A PDP Approach to Localized Fractal Dimension Computation with Segmentation Boundaries

George W. Rogers, Jeffrey L. Solka and Carey E. Priebe

SIMULATION 1995 65: 26
DOI: 10.1177/003754979506500104

TECHNICAL ARTICLE

SIMULATION 65:1, 26-36

ISSN 0037-5497/95 $3.00 + .10
Printed in the United States of America

A PDP Approach to Localized Fractal Dimension Computation with Segmentation Boundaries

George W. Rogers and Jeffrey L. Solka
Advanced Computation Technology Group
Naval Surface Warfare Center
Dahlgren, VA 22448-5000
and
Center for Computational Statistics
George Mason University
Fairfax, Virginia 22030-4444

Carey E. Priebe
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, Maryland 21218

A parallel distributed processing approach to the computation of localized fractal dimension values in imagery is presented. This approach is a further development of the covering method which requires only nearest neighbor communication. A major benefit of our approach is the ability to readily incorporate any boundary information that may be available. Many fractal textures or surfaces are fractal only in distribution. With this in mind, we show that comparison of the fractal dimension distributions via Kullback-Leibler can give an improved texture discrimination capability over comparison of computed fractal dimension. Results are presented for a set of textures.

Key Words: Parallel Distributed Processing, Fractal Dimension, Covering Method, Kullback-Leibler

Introduction

An automated texture recognition/classification capability is an important component of any artificial vision system. One approach to this segment of the vision problem which has shown promise is based on the computation of fractal dimension (*fd*) and/or related power law features [6,7,11,12].

A two-dimensional grayscale image can be thought of as a manifold embedded in a three-dimensional space. From this viewpoint, we can consider the image to have a *fd* that is somewhere between the images topological dimension of two and the dimension of the embedding space, resulting in a value between two and three. The *fd* is thus a characterization of roughness. For our purposes, the defining equation for the *fd* of an image is Richardson's law [5]. This law describes the manner in which a measured property of a fractal varies as a function of the scale of the measuring device. It is given by

$$M(\varepsilon) = K\varepsilon^{(d-D)}, \quad (1)$$

where $M(\varepsilon)$ is the measured property of the fractal at scale ε, K is a constant of proportionality, d is the topological dimension, and D is the fractal dimension. The measured property varies as a power of the scale. Hence, this is a power law relationship and we term features derived from Eq. (1) as "power law features."

If we take the logarithm of Eq. (1) we obtain a linear relationship between the log($M(\varepsilon)$) and D,

$$\log [M(\varepsilon)] = (d - D) \log (\varepsilon) + \log (K) \qquad (2)$$

which is the equation of a straight line with slope (d-D) from which D can be recovered. While exact fractals will, in principle, conform to Richardson's law, texture analysis has as its idealization the concept of "statistical fractals" [12]. For statistical fractals the straight line relationship given in Eq. (2) holds only in distribution and a least squares regression can be used to find the best linear fit to a particular set of observations $\{ \varepsilon_i, M(\varepsilon_i) \}$. The *fd* is estimated by the slope provided by this linear regression. In addition to the slope, regression also yields the y-intercept, log(K). Furthermore, for objects that do not strictly obey Eq. (2), a measure of the goodness of fit based on an F-test provides a third useful feature. The *fd*, the y-intercept, and the F-test constitute three fractal dimension based power law features that are useful features for texture discrimination [11].

Methods of estimating *fd* from Richardson's law use the measured property as a function of scale to estimate D. Two statistical fractal objects may have the same *fd* but different statistical distributions of the estimate of D. Similarly, two fractal objects may have the same *fd* but different values of K, or the same mean K values but different distributions. Finally, most objects are either non-fractal or fractal-like (obey Richardson's law) only over some limited range of scales, which makes the F-test value/distribution an important feature for texture discrimination, similar in concept to the so-called lacunarity [13]. In order to most fully characterize textures in terms of Richardson's law, one should compare the computed distributions of the power law features instead of just the mean values. With this in mind, we present an alternate method of computing *fd* derived power law features that generalizes the covering method approach of Peli [7] while requiring only nearest neighbor communication at all stages of the computation. We also present methods of estimating and comparing the resulting distributions.

We are motivated for this alternate approach by the need to perform texture computations in near real time as well as to readily be able to incorporate any *a priori* segmentation or boundary information that may be available. This has led us to develop a formulation suitable for a massively parallel implementation.

The bounding surfaces used in the covering method of computing *fd* are computed at each point (pixel) based on the characteristics of the image in some neighborhood of that point. The presence of a boundary between textures in this neighborhood can result in a greatly perturbed set of bounding surfaces, especially if there is a large difference in luminance between the two textures. This in turn can substantially perturb the computed *fd*. Segmentation schemes generally include the goal of detecting boundaries between texture types. This raises the question of how to incorporate any segmentation information that may be available into the texture feature computation so as to avoid basing features on both textures (as well as the luminance difference) near a boundary.

In this paper we present a formulation based on work by Rogers *et al.* [9] that addresses both of these concerns simultaneously. We begin by presenting this formulation in the next section. We next describe the statistical methods we use to analyze the fractal based features extracted using this new method. This is followed by a section showing some results based on this approach, with the final section devoted to some concluding remarks.

2. Formulation of the Covering Method with Boundary Incorporation

In this section we develop our implementation of the covering method. By basing the implementation on local (nearest neighbor) computations at each step, we have a method that easily incorporates any *a priori* boundary information. The method results in a distribution of values for each of the power law features.

In order to use Richardson's law to estimate the *fd* of an image we follow the covering method of Peli [7] to estimate the surface area of the image in a window about a given pixel. This method makes use of dilation and erosion operators which act recursively to bound the surface above and below at progressively larger scales. This results in a set of volume approximations at different scales which allow us to obtain estimates of the surface area as a function of scale. In order to take a potentially irregular segmentation boundary into account, we introduce the following modification. Assume that a segmentation map M(i,j) is given where M(i,j)=0 on a boundary and M(i,j)=1 for all pixels not on a boundary. Let the dilation and erosion operators for the pixel (i,j) at scale ε be denoted by $U(i,j;\varepsilon)$ and $L(i,j;\varepsilon)$, respectively. Then we introduce the new recursion relations where the zero level recursions are defined as $U(i,j;0) = L(i,j;0) = G(i,j)$ for an original image grayscale value

$$U(i,j;\varepsilon+1) = max\{U(i,j;\varepsilon)+1, [U(i+1,j;\varepsilon)M(i+1,j)+U(i,j;\varepsilon)(1-M(i+1,j))], [U(i-1,j;\varepsilon)M(i-1,j)+U(i,j;\varepsilon)(1-M(i-1,j))], [U(i,j+1;\varepsilon)M(i,j+1)+U(i,j;\varepsilon)(1-M(i,j+1))], [U(i,j-1;\varepsilon)M(i,j-1)+U(i,j;\varepsilon)(1-M(i,j-1))],\} \quad (3)$$

and

$$L(i,j;\varepsilon+1) = min\{L(i,j;\varepsilon)-1, [L(i+1,j;\varepsilon)M(i+1,j)+L(i,j;\varepsilon)(1-M(i+1,j))], [L(i-1,j;\varepsilon)M(i-1,j)+L(i,j;\varepsilon)(1-M(i-1,j))], [L(i,j+1;\varepsilon)M(i,j+1)+L(i,j;\varepsilon)(1-M(i,j+1))] [L(i,j-1;\varepsilon)M(i,j-1)+L(i,j;\varepsilon)(1-M(i,j-1))],\}, \quad (4)$$

$G(i,j)$. In Eqs. (3) and (4), the segmentation map prevents values on the boundary from being used outside the boundary. For example, if the (i+1,j) pixel is on a boundary, while the (i,j) pixel is not, the $U(i,j;\varepsilon)$ value is substituted for the $U(i+1,j,\varepsilon)$ value. Hence, the $U(i+1,j,\varepsilon)$ value will not affect the computation of $U(i,j,\varepsilon+1)$. Since the operators only involve nearest neighbors, we are guaranteed that no information will cross a continuous segmentation boundary at this stage of the computation. With $M(i,j)=1$ for all pixels, these expressions reduce to the previous versions [7,11].

Many segmentation approaches yield boundaries with an associated strength or degree of certainty. Instead of thresholding such a segmentation map to obtain a binary segmentation map we can simply use the continuous values if they are mapped to the unit interval. The continuous segmentation map can then be used for M in Eqns. (3,4) as well as in the steps to follow.

Once the upper and lower surfaces at a given scale have been computed, the bounding area is given by

$$a(i,j;\varepsilon) = [U(i,j;\varepsilon) - L(i,j;\varepsilon)]/2\varepsilon \quad (5)$$

This method can only be expected to give the correct *fd* in distribution. Hence it has been customary to average over the entire patch of texture [6] or over a window [7,11] as in

$$A(i,j;\varepsilon) = \sum_{(l,m)\in W(i,j)} c\left[\frac{U(l,m;\varepsilon) - L(l,m;\varepsilon)}{2\varepsilon}\right] \quad (6)$$

where $W(i,j)$ is some window about (i,j) and c is a constant for all scales that can be used for normalization, if desired. The *fd* can then be estimated on a pixel by pixel basis as a regression of $\log[A(\varepsilon)]$ versus $\log[\varepsilon]$ as indicated by Richardson's law Eq. (1).

The method of computing the areas embodied in Eq. (6) using a fixed window does not readily accommodate the incorporation of segmentation boundaries. If used with boundaries, it will average across the boundary with attendant undesired effects. This problem has led us to introduce a method of averaging that is based on the physics of a resistive grid.

In this method, we model each pixel as though it is connected to its four nearest neighbors by fixed conductances and to a voltage source through a resistance. This is depicted in Figure 1. In order to account for a regular or irregular segmentation boundary we set all conductances to zero that connect to a pixel on the boundary. The circuit simulation then can be viewed as adapting the effective kernel associated with each pixel so as to account for the segmentation boundaries.

The circuit equations can be easily written as

$$V_{out}(i,j) = \{V_{in}(i,j)/R + V_{out}(i-1,j)G(i-1/2,j) + V_{out}(i+1,j)G(i+1/2,j) + V_{out}(i,j-1)G(i,j-1/2) + V_{out}(i,j+1)G(i,j+1/2)\}/\{(1/R) + G(i-1/2,j) + G(i+1/2,j) + G(i,j-1/2) + G(i,j+1/2)\}. \quad (7)$$

where the indexing scheme is that of Figure 1 and V_{in} is given by

$$V_{in}(i,j) = a(i,j;\varepsilon). \quad (8)$$

This presents us with a coupled set of difference equations that model a discrete anisotropic diffusion problem. Standard relaxation methods [2] can be used to obtain a solution. Two updating methods are commonly used. The Jacobi method updates each value based on the values from the previous iteration while the Gauss-Seidel method sequentially updates each value based on the current mix of new/old values. While the Gauss-Seidel method is the method of choice for a uniprocessor computer, the Jacobi method corresponds to a massively parallel implementation where each value is to be updated synchronously. We chose to solve the equations using a successive overrelaxation (SOR) [2] version of the Jacobi method. This method simulated a massively parallel implementation. For a relaxation parameter value of (1/2), convergence to several significant figures always occurred in less than 2000 steps for an effective decay length of $L = (GR)^{1/2} = 16$. For smaller decay lengths the convergence is faster. After convergence, we set

$$A(i,j;\varepsilon) = V_{out}(i,j) \quad (9)$$

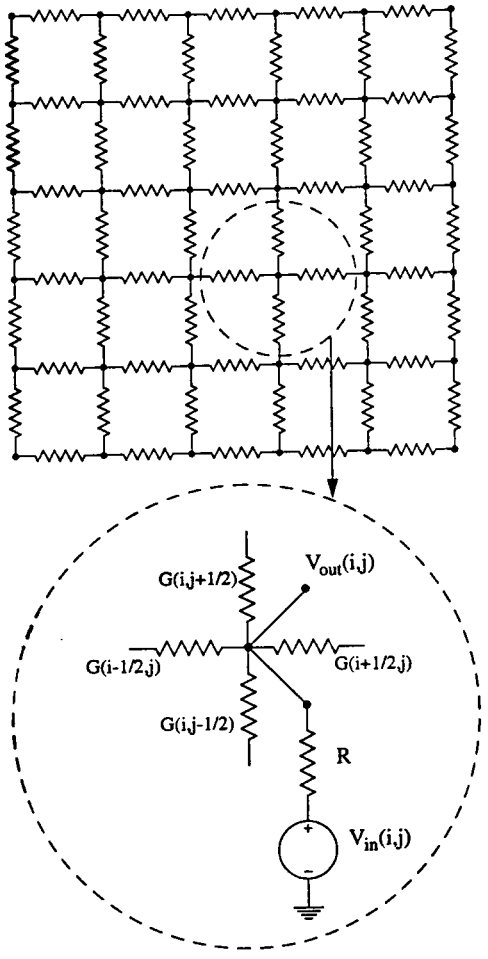


Figure 1. The resistive grid along with the detailed circuit model associated with a single pixel.

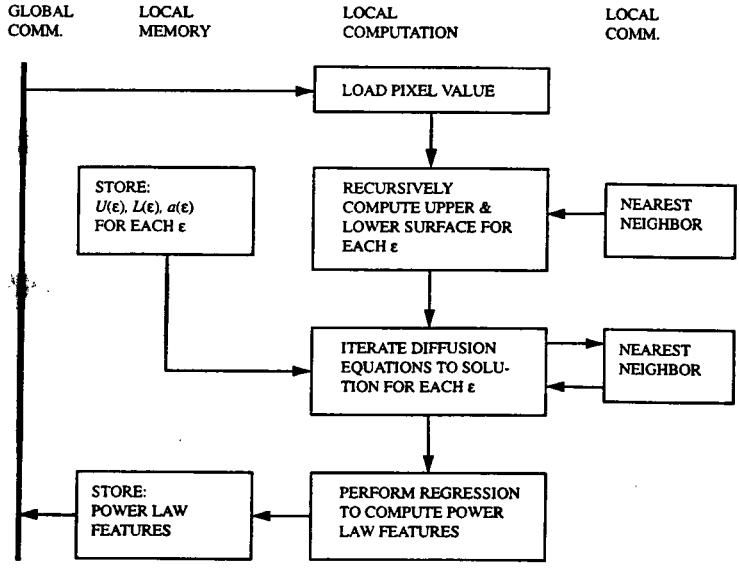


Figure 2. A block diagram of the power law feature computation at each pixel /processor.

for scale ε. Thus, the computation defined by the set of Eqs. (5,8-9) replaces the computation embodied in Eq. (6).

The final step in obtaining *fd* or the associated power law features is to perform a regression on log[A(i,j;ε)] against log[ε] to find the slope for each pixel. This is a local computation (no communication required). The quantity (2 - slope) gives the *fd* while the y-intercept and the F-test of the regression provide the two additional power law features [11].

The computation is depicted schematically in Figure 2 for a massively parallel implementation where one processor is dedicated to each pixel. Global communication is only required to distribute the image and the segmentation map, and to output the results. As can be seen from Figure 2, each pixel is acted on independently. During the course of the computation only nearest neighbor communication is required and local memory requirements are minimal. The same operation is performed in parallel on every element in the data set, making this algorithm appropriate for massively parallel architectures.

A massively parallel implementation [3] of the power law feature computation [11] has been carried out on a Connection Machine. This study yielded real time performance on that method. Based on this result, it is anticipated that the entire power law algorithm presented here could be processed in a similar parallel environment yielding results in a sub-second processing time for an image.

3. Adaptive Mixture Model Probability Density Estimation

Adaptive mixtures [8] is a recursive nonparametric algorithm for the estimation of probability density functions (pdfs). It estimates the density of the data as a mixture of normals, and chooses the number of components in a data driven manner.

The adaptive mixtures estimator is related to both the kernel estimator and to finite mixture models. For the kernel estimator, a fixed kernel is placed at each point in the data, in effect convolving the data with the kernel. One of the problems with the kernel estimator is the computational and storage requirements, particularly for large data sets. Finite mixture models consist of a fixed number of (usually Gaussian) kernels where the parameters (means, variances, etc.) are updated via the EM (for Expectation/Maximization) algorithm which is based on maximum likelihood. The major problems associated with finite mixture models are the fixed, parametric nature and the susceptibility of the EM algorithm to get trapped in local maxima.

The adaptive mixtures estimator addresses these problems by adding terms within the mixture model frame-work in a data driven manner. The ability to add terms allows the escape from local likelihood maxima. The creation of new terms occurs at a much slower rate than with the kernel estimator. This results in a robust density estimator with nonparametric capability and only moderate complexity.

Adaptive mixtures process each observation sequentially and either update the parameter values according to a recursive version of the EM algorithm, or add a new term to the model. The decision to add a new component rather than update the old ones is based on the determination that the latest observation is not well explained by the current mixture. If a point has a low likelihood for each component, that is, is an outlier of each component, then the algorithm adds a component centered at the new data point. This approach results in a robust but generally overdetermined mixture model estimate of the density, giving it the flexibility of a nonparametric approach but with the complexity of a parametric one.

4. Kullback-Leibler Information for Density Comparison

There are many approaches to determining how different two densities are. The one used in this work, the Kullback-Leibler (KL) information [4], is designed to give a measure of discriminatory power. The formula for the KL information of density f compared to density g is:

$$KL(f,g) = \int f(x)\log\left(\frac{f(x)}{g(x)}\right)dx \qquad (10)$$

The log of the likelihood ratio, log ((f (x)) / (g (x))), is the information at x for discrimination in favor of f against g. Thus, the KL information is the expected disciminatory information and hence a measure of the overall discriminatory power of f against g. This gives a single number to use in comparing the two densities and comparing different estimates of a single density.

Since the KL information is nonsymmetric (in general, KL(f,g) ≠ KL(g,f)), we actually get two numbers from the KL processing. As described above, KL(f,g) is the information in f for discriminating against g. Similarly, KL(g,f) is the information in g for discriminating against f.

The KL information gives us a tool for comparing *fd* probability density estimates corresponding to different texture patches.

5. Results

In this section we examine the application of the diffusion based averaging feature extraction method to do texture analysis. Figure 3 shows a quilt of textures [1] used for this analysis. The numbering scheme for the patches used here is the column plus four times the row index. Thus, the texture numbers run from zero to fifteen.

We begin by analyzing the discrimination information in features extracted from two textures in the quilt, textures #9 and #10. These textures are replicated, with a linear boundary, in Figure 4a.

Figures 4b and 4c demonstrate the difference in the computed bounding area (at scale $\varepsilon = 5$) with and without segmentation, respectively. The segmentation boundary, which was assigned *a priori*, is denoted by the black line in Figure 4c. Clearly, there is a significant blending of the computed area in the region about the texture interface in Figure 4b which is not experienced in Figure 4c.

In Figures 5 and 6 we present scatter plots of slope versus y-intercept. Figure 5 shows results with no segmentation. The effect of including segmentation is readily apparent in the greater feature space separation of the textures in Figure 6. It is obvious that the information available for discrimination between these two textures (or the detection of a spatial change point) is much greater for the case presented in Figure 6, which has had the segmentation information incorporated into the feature extraction algorithm.

Figure 7 shows the results of the adaptive mixtures probability density estimate procedure for *fd* features extracted from all 16 textures from Figure 3. The pdfs are based on the fd estimates for those pixels at least one decay length (16 pixels) away from the boundaries of the texture so as to minimize edge effects. Exact segmentation boundaries between textures were used. The nonnormality of these pdfs is noteworthy. In particular it should be noted that there are textures which might have a very similar mean *fd* (say, textures #3 and #4) but which have quite distinct pdfs and could therefore be discriminated if one used the densities rather than mean values.

To support the conjecture that the current approach provides results that are comparable with those previously reported in the literature (Table 3 of Sarkar & Chaudhuri [10]) we also show in Figure 7 the calculated *fd* for those textures for which Sarkar & Chaudhuri report values using five different fractal calculation methods. These reported results lie, for the most part, in the support of our probability density estimates.

We now turn our attention to comparison of the fractal dimension pdfs. The overlap between the probability density estimates f_i and f_j for each pair of textures is measured via the Kullback-Leibler information. The smaller the overlap of the density estimates the greater the KL number. Figure 8 gives a pictorial representation of KL(texture #i, texture #j). The values are zero (light gray) on the diagonal, and the darker values indicate large KL numbers corresponding to more discriminable pairs of textures.

Figure 9 considers a more detailed analysis of boundary effects on texture # 10. This figure shows both the edge effects and the boundary effects for the texture with the largest pdf differences due to these effects. For each segmentation case, the "entire" pdf shows the density based on the computed *fd* from all pixels of the texture. The "interior" pdf shows the density based on the computed *fd* from only those pixels at least one decay length (sixteen pixels) from the texture boundaries. Figure 9a shows the pdfs for the full segmentation scheme. While there are differences between the "entire" and "interior" pdfs, they are reasonably small. For the partial (grayscale) segmentation shown in Figure 9b we see that the pdfs have been degraded significantly from the full segmentation version, with the pdf built from the entire patch preserving even less of the structure than that built on just the interior observations. The no segmentation results depicted in Figure 9c show even more degradation. The large tails for these latter two estimates, especially on the left, mean the discrimination capabilities between this texture and the others will be significantly reduced. Thus, the calculation of the texture features with boundaries incorporated can be a major advantage.

6. Conclusions

The diffusion equation based covering method described in this paper constitutes a boundary gated *fd* algorithm that produces estimates that are in general agreement with previously published values [10]. The largest discrepancies occur for textures #0, #2, and #7 of the quilt. All three of these textures tend to vary significantly only at scales larger than the scales of 3, 4, and 5 pixels used in this work to estimate the *fd*. Thus, these three textures should all yield low estimates of the *fd* at these scales due to relatively large scale of variation. In the other cases (and including texture #0) the mean of the fractal dimension pdf is within the range of *fd* values reported in earlier work.

If the averaging step is left out, the density functions are much more spread out and discrimination

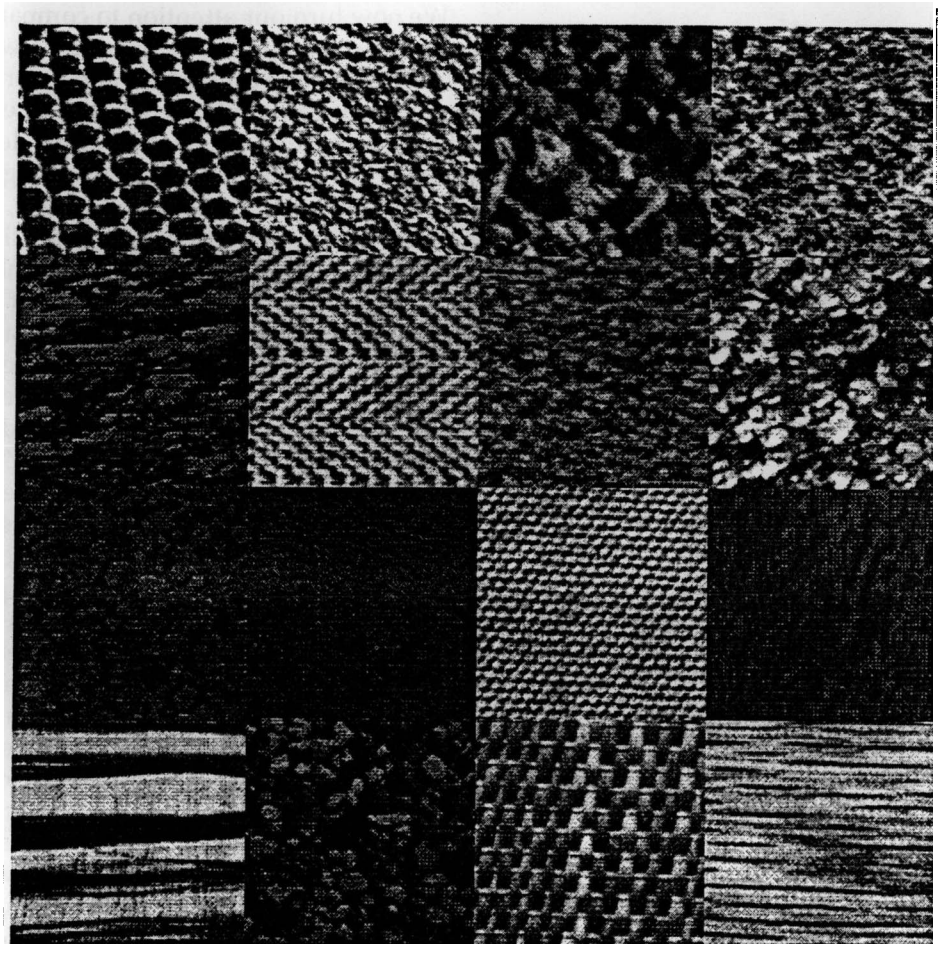


Figure 3. The quilt of textures used in this paper.

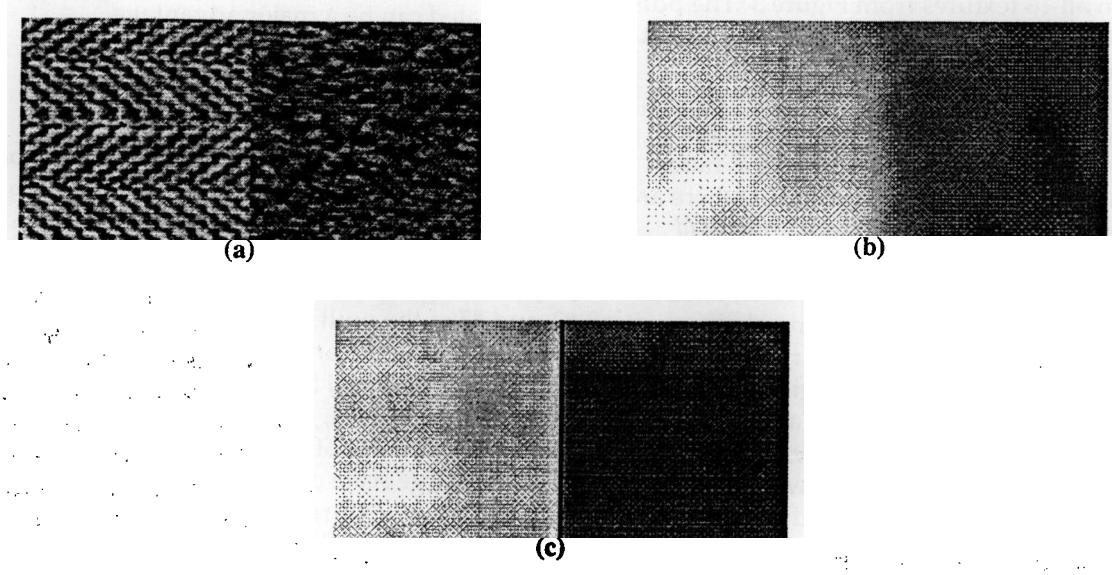


Figure 4. The two textures (#9 & #10) used in the first example are depicted in (a). The area map is shown with no boundary (b) and with boundary (c). In (c), the boundary is shown in black.

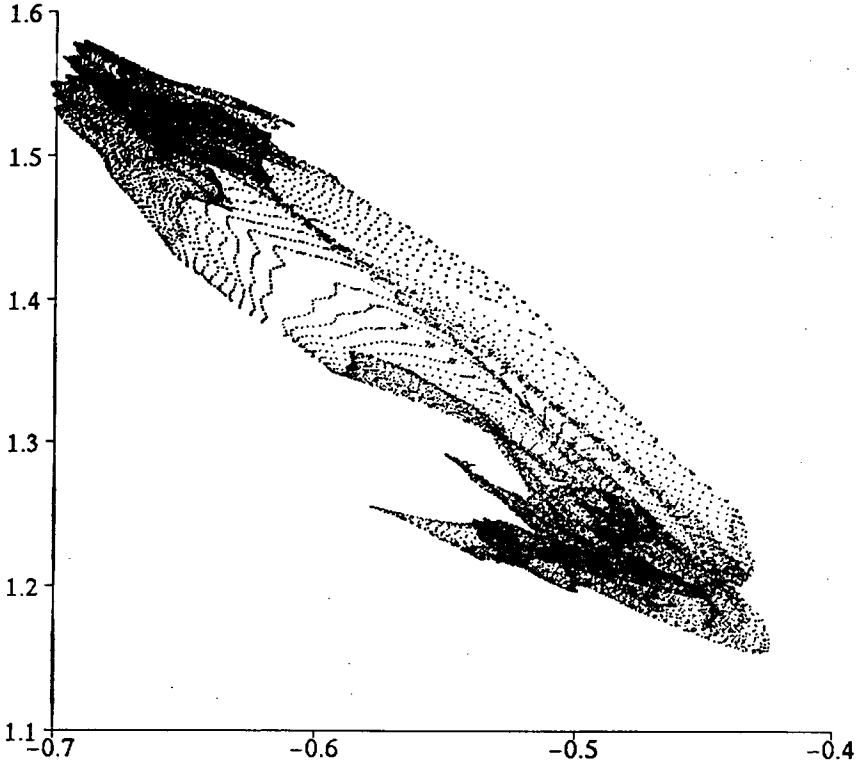


Figure 5. The y-intercept is plotted against the slope for each pixel of the two preceding texture patches with no segmentation.

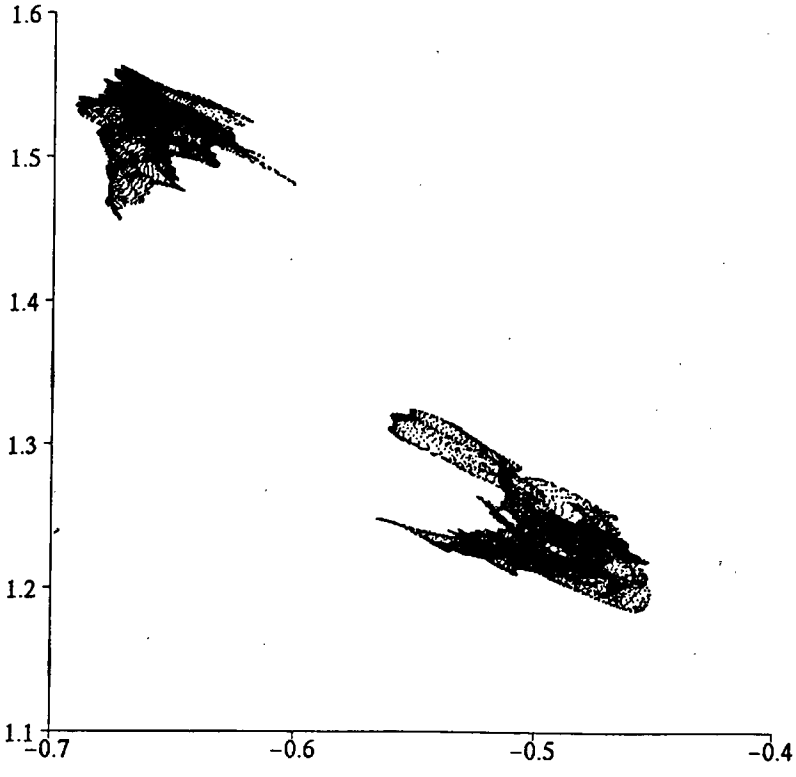


Figure 6. The y-intercept is plotted against the slope for each pixel of the two preceding texture patches with no segmentation.

between them becomes much more difficult. The greater the degree of local averaging, the greater the compactness of the resulting density estimates until, in the limit of a global average, the estimate reduces to a single value. Thus, by varying the decay length and hence the degree of local averaging, we will affect the computed densities. This in turn gives us the freedom to choose a decay length that will enhance differences in the densities corresponding to different textures.

Kullback-Leibler information numbers, computed from the texture fractal dimension pdfs, give a convenient means for quantifying the similarity or difference between pdfs. For non-normal distributions, pdf comparison is a more powerful tool for pattern recognition than simply comparing the distribution means.

The incorporation of known segmentation boundaries can lead to a dramatic reduction in the tail and variance of the *fd* distribution. This in turn can lead

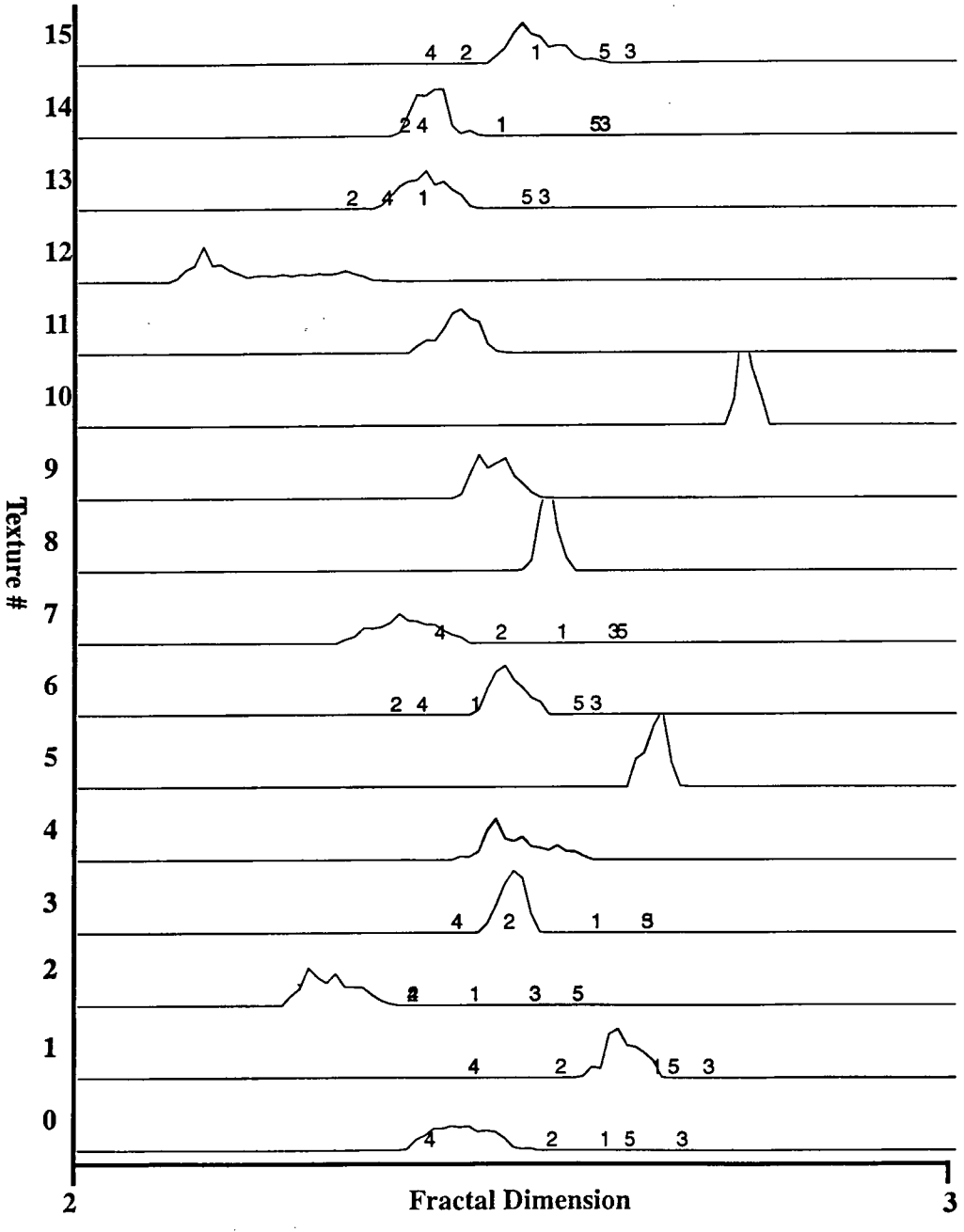


Figure 7. The probability density functions (pdfs) for the fractal dimension (fd) for all sixteen textures.

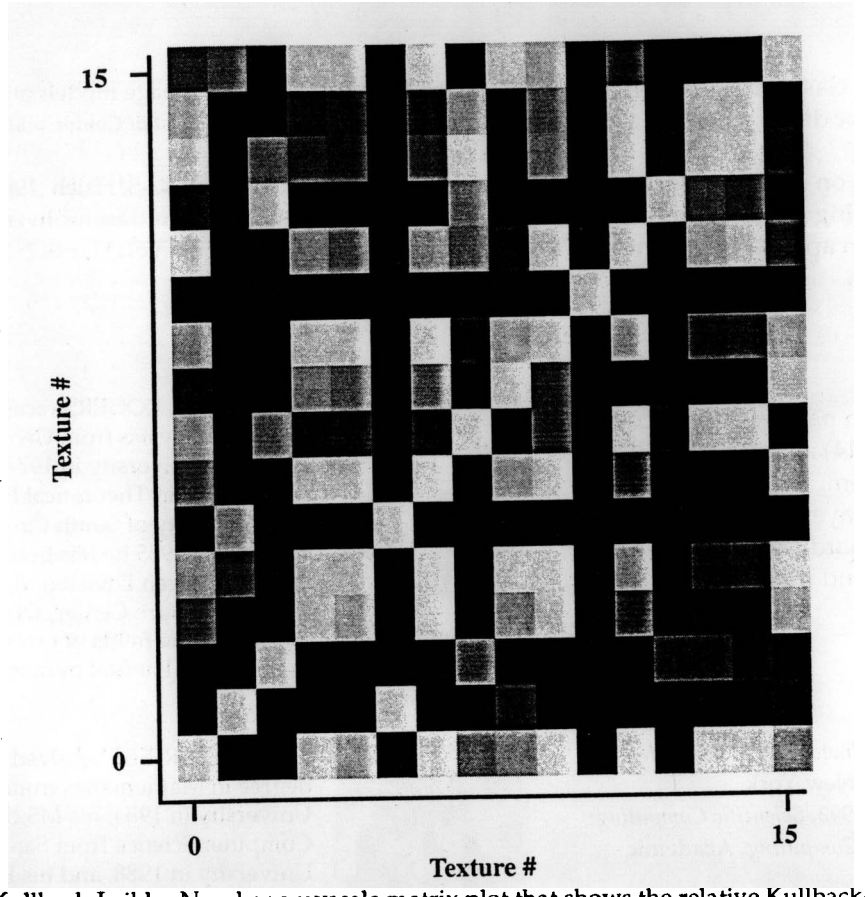


Figure 8. Kullback-Leibler Number grayscale matrix plot that shows the relative Kullback-Leibler numbers comparing different textures.

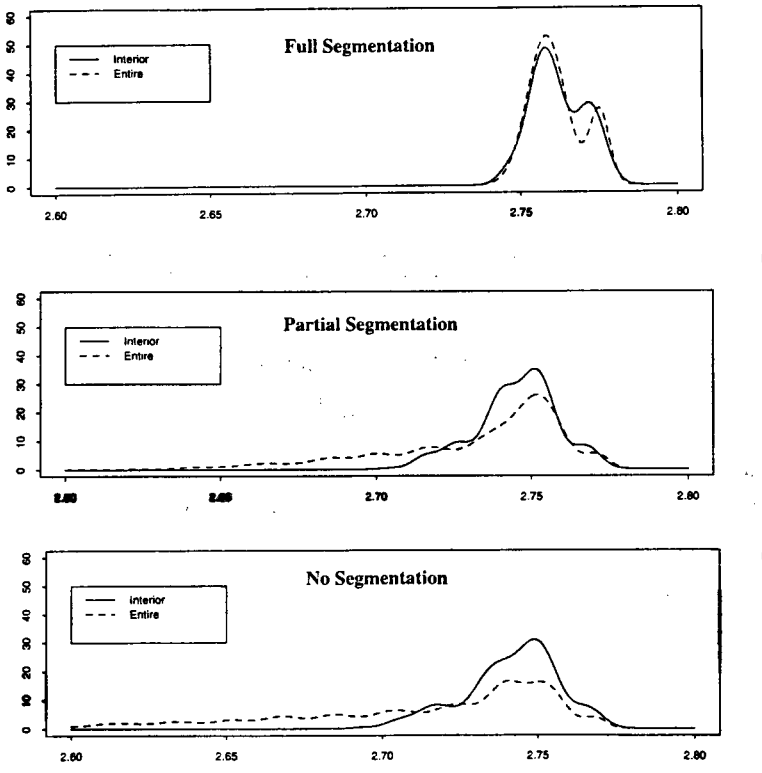


Figure 9. Edge and boundary effects for fractal dimension probability density functions (pdfs) for textures #10. The cases are (a) full segmentation (b) partial segmentation and (c) no segmentation.

to dramatic improvements in classification accuracy, whether in terms of mean *fd* or distribution via Kullback-Leibler numbers.

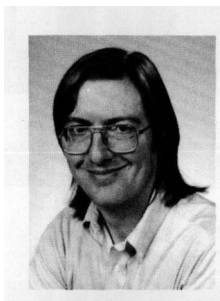
In the future we will report on ongoing work in adaptive kernel based averaging methods for the covering method as well as on applications to medical computer aided diagnosis.

Acknowledgments

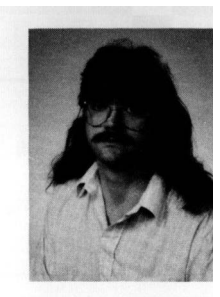
This work was supported in part by the Office of Naval Research (R&T #4424314) and the NSWCDD Independent Research program. The authors gratefully acknowledge Halford Hayes for help with the initial programming and Richard Lorey for his careful reading of the paper and many helpful suggestions.

References

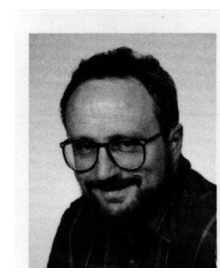
1. Brodatz, P. 1966. *Texture: A Photographic Album for Artists and Designers*. Dover, New York.
2. Golub, G. and J. M. Ortega. 1993. *Scientific Computing: An Introduction With Parallel Computing*. Academic Press, San Diego.
3. Hayes, H. I., J. L. Solka, and C. E. Priebe. 1993. "Parallel Computation of Fractal Dimension." in *Adaptive and Learning Systems II*, F. A. Sadjadi, Ed., *Proc. SPIE* 1962: 219-230.
4. Kullback, S. 1959. *Information Theory and Statistics*. Wiley, New York.
5. Mandelbrot, B. 1977. *The Fractal Geometry of Nature. W. H. Freeman and Company*, New York.
6. Peleg, S., J. Naor, R. Hartley, and D. Avnir. 1984. "Multiple Resolution Texture Analysis and Classification." *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6: 518-523.
7. Peli, T. 1990. "Multiscale fractal theory and object characterization." *Journal of the Optical Society of America A*, Vol. 7, no. 6: 1101-1112.
8. Priebe, C. E. 1994. "Adaptive Mixtures." *Journal of Am. Statist. Assoc.*, vol. 89: 796-806.
9. Rogers, G. W., C. E. Priebe, H. Hayes, and J. L. Solka. 1993. "A Parallel Distributed Processing Algorithm for Power Law Features Which Requires Only Nearest Neighbor Communication." *Proc. Fifth Workshop on Neural Networks*, SPIE vol. 2204: 269-275.
10. Sarkar, N., and B. B. Chaudhuri. 1992. "An Efficient Approach to Estimate Fractal Dimension of Textural Images." *Pattern Recognition*, vol. 25, no. 9: 1035-1041.
11. Solka, J. L., C. E. Priebe, and G. W. Rogers. 1992. "An initial assessment of discriminant surface complexity for power law features." *SIMULATION*, vol. 58, no. 5: 311-318.
12. Stein, M. C. 1987. "Fractal image models and object detection." SPIE, vol. 845 *Visual Comm. and Image Processing II*: 293-300.
13. Wu, C.-M., Y.-C. Chen, and K.-S. Hsieh. 1992. "Texture features for classification of ultrasonic liver images." *IEEE Trans. Medical Imaging*, vol. 11, no. 2: 141-152.



GEORGE W. ROGERS received his BS degree in Physics from Georgia Southern University in 1977 and the PhD degree in Theoretical Physics from the University of South Carolina in 1984. Since 1985 he has been employed at the Dahlgren Division of the Naval Surface Warfare Center, where he has worked in the fields of orbit computation, artificial neural networks, and computational statistics.



CAREY E. PRIEBE received his BS degree in Mathematics from Purdue University in 1984, his MS degree in Computer Science from San Diego State University in 1988, and his PhD in Computational Statistics from George Mason University in 1994. Since 1985 Dr. Priebe has been working in nonparametric estimation and statistical pat-tern recognition, first for the Naval Ocean Systems Center, San Diego, CA, and the Naval Surface Warfare Center, Dahlgren, VA, and currently as an Assistant Professor at The Johns Hopkins University.



JEFFREY L. SOLKA earned the BS degree in Mathematics and Chemistry from James Madison University in 1978, the MS in Mathematics from James Madison University in 1981, the MS in Physics from Virginia Polytechnic Institute and State University in 1989 and his PhD in Computational Statistics from George Mason University in 1995. Since 1984, Dr. Solka has been working in nonparametric estimation and statistical pattern recognition for the Naval Surface Warfare Center.