

Analysis of Texture Using the Fractal Model*

William Navas and Ramón Vázquez Espinosa

Abstract

Properties such as the fractal dimension (FD) can be used for feature extraction and classification of regions within an image. The FD measures the degree of roughness of a surface, so this number is used to characterize a particular region, in order to differentiate it from another. There are two basic approaches discussed in the literature to measure FD: the blanket method, and the box counting method. Both attempt to measure FD by estimating the change in surface area with respect to the change in resolution. We tested both methods but box counting resulted computationally faster and gave better results. Differential Box Counting (DBC) was used to segment a collage containing three textures. The FD is independent of directionality and brightness so five features were used derived from the original image to account for directionality and gray level biases. FD can not be measured on a point, so we use a window that slides across the image giving values of FD to the pixel on the center of the window. Windowing blurs the boundaries of adjacent classes, so an edge-preserving, feature-smoothing algorithm is used to improve classification within segments and to make the boundaries sharper. Segmentation using DBC was 90.8910 accurate.

1. Introduction

The idea behind fractal geometry is that a fractal surface or boundary, when examined in finer detail repeats itself. In other words, a part of it resembles the whole but at a different scale. It can be observed that most of the shapes generated by natural phenomena have this property of self-similarity, lead Mandelbrot [1] and others ([5], [6], [7]) to think that they can be modeled better using fractal geometry rather than Euclidean. Clouds, rivers, coastlines and trees are examples naturally occurring fractals. Our analysis is based on the assumption that most textures exhibit self similarity. Thus, the measurement of fractal dimension (FD) can be used as a discriminator, given that the textures have different degrees of roughness,

2. Fractal Dimension

Traditionally we have regarded points, lines, shapes and objects as having 0, 1, 2, 3, . . . dimensions. Hausdorff and Besicovitch not only found that there could be fractional dimensions, but they redefined the whole concept. The Hausdorff/Besicovitch dimension is defined as:

$$D = \frac{\log(N_r)}{\log(\frac{1}{r})} \quad (1)$$

where N_r is the number of copies of the seed and r is the size of the copy relative to the seed also known as the scaling factor. This means a fractal surface can have a dimension between 2 and 3, and a fractal curve can have a dimension between 1 and 2. Fractal dimension of surfaces can be estimated using the a variety of methods, some of which are the Blanket Method and the Differential Box Counting Method.

The methods described by Mandelbrot [1] for measuring the length of a fractal curve can be expanded to 2 dimensions case and applied to images. One approach is to estimate the surface area of the texture at varying resolutions. As the resolution, ϵ , increases, surface area, $A(\epsilon)$, also increases. Plotting, $A(\epsilon)$ vs. ϵ on a log-log scale yields a line whose slope is the gradient g , of the fractal dimension. The gradient is defined as:

$$g = D - D_T \quad (2)$$

where D is the fractal dimension, D_T is the topological dimension and g is the gradient. If the object is a curve $D_T = 1$, for a surface $D_T = 2$.

Another approach to measure FD is to quantify how much area a curve occupies; or, for the two dimensional case, how much space a surface occupies at a given resolution.

* This work was partially supported by NASA Grant NCCW-0088.

This can be done effectively by the box counting method in which the size of the box relates to a given scale, r . And the number of boxes, N_r , that contain points of the surface, is related to the area. The slope of the curve generated by plotting $\log(N_r)$ vs. $\log(1/r)$ yields D .

3. The Blanket Method

This method is a two-dimensional derivation of the Sausage Method explained in [1], and it is discussed by Peleg in [2]. We will use this approach to classify textures 3 textures taken from the Brodatz Album [3] and one synthetic and compare them with other images. By measuring the area, $A(\epsilon)$ at decreasing resolution (increasing ϵ) and estimating its derivative on a log-log scale,

$$g = \frac{d(\log(A(\epsilon)))}{d\epsilon \log(\epsilon)} \quad (3)$$

we obtain a set of features which contains information about the surface's self-similarity, and its roughness. Since all surfaces are not fractals, the plot of eq.3 does not yield a straight line. We will take advantage of this fact and use the whole curve as a signature for identifying a particular texture. Each point generated by equation 3 is a feature of the texture that is fed to a minimum distance classifier for identification.

3.1. Feature Extraction

Blankets above and below the surface are defined by:

$$u_e(i, j) = \max(u_{e-1}(i, j) + 1, \max(u_{e-1}(m, n)_{(m,n)-(i,j)=1})) \quad (4)$$

$$l_e(i, j) = \min(u_{e-1}(i, j) + 1, \min(u_{e-1}(m, n)_{(m,n)-(i,j)=1}))$$

The point is to calculate the volume between the two surfaces. One approach is to integrate the difference between u_e and l_e , then divide by the thickness which is 2ϵ . But the blankets are not symmetric and this information is lost at integration. A better approach is to measure the upper and lower volumes independently; from the surface to the blanket. If ϵ increases by one, then the area is the difference between the current volume and the one calculated in the previous iteration. The signatures are the slopes of the best fitting line among every three points of the area. For N area measurements the signatures will have $N-2$ points. Comparisons were done calculating the distances squared between all textures taking into account upper and lower signatures for each texture.

$$D(i, j) = \sum_{\epsilon} \left[\left(S_i^+(\epsilon) - S_j^+(\epsilon) \right)^2 + \left(S_i^-(\epsilon) - S_j^-(\epsilon) \right)^2 \right] \quad (5)$$

3.2. Blanket Method Results

Table 1 shows results of the minimum distance classifier. Minimum distances are in bold. It is evident that the two images having the same texture content are the ones that show the smallest difference between their features. For these textures, the fractal signature could discriminate to which class a given texture pertains. Table 1 shows differences between the features of training (rows), and the features of the textures to be classified (columns).

	hmpaper 1	pigsk 1	weave 1	Synth 1
hmpaper 2	0.0414	0.8504	0.4571	0.4616
pigsk 2	0.5997	0.0589	0.6037	0.3991
weave 2	0.4573	0.7324	0.0255	0.7775
Synth 2	0.4383	0.4027	0.7421	0.0522

Table 1. Distances between all textures.

4. Differential Box Counting

Differential Box counting is an approximation of the Blanket Method. Referring to Eq. 1, N_r , determined as follows: An image of size $M \times M$ is scaled down to a size $S \times S$ where $M/2 > S > 1$ and s is an integer, and r is S/M . The (x,y) space is partitioned into an (i,j) of size $S \times S$. On each grid (i,j) there is a column of S boxes. We let the maximum and minimum gray levels of the $(i,j)^{th}$ grid fall on the l^{th} and k^{th} box respectively so that:

$$n_r(i, j) = l - k + 1 \quad (6)$$

$$N_r = \sum_{(i,j)} n_r(i, j) \quad (7)$$

N_r is calculated over 3 values of r . D is obtained substituting on eq. 1 and doing a linear regression.

4.1. Segmentation

Experiments show that the Blanket method is less discriminating than DBC so we used DBC for segmentation as described in Chaudhuri[4]. Since FD does not account for directionality nor absolute gray level, four images were generated from the original. They are called: High Gray, Low Gray, Horizontally and Vertically Smoothed. These preprocessed images are called $1_2, 1_3, 1_4,$ and $1_5,$ respectively. These images are derived from 1_1 which is the original image. The features correspond to the FD of I_i , where $i = \{1,2,3,4,5\}$.

4.2. Features

The first feature is the fractal dimension of the original image calculated on overlapping windows of size $(2w+1) \times (2w+1)$. Since the resulting FD will be between 2 and 3, it is normalized by subtracting 2 so that values for feature 1 will be: $f(i,j) = FD - 2$.

For features 2 and 3 the FD of I_1 and I_2 are taken. These are defined by the following rules:

$$I_2(i, j) = \begin{cases} I_1(i, j) - L_1, & \forall I_1(i, j) > L_1 \\ 0, & otherwise \end{cases} \quad (8)$$

$$I_3(i, j) = \begin{cases} 255 - L_2, & \forall I_1(i, j) > 255 - L_1 \\ I_1(i, j), & otherwise \end{cases} \quad (9)$$

For features 4 and 5 the FD of I_4 and I_5 are taken. These are:

$$I_4(i, j) = \frac{1}{2w+1} \sum_{k=-w}^w I_1(i, j+k) \quad (10)$$

$$I_5(i, j) = \frac{1}{2w+1} \sum_{k=-w}^w I_1(i+k, j)$$

4.3. Methodology

A sliding window of 17×17 pixels scanned the five features of the image in steps of three pixels. The original image was $254 \times 254 \times 256$. $W=17$ was convenient because it can be subdivided into quadrants of 8×8 neighboring a center pixel (i,j) . N_r , can be easily calculated for $r = \{1/2, 1/4, 1/8\}$. The resulting measurement of D is assigned to the point (i,j) at the center of the window. The training samples taken from the original image were analyzed by DBC and D was calculated for all them.

Once the FD is calculated a feature smoothing algorithm is applied. This reduces the misclassification that occurs on the boundaries between one texture and another caused by sliding window over two different textures. The filter used works on the spatial domain. It uses a window divided into 4 quadrants. The mean of the quadrant that has the smallest variance is the assigned to the pixel at the center of the window. The technique is known as Edge Preserving Noise Smoothing Quadrant (EPNSQ).

Finally the minimum distance classifier compared distances squared between features of the original image and the mean of the features of the training set samples. Figure 1 shows a block diagram of the whole process.

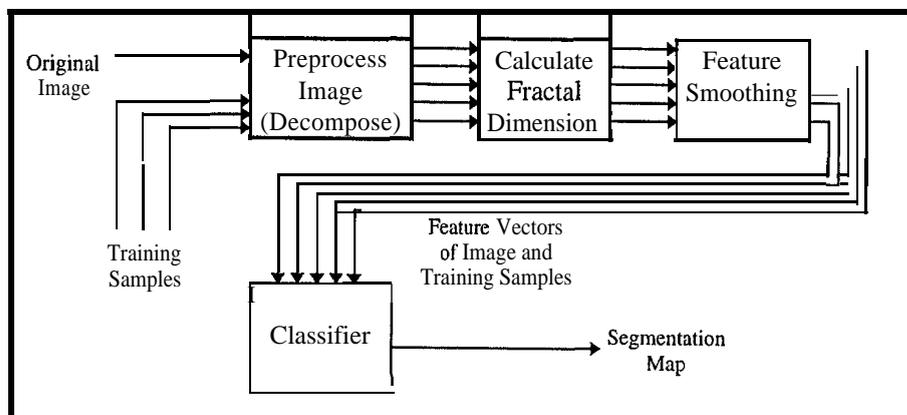


Figure 1. Block diagram of the segmentation process.

4.4. Segmentation Results

The following results were obtained using the method specified on section 4.3. An accuracy of 90.8% on the 3-texture mosaic was achieved. Using only the first feature, the percentage of area correctly classified was 80.3%. To measure accuracy the segmentation map was compared against the actual class map which was crafted by manually specifying the texture class on each sample point.

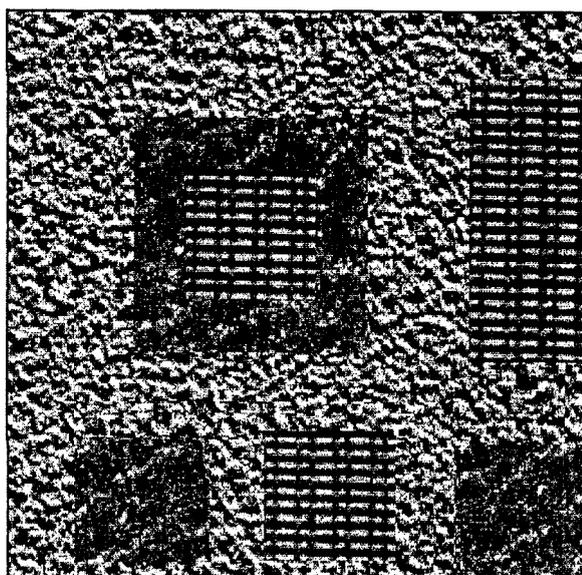


Figure 2. Original Image



Figure 3. Segmented Image.

The second image used was an aerial photograph, taking samples of the water, vegetation, and agricultural drainage trenches. Psycovisual inspection reveals high correlation between the percieved textures on figure 4 and the segmentation map on figure 5.

5. Conclusion

Through these experiments we could compare the banket method and DBC methods in terms of quality and speed. DBC is faster but segmentation was fair no post-filtering of the features had been applied. The

EPNSQ filter greatly improves classification as it denoises features within a segment while preserving its boundary.

The fractal dimension alone is not enough to characterize texture, On the Blanket Method, upper and lower blankets are used independently as features to account for asymmetry of FD measured on the top side vs. the FD measured on the under side. For DBC, the original image was decomposed into low gray, high gray, vertically smoothed and horizontal y smoothed, so that FD could account for gray-level biases and directionality. This does not assure that two different textures will have the different, but is better than having the original image only. More features can be used at the expense of a linear increase in processing time.

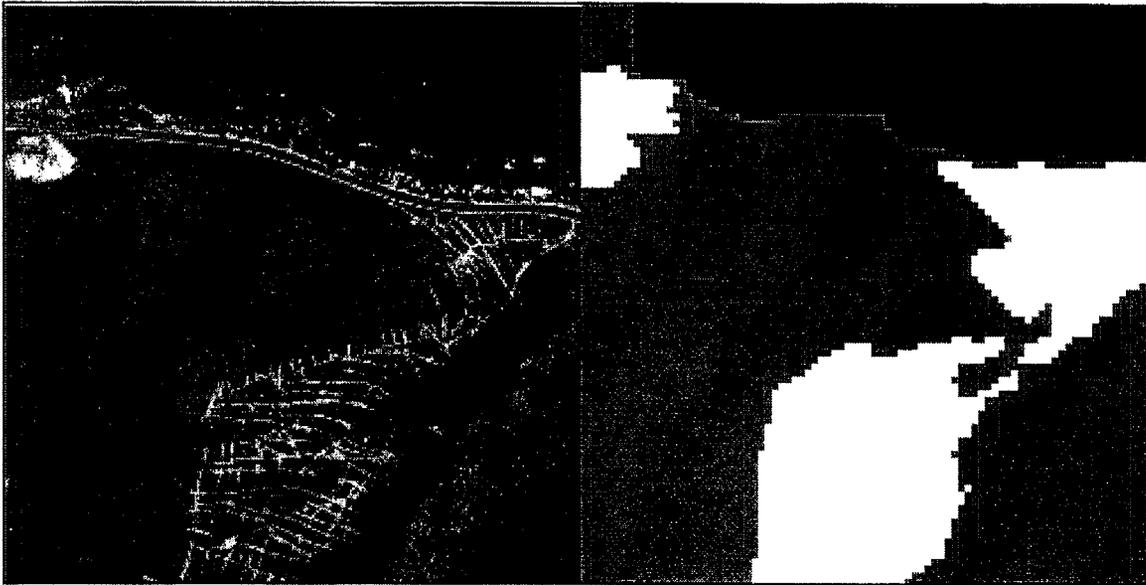


Figure 4. Original Image

Figure 5. Segmented Image

REFERENCES

- [1] B. Mandelbrot - The Fractal Geometry of Nature, W. H. Freeman, NY, 1975
- [2] S. Peleg, J. Naor, R. Hartley, D. Avnir - Multiple Resolution Texture Analysis and Classification, IEEE Transactions, July 1984, PAMI-6 (4), pp. 518-523
- [3] P. Brodatz - Textures, A Photographic Album for artists and Designers, NY, Dover, 1966
- [4] B. B. Chaudhuri, N. Sarkar, Texture Segmentation Using the Fractal Dimension, IEEE Transactions, PAMI- 17 (1), pp72-77.
- [5] R. F. Voss - Random Forgeries, Fundamental Algorithms for Computer Graphics, springer-Verlag, Berlin, 1985) pp.805-835
- [6] Lewis F. Jardine, Ian R. Whitworth - Fractal Based Synthesis of Visual Texture, IEEE Colloquium on 'Application of Fractal Techniques in Image Processing, October 1991,
- [7] L. M. Linnet, S. J. Clarke, C. Graham, D. N. Langhorne - Remote Sensing of the Seabed Using Fractal Techniques, Electronics and Communications Engineering Journal, October 1991.