



Implementación de algoritmos fractales para la clasificación no supervisada de imágenes raster

Implementation of Fractal Algorithms to Classification Unsupervised Raster Images

David A. Suárez Arévalo¹

Para citar este artículo: Suárez, D. (2016). Implementación de algoritmos fractales para la clasificación no supervisada de imágenes raster. *Revista de Topografía Azimut*, (7), 67-74.

Recibido: 12-agosto-2016 / Aceptado: 20-octubre-2016

Resumen

El presente artículo abarca la utilización de la geometría fractal y la estimación de la dimensión fractal como parámetro para clasificar una imagen digital por medio del método no supervisado. Existen múltiples métodos para estimar la dimensión fractal como el Método Fraccional Browniano (fBm, por sus siglas en inglés), los métodos de conteo de cajas y los métodos de determinación de áreas. A la hora de realizar una clasificación de imágenes digitales se utiliza como parámetro de clasificación la textura presente en las imágenes; esta textura es usualmente calculada por el método GLCM (Matriz de Co-ocurrencias de niveles de gris), la dimensión fractal es una alternativa a la hora de extraer esta característica de las imágenes. Se tomó el método de conteo de cajas diferencial (DBC, por sus siglas en inglés) mejorado como base en la determinación de la dimensión fractal y posterior clasificación de las imágenes raster. Las imágenes raster escogidas son imágenes de radar SAR debido a que es en este tipo de imágenes donde se obtiene mayor provecho del análisis de texturas, fuente de información para realizar la clasificación no supervisada, esto

debido a la característica textural que puede ser capturada por medio de este tipo de sensores.

Palabras clave: clasificación, dimensión, fractal, implementación, remotos, SAR, sensores.

Abstract

This paper approach in management of fractal geometry and fractal dimension as parameter to unsupervised classification method in raster images. There are many methods to estimate fractal dimension as Motion Brownian method (fbm), boxes counting methods and Measure Areas. In classification of digital images is used as parameter measure's texture of classification, this parameter is usually calculated with GLCM Method (Gray Level Cooccurrence Matrix), then the fractal dimension is a choice to determine textures images. I was chosen DBC (Differential Box Counting) Improved as basis at estimate of fractal dimension and classification of raster images. SAR images were chosed because these images capture textural features of natural objects, then these are an source of important information.

Keywords: Palabras pendientes.

1 Univesidad Distrital Francisco José de Caldas. Correo electrónico; dasuareza@correo.udistrital.edu.co

INTRODUCCIÓN

Se han propuesto diversos algoritmos para el análisis de patrones, clasificación de imágenes, identificación de objetos y zonas que presentan características texturales similares (Jensen, *et al.*, 1996). En el campo del análisis y extracción de texturas en imágenes digitales se maneja comúnmente el algoritmo GLCM, del cual se obtiene buenos resultados (Presuti, 2004); en los últimos años ha sido utilizado el concepto de geometría fractal para caracterizar las formas de los objetos naturales, desde finales de la década de 1980 y comienzos de la década de 1990 fue introducido el concepto como herramienta para determinar características texturales en una imagen (Clarke, 1986; Pentland, 1984; Sarkar, 1992); desde entonces y hasta la actualidad se han propuesto variedad de algoritmos que utilizan como base la dimensión fractal en el análisis de imágenes digitales (Yu, *et al.*, 2014; Theera, 2002) ya que se ha mostrado que los objetos del mundo real pueden tener estructura fractal, al igual que las imágenes captadas de dichos objetos (Matter, *et al.*, 2009).

Para hablar de la geometría fraccionada o fractal debemos definir la geometría euclidiana, la primera en simular las formas de la naturaleza. Fue el matemático B. Mandelbrot (1967, 1977a, 1977b), quien propuso que algunos elementos y objetos naturales se comportan con características fractales; Mandelbrot definió un objeto fractal como aquel que está compuesto por muchos elementos, cada uno de igual forma al elemento global pero con una reducción en su escala, a su vez, dichos elementos se componen de múltiples elementos de la misma forma pero de escala más reducida (Rubiano, 2008). Otra característica presente en los objetos fractales es que la dimensión topológica de un fractal debe ser mayor a su dimensión

de Hausdorff-Besicovich (Matter, 2009). A partir de estos conceptos se han referenciado múltiples aplicaciones para investigaciones de procesamiento de imágenes SAR (Matter, 2009; Pant, 2008, 2010) y de otras imágenes utilizadas en diversas disciplinas que dan referencia de la utilización de la teoría fractal (Clarke, 1986; Yu, *et al.*, 2014; Lopes, *et al.*, 2009; Pentland, 1984; Sarkar, 1995; Theera, 2002) como herramienta de procesamiento de imágenes.

Las imágenes de Radar SAR, las cuales se almacenan en formato digital con una resolución radiométrica de 8 bits, generalmente, son propicias a la hora de ser clasificadas utilizando como uno de sus parámetros su rugosidad por medio de la geometría fractal y la estimación de DF.

En el presente artículo se presenta una implementación de dos algoritmos basados en la geometría fractal y el algoritmo GLCM utilizados para la extracción de texturas en una imagen de satélite SAR, realizando una comparación de dichos algoritmos a través de la clasificación no supervisada de las imágenes de textura.

DIMENSIÓN FRACTAL

La dimensión fractal está definida por la siguiente ecuación:

$$1 = N_r r^D \Rightarrow D = \frac{\ln(N_r)}{\ln(1/r)} \quad (1)$$

Donde D representa la dimensión fractal (DF), o dimensión de auto-similaridad, de un objeto, N_r representa el número de copias semejantes no superpuestas del objeto reducido por factor de escala $(1/r)$.

Para imágenes de sensores remotos no es fácil realizar el cálculo de DF partiendo de la ecuación (1), en este caso es necesario otro tipo de enfoque de estimación de DF; para calcular DF en una imagen digital se tendrá en cuenta el nivel digital (ND) o nivel de escala de grises presente en la imagen, el tamaño de la imagen y su resolución espacial, luego la imagen se considerara como una superficie plana cuyas coordenadas x (Columnas), y (Filas) corresponden a la ubicación del píxel y una tercera coordenada z será el ND representando así el espacio tridimensional de la imagen.

Métodos para la estimación de DF en imágenes digitales

Variedad de investigadores han propuesto diversos enfoques para la estimación de DF hacia imágenes digitales en escala de grises; una clasificación de los métodos es propuesta por Lopes y Betrouni (2009), realizan una agrupación de tres métodos que son: los métodos de conteo de cajas, el método de movimiento fraccional Browniano (fBm) y los métodos de medición de área; de estos enfoques, el método DBC (perteneciente al grupo de métodos de conteo de cajas) fue escogido en este trabajo por ser uno de los más precisos, más utilizados y menos costosos en cuanto a su implementación computacional (Matter, 2009; Sun, *et al.* 2006). Los pasos a seguir en la estimación de la DF para los dos métodos utilizados son los siguientes:

- Medir el objeto en consideración utilizando varias longitudes de medida.
- Diagramar el gráfico del logaritmo de las cantidades medidas contra el logaritmo de las longitudes de tamaño de medida, luego ajustar los valores mediante una regresión lineal utilizando mínimos cuadrados; "El diagrama log-log es conocido comúnmente como el diagrama de Richardson" (Sun, *et al.* 2006).
- La DF será la pendiente de la recta resultante de la regresión lineal.

Método de conteo de cajas diferencial

El conteo de caja diferencial DBC es un enfoque poco trabajado en el campo de las geociencias, a pesar de que de este se pueden obtener buenos resultados como lo aseguran Matter y Tso (2009) en sus estudios. El método DBC fue propuesto por N. Sarkar y B. Chaudhuri (1992) para calcular la DF en imágenes digitales.

En el DBC el valor de Nr en la Ecuación (2), es contado de la siguiente manera: se tiene una imagen de tamaño $M \times M$ píxeles dividida en una grilla con celdas de longitud $s \times s$ donde $M/2 \geq s > 1$ y s es un entero, entonces se puede deducir que $r = s/M$. En cada celda $s \times s$ se sobrepone una columna con cajas de tamaño $s \times s \times s'$ como se aprecia en la Figura 1.

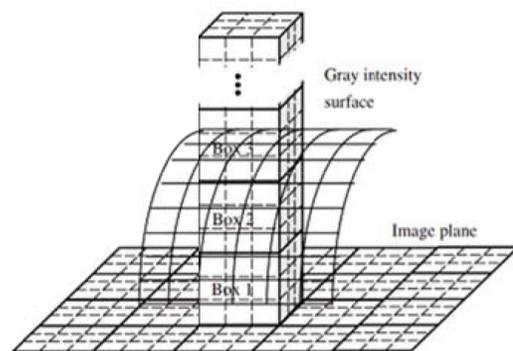


Figura 1. Vista en 3D del método DBC

Si G es el número total de niveles de gris dentro de la imagen (por ejemplo 256), s' es calculado por, $\lceil G/s' \rceil = \lfloor M/s \rfloor$, dejando el mínimo y máximo nivel de gris de la celda (i, j) de la imagen ubicada en la caja número k y l , respectivamente; entonces se tiene que $n r = l - k + 1$ es la contribución de Nr en la (i, j) -ésima celda, como se aprecia en la Figura 1.

Tomando las contribuciones de todas las celdas dentro de la imagen a partir de la Ecuación (2).

$$N_r = \sum_{i,j} n_r(i,j) \quad (2)$$

Para varios valores de r , que es igual a s o tamaños de medida, es calculada la Ecuación (1); luego se calcula D por medio del ajuste lineal por mínimos cuadrados entre el logaritmo de N_r y el logaritmo de $1/r$.

En los últimos años investigadores han propuesto mejoras al algoritmo DBC, estas mejoras han surgido, en su mayoría, debido a la deficiencia de este algoritmo a la hora de realizar el conteo de cajas real para que cubra la totalidad de la imagen. Para el presente trabajo se tuvo en cuenta el método DBC mejorado, propuesto por Liu *et al.*, el cual modifica la expresión $nr = l - k + 1$ por la ecuación (3), donde la expresión "ceil" denota la parte entera superior del valor resultante dentro de los paréntesis, I_{max} e I_{min} representan la intensidad de escala de gris máximo y mínimo respectivamente.

$$nr(i,j) = \begin{cases} \text{ceil}((I_{max} - I_{min} + 1)/s^r), & \text{si } I_{max} \neq I_{min} \\ 1 & \text{si } I_{max} = I_{min} \end{cases} \quad (3)$$

Para mejorar el conteo de caja para el plano x , y Liu *et al.* proponen aplicar un deslizamiento de los bloques de cajas en el plano a determinados píxeles δ . Primero realizan el conteo de cajas para los bloques sin realizar el desplazamiento de δ píxeles y, posteriormente, hacen el conteo de cajas aplicando el desplazamiento de δ píxeles; luego comparan los dos resultados obtenidos y se escoge el máximo valor, como el valor nr , justo como se muestra en la ecuación (4)

$$n_r(i,j) = \max(n_{r_old}, n_{r_new}) \quad (4)$$

Métodos para la estimación de la dimensión multifractal en imágenes digitales

Al igual que con la estimación de la dimensión fractal, R. Lopes y N. Betrouni (2009) proponen una clasificación de métodos multifractales; lo clasifican en métodos por conteo de cajas y métodos basado en ondas. Para el presente trabajo se utilizó el método de conteo de cajas, el cual sigue el mismo principio descrito para la estimación de DF del método de conteo de cajas.

Método multifractal por conteo de cajas:

La estimación de la dimensión fractal de una imagen toma en cuenta las cajas no vacías para cubrir la intensidad de superficie de la imagen, esto sin tener en cuenta el número de puntos en cada caja, describiendo así la geometría de la estructura pero ignorando la distribución de densidad de puntos sobre la estructura misma (Sarkar, 1995). Para superar esta carencia en la medición, se tiene en cuenta el término denominado dimensión fractal generalizada $D(q)$, el cual toma en cuenta el número de puntos en una caja, definida por la

$$D_q = \frac{1}{q-1} \lim_{r \rightarrow 0} \frac{\log \sum_i^{N_r} P_i^q}{\log r}; (q \neq 1) \quad (5)$$

ecuación (5) y (6)

$$D_q = \lim_{r \rightarrow 0} \frac{\sum_i^{N_r} P_i \log P_i}{\log r}; (q = 1) \quad (6)$$

Donde $-\infty \leq q \leq \infty$ representa los momentos para calcular cada orden multifractal y $P_i = U(-Bi)/U(A)$ es la medida normalizada (porcentaje de puntos) de la i -ésima caja, siendo $U(Bi)$ el número de puntos dentro de la caja y $U(A)$ el número total de puntos en el conjunto A . Brandt Tso (2009) realizó un estimado del valor de $D(q)$ a partir de determinar a P_i como la Ecuación (7).

$$P_i = [n_r(i, j)/N_r] \quad (7)$$

Al variar los valores de q en la ecuación (5), se obtendrán diversos valores de $D(q)$, es decir, diversos valores de dimensión fractal, razón por la cual se conoce como dimensión multifractal.

Técnica de movimiento de ventana

Para realizar un análisis de texturas se utiliza una técnica de movimiento de ventana (una subimagen), de forma similar a la técnica realizada en los filtros estadísticos (Canada Center for Remote Sensing); al realizar el cálculo de la textura se tiene en cuenta la relación existente entre un píxel central y sus vecinos dentro de la ventana. Realizando el mismo proceso a los demás píxeles dentro de la imagen, se genera una imagen de texturas a partir de los resultados obtenidos en cada ventana; el anterior proceso se ilustra en la Figura 2, donde se tiene una imagen de tamaño 7x9 píxeles y una subventana o ventana movable de 3x3 píxeles, generando así una imagen de salida de tamaño 5x7 píxeles, cabe anotar que el tamaño de la ventana movable por lo general es impar.

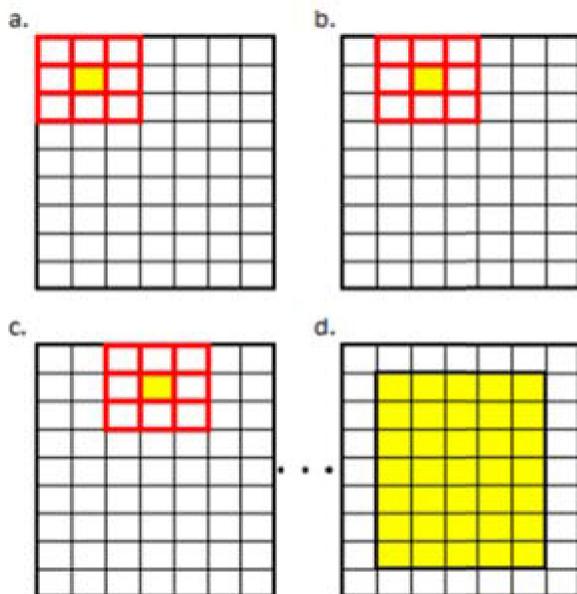


Figura 2. Técnica de movimiento de ventana

METODOLOGÍA

En la metodología se seleccionó un área de estudio de la imagen de satélite, a la cual se le hizo el procesamiento digital por técnicas fractales, la imagen seleccionada debe ser escalada a un rango de 256 niveles de gris y almacenada con una resolución radiométrica de 8 bits. Se implementaron los algoritmos fractales y multifractales por medio de la metodología de conteo de cajas mejorado DBC y el algoritmo GLCM. Posteriormente se realiza la clasificación no supervisada de la imagen fractal resultante y la imagen de texturas GLCM.

Materiales

Los algoritmos fractal y multifractal fueron implementados en el lenguaje de programación Python, utilizando las librerías científicas GDAL, Numpy y Scipy principalmente, de igual forma el algoritmo GLCM; para el manejo y visualización de la imagen se utilizó el software de sistemas de información geográfica Qgis.

Las características del equipo de cómputo donde se realizó la implementación de los algoritmos se muestra en la Tabla 1.

Tabla 1. Equipo de cómputo utilizado

Características Equipo	
Procesador	Intel Core i5
Velocidad Procesador	2,4 GHz
Memoria	4 GB
Espacio en disco proyecto	11 GB
Espacio en disco implementación algoritmo	3 GB
Espacio Total Disco Duro	14 GB

Se utilizó una imagen SAR del sensor Palsar de libre acceso del siguiente enlace <https://vertex.daac.asf.alaska.edu/>, la cual tiene las características referenciadas en la Tabla 2.

Tabla 2. Características imagen SAR

Características Imagen	
Tipo	SAR
Nombre sistema	ALOS Palsar
Fecha captura	07 de agosto 2007
Banda	L
Modo	Fino
Polarización	HH
Resolución radiométrica	32 bits
Resolución Espacial	12,5
Agencia	Japan Aerospace Exploration Agency (JAXA)
Tamaño imagen	6579 x 5780
Zona Geográfica	Bogotá D.C.

De dicha imagen se escogió una zona de tamaño 512x512 píxeles, ya que la implementación funciona sólo para imágenes de tamaño cuadrado. En la Figura 3 se visualiza la imagen trabajada.

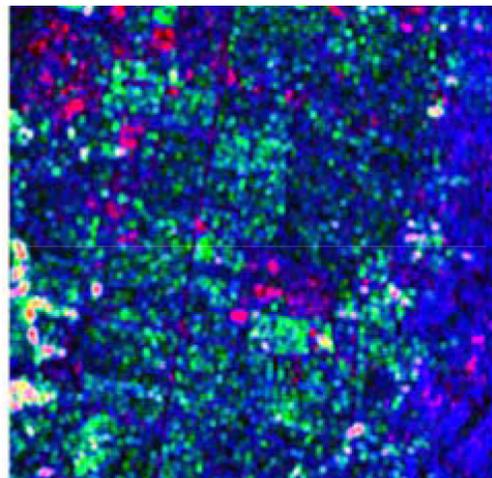
**Figura 3.** Zona escogida de la imagen SAR

La zona de la imagen corresponde a la parte urbana (sector noroccidental) de la ciudad de Bogotá D.C., dicha zona abarca desde las coordenadas geográficas $74^{\circ}4'27,1''$ a $74^{\circ}0'58,8''$ de longitud Oeste y $4^{\circ}41'17,3''$ a $4^{\circ}44'45,7''$ de latitud Norte. Se realiza el cálculo de las imágenes de textura a partir de los momentos $D(q=0,1,4)$ obteniendo tres imágenes fractales y de las imágenes de textura por el método GLCM, calculando las estadísticas ASM (segundo momento angular), contraste y

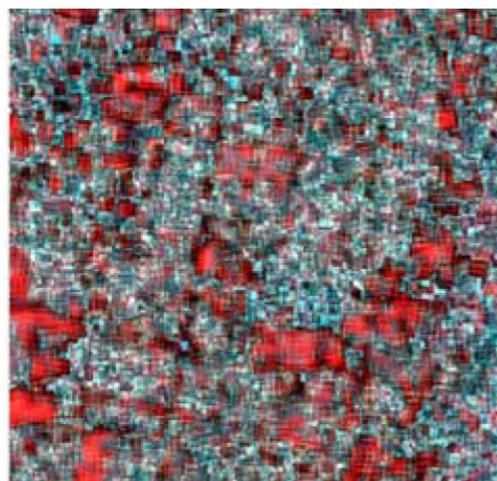
correlación, realizando una combinación de bandas entre las tres imágenes de textura para el método multifractal y el método GLCM, se obtienen las imágenes mostradas en la Figura 4(a) y 4(b).

RESULTADOS

Para el análisis fractal se utilizó una ventana móvil de tamaño 12 píxeles, por otro lado, para el análisis por medio del método GLCM se utilizó un tamaño de ventana móvil de 7 píxeles, esta elección se tomó de forma manual, determinando visualmente el tamaño de ventana más adecuado.



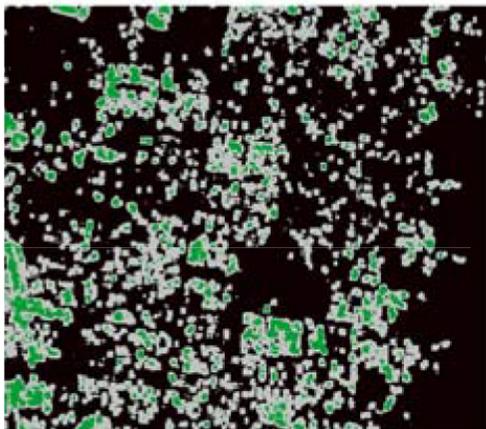
(a)



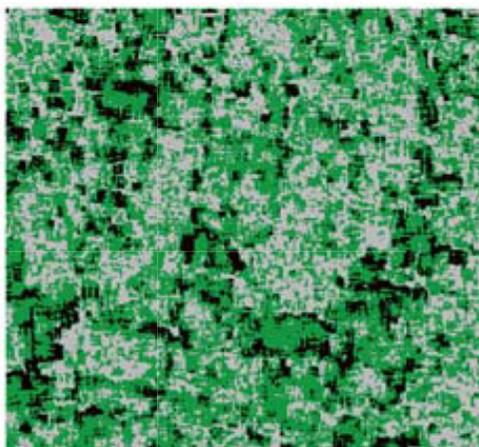
(b)

Figura 4. (a) Composición de bandas de estadísticas GLCM. (b) Composición de bandas multifractales

Posteriormente, se obtuvieron los resultados observados en la Figura 5, donde se puede observar el resultado de haber realizado la clasificación no supervisada de las imágenes de texturas combinadas. Volviendo a la clasificación obtenida de la Figura 5(b), se tienen mapas resultantes con agrupación de tres clases donde la zona con color negro representa superficies muy lisas, por ejemplo, vías nuevas, pistas de aterrizaje y objetos con textura fina; el color verde representa un nivel de rugosidad medio y, por último, el color gris representa un nivel considerablemente alto en la textura, por ejemplo, construcciones muy antiguas o carreteras no pavimentadas o zonas de montañas muy rocosas.



(a)



(b)

Figura 5. (a) Mapa clasificación no supervisada GLCM, (b) Mapa clasificación no supervisada método multifractal para $q=0,1,4$

Las imágenes de texturas individuales no ofrecen mucha información a la hora de realizar una clasificación no supervisada, lo que hace necesario realizar una combinación de las 3 imágenes, o más, para así obtener un resultado más óptimo.

Como se observa en la Figura 5, es muy complicado realizar una comparación exacta entre métodos, en primer lugar porque las ventanas móviles no son del mismo tamaño para cada método, además, los resultados obtenidos muestran una considerable diferencia sobre todo para el método GLCM, ya que como se ve en la imagen hay muchas zonas con el mismo valor de rugosidad pero se combina con otras unidades. En el método multifractal se observa una mejor distribución en la agrupación de las clases agrupadas.

DISCUSIÓN DE RESULTADOS

Como se observó en las imágenes anteriores, se puede determinar que la utilización de los algoritmos fractales y multifractales no ofrecen mucha información a la hora de realizar una eventual clasificación no supervisada, esto a partir de la observación de la imagen.

El método DBC para generar la imagen fractal mejora un poco en cuanto la agrupación de zonas con características de rugosidad similar, aun así este método presenta inconsistencias ya que la dimensión fractal de objetos con texturas diferentes en algunos casos resulta ser la misma; entonces se tiene la dimensión multifractal la cual resulta ser un método más óptimo a la hora de extraer y analizar texturas, en este método se supera la dificultad que presenta la dimensión monofractal anteriormente explicada.

CONCLUSIONES

El software de programación Python facilitó, en gran medida, la implementación de los algoritmos de geometría fractal para extracción de texturas

debido a que es un lenguaje de programación muy fácil de aprender, además de que existen múltiples librerías para el manejo de datos raster, lo cual agilizó el desarrollo del proyecto.

En cuanto a lo que tiene que ver a los parámetros de entrada para la obtención de imágenes de textura por medio de los algoritmos disponibles, según la bibliografía consultada y los resultados obtenidos, se tiene que la elección del tamaño de ventana móvil es una tarea complicada, ya que debe realizarse el proceso de forma mecánica probando uno a uno los tamaños de ventana hasta encontrar el tamaño más preciso e indicado.

A pesar de las desventajas presentadas dentro del análisis de texturas por medio de algoritmos fractales, estos se convierten en alternativa a los algoritmos comúnmente utilizados, ya que continuamente surgen mejoras y nuevas propuestas desde el campo de la geometría fractal para el procesamiento de imágenes de teledetección como el análisis y extracción de texturas.

REFERENCIAS

- Canada Center for Remote Sensing CCRS/CCT, *Fundamentals of Remote Sensing*. Canada: Autor.
- Clarke, C. (1986). Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method. *Computers & Geosciences*, 12(5), 713-722.
- Jensen, J. (1996). *Introductory Digital Image Processing*. Ed. Prentice Hall.
- Lopes, R., Betrouni, N. (2009) *Fractal and multifractal analysis: A review*.
- Mandelbrot, B. (1967). *How long is the coast of Britain? Statistical self-similarity and fractional dimension*. Londres:
- Mandelbrot, B. (1977). *Fractals: Form, chance and dimension*. CA: San Francisco.
- Mandelbrot, B. (1977). *The Fractal Geometry of Nature*, Londres:
- Matter, P. and Tso, B. (2009). *Classification Methods for Remotely Sensed Data*, 2 ed.
- Pant, T., Singh, D., Srivastava T. (2008). Multifractal Analysis of SAR Images for Unsupervised Classification. *Microwave*, 8, 21-24.
- Pant, T., Singh, D., Srivastava T. (2010). Advanced fractal approach for unsupervised classification of SAR images. *Advances in Space Research*, 45(11), 1338-1349.
- Pentland A. (1984). Fractal based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 661-74.
- Presuti, M. (2004). *La Matriz de Co-Ocurrencia en la clasificación multiespectral: Tutorial para la enseñanza de medidas texturales en cursos de grado universitario*. Buenos Aires: La Plata.
- Rubiano G. (2008). *Iteración y fractales con Matemática*, Bogotá D.C.: Universidad Nacional de Colombia, Facultad de Ciencias.
- Sarkar, C. (1992). An Efficient Approach to Estimate Fractal Dimension of Textural Images. *Pattern Recognition*, 25(9), 1035-1041.
- Sarkar, C. (1995). *Multifractal and generalized dimensions of gray-tone digital images*. Signal Processing, 42(2), 181-190.
- Sun, W., Xu, G., Gong, P. and Liang, S. (2006). Fractal analysis of remotely sensed images: A review of methods and applications. *International Journal of Remote Sensing*, 27(22), 4963-4990.
- Theera, N. (2002). Fractal Dimension Estimation using Modified Differential BoxCounting and Its Application to MSTAR Target Classification.
- Yu, L., Lingyu, C., Wang, H., Jiang, L., Zhang, L., Zhao, J., Wang, D., Zhao, Y. and Song, Y. (2009). An improved differential box-counting method to estimate fractal dimensions of gray-level images. *Pattern Recognition*, 42, 2460-2469