

1. (3 puntos). Transformada de Hough. Fundamento y aplicaciones.

2.- (3 puntos) La figura 1 muestra el histograma de una imagen.

- Justifique qué pasos serían necesarios para obtener la función de distribución de la imagen.
- A la vista del histograma, describa las características que cree que presenta la imagen original.
- ¿Qué operación realizaría para mejorar la calidad visual de la imagen? Justifíquela verbalmente e indique su formulación matemática.

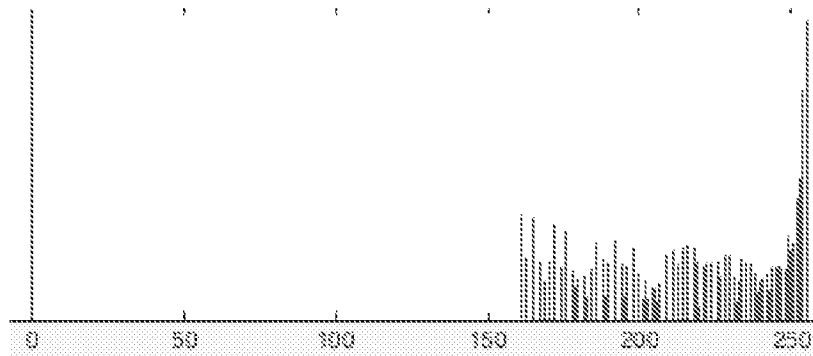


Figura 1

3.- (4 puntos) Dada la imagen de la figura 2, en la que los objetos son máscaras humanas que presentan niveles de gris más oscuros que el fondo (intensidad del fondo > 245 y $80 < \text{intensidad del objeto} < 120$), desarrolle un programa en pseudocódigo para realizar las siguientes operaciones:

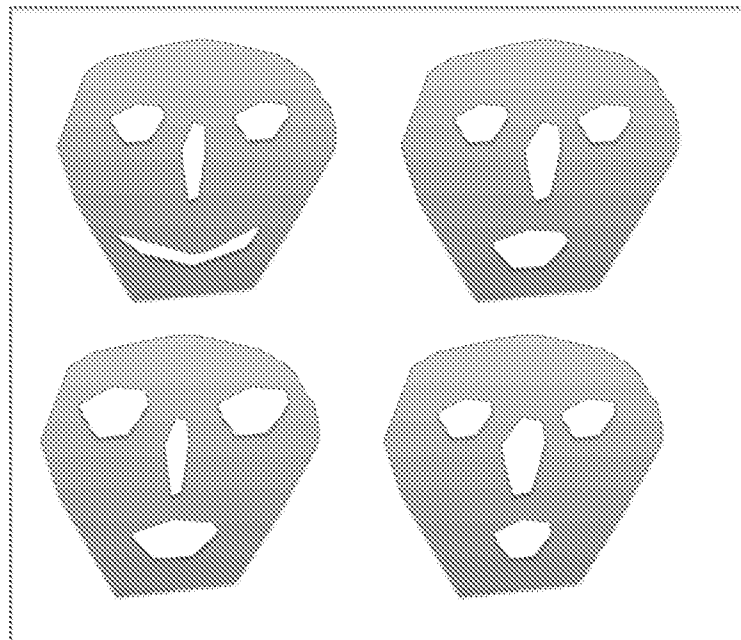


Figura 2

(continúa en la siguiente hoja del enunciado)

- a) localizar la región de la imagen correspondiente a cada máscara, es decir, los blobs correspondientes a cada máscara.
- b) localizar y distinguir la boca (considere que es el agujero que está más abajo en cada máscara).
- c) Localizar la boca más alargada.
- d) Genere y muestre una imagen del mismo tamaño de la imagen de entrada que contenga únicamente la máscara con la boca más alargada.

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 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 comentar el código para justificar las decisiones tomadas.**