

# IEE239 - Procesamiento de Señales e Imágenes Digitales

## Laboratorio 05 - Guía Práctica

### Segundo Semestre 2017

Martes, 14 de noviembre del 2017

#### Horario 08M1

- Duración: 2 horas, 30 minutos.
- Está permitido el uso de material adicional.
- La evaluación es **estrictamente** personal.
- **Está terminantemente prohibido copiar código externo (ejemplos de clase, material en línea, etc.)**

#### 1. (3 puntos) **Supresión del patrón de Moiré presentado en radiografías.**

Las radiografías realizadas en zonas donde el tejido es mayor a 10 cm de espesor se adquieren típicamente a través de una rejilla Bucky, un patrón fino de tiras alternas de plomo y plástico que suprimen la radiación dispersa y por lo tanto mejoran el contraste de la imagen. Desafortunadamente, cuando se muestrea la imagen, pueden producirse patrones de Moiré. En este problema se estudia cómo reducir los patrones de Moiré para preservar adecuadamente las características salientes para el diagnóstico.

- Leer la imagen '**radiografia1.jpg**'<sup>1</sup>. Aplicar un filtro mediana de orden  $N \times N$  (usar **medfilt2()**). Ajustar el tamaño de  $N$  de forma que se logre remover el patrón de Moiré lo máximo posible (probar con  $N = 3, 5, 7$ ). Comentar sobre la calidad de la imagen luego del filtrado y graficar la imagen original y la imagen filtrada en una misma ventana usando **imshow()**.
- Calcular la DFT 2D sobre la imagen filtrada y graficar el espectro de magnitud y fase, usando **fft()**, **fftshift()**, **unwrap()**, **angle()** y **abs()**. Identificar las regiones correspondientes al ruido y almacenar en variables sus coordenadas centrales  $(x, y)$  (ver Figura 1a). **Nota:** Dado que se trata de señales reales, los centros de ruido más cercanos al origen son simétricos entre sí (misma posición, solo con cambio de signo). Lo mismo sucede para el ruido más alejado del origen.
- Diseñar un filtro rechazabanda para eliminar cada componente de ruido en la imagen, el cual está descrito por:

$$H(u, v) = 1 - \exp\left(-\frac{(u - u_0)^2 + (v - v_0)^2}{\sigma^2}\right), \quad (1)$$

donde  $(u_0, v_0)$  es la coordenada central del ruido. Para obtener los pares ordenados  $(u, v)$  usar **meshgrid()**.

---

<sup>1</sup>La imagen está almacenada en la carpeta 'laboratorio/lab05/08m1/Guia/'



Figura 1: Procedimiento para remover el patrón de Moiré en la imagen. (a) Espectro de magnitud correspondiente a la imagen original filtrada, donde se señala la localización del ruido (círculos amarillos). (b) Filtro rechazabanda diseñado para eliminar las componentes de ruido.

Para los picos más cercanos al centro del espectro de magnitud, usar  $\sigma = 10$ , mientras que para los más lejanos usar  $\sigma = 20$ . Graficar la máscara correspondiente al filtro rechazabanda a aplicar sobre la imagen, la cual debe ser similar a la mostrada en la Figura 1b.

- d. Filtrar aplicando producto en frecuencia. Luego, calcular la transformada inversa de la multiplicación con `ifft2()` e `ifftshift2()`. Extraer la parte real y normalizar el resultado. Posteriormente, ecualizar el histograma usando `histeq()` y graficar la imagen restaurada. ¿Se logró remover el patrón totalmente?. Explicar por qué se degrada el fondo negro de la imagen en base al filtro rechazabanda aplicado.

2. (4 puntos) **Identificación y segmentación de personas a partir de una imagen aérea.** Mediante el uso de una cámara aérea se monitorea la cantidad de transeúntes durante el día por una determinada avenida. A partir del video, se extrae una imagen, la cual es mostrada en la Figura 2a.

- a. En primer lugar, leer la imagen '`camera.jpg`'<sup>2</sup>. Transformar la imagen a escala de grises usando `rgb2gray()` y graficar el resultado.
- b. Calcular la magnitud de la gradiente. Para ello, crear un filtro sobel usando `fspecial()` y aplicarlo sobre la imagen en escala de grises usando `conv2()` (filtro en dirección  $x$ ), obteniendo  $I_x \triangleq \frac{d}{dx}I(x, y)$ . A continuación, usar la misma máscara para obtener  $I_y \triangleq \frac{d}{dy}I(x, y)$ . Luego, calcular la magnitud de la gradiente según (2) y normalizarla. También calcular la fase de la gradiente, y graficar ambos en una misma ventana.

$$|\nabla I| = \sqrt{I_x^2 + I_y^2},$$

$$\angle(\nabla I) = \arctan\left(\frac{I_y}{I_x}\right), \quad (2)$$

- c. Realizar una transformación sobre la magnitud de la gradiente:

$$|\widehat{\nabla I(x, y)}| = \begin{cases} |\nabla I|/5 & \text{si } |\nabla I| < 0,1 \\ |\nabla I|/2 & \text{si } |\nabla I| \geq 0,1 \end{cases} \quad (3)$$

Luego, graficar el histograma correspondiente empleando 100 bins (usar `hist()`). Por inspección, hallar un valor de umbral adecuado para binarizar la imagen. Indicar dicho valor en los comentarios.

---

<sup>2</sup>La imagen está almacenada en la carpeta '`laboratorio/lab05/08m1/Guia/`'

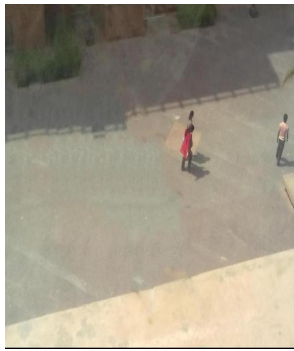
- d. En base al umbral hallado en el ítem anterior, binarizar la imagen y graficarla. Para mejorar la segmentación, se requiere conectar los bordes inconexos. Para esto, hacer

$$B = \text{imclose}(A, \text{ones}(10));$$

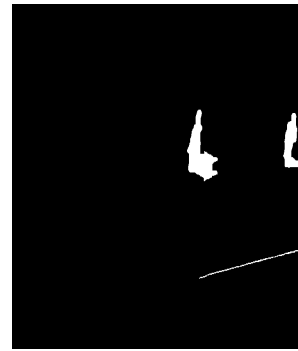
donde A es la imagen binaria y B la imagen resultante. El término 'ones(10)' es una matriz de 1's que recorre la imagen, en primera instancia 'expandiendo' la imagen (dilatación) y luego opera sobre la imagen dilatada, 'contrayéndola' (erosión).

Posteriormente, rellenar espacios vacíos usando **imfill(..., 'holes')**. A continuación, usar **bwlabel()** (la cual le asigna un índice distinto a cada objeto dentro de la imagen binaria) e identificar el índice correspondiente a las personas en la zona central y la persona en la izquierda. Con esto, generar una máscara binaria donde sólo aparezcan las personas.

- e. Refinar la segmentación: recortar 4 píxeles de cada lado de la imagen segmentada y para eliminar pequeñas regiones usar **bwareaopen(..., 500)** (que elimina pequeños objetos inconexos). Por último, mostrar la imagen original y la segmentada en un mismo gráfico. Se espera un resultado al presentado en la Figura 2b.



(a)



(b)

Figura 2: (a) Imagen de entrada cuyo ruido sigue un patrón. (b) Imagen sin ruido filtrada en frecuencia.

3. (4 puntos) Se va a trabajar con la imagen de una pirámide, sobre la cual se aplicarán filtros para detección de bordes.
  - a. Leer la imagen '**piramide.jpg**'<sup>3</sup> y convertirla a escala de grises usando **rgb2gray()**. Calcular el espectro de magnitud y de fase de la imagen y graficarlos. Usar **fft()**, **fftshift()**, **unwrap()**, **angle()**, **abs()** y **log()** para mostrar el espectro.
  - b. A continuación, calcular la conjugada de la DFT 2D hallada en el ítem anterior. Luego, proceder a calcular la IDFT 2D usando **ifft()**, **ifftshift()**. ¿Qué sucedió con la imagen de entrada? Explicar por qué la imagen presenta tal transformación. Se espera obtener una imagen como la mostrada en la Figura 3.
  - c. Añadir ruido gaussiano a la imagen original en escala de grises usando **imnoise()**, con varianza 0.05. A continuación, usar **fspecial()** para generar un filtro laplaciano, con  $\alpha = 0,4$ . Luego, filtrar la imagen con ruido usando el filtro laplaciano en el dominio de la frecuencia (**fft2()**, **fftshift()**, **ifft()**, **ifftshift()** y **real()**). ¿Se puede visualizar correctamente los bordes de la imagen? En caso no sea posible, identificar el factor que lo impide.

<sup>3</sup>La imagen está almacenada en la carpeta 'laboratorio/lab05/08m1/Guia/'



(a)



(b)

Figura 3: Transformación de la DFT 2D de una imagen. (a) Imagen de entrada, (b) Imagen reconstruida tras hallar la conjugada de la DFT 2D.

- d. Ahora, se propone implementar un filtro Laplaciano de Gaussiano para la detección de bordes en la imagen, que viene dado por

$$LoG(x, y) = \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right)$$

Verificar el efecto del filtro LoG para tal finalidad, trabajando con distintos valores de  $\sigma = 0.5, 1.4, 3$  y  $5$  y tamaño  $3\sigma + 1$  y posteriormente umbralizando el filtrado usando **im2bw()**. ¿Cuál es el efecto de incrementar el valor de  $\sigma$  en la detección de bordes? ¿Por qué este filtro permite distinguir los bordes mejor que el propuesto en el ítem anterior?