

# IEE239 - Procesamiento de Señales e Imágenes Digitales

## Laboratorio 5 - Guia Práctica

### Primer Semestre 2018

Martes, 19 de junio del 2018

#### Horario 08m2

- 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. (3.5 puntos) Se requiere dar color de forma automática a imágenes de *manga* (cómic japoneses). Para ello, un primer paso a realizar es la eliminación de artefactos (interferencia, ruido) en la imagen y la eliminación de *screeentones* o sombreados de textura. En esta pregunta se propondrá un método básico para realizar este proceso.
  - a. (1 punto) Cargar la imagen **alchemist.png**<sup>1</sup>, la cual fue contaminada por un tono sinusoidal debido a un error en el escaneo. En primer lugar, hallar y graficar la magnitud de la transformada de Fourier de la imagen considerando que el tamaño de la DFT es el doble de la imagen. A partir del espectro de magnitud, determinar la posición  $(u, v)$  del tono de interferencia con respecto del origen de coordenadas.  
e identificar visualmente el tono de interferencia.

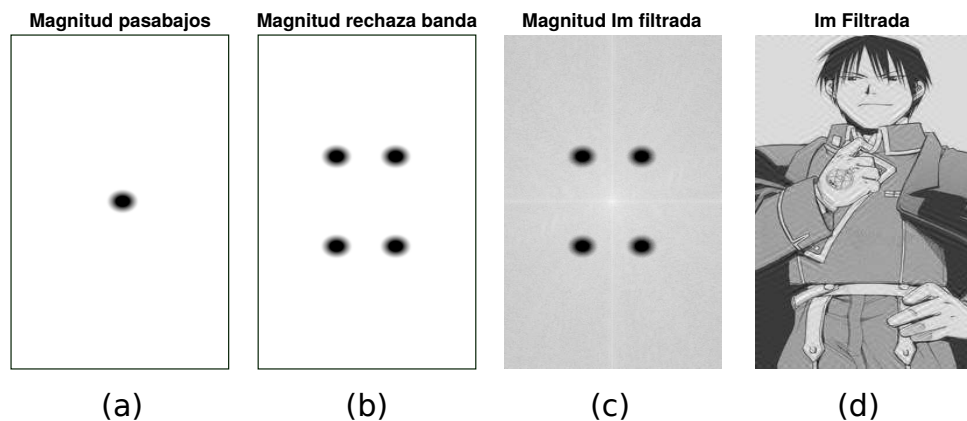


Figura 1: Imágenes referenciales Pregunta 1b. (a) Espectro magnitud filtro pasaaltos Butterworth, (b) Espectro magnitud filtro rechazabanda, (c) Espectro de magnitud de la imagen filtrada y (d) Imagen filtrada

<sup>1</sup>El archivo está ubicado en laboratorio/lab05/08m2/Guia.

- b. (1 punto) Considerar como prototipo el filtro pasabajos Butterworth definido en frecuencia como:

$$H_{HP}(u, v) = 1 - \frac{1}{1 + \left( \frac{(u^2 + v^2)^{\frac{1}{2}}}{D_0} \right)^{2n}},$$

donde  $D_0 = 27$  es el ancho de banda,  $n = 21$  es el orden y el tamaño de la DFT es el doble de la imagen. Diseñar un filtro rechaza banda que elimine la interferencia presente en la imagen original. Para ello, deberá desplazar el filtro pasaaltos en las posiciones halladas en el inciso anterior utilizando el comando `circshift()`<sup>2</sup>. Realizar el filtrado en frecuencia y graficar el espectro resultante. Adicionalmente, transformar el espectro filtrado al espacio de muestras y graficar la imagen filtrada denotada como  $I_1(x, y)$ .

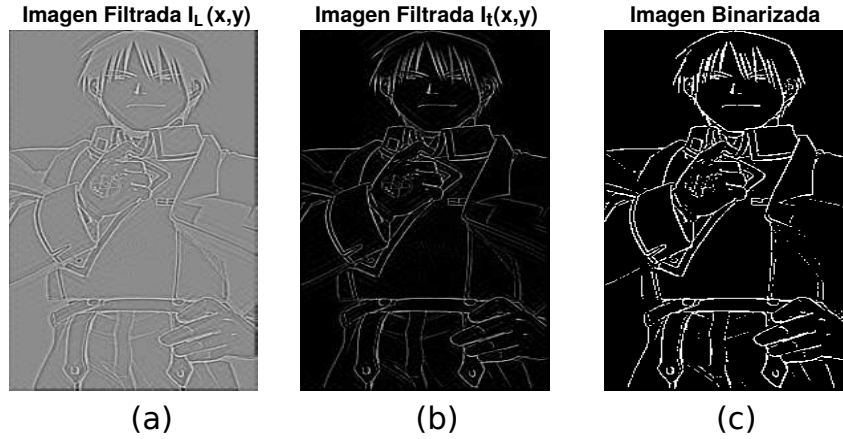


Figura 2: Imágenes referenciales (a) pregunta 1c y (b), (c) pregunta 1d.

- c. (1 punto) Definir un filtro basado en el operador Laplaciano de Gaussiano (LoG):

$$\text{LoG}(x, y) = \frac{1}{\pi\sigma^4} \left( \frac{x^2 + y^2}{2\sigma^2} - 1 \right) e^{-\frac{x^2 + y^2}{2\sigma^2}},$$

de dimensiones  $9 \times 9$  y  $\sigma = 0.8$ . Obtener la imagen filtrada  $I_L(x, y) = I_1(x, y) * \text{LoG}(x, y)$  usando el comando `conv2()` con el parametro 'same'. Graficar  $I_L(x, y)$  e  $I_1(x, y)$  en una misma ventana y comentar si el filtro tiene alguna predilección a bordes con cierta orientación.

- d. (0.5 puntos) Anular los elementos negativos obtenidos, es decir, obtener

$$I_t(x, y) = \begin{cases} I_L(x, y) & , \text{ si } I_L(x, y) \geq 0, \\ 0 & , \text{ otros casos .} \end{cases}$$

Finalmente, a partir del análisis del histograma de  $I_t(x, y)$  utilizando el comando `hist()`, seleccionar de forma manual un umbral  $\tau$  que permita obtener una imagen binarizada que contenga las líneas de la imagen de *manga* sin los *screentones*. Comentar sobre la calidad de los bordes obtenidos.

<sup>2</sup>`circshift(X,[K J])` realiza el desplazamiento circularmente de los los elementos de X en las posiciones K y J (escalares enteros) a lo largo de la primera y segunda dimensión, respectivamente. Por ejemplo, si K es positivo, los valores de X se desplazan hacia abajo. Si es negativo, los valores de X se desplazan hacia arriba. Si J es positivo, los valores de X se desplazan hacia derecha. Si J es negativo, los valores de X se desplazan hacia izquierda.

2. (3.5 puntos) La segmentación de los objetos móviles en un video es un primer paso esencial en aplicaciones de seguridad, de robótica o de análisis de comportamiento y acciones. En el archivo `bebop.avi`<sup>3</sup> se provee un video de 10 de frames. Se procederá a segmentar el objeto móvil de esta secuencia y calcular su velocidad.

- a. (0.5 puntos) Usar la función adjunta `video2frames(nombre)`, donde `nombre` es el nombre del archivo de vídeo. Esta función devuelve un arreglo de 4 dimensiones  $V(x, y, c, m)$  donde los ejes  $x, y, c$  y  $m$  corresponden a filas, columnas, canales y número de frame, respectivamente. Para hallar los objetos móviles primero se debe realizar una estimación del fondo  $B(x, y, c)$ . Calcular esta estimación mediante:

$$B(x, y, c) = \text{median}(V(x, y, c, :))$$

- b. (0.5 puntos) Hallar la diferencia de cada cuadro con la estimación del fondo para hallar los objetos móviles en un arreglo  $S(x, y, c, n)$  definido por:

$$S(x, y, c, n) = V(x, y, c, n) - B(x, y, c)$$

Usando `imshowpair` con la opción `montage` mostrar en una misma ventana el primer y último frame de  $S$ .

- c. (1 punto) Hallar y graficar el histograma del canal R del primer frame de  $S(x, y, c, n)$  utilizando el comando `hist()`. El histograma deberá estar dominado por un pico en cero (correspondiente al fondo que se ha eliminado). Determinar un umbral  $\tau$  adecuado y definir una binarización  $M(x, y, n)$  para cada cuadro de  $S$ :

$$M(x, y, n) = \begin{cases} 1, & \text{si } S(x, y, 1, n) > \tau \\ 0, & \text{otros casos.} \end{cases}$$

Usando `imshowpair` con la opción `montage` mostrar en una misma ventana el tercer y octavo frame de  $M$ .

- d. (1 punto) Para cada frame de  $M(x, y, n)$ , usar el comando `stats = regionprops()` para hallar el centroide del objeto móvil. En la variable `stats.Centroid` se devuelve un vector correspondiente al centroide de la imagen binaria en el formato  $(x, y)$ . Deberá obtener una serie  $C_n, n \in \{1, \dots, 10\}$  de 10 centroides (uno por cada frame). Se pide hallar la velocidad en  $x$ . La velocidad en píxeles por cuadro  $u_x$  se puede calcular como:

$$u_x = \frac{1}{9} \sum_{n=2}^{10} C_n^x - C_{n-1}^x,$$

donde  $C_n^x$  es la componente  $x$  del  $n$ -ésimo centroide. Finalmente, si se sabe que cada pixel de la imagen representa una región de  $5 \times 5$  metros en la vida real y el video se tomó con una tasa de cuadros de 2 cuadros por segundo, convertir  $u_x$  a una velocidad física en m/s.

3. (3 puntos) En esta pregunta se hallará mediante segmentación de bordes y transformada de Hough el punto de fuga de una imagen. Por la perspectiva, en una imagen las líneas que son paralelas en la realidad parecen converger en un punto denominado “punto de fuga”. El punto de fuga define las características de perspectiva en el dibujo y fotografía.

---

<sup>3</sup>El archivo está ubicado en laboratorio/lab05/08M2/Guia.

- a. (1 puntos) Cargar la imagen `yourname.jpeg` y definir la matriz  $I(x, y)$  convirtiéndola a escala de grises y haciendo cast a double y normalizándola. Hallar las aproximaciones discretas de las derivadas  $D_x(x, y)$  y  $D_y(x, y)$  a partir del producto en frecuencia entre los operadores de Sobel:

$$G_y(x, y) = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad G_x(x, y) = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{pmatrix}.$$

y la imagen normalizada. Considerar que el tamaño de las DFTs es el doble de la imagen normalizadas. Graficar en una misma ventana tanto  $D_x(x, y)$  como  $D_y(x, y)$ .

- b. (1 punto) Hallar las matrices  $M(x, y)$  y  $\theta(x, y)$  correspondientes a la magnitud y ángulo del gradiente. Para este último utilizar el comando `atan2()`. Segmentar las líneas de interés definiendo la matriz  $T(x, y)$  como:

$$T(x, y) = \begin{cases} 1, & \text{si } M(x, y) > 1.2 \text{ y } \frac{\pi}{7} \leq \theta(x, y) \leq \frac{\pi}{2} \\ 0, & \text{otros casos.} \end{cases}$$

- c. (1 puntos) Usar el comando `[H,theta,rho] = hough(...)` para hallar la transformada de Hough  $H$  de la imagen binarizada  $T$ . Luego, utilizar la función `houghpeaks` para encontrar los dos picos dominantes de  $H$ . Finalmente usar el comando `lines = houghlines(I,theta,rho,peaks)` donde  $I$  es la imagen en escala de grises. En `lines.point1` y `lines.point2` se devuelven los dos puntos que definen las dos rectas dominantes de la imagen
- d. (0.5 pto) Graficar en un mismo eje usando `imshow`, `plot` y `hold on`, la imagen original a color y las dos rectas halladas en el inciso anterior (Deberá obtener un resultado similar al mostrado en la 3). Con la herramienta `tooltip` de Matlab hallar las coordenadas del punto de fuga de la imagen, el cual corresponde a la intersección de ambas rectas. Escribir en comentarios la expresión analítica cartesiana de las rectas .

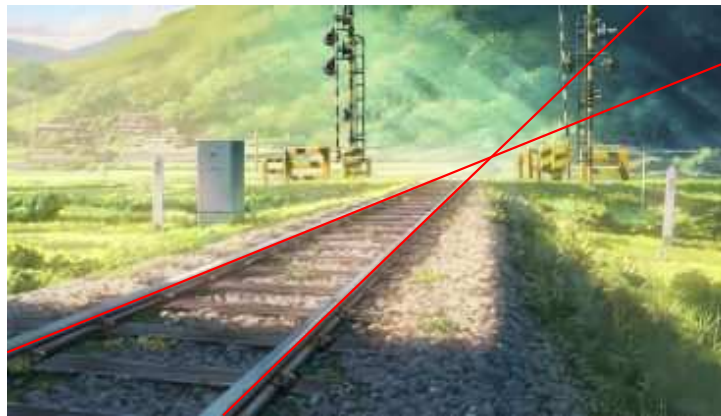


Figura 3: Ejemplo de como hallar el punto de fuga con Transformada de Hough.