DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

# Laboratorio 6

## PROCESAMIENTO DE SEÑALES E IMÁGENES

Profesores:

- Violeta Chang C.
- Leonel E. Medina

Ayudante: Luis Corral

Alumno: John Serrano C.

# Actividades

1. Utilizar la imagen ‘Q2_2.tif’, y escribir una función que lea la imagen, reduzca el ruido y guarde la imagen filtrada con el nombre de 'Q2_2-sr.jpg'. Comparar visualmente con resultado de aplicar función *medfilt2* de Matlab.
2. Utilizar la imagen ‘luna.tif’, y escribir una función que lea la imagen, aplique el filtro de énfasis con factor 4 y guarde la imagen filtrada con el nombre de 'luna-ea.jpg'.
3. Utilizar la imagen 'circuit.tif', y escribir una función que lea la imagen, aplique el operador de Kirsch para detección de bordes y guarde la imagen filtrada con el nombre de 'circuit-kirsch.jpg'.
4. Utilizar la imagen 'coins.png' y aplicar la función de Matlab *edge* para detectar los bordes de los objetos usando el detector de Canny experimentando con los valores de los umbrales *thr1* y *thr2* hasta obtener el mejor resultado posible. Guardar la imagen resultante como 'coins-canny-*thr1*-*thr2*.jpg', donde *thr1* y *thr2* corresponden a los mejores valores de los parámetros correspondientes.

# Tarea

Entregue el resultado de la actividad 4 en formato mlx explicando la selección de los valores *thr1* y *thr2*. Se evalúa los conceptos en formato de texto, los comentarios dentro del código, la exactitud del algoritmo y la calidad de los gráficos generados. Muestre solo los valores más importantes.

# Desarrollo

Lo primero es cargar la imagen en Matlab, para así saber con que estamos trabajando:

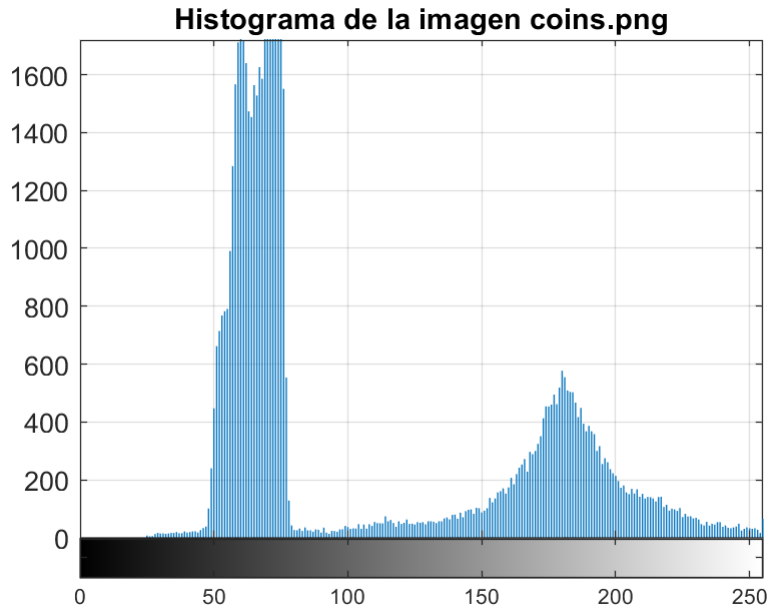
```
% Carga de la imagen coins.png

imagen = imread("coins.png"); % Se carga la imagen
figure;                       % Se inicializa una figura
imshow(imagen);               % Se muestra la imagen
title("coins.png")            % Titulo de la imagen
```


Ahora, necesitamos obtener los valores *thr1* y *thr2* para poder aplicarnos junto con la **función edge()**, usando el **detector de canny.** Para ello, vamos a crear el histograma de la imagen junto con el histograma acumulado para así obtener *thr2.*

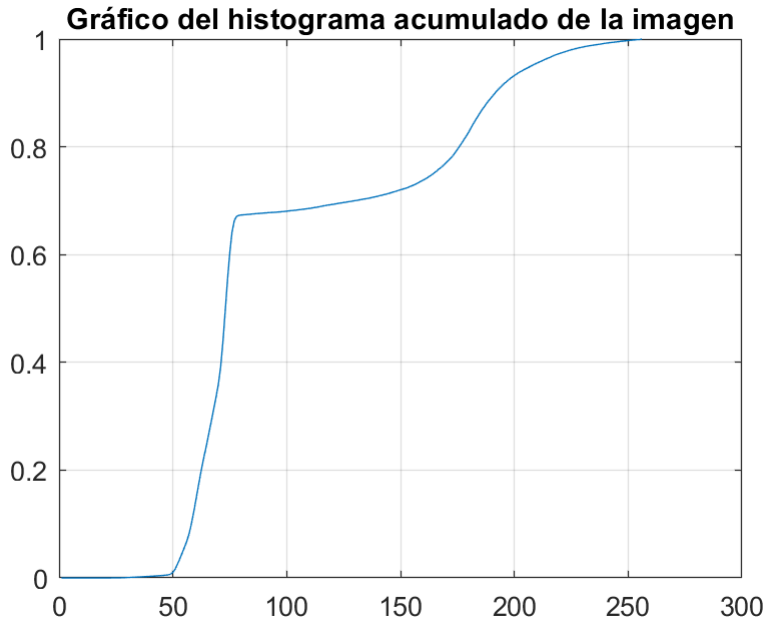
```
% Histograma de la imagen
histograma = imhist(imagen); % Se guarda el histograma de la imagen
figure;                      % Se inicializa una figura
imhist(imagen);              % Se muestra el histograma de la imagen

% Configuraciones adicionales del gráfico
title("Histograma de la imagen coins.png")
grid on
```




```
% Histograma acumuado de la imagen
histograma_acum = cumsum(histograma); % Se obtiene el histograma acumulado
% Se normaliza el histograma acumulado
histograma_acum = histograma_acum/max(histograma_acum);
% Gráfico del Histograma acumulado
figure; % Se inicializa una figura para graficar
plot(histograma_acum) % Se grafica el histograma acumulado

% Configuraciones adicionales del gráfico
title("Gráfico del histograma acumulado de la imagen")
grid on
```




Del gráfico anterior, podemos observar que *thr2* toma aproximadamente un valor correspondiente a 0,66. De la teoría, sabemos que se debe aplicar una heurística para obtener *thr1*. Aplicando la heurística:

```
thr2 = 0.66        % Valor del umbral alto
```

```
thr2 = 0.6600
```

```
thr1 = 0.66 * 0.4  % Valor del umbral bajo
```

```
thr1 = 0.2640
```

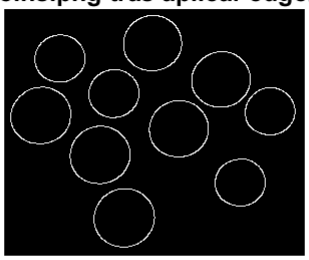
Ahora, podemos aplicar la función edge usando el detector de Canny. Como valores del Umbral de sensibilidad, vamos a utilizar los valores obtenidos anteriormente,  0.2640 y 0.66. Por lo tanto, edge va a ignorar todos los bordes cuya intensidad es menor que la del umbral más bajo (0.2640) y conserva todos los bordes cuya intensidad es mayor que la del umbral más alto (0.66).

```
% Aplicando la función edge

imagen2 = edge(imagen,"canny",[0.2640 0.66]); % Se aplica la función edge con "canny"
figure;                     % Se inicializa una figura
imshow(imagen2)             % Se muestra la imagen

% Configuraciones adicionales del gráfico
title("coins.png tras aplicar edge()");
grid on;
```

**coins.png tras aplicar edge()**



Como se puede apreciar, gracias a la función edge y los valores *thr1* y *thr2*, solamente se logra mantener los bordes de las monedas, eliminando el resto del contenido de la imagen. Ahora, vamos a guardar la imagen.

```
% Se guarda la imagen con el nombre solicitado.
imwrite(imagen2, sprintf('coins-canny-%f-%f.jpg', thr1, thr2));
```