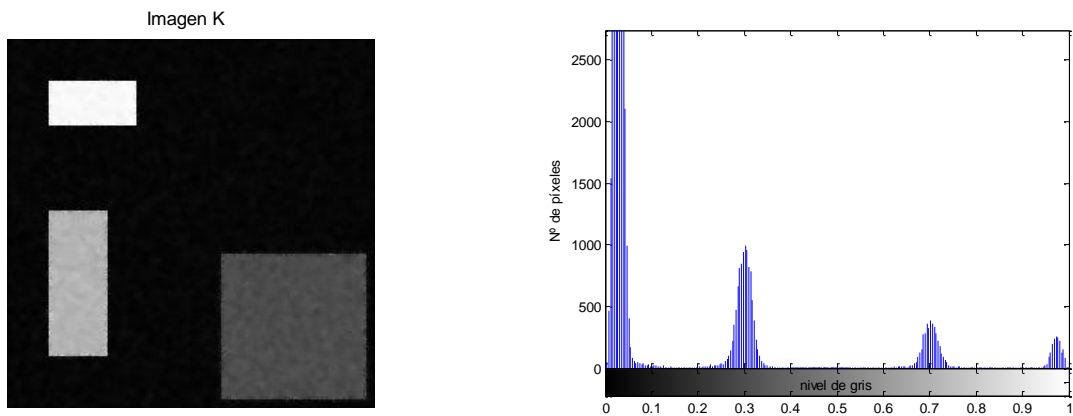


1. SEGMENTACIÓN BASADA EN EL HISTOGRAMA

1.2. Segmentación basada en el histograma, supervisada mediante umbral global

Observando el histograma, se localizan en él los distintos objetos y se colocan umbrales para quedarse únicamente con los objetos de interés. Ejecute el siguiente ejemplo:

```
BW1=zeros(256,256); %Crea imagen (toda a nivel negro)
BW1(30:60,30:90)=1; %Crea rectangulo blanco dentro de BW1
BW1(120:220,30:70)=0.7; %Crea rectangulo gris claro dentro de BW1
BW1(150:250,150:250)=0.3; %Crea cuadrado gris oscuro dentro de BW1
J=imnoise(BW1,'gaussian',0,0.005); %Introduce ruido gaussiano en BW1
K=wiener2(J,[5 5]); %Filtra la imagen
imshow(BW1); title('Imagen BW1 ');
figure; imshow(K); title('Imagen K ');
figure;imhist(K);xlabel('nivel de gris');ylabel('Nº de píxeles');
```



A la vista del histograma, se observa como aplicando un umbral de 0.2, se segmentarán (previsiblemente) todos los objetos de la imagen (dado que el objeto más numeroso en nº de píxeles es el fondo: cercano al valor 0 (negro)):

```
Um=(K >= 0.2); %Imagen binaria con pixels a "1" donde K >= 0.2 y "0" en el resto de pixels.
figure; imshow(Um); title('Resultado de la segmentación con el umbral puesto');
```

Resultado de la segmentación con el umbral puesto



Pruebe a segmentar los distintos objetos en imágenes distintas.

1.3. Segmentación supervisada por modelos gaussianos (no lineal)

El mayor problema de la segmentación anterior es el cálculo de los umbrales. Además, los objetos se suelen distribuir en el histograma de forma gaussiana. Se puede hacer una segmentación mejor caracterizando los niveles de gris de los objetos de la imagen y del fondo mediante modelos basados en funciones de densidad de probabilidad gaussianas (véase la Figura 1) y obteniendo, luego, el umbral óptimo o viendo el grado de pertenencia de un píxel (con un determinado nivel de gris) a uno u otro modelo (objeto, fondo...) y clasificando después el píxel al modelo cuyo grado de pertenencia es mayor.

Para la creación de estos modelos gaussianos de los objetos únicamente se necesitan los parámetros media y varianza de los niveles de gris del objeto.

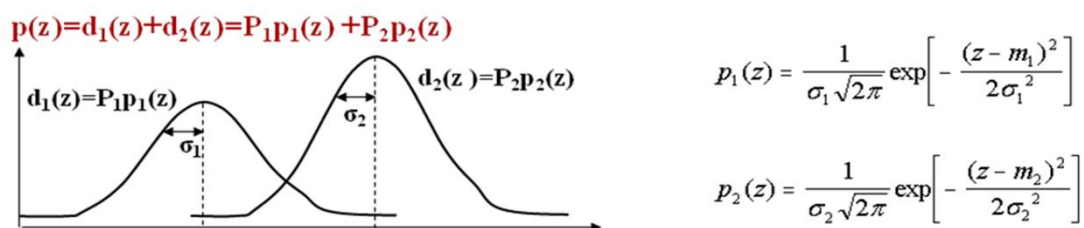


Figura 1.- Histograma con dos objetos modelados mediante funciones gaussianas

Localice, del modo mostrado en el listado de debajo, usando la función *impixel*, un conjunto de puntos representativos del fondo y de los granos de la imagen 'rice.tif' obteniendo el valor de gris de los mismos.

```
imfinfo('rice.tif') %Información sobre la imagen
X=im2double(imread('rice.tif')); %imagen cuadrada de 256x256; length(X)=256
figure;imhist(X);
disp('Selecciona 1º, con el raton (botón izqu.), unos cuantos pixels de fondo. Luego enter.');
```

```
disp('Selecciona después unos cuantos pixels de objetos. Luego enter');
```

```
val=impixel(X),title('Selecciona pixeles del fondo.');
```

```
fondo=val;
```

```
val=impixel(X),title('Selecciona pixeles de los objetos');
```

```
granos=val;
```

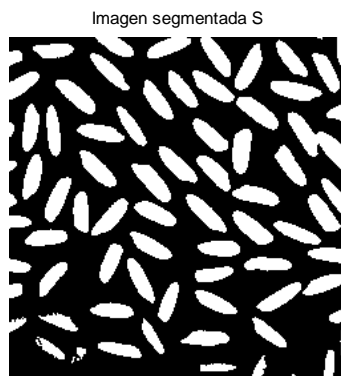
A partir de estos píxeles, con las siguientes líneas de código, obtenga los estadísticos de los dos conjuntos de píxeles (clase fondo y clase grano):

```
mf=mean(fondo(:,1)); %Media de la clase fondo
mg=mean(granos(:,1)); %Media de la clase grano
sf=std(fondo(:,1)); %Desviacion tipica de la clase fondo
sg=std(granos(:,1)); %Desviacion tipica de la clase grano
```

Seguidamente, segmente la imagen clasificando cada píxel de la misma (en función de su nivel de gris) en el modelo para el cual la función densidad de probabilidad (fdp) sea

mayor (grado de pertenencia mayor). Se suponen las clases equiprobables ($P_1=P_2$ en la Fig. 1).

```
tic %para computar cuanto tarda este algoritmo desde este punto
S=zeros(length(X),length(X)); %Imagen a segmentar S inicializada ceros (fondo -> todo negro)
for i=1:length(X) %Para todas las filas de la imagen
    for j=1:length(X) %Para todas las columnas de la imagen
        x=X(i,j); %Se toma cada pixel "x" de la imagen
        bayesf=(1/(sqrt(2*pi)*sf))*exp(-0.5*(x-mf)'*sf^(-2)*(x-mf)); %fdp normal
        %Nota.- Se puede poner (x-mf)' o (x-mf) pues es un número y (x-mf)=(x-mf)'
        bayesg=(1/(sqrt(2*pi)*sg))*exp(-0.5*(x-mg)'*sg^(-2)*(x-mg)); %fdp normal
        %Nota.- Se puede poner (x-mg)' o (x-mg) pues es un número y (x-mg)=(x-mg)'
        if bayesg > bayesf %Pixel en posicion (i,j) tiene > pertenencia a la fdp normal clase grano
            S(i,j)=1; %Pone a "1" (blanco) el pixel en posicion (i,j) pues pertenece a la clase grano
        end
    end
end
figure,imshow(S);title('Imagen segmentada S');
toc %devuelve cuanto tardo el algoritmo desde el inicio del tic
```



El código con for anidados anterior se puede sustituir, aprovechando la capacidad de Matlab de trabajar vectorial y matricialmente, por este otro, más rápido, donde se ha llamado SSS a la matriz con los resultados de la segmentación (en lugar de S)

```
tic %para computar cuanto tarda este algoritmo desde este punto
SSS=zeros(length(X),length(X)); %Imagen a segmentar inicializada ceros(fondo -> todo negro)
bayesf_total=(1/(sqrt(2*pi)*sf))*exp(-0.5*(X-mf)*sf^(-2).*(X-mf)); %fdp normal
bayesg_total=(1/(sqrt(2*pi)*sg))*exp(-0.5*(X-mg)*sg^(-2).*(X-mg)); %fdp normal
SSS= (bayesg_total > bayesf_total);
figure,imshow(SSS); title('Imagen segmentada SSS');
toc %devuelve cuanto tardo el algoritmo desde el inicio del tic
```

Teclee (si quiere comprobar que el resultado es el mismo con for anidados o trabajando matricialmente):

```
find (S-SSS)
```