

# Clasificación y Segmentación de Texturas usando Dimensión Fractal y Contornos B-spline Deformables

Horacio Abbate<sup>1</sup>, María E. Buemi<sup>1</sup>, Juliana Gambini<sup>1</sup>, Claudio Delrieux<sup>2\*</sup>

<sup>1</sup>Facultad de Ciencias Exactas y Naturales, UBA, Buenos Aires, Argentina  
{habbate; mebuemi; jgambini}@dc.uba.ar

<sup>2</sup>Departamento de Ingeniería Eléctrica y Computadoras, UNS  
Bahía Blanca, Argentina, claudio@acm.org

## Resumen.

En este trabajo se estudia el problema del reconocimiento y la segmentación de texturas en imágenes. Se presenta una técnica basada en la dimensión fractal (DF) y contornos B-spline deformables para hallar el borde de un objeto de interés. Sobre un imagen original se aplican 7 características de DF y una de multifractalidad. Para estimar la DF se propone un enfoque box-counting modificado combinado con la característica de suavizado por difusión anisotrópica para disminuir regiones espúreas. Se utiliza el método de clasificación no supervisada mediante *K-medias*. Se muestran varios ejemplos con imágenes sintéticas de diferentes texturas, en los cuales se observa que el uso de la dimensión fractal local, como descriptor para la búsqueda de texturas, es adecuado para la extracción de contornos en este tipo de imágenes.

**Palabras Clave:** PROCESAMIENTO DE IMÁGENES; DETECCIÓN DE CONTORNOS; TEXTURAS; ANÁLISIS MULTIFRACTAL

## 1. Introducción

El análisis de la dimensión fractal se ha utilizado como descriptor en la última década, fundamentalmente para la segmentación basada en texturas y rugosidad. Sin embargo, hasta ahora no había sido utilizado en combinación con métodos locales de segmentación como los contornos activos. Las características de texturas estudiadas aquí están basadas,

---

\*Parcialmente financiado por la SECyT-UNS

en la geometría fractal de imágenes, partiendo de que la dimensión fractal (DF) es relativamente insensible a cambios de escala de la imagen y muestra una fuerte correlación con el criterio del ojo humano de rugosidad de superficies.

La técnica propuesta para extracción de contornos, está basada en el trabajo de A. Blake *et al.* [1] sobre ajuste de contornos con curvas B-spline.

Este trabajo está organizado de la siguiente manera: en la sección 2 se describe el modelo utilizado para estimar la dimensión fractal. En la sección 3 se presenta la forma de calcular los 8 descriptores de una textura. En la sección 4 se describe el algoritmo de Difusión Anisotrópica utilizado para suavizar las características. En la sección 5 se presenta un resumen de la representación de curvas con B-spline, el algoritmo de detección de puntos de borde, y una forma de hacer más eficiente este algoritmos reduciendo la cantidad de puntos de borde. En la sección 6 se muestran los resultados obtenidos, y finalmente, en la sección 7 se presentan las conclusiones.

## 2. Estimación de la Dimensión fractal

En imágenes, así como en otros conjuntos cuya función característica es no determinística, existe una relación directa entre la dimensión fractal y el coeficiente de autocorrelación de la función [2, 3]. Esta relación justifica la posibilidad de estimar la dimensión fractal en este tipo de conjuntos por métodos estadísticos que extienden a las definiciones clásicas de fractalidad como por ejemplo la dimensión de Hausdorff (o de autosimilitud). Existen diversos métodos para estimar la dimensión fractal de un conjunto dado, en este trabajo se utiliza la dimensión fractal diferencial, propuesta por Chaudhuri *et al.* [4].

Un conjunto acotado  $A$  en un espacio, es *autosimilar* si se puede mostrar que  $A$  es la unión de  $N_r$  copias distintas (no solapadas) de sí mismo escalada por encima o por debajo de un radio  $r$ . En este caso, la dimensión fractal  $DF$  de  $A$  está dada por la relación:

$$DF = \lim_{r \rightarrow 0} \frac{\log(N_r)}{\log(1/r)} \quad (1)$$

El concepto de autosimilaridad puede ser utilizado para estimar la dimensión fractal de un conjunto arbitrario. Existen diversos métodos para calcular esta estimación, la mayoría de los cuales se aplican a imágenes que deben estar previamente binarizadas. Chaudhuri *et al.* describen un método que tiene la ventaja de que puede aplicarse a imágenes que tengan niveles de gris entre  $[0, 255]$ . La ecuación 1, es la base de la estimación de la dimensión fractal. En este caso  $N_r$  está determinada de la siguiente manera: dada una imagen de  $M \times M$  pixels, se hace una partición de  $s \times s$  donde  $M/2 \geq s > 1$  y  $s \in \mathbb{Z}$ . Luego el radio es  $r = s/M$ . Se considera la imagen perteneciente a un espacio tridimensional, donde cada elemento es una terna  $(x, y, z)$ , con  $(x, y)$  la posición del pixel y  $z$  es el nivel de gris correspondiente al pixel  $(x, y)$ .

El espacio  $(x, y)$  está particionado en una grilla cuya celdas son de lado  $s$ . Sobre cada una de las celdas se considera una columna de cajas de tamaño  $s \times s \times s$ , es decir se hace una partición de los niveles de gris sobre cada celda. Si el máximo y mínimo nivel de

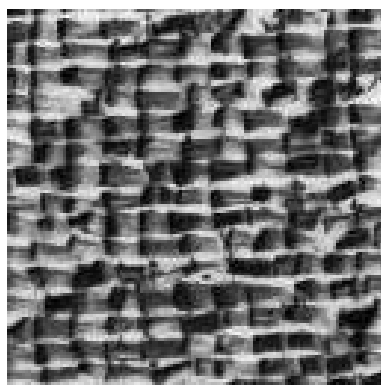
gris caen en la  $k$ -ésima y  $l$ -ésima celdas de la columna que se levanta sobre la celda  $(i, j)$ , entonces

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

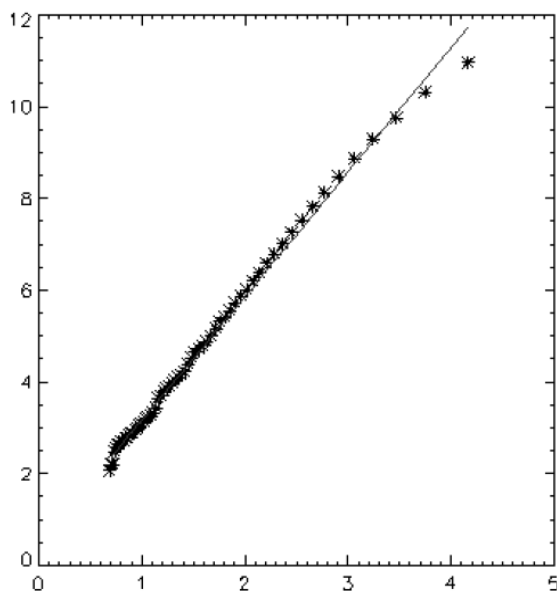
es la contribución de  $N_r$  en la  $(i, j)$ -ésima celda. Tomando todas las contribuciones sobre la grilla se obtiene:

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

$N_r$  se calcula para diferentes valores de  $r$ . Luego, la dimensión fractal  $DF$  se estima como la pendiente de la recta de mínimos cuadrados que ajusta  $\log(N_r)$  vs.  $\log(\frac{1}{r})$ . La Figura 1 muestra una textura tomada del álbum de Bordatz ([5]) y la recta de mínimos cuadrados correspondiente.



(a)



(b)

Figura 1: Ejemplo de la estimación de la Dimensión Fractal de una imagen de textura. (a) Imagen D84 del álbum de Brodatz. (b) La recta de mínimos cuadrados que ajusta los datos, calculada por el método.

### 3. Selección de Características

Mandelbrot y Van Ness [3] han puntualizado que diferentes texturas tienen la misma dimensión fractal. En este trabajo se propone usar 8 características  $f_i$ , ( $1 \leq i \leq 8$ ) para discriminar aspectos de rugosidad y orientación. Estas características están basadas en el cálculo de la dimensión fractal de la imagen original, la DF de los píxeles de la imagen con altos valores de luminancia, la DF sobre bajos valores de luminancia, la DF de la imagen convolucionada con filtros de suavizado horizontal, suavizado vertical, suavizado diagonal, y la dimensión multifractal de exponente 2 de la imagen original. Todas las características  $f_i$  son normalizadas de manera que  $f_i \in [0, 1]$ .

CARACTERÍSTICA 1: La dimensión fractal de la imagen original  $I_1$  se calcula sobre ventanas solapadas de tamaño  $(2W + 1) \times (2W + 1)$ . Por lo tanto en el píxel  $(i, j)$  la primera característica  $F_1(i, j)$ , está definida como

$$F_1(i, j) = DF\{I_1(i + l, j + k)\}$$

donde

$$-W \leq l, k \leq W$$

Dado que  $2 \leq F_1(i, j) \leq 3$ , se define la característica normalizada como  $f_1(i, j) = F_1(i, j) - 2$ , de tal modo  $0 \leq f_1(i, j) \leq 1$ .

CARACTERÍSTICAS 2 Y 3: Considerando dos imágenes  $I_2$  y  $I_3$ , definidas como

$$I_2(i, j) = \begin{cases} I_1(i, j) - L_1 & \text{si } I_1(i, j) > L_1 \\ 0 & \text{en todo otro caso} \end{cases}$$

$$I_3(i, j) = \begin{cases} 255 - L_2 & \text{si } I_1(i, j) > (255 - L_2) \\ I_1(i, j) & \text{en todo otro caso} \end{cases}$$

donde  $L_1 = g_{min} + \frac{av}{2}$ ;  $L_2 = g_{max} - \frac{av}{2}$  mientras  $g_{max}$ ,  $g_{min}$  y  $av$  denotan el máximo, mínimo y porcentaje de valores de gris en  $I_1$  respectivamente. Si dos imágenes  $I_1$  y  $J_1$  tienen la misma dimensión fractal, sus imágenes  $I_2$  y  $J_2$  no tienen una rugosidad idéntica y sus dimensiones fractales son diferentes. Lo mismo ocurre con  $I_3$  y  $J_3$ . Las características normalizadas  $f_2$  y  $f_3$  se computan de  $I_2$  y  $I_3$  de la misma forma que  $f_1$  desde  $I_1$ .

CARACTERÍSTICAS 4 A 7: Para una textura altamente orientada, la DF se ve menos afectada si se realiza un suavizado en su orientación dominante. Esto motiva el hecho de tomar como descriptores de la textura, la estimación de la dimensión fractal de cuatro imágenes suavizadas en cuatro direcciones, vertical, horizontal y diagonales,

de la siguiente manera:

$$\begin{aligned}
I_4(i, j) &= \frac{1}{2w+1} \sum_{k=-w}^w I(i, j+k), \\
I_5(i, j) &= \frac{1}{2w+1} \sum_{k=-w}^w I(i+k, j), \\
I_6(i, j) &= \frac{1}{2w+1} \sum_{k=-w}^w I(i+k, j+k), \\
I_7(i, j) &= \frac{1}{2w+1} \sum_{k=-w}^w I(i+k, j-k).
\end{aligned}$$

La DF normalizada de las características  $f_4$  a  $f_7$  son calculadas de la misma forma que  $f_1$ .

**CARACTERÍSTICA 8:** Esta última característica está basada en el cálculo de la dimensión multifractal con exponente de orden 2, para más detalles ver [6]. Un conjunto  $A$  es *multifractal* si la Ec. 1 converge a diferentes valores al utilizar diferentes exponentes. Sea  $\mu_r(x)$  la medida del conjunto  $A$  en una esfera de radio  $r$  alrededor del punto  $x$ . La distribución en la posición  $(i, j)$  es  $\mu_r(i, j) = \frac{n_r(i, j)}{N_r}$ , donde  $n_r(i, j)$  y  $N_r$  están definidas por las ecuaciones 2 y 2 respectivamente.

Se define la función de partición

$$\chi(q, r) = \sum_{(i, j)} [\mu_r(i, j)]^q.$$

La DF generalizada  $D(q)$  resulta

$$(q-1)D(q) = \lim_{r \rightarrow 0} \frac{\ln \chi(q, r)}{\ln r},$$

donde  $q \neq 1$ . Luego, la octava característica normalizada es

$$f_8 = TD - D(2),$$

siendo TD la dimensión topológica.

Para el cómputo de  $f_8$  se buscaron las ventanas que experimentalmente produjeran buenos resultados sin tiempos excesivos de computo, siendo los ideales los valores  $W = 17$  y  $w = 7$ .

## 4. Suavizado y Segmentación

Si las características son utilizadas directamente para segmentación, pueden ocurrir errores en la clasificación tanto en las regiones interiores, como en las regiones de bordes. El suavizado reduce los errores de clasificación dentro de las regiones con texturas.

En este trabajo se utiliza el algoritmo de Difusión Anisotrópica para suavizar todas las características desarrollado por Perona y Malik en [7]

La formulación de Perona y Malik presenta un algoritmo que remueve el ruido manteniendo las características de borde. Este algoritmo modifica la imagen de acuerdo a la siguiente ecuación:

$$\frac{\partial I(x, y, t)}{\partial t} = \text{div}[g(\|\nabla I\|)\nabla I]$$

donde  $I(x, y, 0) : \mathbb{R}^2 \rightarrow \mathbb{R}^+$  es la imagen original,  $t$  es un parámetro de tiempo artificial,  $\nabla I$  es el gradiente de la imagen,  $\|\nabla I\|$  es la magnitud del gradiente de la imagen y  $g(\|\nabla I\|)$  es una función de detección de borde, con la única condición de que  $g(x) \rightarrow 0$  cuando  $x \rightarrow 0$ . La clasificación de texturas se realiza con el conocido algoritmo de clustering *K-medias* (ver [8]).

## 5. Detección de bordes utilizando representación B-spline para curvas

En esta sección presentamos el algoritmo de detección de bordes utilizado en este trabajo para detectar los cambios de texturas dentro de la imagen y hallar el contorno del objeto de interés. Con este propósito se utiliza la representación B-spline para curvas, que tiene la ventaja de que la ecuación de la curva queda representada por sus puntos de control, lo cual reduce ampliamente la cantidad de parámetros y por lo tanto el costo computacional. Además puede elegirse el orden de la representación, de acuerdo a la suavidad necesaria para la curva. Con el objetivo de que este trabajo sea autocontenido se presenta, en primer lugar, un breve resumen de la representación B-spline de contornos y de la interpolación de puntos por medio de B-spline utilizado para el ajuste; para más detalles sobre este tema consultar [1, 9].

### 5.1. Representación B-spline

Sea  $\{Q_0, \dots, Q_{N_B-1}\}$  un conjunto de puntos de control, donde  $Q_n = (x_n, y_n)^t \in \mathbb{R}^2$ ,  $0 \leq n \leq N_B - 1$ , y sea  $\{s_0 < s_1 < s_2 < \dots < s_{L-1}\} \subset \mathbb{R}$  un conjunto de  $L$  nodos. Una curva B-spline de orden  $d$  se define como una suma ponderada de  $N_B$  funciones polinomiales  $B_{n,d}(s)$  de grado  $d - 1$ , dentro del intervalo  $[s_i, s_{i+1}]$  con  $0 \leq i \leq L - 1$ . La curva B-spline se contruye como  $r(s) = (x(s), y(s))^t$ ,  $0 \leq s \leq L - 1$ , siendo

$$r(s) = \sum_{n=0}^{N_B-1} B_{n,d}(s)Q_n,$$

y

$$\begin{aligned} x(s) &= B^t(s).Q^x \\ y(s) &= B^t(s).Q^y \end{aligned}$$

donde el vector de funciones base  $B(s)$  de  $N_B$  componentes está dado por  $B(s) = (B_{0,d}(s), \dots, B_{N_B-1,d}(s))^t$ . Los vectores de pesos  $Q^x$  y  $Q^y$  son las primeras y segundas componentes de  $Q_n$ , respectivamente.

Las curvas usadas en este trabajo para extracción de contornos son cerradas, de orden  $d = 3$  ó  $d = 4$ , especificadas por funciones B-spline periódicas.

Para resolver el problema de encontrar los  $N_B$  puntos de control de una curva B-spline que ajuste un contorno, utilizamos el siguiente esquema desarrollado en [9].

Dado un conjunto de  $k$  puntos en el plano de la imagen  $\{D_0, D_1, \dots, D_{k-1}\}$ , donde  $D_i = (x_i, y_i)^t$ ,  $i = 0, \dots, k-1$ , buscamos la curva B-spline de  $N_B$  puntos de control que mejor los ajusta. Entonces, por las ecuaciones 2 y 2, las componentes  $D_i$  deben satisfacer

$$\begin{aligned} x_i &= B^t(t_i)Q^x, \\ y_i &= B^t(t_i)Q^y, \end{aligned}$$

para ciertos valores de  $t_i$ , donde  $i = 0, \dots, k-1$  y  $N_B \leq k$ .

Este sistema lineal puede ser escrito en forma matricial como  $D = K(Q^x \quad Q^y)$ , donde los  $k \times N_B$  elementos de la matriz  $K$  están dados por  $K_{ij} = B_{j,d}(t_i)$ , con  $i = 0, \dots, k-1$ ,  $j = 0, \dots, N_B-1$ , y  $D = (D_0, D_1, \dots, D_k)^t$ .

En el caso más general es  $N_B < k$  y por lo tanto  $K$  no es una matriz cuadrada. En este caso se utiliza la matriz pseudo-inversa para encontrar una solución aproximada del sistema, de donde resulta

$$(Q^x \quad Q^y) = K^+ D$$

El conjunto de valores de los parámetros  $\{t_0, \dots, t_{k-1}\}$  está dado por

$$t_0 = 0, \quad ; t_\ell = \frac{\sum_{i=1}^{\ell} \|D_i - D_{i-1}\|}{\sum_{i=1}^{k-1} \|D_i - D_{i-1}\|}, \quad \ell \geq 1.$$

## 5.2. Algoritmo de detección de bordes

En esta sección se describe el algoritmo desarrollado para detectar el borde del objeto de interés. Sea  $E$  una escena compuesta por el fondo  $B$  y una región  $R$  con su borde  $\partial R$ . Se desea encontrar la curva  $C_B$  que ajusta al borde de la región  $\partial R$  en la imagen. Definimos una región inicial de búsqueda, especificada por un polígono, cuyos vértices son los puntos de control que generan una curva B-spline.

Si un punto pertenece al borde de la imagen, entonces una muestra tomada sobre un entorno del punto exhibe un cambio en la intensidad de los niveles de gris y es considerado un punto de transición. Entonces se consideran  $N$  segmentos  $a^{(i)}$ ,  $i \in \{1, \dots, N\}$  con la forma  $a^{(i)} = \overline{CP_i}$ . Aquí  $C$  es el centroide de la región inicial, el extremo  $P_i$  es un punto fuera de la región y  $\theta = \text{ang}(a^{(i)}, a^{(i+1)}) \forall i$  es el ángulo entre dos segmentos consecutivos como muestra la Figura 2. El segmento  $a^{(i)}$  es un arreglo de  $m$  elementos provenientes de la discretización una recta sobre la imagen.

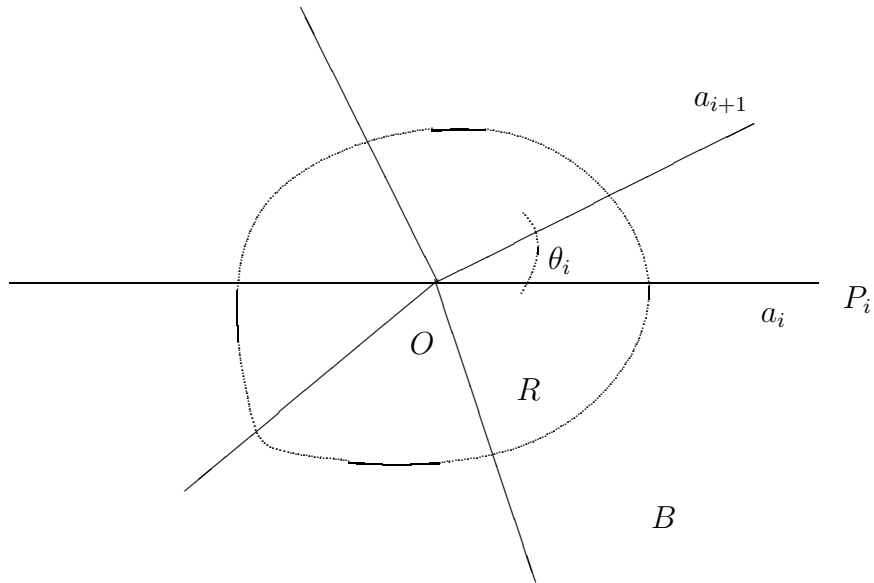


Figura 2: Rectas radiales.

Después de aplicar la clasificación explicada en la sección 4, la imagen es binaria, entonces el punto de borde sobre cada segmento se encuentra convolucionando los datos del segmento con una máscara dada por  $[-1, 1]$ . Una vez encontrados los puntos de borde  $\{b_1, \dots, b_N\}$  que están sobre cada segmento se contruye la curva B-spline interpolante como se explica en la sección 5.1

### 5.3. Algoritmo de elección de puntos de borde

El algoritmo de la sección 5.2 genera una cantidad exagerada de puntos de borde. No todos son necesarios para construir una buena aproximación del contorno del objeto de interés. En esta sección se explica una forma de hallar los puntos de borde más significativos en la descripción del contorno, de manera de generar la misma curva aproximante con la menor cantidad de puntos de borde posible. La curva B-spline interpolante se genera con 15 puntos de control.

Dado un conjunto de puntos borde  $A = \{b_1, \dots, b_N\}$  se quiere determinar los puntos más importantes en la descripción de la curva (puntos críticos), eliminando los puntos que son redundantes. Por ejemplo, los puntos donde la curva cambia de dirección, o los puntos en los que la curvatura es máxima. El método que proponemos fue desarrollado originalmente por P. Zhu (see [10]) y trata de detectar puntos críticos, dentro del conjunto de puntos de borde  $A$ , eligiendo aquellos que tienen curvatura máxima pero sin calcular la curvatura.



El conjunto de puntos  $A$  puede ser particionado en dos subconjuntos  $D$  (puntos críticos) y  $E$  (puntos redundantes), de manera que:

$$\begin{aligned} C \cap R &= \emptyset \\ C \cup R &= B \end{aligned}$$

Tomamos las coordenadas polares de cada punto  $b_i$

$$b_i = (\rho_i, \alpha_i)$$

donde  $\rho_i$  y  $\alpha_i$  son el módulo y el ángulo respectivamente, y con lo que el punto  $b_i$  queda unívocamente determinado.

El conjunto  $A$  es una secuencia y por lo tanto cada punto tiene solo dos puntos geoméricamente vecinos. Si  $b_{i-1}$ ,  $b_i$  y  $b_{i+1}$  son tres puntos consecutivos, entonces habrá un máximo o un mínimo local si se cumple

$$\text{sign}(\rho_i - \rho_{i-1}) \neq \text{sign}(\rho_{i+1} - \rho_i)$$

ó

$$\text{sign}(\alpha_i - \alpha_{i-1}) \neq \text{sign}(\alpha_{i+1} - \alpha_i)$$

De esta manera se obtiene un subconjunto  $M$  de puntos máximos o mínimos locales,

$$M = \{b_{c_1}, \dots, b_{c_m}\}$$

Luego, por cada par de puntos adyacentes  $b_{c_j}, b_{c_{j+1}}$  de  $M$  se calcula la recta  $S_j$  que pasa por esos puntos y luego la distancia  $d_i$  que existe entre los puntos  $p_i$  pertenecientes al conjunto  $A - M$  y la recta  $S_j$ . Entonces se obtienen los puntos que tienen máxima y mínima distancia  $d_i$  a la recta  $S$ , seleccionando aquellos que cumplen la condición:

$$\text{sign}(d_i - d_{i-1}) \neq \text{sign}(d_{i+1} - d_i) \quad (2)$$

para  $c_j < i < c_{j+1}$ . Queda entonces definido el conjunto

$$\Gamma = \{\tilde{p}_{b_0}, \tilde{p}_{b_1}, \dots, \tilde{p}_{b_{r-1}}\}$$

compuesto por todos los puntos que cumplen (2).

Teniendo ahora  $M$  y  $\Gamma$ , se obtiene  $C_0 = \text{sort}(M \cup \Gamma)$  donde se han reordenado los puntos de acuerdo al orden original que tenían en  $A$ . A estos puntos se los llama pseudo-críticos.

El paso siguiente es elegir los puntos críticos dentro de el conjunto  $C_0$ . Para esto definimos el nivel crítico  $l(p_i)$  de la siguiente manera: se toman tres puntos consecutivos  $p_{i-1}, p_i, p_{i+1}$  y se calcula

$$l(p_i) = \text{area}(p_{i-1}, p_i, p_{i+1})^{\Delta}$$

el área del triángulo que forman.

Luego, de manera recursiva, se eliminan los puntos donde  $l(p_i) < l$ , donde  $l$  es un umbral que puede elegirse arbitrariamente o con algún criterio.  $l(p_i)$  da una medida de la importancia que tiene un punto para la descripción de la curva.

$$D = \{p_i \in C_0, l(p_i) \geq l\}$$

$$E = \{p_i \in C_0, l(p_i) < l\}$$

En este trabajo se ha tomado  $l = \min \left\{ \frac{l(p_i)}{3}, p_i \in A \right\}$ .

## 6. Resultados

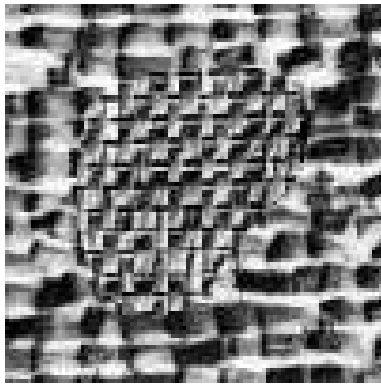
En las Figuras 3 y 4 se muestra el resultado de aplicar todo el algoritmo a dos imágenes, ambas compuestas por un objeto y el fondo con texturas distintas, tomadas del álbum de Brodatz ([5]). Las Figuras 3(a) y 4(a) muestran las imágenes originales a las que se les aplican los algoritmos.

Las Figuras 3(b) y 4(b) muestran la clasificación realizada por los algoritmos de las secciones 3 y 4. Las Figuras 3(c) y 4(c) muestran el resultado de aplicar además el algoritmo de segmentación utilizando curvas B-spline. Se observa en estas figuras, el suavizado del borde, característico de las curvas B-spline.

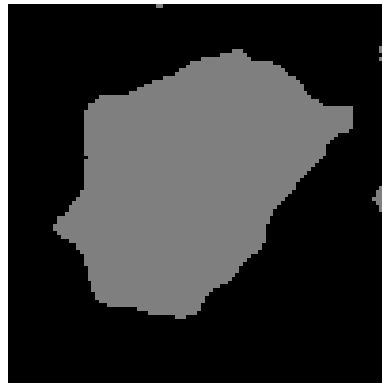
## 7. Conclusiones

En este trabajo se describe un enfoque en segmentación de imágenes de texturas usando una técnica de clasificación basada en la estimación de la dimensión fractal y contornos deformables. La dimensión fractal de dos texturas distintas puede coincidir, pero tomando ocho descriptores basados en la dimensión fractal se obtiene un espacio 8-dimensional, en el cual cada elemento corresponde a una textura diferente. Esto resulta un algoritmo eficiente para clasificar. El algoritmo de contornos deformables es una herramienta muy útil para encontrar bordes de regiones por su eficiencia computacional. Además, trabaja sobre regiones previamente especificadas lo cual reduce el costo computacional. Por último, tiene la ventaja que produce una curva suave para el contorno del objeto de interés con poca cantidad de parámetros.

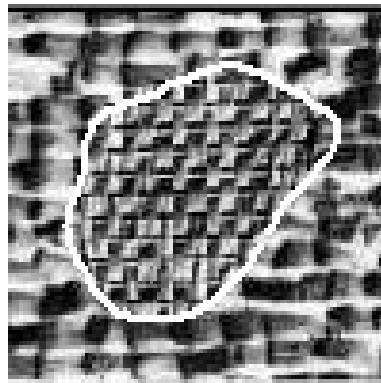
En el primer paso se calculan los descriptores basados en la dimensión fractal y multifractal de orden 2, luego se procede al suavizado utilizando difusión anisotrópica para hacer la clasificación utilizando el algoritmo de K-medias. A la imagen clasificada, y por lo tanto binarizada, se aplica un proceso de detección de puntos de borde sobre una serie de segmentos y luego se construye la B-spline interpolante. Se obtiene también una fórmula matemática para el borde del objeto buscado. Estas ideas fueron aplicadas a imágenes en las cuales los algoritmos de segmentación de áreas o fronteras no son adecuados. Los resultados experimentales consignados en este trabajo son excelentes.



(a)



(b)

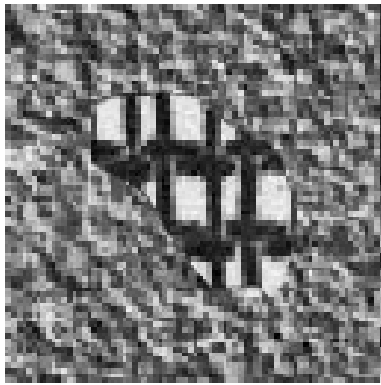


(c)

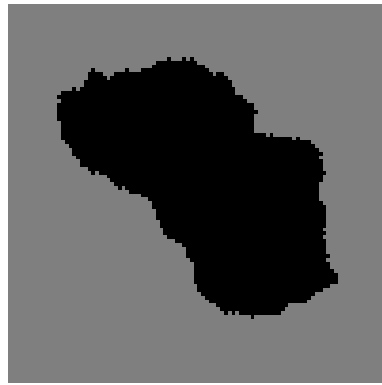
Figura 3: (a) Imagen que muestra un objeto con una textura distinta que la textura del fondo, ambas tomadas del álbum de Brodatz. (b) Resultado de aplicar a la imagen mostrada en (a), el algoritmo de clasificación utilizando los 8 descriptores como se explica en la sección 3. (c) Resultado de la segmentación utilizando el algoritmo de contornos.

## Referencias

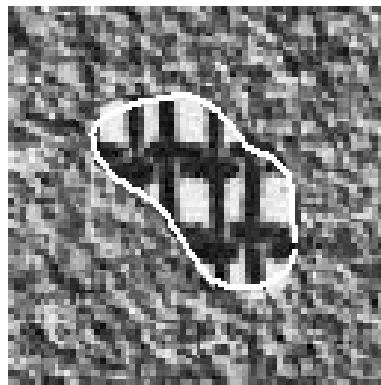
- [1] A. Blake and M. Isard, *Active Contours*. Springer Verlag, 1998.
- [2] H. O. Peitgen and D. Saupe, *The Science of Fractal Images*. Springer-Verlag, 1986.
- [3] B. Mandelbrot and J. van Ness, “Fractional brownian motion, fractional noises and applications.” *Siam Review*, vol. 10, pp. 422–437, 1983.
- [4] B. Chaudhuri and N. Sakar, “Texture segmentation using fractal dimension,” *IEEE Trans. Pattern Anal. Machine Intell*, vol. 25, no. 17, pp. 72–76, 1995.
- [5] P. Brodatz, *Texture: A Photographic Album for Artist and Designers*. New York: Dover, 1966.
- [6] L. Pietronero and R. Kusters, *Fractals in physics*. Amsterdam: L. Pietronero and E. Tosatti, eds. North-Holland, 1986.



(a)



(b)



(c)

Figura 4: (a) Imagen que muestra un objeto con una textura distinta que la textura del fondo, ambas tomadas del álbum de Brodatz. (b) Resultado de aplicar a la imagen mostrada en (a) el algoritmo de clasificación utilizando los 8 descriptores como se explica en la sección 3. (c) Resultado de la segmentación utilizando el algoritmo de contornos.

- [7] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 7, pp. 629–639, 1990.
- [8] J. S. Lim, *Two-dimensional Signal and Image Processing*. New York, USA: Prentice Hall, 1990.
- [9] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, 2nd ed. New York, USA: McGraw-Hill, 1990.
- [10] P. Zhu and P. Chirlian, "On critical point detection of digital shapes." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 737–748, 1995.