

---

## IEE239 - PROCESAMIENTO DE SEÑALES E IMÁGENES DIGITALES

### LABORATORIO 04 - GUÍA PRÁCTICA

LUNES, 24 DE OCTUBRE DEL 2016

Horario: 07M1.

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.).**

---

### Pregunta 1 (4 puntos)

Se desea detectar las coordenadas de los **bounding boxes** de la chapa y etiqueta de una botella en una imagen a color (Ver figura 1). Además se cuenta con una máscara binaria con la ubicación de la botella.

Imagen original coke.png



Imagen binaria usando solo el color rojo

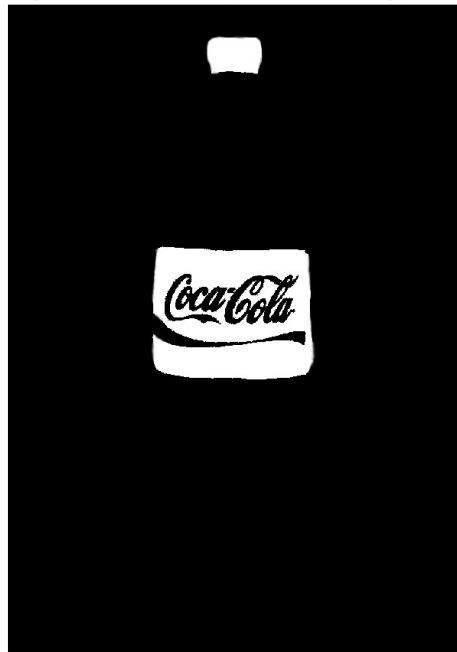


Fig. 1: Imagen de la botella a color, y de Imagen Binarizada usando solo el color rojo

- Leer la imagen **coke.png**, la cual es una imagen a color (RGB) de una botella (cada capa de la matriz de  $[M \times N \times 3]$  ( $M \times N$  es el tamaño de la imagen) representa un color), usando la función `imread()`. Cargar la máscara binaria `mask_coke.mat` y mostrar tanto la imagen normalizada como máscara binaria correctamente rotulados en una sola figura. (Usar `subplot()`) (**Nota: Para acceder a una capa específica de una imagen RGB se hace un direccionamiento de la siguiente forma  $\text{Capa}_x = I.\text{rgb}(:, :, x)$ , donde  $x = \{1, 2, 3\} \sim \{R, G, B\}$** )

- 
- b. Hallar los histogramas normalizados de cada color tomando en consideración la máscara binaria, es decir, se considerará solo las regiones de la imagen donde la máscara binaria es igual a 1. Graficar los histogramas normalizados con las etiquetas adecuadas de cada canal de color (RGB) de la imagen en una misma figura (usar `imhist()` y `subplot()`).
  - c. A partir de los histogramas normalizados obtener los valores RGB del rojo más oscuro y el rojo más intenso de la imagen original normalizada. El rojo más oscuro es aquel que en la capa R su nivel de intensidad es cercano a 1, y los niveles de intensidad en las capas G y B son bajos. El color rojo más intenso, es aquel en el que la capa R es 1 y las capas G y B son 0. (Puede usar la herramienta `Data Cursor` para verificar los valores RGB encontrados).
  - d. Obtener una imagen binaria utilizando los umbrales de cada capa encontrados en el ítem anterior. (**Nota: Puede encontrar una imagen binaria de cada capa y realizar una multiplicación punto a punto**)
  - e. Calcular la cantidad de '1's por cada línea horizontal y por cada línea vertical de la Imagen binarizada. Mostrar en una misma figura los vectores encontrados en función de los ejes respectivos y etiquetarlos adecuadamente. Encontrar las coordenadas para la tapa y la etiqueta a partir de estas gráficas.

## Pregunta 2 (3 puntos)

Se está armando uno de los rompecabezas más complicados, el Krypt. Se desea encontrar la pieza faltante de la figura 2. Se ha logrado reducir las posibilidades a 2 piezas, `pieza1.png` y `pieza2.png`, las cuales se sabe que presentan una rotación en un ángulo  $\theta = \{\pi/3, \frac{37}{180} \cdot \pi, 0.25\pi, \pi/6\}$ , además se tiene la máscara de la imagen a encontrar en `mask.mat`. Se pide realizar lo siguiente:

- a. Encontrar el nivel de umbralización adecuado para las imágenes proporcionadas de las piezas. Usar este nivel para convertirlas a imágenes binarias. (Usar las funciones `graythresh()` y `im2bw()`)
- b. Usando la función `imrotate()`, rotar las imágenes binarizadas en los ángulos dados. Considerar interpolación bilineal en la función, además la imagen rotada debe ser del mismo tamaño que la imagen original.
- c. Una forma de verificar la similitud entre dos imágenes es calculando la norma de la diferencia entre estas ( $e = \|I_1 - I_2\|$ ). Para dos imágenes iguales la norma será igual a cero. (Usar la función `norm()`)
- d. Realizar esto para todos los valores de  $\theta$  para las dos piezas. Indicar que pieza es la que se está buscando y el ángulo en que se encontraba rotada.

## Pregunta 3 (3 puntos)

Se tiene un audio con ruido `queen_noise.wav`. Se sabe que el ruido presente es un proceso WSS, con media cero, y  $\sigma^2 = 0.4$  aditivo Gaussiano  $w[n] \sim \mathcal{N}(0, 0.4)$ . Se desea filtrar este ruido utilizando un filtro Wiener siguiendo la estructura de la figura 3. El audio que se desea estimar se encuentra en el archivo `queen.wav`

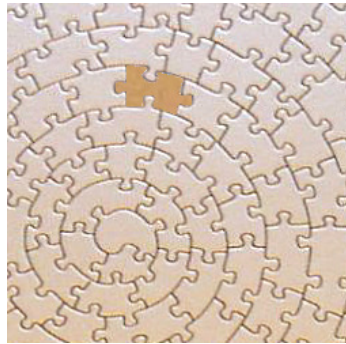


Fig. 2: Rompecabezas Krypt

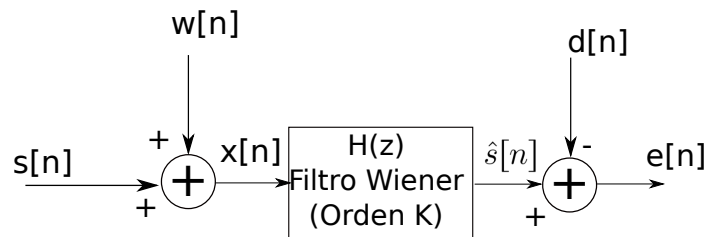


Fig. 3: Sistema propuesto

- a. Cargar el audio `queen.wav` con `audioread()`
- b. Hallar los coeficientes del filtro Wiener para un orden  $K = 50$ .
  - i. Generar la matriz de autocorrelación del audio con ruido y el vector de correlación cruzada del audio con ruido y el audio deseado para el orden solicitado. **Nota:** para la matriz de autocorrelación, utilizar primero la función `xcorr()` para autocorrelacionar la señal  $x[n]$  y a continuación usar la función `toeplitz()` para generar la matriz.
  - ii. Hallar el vector de coeficientes del filtro para el orden solicitado.
  - iii. Generar la señal estimada  $\hat{s}[n]$  utilizando la función `conv()` con los coeficientes del filtro.
  - iv. Graficar la señal original y la señal estimada  $\hat{s}[n]$  en una misma figura para las muestras  $n \in [2000, 3000]$ . Utilice `subplot()`
- c. Repetir los pasos anteriores para ordenes  $K = 25$  y  $K = 500$ .
- d. ¿Para que ordenes se obtiene una mejor estimación del audio? (Calcular el error usando la función `norm()`)