

1.- (3 puntos) Ruido en una imagen: Describa 3 tipos de ruido que puede presentar una imagen, indicando el efecto sobre la imagen y los problemas o dificultades que pueden causar a un sistema de visión por computador.

2.- Normalización e histogramas (3 puntos)

- a) Indique la diferencia entre histograma y función de distribución de intensidad de una imagen.
- b) Dada una imagen I , cuyos píxeles pueden presentar valores de intensidad enteros en el rango 0-255, indique cuántos bytes serían necesarios, como mínimo, para codificar cada pixel.
- c) Indique la fórmula para normalizar las intensidades de la imagen I al intervalo $[0,1]$.

3.- (4 puntos) Una cámara térmica es capaz de segmentar imágenes de caras. Después de un preprocesado inicial, se obtiene una imagen I_0 en donde la intensidad de las zonas más calientes es 100 y la del fondo es 255. Las zonas correspondientes a la nariz, los ojos y la boca tienen los niveles del fondo (255).

Desarrolle un programa en pseudocódigo para realizar las siguientes operaciones:

- a) Obtener I_1 : Segmentar la imagen I_0 para separar la región de interés, correspondiente a la cara, del fondo (En I_1 , tanto la región de la cara como las regiones correspondientes a los ojos, nariz y boca deben quedar a "1" y el fondo a "0").
- b) Generar una nueva imagen I_2 en la que se muestre la imagen I_0 pero en la que los píxeles correspondientes a los ojos tomen el valor "0".

Para realizar el programa en pseudocódigo dispone de los siguientes operadores y estructuras de programación (continúa en la siguiente hoja de enunciado)



Figura 1: Imagen I_0

- a) Estructuras de control habituales (**for**, **if**, **while**, ...).
- b) **Operaciones aritméticas y lógicas** sobre variables de tipo matriz, tanto en operación matricial como elemento a elemento.
- c) Operadores de procesamiento de imagen:
- Sea **Ib** una imagen binaria
 - Etiquetado: **L = bwlabel(Ib)** devuelve en **L** una imagen de las mismas dimensiones que **Ib** y con etiquetas distintas (1,2,3,...N) en cada uno de los N blobs independientes en conectividad 8 encontrados en **Ib**. Se supondrá que el fondo tiene la etiqueta "0".
 - Rellenado de agujeros de los objetos: **Ib2 = bwfill(Ib)** Devuelve una imagen binaria de las mismas dimensiones que **Ib** y con los objetos de la imagen **Ib** y sus agujeros a "1".
 - Mostrar imagen en pantalla: **imshow(I)**
- d) Otros operadores:
- [p] = find(X)** encuentra los índices de **X** con valor distinto de cero.
 - [u] = unique(X)** devuelve los valores de **X** sin repetición.
 - Paso de índice lineal a subíndice (fila, columna) y viceversa
 - [linearInd] = sub2ind(matrix2DSize, filaInd, colInd)**
 - [filaInd, colInd] = ind2sub(siz, linearInd)**
 - N = length(v)** devuelve en **N** el número de elementos del vector **v**.
 - [f,c] = size(M)** dimensiones en filas y columnas de la matriz **M**.
 - v = M(:)** pasa una matriz N-dimensional **M** a vector unidimensional.
 - zeros(f,c)** : genera una matriz de zeros de **f** filas y **c** columnas.

* Es muy probable que con los operadores descritos se pueda resolver el problema. Sin embargo, si considera que necesita más operadores, puede utilizarlos siempre que los justifique.

**** Atención: No basta con escribir el programa en pseudo-código. Debe comentarlo para justificar las decisiones tomadas.**