UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA UNIVERSITARIA
DE INGENIERIA TECNICA INDUSTRIAL
DE MADRID

DEPARTAMENTO DE ELECTRÓNICA, AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

Prácticas de Procesado de Señales e Imágenes

Práctica 1

Prácticas de procesamiento disperso de las imágenes

1 Técnicas de procesamiento disperso de las imágenes

En esta práctica se tratará de experimentar con las técnicas clásicas del procesado de imágenes empleando *"Image Procesing Toolbox"* de Matlab. Se revisarán las que derivan de procesamiento lineal de señales e imágenes. En primer lugar, se analizarán los conceptos de convolución discreta 2D y de respuesta en frecuencia.

1.1 Repaso a la convolución y a las transformadas de Fourier

Se emplea las aplicaciones demo sobre la convolución continua y la serie de Fourier ubicadas en:

http://www.jhu.edu/~signals/convolve/index.html

www.jhu.edu/~signals/fourier2/index.html

Realice la convolución entre un pulso con una exponencial decreciente y obtenga la descomposición ortogonal de un pulso.

La operación de convolución discreta corresponde con el procesado lineal de las imágenes. Se fundamenta en aplicar una máscara de convolución que corresponda a un tipo de filtro determinado. Los filtros paso bajo suavizarán las imágenes y los filtros paso alto permitirán determinar los bordes.

El filtro binomial es un filtro de fase lineal empleado para el suavizado y eliminación del ruido. En 1D tiene el aspecto de:

```
bino1D = [1 2 1]./4;
[G,W] = freqz (bino1D, 1,64);
figure(1);
plot(W,abs(G));
figure(2);
plot(W,angle(G).*(180/pi));
```

Su extensión a 2D será:

```
bino2D = conv2(bino1D,bino1D');
resp_frBino2D = fft2(bino2D,64,64);
figure(1);
surf(abs(resp_frBino2D));
figure(2);
surf(fftshift(abs(resp_frBino2D)));
```

Interprete la respuesta en frecuencia. ¿Cuál es el efecto de ***fftshift()*** ?. Decida sobre la simetría de la transformada de Fourier y qué zonas corresponde a la baja frecuencia y qué otras a la alta.

Para ver el efecto de la máscara de convolución aplíquese a la imagen del hombre de la cámara:

```
imgEnt = imread('cameraman.tif');
imgSal = imfilter(imgEnt,bino2D);
imshow([imgEnt,imgSal]);
```

Observe el efecto de los bordes. Utilice las condiciones de Neumann (replicar los bordes). Considérese la máscara de Sobel y aplíquese con la anterior imagen:

```
maskSobel = fspecial('sobel')
imgSal = imfilter(imgEnt,maskSobel,'replicate');
imshow([imgEnt,imgSal]);
```

Efectivamente, la máscara es utilizada para implementar la derivada en el eje X. Varíe el código para obtener la derivada en el eje Y. Presente los resultados de manera que quede como en la siguiente figura:

1.1.1 Transformadas de Fourier

Para una mejor comprensión de las transformadas de Fourier, hágase la antitransformada de la segunda componente y de la dieciseisava en la dirección de franjas verticales

```
V=zeros(64,64);V(1,1+2)=2000;V(1,64-2)=2000;
figure(1);
surf(fftshift(V));title('Componente 2 eje de las frec. vertical');
figure(2);imshow(abs(ifft2(V,64,64)));
pause;

figure(1);
V=zeros(64,64);V(1,1+16)=2000;V(1,64-16)=2000;
surf(fftshift(V));title('Componente 16 eje de las frec. vertical');
figure(2);imshow(abs(ifft2(V,64,64)));
```

Modifique la transformada de Fourier para que aparezcan franjas horizontales.

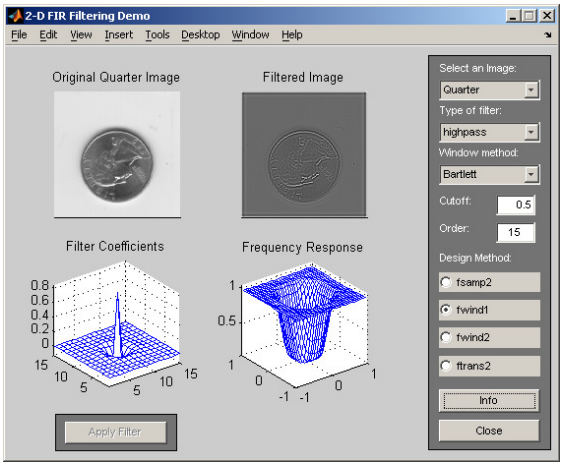
Véase la utilidad de eliminar las componentes de alta frecuencia de una imagen contaminada por el ruido:

```
%% Filtro paso bajo
imagen= imnoise(imread('cameraman.tif'));
n = size(imagen,1);
k = round(.8*n); k = round(k/2)*2;
V=fft2(double(imagen));
V(n/2-k/2+2:n/2+k/2, n/2-k/2+2:n/2+k/2) = 0;
imshow([imagen,uint8(ifft2((V)))])
```

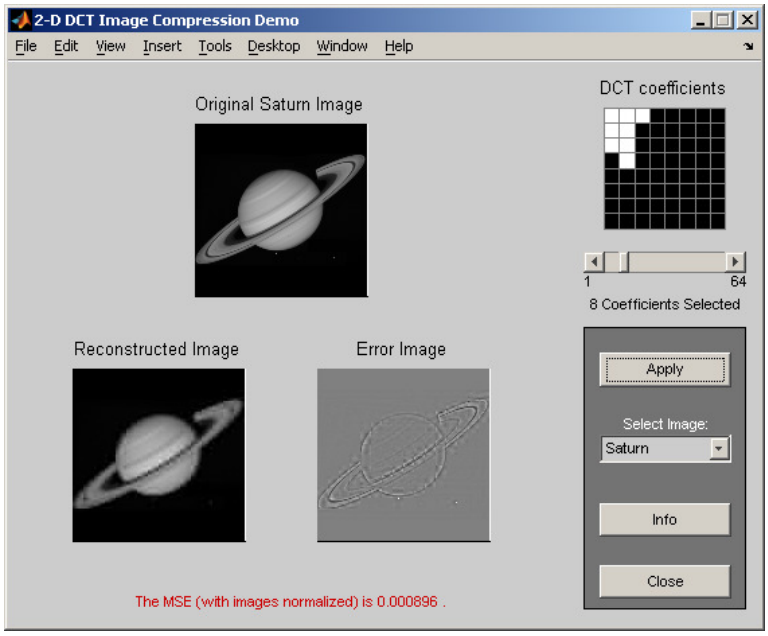
Y de cancelar las componentes de baja frecuencia:

```
%% Filtro paso alto
imagen= imnoise(imread('cameraman.tif'));
V=fft2(double(imagen));
V(1:10,1:10)=0;V(256-9:256,1:10)=0;
V(1:10, 256-9:256)=0;V(256-9:256,256-9:256)=0;
imshow([imagen,uint8(real(ifft2((V))))]);
```

No obstante, este proceder no es adecuado por el alto coste computacional. Es preferible diseñar filtros e implementarlos mediante máscaras de convolución. Emplearemos la demo *firdemo* para ver la potencia del diseño de los filtros, su respuesta en frecuencia y sus resultados al convolucionarlos con las imágenes



Las transformas de Fourier si que son aplicadas para la compresión de las imágenes. Aunque realmente se emplean las transformadas discretas del coseno. Mediante la demo *dctdemo* se puede observar cómo sólo almacenado unas pocas componentes es posible la reconstrucción con muy poca perdida de la información.



1.2 Introducción al procesamiento lineal con Fourier

En este apartado se sigue el guión de Peyré en la referencia:

http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/introduction_3_image/

Se repasará algunas operaciones básicas en las imágenes y se procederá a realizar procesamiento lineal con las transformadas de Fourier.

1.3 Wavelet con el toolbox de Matlab

Con el toolbox de wavelets de Matlab se hacen las primeras operaciones de análisis y síntesis. Dada una señal 1D:

```
close all;
x =0:1/1023:1;
signal = (20*x.^2).*((1-x).^4).*cos(12*pi*x);
% Visualizacion
figure(1);
plot(signal);
axis([0 1024 -.5 .5]);
```

Se procede a obtener el primer nivel de escalamiento (fase de análisis) mediante la función **wavedec( )**. Ésta calcula los productos escalares de la señal con las wavelets. Para extraer los coeficientes de la escala y detalle se emplea **appcoef( )** :

```
scaling_level = 1;
%C vector de escala y detalles
[C,L]=wavedec(signal, scaling_level ,'haar');
cA1 = appcoef(C,L,'haar',1); %coeficiente de aproximacion nivel 1
cD1 = detcoef(C,L,1); % coeficientes de detalle nivel 1
```


```
%Visualizacion
figure(1);
subplot(1,2,1);plot(cA1);axis([1 length(cA1) -1 1]);
subplot(1,2,2);plot(cD1);axis([1 length(cD1) -.05 .05]);
```

Obsérvese que es de la mitad de tamaño y en diferentes magnitudes. Para la reconstrucción se usa la función wavedec(). La señal será la suma de detalles más el de escala:

```
%% Reconstrucción
A1 = wrcoef('a',C,L,'haar',1);
D1 = wrcoef('d',C,L,'haar',1);
F = A1 + D1;
fprintf('dif sum %f\n',sum((F(:)-signal(:)).^2));

%Visualizacion
figure(1);
subplot(1,2,1);plot(x,A1,'r',x,signal,'b');
axis([min(x) max(x) -1 1]);
subplot(1,2,2);plot(D1);axis([1 length(D1) -.05 0.05]);
figure(2);
subplot(1,2,1);plot(x,signal,'b');axis([min(x) max(x) -1 1]);
subplot(1,2,2);plot(x,F,'r');axis([min(x) max(x) -1 1]);
```

Ejercicio: Realizar una descomposición a 3 niveles con wavelets Haar.

Matlab tiene un entorno gráfico para el análisis de señales e imágenes ejecutando:

```
% Entorno gráfico
wavemenu
```

Importando la señal empleada, vea la descomposición al emplear Haar a nivel 3.

1.4 Wavelets Haar con el toolbox de Peyré

En este apartado se sigue el guión de Peyré en la referencia:

http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/wavelet_1_haar1d/

http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/wavelet_2_haar2d/

El procesamiento con las wavelets Haar son aplicadas a señales e imágenes 2D.