1.- (3 puntos) Filtro de mediana

a) Fundamento y características de un filtro de mediana.

b) Suponiendo que disponemos de un operador llamado `median(x)`, donde x es un vector unidimensional, escriba una función en pseudocódigo, `medianFilter(I,w)`, para aplicar un filtro de mediana a una imagen. La función tendrá como parámetros de entrada la propia imagen sobre la que se desea aplicar el filtro y el tipo de ventana del filtro. Se deberá poder elegir entre una de las ventanas de la figura 1, las cuales, estando centradas en el pixel al que se desea aplicar el filtro, tienen X en los términos que se evaluarán en el cálculo de la mediana).

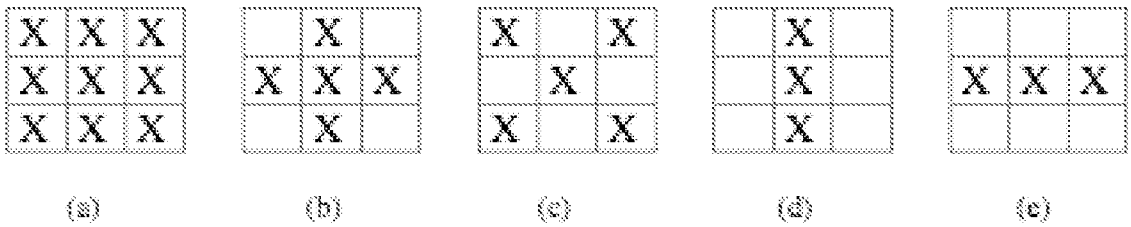


Figura 1

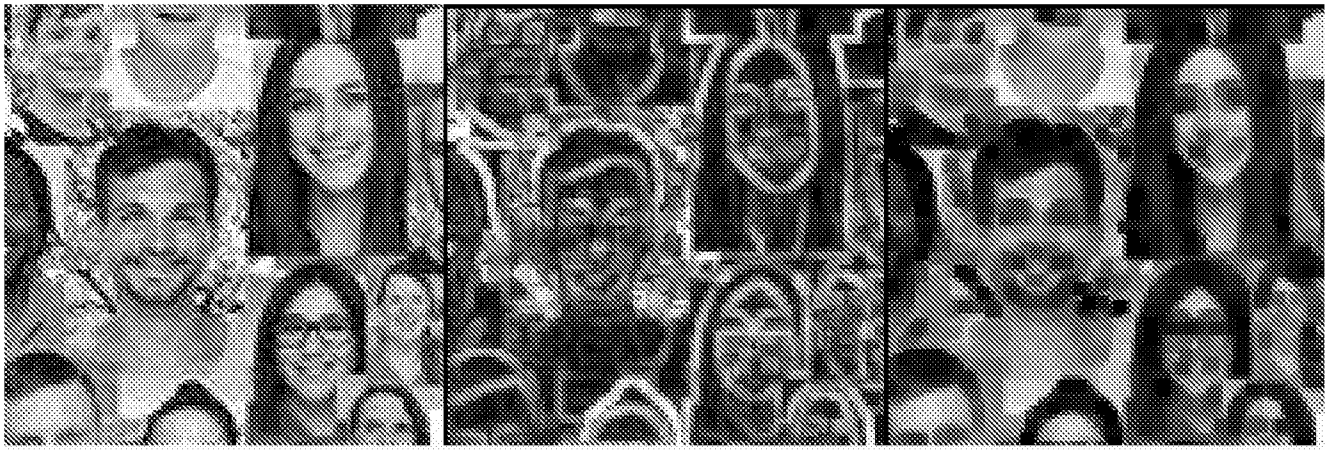
2.- Filtros min-max (3 puntos).

a) Describa el efecto de aplicar el siguiente filtro a una imagen en niveles de gris.

$$g(x,y) = fmax - fmin$$

donde fmax y fmin son los valores máximo y mínimo en una ventana de 3x3 centrada en el pixel(x,y)

b) ¿Cuál de las salidas mostradas en las figuras Out1 y Out2 cree que corresponde a aplicar este filtro sobre la imagen Iorig?. Justifique la respuesta.



Iorig **Out1** **Out2**

3.-(4 puntos) Dada la imagen de la figura 2, que contiene varias máscaras de color más oscuro que el fondo (considere: intensidad del fondo<42; 44<intensidad de máscara <180; intensidad huecos máscara >190). Desarrolle un programa en pseudocódigo para realizar las siguientes operaciones:

a) Segmentar la imagen para separar las máscaras del fondo.
b) Localizar y distinguir la máscara A con la boca más pequeña (boca de menor área)
c) Localizar la máscara más alejada de la máscara con la boca más pequeña.
d) De ésta máscara más alejada, cerrar todos los agujeros y mostrar la imagen.

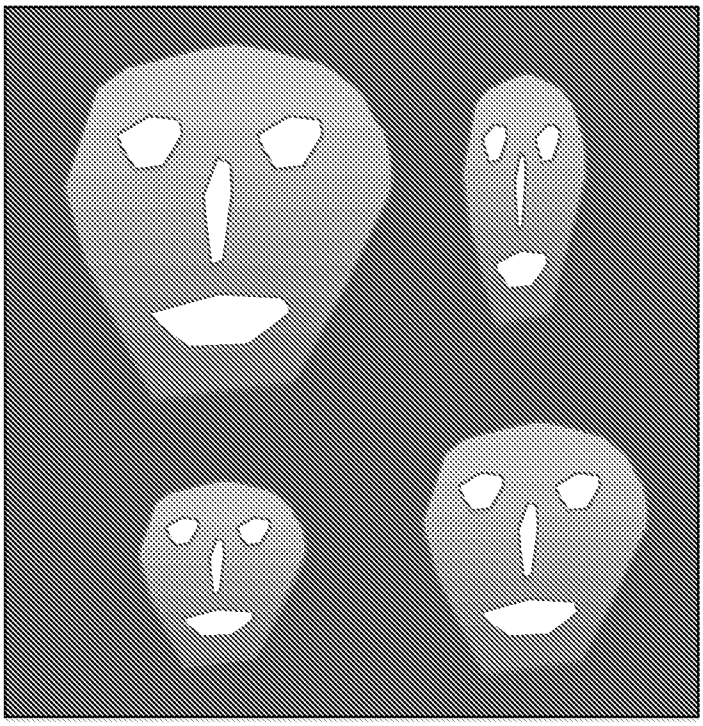


Figura 2

Para realizar el programa en pseudocódigo dispone de los siguientes operadores y estructuras de programación

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 procesado 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".

(continúa en la siguiente hoja de enunciado)

- 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 del vector X con valor distinto de cero.

[u] = unique(v) devuelve los valores del vector v 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.**