

## Tarea 2

### Filtrado en el dominio de la frecuencia

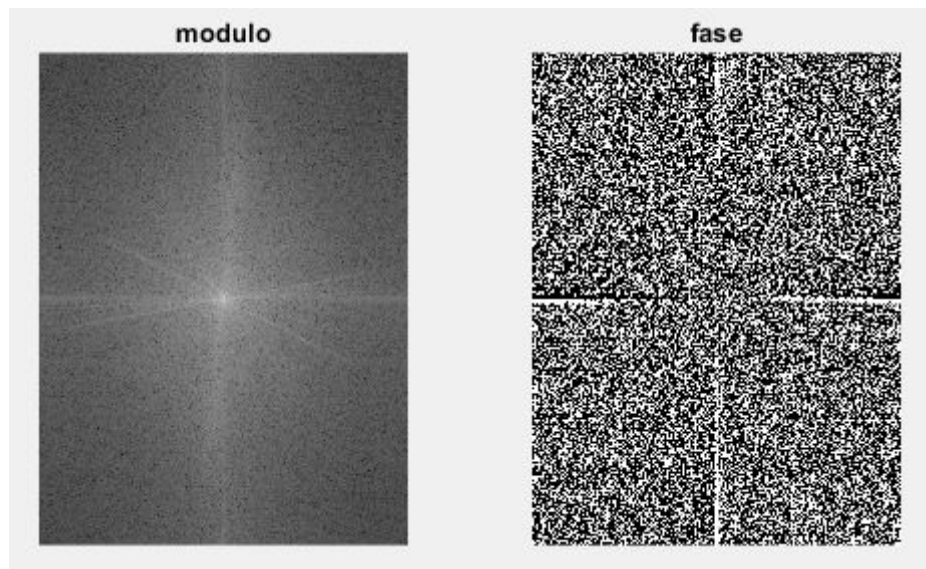
### INFORME

**Integrante 1:** Ana Poveda García Herrero.

**Integrante 2:** Cristina Taboada Mayo.

**Objetivo de la parte obligatoria:** Obtener la FFT de la imagen original, obtenga el módulo y la fase. Aplique, si lo considera oportuno, alguna transformación para visualizar mejor el módulo de la FFT. Justifique en qué componentes frecuenciales se encuentra la mayor parte de la energía. Además, se pide obtener una imagen binaria donde sólo aparezcan como primer plano los píxeles asociados a los cambios espaciales de intensidad.

El primer objetivo de esta parte eran obtener el módulo y la fase de la FFT de la imagen original y visualizarlas. Para ello, primero hemos cargado la imagen con el comando `imread()`. A continuación, para realizar el cálculo de la FFT necesitamos una imagen en 2 dimensiones, en este caso, nuestra imagen es de 3 dimensiones, por lo tanto la transformamos a una imagen en escala de grises que tendrá valores de 0 a 255. Con la imagen obtenida realizamos la FFT con el comando `fft2()` pasando como parámetros la imagen (de tipo double) y sus dimensiones espaciales, y a su vez al resultado se le aplica el comando `fftshift()` que centra las bajas frecuencias en el espectro. (Anexo 1.1) Tras obtener la FFT de la imagen, calculamos su módulo y su fase y finalmente los visualizamos. *Figura 1.*



*Figura 1. Visualización de Módulo y Fase de la FFT imagen original*

Para visualizar correctamente el módulo de la FFT de la imagen, hemos aplicado una transformación logarítmica ( $\log_{10}(1+FFT\_modulo)$ ), para aumentar el contraste y que se aprecie mejor.

El siguiente objetivo, obtener imagen binaria donde solo aparecen como primer plano los píxeles asociados a los cambios espaciales de intensidad. Primero de todo, creamos un

filtro paso bajo (con el comando *lpfilter()*), a partir de cual creamos el filtro paso alto calculando su inverso,  $1-H$ . Con el filtro paso alto creado calculamos su módulo y fase, y lo visualizamos. *Figura 2.*



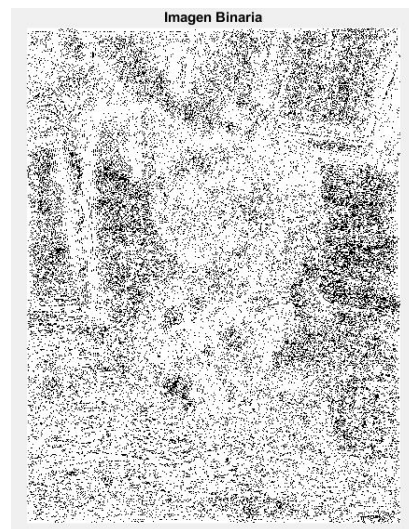
*Figura 2. Visualización del módulo y fase del filtro paso alto.*

Tras esto, hacemos el filtrado de la FFT de la imagen multiplicando pixel a pixel por el filtro paso alto, obteniendo así nuestra imagen filtrada en frecuencia. Finalmente hacemos la transformada rápida de Fourier inversa, para recomponer la imagen y poder visualizarla. Añadimos el valor absoluto ya que al estar en escala de grises los niveles están limitados de 0 - 255, y no valores negativos. *Figura 3.*



*Figura 3. Visualización de la imagen filtrada paso alto.*

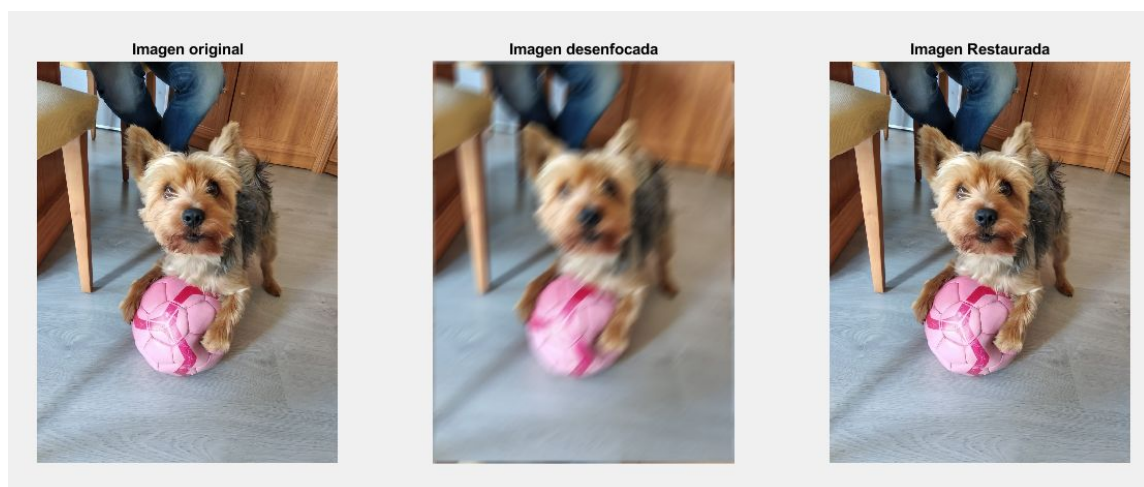
Finalmente, umbralizamos la imagen y visualizamos el resultado final. *Figura 4.*



*Figura 4. Visualización de la imagen umbralizada.*

**Objetivo de la parte creativa:** Obtener la imagen restaurada a partir de una imagen con desenfoque de movimiento.

En esta parte hemos decidido restaurar una imagen utilizando el filtro de Wiener. En primer lugar, cargamos la imagen y la transformamos a tipo *double*, y creamos la imagen degradada con un efecto de movimiento y desenfoque. La degradación de la imagen se consigue creando la PSF mediante el comando *fspecial()* introduciendo como parámetros el efecto 'motion', LEN (número de píxeles con movimiento lineal) y THETA (ángulo de desenfoque). Tras esto, aplicamos la PSF a nuestra imagen, obteniendo la imagen desenfocada mediante convolución circular. Para restaurar la imagen original a partir de la imagen desenfocada, utilizamos el filtro de Wiener ya que este es utilizado cuando tenemos ruido y distorsión de tipo difuminado, esto se hace mediante el comando *deconvwnr()* pasándole como parámetros la imagen desenfocada, la PSF utilizada para la degradación y considerando la NSR nula. (Anexo 1.2) Finalmente visualizamos los resultados. *Figura 5.*



*Figura 5. Visualización de la imagen Original, la imagen desenfocada y la imagen restaurada.*



## ANEXO:

### Anexo 1.1:

```
%% ----- PARTE OBLIGATORIA -----
clear all, close all, clc
% Cargamos la imagen.
I = imread('G15.jpeg');
figure, imshow(I), title('Imagen original')

% Transformamos la imagen a escala de grises.
X = rgb2gray(I);
figure, imshow(X), title('Imagen original en escala de grises')

% Calculo de la FFT de la imagen original.
X_FFT = fftshift(fft2(double(X), 1600, 1200));

% Calculo del módulo y la fase de la FFT (esto lo calculamos para poder
% visualizarlos, ya que la FFT no puede visualizarse directamente)

FFT_modulo = abs(X_FFT); % Calcula el modulo del resultado de la FFT
FFT_fase = angle(X_FFT); % Calcula la fase del resultado de la FFT

% Visualizamos el modulo y la fase de la FFT.
% Para poder visualizar el modulo debemos hacer la siguiente
% transformación: log10(1+FFT_modulo), para aumentar el contraste, y
% visualizar mejor la imagen resultante.
figure, subplot(1,2,1), imshow(log10(1+FFT_modulo), []), title('modulo')
subplot(1,2,2), imshow(FFT_fase), title('fase')

% Obtener imagen binaria donde solo aparecen como primer plano los píxeles
% asociados a los cambios espaciales de intensidad.
clear all, close all, clc

% Filtrado paso alto con el filtro ideal.
H = lpfilter('ideal', 1600, 1200, 100);

H = 1-H; % Con este comando, transformamos el FPB en el FPA.
FFT_moduloH = abs(H);
FFT_faseH = angle(H);
figure, mesh(FFT_moduloH), title('Mesh del FFT Modulo del FPA')
figure, mesh(FFT_faseH), title('Mesh del FFT Fase del FPA')
figure, subplot(1,2,1), imshow(log10(1+FFT_moduloH), []), title('modulo FPA')
subplot(1,2,2), imshow(FFT_faseH), title('fase FPA')

% Cargamos la imagen
I = imread('G15.jpeg');

% Transformamos la imagen a escala de grises.
X = rgb2gray(I);

% Aplicamos el filtro paso alto a la imagen original.
F = fft2(double(X));
Filtrada_freq = H.*F;
FPA_ideal = abs(real(ifft2(Filtrada_freq)));
figure, imshow(FPA_ideal, []), title('Imagen Filtrada Paso Alto')

% Aplicamos umbralización para visualizar la imagen.
X_U = im2bw(FPA_ideal, 230/255);
figure, imshow(X_U), title('Imagen Binaria')
```

## Anexo 1.2:

```
%% ----- PARTE CREATIVA -----
clear all, close all, clc

% Cargamos la imagen y la pasamos a tipo double.
I = im2double(imread('G15.jpeg'));

% Simulamos un efecto de movimiento (desenfocado) en la imagen original.
LEN = 50; % Número de pixel con movimiento lineal.
THETA = 45; % Angulo de desenfoque
PSF = fspecial('motion', LEN, THETA); % Point Spread Function (Respuesta al impulso espacial, nuestra H)
desenfocado = imfilter(I, PSF, 'conv', 'circular'); % Aplica el filtro como una convolucion circular.

% Restauramos la imagen desenfocada utilizando un filtro de Wiener.
F_Wiener = deconvwnr(desenfocado, PSF, 0);

% Visualizamos el resultado.
figure, subplot(1,3,1), imshow(I), title('Imagen original')
subplot(1,3,2), imshow(desenfocado), title('Imagen desenfocada')
subplot(1,3,3), imshow(F_Wiener), title('Imagen Restaurada')
```