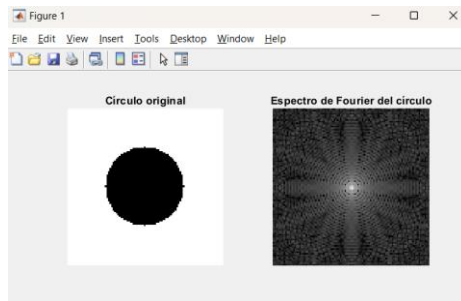


Práctica 5

10. Transformada de Fourier discreta

a) Determina la transformada de Fourier discreta de la imagen.



```
% Defino las dimensiones de la matriz
filas = 100;
columnas = 100;

% Creo una cuadrícula de coordenadas
[X, Y] = meshgrid(1:columnas, 1:filas);

% Defino el centro y el radio del círculo
centro_x = columnas / 2;
centro_y = filas / 2;
radio = min(filas, columnas) / 4; % Radio del círculo

% Calculo la máscara del círculo
circulo = ((X - centro_x).^2 + (Y - centro_y).^2 <= radio.^2);

% Invierto la máscara para que el círculo sea negro y el fondo blanco
circulo = ~circulo;

% Calculo la transformada de Fourier bidimensional
transformada = fft2(circulo);

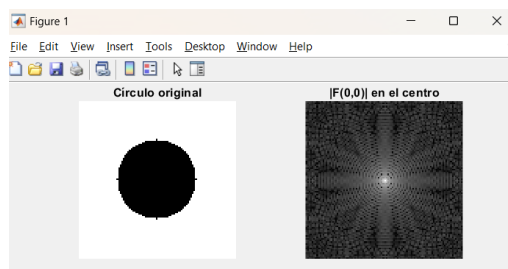
% Calculo el espectro de la transformada de Fourier
espectro = abs(fftshift(transformada));

% el círculo original y su espectro de Fourier en una sola figura
figure;

subplot(1, 2, 1);
imshow(circulo);
title('Circulo original');

subplot(1, 2, 2);
imshow(log(1 + espectro), []);
title('Espectro de Fourier del círculo');
```

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



Las zonas oscuras se pueden apreciar algo más

Oscuras en comparación con la del apartado a.

```
% Defino las dimensiones de la matriz
filas = 100;
columnas = 100;

% Creo una cuadrícula de coordenadas
[X, Y] = meshgrid(1:columnas, 1:filas);

% Defino el centro y el radio del círculo
centro_x = columnas / 2;
centro_y = filas / 2;
radio = min(filas, columnas) / 4; % Radio del círculo

% Calculo la máscara del círculo
circulo = ((X - centro_x).^2 + (Y - centro_y).^2 <= radio.^2);

% Invierto la máscara para que el círculo sea negro y el fondo blanco
circulo = ~circulo;

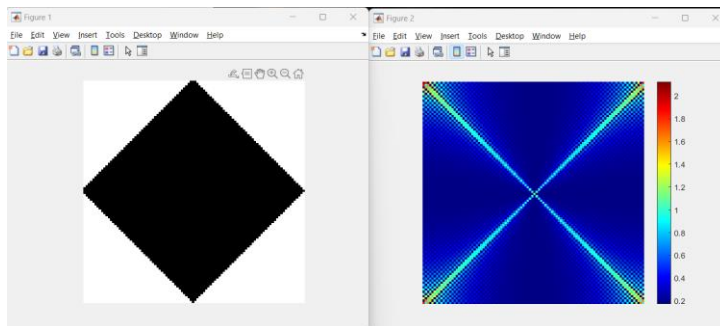
% Calculo la transformada de Fourier bidimensional del círculo
dft_circulo = fft2(circulo);

% Reorganizo las frecuencias para centrar [F(0,0)]
dft_centro = fftshift(dft_circulo);

% Calculo la magnitud de la DFT (opcional)
magnitud_dft_centro = abs(dft_centro);

% Visualizo el círculo original y su DFT con [F(0,0)] en el centro
subplot(1,2,1), imshow(circulo), title('Circulo original');
subplot(1,2,2), imshow(log(1 + magnitud_dft_centro), []), title('|F(0,0)| en el centro');
```

c) Determina la transformada de Fourier discreta de la imagen.



```
% Inicializo una matriz de unos
I = ones(100,100);

% Itero sobre cada pixel de la matriz
for i = 1:100
    for j = 1:100
        % Defino las coordenadas del centro del rombo
        centro_x = 50;
        centro_y = 50;

        % Calculo la distancia entre el pixel actual y el centro del rombo
        distancia_x = abs(i - centro_x);
        distancia_y = abs(j - centro_y);

        % Calculo la suma de las distancias
        suma_distancias = distancia_x + distancia_y;

        % Establezco el valor del pixel en función de la suma de las distancias
        if suma_distancias <= 50 % El rombo tendrá un tamaño de 100x100
            I(i,j) = 0;
        end
    end
end

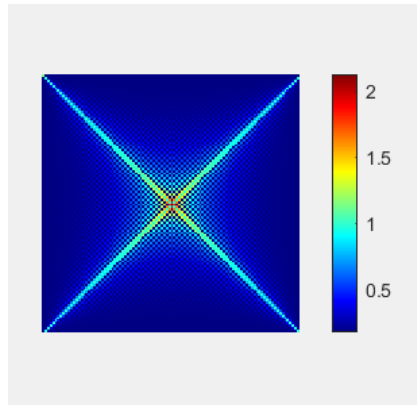
% Muestro el rombo
imshow(I, 'InitialMagnification', 'fit')

% Calculo la transformada de Fourier bidimensional
F = fft2(I);

% Calculo el logaritmo de la magnitud de la transformada de Fourier
F1 = log(1 + abs(F));

% Muestro la transformada de Fourier
figure;
imshow(F1 / 4, [], 'InitialMagnification', 'fit');
colormap(jet);
colorbar;
```

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



```
% Inicializo una matriz de unos
I = ones(100,100);

% Itero sobre cada pixel de la matriz
for i = 1:100
    for j = 1:100
        % Defino las coordenadas del centro del rombo
        centro_x = 50;
        centro_y = 50;

        % Calculo la distancia entre el pixel actual y el centro del rombo
        distancia_x = abs(i - centro_x);
        distancia_y = abs(j - centro_y);

        % Calculo la suma de las distancias
        suma_distancias = distancia_x + distancia_y;

        % Establezco el valor del pixel en función de la suma de las distancias
        if suma_distancias <= 50 % El rombo tendrá un tamaño de 100x100
            I(i,j) = 0;
        end
    end
end

% Muestro el rombo
imshow(I,'InitialMagnification','fit')

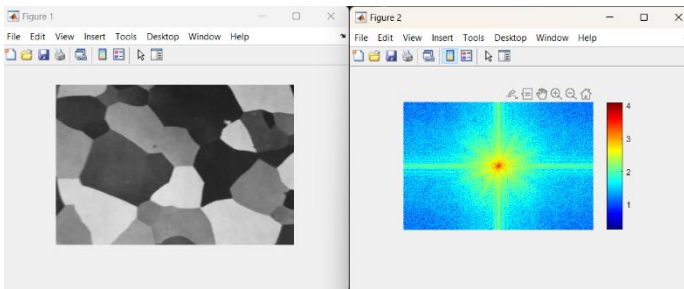
% Calculo la transformada de Fourier bidimensional de la imagen
F = fft2(I);

% Reorganizo las frecuencias para centrar |F(0,0)|
F_centro = fftshift(F);

% Calculo el logaritmo de la magnitud de la transformada de Fourier
F1 = log(1 + abs(F_centro));

% Muestro la transformada de Fourier con el valor de |F(0,0)| en el centro de la imagen
figure;
imshow(F1 / 4, []);
colormap(jet);
colorbar;
```

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



```
% Leo la imagen
imagen = imread('alumngrns.JPG');

% Convierto la imagen a escala de grises si es necesario
if size(imagen, 3) == 3
    imagen_gris = rgb2gray(imagen);
else
    imagen_gris = imagen;
end

% Muestro el rombo
imshow('alumngrns.JPG')

% Calculo la transformada de Fourier bidimensional de la imagen
F = fft2(imagen_gris);

% Reorganizo las frecuencias para centrar |F(0,0)|
F_centro = fftshift(F);

% Calculo el logaritmo de la magnitud de la transformada de Fourier
F1 = log(1 + abs(F_centro));

% Muestro la transformada de Fourier con el valor de |F(0,0)| en el centro de la imagen
figure;
imshow(F1 / 4, []);
colormap(jet);
colorbar;
```

f) Ejecuta el siguiente código y razona los valores que toma la transformada de la imagen.

Estamos ante una transformada de Fourier de una imagen sinusoidal que muestra componentes de baja frecuencia en el centro y componentes de alta frecuencia en los bordes. Los valores significativos se concentran en el centro debido a la naturaleza suave de la función sinusoidal.

Preguntas:**¿Qué hace *fft2*?**

Calcula la Transformada de Fourier Discreta Bidimensional (DFT) tomando una matriz de entrada de datos (en este caso I) y la convierte desde el dominio espacial al dominio de la frecuencia.

¿Para qué se genera la variable F1?

Calcula el logaritmo de la magnitud de DFT para obtener una representación visual de las frecuencias.

¿Hay diferencia si el rectángulo es vertical?

Si. Afectaría a la orientación de las frecuencias principales en la DFT y la distribución de energía en la transformada.

¿Qué cambio hay entre la Figura 42 y 43 del guion?

Al aplicarse la función *fftshift*, se ha dividido la matriz de frecuencias en cuatro cuadrantes, intercambiando los cuadrantes superior izquierdo con el inferior derecho e intercambiando los cuadrantes superior derecho con el inferior izquierdo.

¿Qué hace *fftshift*?

Es una función que se utiliza para cambiar el centro de la matriz de frecuencias al centro de la imagen, permitiendo que los valores de frecuencias bajas estén ubicados en el centro de la matriz, mientras que los valores de frecuencias altos estén ubicados en las esquinas de la matriz.

¿Y si en lugar de usar 1 se usa 10, 100, 0.1 o 0.001?

Al poner valores más grandes como 10 o 100 permite aumentar el contraste en la imagen resultante, haciendo que las zonas con frecuencias bajas sean más visibles.

Por otro lado, al usar valores más pequeños como 0.1 o 0.001 reduce el contraste, permitiendo suavizar la imagen resultante.