

1.- (3 puntos) Flujo optico: fundamentos, aplicaciones y limitaciones.

2.- (3 puntos) La función de distribución de la figura 1 se ha obtenido a partir del histograma de una imagen.

- Justifique qué pasos han sido necesarios para obtenerlo si la imagen de entrada estaba en el rango  $[0,255]$ .
- Describa las características que cree que presenta la imagen original
- ¿Qué operación realizaría para mejorar la calidad de la imagen? Justifíquela verbalmente e indique su formulación matemática.

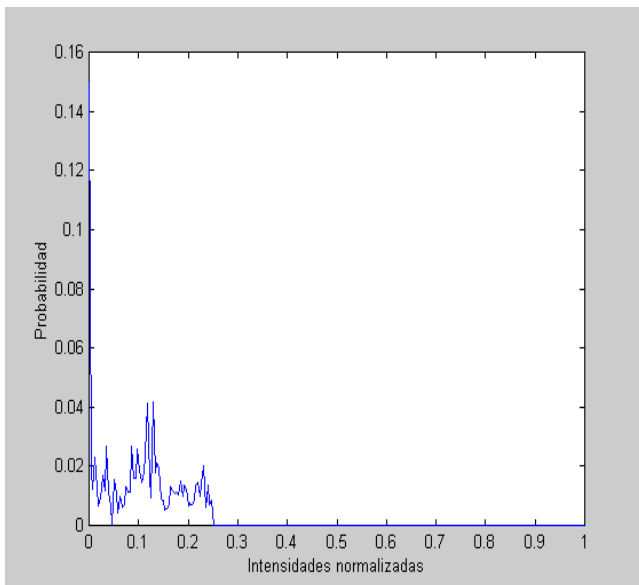


Figura 1



Figura 2

3.- (4 puntos) Dada la imagen de la figura 2, que contiene un objeto más oscuro que el fondo (considere intensidad del fondo=255 e intensidad del objeto  $<160$ ). Desarrolle un programa en pseudocódigo para realizar las siguientes operaciones:

- localizar la región de la imagen correspondiente al objeto. Se supone que sólo hay un objeto en la imagen, ¿cómo podría comprobarlo?
- localizar y distinguir los agujeros del objeto
- localizar el agujero de mayor área.
- muestre una imagen con únicamente el agujero de mayor tamaño.

**\* Debe comentar el código para justificar las decisiones tomadas.**

(continúa en la siguiente hoja de enunciado)

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.