

## Laboratorio de Procesamiento de Imágenes

### 1. Representación de imágenes digitales monocromáticas

- a) Construye la matriz vinculada a la imagen de la figura 1.

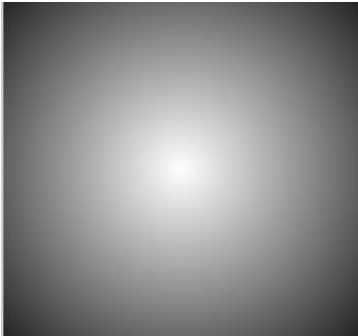


Figura 1. Iluminación por focos.

```
Programa
%Coordenadas del foco (a,b)
a=150;
b=150;
%Construcción imagen MxN
M=300; N=300;
for x=1:M;
    for y=1:N;
        I(x,y)=(255-sqrt((x-a)^2+(y-b)^2))/255;
    end
end
imshow(I)
```

- b) Construye la *matriz vinculada* a la imagen de la figura 2.

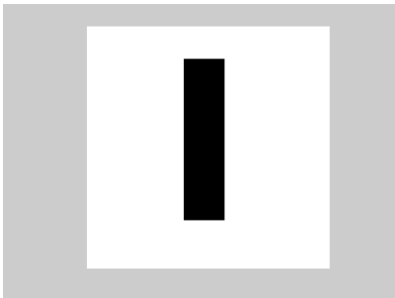


Figura 2. Rectángulo negro.

```
Programa
%imagen de tamaño 30x30
I=ones(30,30);
I(5:24,13:17)=0;
imshow(I,'initialMagnification','fit')
```

- c) Extrae de la imagen de la figura 3 la *región de interés* mostrada en la figura 4.

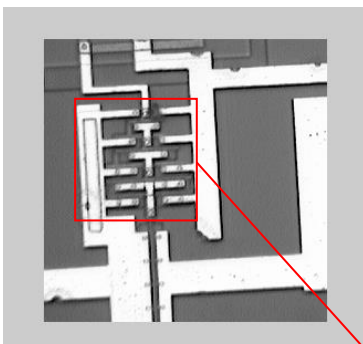


Figura 3. Imagen de un circuito.



Figura 4. Región de interés.

```
Programa
I=imread('ic.tif');
imshow(I)
J=imcrop;
%seleccionar con el ratón la región de
%interés
imshow(J)

%También se puede hacer fijando
%[xmin ymin ancho y alto]
I=imread('ic.tif');
I1=imcrop(I,[60 40 100 90]);
imshow(I)
figure,imshow(I1)
```

## 2. Operaciones aritméticas y geométricas con imágenes

- Multiplica la imagen de la figura 5 por 1.3. Súmale 50 a cada píxel de la imagen de la figura 5. Compara las imágenes resultantes. ¿Qué conclusiones sacas?
- Aumenta una cierta región de la imagen de la figura 5 para verla mejor.

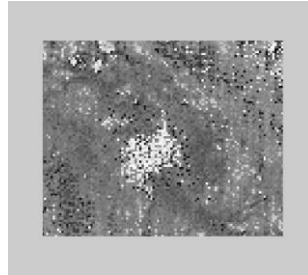
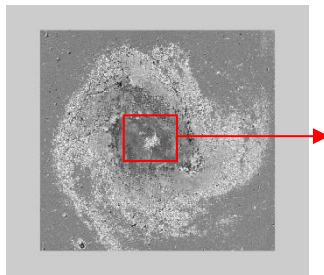


Figura 5. Galaxia.

### Programa

```
I=imread('m83.tif');
imshow(I)
zoom on
%marcar la región a aumentar
%aparecerá la imagen de la región
%marcada
```

- Modifica el tamaño de la imagen de la figura 5.

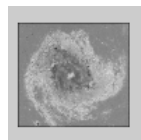


Figura 6. Imagen reducida.

### Programa

```
I=imread('m83.tif');
J=imresize(I,0.5,'bilinear');
imshow(J)
%también se puede poner
J1=imresize(I,[30 40]) %reduce la
%imagen al tamaño 40x30; 30 filas y
%40 columnas
figure, imshow(J1)
```

- Gira 30° la imagen anterior (figura 6)

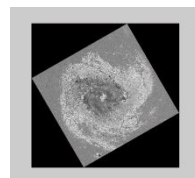


Figura 7. Imagen girada.

### Programa

```
IR=imrotate(J,30);
imshow(IR)
```

## 3. Conectividad de píxeles

- Determina el objeto de la imagen de la figura 7 que está en la posición (columna 90, fila 197) y el que está en la posición (16, 67), suponiendo que un objeto viene dado por un conjunto de píxeles de igual tonalidad y conectados por vecindad (entornos de 8 vecinos).

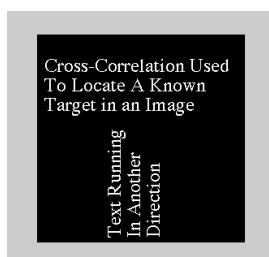


Figura 7. Caracteres de texto.

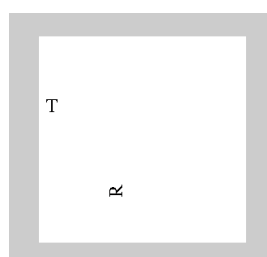


Figura 8. Objetos encontrados.

### Programa

```
I=imread('text.tif');
c=[90 16]; r=[197 67];
I0=bwselect(I,c,r,8);
imshow(I)
figure, imshow(I0)
```

## 4. Histograma de una imagen

Determina el histograma de la imagen de la figura 9.



Figura 9. Espina dorsal.

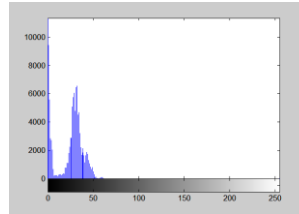


Figura 10. Histograma.

### Programa

```
I=imread('spine.tif');
imshow(I)
figure, imhist(I)
```

## 5. Transformaciones puntuales

- a) Aplica a la imagen de la figura 9 una transformación que ajuste los tonos de gris al rango [0 255]

La transformación más sencilla es la siguiente:

$$y = T(x) = \begin{cases} ax & \text{si } x \in [0 \ 63] \\ 255 & \text{si } x \in (63, 255] \end{cases}$$

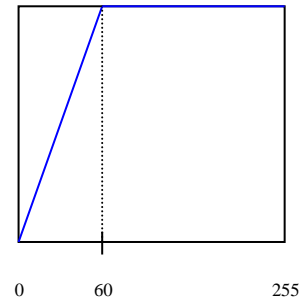


Figura 11. Transformación.

El valor de la pendiente  $a$  se determina por la condición  $T(63)=255$ , es decir,  $a \cdot 60 = 255$ . Por lo tanto,  $a \approx 4.05$ . Como la pendiente es mayor que uno la imagen aumenta su contraste en el intervalo [0 60].

### Programa

```
I=imread('spine.tif'); imshow(I);
I1=im2double(I); I2=I1;
for n=1:490,
    for m=1:367,
        I2(m,n)=I1(m,n)*255/63;
        if I2(m,n)>1,
            I2(m,n)=1;
        end
    end
end
figure, imshow(I2)
```



Figura 12. Espina dorsal.

También se puede realizar un ajuste de los tonos de gris del intervalo [0 60] al intervalo [0 255] mediante el comando `imadjust` de MATLAB:

### Programa

```
I=imread('spine.tif'); imshow(I);
J=imadjust(I,[0 0.247],[])
imshow(J)
```

Observa que  $63/255 \approx 0.247$ .

- b) Aplica a la imagen de la figura 13(a) una transformación que aumente el contraste en las regiones más claras a costa de reducirlo en las más oscuras.

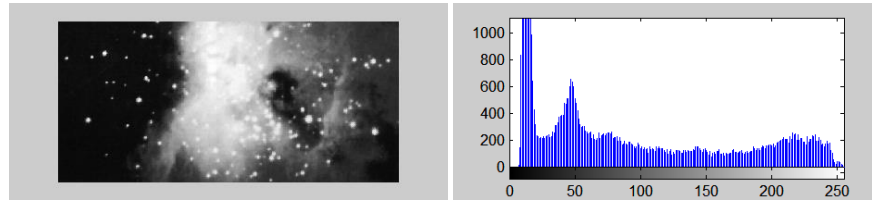


Figura 13. (a) Nebulosa. (b) Histograma.

Una transformación sencilla, suponiendo que el rango de la imagen es el intervalo  $[0,1]$ , es la siguiente

$$y = T(x) = \begin{cases} ax & \text{si } x \in [0, c] \\ \frac{1-ac}{1-c}(x-c) + ac & \text{si } x \in (c, 1] \end{cases}$$

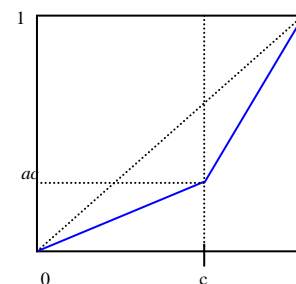


Figura 14. Transformación.

El valor de la pendiente de la recta en el tramo  $[0, c]$  es  $a$ , siendo  $a < 1$ , por lo que el rango  $[0, c]$  lo reduce al rango  $[0, ac]$ , es decir, se pierde contraste en dicho tramo (zonas más oscuras) mientras que en el tramo  $[c, 1]$  la recta tiene de pendiente mayor que 1, es decir, los valores de dicho intervalo se reducen al intervalo  $[ac, 1]$ , por lo que se aumenta el contraste en dicho tramo (valores más claros).

```

Programa
I=imread('ngc40241.tif'); imshow(I);
I1=im2double(I); %Rango de la imagen en [0,1]
a=0.5;
c=220/255;
for n=1:368,
    for m=1:174,
        if I1(m,n) < c,
            I2(m,n)=I1(m,n)*a;
        else
            I2(m,n)=a*c+(I1(m,n)-c)*(1-a*c)/(1-c);
        end
    end
end
figure, imshow(I2)

```

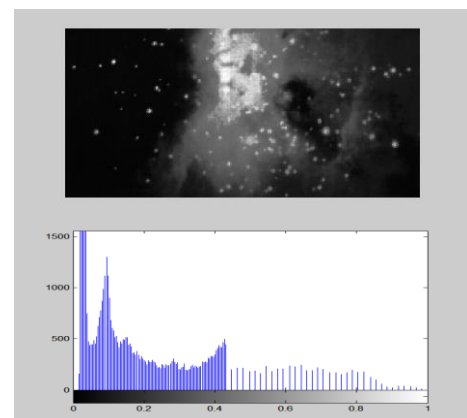


Figura 15. Imagen resultante y su histograma.

- c) Aplica una transformación a la imagen de la figura 6 que aumente el contraste en los tonos de gris intermedios (intervalo  $[0.4, 0.6]$ ).
- d) Aplica una transformación no lineal de la forma  $T(x)=ax^2$  que mejore el contraste en las zonas más claras. ¿Qué valor de  $a$  seleccionarías?
- e) Consigue una imagen binaria con fondo negro donde sólo aparezcan los objetos más brillantes.

## 6. Igualación del histograma

a) Aplica la técnica de igualación del histograma para mejorar la calidad de la imagen de la figura 16(a).

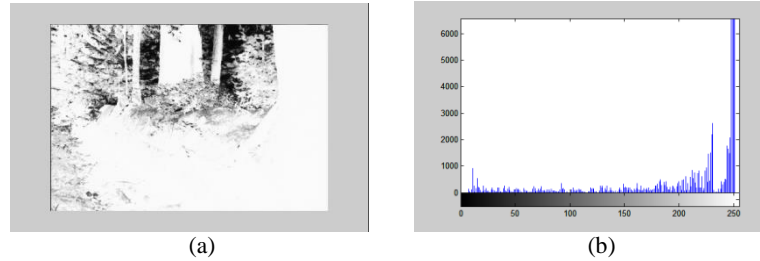


Figura 16. a) Bosque. b) Histograma de la imagen.



Figura 17. Imagen igualada y su histograma.

Si restamos a la imagen original la imagen igualada se obtiene la imagen de la figura 18, donde también podemos ver su histograma, que pone de manifiesto las diferencias entre las dos imágenes.

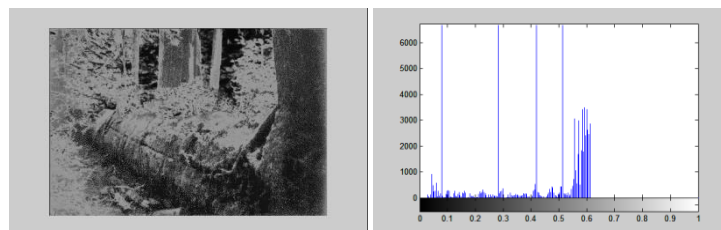


Figura 18. Imagen diferencia y su histograma.

b) Aplica la técnica de igualación del histograma a la imagen de la figura 19(a) (moon.tif). Comenta el resultado.

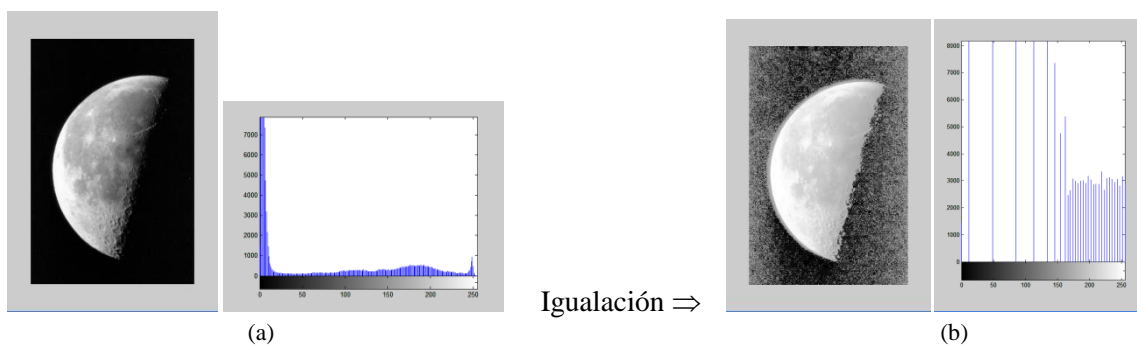


Figura 19. a) Imagen de la Luna y su Histograma. b) Imagen igualada y su Histograma.

La imagen igualada 19(b) es de peor calidad puesto que se han mejorado los detalles del fondo (parte más oscura) a costa de reducir el contraste del objeto (la luna) que es la parte más clara.



## Escuela Técnica Superior de Ingeniería Informática Universidad de Málaga



Para ver la transformación que hemos realizado para tratar de conseguir una igualación del histograma, escribimos el siguiente programa:

```
Programa  
I=imread ('moon.tif');  
[J,T]=histeq(I);  
plot((0:255)/255,T);
```

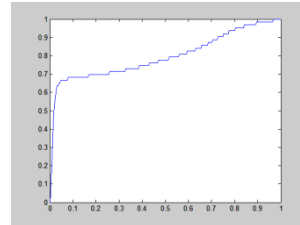


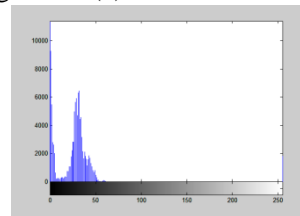
Figura 20. Transformación realizada.

### 7. Restauración mediante filtrado en el dominio espacial

a) Mejora la calidad de la imagen con ruido de la figura 21 (a).



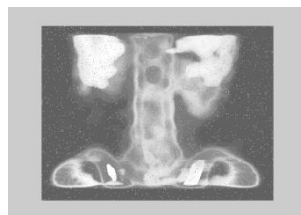
(a)



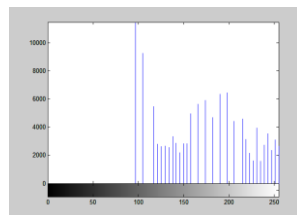
(b)

Figura 21. a) Imagen radiológica de la espina dorsal. b) Histograma de la imagen.

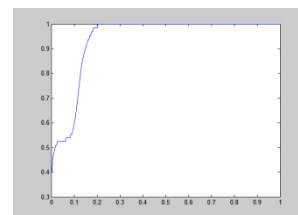
Para obtener la imagen de la figura 21(a) le añadimos primero ruido del tipo “sal y pimienta” (puntos blancos y negros) con una densidad de ruido igual a 0.02 (afecta al 2% de los píxeles). A continuación le aplicamos la transformación de igualación del histograma para mejorar la calidad de la imagen y comprobamos que el ruido no se ha eliminado. Para eliminar o reducir dicho ruido podemos aplicarle un filtro de paso baja como el *filtro mediana* 3×3 y se obtiene, finalmente, la imagen de la figura 23.



(a)



(b)



(c)

Figura 22. (a) Imagen igualada. b) Histograma. c) Transformación para la igualación.

```
Programa  
I=imread('spine.tif');  
I=imnoise(I,'salt & pepper',0.02);  
[J,T]=histeq(I);  
plot((0:255)/255,T);  
M=medfilt2(J,[3 3]);  
imshow(M)
```



Figura 23. Imagen restaurada.

Si en lugar de aplicarle dicho filtro le aplicamos el *filtro media* obtenemos la imagen de la figura 24 en la que se puede ver que el ruido se ha atenuado pero no se ha eliminado:

```
Programa
%continuación del anterior
%primero generamos el filtro media
g=fspecial('average',[5 5])
M=filter2(g,J)/255;
imshow(M)
```



Figura 24. Imagen mejorada con el filtro media.

b) Añade *ruido gaussiano* a la imagen de la figura 21(a) y después restaura dicha imagen con el filtro *media* y con el filtro *mediana*. Comenta los resultados.

```
Programa
I=imread('spine.tif');
J=imnoise(I,'gaussian',0,0.01);
g=fspecial('average',[5 5])
M=filter2(g,J)/255;
imshow(M)
M1=medfilt2(J,[5 5]);
figure, imshow(M1)
```

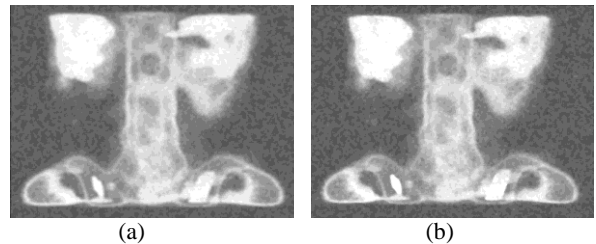


Figura 25. (a) Restaurada con el filtro media. (b) Restaurada con el filtro mediana.

## 8. Detección de bordes mediante filtrado en el dominio espacial

a) Utiliza un operador basado en un filtro de paso alta para la detección de los contornos de los granos de arroz de la imagen de la figura 26.

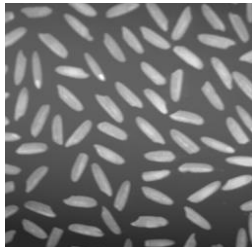
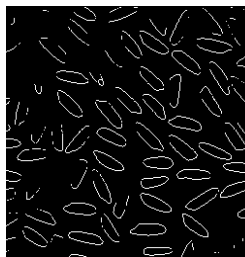
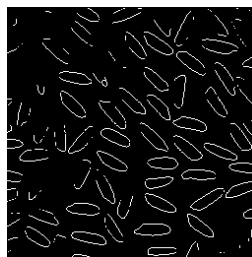


Figura 26. Imagen de arroz.

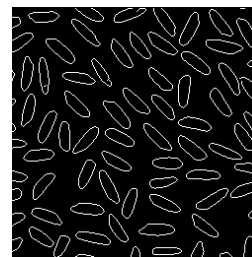
```
Programa
I=imread('rice.tif');
imshow(I)
B1=edge(I,'sobel');
B2=edge(I,'prewitt');
B3=edge(I,'canny');
imshow(B1)
figure, imshow(B2)
figure, imshow(B3)
```



(a)



(b)



(c)

Figura 27. a) Imagen con filtrado de Sobel. b) Imagen con filtrado de Prewitt. c) Imagen con filtrado de Canny.

b) Detecta los bordes horizontales de la imagen de la figura 26 aplicando un filtro de paso alta.

```
Programa
I=imread('rice.tif');
g=[1 1 1;0 0 0; -1 -1 -1];
J=filter2(g,I);
J=abs(J);
B=J>0.27*(max(J(:))-min(J(:)));
imshow(B)
```



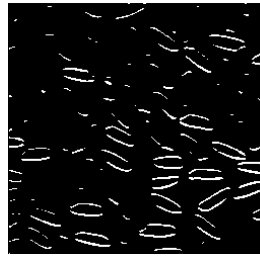


Figura 28. Imagen de bordes horizontales.

- c) Detecta los bordes verticales de la imagen de la figura 29 (wafer1.tif).

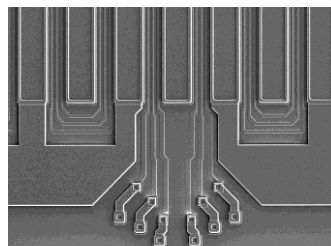


Figura 29. Imagen wafer1.tif.

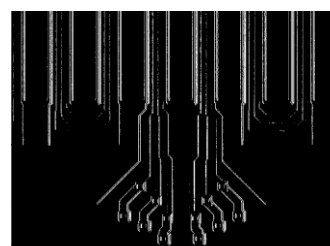


Figura 30. Imagen de bordes verticales.

- d) Aplica un filtro de pasa alta a la imagen de la figura 31 y muestra la imagen resultante.

```
Programa  
I=imread('saturno.tif');  
h=[1 2 1; 0 0 0; -1 -2 -1];  
I1=filter2(h,I);  
imshow(I1,[]), colorbar
```

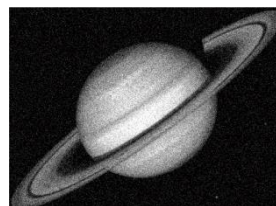


Figura 31. Saturno.

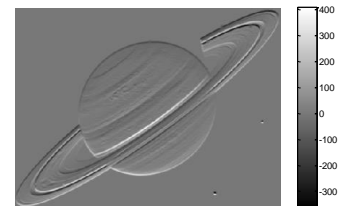


Figura 32. Imagen filtrada.

## 9. Realzado de imágenes mediante filtrado

- a) Acentúa (perfila) los contornos a la imagen de la figura 31.

```
Programa  
I=imread('saturno.tif');  
h=fspecial('unsharp');  
J=abs(filter2(h,I));  
J1=J/255; % /max(J(:));  
imshow(J1)
```

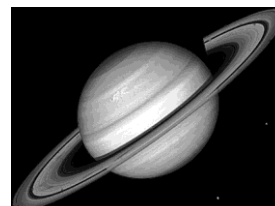
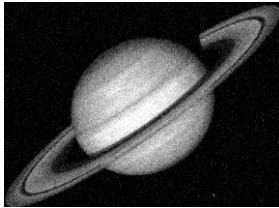


Figura 33. Imagen realzada.

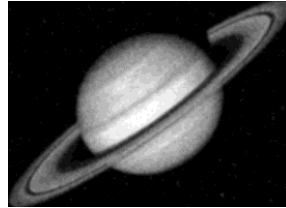
- b) Añade ruido gaussiano a la imagen de la figura 31 y a continuación restaura y realza la imagen resultante.

```
Programa  
I=imread('saturno.tif');  
J=imnoise(I, 'gaussian', 0, 0.01);  
imshow(J)  
g=fspecial('average', [5 5])  
M1=filter2(g,J)/255;  
h=fspecial('unsharp')  
M2=abs(filter2(h,M1));  
imshow(M1); figure, imshow(M2)
```

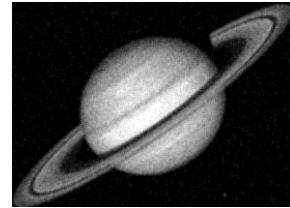




(a)



(b)



(c)

Figura 34. a) Imagen tratada con ruido gaussiano. b) Imagen restaurada con el filtro media 5×5. c) Imagen realzada con el filtro “unsharp” de perfilado.

- c) Realza la imagen de la figura 26 utilizando un filtro lineal.

```
Programa
I=imread('rice.tif');
gv=[-1 0 1;-1 1 1; -1 0 1];
gh=[1 1 1;0 1 0; -1 -1 -1];
Jv=abs(filter2(gv,I))/255;
Jh=abs(filter2(gh,I))/255;
imshow(Jv); figure, imshow(Jh)
```

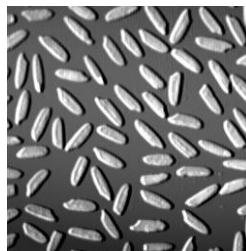


Figura 35. Realzado vertical

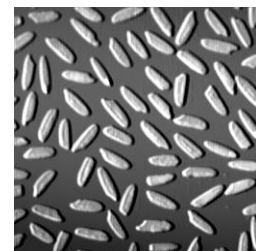


Figura 36. Realzado horizontal

- d) Realza la imagen ecualizada de la espina dorsal (figura 22(a)) utilizando la imagen de bordes.

```
Programa
I=imread('spine.tif');
I=histeq(I);
imshow(I);
B=edge(I,'canny');
figure, imshow(B);
R=double(I)/255+0.1*double(B);
figure, imshow(R)
```

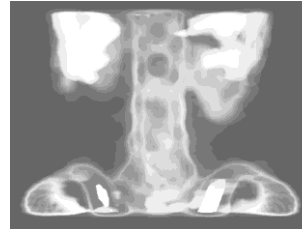


Figura 37. Imagen realzada.

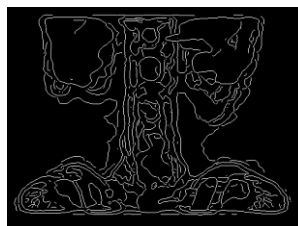


Figura 38. Imagen de bordes.

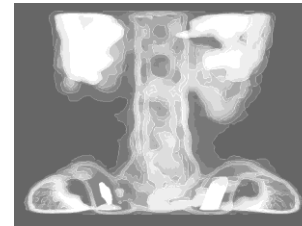


Figure 39. Imagen realzada.

- e) Mejora la imagen de un insecto (figura 40) realzando sus bordes horizontales.

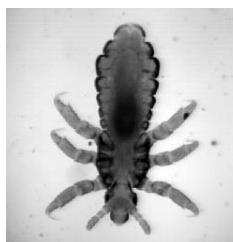


Figura 40. Insecto.



Figura 41. Imagen realzada.

## 10. Transformada de Fourier discreta

a) Determina la *transformada de Fourier discreta* de la imagen de la figura 42(a).

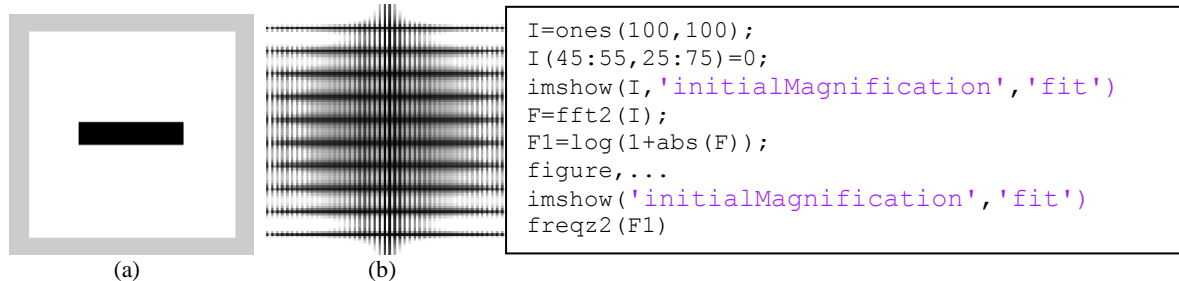


Figura 42. a) Rectángulo. b) Espectro de Fourier.

b) Determina la *transformada de Fourier discreta* de la imagen de la figura 42(a) pero con el valor  $|F(0,0)|$  en el centro de la representación.

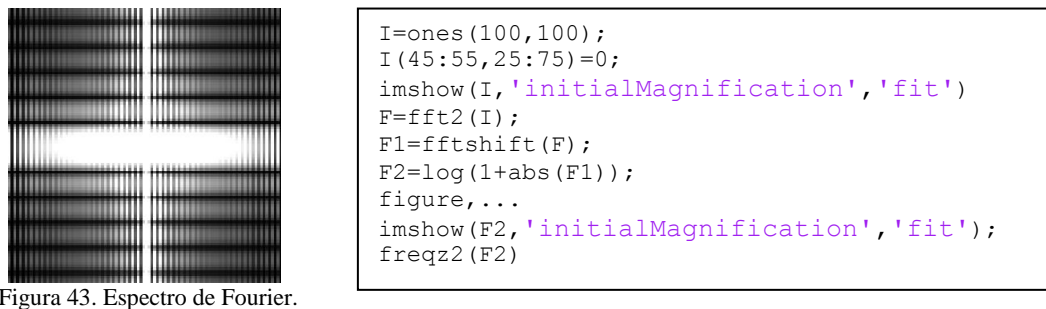


Figura 43. Espectro de Fourier.

c) Determina la *transformada de Fourier discreta* de la imagen de la figura 44(a).

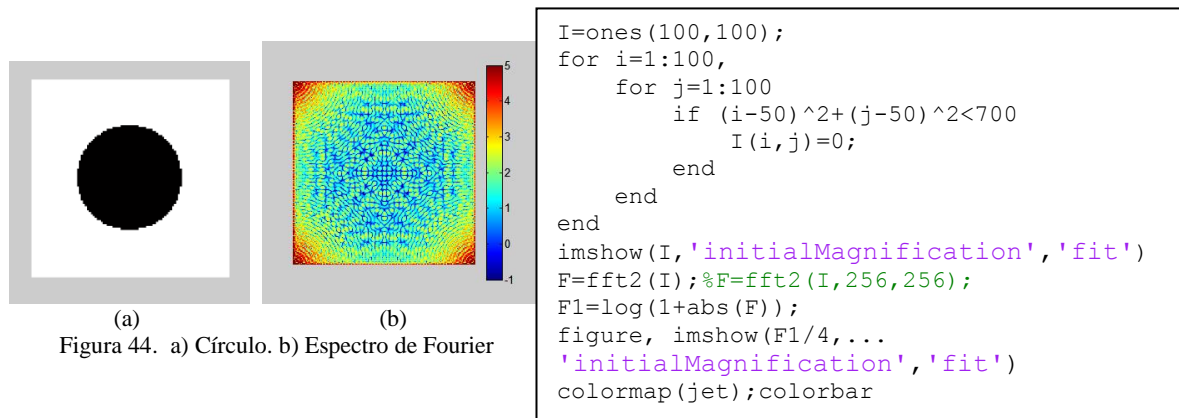


Figura 44. a) Círculo. b) Espectro de Fourier

d) Determina la transformada de Fourier discreta de la imagen de la figura 45(a) pero con el valor de  $|F(0,0)|$  en el centro de la imagen.

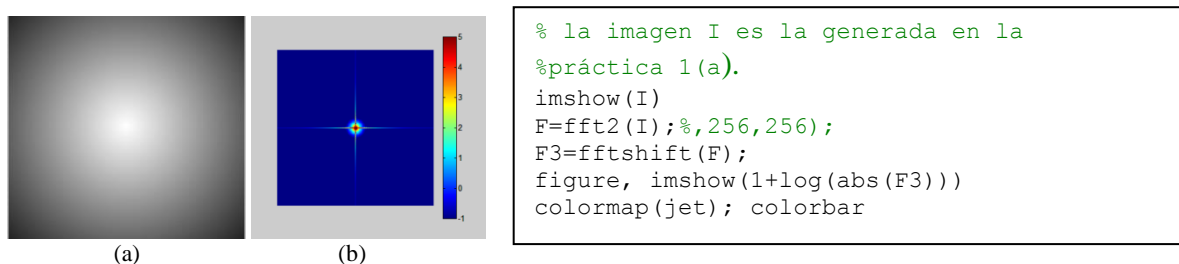


Figura 45. a) Punto luminoso. b) Espectro de Fourier.



- e) Determina la transformada de Fourier discreta de la imagen de la figura 46(a) pero con el valor de  $|F(0,0)|$  en el centro de la imagen.

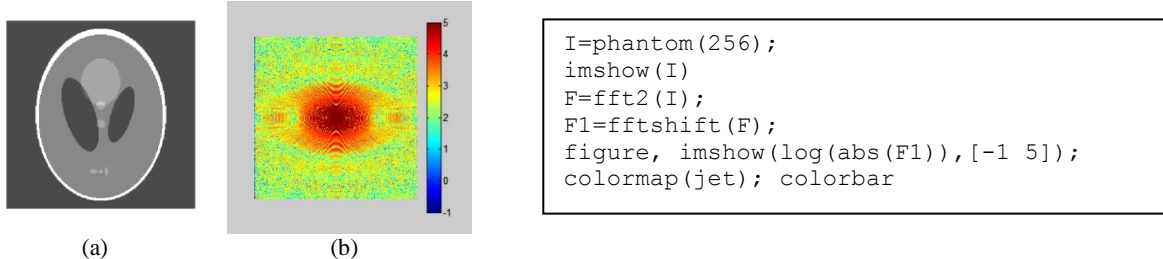


Figura 46. a) Tomografía. b) Espectro de Fourier

## 11. Diseño de filtros en el dominio de las frecuencias

- a) Construye un filtro ideal de paso baja en el dominio de las frecuencias

```
[u,v]=freqspace(25,'meshgrid');  
H=sqrt(u.^2+v.^2)<0.5;  
meshz(u,v,H)
```

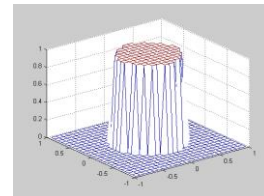


Figura 47. Filtro ideal.

- b) Construye el filtro en el dominio de las frecuencias (respuesta de frecuencias) correspondiente al *filtro de Laplace*  $3 \times 3$  definido en el dominio espacial.

```
h=[0 -1 0;-1 4 -1;0 -1 0];  
freqz2(h)
```

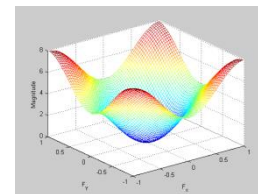


Figura 48. Filtro de Laplace.

- c) Determina el filtro en el dominio de las frecuencias que le corresponde al *filtro gaussiano*  $3 \times 3$ .

```
h=fspecial('gaussian')  
freqz2(h)
```

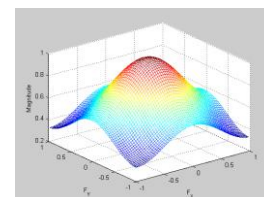


Figura 49. Filtro gaussiano.

- d) Construye un filtro de paso baja por el *método de ventanas*.

Este método construye un filtro cuya respuesta de frecuencias es una suavización (aproximación) de una respuesta de frecuencias dada (deseada),  $H$ , utilizando una *función de ventana* (la ventana de Hamming).

```
[f1,f2]=freqspace(15,'meshgrid');  
H=zeros(15,15);H(5:10,5:10)=1;  
mesh(f1,f2,H),axis([-1 1 -1 1 0 1.2]),colormap(jet(64))  
h=fwind1(H,hamming(15));  
figure,freqz2(h,[32 32]),axis([-1 1 -1 1 0 1.2])
```

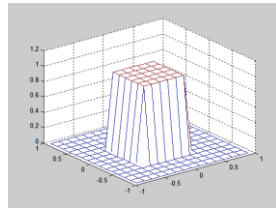


Figura 50. Filtro de paso baja.

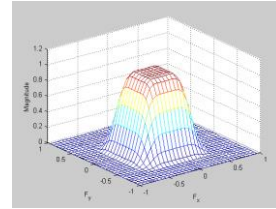


Figura 51. Filtro de paso baja.

e) Construye un filtro de paso baja por el *método de muestreo de las frecuencias*.

Este método crea un filtro cuya respuesta de frecuencias pasa a través de un conjunto de puntos dados (respuesta de frecuencias deseada) que definen la forma de la respuesta de frecuencias a construir, sin poner condiciones en el comportamiento de la respuesta de frecuencias entre los puntos dados (presentando generalmente rizados entre dichos puntos).

```
[f1,f2]=freqspace(15,'meshgrid');
Hd=zeros(15,15); Hd(4:11,4:11)=1;
mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
h=fsamp2(Hd);
figure,freqz2(h,[32 32]),axis([-1 1 -1 1 0 1.2])
```

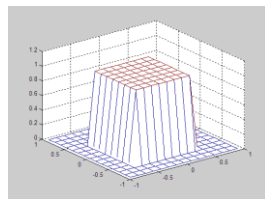


Figura 52. Filtro de paso baja.

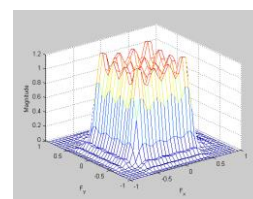


Figura 53. Filtro de paso baja.

## 12. Aplicación de filtros en el dominio de las frecuencias

a) Aplica en el dominio de las frecuencias el filtro Gaussiano de paso baja con diferentes valores del radio,  $D_0$ .

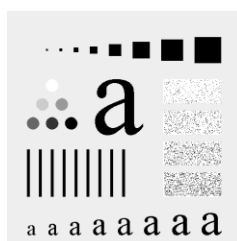
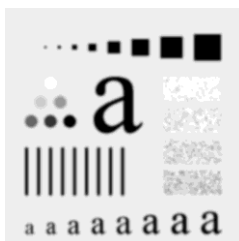
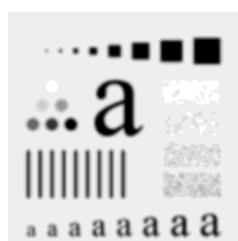


Figura 54. Letras.

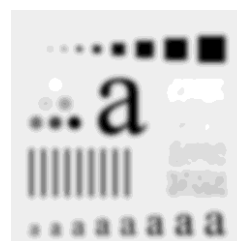
```
I0=imread('letras.jpg');
I=im2double(I0); imshow(I); [M,N]=size(I);
for i=1:2:7
    F=fft2(I);
    H=lpfilter('gaussian',M,N,5*i);
    G=H.*F;
    I1=real(ifft2(G));
    figure,imshow(I1)
end
```



(a)



(b)



(c)



(d)

Figura 55. (a) Imagen filtrada con  $D_0=5$ . (b)  $D_0=15$ . (c)  $D_0=25$ . (d)  $D_0=35$ .

- b) Aplica el filtro ideal de paso baja a la imagen de la figura 56 para diferentes valores del parámetro.

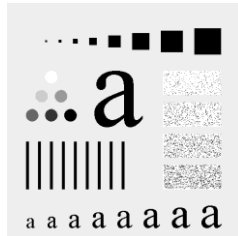


Figura 56. Letras.

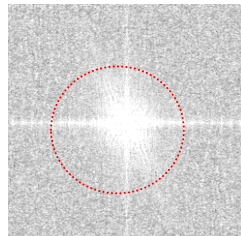
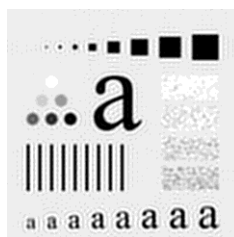
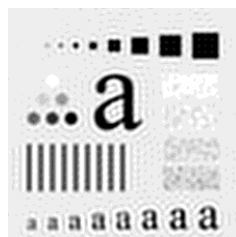


Figura 57. Transf.. Fourier

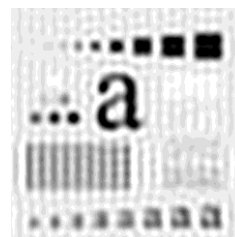
```
I0=imread('letras.jpg');
I=im2double(I0);
imshow(I)
[M,N]=size(I);
for i=1:3:10
F=fft2(I);
H=lpfilter('ideal',M,N,5*i);
G=H.*F;
I1=real(ifft2(G));
figure,imshow(I1)
end
```



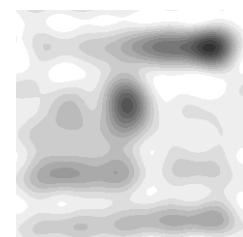
(a)



(b)



(c)



(d)

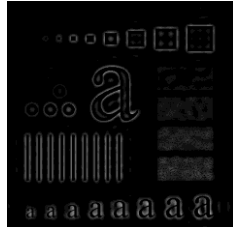
Figura 58. Imágenes filtradas. (a)  $D_0=5$ . (b)  $D_0=20$ . (c)  $D_0=35$ . (d)  $D_0=50$ .

- c) Aplica en el dominio de las frecuencias un filtro de paso alta deducido de otro de paso baja.

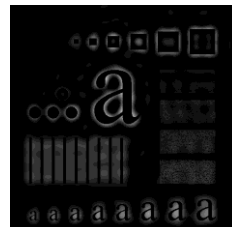
```
H=1-lpfilter('ideal',M,N,5*i);
```



(a)



(b)



(c)

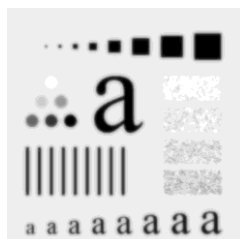


(d)

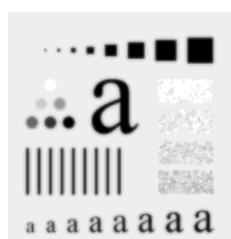
Figura 59. Imágenes filtradas. (a)  $D_0=5$ . (b)  $D_0=20$ . (c)  $D_0=35$ . (d)  $D_0=50$ .

- d) Aplica el filtro de paso baja de Butterworth a la imagen de la figura 56.

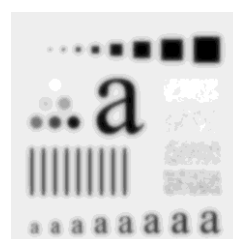
```
H=lpfilter('btw',M,N,5*i);
```



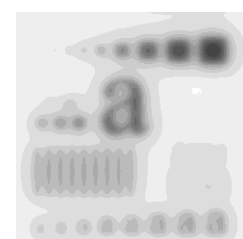
(a)



(b)



(c)



(d)

Figura 60. Imágenes filtradas. (a)  $D_0=5$ . (b)  $D_0=20$ . (c)  $D_0=35$ . (d)  $D_0=50$ .



Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga



- e) Atenúa el ruido del tipo sal y pimienta aplicando un filtro de pasa baja en el dominio de las frecuencias.

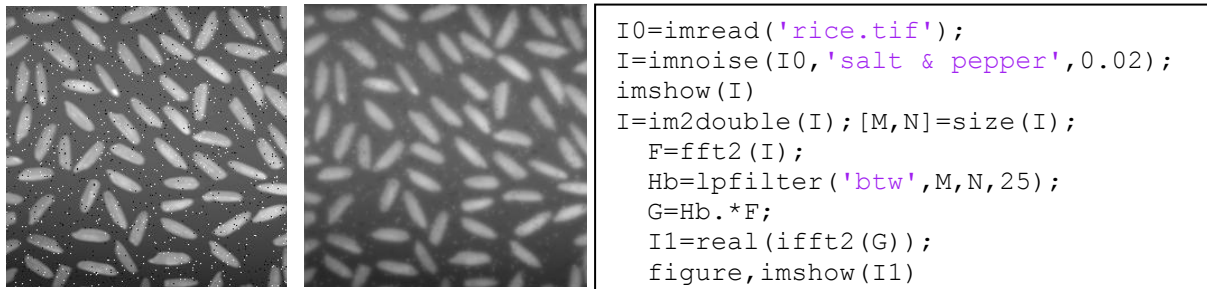
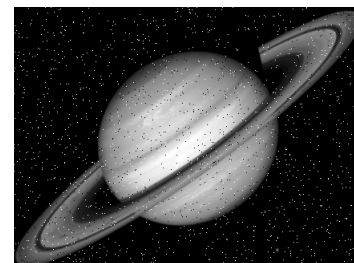


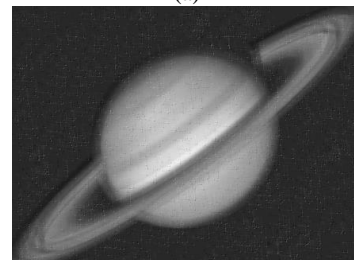
Figura 61.(a) Imagen con ruido. (b) Imagen filtrada.

- f) Construye un filtro en el dominio espacial que corresponda a un filtro ideal de paso baja en el dominio de las frecuencias y aplícaselo a una imagen que tenga ruido del tipo “sal y pimienta”.

```
I=imread('saturno.tif');  
I=imnoise(I,'salt & pepper',0.02);  
F=fft2(I);  
[f1,f2]=freqspace(25,'meshgrid');  
H=sqrt(f1.^2+f2.^2)<0.5;  
H1=H(13:25,13:25);  
h1=ifft2(H1);  
B=filter2(real(h1),I);  
imshow(I),  
figure,imshow(B,[min(min(B)) max(max(B))])
```



(a)

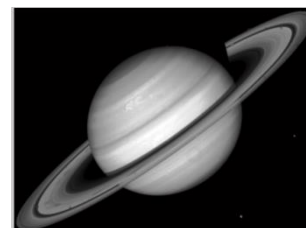


(b)

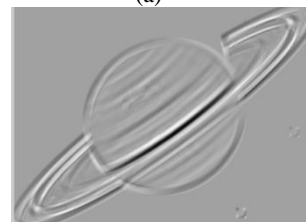
Figura 62.a) Imagen de Saturno. b) Imagen filtrada.

- g) Diseña un filtro en el dominio espacial que se corresponda con un filtro ideal de paso alta en el dominio de las frecuencias y aplícaselo a una imagen.

```
I=imread('saturno.tif');  
F=fft2(I);  
[f1,f2]=freqspace(25,'meshgrid');  
H=sqrt(f1.^2+f2.^2)>1.1;  
H1=H(13:25,13:25);  
h1=ifft2(H1);  
B=filter2(real(h1),I);  
imshow(I),  
figure,...  
imshow(B,[min(min(B)) max(max(B))])
```



(a)



(b)

Figura 63.a) Imagen de Saturno. b) Imagen filtrada.





- h) Aplica a la imagen de la figura 63(a) un filtro de paso alta diseñado por el *método de la ventana*.

```
[f1,f2]=freqspace(15,'meshgrid');  
H0=zeros(15,15);H0(12:15,12:15)=1;  
H=fwind1(H0,hamming(15));colormap(jet(64))  
freqz2(H,[32 32])  
pause  
mesh(f1,f2,abs(H))  
pause  
H1=H(7:15,7:15)  
h=ifft2(H1);  
I=imread('saturno.tif');  
B=filter2(real(h),I);  
imshow(B,[min(min(B)) max(max(B))])
```



Figura 64. Imagen filtrada.

- i) Aplica a la imagen de la figura 63(a) un filtro de paso de banda.

```
[f1,f2]=freqspace(16,'meshgrid');  
H=zeros(16,16);H(5:12,5:12)=1;H(7:10,7:10)=0;  
freqz2(H,[32 32])  
pause  
mesh(f1,f2,H)  
pause  
H1=H(9:16,9:16);  
h=ifft2(H1);  
I=imread('saturno.tif');  
B=filter2(real(h),I);  
imshow(B,[min(min(B)) max(max(B))])
```

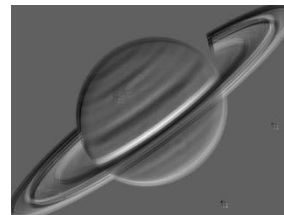


Figura 65. Imagen filtrada.



### 13. Restauración de imágenes degradadas o distorsionadas

- a) Distorsiona la imagen de la figura 68 utilizando una función de distorsión que produzca el efecto de “imagen movida” y corrígela mediante el *filtro inverso*.



Figura 68. Imagen aérea.



Figura 69. Imagen movida.

```
I=imread('concordaerial.png');  
long=31;  
ang=11;  
h=fspecial('motion',long,ang);  
B=imfilter(I,h,'circular','conv');  
imshow(B)  
W=deconvwnr(B,h);  
imshow(W)
```



Figura 70. Imagen restaurada.

- b) Restaura la imagen movida de la figura 71.

```
I=imread('Fig5.29(g).jpg');  
imshow(I)  
long=100;  
ang=135;  
h=fspecial('motion',long,ang);  
W=deconvwnr(I,h);  
figure,imshow(W)
```



Figura 71. Imagen movida.



Figura 72. Imagen restaurada.

- c) Restaura la imagen movida de la figura 73(b) que tiene además añadido ruido gaussiano utilizando primero el *filtro inverso* y después el *filtro de Wiener*.



(a)



(b)

Figura 73. a) Imagen de flores original. b) Imagen movida y con ruido gaussiano.

```
I=imread('flowers.tif');
I=I(10+[1:256],222+[1:256],:);
imshow(I)
long=20;
ang=12;
h=fspecial('motion',long,ang);
J=imfilter(I,h,'circular','conv');
figure,imshow(J);
ruido=0.1*randn(size(J));
B=imadd(J,im2uint8(ruido));
figure,imshow(B);
W=deconvwnr(B,h);
figure,imshow(W);
NSR=sum(ruido(:).^2)/sum(im2double(I(:)).^2);
W1=deconvwnr(B,h,NSR);
figure,imshow(W1)
```



Figura 74. Restaurada con el filtro inverso.



Figura 75. Restaurada con el filtro de Wiener.

- d) Corrige la imagen movida de la figura 76 (superior izquierda) que tiene además añadido ruido gaussiano utilizando el método de desconvolución a ciegas (algoritmo de Lucy-Richardson).

```
I=imread('cameraman.tif');
imshow(I)
h=fspecial('gaussian',7,10);
B=imfilter(I,h,'symmetric','conv');
figure,imshow(B);
h1=ones(size(h)-4);
%1ª restauración
[J1 P1]=deconvblind(B,h1);
figure, imshow(J1);
%2ª restauración
h2=padarray(h1,[4:4],'replicate','both');
[J2 P2]=deconvblind(B,h2);
figure, imshow(J2);
%3ª restauración
h3=padarray(h1,[2 2],'replicate','both');
[J3 P3]=deconvblind(B,h3);
figure, imshow(J3);
peso=edge(I,'sobel',.3);
ee=strel('disk',2);
peso=1-double(imdilate(peso,ee));
peso(:,1:3 end-[0:2]))=0;
figure,imshow(peso);
[J P]=deconvblind(B,h3,30,[],peso);
figure,imshow(J)
```



Figura 76. Proceso de restauración a ciegas.

## 14. Análisis morfológico

- a) Identifica los objetos rectangulares más grandes de la imagen de la figura 76 utilizando operadores morfológicos.

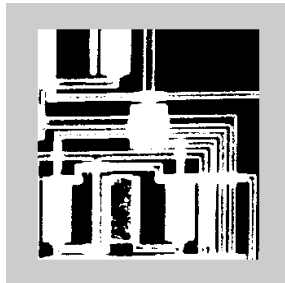


Figura 76. Circuito.

```
B=imread('circbw.tif');
h=strel('rectangle',[40 30]);
BE=imerode(B,h);
imshow(BE)
BD=imdilate(BE,h);
imshow(BD)
```

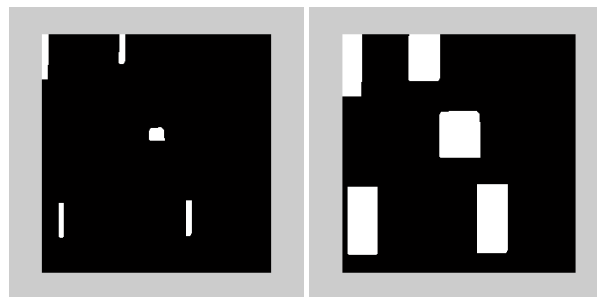


Figura 77 a) Imagen erosionada. b) Imagen dilatada.

- b) Describe un algoritmo para detectar *microestructuras* basado en la segmentación de la imagen de la figura 78 mediante operadores morfológicos.

```
I = imread('pearlite.png');
figure, imshow(I)
Ic = imcomplement(I);
B = im2bw(Ic, graythresh(Ic));
figure, imshow(Ic), figure, imshow(B);
h = strel('disk', 6);
Bc = imclose(B, h);
Bo = imopen(Bc, h);
figure, imshow(Bc),
figure, imshow(Bo);
Is= B & Bo;
figure, imshow(Is)
```

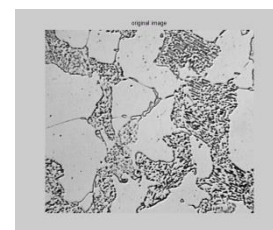


Figura 78. Imagen de Microscopio.

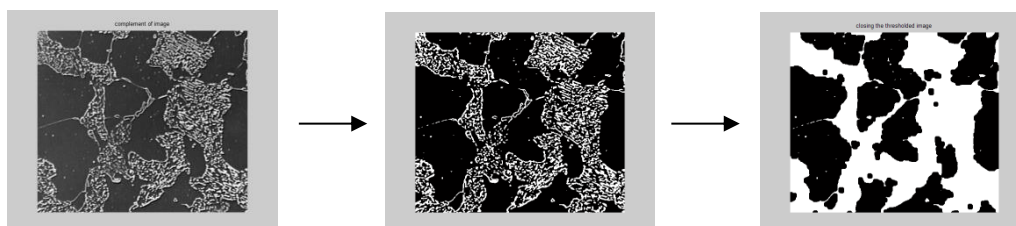




Figura 79. Secuencia de las imágenes obtenidas.

c) Atenúa el ruido de la imagen de la huella dactilar de la figura 80 y realiza un refinado de las *crestas papilares* utilizando operadores morfológicos.

```
I0=imread('Fig9.11(a).jpg');
es=strel('square',3);
I1=imopen(I0,es); %Eliminación de ruido
I2=imclose(I1,es);
D1=bwmorph(I1,'thin',1);
DF=bwmorph(I2,'thin',Inf);
imshow(I0), figure, imshow(I1),
figure, imshow(D1), figure, imshow(DF)
```

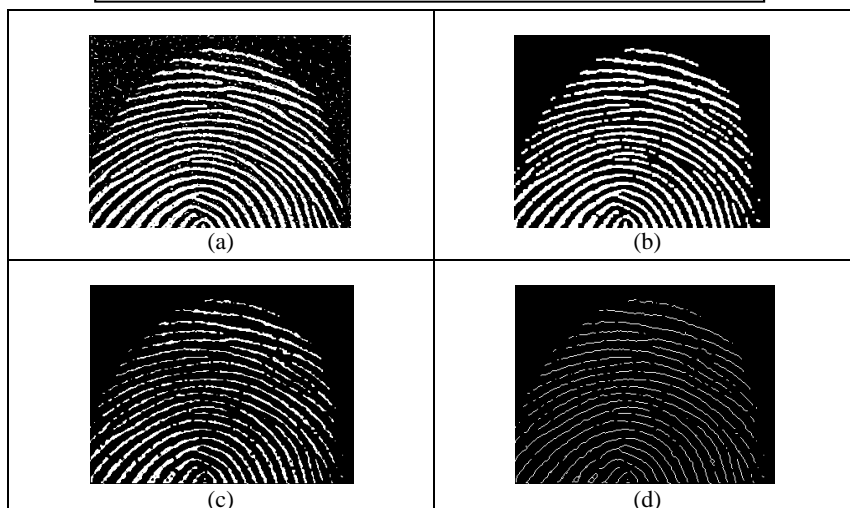


Figura 80. (a) Huella dactilar. (b) Abertura seguida de cierre de la imagen. (c) Imagen con crestas adelgazadas. (d) Imagen con crestas muy delgadas.

d) Detecta los rectángulos de tamaño  $9 \times 6$  que aparecen en la figura 81(a) utilizando el *operador de acierto y fallo*.

```
I=imread('testpat2.tif');
imshow(I)
B=zeros(11,9);
B(2:10, 2:7)=1;
B1=B;
B=ones(11,9);
B(2:10,2:7)=0;
B(1:11,9)=0;
B2=B;
C=bwhitmiss(I,B1,B2);
[fila col]=find(C);
C(fila-4:fila+4,col-3:col+2)=1;
figure, imshow(C)
```

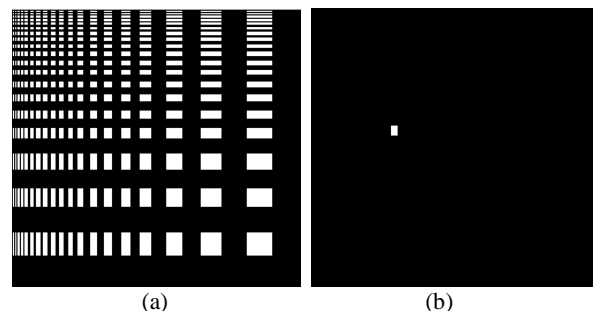


Figura 81. Imagen de rectángulos. (b) Rectángulo detectado.



## Escuela Técnica Superior de Ingeniería Informática Universidad de Málaga



e) Describe un algoritmo que detecte la célula de la figura 82 y que determine su perímetro.

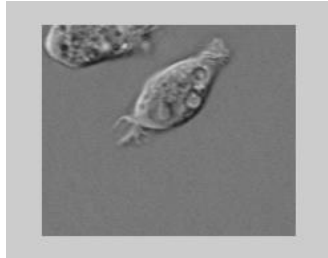


Figura 82. Célula.

```
I = imread('cell.tif');  
imshow(I)  
B = edge(I, 'sobel', (graythresh(I)*.1));  
figure, imshow(B)  
h90 = strel('line', 3, 90);  
h0 = strel('line', 3, 0);  
Bd = imdilate(B, [h90 h0]);  
figure, imshow(Bd)  
Br = imfill(Bd, 'holes');  
figure, imshow(Br)  
Bs = imclearborder(Br, 4);  
figure, imshow(Bs)  
hd = strel('diamond', 1);  
Bfinal = imerode(Bs, hd);  
figure, imshow(Bfinal)  
BL = bwperim(Bfinal);  
Segout = I; Segout(BL) = 255;  
figure, imshow(Segout)
```

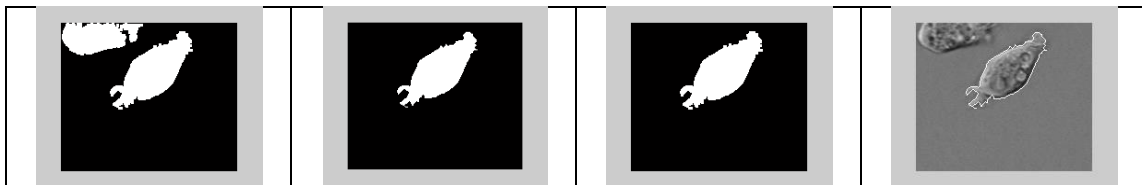
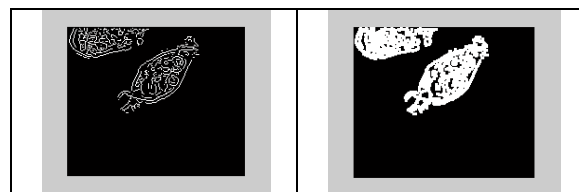


Figura 83. Secuencia de imágenes obtenidas por el algoritmo.

f) Corrige el efecto de una iluminación no uniforme en la imagen de la figura 84.

```
I=imread('rice.tif');  
J=im2double(I);  
imshow(J)  
B=blkproc(J, [32 32], 'min(x(:))');  
B1=imresize(B, [256 256], 'bicubic');  
imshow(B1);  
D=J-B1;  
imshow(D)  
A=imadjust(D, [0 max(D(:))], [0 1], 1);  
imshow(A)
```

```
% Es una aplicación directa del  
% operador TopHat  
I=imread('rice.tif');  
se=strel('disk', 10);  
U=imtophat(I, se)  
imshow(U)
```

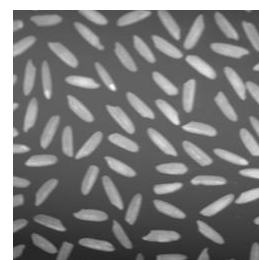


Figura 84. Imagen con iluminación no uniforme.

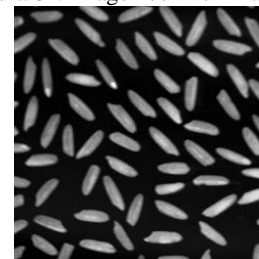
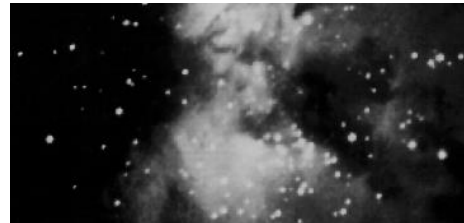


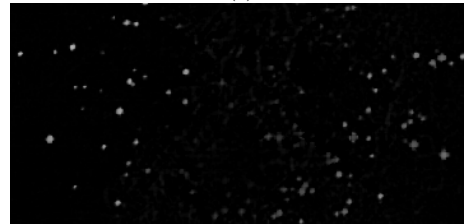
Figura 85. Imagen corregida.

g) Consigue una imagen constituida sólo por los objetos de la figura 86 (a) que están contenidos en un disco de radio tres píxeles utilizando operadores morfológicos.

```
I=imread('ngc4024m.tif');
imshow(I)
se = strel('disk',3);
E = imerode(I,se);
D = imdilate(E,se);
%D=imtophot(I);
F=I-D;
figure, imshow(F)
```



(a)



(b)

Figura 86. (a) Nebulosa. (b) Imagen con los objetos más pequeños.

## 15. Segmentación

a) Describe un algoritmo que determine el número de objetos de la imagen de la figura 87.

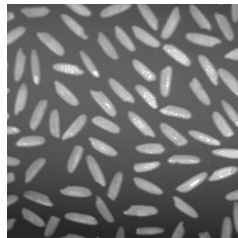
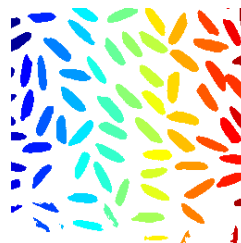


Figura 87. Granos de arroz.

```
I=imread('rice.tif');
nivel=graythresh(I);
B=im2bw(I,nivel);
figure,imshow(B)
whos
[etiquetas, nobjetos]=bwlabel(B,4);
nobjetos
E=label2rgb(etiquetas);
figure, imshow(E)
```



(a)



(b)

Figura 88. (a) Imagen binaria. (b) Imagen etiquetada.

b) Determina las *componentes conectadas* de la imagen de la figura 89(a) y el centro de cada una de las regiones correspondientes.

```
I=imread('coins.png');
nivel=graythresh(I);
B=im2bw(I,nivel);
figure,imshow(B);
[etiquetas,nobjetos]=bwlabel(B,4);
E=label2rgb(etiquetas);
figure, imshow(E)
%Determinación del centro
for k=1:nobjetos
[filas col]=find(etiquetas==k);
cy=mean(filas);
cx=mean(col);
```

```
plot(cx,cy,'Marker','o',...
'MarkerEdgeColor','k',...
'MarkerFaceColor','k',...
'MarkerSize',10)
hold on
plot(cx,cy,'Marker','*',...
'MarkerEdgeColor','w')
end
```

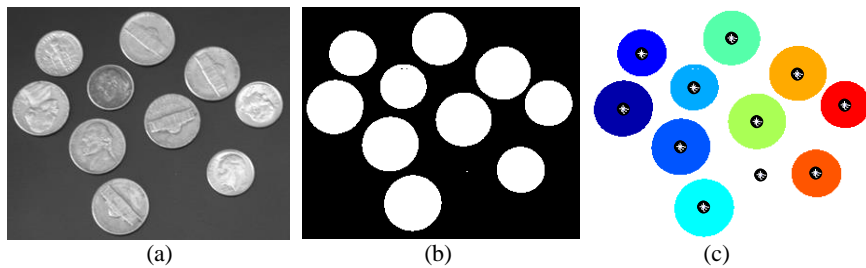


Figura 89 (a) Monedas. (b) Imagen binaria. (c) Imagen etiquetada y centros.

c) Identifica las monedas de la imagen de la figura 89 cuyo radio tiene una longitud de 29 a 30 píxeles mediante la *Transformada de Hough*.

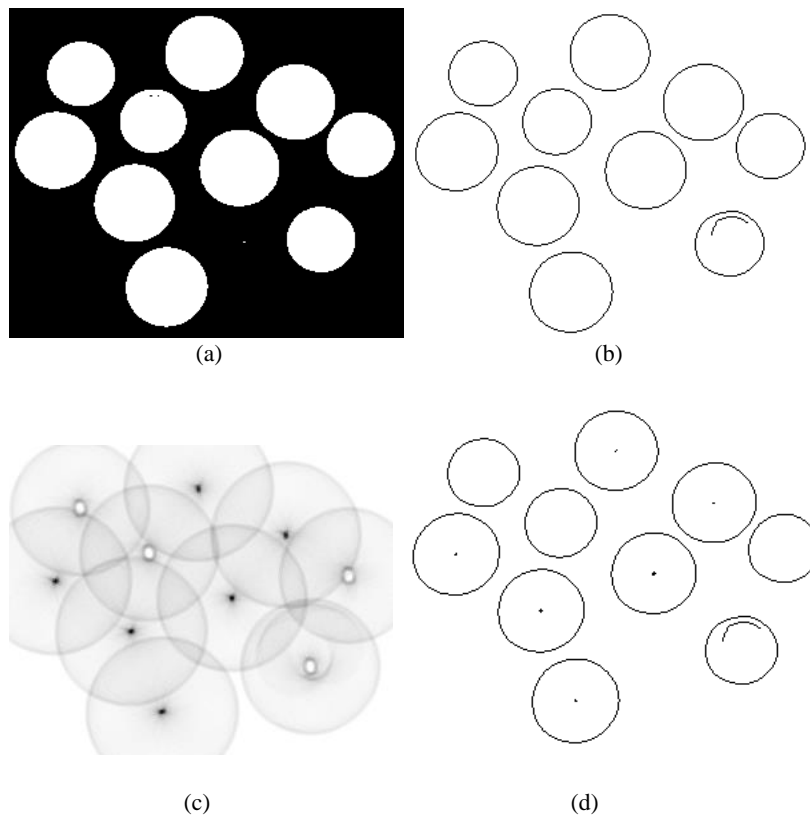


Figura 90. (a) Imagen Binaria. (b) Imagen de contornos.  
(c) Imagen de los acumuladores. (d) Imagen con los centros de las monedas identificadas.