

DCC Universidad de Chile  
CC5508 Procesamiento y Análisis de Imágenes  
Tarea Nro 5: Mejoramiento de Huellas Dactilares

Profs. José M. Saavedra & Mauricio Cerda  
Ayudante: Camila Álvarez

## 1. Objetivo

Entender e implementar técnicas de mejoramiento de huellas dactilares.

## 2. Descripción

La tarea consiste en implementar un programa en Python que sea capaz de mejorar una imagen de huella dactilar usando filtros de Gabor. Un ejemplo del resultado esperado se muestran en la Figura 1.



Figura 1: Izquierda: Imagen de huella dactilar. Derecha: Imagen mejorada.

## 2.1. Filtro de Gabor

Recordemos que un filtro de Gabor es un filtro orientado que se representa por una función *coseno* modulada por una gaussiana y se define como:

$$G(x, y; \theta, \lambda, \sigma_x, \sigma_y) = \exp \left[ -\frac{1}{2} \left( \frac{x_\theta^2}{\sigma_x} + \frac{y_\theta^2}{\sigma_y} \right) \right] \cos \left( 2\pi \frac{1}{\lambda} x_\theta \right) \quad (1)$$

donde:

- $x, y$ : es una coordenada en el filtro. Si definimos un filtro con radios  $rx, ry$ , entonces  $x \in [-rx, rx]$  e  $y \in [-ry, ry]$ . Recuerde, además, que el tamaño de un filtro gaussiano depende de su desviación estándar correspondiente.
- $\theta$ : orientación del filtro.
- $\lambda$ : *bandwidth* de la función coseno, es el inverso de la frecuencia.
- $\sigma_x$ : desviación estándar con respecto a las coordenadas  $x$ 's.
- $\sigma_y$ : desviación estándar con respecto a las coordenadas  $y$ 's.
- $x_\theta, y_\theta$ : son las coordenadas orientadas que se definen como:

$$x_\theta = x * \cos(\theta) - y * \sin(\theta) \quad (2)$$

$$y_\theta = y * \sin(\theta) + x * \cos(\theta) \quad (3)$$

Un muestra de diversos filtros de Gabor para diferentes valores de  $\theta$  y  $\lambda$  se muestran en la Figura 2.

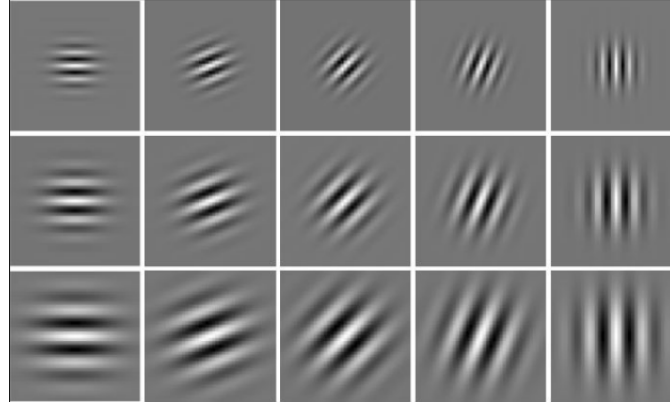


Figura 2: Ejemplo de filtros de Gabor calculados para diferentes orientaciones y frecuencias

**El mejoramiento de una huella dactilar consiste en aplicar un proceso similar a la convolución, pero en el que el filtro se va modificando para cada punto de acuerdo con la orientación y frecuencia de la huella en tal punto.**

## 2.2. Estimación de Orientaciones

La orientación de los puntos de las huellas siguen un comportamiento local, es decir varios puntos en una misma localidad o región siguen las misma orientación. Por lo tanto, podemos diseñar un método que estime localmente un mapa de orientaciones de una huella. Un buen algoritmo para realizar esta tarea se describe a continuación:

**Data:** I: imagen de huella, W: #celdas por fila y columna

**Result:** m\_orienta: mapa de orientaciones

dividir la imagen en WxW celdas, cada celda corresponde a una localidad. Además cada celda tiene una posición  $(i_c, j_c)$ ;

Gx=calcular gradiente en x de I (usar Sobel);

Gy=calcular gradiente en y de I(usar Sobel);

Mag= $\sqrt{Gx^2 + Gy^2}$ ;

A=atan2(Gy,Gx) //estimación de ángulos usando los gradientes;

m\_sines=crear matriz de WxW e inicializar todo en 0;

m\_cosines=crear matriz de WxW e inicializar todo en 0;

**for**  $(i, j)$  de I **do**

    determinar la celda  $(i_c, j_c)$  a la que pertenece  $(i, j)$  ;

$\alpha = A(i, j)$ ;

    mag=Mag(i,j);

    m\_sines = m\_sines +  $2 * \cos(\alpha) \sin(\alpha) * mag$ ;

    m\_cosines = m\_cosines +  $\cos^2(\alpha) - \sin^2(\alpha) * mag$

**end**

m\_orienta=atan2(m\_sines, m\_cosines);

m\_orienta=m\_orienta\*0.5; //pues habíamos usado ángulos dobles ;

m\_orienta=m\_orienta +  $\pi * 0,5$  //para obtener la orientación de los bordes

**Algoritmo 1:** Algoritmo para estimar orientación local

Para efectos de visualización es posible graficar las orientaciones por celda como se muestra en la Figura 3.

## 2.3. Estimación de Frecuencias

En esta tarea pueden suponer que la frecuencia es fija para una cada huella a procesar.

## 2.4. Resultados del mejoramiento

Usando el mapa de orientaciones y definiendo una frecuencia fija, es posible procesar una huella dactilar como se ha descrito anteriormente y obtener resultados como los mostrados en Figura 4

## 3. Especificaciones

1. Implementar un programa en Python que permita mejorar una huella dactilar siguiendo el algoritmo antes mencionado.



Figura 3: Izquierda: Imagen de de huella dactilar. Derecha: Mapa de orientaciones locales.

2. El programa debe contener:
  - a) Función para calcular filtro de Gabor, según Ecuación 1. Es recomendable, que represente gráficamente el filtro obtenido con su función. ¿Es el esperado?
  - b) Función para estimar orientaciones, según Algoritmo 1.
  - c) Función para mejorar una huella dactilar, que haga uso de las dos funciones anteriores.
3. Evaluar el programa con imágenes de huellas adjuntas.
4. Presentar un informe tipo paper.

## 4. Informe

Se debe presentar un informe en formato tipo *paper*, el que debe incluir:

1. Introducción
2. Desarrollo. Aquí describa todo lo referente a cómo se abordó el problema y presente una descripción detallada de sus programas.
3. Presentación y Análisis de Resultados.
4. Conclusiones

## 5. Entrega

- La tarea debe ser entregada hasta el 27 de junio de 2017 a través de u-cursos.



Figura 4: Algunos resultados del procesamiento de huellas dactilares para  $\lambda = 10$  y  $\sigma_x = \sigma_y = 10$

- No se aceptan atrasos.
- La tarea se desarrolla en forma individual.
- Solamente se evaluarán tareas con informe.