (They are generated in exactly the same way.)

5. All the operations of combining are operations of weighted addition.

6. For the scales of primary interest to us, the weights are all equal to 1.

7. The fraction of the variance of the observed signal that can be expected to be generated between any two scales σ_1, σ_2 depends only on the ratio of the two scales assuming we can apply assumption 6.

8. The elementary signals are step functions with at most one nonzero piece. Since there is no preferred orientation, the region of support of the signal is a sphere whose radius is the scale of the elementary signal.

In the nonlinear case we will have to give up one or all of 5, 6, 7, 8.

Our task now is: given the observed signal at scale 1 which satisfies the above assumptions, recover some statistical information about the signal at other scales. This task is made easier by the fact that we have a kind of Markovian property: if we want to estimate the signal at scale σ and we only have complete information about the signal at all the scales less than or equal to $\tau < \sigma$ and we are given no information about other scales, then actually if all we care about is f at scale σ, it suffices to throw away all the data except f at scale τ.

Indeed by linearity, to know the mean value of the signal at scale σ it is enough to know the mean value at scale τ. In the case where statistics are Gaussian, the mean is especially significant because it is also the most likely value as well as the median value. So let us try to estimate the mean value of f at all scales given only the signal at scale 1. By the Markovian condition, to estimate the signal at scale σ, we need only know the signal at scale $\sigma - d\sigma$. Thus we have a diffusion equation

$$\frac{\partial f}{\partial \sigma} = ?. \tag{9}$$

A simple calculation will show that the right-hand side is $\sigma \ \nabla^2 f$. We treat in detail the one dimensional case.

Let us discretize space with grid size dx where dx is small. We also assume $d\sigma$ small and finite. In fact, to make things easy for ourselves let us choose the unit of length and the discretization so that $dx = d\sigma = 1$. Thus 1 is no longer necessarily the size of a pixel. To estimate the mean of $\frac{\partial f}{\partial \sigma}$, we will estimate the mean of the $h_{\sigma,y}$ that are nonzero at x and then add. This works because df is formed by addition of those local features.

The only reasonable way to estimate the means of the h's using the information at scale approximately σ is to use the information in the partial derivatives of f with respect to space. Thus $\frac{\partial f}{\partial x}$ evaluated at scale σ and point x can be written as an integral (which is really a sum because we are discretizing).

$$\frac{\partial f}{\partial x} = \int_{\sigma}^{\infty} h_{\sigma,x+dx+\sigma}(x) - h_{\sigma,x-\sigma}(x) d\sigma. \tag{10}$$

This can be verified by seeing which h's contribute to $f(\sigma, x+dx)$ and not to $f(\sigma, x)$ and vice versa.

When dx is small, we can use the approximation

$$h_{\sigma,x+dx+\sigma} = h_{\sigma,x+\sigma}. \tag{11}$$

If we recall that the h's are independent and that we know their variances, we get an estimate of the size of $h_{\sigma,x+\sigma}$ from knowledge of the size of $\frac{\partial f}{\partial x}$.

From our discussion of how the variance v varies with the scale, we see that the contribution of scale τ to the variance of $\int h_{\sigma,x+\sigma} d\sigma$ is proportional to τ^{-2} and

$$\int_{\sigma}^{\infty} \tau^{-2} d\tau = 1/\sigma \tag{12}$$

while the variance at scale σ contributes an amount proportional to $1/\sigma^2$. Thus we can see that $1/\sigma$ of the variance is contributed at the scale of interest (namely σ) if $d\sigma = 1$. (Otherwise the fraction of the variance contributed at that scale is $\frac{d\sigma}{\sigma}$.) We have the estimate

$$h_{\sigma,x+\sigma} - h_{\sigma,x-\sigma} = 1/\sigma * \frac{\partial f}{\partial x} \tag{13}$$

at point x, or to estimate only one of the h's:

$$h_{\sigma,x+\sigma} = 1/2 * 1/\sigma * \frac{\partial f}{\partial x}. \tag{14}$$

This last equation follows because half of the variance in the difference of the independent h's is contributed by each h. If we also use the information at $x + 2 * \sigma$ we would get the estimate

$$h_{\sigma,x+\sigma} = 1/2 * 1/\sigma * (\frac{\partial f}{\partial x}(x) - \frac{\partial f}{\partial x}(x + 2 * \sigma)). \tag{15}$$

The reason we add the estimate from the data at $x + 2 * \sigma$ to the one from the data at x is that if we invert our linear model, which gives us the f values at all different scales as a linear function of the h's and thus gives the spatial derivatives at all different scales and positions as a linear function of the h's, we obtain a formula of the form $h = z_1 z(x) + z_2 z(x+2*\sigma) + \cdots$ where z stands for the spatial derivative at a point and the z_1, z_2 are coefficients.

Our formula (15) is translation-invariant so it can be used to estimate any h_σ and in particular to estimate any $h_{\sigma,x+\tau}$ with τ less than or equal to σ. What we want is

$$-\frac{\partial f}{\partial \sigma} = \int_{-\sigma}^{\sigma} h_{\sigma,x+\tau}(x) d\tau = 1/2 * 1/\sigma * \int_{x-2*\sigma}^{x} Df(x) - Df(x + 2 * \sigma) dx \tag{16}$$

where D represent the spatial differentiation operation. But

$$1/2 * 1/\sigma * \int_{x-2*\sigma}^{x} Df(x) - Df(x+2*\sigma)dx = -1/2 * 1/\sigma * \frac{\Delta^2 f}{\Delta x^2} \tag{17}$$

where the second difference quotient is taken with grid size $2 * \sigma$.

Next we will use a trick to enable us to compute the signal at scale $\sigma + \delta\sigma$ knowing the signal at σ. By the Markov property, we can divide the step from σ to $\sigma + \delta\sigma$ into small pieces of equal size. Thus let us try to compute $f(x, \sigma + md\sigma)$ by applying an operator $J_{m,\sigma}$ to f at scale σ with m large and $d\sigma$ very small. The J operator applied to a signal at fine scale produces the signal at coarse scale. Of course the value of the signal at point x and coarse scale depends not only on the value of the signal at point x and fine scale but also on the value of the signal at other points.

We just saw in (17) that J is linear and in fact a convolution. If we are dealing with small enough increments, $J_{1,\sigma+rd\sigma}$ with $r < m$ is almost the same function for all the different r and we can write $J_m = {J_1}^m$ because J_m is obtained by iterative application of the different $J_{1,\sigma+rd\sigma}$. Here we have written J_m, J_1 when the scale subscript is understood to approximately equal σ.

Let the operator J_m be convolution by the kernel j_m. Just as there is a central limit theorem that says that the sum of a large number of independent observations has asymptotically Gaussian distribution, there is also a related theorem saying that if we have a kernel j that is normalized to integrate to 1 and convolve it with itself a large number of times we approach Gaussianness. The theorem is easy to prove if we contemplate the fact that if the probability distribution of the results of one independent experiment is given by j, then the sum of the results of m independent experiments is given by j convolved with itself m times [21]. We note that $j_m = j_1 * j_1 * j_1 * \cdots (m\ j_1\text{'s})$ approximately. The normalization to make j a probability distribution is a technical trick; we can always apply the theorem to $\frac{j}{\int j(z)dz}$. What is important that the exact shape of j does not matter and does not greatly affect $j * j * j * \cdots$ if the number of convolutions is large. Letting I represent the identity operator, we can thus replace $J_1 = I + d\sigma * 1/2 * 1/\sigma * \Delta^2$ where the difference quotient has grid size $2 * \sigma$ by the expression $I + d\sigma * \sigma * \Delta^2$ with unit grid size. We are free to perform the replacement because the mean and variance of the associated convolving kernel are not changed thereby. By the theorem about repeated convolutions approaching Gaussianness, only the mean and variance matter. A slight manipulation of this result will give us that the change in signal divided by the change in scale is $\sigma * \Delta^2$. This does employ unit scale dimension grid size. But the only significance of unit length up to this point is that the scale dimension is measured in such a way that it is discretized with unit grid. To compute the needed spatial derivatives the size of the spatial grid must be unit or less. But we can always change the unit of length in such a way as to make it increasingly smaller and approach the infinitesimal. In the limit we get the desired diffusion equation.

Notice that if we carefully count the variance components we get not only an estimate of the mean size of $\frac{\partial f}{\partial \sigma}$ but also a variance estimate that will tell us how much our estimate is in error. If $d\sigma = dx = 1$ is small, let μ be our estimate of the mean value of the derivative of f with respect to scale and ν be the fraction of the variance of μ that is explained by error. (Thus the estimate actually should be written $(1 \pm \nu^{1/2}) * \mu$.) Since only $1/\sigma$ of the components of the spatial derivative $\frac{\partial f(x,\sigma)}{\partial x}$ are generated at scale σ we will have $\nu = 1 - 1/\sigma$ and we can actually show that more generally the error in estimating the signal at scale $\sigma + \Delta\sigma$ given f at σ is approximately $(1 - 1/\sigma)^{\Delta\sigma} = ((1 - 1/\sigma)^{\sigma})^{\Delta\sigma/\sigma}$. Here by the error we mean the fraction of the variance caused by error and this estimate is computed using discretized scale and space dimensions with grid size $d\sigma = dx = 1$. As the size of the grid becomes infinitesimal, σ/dx approaches infinity and thus using the result that $e^x = \lim_{n\to\infty}(1 + 1/n)^{nx}$ an error ratio is obtained of $e^{-\frac{\Delta\sigma}{\sigma}}$ for the case where $dx = d\sigma$ is truly infinitesimal. Thus the real Gaussian linear scalespace should have the equation

$$f(x, \sigma + \Delta\sigma) - f(x, \sigma) = (G_{\Delta\sigma} * f(x, \sigma) - f(x, \sigma))(1 \pm e^{-\frac{\Delta\sigma}{\sigma}}). \qquad (18)$$

If $\Delta\sigma$ is small, we are essentially computing a derivative. We expect derivatives to be hard to compute. Therefore, we are not surprised by the large relative error in this case. If $\Delta\sigma$ is large, we are estimating a difference between signals at very different scales and we have a much smaller relative error. It is standard that if we average a large number of noisy derivatives, we get a difference that is much easier to estimate because the errors in estimation of the derivatives cancel.

We could also compute the effects of quantization and gridding and the finite size of images. All these give us clues as to what thresholds to use when we are looking for important information. We do not actually implement all these error measures in this paper because we are really interested in the more complicated nonlinear, nonhomogeneous, nonisotropic case, but we do want to emphasize they might be of some use in a rigorous theory of scalespace error that does not merely study what happens with two or three ideal edges corrupted by independent Gaussian noise.

The hierarchical model described in this section can be considered an extension of the multiscale autoregressive model discussed in [2, 10]. But we allow the set of scales to be continuous rather than restricted to powers of 2 and allow for a much finer sampling at coarse scales of the function being generated. Our model is also much more naturally extendible to the nonlinear case.

3 Nonlinear Scalespace

In all of the previous section, we were assuming that $k = 1$ in the region of interest. If k were not 1, the diffusion equation would have to be slightly modified. We would have the equation

$$\frac{\partial f}{\partial \sigma} = \frac{\sigma}{k}\frac{\partial^2 f}{\partial x^2} \qquad (19)$$

in the one-dimensional case. More generally k could vary with position thus reflecting that certain parts of the image have more information at a certain scale then other parts of the image. We might also want k to vary with orientation because at any given position and scale the image is smoother in some directions than in others. Thus we would have the diffusion equation

$$\frac{\partial f}{\partial \sigma} = \sum_i \frac{\sigma}{k_i}\frac{\partial^2 f}{\partial {x_i}^2}. \qquad (20)$$

Here the subscripts represent the different possible directions. The demonstration of this is routine given the previous results: we just have replaced certain sums by weighted sums where the weights are $1/k$.

We can give a natural interpretation of the case where k is greater than 1. That represents the situation where there is an obstacle to smoothing. A perfect obstacle would occur when k was infinity while k between 1 and infinity would represent a partial obstacle or a partial edge. Values of k less than one are harder to interpret, but we wish to allow k to be very small so that in very smooth regions we smooth a lot more than average. Thus k represents the interaction of two effects: one effect is that sharp edges will results in big k's. The second is that k can vary because certain regions of an image or certain scales of an image are more prominent than others. If there is a large amount of signal at some scale τ and very little signal at the scales between τ and $\tau + \delta\tau$, then we need to smooth a lot at scale τ. We can reflect this in our model by appropriate choice of k.

We will modify our hierarchical model to allow for edges and k being bigger than 1. We note that in our model edges (and later corners) are part of the model, part of the representation and not just features that can be computed given the representation. It is not a real-world fact whether or not edges are part of the representation or are simply features. We do things in the way we find most convenient. To model the statistics of the f-field (the field of image intensity values), we need to introduce an edge-field as an intermediate variable. To model the edges better we need to introduce corners.

We need to modify our model of the f-field because the linearity assumptions are unrealistic and so are the Gaussianness assumptions. We, to be sure, could still use the linear model if we had some easy way to know what k should be at each scale and position and orientation. But we do not and thus will have to resort to nonlinearity to construct a diffusion equation with the appropriate k values.

We modify the model of the f-field by changing the definition of the basic functions h. They can no longer simply be step functions. In the linear case we can think of the elementary function $h_{\sigma,\vec{x}}$ starting out by giving the point $\vec{x}$ some value and then that value diffuses out from that central point until it hits the boundary of the sphere of radius σ centered on $\vec{x}$. In the nonlinear case there are obstacles or boundaries or edges that inhibit the diffusion and thus as we move out from the center the signal gets attenuated. The fundamental idea here is that an elementary signal can not diffuse across a true edge and can only partially diffuse across a partial edge or partial boundary. Thus if c represents the central value, we have

$$h_{\sigma,\vec{x}} = a_{\sigma,\vec{x}}\ c \tag{21}$$

where $a_{\sigma,\vec{x}}$ is an attenuation function we need to define.

We know this function equals 1 at the center of the sphere of support. We will define the function $a_{\sigma,\vec{x}}$ by showing how it changes with position. Thus we are interested in the derivatives of $a_{\sigma,\vec{x}}$ with respect to space. Change in the quantity a should be due to some objective feature of a point, namely, the degree to which the point is an edge pixel and thus all attenuation functions will be affected in the same way when they hit an edge in their attempt to diffuse outward from the center of their zone of support. Now if the effect of a true edge is to prevent any smoothing across the edge, the effect of a partial edge is to allow some fraction of a signal to diffuse from one side of the edge to the other. This is a kind of "linearity" assumption that the partial edge attenuates all signals by some constant fraction and this assumption simplifies the mathematics because it allows us to change the units in which f, which could be light intensity, is measured and not change anything significant. Thus we have seen what should be invariant would be the ratio of the values of the function a on the two sides of the edge or the difference between the values of the logarithms on the two sides of the edges. Hence we have the equation

$$\frac{\partial \log a_{\sigma,\vec{x}}(\vec{y})}{\partial y_i} = -\frac{y_i - x_i}{|y_i - x_i|} o_i(\sigma, \vec{y}). \tag{22}$$

The term in $\frac{y_i - x_i}{|y_i - x_i|}$ is necessary because the derivative should be with respect to the distance between y_i and x_i. Here x_i matters because it is the center of the nonzero region of the basic signal $h_{\sigma,\vec{x}}$. It is significant that the obstacle field $\vec{o}$ does not vary with $\vec{x}$ because $\vec{o}$ as we have mentioned earlier is an intrinsic feature that measures how much of an edge a point might be.

The last equation that we wrote is something that ideally we would like to be true of the obstacle field $\vec{o}$, but there are certain consistency conditions we have to worry about. The problem can be seen if there is a partial corner. The elementary signal is trying to diffuse to the Northwest. If it diffuses first North for some distance and then West, it might have to diffuse across a strong North boundary and then a strong West boundary and wind up very much attenuated before it reaches a certain point of interest to the Northwest. But it might be able to travel to the same point by another route whereby it only hits weak boundaries and thus is not that much attenuated. We thus have to modify the equation for the obstacle field or the meaning of the attenuation field. Or we can just assume we do not run into the problem of the path dependence of the attenuation field. This is saying that the obstacle field is not and cannot be arbitrary.

What we actually do is smooth one dimension at a time and compute the obstacle field one dimension at a time so that again path dependence does not matter. But we are implicitly assuming that the obstacle field has a special character. The field would have this special character if it were the gradient of a scalar field. If it did not have this special character, we would have to use somewhat different equations, but we assume that the algorithm we actually use will work anyway.

Knowing the statistics of the obstacle field, we know all we need to know in order to determine the statistics of the signal. Estimation of coarse level signals given fine level signals is now much harder. We can no longer rely on Gaussianness and linearity. The mean value no longer need be the most likely value. It is still the easiest thing to estimate; thus we will try to recover this value anyway. Since we want a local diffusion equation we assume it is legitimate to use a diffusion equation that is the same as the standard equation except for the weightings of the second derivatives by k_i.

Actually, since we are working with discrete data and edges can be rather thin, it is important even in one dimension that there can be a much greater obstacle to smoothing on one side rather than another of some pixel. And if k measures the obstacle to smoothing, we now need two k's to measure the obstacle on each side of the pixel. So instead of working with $\frac{\sigma}{k_i}\Delta_i^2$ to smooth in direction i, we will instead use $\frac{\sigma}{k_i^+}\Delta_i^+ - \frac{\sigma}{k_i^-}\Delta_i^-$ where Δ^+ and Δ^- represent forward and backward difference quotients respectively in the direction i. This expression reduces to what we had before if the two k's are the same. Of course, there should be some redundancy here; the k^+ at pixel 8 should be the same as the k^- at pixel 9.

3.1 Some Details of Implementation or How We Determine k in Practice

The problem is to determine k. One image really does not provide enough information to allow us to come up with a reasonable probability distribution for k. Since we can only get a certain amount of information about k anyway from an image, we wanted to see in this preliminary investigation how far we could go using only a few assumptions. One simple assumption is that the difference quotients of f with respect to scale have a Gaussian a priori distribution. This assumption is true in the linear case (according to our model) and we want to stay as close to that situation as possible.

More generally we would have to transform the f-field before we could apply the Gaussianness assumption and the problem would be finding the right transform. Thus perhaps there is some operator Z that can be applied to the f so that the derivatives of $Z(f)$ with respect to scale are a priori Gaussian. This operator cannot just be a linear operation such as convolution but something more complex.

Another assumption is that k's depend only on difference quotients with respect to space, $\frac{\partial f}{\partial x_i}$.

The problem is we do not know which difference quotients should be used. The central limit theorem trick we used in the linear case will not work here and in fact it is not surprising that if we want to obtain information about phenomena of size σ it is difficult to get accurate answers using strictly local information. Our attempts to find a strictly local computation of k failed. Another thing that we tried and that also failed is

1. at scale 1, estimate k;
2. find some locally computable Gaussian variable t that is a function of k (i.e. somehow we apply some locally computable nonlinear operator to the k-field and obtain a variable with Gaussian a priori statistics);
3. smooth t (which is really t at scale 1) to obtain t at coarse scale;
4. apply some locally computable inverse transform to coarse scale t to get k at coarse scale.

The problem with this approach is that there is some nonGaussianness that is irreducibly nonlocal and our results with this approach were not that much better than linear smoothing. The approach is justified by our theory that everything should reduce to the linear Gaussianness case. This approach says instead of modeling k or rather modeling $\vec{o}$, the obstacle field, which determines k and which is highly nonGaussian, all we need model is a field t that satisfies the linear Gaussian hierarchical statistical model we described in the last section and a transformation function π which lets us obtain $\vec{o}$ given t (not a statistical model of the transformation but one single transformation function which can vary with scale in some systematic way but should not vary from image to image). Unfortunately we do not know the transformation function; the transformation function π induces a transformation function from k to t which is difficult to compute strictly locally.

To return to describing the procedure we actually did follow: An additional assumption we have made to facilitate our calculations is that the k_i's can be determined separately in each direction and to determine the k at scale σ we need a histogram of Δ_i with grid size σ. Big gradients should mean little smoothing and small gradients should mean much smoothing.

We describe our procedure more precisely:
Assume we wish to determine k; we do that indirectly by first determining the desired change $C = \frac{\Delta f}{\Delta \sigma}$ or more exactly the amount of change that should be desired when all we know is that the first spatial difference quotient of f in a particular orientation (x or y) and direction (forward or backward) has a certain value . Thus we need a formula telling us how much smoothing of the f-field we need to do when we see a certain value in a certain spatial derivative.

We write

$$k = \frac{\sigma \frac{\Delta f}{\Delta x}}{C}, \tag{23}$$

which is just another way of writing the diffusion equation. Here we have omitted to write the subscripts and superscripts indicating orientation and whether forward or backward differences are used. We need to determine k and not just C. Although all we ultimately need to estimate is C, in order to implement corners, we will need to obtain and smooth k.

We have indicated that C should be locally computable and in fact computable from the first spatial differences of f. To compute C, we need a transformation function that goes from the histogram of first differences Df to the desired Gaussian histogram of C. We want to actually compute histograms because when we do not know the desired function from Df to C, rank information is relatively easy to use and can allow us to compute with a fairly simple algorithm quite complex functions from Df to C.

We want the histogram of C to have zero mean in order to simplify calculations. The average difference between the coarse and the fine scale signal should be zero. Because of symmetry considerations and problems that can arise with division by zero, we need the histogram of Df to have zero median (this will make it easy for zero values of Df to transform to zero values of C); we, therefore, preprocess the image to insure that this happens. This involves subtracting a linear function from f. (We will have to subtract some constant from all the Df's.) This subtraction has the desirable effect of protecting us from the effect of global lighting conditions. We assume all our models apply to the preprocessed image rather than the original image.

We next require that the transformation from Df to C be symmetric so if d transforms to c then $-d$ goes to $-c$. Thus we really use a histogram of the absolute value of Df.

The most typical value of this histogram is the standard deviation sd and the most typical points have $Df = sd$. Points with bigger difference quotient than sd will tend to be edge points and they should transform to points in the histogram of C with small values. Points with difference quotient smaller than sd should transform to points with small change because there is not a large enough gradient to justify a big change.

The simplest transformation satisfying these two conditions is piecewise monotonic with two pieces (four pieces if we consider both negative and positive difference quotients.).

To say more we need to define the variance of C, which we know should have zero mean. (We need to know exactly which Gaussian histogram we are transforming to when we map from the histogram of first differences to the histogram of desired changes.) The variance of the C's should be the same as that of $\sigma\frac{\Delta f}{\Delta x} * d\sigma$ if we are using the diffusion equation with k everywhere approximately equal to 1. This follows from the fact that in one dimension the difference quotient of f with respect to scale is computed as a difference of two expressions of the form $\frac{\sigma}{k^j}\Delta^j f$ where j represents whether forward or backward differences are taken. We are computing the change due to one of the expressions and assuming (not for the purpose of computing the k or the desired change at each point but just for the purpose of computing the variance of C) that $k = 1$ or is at most points close enough to 1 to allow us to use this approximation when computing the desired variance of C.

Now we can write the simple formula for the desired transformation ω. We really are interested in the transformation from the positive part of the histogram of Df to the positive part of the histogram of C. We work with percentiles of the positive histograms. Thus ω maps from a percentile to a percentile. Let ψ be the percentage of points of the positive histogram of Df that are less than sd. Then the transformation is

$$\omega(\rho) = \begin{cases} \frac{100\rho}{\psi}, & \text{if } \rho < \psi \\ 100 - (\frac{100*(\rho-\psi)}{100-\psi}), & \text{if } \rho \geq \psi \end{cases} \tag{24}$$

This is a complex transformation but is the simplest transformation that is monotonicly increasing for small values and monotonicly decreasing for large values. We tried working with a strictly monotonic transformation, but the result was edges were insufficiently sharp even though some of the k's at edge points were quite large.

To implement the histogram transformation we need to be able to compute the histogram of Df when scale is a fraction and grid size is a fraction. First we need to define f at nonintegral values of $\vec{x}$ which is easily done through linear interpolation. The justification for using the linear interpolation is that the linear scalespace could also be obtained by solving the variational condition of minimizing $\int (f(\sigma + d\sigma) - f(\sigma))^2 + \sigma(\nabla f)^2 d\vec{x}$ given the observed values of f at scale σ and solving this condition requires that we use linear interpolation to obtain f at fractional positions. The argument from the hierarchical model is somewhat more complex.

Next we need to know at which values do we actually compute Df. The answer is we are only interested in first differences of the form $f(x + .5 + .5 * \sigma) - f(x + .5 - .5 * \sigma)$. This is the region of length σ centered at the boundary between two neighboring pixels. One additional problem is points being too close to the border of the image. We cannot compute a Df centered around such points. For the purpose of computing the transform from Df to C we ignore such points (otherwise we would have problems with double counting the same data.). For the purpose of computing k at such a border point x, we first find the nearest nonborder point y where we have no trouble computing $C(y)$ and use $C(y)$ and the first difference at y to compute k at x. Here we are using $C(y)$ to represent the desired change at point y. Thus if the scale is 10, we cannot compute a central difference in the x coordinate with center (3.5, 32). The nearest central difference we can compute is the one centered around (5.5, 32) and that is the one we need to use.

As usual with histogram transformations we use binning (i.e. we only directly transform some of the values from Df to C and the rest of the function is computed through linear interpolation). Thus we only directly compute the transform for values of Df that are at the middle of bins. We bin in order to lessen computation and increase resistance to noise. Provided certain constraints are enforced about allowing a maximum number of pixels in each bin, the results are not very sensitive to details of adaptive histogram computation. There is one problem: What to do with very large Df; there is a largest bin and depending on exactly how we set certain limits the value of k at points where the gradient is large can vary greatly. This does not much affect the final image but it does affect the interpretation of the k-field and interferes with the smoothing of edges by the corner operator.

3.2 On Computational Complexity

We have not done all we could to optimize the speed of our calculations. We have after all described a computationally very intensive algorithm particularly if we solve the diffusion equation by brute force methods. Things are not quite so bad as that. There are approximations that work well under the assumption that k varies slowly with scale. If k is actually constant we obtain Lindeberg's scalespace [8]. If k varies with position and not scale or varies slowly with scale we approximate by first doing linear interpolation at the finest scale to make our signal continuous and then apply a change of parameterization to the spatial variables in order to make $k = 1$ everywhere and then we just have ordinary Gaussian scalespace which is easy to compute. Of course, to get usable results we eventually have to invert the transformation that reparameterized the spatial coordinates. A variant of this which will work in the discretized case just uses the smoothing kernel of Lindeberg and the reparameterization discussed above. In fact if we integrate the diffusion equation by brute force and we take small enough steps, then provided we renormalize the sum of coefficients so that they add up to 1 we do not even have to use the full Lindeberg kernel or the full Gaussian smoothing kernel but instead only consider the three biggest coefficients and renormalize.

We could have developed a more efficient way of solving the diffusion equation at some cost in accuracy, but our goal in this paper is simply to show that the nonlinear smoothing produces useful results and to introduce the hierarchical statistical model underlying it.

4 On Corners and Insuring the Smoothness of Edges

Just as we used scalespace to nonlinearly smooth the observed intensity field because intensity fields tend to be smooth, we might also want to smooth the edge measures k_x, k_y. We might want to smooth the k's because edge pixels tend to cluster and thus information about the proper k value of a given point can be found by looking at nearby points. Thus if nearby points to point P have big k's so might P. However, the effect of this smoothing might be to make edges too fuzzy and corners too rounded and indistinct.

The k variables are emphatically not Gaussian or near Gaussian in the way the f-field might be nearly Gaussian. Thus it is harder to know exactly what smoothing should be applied to the k-vector field. In our theory we really have a hierarchical almost Gaussian model of some other variable t that gets transformed into o, but we do not know the transformation and thus do not know t. And since we do not have to work with t directly, we ignore trying to find the transformation and instead try to find directly how to smooth k. We know the smoothing cannot be linear because we do not want to make edges excessively fuzzy.

In fact, we choose to smooth not k but $\tilde{k}$ which is the same as k except for sign. The sign of $\widetilde{k_x}^+$ is the same as that of $\frac{\Delta^+ f}{\Delta x}$ and similarly for $\widetilde{k_y}^+$. We do not really have to worry about the k^-'s because they are the same as the k^+'s except for being translated one pixel. We use signed k's in order for it to be possible for a big positive jump in k to kill a big negative jump in k. This will increase the tendency for a step function of size σ to in fact disappear at a scale approximately equal to σ.

We would like to smooth the components of the k vector exactly in the same way that we smoothed the intensity field f. But we cannot rightly do so because of the problem with very large k's (especially when the exact size of a large k depends on an arbitrary limit) and because of the related problem of the extreme nonGaussianness of k. One way to reduce the sensitivity to the exact size of large k's is to take logarithms and then smooth. Of course, we cannot really take logarithms because we have signed $\tilde{k}$. Also there is something slightly ad hoc in deciding to suddenly take logarithms here; thus, we would like small k's to be left alone. Thus we define a transformation function $\gamma(v)$ such that $\frac{d\gamma}{dv} = 1$ if $|v| < V_0$ and $|\frac{1}{v}|$ otherwise. Here V_0 is a large constant, large enough so the results are not very sensitive to exactly how large the constant is. The threshold V_0, however, should be large enough that a point having k bigger than the threshold in question is likely to be a point where a sharp edge is present. Thus below threshold we have the identity function and above the threshold the derivative is the same as it would be if we were keeping the sign and taking the logarithm of the unsigned k's. We would like to smooth $\gamma(\widetilde{k_x})$ in the same way we smooth f but in that case the corner field values would be determined by computing a difference of $\gamma(\widetilde{k_x})$ values at different points and comparing to a histogram of similar differences. This is too sensitive to point to point variation in k. This variation can be huge because the k field will have a different structure than the f field. The k-field even after it has been transformed by γ is still too noisy and varies too much from point to point for us to get optimal results for the corner field. As the scale is increased slightly, edge features or more precisely large derivatives of f with respect to position can suddenly appear. This is acceptable. We want features of size 10 to suddenly almost disappear at scale 10. What we do not want is that they suddenly reappear at scale 10.2. This does not quite happen but if we look at the rate of change of f (i.e. $\frac{\partial f}{\partial \log \sigma}$), we can see how there can be a very rapid rate of disappearance of features followed by a very rapid reappearance of features. Visually nothing looks abnormal when we look at the smoothed images but computation shows there is an unphysical artifact.

To prevent the excessive influence of point-to-point variation we need to perform another operation before computing histograms. This operation should be a kind of smoothing. What we actually do is a kind of addition that allows the differences we compute to be influenced by more points than just the two where the difference is being computed. Thus if

k_x somehow measures the barrier between two adjacent pixels, by adding nearby k values we might obtain the barrier between two pixels that are not adjacent. Since k_x only measures the barrier to homogeneity in the x-direction, we should only add in the x-direction. Actually we do not add k values but k values as transformed by γ. The summing transformation is simply then $k_x{}^*[i][j] = \sum_{l=0}^{l=j} \gamma(\widetilde{k_x}[i][l])$ and we then use the resulting $k_x{}^*$ as input to the operation of computing histograms of differences and mapping from differences to desired changes. Thus we are using differences in the $k_x{}^*$'s to determine how much to smooth the k-field.

Here we have described what we do with the k_x; what we do with the k_y is the same except the summation is taken in the y-direction. Just as in the process of computing the f-field we have to compute the edge-fields, k_x, k_y, in the process of smoothing the x-direction edges we calculate the corner fields b_{xx}, b_{xy} and smoothing the y-direction edges gives us b_{yx}, b_{yy}. We have

$$\frac{\Delta k_x}{\Delta \sigma} = \sigma * (1/b^+_{xx}\frac{\Delta^+ k_x}{\Delta x} - 1/b^-_{xx}\frac{\Delta^- k_x}{\Delta x} + 1/b^+_{xy}\frac{\Delta^+ k_x}{\Delta y} - 1/b^-_{xy}\frac{\Delta^- k_x}{\Delta y}). \tag{25}$$

A similar equation holds in the y-direction.

4.1 How Corners and Edges Together Affect the Smoothing of the Intensity Field

In this section we summarize our basic smoothing algorithm and show how it uses edge and corner information.

The first step is to use the given intensity data to compute the k's at scale 1. Then we can use these k's to compute the b's at scale 1.

At subsequent iterations, we histogram difference quotients of f in order to get an initial estimate of k. We then use the histograms of difference quotients of k or more precisely k as transformed by γ in order to get a new b. Next we use this new b to help us smooth our initial estimate of k. Finally the revised k is used to smooth f.

5 Calculating Results

There are numerous ways results might be computed that will enable us to compare various smoothing filters. One possibility would be to look for the most interesting points in scalespace. These should be the points where the change in f as we change scale (or more precisely as we change the logarithm of scale [ratios rather than differences of scales are significant]) is the most significant. We could find these extrema by looking for zero-crossings of an operator that takes the second derivative of f with respect to the logarithm of scale. The results are too noisy and there are too many extrema with small first derivative. This suggests we should at each point P of scalespace take the average of f over a rectangular solid in scalespace and then take the second derivative.

But actually the results are most transparently evident if we simply graph the effect of using different filters. We display the first derivative of f with respect to the logarithm of scale. Or more precisely we chart the amount of change in f caused by a small change in scale (one step in our iterative algorithm, one small increment in scale) where the size of the small changes in scale is an exponential function of scale. There are points that are boundaries of features of known size and where we want the f not to change much until we reach the scale of approximately the size in question. We do see in the examples below that the results for the filter using b and k are better than for the one using just k and the one using k is better than the linear filter. We can also visually compare the smoothed images produced by the linear and nonlinear filters although there are not very large differences between the two nonlinear filters and thus we only display the nonlinearly smoothed images produced using both b and k.

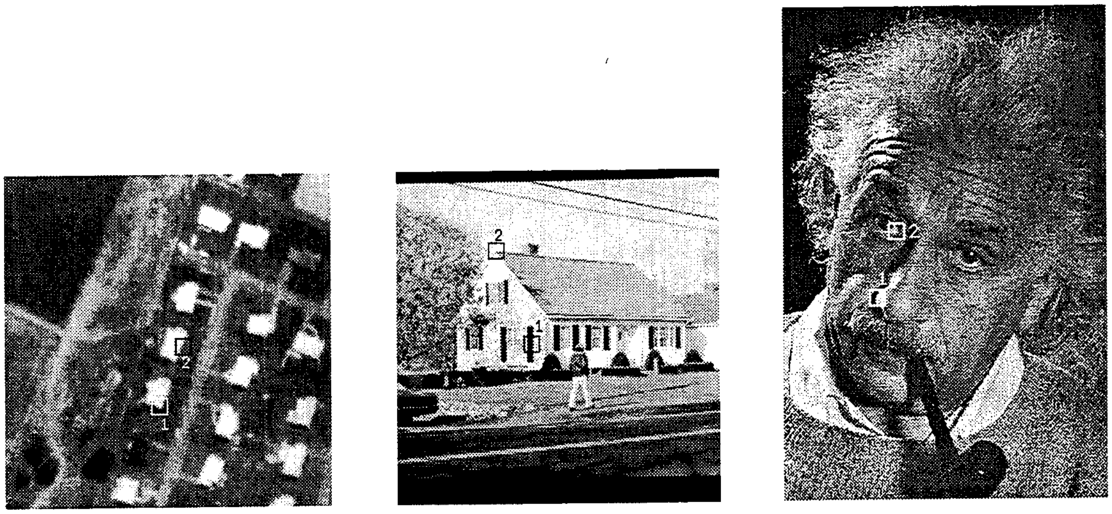
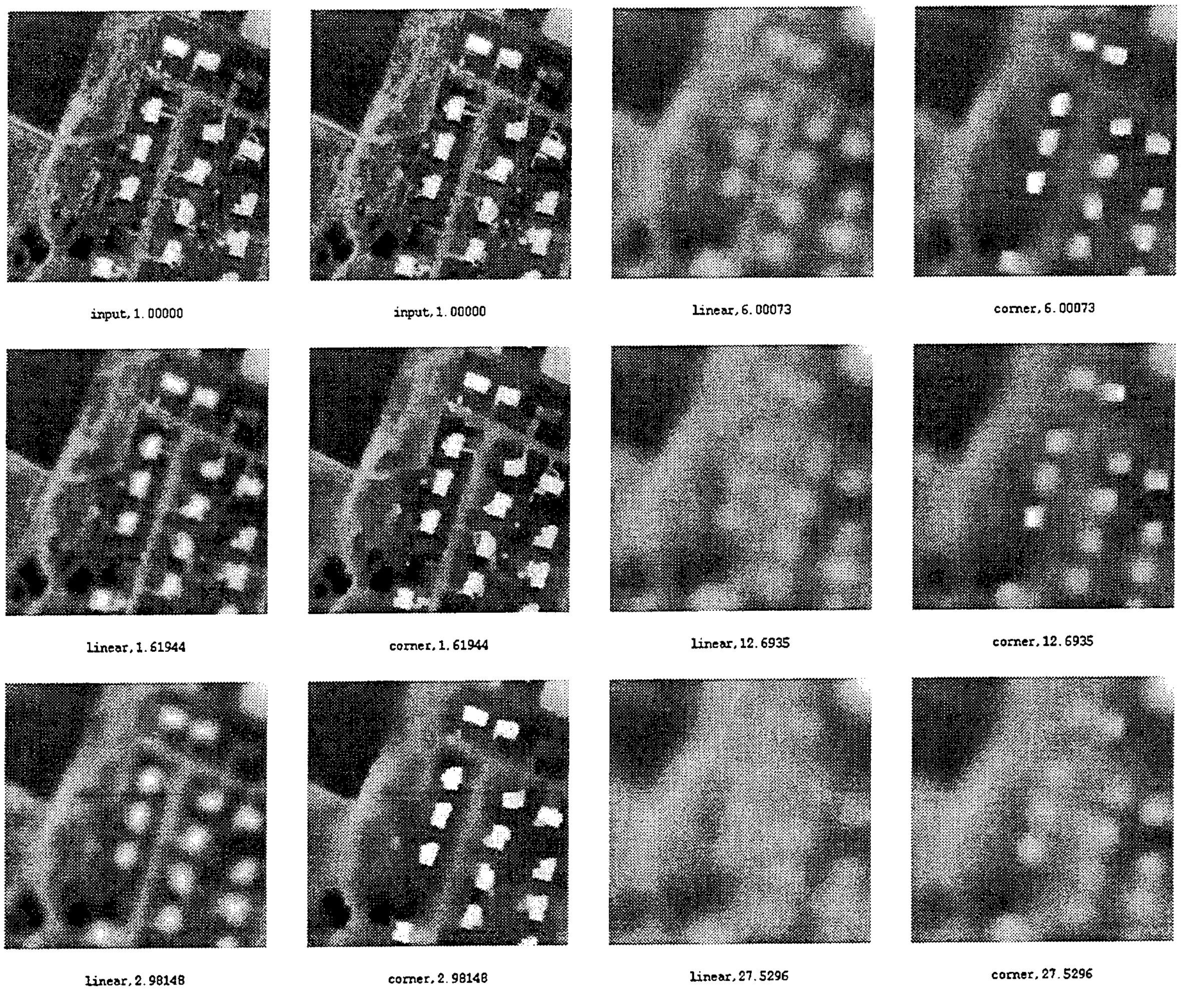


Figure 1: Test images with indicated trace points

6 Images

We show in Figure 1 three images to which we applied both linear and nonlinear smoothing (Figures 2–4). The two kinds of smoothing are labeled "linear" for linear smoothing and "corner" for nonlinear smoothing using both edge and corner features. A number labeling an image represents the scale of resolution shown. We have three images: one is a picture of a road scene with many small houses, the second is a picture of a house and yard and the third is a truncated picture of Albert Einstein.

We see a very distinct difference between the linear and corner smoothers. We notice that the corner smoother does in fact make blurry regions even blurrier than the linear smoother (k is very small) and thus it does recognize that some regions are smooth, very smooth, but distinct features do become more prominent and this is not because they are not smoothed—instead they are smoothed less.



input, 1.00000

input, 1.00000

linear, 6.00073

corner, 6.00073

linear, 1.61944

corner, 1.61944

linear, 12.6935

corner, 12.6935

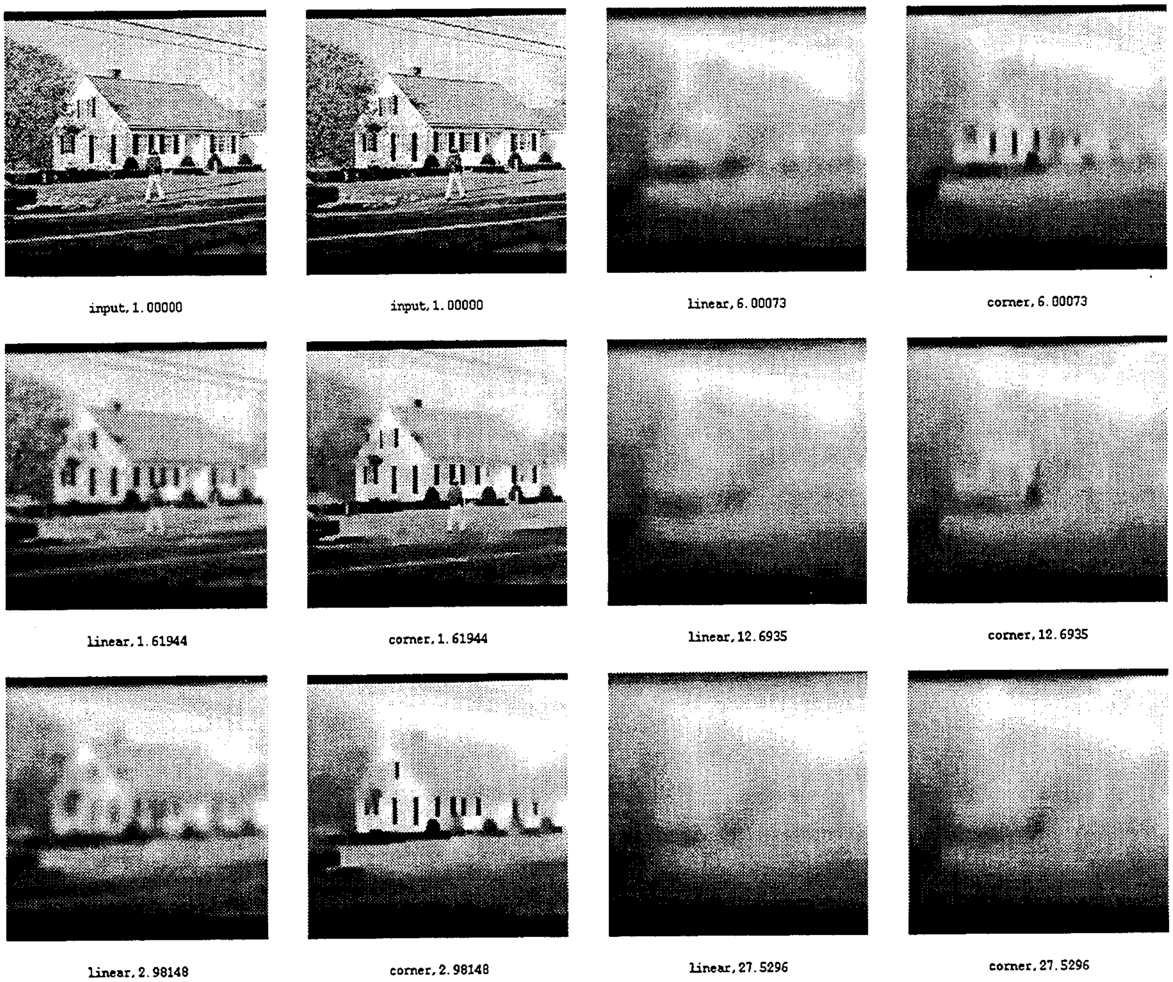
linear, 2.98148

corner, 2.98148

linear, 27.5296

corner, 27.5296

Figure 2: Many-houses image, linear and corner smoothing



input, 1.00000 input, 1.00000 linear, 6.00073 corner, 6.00073

linear, 1.61944 corner, 1.61944 linear, 12.6935 corner, 12.6935

linear, 2.98148 corner, 2.98148 linear, 27.5296 corner, 27.5296

Figure 3: One-house image, linear and corner smoothing

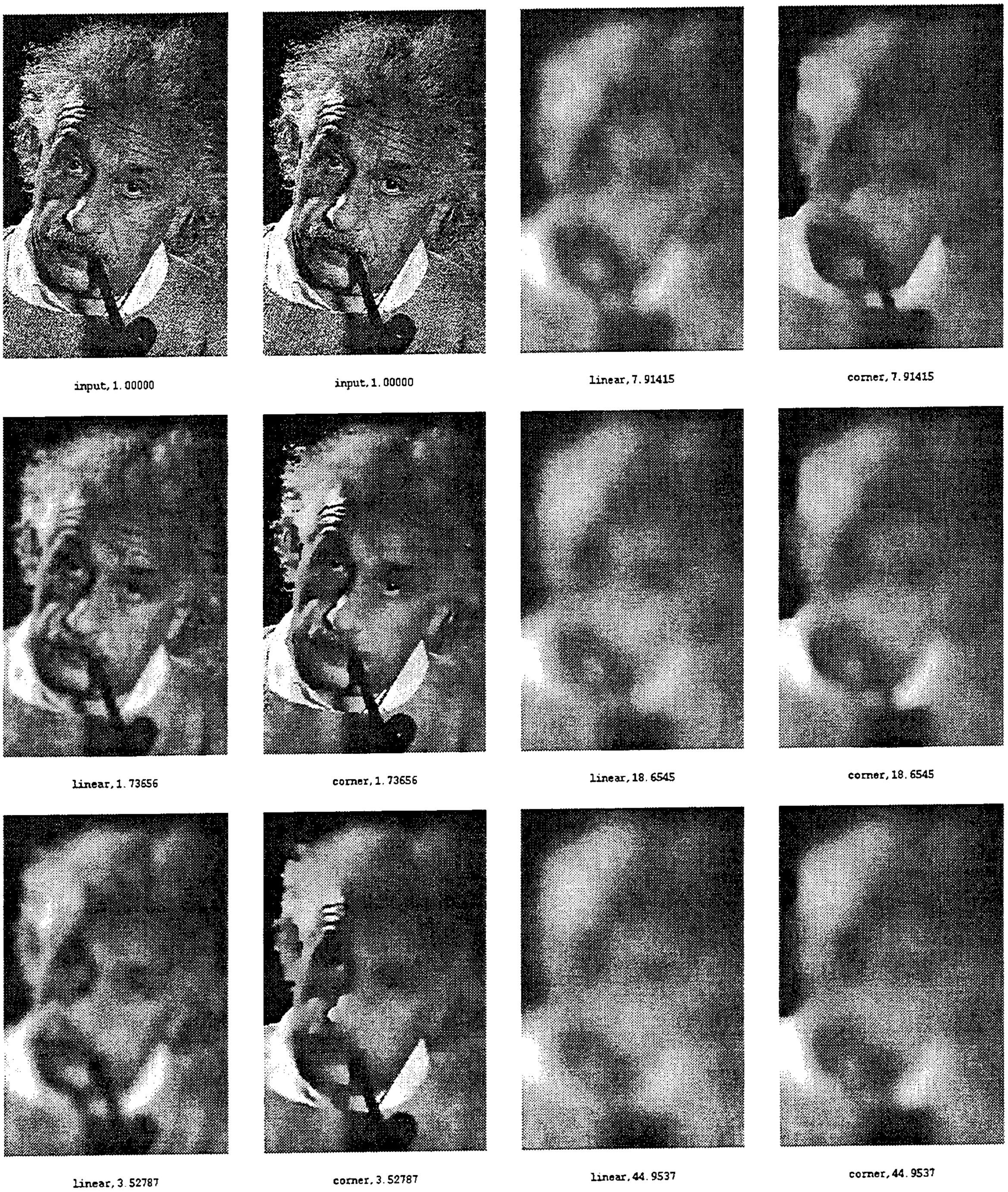
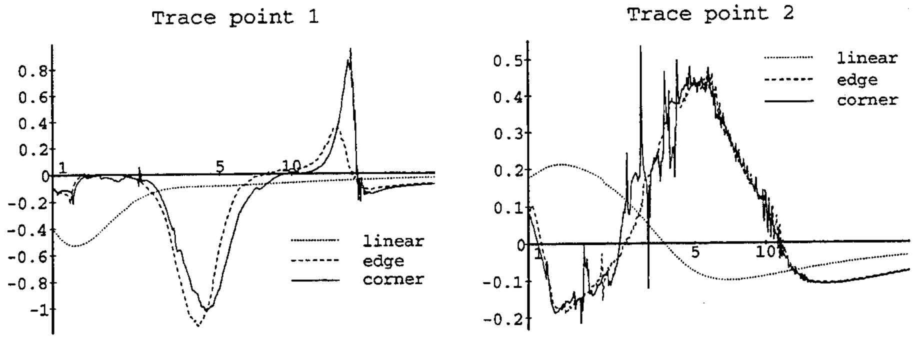


Figure 4: Einstein image, linear and corner smoothing

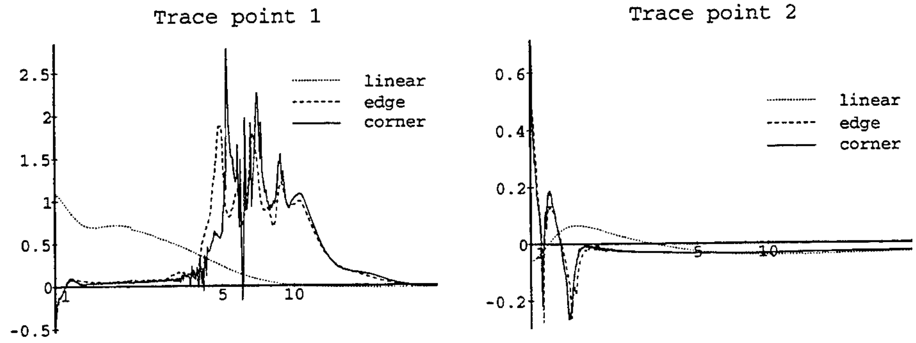
7 Graphs of Results

In this section we display some graphs (Figures 5–7). We display how much the image intensity value changes at a particular pixel value as we change scale; this is the derivative of intensity with respect to the logarithm of scale. The general result we get is that the graph for the linear smoother is in fact very smooth; normally we observe big change at small scales, the change becomes less as we increase scale, and sometimes we overshoot and the change decreases to zero and then changes sign and then starts decreasing again to zero. We show results for two different nonlinear smoothers, one just using k, which we call "edge", and the other using both b and k, which as before we call "corner". The graphs for the edge and the corner smoother are on the other hand much more peaked, especially is this true of the corner smoother, although the corner smoother still needs perfecting because there is substantial overshoot and irregularities in the corner curves. The overshoot was significantly worse before we decided to apply the summation step to the k^* field. It is also interesting to note that the threshold we need to use for the γ logarithmic transform is quite high. Most k's are not changed but very large k's can cause artifacts and slight changes in how we do the binning can affect the size of large k's. Hence we need to prevent the transformed k's from being too big but most k's must be kept the same (in fact it is mostly sharp edge points that are subject to being changed by the logarithmic transformation.


Trace point 1

Trace point 2

Figure 5: Many-houses image, $\partial f/\partial \log(\text{scale})$


Trace point 1

Trace point 2

Figure 6: One-house image, $\partial f/\partial \log(\text{scale})$

In all these figures it will be noted that the edge and corner smoother graphs will look noisy. This is true for many reasons. One is that the change recorded at a given pixel

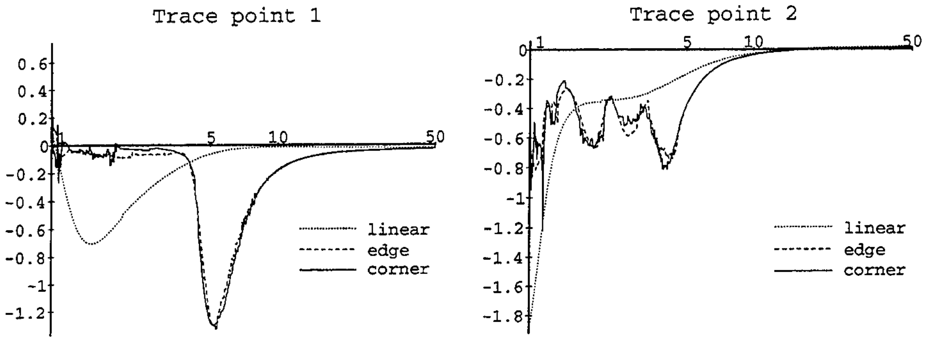


Figure 7: Einstein image, $\partial f/\partial \log(\text{scale})$

depends primarily on the nearby features of the appropriate scale and we could be at a point with many nearby features of approximately the same scale but different average intensity; these features will become prominent at slightly different scales. A more basic reason is that, as we explained while justifying the diffusion equation, it is not an exact but a noisy equation. In the nonlinear case places of big change can arise because an area which has not been much smoothed suddenly is subject to a large amount of smoothing (i.e. the edge has begun to disappear). The noise is proportional to the size of the change in intensity. We, of course, have further problems, because our estimates of k are imperfect or in the case of the corner smoother we are using two noisy diffusion equations. All this is besides the main point which is that the nonlinear smoothers cause the sizes of the changes to peak in the vicinity of the region where we would expect them to as determined by the size of the features in the vicinity of the pixel whose graph is being recorded.

8 Conclusion

This is a preliminary investigation into the statistical structure of scalespace. We have provided a simple, elegant model that underlies the computation of Gaussian scalespace. This model naturally extends to a nonlinear model if there are edges or partial edges which act as obstacles or partial obstacles to smoothing. We have not fully solved the problem of discovering the underlying nonlinear hierarchical statistical model for a given image, but we have obtained some interesting results showing how the resulting nonlinear statistical smoothers will in a simple and natural way isolate the scales of interest in the vicinity of a given pixel.

But our primary interest extends beyond the results in this paper. The model we used could be modified to allow for only binary edges (but then we would have to estimate the probability that a given pixel is an edge and again this is a continuous quantity. We could allow for a discretized gridded scalespace where the size of the grid is scale-dependent (how finely we can localize features is scale-dependent) or we could extend our model to allow for explicit representation of occlusions and transparency or take symmetries into account and construct a model that prefers symmetric regions over nonsymmetric regions. Indeed we recall that the general statistical technique of hierarchical modeling is just that, a general technique, and it should be possible to usefully apply it to any problem to which one might think to apply multigrid or continuous wavelet transform techniques. We are currently researching further applications of the hierarchical statistical paradigm.

Even the application discussed in this paper can be extended much further. We are currently investigating mean field theory approaches to parameter estimation. If we only want to estimate the expected value of a quantity, we do not need to and in fact should not try to obtain the best values for the auxiliary parameters that we need to know in order to estimate the quantity of interest. If we have some prior probabilistic information about the parameters (information that might be learned from experience with previous images) and we only try to estimate a small number of parameters per image, then we should be able to compute a mean value for the quantities of interest such as image intensity value at a given scale. And the estimate should be fairly robust. Ideally we should at the start declare what quantities we are really interested in knowing (and some of these quantities will actually be qualitative values such as whether or not a certain feature occurs within a certain distance of a certain pixel) and just estimate their mean values; this will prevent us from trying to find some exact parameter values we will never know or even from trying to estimate some edge and corner features that really do not directly interest us and are not necessary in the indirect computation of the needed mean values. There are some hard issues here that are still to be solved, but we want to assert the general value and power of the hierarchical statistical paradigm.

References

[1] J. Babaud, A.P. Witkin, M. Baudin, and R.O. Duda, "Uniqueness of the Gaussian Kernel for Scale-Space Filtering", *IEEE PAMI*, 8:26–33, 1986.

[2] M. Basserville, A. Benveniste and A. Willsky, "Multiscale Autoregressive Processes, Parts I and 2," *IEEE Signal Processing*, 40:1915–1954, 1992.

[3] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, 1987.

[4] B. Gidas, "A Renormalization Approach to Image Processing Problems", *IEEE PAMI*, 11:164–180, 1989.

[5] B.B. Kimia, A. Tannenbaum, and S.W. Zucker, "Entropy Scale Space", *Proc. Visual Form Workshop*, Plenum, 1991.

[6] J.J. Koenderink and A.J. Van Doorn, "The Structure of Images", *Biological Cybernetics* 50:363-370, 1984.

[7] H.S. Lim and T.O. Binford, "Stereo Correspondence: A Hierarchical Approach", *Proc. Image Understanding Workshop*, 234–241, 1987.

[8] T.P. Lindeberg, "Scale-Space for Discrete Signals", *IEEE PAMI*, 12:234–245, 1990.

[9] B.F. Logan, "Information in the Zero-Crossings of Bandpass Signals", *Bell System Technical Journal*, 1977.

[10] M.R. Luettgen, *Image Processing with Multiscale Stochastic Models*, MIT Ph.D. thesis.

[11] S. Mallat and S. Zhong, "Characterization of Signals from Multiscale Edges", *IEEE PAMI*, 14:710–733, 1992.

[12] D. Marr, *Vision*, Freeman, 1982.

[13] A. Montanvert, P. Meer, and A. Rosenfeld, "Hierarchical Image Analysis Using Irregular Tessellations", *IEEE PAMI*, 13:307–316, 1991.

[14] N. Nordstrom, "Biased Anisotropic Diffusion—A Unified Regularization and Diffusion Approach to Edge Detection", *Image and Vision Computing*, 8:318–327, 1990.

[15] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Nonisotropic Diffusion", *IEEE PAMI*, 12:629–639, 1990.

[16] R.W. Rodieck, *The Vertebrate Retina*, Freeman, 1973.

[17] B.M. ter Haar Romany, L.M.J. Florack, J.J. Koenderink, and M. Viergever, "Scalespace, Its Natural Operations and Differential Invariants" in A.C.F. Colchester, D.J. Hawkes (eds.), *Information Processing in Medical Imaging*, 239–251, 1991.

[18] A. Rosenfeld and M. Thurston,, "Edge and Curve Detection for Visual Scene Analysis", *IEEE Computers*, 20:562–569, 1971.

[19] G. Sapiro and A. Tannenbaum, "Affine Invariant Scale-Space", *International Journal of Computer Vision*, 11:25–44, 1993.

[20] E. Saund, "Adding Scale to the Primal Sketch", *Proc. CVPR*, 70–78, 1989.

[21] D. Stroock, *Probability Theory, an Analytic View*, Cambridge University Press, 1993.

[22] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE PAMI*, 8:129–139, 1986.

[23] R.T. Whitaker and S.M. Pizer, "A Multiscale Approach to Nonuniform Diffusion", *CVGIP Image Understanding*, 57:99–110, 1993.

[24] A.P. Witkin, "Scale-Space Filtering", *Proc. IJCAI*, 1019–1022, 1983.

[25] A. Yuille and T. Poggio, "Scaling Theorems for Zero-Crossings", *IEEE PAMI*, 8:15–25, 1986.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
October 1994

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Nonlinear Scalespace via Hierarchical Statistical Modeling

5. FUNDING NUMBERS
N00014-93-1-0257

6. AUTHOR(S)
David Shulman and Tomas Brodsky

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER

CAR-TR-742
CS-TR-3366

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The content of the information in this report does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.
Distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

Nonlinear scalespace should be based on a hierarchical statistical model of the image intensity function. This model should contain an explicit representation of the multiscale structure of edges and corners. Using this model we can have a non-ad-hoc basis for computing the parameters we need to determine how much smoothing we should do at points that appear to be edge points. We also have a basis for computing the apparent error in our scalespace calculations.

Hierarchical statistical modeling is a technique that can be applied to other problems in low-level vision, but in this introductory paper we just present the application of our scalespace theory to image smoothing.

14. SUBJECT TERMS
Hierarchical models, image models, scale space, statistical models

15. NUMBER OF PAGES
28

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	-	Contract	PR	-	Project
G	-	Grant	TA	-	Task
PE	-	Program Element	WU	-	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.