USC–SIPI REPORT #266

Texture Segmentation via Haar Fractal Feature Estimation

by

Lance M. Kaplan and C.-C. Jay Kuo

August 1994

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

Texture Segmentation via Haar Fractal Feature Estimation *

Lance M. Kaplan[†] and C.-C. Jay Kuo[†]

July 19, 1994

Abstract

We examine an approach for texture segmentation by using the fractal dimensions along the 1-D cross sections of 2-D texture data as image features, where an effective Haar transform fractal estimation algorithm is utilized. The major advantage of the Haar fractal estimator is its computational efficiency along with robustness. The method is fast due to the pyramid structure of the Haar transform and nearly optimal in the maximum likelihood sense for fBm data. We compare the low complexity of this new algorithm with the complexity of existing fractal feature extraction techniques, and test our new method on fBm data, real Brodatz textures, and natural scenes.

1 Introduction

Over the past years, an unified definition of texture has been elusive. One can think of textures as images which contain some random or deterministic patterns. Two textures are distinct if they can be separated visually. Examples of textures include grass, cloth, mammographic images, and clouds. While for most cases, the eye can easily distinguish between two textures, the classification and segmentation of textures by a computer has proven to be a challenging problem. The technology of texture classification and segmentation is useful for the automation of quality control in industrial monitoring, searching for earth resources in remote sensing, medical diagnosis using computer tomography, and target detection in SAR images.

Early research work such as SGLDM [16] and correlation [8], [12] was based on the second-order statistics of textures. In the 80's, researchers developed the Gaussian Markov Random Field (GMRF) [9] and Gibbs distribution [15] texture models, where the gray levels

[*] This work was supported by the National Science Foundation Presidential Faculty Fellow (PFF) Award ASC-9350309.

[†] The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California 90089-2564. E-mail: lancekap@sipi.usc.edu and cckuo@sipi.usc.edu.

between nearest neighboring pixels are characterized by a certain stochastic relationship. Laws [26] proposed a simple scheme which used local linear transformations and energy computation to extract texture features. The simple scheme often gives reasonably good performance, and has been studied and improved by many researchers [18], [38].

Many textures like cloth can be modeled as quasi-periodic. As a result, many researchers have concentrated on multichannel or multiresolution analysis to capture features representing the dominant harmonics [2], [39]. These methods often outperform traditional methods based on the second-order statistics or the GMRF model. The main problem of traditional texture analysis algorithms is that they did not properly capture the features at different scales. Recent work has concentrated around spatial/frequency or spatial/scale analysis. For instance, algorithms using the Gabor transform have been reported [3], [37]. More recently, Chang and Kuo [5] utilized a wavelet packet scheme to adaptively search out the dominate frequency bands. While the subband techniques are good to classify a wide variety of textures, most of the algorithms fail to distinguish between many natural textures.

Many natural textures have power spectrums which follow a $1/f$ law. As a result, these textures only have a dominant DC band. Flandrin [13] has shown a connection between the fractal dimension and the rate of decay of a texture's power spectrum. Pentland [33] and Kumar *et al.* [25] have demonstrated the connection between fractal dimension and surface roughness. As a result, many researchers have utilized the fractal dimension as a feature to distinguish natural textures. For example, texture classification algorithms were developed for medical applications [7], [28]. Texture segmentation algorithms based on fractal dimension estimates are presented in [6], [17], [21], [23], [33], [35]. Moreover, a fractal dimension estimator was used for speech segmentation [10]. While many algorithms measure the fractal dimension of surfaces or curves by the box counting method [6], [23], many others use a model based approach with the fractional Brownian motion (fBm) model [17], [35].

The fBm has proven to be a very good model for the analysis and synthesis of natural phenomena [29], [30]. For instance, fractal models have been successfully applied to texture analysis and synthesis [22], [33] and landscape modeling [32]. In this research, we propose a fast Haar fractal estimator to capture features for texture segmentation based on the fBm model. The major advantage of the Haar fractal estimator is its computational efficiency along with robustness. The pyramid structure of the Haar transform provides a fast implementation of the fractal dimension estimator. We will demonstrate the robustness of the algorithm in Section 3.

This paper is organized as follows. In Section 2 , we give a review of the properties of fractals, fBm, and the Haar transform applied to fBm. The new Haar fractal feature extraction algorithm is detailed in Section 3. The texture segmentation results are shown in Section 4, and some concluding remarks are given in Section 5.

2 Fractals, Fractional Brownian Motion and Haar Transform

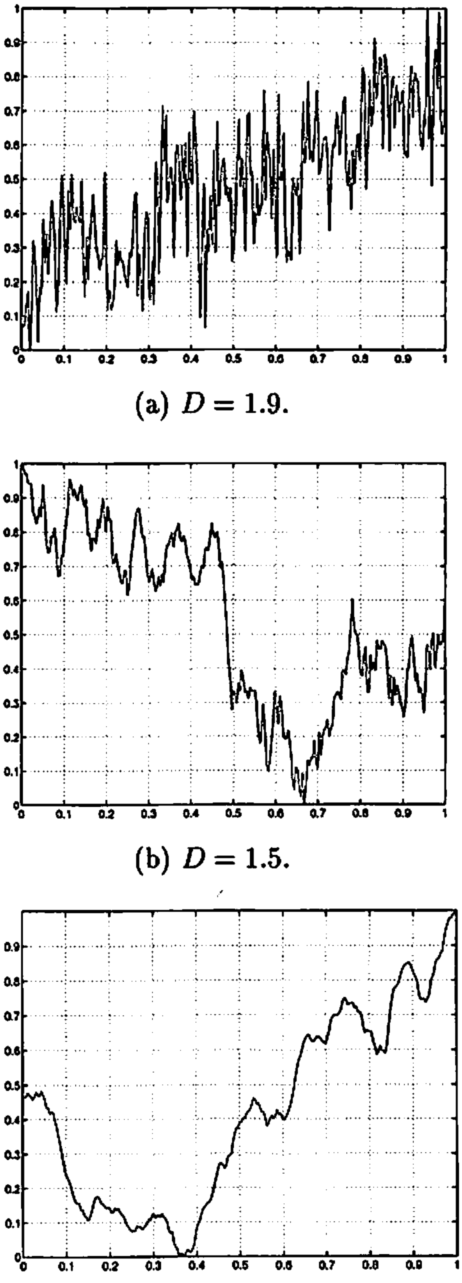
2.1 Fractal Curves and Surfaces

To model the textured details of nature, Mandelbrot popularized fractal geometry [29]. He demonstrated how self-similar patterns can mimic patterns found in nature. A strict mathematical treatment of fractals can be found in [11]. Fractals are defined as objects, or sets, whose Hausdorff dimension (or fractal dimension) is greater than its topological dimension. In this paper, an image is treated as surface with a topological dimension of two. The surface is defined over a square lattice that represent pixels, and the gray level of the pixels represent the location of the surface on the z axis. Then, any cross section of the surface represents a curve whose topological dimension is one. The fractal dimension of the surface (or curve) can loosely be interpreted as the amount of three (or two) dimensional space that the object occupies. As a result, the fractal dimension provides quantitative information about the roughness of the surface (or curve). For example, Fig. 1 displays curves of varying fractal dimension defined over the interval $[0, 1]$. The curves all have a minimum value of zero and a maximum value of one. It is easy to see that the curves of higher fractal dimension appear rougher and cover more space in the one square unit of area.

As shown in [29], the fractal dimension can be determined by trying to cover the object of interest with balls of diameter ϵ. If the diameter is small enough, the number of balls of size ϵ to cover the object is related to the fractal dimension d by

$$N(\epsilon) \propto \epsilon^{-d}. \tag{2.1}$$

Obviously, as ϵ goes to zero, a larger fractal dimension d means that a larger the number of balls or space is necessary to cover the object.



(a) $D = 1.9$.

(b) $D = 1.5$.

(c) $D = 1.1$.

Figure 1: Fractal curves of various fractal dimensions.

2.2 Fractional Brownian Motion

Fractional Brownian motion (fBm) is the most common stochastic model for random fractal textures. Many properties of fBm have been examined, say, [1], [13], [14], [19], [30], [32], [34], [36], and [40]. We will review some basic results of 2D-fBm in this section. Two dimensional fBm $B_H(\vec{t})$ is a zero-mean Gaussian random process such that,

$$B_H(\vec{0}) = 0, \tag{2.2}$$

and

$$\mathrm{var}[B_H(\vec{t}+\vec{s}) - B_H(\vec{t})] = \sigma^2 \|\vec{s}\|_2^{2H}, \tag{2.3}$$

where $0 < H < 1$, and $\|\cdot\|_2$ is the standard Euclidean norm. Condition (2.3) is known as the self-similarity condition, and it means that the variance of any increments is independent of orientation and dependent only on the length of the increment. The parameter H is known as the Hurst parameter, and the bounds on H assure that the fBm process is mean squared continuous and that the correlation function possesses the properties of an inner product [30]. The self-similarity condition leads to the fact that the realizations of 2D-fBm are fractals surfaces whose fractal dimension is related to H via [32]

$$D = 3 - H.$$

Thus, the Hurst parameter determines the visual roughness of the fBm realization. Moreover, an estimate of H also provides an estimate of the fractal dimension. Based on (2.2) and (2.3), the correlation function of $B_H(\vec{t})$ can be derived as

$$r_{B_H}(\vec{s},\vec{t}) = \frac{\sigma^2}{2}(\|\vec{s}\|^{2H} + \|\vec{t}\|^{2H} - \|\vec{s}-\vec{t}\|^{2H}).$$

Thus, fBm is a nonstationary process, and if $H = 1/2$, the fBm process has the correlation structure of normal Brownian motion.

A nice property of 2D fBm is that each one dimensional slice forms a 1D fBm curve. Specifically, given a normalized directional vector $\vec{n}$ and an offset $\vec{o}$, the 1D slice of fBm,

$$B_H^{(1)}(t) = B_H(t\vec{n} + \vec{o}),$$

satisfies the self-similarity condition. While $B_H^{(1)}(t)$ is still a nonstationary process, its increments,

$$X_H(m, \Delta x) = B_H^{(1)}(\Delta x(m+1)) - B_H^{(1)}(\Delta x m), \qquad m \in \mathrm{Z},$$

form a stationary process. The fBm increments defines a sequence known as *discrete fractional Gaussian noise (dfGn).* The correlation function for dfGn of incremental length Δx is

$$r_{X_H}(k, \Delta x) = \frac{\sigma^2}{2}|\Delta x|^{2H}(|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H}). \tag{2.4}$$

Due to stochastic self-similarity, the shape of the correlation function or power spectrum of dfGn is the same for all values of the incremental length. Moreover, dfGn can be divided into three classes. Namely, when $H < 1/2$, the increments are negatively correlated, when $H > 1/2$, the increments are positively correlated, and when $H = 1/2$, the increments are uncorrelated (or white).

Although fBm is nonstationary and thus has no formal power spectrum, Flandrin [13] showed that the "average" power spectrum of 1D fBm follows a $1/f$ law, i.e.

$$S(f) \propto \frac{1}{f^{2H+1}}. \tag{2.5}$$

The spectral behavior of fBm provides good motivation to apply fBm as a model for $1/f$ processes.

2.3 The Haar Transform and FBM

The Haar transform is the most basic member of the class of orthogonal wavelet transforms, and it is the easiest to implement in hardware. In previous work [19], we have shown some very useful properties of the wavelet coefficients when the Haar transform is applied to discrete fGn (dfGn). We summarize some main results in this section.

The Haar transform provides a multi-scale representation of a signal where coefficients defining the signal details at different scale are computed. Given an approximation of the signal at the finest available scale $a_0(t)$ which is usually the discrete signal samples, the approximation and detail coefficients of the signal at the next coarser level is computed for $m \geq 0$ by

$$a_{m+1}(t) = a_m(2t) + a_m(2t+1), \tag{2.6}$$

$$d_{m+1}(t) = a_m(2t) - a_m(2t+1). \tag{2.7}$$

Equations (2.6) and (2.7) are computed recursively to obtain detail wavelet coefficients for different scales. It is known that the the Haar transform can effectively *whiten* the stationary dfGn process [19]. It is important to point out, however, that the Haar transform cannot whiten 1D fBm for $0 < H < 1$. Thus, the simplicity of the Haar transform cannot be exploited by a fractal wavelet estimator applied directly to the fBm data [41]. Based on our previous results [19], we can show some nice properties of the Haar transform applied to dfGn. When $a_0(t)$ is set equal to the dfGn process, the approximation coefficients of scale m are equivalent to the increments of the fBm curve for an incremental length of $\Delta x = 2^m$. By (2.4), the correlation function of $a_m(t)$ is the same as that of $a_0(t)$ scaled by a factor of 2^{2Hm}. Then, it can be shown that the variance of the detail wavelet coefficients $d_m(t)$ are related to scale by

$$\sigma_m^2 = \text{var}[d_m(t)] = 2^{2H(m-1)}\sigma^2(4 - 2^{2H}). \qquad (2.8)$$

The simple exponential variance progression of the detail wavelet coefficients via the discrete wavelet transform applied to dfGn using the Haar basis does not generalize to higher order (e.g. more regular) filters. Besides, the Haar transform provides for a natural separation of data segments as discussed in Section 3. Thus, we concentrate on the Haar transform.

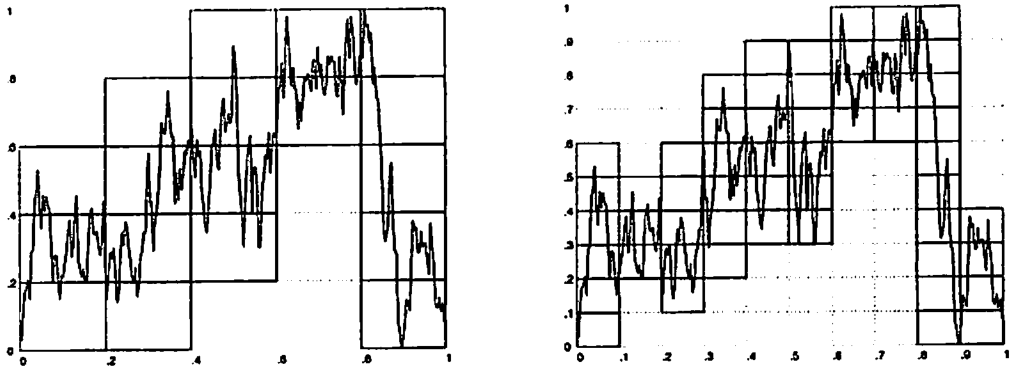
3 Fractal Feature Estimation via Haar Transform

For 2-D fractal surfaces, the fractal dimension can be measured by considering the surface to be isotropic where measurements are averaged over all possible directions. Another choice is to consider 1-D cross sections of the surface as fractal curves and measure the fractal dimension of the curves. The non-isotropic measurement technique provides more information than the isotropic counterpart when a real texture appears rougher on a preferred direction. Moreover, the fractal estimation of image strips can be computed in parallel to reduce the computation time. Thus, we consider fractal dimension estimators using the directional measurement approach in this work.

3.1 Previous Fractal Dimension Estimators

Many methods to calculate the fractal dimension of a fractal curve have been given in the literature. Three of the most popular algorithms are the box counting, the variance scaling, and the power spectral methods [33]. All three methods require that linear or nonlinear regression is used on the logarithm of some measurements versus the logarithm of the scale where the calculated slope is the estimate for H.

The basic idea behind the box counting method is to count the number of boxes on a grid of size ϵ necessary to cover a function on a compact interval as shown in Fig. 2. Since the number of boxes to cover the curve should follow the law given by (2.1), the slope of the



(a) $\epsilon = 0.2$.

(a) $\epsilon = 0.1$.

Figure 2: Box counting at different scales.

curve of $\log(N(\epsilon))$ versus $\log(\epsilon)$ is the fractal dimension. Usually, one uses linear regression to calculate the slope. In reality, the box counting dimension is greater than or equal to the true fractal dimension.

The power spectral method uses the $1/f$ law for the power spectrum (i.e. (2.5)) to relate the coefficients of the fast Fourier transform (FFT) to the frequency. Specifically, the slope of the log of the power spectrum versus the log of the frequency is calculated via linear regression. Then, the Hurst parameter or fractal dimension is obtained based upon the slope.

The variance method uses the self-similarity condition (2.3) where the variance of the increments for various lags are plotted as a function of the lag. Again, the slope of the log of the variance versus the log of the lag determines the Hurst parameter or fractal dimension. While the variance method is motivated by the self-similarity condition of the fBm, it does provide a good fractal dimension estimate for general fractal functions. Besides, the variance method can be related to the box counting method. For instance, the variance method measures the expected change in a function value over a distance of ϵ. To cover the function in this interval of size ϵ will require about $\sqrt{\mathrm{VAR}[B(s+\epsilon) - B(s)]}/\epsilon$ boxes. Then the total number of boxes to cover the function should be about

$$N(\epsilon) \approx K\sqrt{\mathrm{VAR}[B(s+\epsilon) - B(s)]}/\epsilon^2.$$

By using the self-similarity condition (2.3) and (2.1), it is easy to understand the relation between fractal dimension and H for a fractal curve, i.e. $D = 2 - H$.

3.2 Haar Fractal Dimension Estimation

In this section, we describe the Haar fractal estimator and its detailed implementation.

3.2.1 Haar Fractal Dimension Estimator

Given the Haar transform of a dfGn sample and using the assumption that the detail Haar coefficients are uncorrelated, one can compute the log-likelihood function as

$$L(\sigma^2, H) = -\frac{1}{2}\sum_{m=1}^{M} N(m)\left[\frac{\hat{\sigma}_m^2}{\sigma_m^2} + \ln(2\pi\sigma_m^2)\right], \tag{3.1}$$

where $N(m)$ is the number of wavelet coefficients available at scale m (usually $N(m) = 2^{M-m}$), $\hat{\sigma}_m^2$ is the sample variance of the wavelet coefficients, σ_m^2 is given by (2.8), and M is the number of scales. We maximize the likelihood function to obtain the estimate of H. As shown in the Appendix, the estimate of H is found by first solving the following polynomial for β,

$$\sum_{m=1}^{M} C_m N(m)\hat{\sigma}_m^2 \beta^{M-m} = 0, \tag{3.2}$$

where

$$C_m = \frac{m}{\sum_{m=1}^{M} mN(m)} - \frac{1}{\sum_{m=1}^{M} N(m)}.$$

Then, the Hurst parameter which maximizes the likelihood function is related to β by

$$\hat{H} = \frac{1}{2}\log_2(\beta). \tag{3.3}$$

The Haar fractal method can be directly compared to the variance method. By examining (2.6) for the case that the finest scale approximation coefficients correspond to the increments of length one, one can see that the variance method in effect performs the regression analysis on the approximation coefficients. In other words, the Haar method adds the wavelet filtering step (2.7) to "whiten" the coefficients so that a maximum likelihood estimate of H is easy to perform. Even without the maximum likelihood formulation, a linear regression analysis after the wavelet filtering step should provide more accurate estimates because the coefficients are virtually independent.

3.2.2 Efficient Implementation

To compute the local fractal measurement in the y direction, we take a 9×17 block (or a block consisting of 9 columns and 17 rows). For each 1×17 slice of the block, we compute the 16 increments, pass the 16 increment through the four level Haar transform, and compute the sample variance of the detail Haar coefficients at each scale. Then, the third order polynomial (3.2) is solved, and $\hat{H}$ is calculated. The nine different estimates of H are averaged to provide a local y-directed Hurst parameter H_y for the middle pixel in the 9×17 block of concern. We then slide the block over the entire image to compute local fractal measurements for every pixel. Similarly, we use a 17×9 block to compute the x-directed Hurst parameter H_x. More generally, one can consider rotations of the 9×17 block to calculate the directed Hurst parameters at other angles. Although we are concentrated on the x- and y-directed Hurst parameters in the experiments in Section 4, situations may arise where diagonal directions may provide additional interesting texture features in the real world problems.

To compute the local fractal features effectively, one can divide an $N \times N$ image into strips of width one pixel and length N pixels. The orientation of the strips depends on the directional Hurst parameter that one wants to calculate. For each strip, the increments are computed, and $\log_2(W)$ levels of the Haar transform are applied to the increments via (2.6) and (2.7). We use $W = 16$ and, therefore, 4-level Haar transform in our experiments. The energy of the detail Haar coefficients is computed for each scale in each block, and Equation (3.2) is used to calculate the local Hurst parameter for each block.

The pyramid structure of the Haar transform provides a fast method to slide the analysis window block. To illustrate this idea, we show the pyramid structure of the Haar transform for $W = 8$ in Fig. 3 as an example. First, the local Hurst parameters for the pixels marked with zero are calculated. In the next step, the local Hurst features for the pixels marked with one are computed. This can be easily achieved by switching the neighbor of the second coarsest approximation coefficients used to calculate the coarsest wavelet coefficients. The new calculation scheme is indicated by the broken lines in Fig. 3. Once the coarsest coefficients are modified, (3.2) is reevaluated. In general, a new step will slide the analysis window by 2^m pixels when the approximation coefficients neighbor grouping at scale m is switched, and the wavelet coefficients and energy values are recalculated by (2.6) and (2.7). Fig. 3 shows the pixel ordering for the feature calculation. It turns out that the local fractal features for each pixel in a W length segment is computed by considering shifting

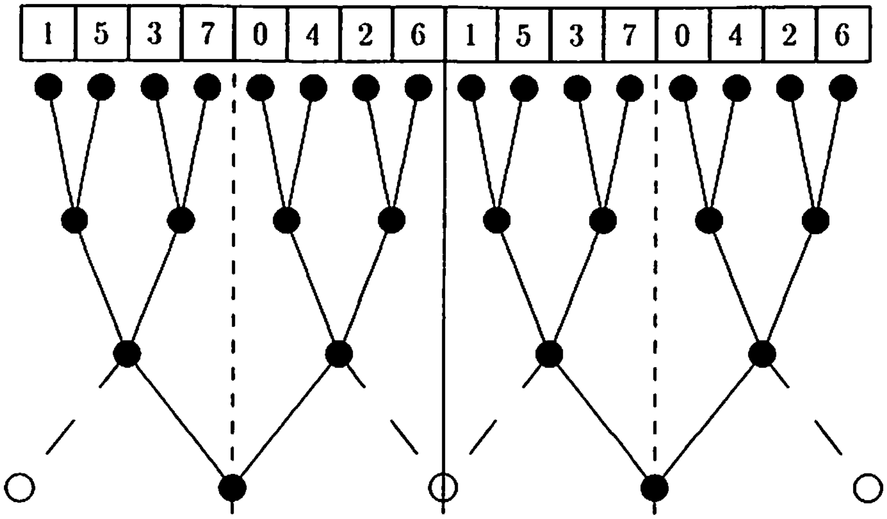


Figure 3: Segmentation of the Haar transform for $W = 8$.

the analysis window by the bit reversed value of the step number.

After the wavelet features are computed for all N strips, V strips are averaged by a sliding window. When $V = 9$ and $W = 16$, it is the method described above and used in the experiment. Note that to calculate the Hurst and incremental power features for all pixels in the image, one must consider extensions around the image borders. In our work, we consider the image to extend symmetrically about all the edges.

3.2.3 Computational Complexity

When the method presented above is used to calculate the local fractal features, only a limited number of coefficients need to be altered for each step before the nonlinear regression analysis. Table 1 shows the number of times the energy of the coefficients for each scale needs to be calculated and the complexity to compute the energy values at each scale for $W = 16$. In general, our wavelet extraction method takes about $4W \log_2(W)$ flops for each of the N/W segments in a strip. Given N strips, the total cost to compute one directional local fractal dimension measurement for each pixel is about $4 \log_2(W)N^2$ flops. The cost per pixel is only $O(\log_2(W))$.

The computations necessary to set up the regression analysis are higher by using other fractal methods. For instance, the box counting method must be computed in $O(W)$ flops per pixel. The power spectral method has a complexity of $O(W \log_2(W))$ per pixel because

Scale (m)	Calculation of (2.6) in Flops	Calculation of (2.7) in Flops	Calculation of Energy in Flops	Flops to Calculate Energy Value	Number of Times Energy Values are Calculated	Total Flop Count
1	8	8	15 (≈ 16)	31 (≈ 32)	2	62 (≈ 64)
2	4	4	7 (≈ 8)	15 (≈ 16)	4	60 (≈ 64)
3	2	2	3 (≈ 4)	7 (≈ 8)	8	56 (≈ 64)
4	1	1	1 (≈ 2)	3 (≈ 4)	16	48 (≈ 64)

Table 1: Number of flops to implement the Haar fractal dimension extraction method on a segment of size $W = 16$.

one must compute an FFT of length W for each pixel. A straightforward implementation of the variance method would result in a complexity of $O(W^2)$. However, the variance scaling can be implemented in about $3W$ flops per pixel. For our experiments where we use segments of length $W = 16$, the approximate flop count to set up the regression analysis is about 48 flops for variance scaling and only about 16 flops for the Haar transform. Note that computation of the V length smoothing filter is negligible in our analysis.

An important point is that the directionally based fractal estimators can be computed in parallel. Specifically, fractal estimates on each strip can be computed separately from other strips. Moreover, the averaging step to compute the final Hurst estimate at each pixel can also be computed in parallel.

3.2.4 Robustness of Haar Fractal Dimension Estimator

We tested the robustness of the proposed wavelet method on true fBm images of size 256 × 256. The synthetic fBm images were generated by using a new spectral synthesis technique that creates realization which represent the true statistics of fBm [20]. The means and standard deviations of the x and y oriented fractal features over all 65,536 pixels are listed in Table 2 with various values of H. These statistics are comparable to results obtained by using isotropic fractal feature methods [35]. The fact that the Hurst parameter is always underestimated may be due to the nonlinear regression analysis performed by (3.2). Most importantly, the measurements of H appear consistent. Table 2 also shows that as H approaches one, the wavelet estimator error increases. This phenomena is most likely due to the fact that as H increases, the correlation of pixels increase. In fact, other experiments that we have performed may suggest that the field of local Hurst parameter

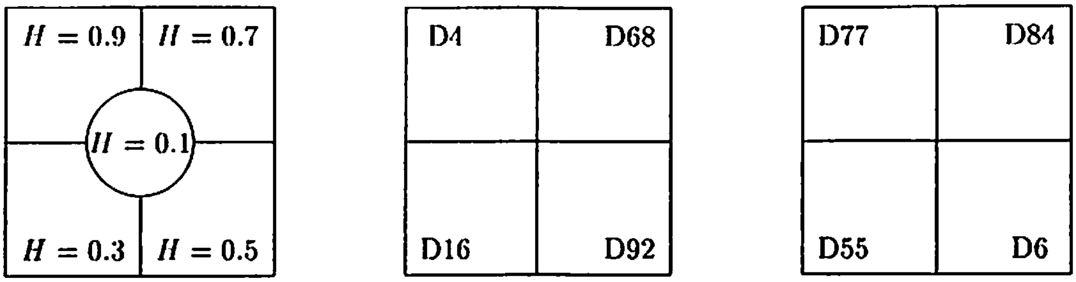
Fractal	True	$\hat{H}_x$		$\hat{H}_y$	
Dimension	H	mean	std.	mean	std
2.9	0.1	.053544	.111746	.047463	.110777
2.8	0.2	.146877	.118725	.149234	.123621
2.7	0.3	.248023	.128212	.245142	.132294
2.6	0.4	.353126	.143719	.357044	.145934
2.5	0.5	.453680	.153522	.441098	.152108
2.4	0.6	.540792	.165553	.549654	.164069
2.3	0.7	.649483	.170730	.642618	.170432
2.2	0.8	.740592	.179003	.728553	.175834
2.1	0.9	.804649	.181956	.810099	.190086

Table 2: Results of the Haar fractal dimension estimator applied to real fBm realizations.

estimates does not form an ergodic process when $H > 1/2$. The same problem was reported by Hoefer *et al.* [17].

3.3 Fractal Power Estimation

Medioni and Yasumoto have pointed out that the fractal dimension alone will not classify or segment all real textures [31]. Thus, we also consider the estimate of the fractal power parameter i.e. σ^2 in (2.3), as a feature for texture segmentation. In fact, the fractal power parameter is similar to a morphometric parameter that is used in geology to determine the roughness of landscapes [24]. While the power parameter is sensitive to the contrast of the texture, one can assume the lighting condition are fairly homogeneous over the entire texture so that the power parameter is a valid feature in the segmentation problem. To estimate the fractal power parameter, we compute the x- and y-directed increments of an image and calculate the average energy of each of the two increments over a 9×9 block. Note that efficient computation of the averaging window can be accomplished by exploiting its separable property. Then, we use the logarithm of the two average energies as two local features for the middle pixel. The logarithm is used so that the resulting features vary linearly instead of exponentially with respect to H when two fBm textures have equal dynamic range for their grey level values. Finally, we slide the 9×9 block over the entire image.



(a) fBm (b) Random Brodatz (c) Periodic Brodatz

Figure 4: Texture segment labels for the three test images.

4 Experimental Results

4.1 Segmentation of Texture Mosaics

The fractal features can be combined with any other texture features to perform texture classification of segmentation. To validate, the new fractal features, we used only the x- and y-directed Hurst and incremental power estimates for a total of four features. The segmentation results that we present may be improved by adding other features such as subband features obtained from Laws masks.

To perform the segmentation, we first smoothed the four computed features via a 9×9 EPNSQ filter [18]. Based upon the smoothed features, we search for different clusters. In our experiments, we assume that the number N of texture types in an image is known a priori, and we apply algorithms which search for N clusters in the feature space. To search for N clusters, we employ a splitting algorithm [27] to perform an initial search for cluster centers. Then, we apply a k-means algorithm to accurately find the centroids of the clusters. Each pixel is classified to the closest cluster centroid in the feature space. The segmented image contains N gray level where each gray level represents the texture class of the pixel. Since one can assume that the actual segments are more than just a few pixels in size, postsmoothing is performed to remove the small spots that may appear in the segmented image. In our work, we used a mode filter such that a pixel's gray level is changed to the gray level that appear most frequently in a 9×9 surrounding window. The mode filter has many similar properties to the median filter such as the ability to preserve boundaries.

The segmentation method was applied to three images of size 256×256, and the ground truth of the segmented textures is displayed in Fig. 4.

Test Image 1: Texture mosaic with 5 fBm images.

The first test image is a texture mosaic with 5 synthetic fBm textures of varying H as shown in Fig. 5 (a). The actual segmentation result is provided in Fig. 5(b). A similar problem was tested by Hoefer, *et al.* [17]. Fig. 5(b) shows that no misclassifications occurred in the interior of the different textures. There are segmentation errors along the texture borders. It is evident that the border effects are strongest between segments with large differences between their Hurst parameter. These edge effects occur because the calculation of the incremental power is an edge detecting operator, and in many cases, the local Hurst parameter will have different values along texture edges (see Section 4.2).

Test Image 2: Texture mosaic with 4 random Brodatz textures.

The test image contains four random Brodatz textures [4], i.e. pressed cork (D4), herring bone (D16), pigskin (D92) and wood grain (D68), as shown in Fig. 6(a). The second experiment was designed to test the ability to classify natural random textures which do not contain obvious harmonics. The segmentation result is depicted in Fig. 6(b). For this test case, the directional fractal dimension measurements were very useful to separate the wood texture as shown in Figs. 7(a) and (b). The border effects are not as pronounced as in Test Image 1 because the random textures do not provide an accurate fit to the fBm model. Unfortunately, another result of the lack of a perfect model fit is sporadic classification errors in the interior of the different textures. The misclassification between pressed cork and pigskin is understandable since these textures are visually similar. The misclassification between the wood to the herring bone is a result of some repeating fine grain structure of the wood. Fig. 7 shows that the y-directed Hurst parameter cannot distinguish between these two textures and that the x-directed Hurst provides classification errors in the wood texture along the left border of the wood.

Test Image 3: Texture mosaic with 4 periodic Brodatz textures.

The test image includes four Brodatz textures which represent woven aluminum wire (D6), straw matting (D55), cotton canvas (D77), and raffia (D84) as shown in Fig. 8(a). The segmentation result is provided in Fig. 8(b). Even though the fractal features are designed to distinguish the random textures via roughness, they seemed to do a very good job in segmenting periodic type of textures as demonstrated by this example. The segment border effects are the least pronounced for this test case. The only significant errors occur in the interior of the raffia texture where the texture is classified as cotton canvas. It should be noted that the weave between these two textures do appear alike except that the strands

Image	Number of Textures	Classification Accuracy
1. fBm Collage	5	95.15%
2. Random Brodatz	4	91.32%
3. Periodic Brodatz	4	95.62%

Table 3: Classification accuracy of different test images.

of the raffia are larger.

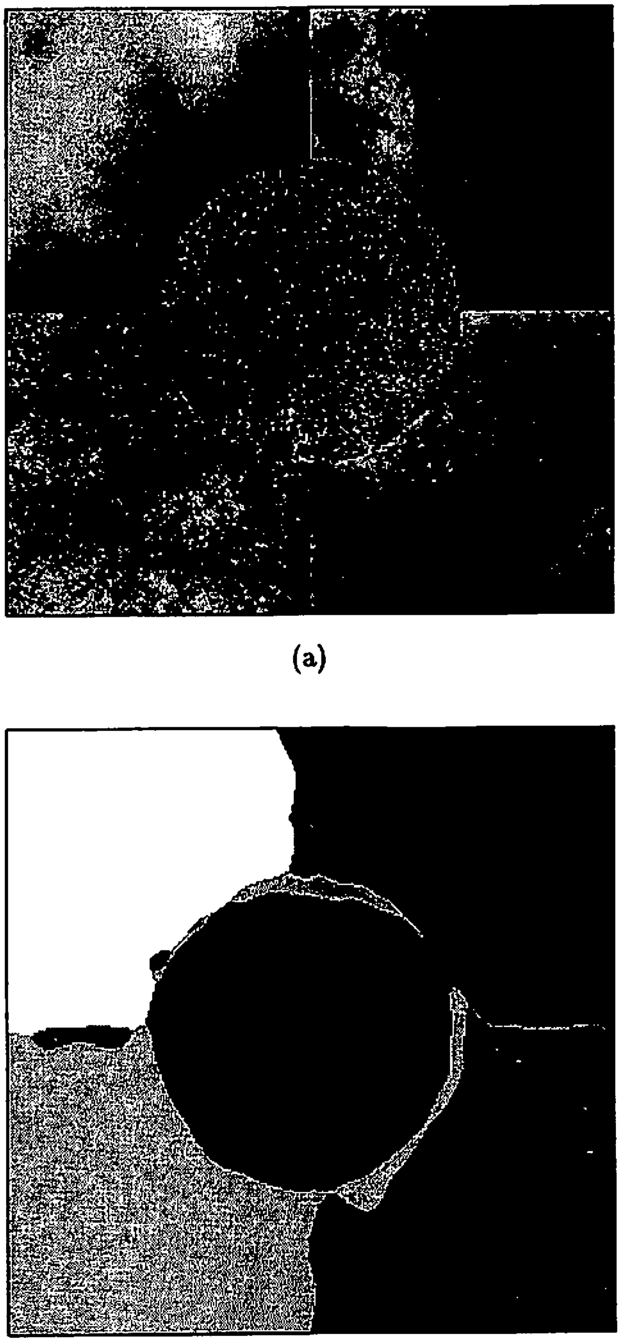
The segmentation results of the three test cases are very good. They are comparable to other fractal segmentation results as provided in [6]. The segmentation results of this section do show that our fractal feature extraction method provides useful features for texture segmentation. The classification accuracy of the three test cases is shown in Table 3. The segmentation results can be further improved by considering to measure some local multifractal or lacunarity parameter as used in [6] and [23] respectively. These parameters help to segment textures based upon "sparseness" while the fractal features measure "roughness."

4.2 Natural Scenes

The incremental power feature is very useful for segmenting water and land from aerial photographs. The fractal feature, itself, will not distinguish land and water, but the feature does provide higher values for the Hurst estimate at the water/land boundaries. The edge detection abilities of the fractal feature have been documented in [7], [33]. In our application, the edges will allow for further refinement of the segmented photograph when the incremental power feature of the land dominates a small lake or river.

Fig. 9(a) shows an aerial view of the San Francisco Bay Area in an image of size 512 × 512. The average of the x- and y-directed power features of the San Francisco image is displayed in Fig. 9(b), and the histogram of Fig. 9(b) is shown in Fig. 9(c). Based on the histogram, we choose the gray level 130 as a threshold between the water and the land. The initial segmentation result is given in Fig. 9(d). The figure shows good separation of the land and water. However, some smaller areas of water such as the the river (Area D) and bay inlet (Area E) disappear.

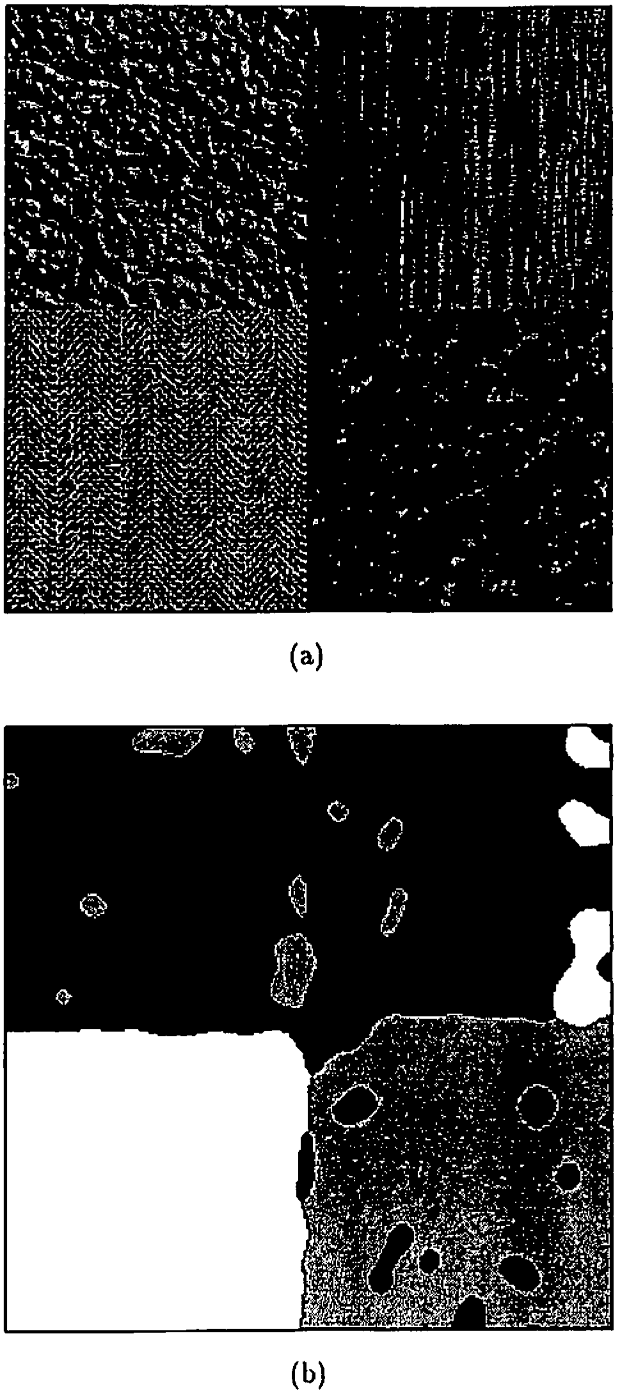
While the incremental power feature provide good segmentation, the result can be im-



(a)

(b)

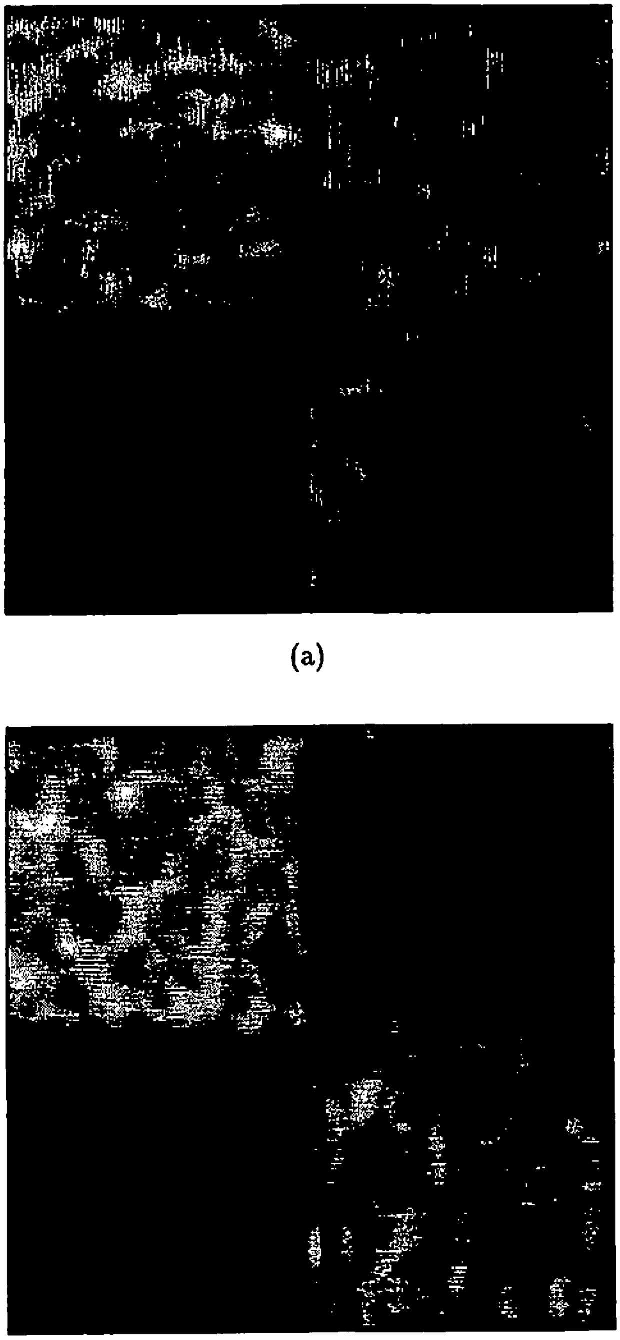
Figure 5: (a) FBm test image and (b) segmentation result.



(a)

(b)

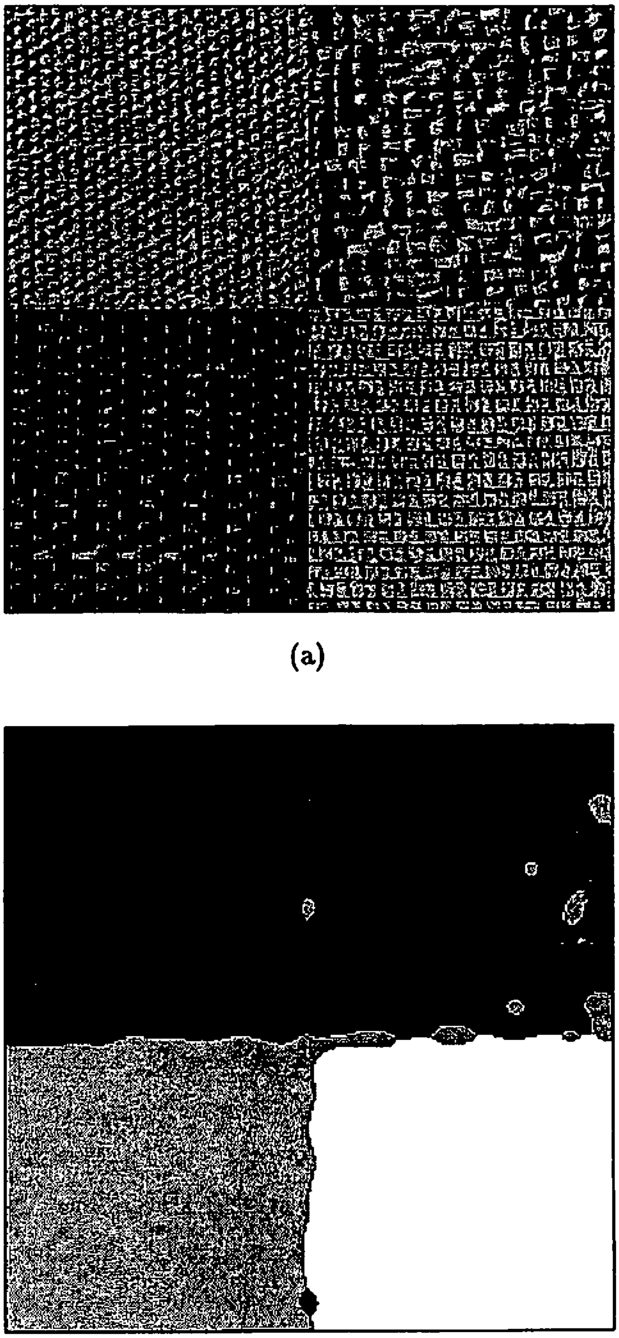
Figure 6: (a) texture mosaic consisting of random Brodatz textures, and (b) segmentation result.



(a)

(b)

Figure 7: Smoothed (a) x- and (b) y- directed Hurst features.



(a)

(b)

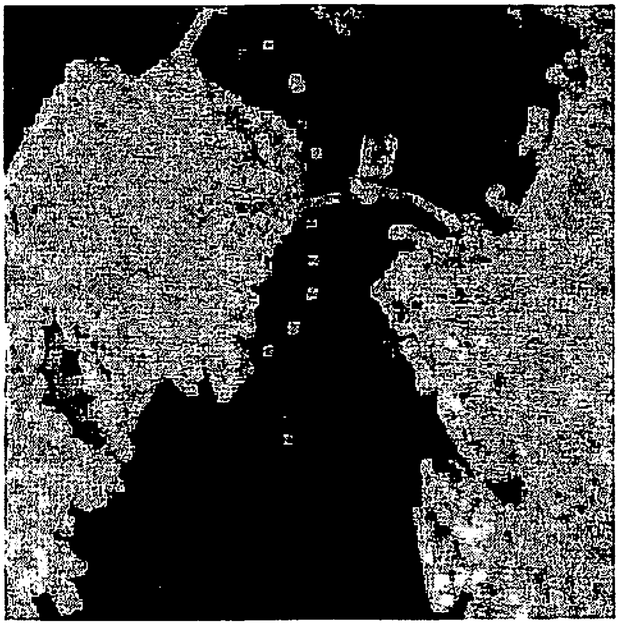
Figure 8: (a) texture mosaic consisting of periodic Brodatz textures, and (b) segmentation result.

proved by integrating fractal dimension estimates. The x-directed Hurst feature of the San Francisco image is given in Fig. 9(e). The figure does provide larger value for H around the water/land interfaces. In fact, edges appear in the figure that represent the river and water inlet which were misclassified as land. To integrate the fractal feature to obtain a final segmentation result for the photograph, we first used a threshold to consider only the largest 2.5% values of the x- and y-directed values of H to represent edges. Then, we applied one step of mode filtering to remove any spurious edges. Finally, the edges are combined with the initial segmentation result. Specifically, if an edge is detected at a pixel and the majority of surrounding pixels in a 9×9 window are also edges, then the classification of the pixel from the original segmentation step is reversed. The final segmentation result is shown in Fig. 9(f). The figures does show that more of the river and water inlet are classified as water. The indentation that appears around Area A is due to lake that is very close to the bay, and Lake Merced (Area C) is detected. The San Bruno Mountains (Area B) provided some problems to the segmentation algorithm. Other experiments suggest that other mountain ranges causes difficulties for the algorithm. Other false lakes appeared around the Oakland Airport (Area F) for unknown reasons. All in all, the incremental power and fractal dimension features provide very good separation of water and land in the San Francisco image.

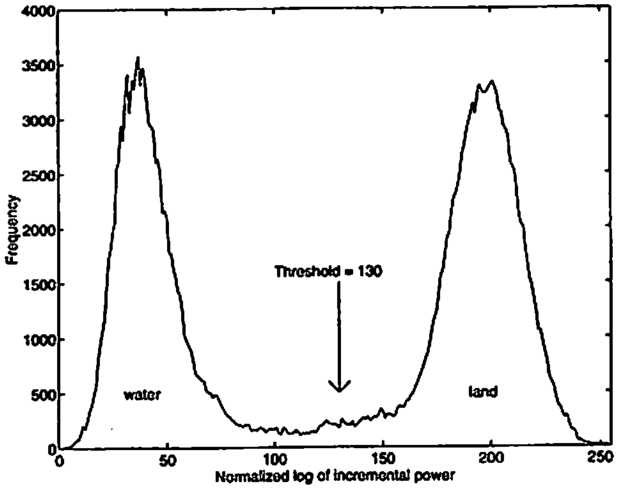
5 Conclusions and Extensions

We presented a new fractal feature extraction method by using the Haar transform. The new algorithm is fast and provides good segmentation results with only four features as demonstrated in the experiments. More segmentation problems can be performed to evaluate our new method. It would also be useful to find a way, if possible, to extend the Haar transform to higher regular filters without losing the efficient algorithmic architecture presented in Section 3. Additionally, Stewart *et al.* [35] reported that tree and grass clutters represent dfGn instead of fBm and, therefore, we should investigate the cases where the dfGn model (i.e. do not calculate the initial increments) provides better segmentation results for real textures. Other experiments which combine the fractal features with other texture features may prove to be useful. Finally, the effect of noise on the Haar fractal feature extraction method should be studied.

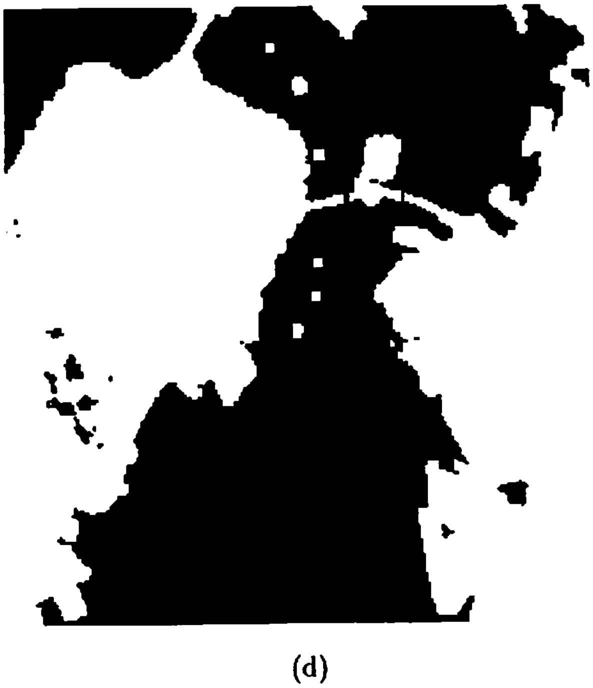
(a)



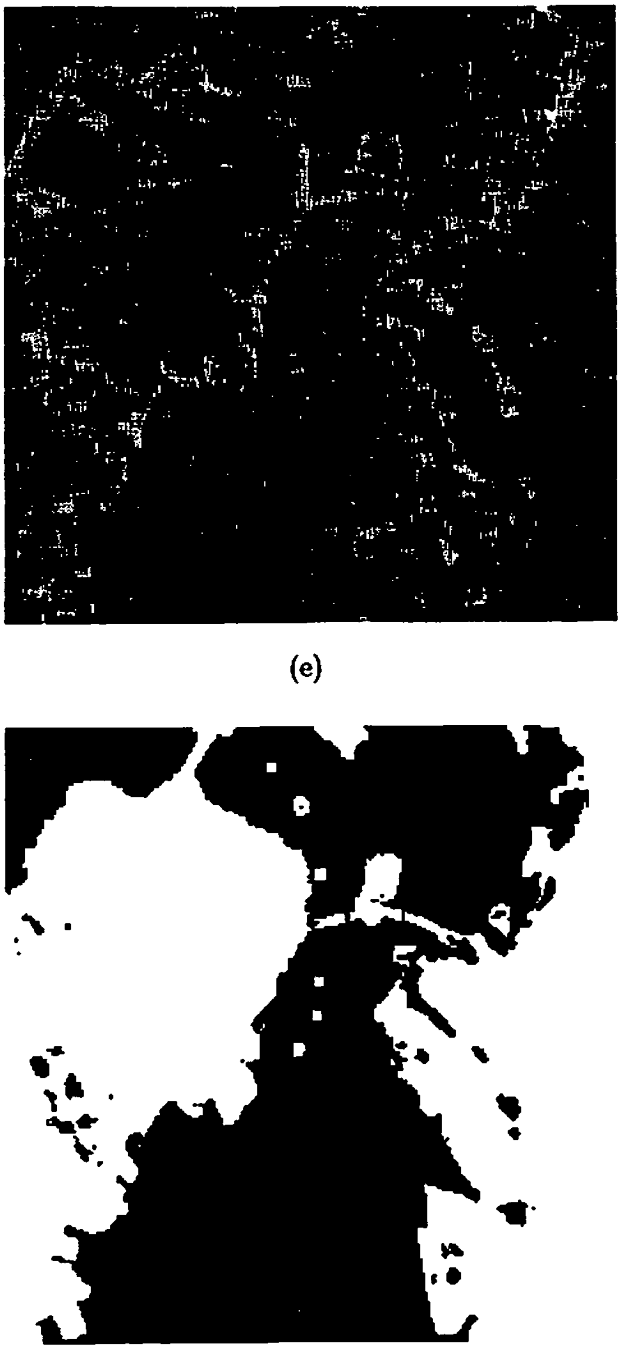
(b)


4000
3500
3000
2500
2000
1500
1000
500
0
Frequency
0
50
100
150
200
250
Normalized log of incremental power
Threshold = 130
water
land

(c)



(d)



(e)

(f)

Figure 9: (a) San Francisco image, (b) average incremental power (normalized), (c) histogram of average incremental power, (d) initial segmentation result, (e) x-directed Hurst estimate, (f) final segmentation result.

Appendix: Estimation of H

Under the assumption that (3.1) is the likelihood function for the curve of interest, the maximum likelihood estimates $\hat{\sigma}^2_{ML}$ and $\hat{H}_{ML}$ are the values of σ^2 and H which maximizes (3.1). To make the notation easier, the following terms are defined:

$$\beta = 2^{2H}, \tag{A.1}$$

$$\sigma'^2 = (2^{2-2H} - 1)\sigma^2, \tag{A.2}$$

so that the variance progression given by (2.8) can be rewritten as

$$\sigma_m^2 = \sigma'^2 \beta^m.$$

To minimize the likelihood function (3.1), the derivatives of the likelihood function with respect to β and σ'^2 is set to zero to provide the following two relations,

$$\hat{\sigma}'^2 = \frac{\sum_{m=1}^{M} N(m)\hat{\sigma}_m^2 \hat{\beta}^{-m}}{\sum_{m=1}^{M} N(m)}, \tag{A.3}$$

and

$$\frac{1}{\hat{\sigma}'^2} \sum_{m=1}^{M} mN(m)\hat{\sigma}_m^2 \hat{\beta}^{-m-1} - \hat{\beta}^{-1} \sum_{m=1}^{M} mN(m) = 0. \tag{A.4}$$

Substitution of (A.3) in (A.4) yields

$$(\sum_{m=1}^{M} N(m))(\sum_{m=1}^{M} mN(m)\hat{\sigma}_m^2 \hat{\beta}^{-m-1}) - \hat{\beta}^{-1}(\sum_{m=1}^{M} mN(m))(\sum_{m=1}^{M} N(m)\hat{\sigma}_m^2 \hat{\beta}^{-m}) = 0.$$

Then, multiplication by $\frac{\hat{\beta}^{M+1}}{(\sum_{m=1}^{M} N(m))(\sum_{m=1}^{M} mN(m))}$ provides Equation (3.2). Note that Wornell and Oppenheim have shown that when at least one value of $\hat{\sigma}_m^2$ is non zero and $M \geq 0$, then there exist a unique positive real solution to (3.2) [41]. Finally, the maximum likelihood estimate of H is determined by the inverse of (A.1), i.e. (3.3).

References

[1] R. J. Barton and H. V. Poor, "Signal detection in fractional Gaussian noise," *IEEE Trans. on Information Theory*, Vol. 34, pp. 943–959, Sept. 1988.

[2] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 99–113, Feb. 1991.

[3] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. on Signal Processing*, Vol. 39, pp. 2025–2043, Sept. 91.

[4] P. Brodatz, *Textures: A Photographic Album for Artist & Designers*, New York: Dover Publications, Inc., 1966.

[5] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. on Image Processing*, Vol. 2, pp. 429–441, Oct. 1993.

[6] B. B. Chaudhuri, N. Sarkar, and P. Kundu, "Improved fractal geometry based texture segmentation technique," *IEE Proceesings-E*, Vol. 140, pp. 233–241, Sept. 1993.

[7] C.-C. Chen, J. S. Daponte, and M. D. Fox, "Fractal feature analysis and classification in medical imaging," *IEEE Trans. on Medical Imaging*, pp. 133–142, June 1989.

[8] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 64–69, Jan. 1983.

[9] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 25–39, Jan. 1983.

[10] M. Deriche and A. H. Tewfik, "Signal modeling with filtered discrete fractional noise processes," *IEEE Trans. on Signal Processing*, Vol. 41, pp. 2839–2849, Sept. 1993.

[11] K. J. Falconer, *The Geometry of Fractal Sets*, New York: Cambridge University Press, 1985.

[12] O. D. Faugeras and W. K. Pratt, "Decorrelation methods of texture feature extraction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 2, pp. 323–332, July 1980.

[13] P. Flandrin, "On the spectrum of fractional Brownian motions," *IEEE Trans. on Information Theory*, Vol. 35, pp. 197–199, Jan. 1989.

[14] P. Flandrin, "Wavelet analysis and synthesis of fractional Brownian motion," *IEEE Trans. on Information Theory*, Vol. 38, pp. 910–917, Mar. 1992.

[15] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 721–741, Nov. 1984.

[16] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification," *IEEE Trans. on System, Man, and Cybernetic*, Vol. 8, pp. 610–621, Nov. 1973.

[17] S. Hoefer, F. Heil, M. Pandit, and R. Kumaresan, "Segmentation of textures with different roughness using the model of isotropic two-dimensional fractional Brownian motion," in *IEEE ICASSP-93*, vol. 5, pp. 53–56, Apr. 1993.

[18] J. Y. Hsiao and A. A. Sawchuk, "Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 1279–1292, Dec. 1989.

[19] L. M. Kaplan and C.-C. J. Kuo, "Fractal estimation from noisy data via discrete fractional Gaussian noise (dfGn) and the Haar basis," *IEEE Trans. on Signal Processing*, Vol. 41, pp. 3554–3562, Dec. 1993.

[20] L. M. Kaplan and C.-C. J. Kuo, "An improved method for two dimensional self-similar image synthesis," Tech. Rep. SIPI 249, Univ. of Southern California, June 1994. submitted to *IEEE Trans. on Image Processing.*

[21] R. L. Kashyap and K.-B. Eom, "Texture boundary detection based on the long correlation model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 58–67, Jan. 1989.

[22] R. L. Kashyap and P. M. Lapsa, "Synthesis and estimation of random fields using long-correlation models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 800–809, Nov. 1984.

[23] J. M. Keller, S. Chen, and R. M. Crownover, "Texture description and segmentation through fractal geometry," *Computer Vision, Graphics, and Image Processing*, Vol. 45, pp. 150–166, 1989.

[24] B. Klinkenberg and M. F. Goodchild, "The fractal properties of topography: a comparison of methods," *Earth Surface Processes and Landforms*, Vol. 17, pp. 217–234, May 1992.

[25] T. Kumar, P. Zhou, and D. A. Glaser, "Comparison of human performance with algorithms for estimating fractal dimension of fractional Brownian statistics," *J. Opt. Soc. Am.A*, Vol. 10, pp. 1136–1146, June 1993.

[26] K. I. Laws, "Texture image segmentation," Ph.D. dissertation 940, Image Processing Inst., Univ. of Southern California, Jan. 1980.

[27] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. on Communication*, Vol. 28, pp. 84–95, Jan. 1980.

[28] T. Lundahl, W. J. Ohley, S. M. Kay, and R. Siffert, "Fractional Brownian motion: A maximum likelihood estimator and its application to image texture," *IEEE Trans. on Medical Imaging*, Vol. 5, pp. 152–161, Sept. 1986.

[29] B. B. Mandelbrot, *The Fractal Geometry of Nature*, San Francisco: Freeman, 1982.

[30] B. B. Mandelbrot and J. W. V. Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM Review*, Vol. 10, pp. 422–437, Oct. 1968.

[31] G. Medioni and Y. Yasumoto, "A note on using fractal dimension for segmentation," in *IEEE Computer Vision Workshop*, pp. 25–30, 1984.

[32] H. O. Peitgen and D. Saupe, eds., *The Science of Fractal Images*, New York: Springer-Verlag, 1988.

[33] A. P. Pentland, "Fractal-based description of natural scenes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 661–674, Nov. 1984.

[34] J. Ramanathan and O. Zeitouni, "On the wavelet transform of fractional Brownian motion," *IEEE Trans. on Information Theory*, Vol. 37, pp. 1156–1158, July 1991.

[35] C. V. Stewart, B. Moghaddam, K. J. Hintz, and L. M. Novak, "Fractional Brownian motion models for synthetic aperture radar imagery scene segmentation," *Proceedings of the IEEE*, Vol. 81, pp. 1511–1522, Oct. 1993.

[36] A. H. Tewfik and M. Kim, "Correlation Structure of discrete wavelet coefficients of fractional Brownian motion," *IEEE Trans. on Information Theory*, Vol. 38, pp. 904–909, Mar. 1992.

[37] M. R. Turner, "Texture discrimination by Gabor functions," *Biological Cybernetics*, Vol. 55, pp. 71–82, 1986.

[38] M. Unser, "Sum and difference histograms for texture classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 118–125, Jan. 1986.

[39] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 717–728, July 1989.

[40] G. W. Wornell, "A Karhunen-Loève-like expansion for $1/f$ processes via wavelets," *IEEE Trans. on Information Theory*, Vol. 36, pp. 859-861, July 1990.

[41] G. W. Wornell and A. V. Oppenheim, "Estimation of fractal signals from noisy measurements using wavelets," *IEEE Trans. on Signal Processing*, Vol. 40, pp. 611–623, Mar. 1992.