

Segmentación de Texturas con funciones de Gabor

Dan Williams Robledo Cruz
Visión por Computadora
Laboratorio de Tecnologías de Información,
Cinvestav
Cd. Victoria, Tamaulipas
Email:drobledo@tamps.cinvestav.mx

I. INTRODUCCIÓN

El objetivo de la practica se presenta el problema del reconocimiento y segmentación de texturas en una imagen. Se presenta una tecnica basada en funciones de gabor combinado con el S método de clasificación no supervisada mediante K-medias.

Una imagen está conformada por un arreglo bidimensional de pixeles, y el objetivo es asignarles a cada uno una etiqueta que indique la región a la cual pertenece; una región se define como el conjunto de pixeles(generalmente conectados) que comparten una etiqueta asignada.

Una Textura se puede definir como la estructura física característica de un objeto de acuerdo a su tamaño, forma, arreglo y proporciones de sus partes contribuyentes. **Fig. 1**

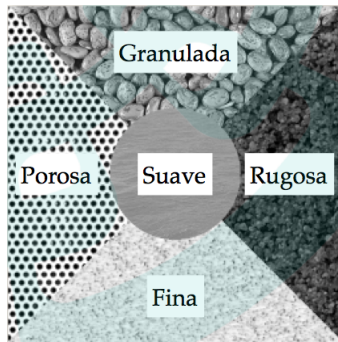


Fig. 1. Ejemplo de texturas en una imagen

El proceso de segmentación de imágenes texturadas, consiste en realizar una partición de la imagen de entrada, en un conjunto de regiones disjuntas, haciendo que cada región sea homogénea con respecto a la característica de la textura.

22 Marzo, 2014

II. DESARROLLO

Funciones de Gabor

Las células de la corteza visual del humano responde a estímulos con diferentes frecuencias, orientaciones y direcciones, por lo que se pueden interpretar como filtros que seleccionan ciertas bandas de frecuencia, orientación y

dirección.

El comportamiento de estas células han sido modeladas a través de las funciones de Gabor, debido a que estas funciones pueden ser sintonizadas a un ancho de banda, frecuencia y orientación específicos.

El función de Gabor es un filtro lineal cuya respuesta de impulso es una función sinusoidal multiplicada por una función gaussiana.

En el dominio de la frecuencia, la función de Gabor se define como una función Gaussiana desplazada a partir del origen, a una frecuencia radial U_0 y una orientación θ especificadas, y se define como:

$$H(u, v) = \exp\left(-\frac{1}{2}\left[\frac{(u' - u_0)^2}{\sigma_u^2} + \frac{v'^2}{\sigma_v^2}\right]\right) + \exp\left(-\frac{1}{2}\left[\frac{(u' + u_0)^2}{\sigma_u^2} + \frac{v'^2}{\sigma_v^2}\right]\right)$$

Donde :

$$\begin{aligned} u' &= u \cos(\theta) + v \sin(\theta) \\ v' &= v \cos(\theta) - u \sin(\theta) \\ \sigma_u &= \frac{1}{2\pi\sigma_x}, \sigma_v = \frac{1}{2\pi\sigma_y} \end{aligned}$$

Algoritmo de segmentación de texturas

Malik y Perona propusieron un algoritmo de segmentación de texturas no supervisado inspirado en el filtrado multi-canal, el cual consta de 3 pasos básicos:

- 1) Descomposición multi-canal de la imagen de entrada usando filtros de Gabor.
- 2) Extracción de características.
- 3) Agrupamiento.

Implementación

La implementación de de este método fue mediante la realización de un script en MatLab, el cual recibe como parámetro de entrada una imagen con diferentes texturas a segmentar **Fig 2**.

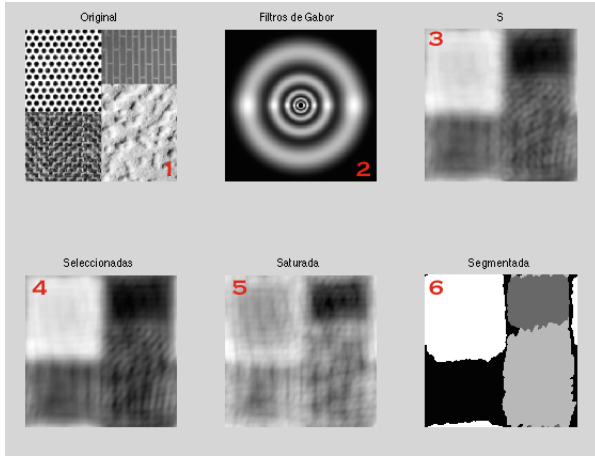


Fig. 2. Interfaz de la aplicación

Donde:

- 1) Imagen Original.
- 2) Muestra el progreso de como se generan los filtros de Gabor.
- 3) La suma de todas las imágenes filtradas .
- 4) Selección de filtros.
- 5) Saturación de imágenes.
- 6) Imagen Segmentada.

Algoritmo 1 Segmentación de texturas con filtros de Gabor

Entrada: Imagen con texturas

Salida: Imagen segmentada.

- 1: *Selección parametros()*;
 - 2: *Descomposicion multicanal()*;
 - 3: *Selección filtros()*;
 - 4: *Selección características()*;
 - 5: *Agrupamiento()*;
-

Selección de parámetros

La imagen de entrada se filtra con un banco de filtros de Gabor, los cuales poseen diferentes escalas, orientaciones y frecuencias. Las orientaciones θ se determinan en el rango $[0, \pi]$ con intervalos constantes.

Grados	0	10	15	20	30	45	90	135	180
Radianes	0	$\frac{\pi}{18}$	$\frac{\pi}{12}$	$\frac{\pi}{9}$	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π

Descomposición multi-canal.

La descomposición multi-canal consiste en filtrar la imagen de entrada (en el dominio de la frecuencia) usando cada filtro de Gabor, para generar un conjunto de imágenes con la respuesta de cada filtro.

Una forma de homogeneizar cada región es aplicar un suavizado Gaussiano a cada imagen filtrada, cuya varianza esta

directamente ligada a la de su respectivo filtro.

$$G(u, v) = \exp \left[-\frac{1}{2} \left(\frac{u^2}{\sigma^2} + \frac{v^2}{\sigma^2} \right) \right]$$

$$\sigma = k \frac{N_c}{u_o}$$

Selección de filtros

Para reducir el costo computacional se debe seleccionar un subconjunto de filtros que se aproxime a las variaciones de intensidad de la suma de todas. El error entre ambos conjunto puede medirse como:

$$SSE = \sum_{x,y} [\hat{s}(x, y) - s(x, y)]^2$$

Sea $s(x, y)$ la suma de todas las imágenes filtradas y $\hat{s}(x, y)$ la suma parcial de un subconjunto de imágenes filtradas.

La fracción de variación de intensidad se puede medir con el coeficiente de determinación.

$$R = 1 - \frac{SSE}{SSTOT}$$

$$SSTOT = \sum_{x,y} s(x, y)^2$$

Extracción de características

La extracción de características se lleva a cabo mediante la saturación del subconjunto de imágenes. **Fig. 9**

$$\tanh(\alpha * t) = \frac{1 - e^{-2\alpha * t}}{1 + e^{-2\alpha * t}}$$

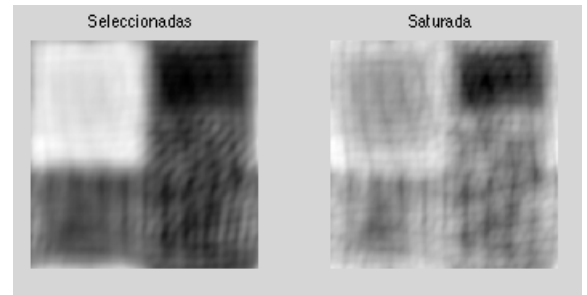


Fig. 3. Saturación

Agrupamiento

Como paso final consiste en agrupar los píxeles en k grupos predefinidos que representan las diferentes texturas en la imagen.

El agrupamiento se realiza utilizando el algoritmo de k-means utilizando como información las imagenes saturadas.

Salida del programa

Los resultados arrojados por el programa se pueden apreciar a continuación.

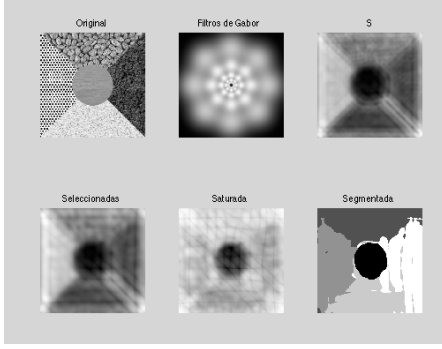


Fig. 4. Instancia 1 $k=5$, $b=1$, $\theta=45$, $\alpha=0.25$, $T=1$

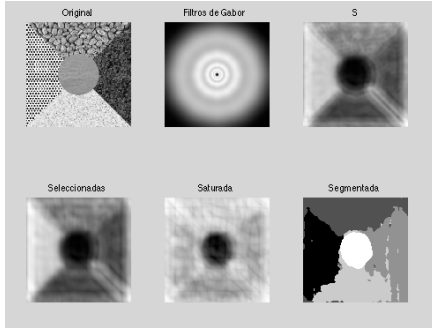


Fig. 5. Instancia 1 $k=5$, $b=1$, $\theta=30$, $\alpha=0.25$, $T=1$

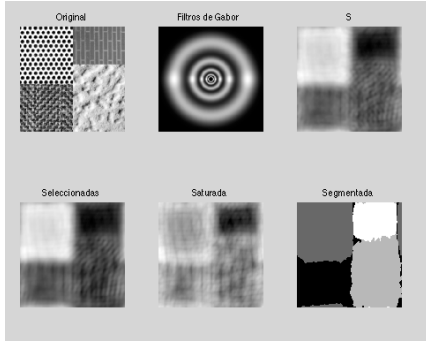


Fig. 6. Instancia 2 $k=4$, $b=0.5$, $\theta=10$, $\alpha=0.25$, $T=1$

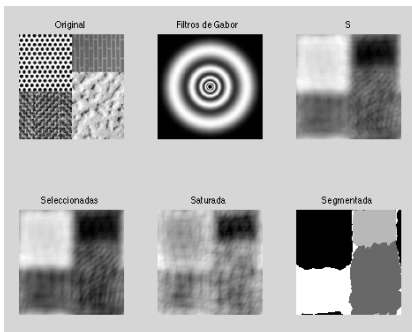


Fig. 7. Instancia 2 $k=4$, $b=0.5$, $\theta=15$, $\alpha=0.25$, $T=1$

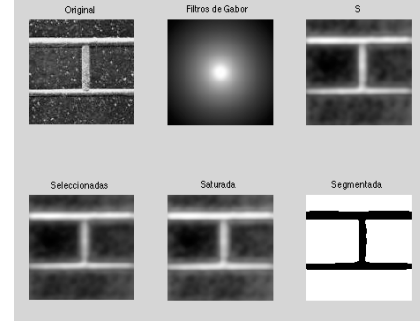


Fig. 8. Instancia 3 $k=2$, $b=3$, $\theta=35$, $\alpha=0.01$, $T=1$

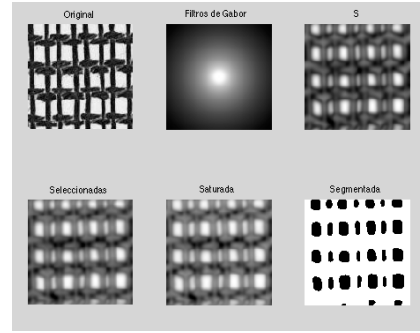


Fig. 9. Instancia 4 $k=2$, $b=2$, $\theta=20$, $\alpha=0.01$, $T=1$

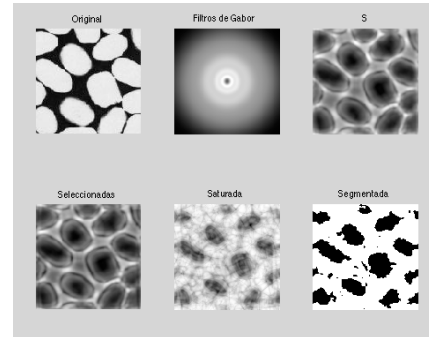


Fig. 10. Instancia 5 $k=2$, $b=1.3$, $\theta=20$, $\alpha=0.25$, $T=3$

III. CONCLUSIONES

Como se puede observar en las instancias de pruebas, la segmentación no siempre es la correcta, esto se debe al funcionamiento del algoritmo K-means, una solución sería usar algún otro algoritmo de inteligencia computacional para poder agrupar los píxeles de una manera óptima.