

PROCESAMIENTO DE IMÁGENES

PRÁCTICA 8: RESTAURACIÓN DE IMÁGENES DEGRADAS O DISTORSIONADAS

¿CÓMO REALIZAR LA PRÁCTICA?: Para lograr la máxima puntuación, en cada ejercicio:

- Utiliza una imagen DIFERENTE a la propuesta.
- Añade comentarios al código.
- Responde a todas las preguntas y si no hubiera preguntas, comenta los resultados.
- Presenta los resultados legibles y ordenados.

El incumplimiento de alguna de estas reglas podrá restar puntuación de la nota de la práctica por cada ejercicio en el que se incumpla. Sube al CV un documento con cada uno de los apartados resueltos.

- 1) Restaura una imagen movida y suavizada como la de la siguiente figura y calcula el ECM de restauración. ¿Qué método de restauración estamos usando?, ¿por qué no es nulo el ECM? Prueba a variar el grado de suavizado, ¿varía significativamente el ECM? ¿Por qué?

```
I0=imread('Fig5.26(a).jpg');
I=im2double(I0);
figure, imshow(I)

[M,N]=size(I);
F=fft2(I);
h1=fspecial('gaussian',7,20);
H1=fft2(h1,M,N);
long=100;
ang=135;
h2=fspecial('motion',long,ang);
H2=fft2(h2,M,N);
G=H1.*H2.*F;
I1=real(ifft2(G));
figure,imshow(I1)

HR=(1./(H1.*H2));
G_rest=HR.*G;
I2=real(ifft2(G_rest));
figure,imshow(I2)
ECM=sum((I2-I).^2,"all")
```



- 2) a) Añade ruido uniforme antes de desplazar una imagen y restáurala usando un filtro inverso, ¿se restaura sin ruido? ¿Por qué?

b) Y si añades ruido después de desplazar la imagen, ¿hay alguna diferencia con el caso anterior? ¿Por qué?

```
%% 2.a)

I0=imread('Fig5.26(a).jpg');
I=im2double(I0);
ruido=0.1*randn(size(I));
B=I+ruido;
figure, imshow(B)
F=fft2(B);
long=100;
ang=135;
h=fspecial('motion',long,ang);
[M,N]=size(I);
H=fft2(h,M,N);
G=H.*F;
I1=real(ifft2(G));
figure, imshow(I1)
HR=(1./H);
G_rest=HR.*G;
I2=real(ifft2(G_rest));
figure, imshow(I2)
```

```
%% 2.b)

I0=imread('Fig5.26(a).jpg');
I=im2double(I0);
F=fft2(I);
long=100;
ang=135;
h=fspecial('motion',long,ang);
[M,N]=size(I);
H=fft2(h,M,N);
G=H.*F;
I1=real(ifft2(G));
figure, imshow(I1)
ruido=0.1*randn(size(I));
B=I1+ruido;
figure, imshow(B)
F2=fft2(B);
HR=(1./H);
G_rest=HR.*F2;
I1=real(ifft2(G_rest));
figure, imshow(I1)
```



- 3) Distorsiona una imagen utilizando una función de distorsión que produzca el efecto de “imagen movida” y corrígela mediante el filtro Wiener. Prueba dos desplazamientos diferentes y comenta las diferencias si las hubiera.

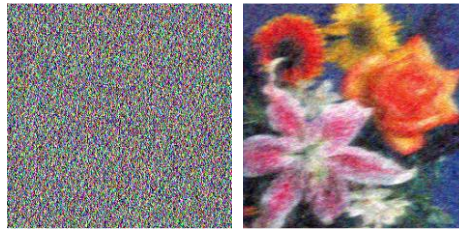
```
I0=imread('concordaerial.png');
I=im2double(I0);
long=31;
ang=11;
h=fspecial('motion',long,ang);
B=imfilter(I,h,'circular','conv');
figure, imshow(B)
W=deconvwnr(B,h);
figure, imshow(W)
```



- 4) Restaura una imagen movida que tiene añadido además ruido gaussiano utilizando primero el filtro inverso y después el filtro de Wiener. ¿Ambos restauran bien la imagen? ¿Por qué?

```
I=imread('flowers.tif');
I=I(10+[1:256],222+[1:256],:);
%Cogemos una matriz vinculada
%cuadrada => debido a una limitación de
%de deconvwnr
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)
```





- 5) Restaura una imagen que ha sido suavizada con un filtro gaussiano utilizando el método de desconvolución a ciegas (algoritmo de Lucy-Richardson). ¿Cuál es la matriz/función de degradación? ¿ha sido necesario conocer el valor concreto de dicha matriz para restaurar la imagen? ¿por qué?

→EN ESTE EJERCICIO **NO** DEBES EXPLICAR EL CÓDIGO.

```
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)
```



...



...

