# introducción

Procesamiento digital de imágenes  
Práctica 5: “Interpolación”



El problema de construir una función continua a partir de datos discretos es inevitable cuando estos datos deben ser manipulados de cierta manera que se necesita información no incluida explícitamente. Para resolver este problema, el esquema más utilizado es la interpolación que consiste en contruir una función que aproxime de la manera más perfecta a la función original desconocida en los puntos de la medición.

La interpolación de imagenes es una opción muy importante usada en imagenelolgía médica, procesamiento de imágenes y graficación por computadora. Existe una gran variedad de métodos de interpolación reportados en la literatura. La interpolación de imágenes es necesaria en una gran variedad de situaciones como las que se mencionan a continuación:

1. Representar imagenes o volúmenes a un nivel deseado de discretización modificando para ello la tasa de muesteo de los pixeles o voxels.

2. Cambiar la orientación de alguna regilla de discretización.

3. Combinar la información sobre un mismo objeto desde múltiples modalidades en una sola imagen (fusión de imagenes). Por ejemplo, una resonancia magnética y una tomografía por emisiones de positrones.

4. Cabio de de rijilla de discretización, por ejemplo de polares rectangulares.

Probablemente el uso más común de la interpolación sea la mencionada en el punto numero 1, donde por lo regular se desean analizar ciertos detalles de una imagen a una escala y otros detalles a una escala más fina (zoom).

Algunas transformaciones pueden involucrar un cambio de coordenadas, por ejemplos, la función de conversión de coordandas polares, adquiridas a través de un transductor de ultrasonido, a coordenadas cartesianas necesario para la visualización de la imagen en un monitor. En general, casi cualquier transformación geométrica sobre una imagen o un volumen necesita que se efectué una interpolación.

La calidad de la imagen o volumen obtenidos dependerá del proceso utilizado para realizar la interpolación así como también del trabajo necesario para que una computadora lo ejecute en un tiempo razonable.

# DESARROLLO

Para todos los puntos siguientes y con la finalidad de poder observar el desempeño de los distintos interpoladores usar una imagen nítida de baja resolución. Se recomienda usar imágenes desde 128x128 hasta 256x256 pixeles como máximo y la imagen pentagon que se encuentra disponible en la sección de imágenes del curso.

1. Sobremuestro espacial: Obtenga el sobremuestreo de la imagen original insertando ceros entre los pixeles de la misma con factores T ↑= 2×2 y T ↑= 4×4.

image = imread('pentagon256x256.tif');

S\_T2 = zeros(size(image,2)\*2);

for x = 1:size(image,2)

for y = 1:size(image,2)

S\_T2(2\*x-1,2\*y-1) = image(x,y);

end

end

S\_T4 = zeros(size(image,2)\*4);

for x = 1:size(image,2)

for y = 1:size(image,2)

S\_T4(4\*x-3,4\*y-3) = image(x,y);

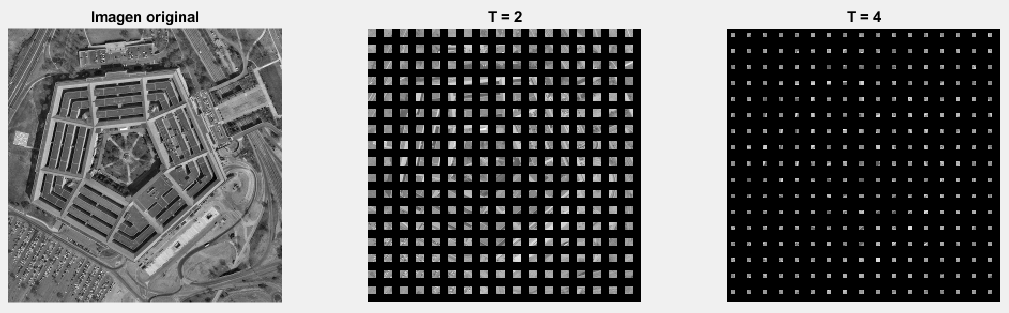
end

end

subplot(1,3,1),imshow(image,[]),title('Imagen original')

subplot(1,3,2),imshow(S\_T2,[]),title('T = 2')

subplot(1,3,3),imshow(S\_T4,[]),title('T = 4')



*Figura 1: Sobremuestreo espacial*

Interpretación:

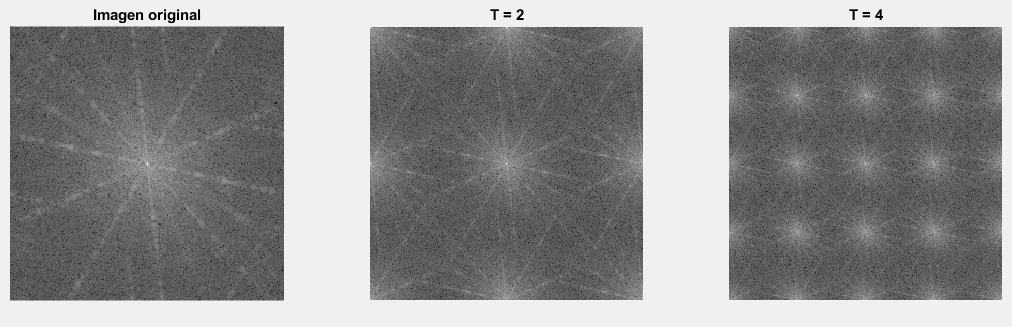
Apartir de la imagen original, generamos dos sobremuestros con el objetivo de agergar espacio para poder inteprolar.

* Obtenga la magnitud del espectro de la DFT (abs) de cada una de las imágenes
* sobremuestreadas y de la imagen original. Despliegue los resultados en una misma figura para efectos de comparación (en Matlab se puede usar el comando subplot). Recuerde usar fftshift para centrar los espectros y una función de escalamiento para el despliegue, ejemplo: ImFDespliegue=log(1.0 + ImF), donde ImF es la DFT de la imagen con sobremuestreo.

subplot(1,3,1),imshow(fftshift(log(abs(fft2(image)))),[]),title('Imagen original')

subplot(1,3,2),imshow(fftshift(log(abs(fft2(S\_T2)))),[]),title('T = 2')

subplot(1,3,3),imshow(fftshift(log(abs(fft2(S\_T4)))),[]),title('T = 4')



*Figura 2: Sobremuestreo en frecuencia*

Interpretación:

Cuando transformamos las imagenes a su espectro en frecuencia, observamos que

la insersión de ceros en la imagen en su espectro oringial, general en frecuencia imagenes

‘aisladas‘

* ¿Qué sucedería si se intercalaran ceros entre los valores del espectro de la DFT?

Se generaría un resultado diferente, ya que en lugar de separar la imagen en subconjuntos, se parariamos los elementos del espectro en frecuencia.

1. Interpolación espacial. Interpole las imágenes con sobremuestreo obtenidas en el inciso anterior (con factores T ↑= 2×2 y T ↑= 4×4) usando interpoladores:

* De orden cero

f\_2 = ones(2);

f\_4 = ones(4);

%Orden cero T=2:

temp = conv2(S\_T2,f\_2,'full');

Z\_T2 = temp(1:size(S\_T2,2),1:size(S\_T2,2));

%Orden cero T=4:

temp = conv2(S\_T4,f\_4,'full');

Z\_T4 = temp(1:size(S\_T4,2),1:size(S\_T4,2));

figure(),

subplot(1,2,1),imshow(Z\_T2,[]),title('T = 2')

subplot(1,2,2),imshow(Z\_T4,[]),title('T = 4')



*Figura 3: Orden cero*

* Lineal

f\_2 = [1/2 1 1/2]'\*[1/2 1 1/2];

f\_4 = [1/4 1/2 3/4 1 3/4 1/2 1/4]'\*[1/4 1/2 3/4 1 3/4 1/2 1/4];

%Lineal para T=2

temp = conv2(S\_T2,f\_2,'full');

L\_T2 = temp(1:size(S\_T2,2),1:size(S\_T2,2));

%Lineal para T=4

temp = conv2(S\_T4,f\_4,'full');

L\_T4 = temp(1:size(S\_T4,2),1:size(S\_T4,2));

figure(),

subplot(1,2,1),imshow(L\_T2,[]),title('T = 2')

subplot(1,2,2),imshow(L\_T4,[]),title('T = 4')



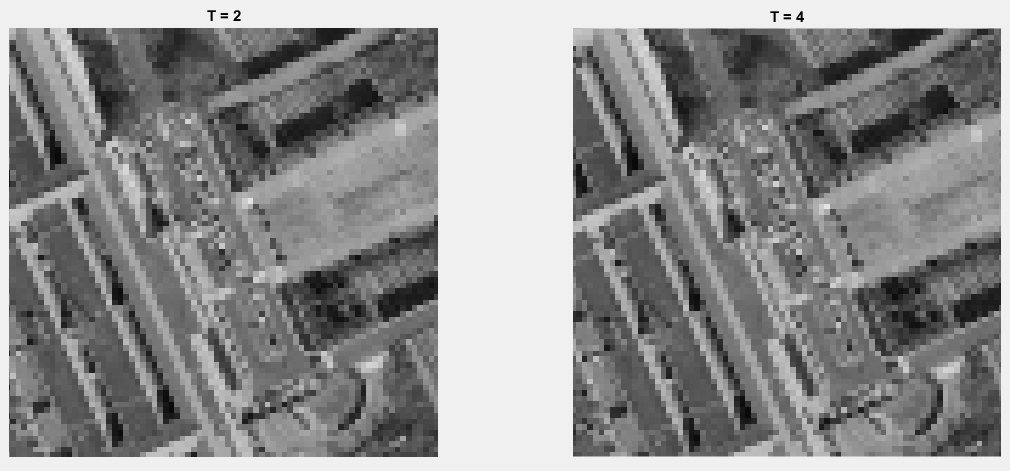
*Figura 4: Lineal*

Interpretación de figura 3 y 4:

Observamos que la corrección, mediante interpolación, de las imagenes ayuda a reducir a aparición de pixeles, sin embargo se genera una atenuación el las imagnes debido a la ponderación que existe al calcular los pixeles intermedios

* Para cada caso y factor de interpolación despliegue con alguna función de acercamiento (zoom) una misma región seleccionada de las imágenes interpoladas. Compare y explique los resultados de los distintos interpoladores en una misma figura.

Orden Cero



*Figura 5: Orden cero con zoom*

Lineal



*Figura 6: Lineal con zoom*

Interpretación de imagen 5 y 6:

Del mismo modo con en la interpretación previa, se genera una atenuación debido a

el calculo ponderado de los pixeles intermedios. Sin embargo en esta ocasión, existe tanto daño que es complicado no generar artefactos.

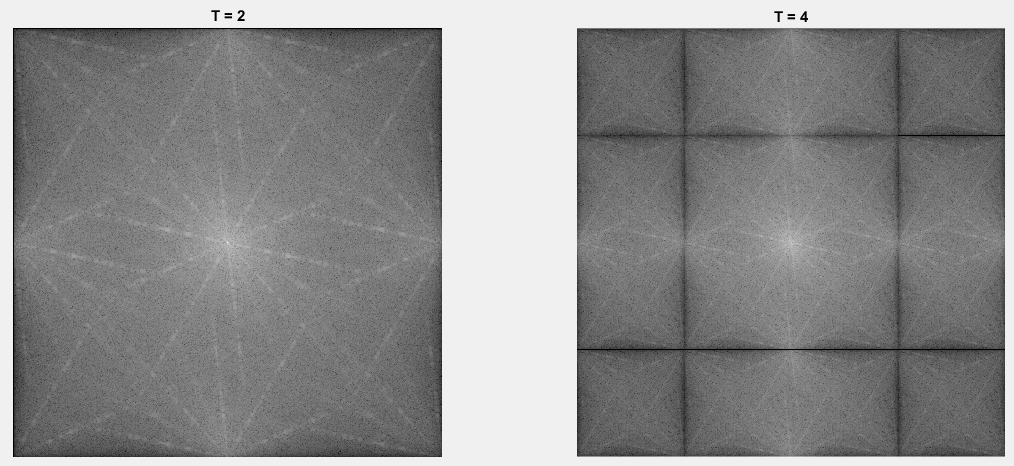
* Obtenga la magnitud del espectro de la DFT (abs) de cada una de las imágenes interpoladas. Despliegue los espectros en una misma figura para compararlos. Explique el efecto de los distintos tipos de interpolación en los espectros de las DFTs.

Orden Cero en frecuencia

figure(),

subplot(1,2,1),imshow(fftshift(log(abs(fft2(Z\_T2)))),[]),title('T = 2')

subplot(1,2,2),imshow(fftshift(log(abs(fft2(Z\_T4)))),[]),title('T = 4')



*Figura 7: Orden cero en frecuencia*

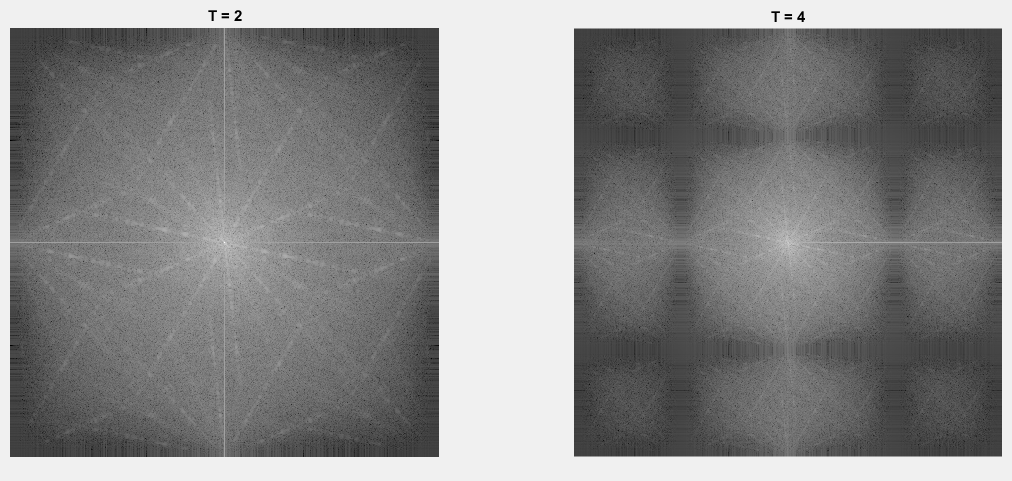
Lineal

%En frecuencia

figure(),

subplot(1,2,1),imshow(fftshift(log(abs(fft2(L\_T2)))),[]),title('T = 2')

subplot(1,2,2),imshow(fftshift(log(abs(fft2(L\_T4)))),[]),title('T = 4')



*Figura 8: Lineal en frecuencia*

1. Interpolación en frecuencia:

* Obtenga la DFT de la imagen original y despliegue su magnitud.

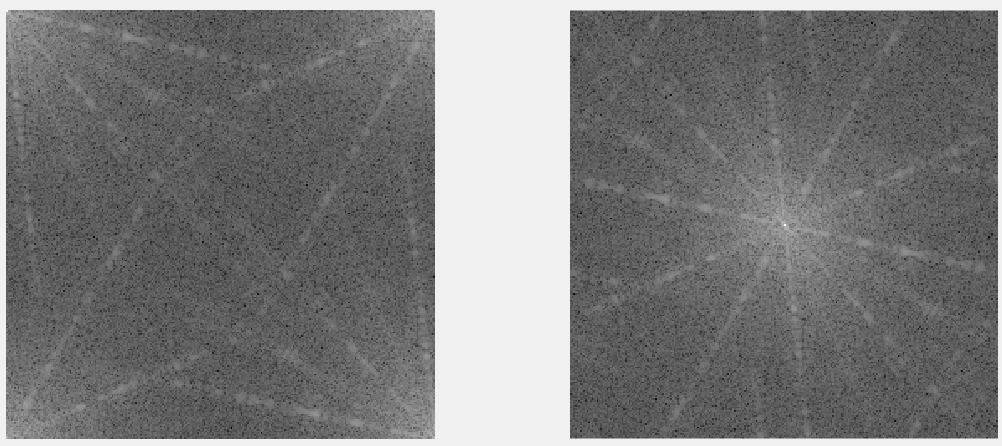
Interpolacion\_frecuencia = fft2(image);

%Despliegue de espectros, centrado y no centrado.

figure('Name','DFT imagen original.') ,

subplot(1,2,1),imshow(log(abs(Interpolacion\_frecuencia)),[])

subplot(1,2,2),imshow(fftshift(log(abs(Interpolacion\_frecuencia))),[])



*Figura 9: DFT imagen original (No centrada)*

Interpretación:

En este caso vemos la insersión de ceros dentro de los espectros de frecuencia en lugar de las

imagenes, sin embargo observamos diferencias en las franjas de ceros teninendo en la primer

un ingreso de linea solida y en la segunda un barrido degradado

* Centre el espectro de la DFT (fftshift) y agregue ceros alrededor del mismo hasta completar y obtener dos espectros cuyas dimensiones correspondan a las dimensiones de la imagen original multiplicadas por los factores de interpolación T ↑= 2×2 y T ↑= 4×4 respectivamente.

aux = fftshift(Interpolacion\_frecuencia);

%Interpolacion en frecuencia con T=2:

I\_frec\_t2 = complex(zeros(size(image,2)\*2));

I\_frec\_t2(size(image,2)/2+1:size(image,2)+size(image,2)/2,size(image,2)/2+1:size(image,2)+size(image,2)/2) = aux;

I\_frec\_t2\_final = ifft2(I\_frec\_t2);

%Interpolación en frecuencia con T=4:

I\_frec\_t4 = complex(zeros(size(image,2)\*4));

I\_frec\_t4(385:640,385:640) = aux;

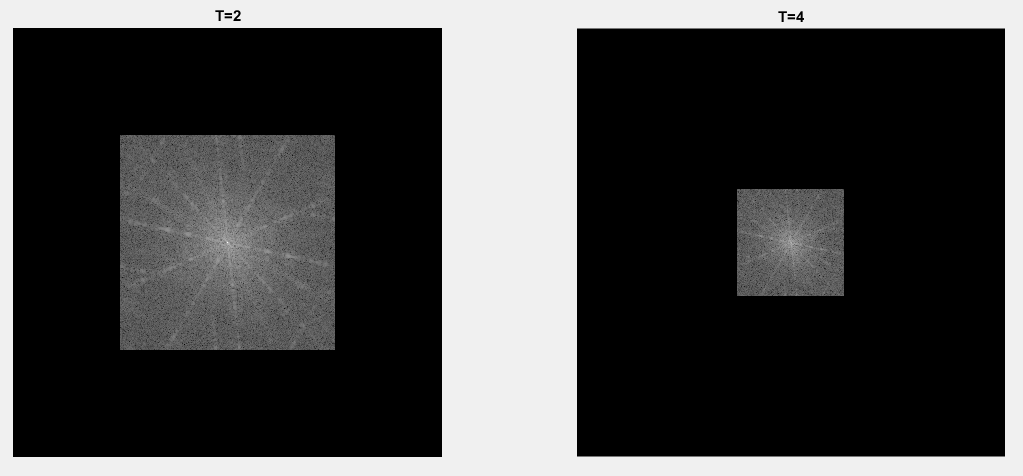
I\_frec\_t4\_final = ifft2(I\_frec\_t4);

* Despliegue las magnitudes de las DFTs (abs) con ceros alrededor. Use una función de escalamiento para el despliegue.

figure('Name','DFT con ceros alrrededor.'),

subplot(1,2,1),imshow(log(abs(I\_frec\_t2)),[]),title('T=2')

subplot(1,2,2),imshow(log(abs(I\_frec\_t4)),[]),title('T=4')



*Figura 10: DFT Interpolacion en frecuencia*

* Compare en una misma figura las magnitudes de las DFTs (abs) de las imágenes interpoladas en el punto 2 con la magnitudes de las DFTs (abs) con ceros alrededor. Explique los resultados y diferencias.

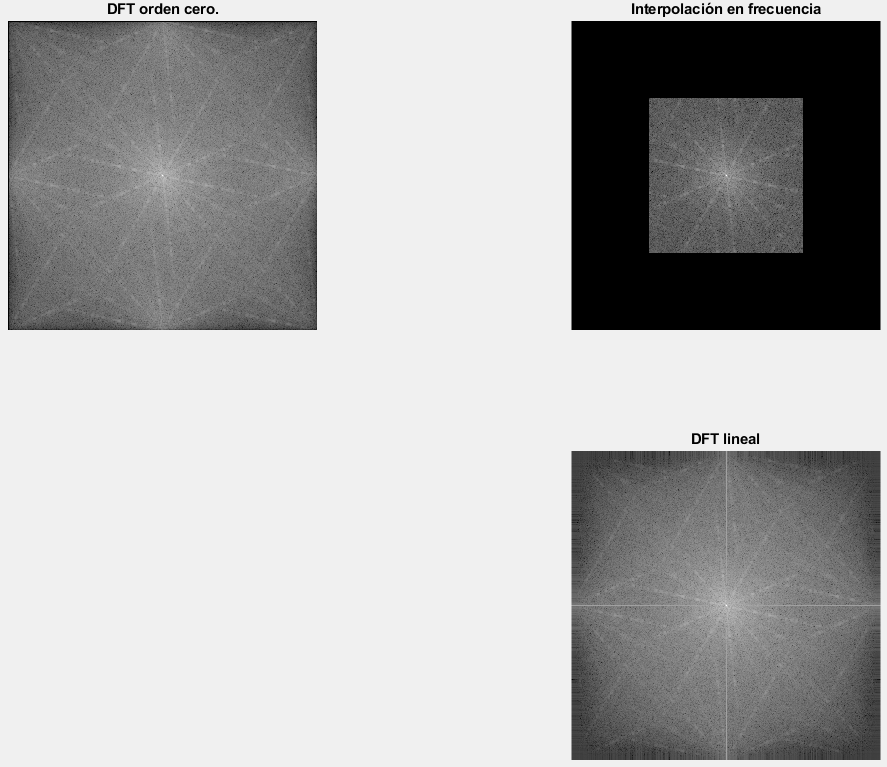
%Comparativa de las DFT T=2.

figure('Name','Comparando DFT cuando T=2.'),

subplot(2,2,1),imshow(fftshift(log(abs(fft2(Z\_T2)))),[]),title('DFT orden cero.')

subplot(2,2,2), imshow(log(abs(I\_frec\_t2)),[]),title('Interpolación en frecuencia')

subplot(2,2,4), imshow(fftshift(log(abs(fft2(L\_T2)))),[]),title('DFT lineal')



*Figura 11: Varios DFT cuando T=2*

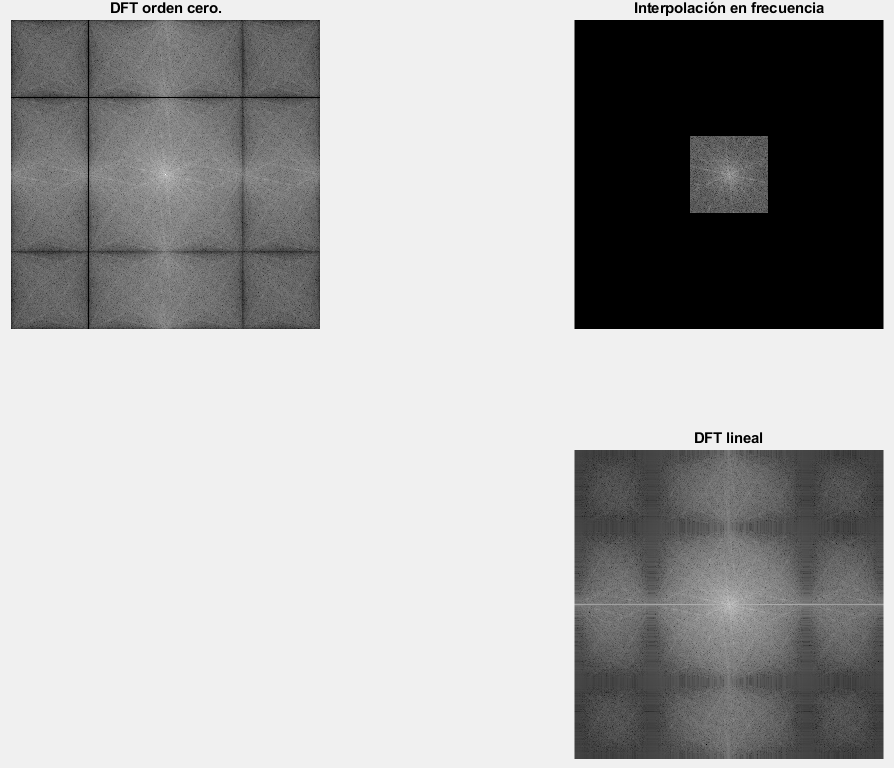
%Comparativa de las DFT T=4.

figure('Name','Comparando DFT cuando T=4.'),

subplot(2,2,1),imshow(fftshift(log(abs(fft2(Z\_T4)))),[]),title('DFT orden cero.')

subplot(2,2,2),imshow(fftshift(log(abs(fft2(L\_T4)))),[]),title('DFT lineal')

subplot(2,2,4),imshow(log(abs(I\_frec\_t4)),[]),title('Interpolación en frecuencia')



*Figura 12: Varios DFT cuando T=4*

* Calcule la inversa de la DFT (IDFT) de las DFTs con ceros alrededor.

%Interpolación en frecuencia

figure('Name','IDFT en frecuencia.'),

subplot(1,2,1),imshow(abs(I\_frec\_t2\_final),[]),title('T = 2')

subplot(1,2,2),imshow(abs(I\_frec\_t4\_final),[]),title('T = 4')



*Figura 13: IDFT en frecuencia*

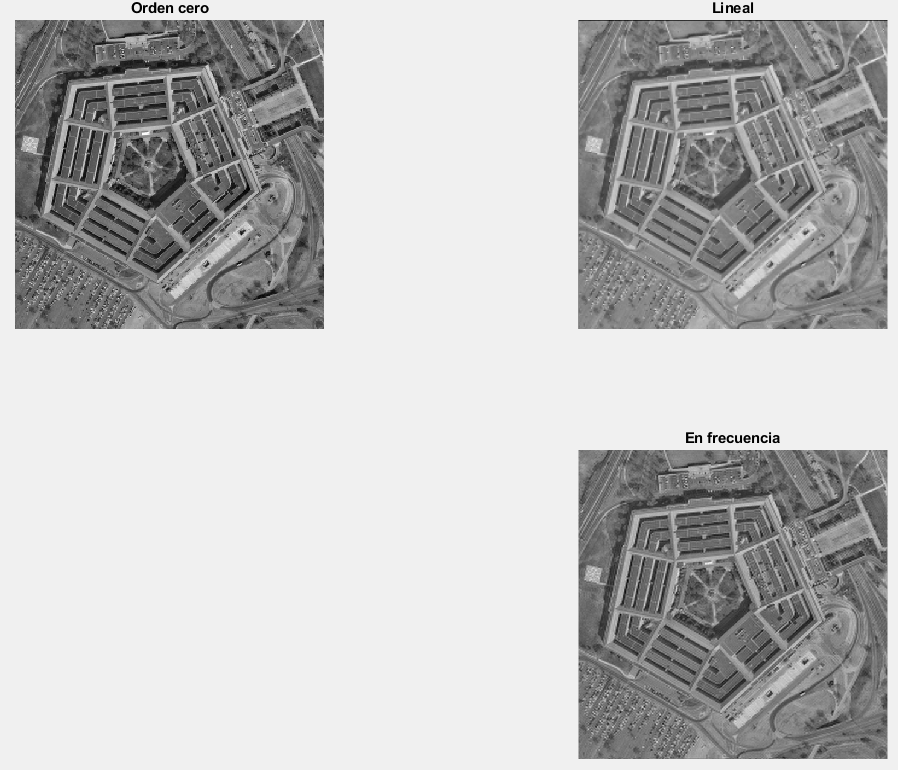
1. Compare en una misma figura los resultados de las imágenes interpoladas que se obtuvieron con los cuatro métodos

figure('Name','Comparando interpolación T=2.'),

subplot(2,2,1),imshow(Z\_T2,[]),title('Orden cero')

subplot(2,2,2),imshow(L\_T2,[]),title('Lineal')

subplot(2,2,4),imshow(abs(I\_frec\_t2\_final),[]),title('En frecuencia')



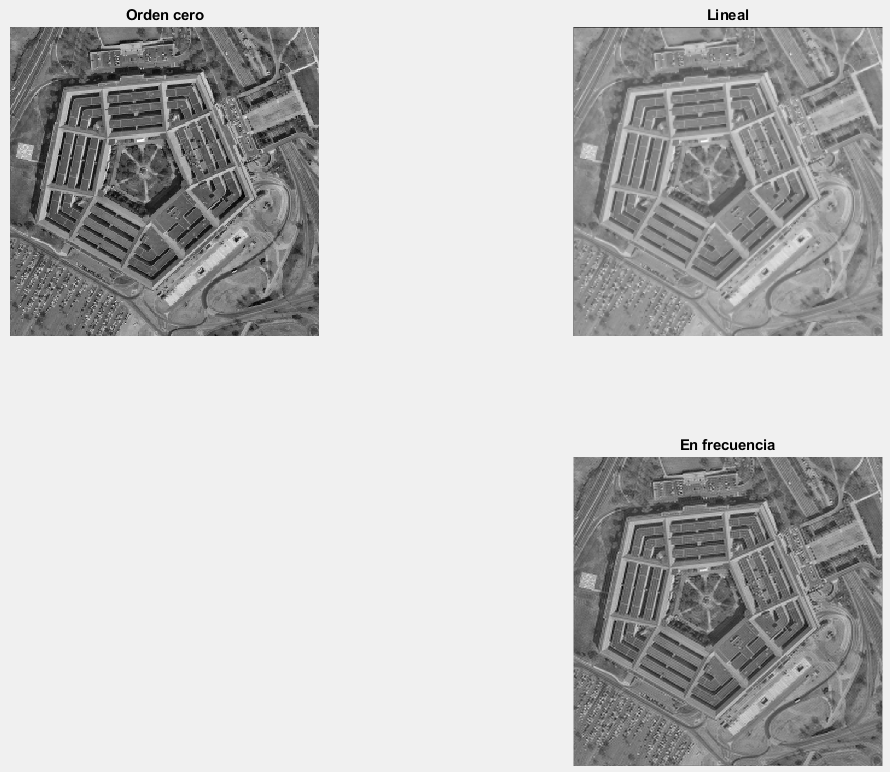
*Figura 14: Comparacion de interpolaciones con T=2*

figure('Name',Comparando interpolación T=4.'),

subplot(2,2,1),imshow(Z\_T4,[]),title('Orden cero')

subplot(2,2,2),imshow(L\_T4,[]),title('Lineal')

subplot(2,2,4),imshow(abs(I\_frec\_t4\_final),[]),title('En frecuencia')



*Figura 15: Comparacion de interpolaciones con T=4*

Interpretación de imagen 14 y 15:

Con los analisis resueltos, observamos mejores calidades de imagenes en el proceso de interpolación cuando el proceso de interpolación es realizado en el espectro de freucia, dejando una imagen mas alzada en tonos y con una disminución de fragmentos permitiendo apreciar mejos la imagen.

# CONCLUSIÓN

# referencias

E.Meijeringa, A Chronology of Interpolation: From ancient Astronomy to Modern Signal and Image Processing, Procedings of the IEEE, Vol 90, No. 3, 2002.

E. Maeland, On the Comparison of Interpolation Methods, IEEE Transactions on Medical Imaging, Vol 1, No 3, 1998.

B. Escalante,Procesamiento Digital Imagenes: Prática 5 “Interpolación”.