

Práctica 6: Operadores globales		Grupo	
		Puesto	
Apellidos, nombre		Fecha	
Apellidos, nombre			

El objetivo de esta práctica es presentar al alumno los fundamentos de las transformadas discretas lineales y parte de sus aplicaciones.

Desarrolle cada ejercicio en un fichero de comandos ‘ejercicio_X.m’ separado. Para conocer el funcionamiento preciso de los comandos que se introducen en este gui3n, utilice la ayuda de MATLAB. Para evitar posibles interferencias con otras variables o ventanas recuerde incluir siempre las instrucciones `clear all` y `close all` al principio de cada fichero de comandos.

Justo antes de finalizar la pr3ctica, comprima el gui3n y los ficheros ‘.m’ generados en un 3nico fichero con el nombre ‘Practica_6_ApellidosNombre.zip’, con3ctese al sistema de entrega de pr3cticas de la Intranet y entr3guelo en el grupo de pr3cticas (grupo 3nico).

1 Transformadas discretas lineales bidimensionales

Este apartado gu3a el modo de realizar una transformada y muestra su interpretaci3n como cambio de base.

1.1 Ejercicio 1: c3lculo de una transformada directa y de su inversa

El objetivo de este ejercicio es realizar una transformada discreta lineal y su inversa. Se presenta 3nicamente el caso de las transformadas separables y unitarias, que son las de mayor aplicaci3n en el 3mbito del tratamiento de imagen.

Seg3n las explicaciones de la parte te3rica de la asignatura sabemos que la relaci3n entre una imagen o se3al bidimensional ψ y su imagen transformada Ψ viene dada en forma matricial por las expresiones:

$$\begin{aligned}\Psi &= \mathbf{W}^{*T} \cdot \psi \cdot \mathbf{W}^* \\ \psi &= \mathbf{W} \cdot \Psi \cdot \mathbf{W}^T\end{aligned}$$

, donde la matriz \mathbf{W} es el n3cleo de la transformaci3n.

A lo largo de esta pr3ctica, por simplicidad, asumiremos que las im3genes sobre las que se efect3an transformadas son siempre cuadradas, es decir, de N filas por N columnas. En esta situaci3n el n3cleo de la transformaci3n ser3 tambi3n una matriz de $N \times N$ valores.

Se propone, a modo de ejemplo, obtener la Transformada Discreta del Coseno (DCT) de una imagen. El n3cleo de la DCT viene dado, para cada fila n y columna u , por la expresi3n:

$$\mathbf{W}(n,u) = \begin{cases} \sqrt{\frac{1}{N}} & , u = 0, 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2n+1)u \cdot \pi}{2N}\right) & , 1 \leq u \leq N-1, 0 \leq n \leq N-1 \end{cases} \quad (6.1)$$

Para definirlo en MatLab puede utilizar los siguientes comandos:

```
>> N=image_width; [u,n]=meshgrid([0:N-1],[0:N-1]);  
>> W=(sqrt(2/N))*cos((2*n+1)*pi.*u/(2*N)); W(:,1)=(1/sqrt(N));
```

Utilice este núcleo y las expresiones generales anteriores para calcular la DCT de la imagen `edificio.bmp`¹. Tenga en cuenta que antes de calcular la transformada deberá convertir la imagen a tipo `double` (no utilice `im2double` sino sencillamente `double`). Observe adicionalmente que en el caso de la DCT el núcleo presenta valores reales, por lo que no es necesario conjugarlo para calcular la transformada directa.

Una vez obtenida la transformada aplique sobre ella la operación inversa para obtener de nuevo la imagen original. Para comprobar que el resultado es correcto, calcule la energía de la diferencia entre la imagen inicial (convertida a `double`) y la imagen obtenida tras aplicar la transformada directa e inversa. Observe que aunque la energía obtenida es despreciable frente al rango de variación de las imágenes (0-255), no es completamente nula. Ello se debe a la precisión finita en el cálculo de funciones trigonométricas derivada del uso de magnitudes irracionales (en este caso, del número π).

Finalmente, modifique su código para repetir el ejercicio calculando esta vez la Transformada Discreta del Seno (DST) teniendo en cuenta que su núcleo viene dado por la expresión:

$$W(n,u) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{(n+1)(u+1) \cdot \pi}{N+1}\right), \quad 0 \leq n, u \leq N-1 \quad (6.2)$$

1.2 Ejercicio 2: representación de vectores base e imágenes base

El objetivo de este ejercicio es representar las imágenes de la nueva base que caracteriza cada transformada, para poder obtener una visión intuitiva del efecto que la transformada persigue y una interpretación adecuada del significado de cada coeficiente (recuerde que cada coeficiente o valor de la imagen transformada indica la ponderación de la correspondiente imagen base a la hora de recomponer la imagen original).

Al efectuar una transformada de $N \times N$ valores se está efectuando un cambio de base de la imagen original de $N \times N$ valores en una *imagen* transformada de $N \times N$ coeficientes. Las imágenes base que gobiernan el cambio son por lo tanto $N \times N$ imágenes de tamaño $N \times N$ cada una. Siendo así, para poder representar todas ellas se propone realizar el ejercicio con un núcleo de tamaño reducido, por ejemplo de tamaño $N = 8$, en vez de con el núcleo correspondiente a una imagen como la del ejercicio anterior ($N = 512$) que daría lugar a un número irrepresentable de imágenes base.

Según lo visto en las explicaciones teóricas, en el caso de las transformadas separables el núcleo que la caracteriza al realizar una transformada 2D es el mismo que caracteriza la transformada 1D, sólo que en el caso de la bidimensional se aplica dos veces (sobre ambas dimensiones de la señal a transformar). En el caso de la transformada 1D, sabemos que los vectores columna del núcleo que la define son precisamente los vectores de la nueva base que representa la transformada. En el caso de la transformada 2D, sabemos que el producto tensorial de todas las posibles parejas de vectores base da lugar a las imágenes base de la transformada 2D.

Teniendo en cuenta lo anterior, defina el núcleo de una DCT de tamaño 8×8 y represente en una misma ventana (comando `subplot`) los ocho vectores que definen la nueva base, ordenados desde la primera hasta la última columna. Al representar cada vector superponga sus ocho valores

¹ Descárguese esta imagen de la página *web* de la asignatura.

(comando `stem`) con la curva que los interpola linealmente (comando `plot`). Indique el número de cruces por cero (o *wave number*) de cada uno de los ocho vectores:

Vector columna:	1	2	3	4	5	6	7	8
Cruces por cero:								

A continuación obtenga, dos a dos, los 8×8 productos tensoriales (multiplique cada vector columna por el correspondiente vector columna transpuesto) de los vectores anteriores, lo que resultará en 8×8 imágenes. Represente estas imágenes en una misma ventana (comando `subplot`), formando una matriz de 8×8 imágenes de tamaño 8×8 cada una.

Para que las imágenes base sean fáciles de comparar e interpretar escale cada una de modo que su rango se sitúe entre -1 y $+1$. Para lograrlo, observe en la expresión del núcleo de la DCT (Eq. 6.1) que cada vector columna sigue una función sinusoidal (el primero constante y de valor $\sqrt{1/N}$ y el resto ponderadas por $\sqrt{2/N}$). Por lo tanto, para que los productos tensoriales varíen en $(-1,1)$ deberá multiplicar el resultado por $N/\sqrt{2}$, si el producto involucra el primer vector columna, y por $N/2$ si no lo involucra.

Modifique el código para repetir el mismo ejercicio con la DST. Compare las imágenes base de mínima frecuencia de ambas transformadas e indique si alguna de las dos representa la componente continua o valor medio de la imagen original (es decir, si alguna de las dos es una imagen base constante):

2 Transformadas y compactación de energía

Este apartado presenta la característica de la DCT y de la DST como transformadas para compactar la energía de la señal y el posible uso de esta propiedad en aplicaciones de codificación.

2.1 Ejercicio 3: observación de la distribución de energías de la imagen y su transformada.

El objetivo de este ejercicio es observar la propiedad de la DCT y de la DST de compactar la energía de las señales sobre las que operan.

Partiendo del código del Ejercicio 1, obtenga la DCT de la imagen `edificio.bmp`. Calcule la energía de la imagen y la energía de su DCT y compruebe, según se ha visto en las explicaciones teóricas, que son idénticas. Haga la misma comprobación con la DST.

Para observar la distribución de energía en cada zona de la imagen, divida la imagen original en $K \times K$ subimágenes y obtenga en una matriz de $K \times K$ valores el porcentaje de la energía de la imagen original contenido en cada subimagen. Efectúe este proceso para $K = 4$ (compruebe que el resultado es correcto verificando que la suma de porcentajes es 100). A continuación repita el proceso sobre la imagen DCT y sobre la imagen DST. Transcriba aproximadamente (basta con un dígito significativo) los porcentajes obtenidos en cada caso:

% de energía regiones de la imagen original

% de energía regiones de la imagen DCT

% de energía regiones de la imagen DST

Comente los resultados obtenidos indicando si las transformadas compactan o no y cuánto la energía de la señal, indicando en qué regiones de las transformadas se concentra la mayor parte de la energía, y concluyendo si DCT y DST compactan de manera similar:

2.2 Ejercicio 4: efecto de la eliminación de las componentes de menor energía.

El objetivo de este ejercicio es apreciar la pérdida de calidad que se produce sobre una imagen al ir eliminando (haciendo nulas) componentes de su DCT.

En primer lugar obtenga la DCT de la imagen `edificio.bmp` y represéntela. Como el rango de la DCT es mucho mayor que el de la imagen, en vez de sus valores represente el logaritmo de su valor absoluto.

Tomando como punto de partida el ejercicio anterior, divida la DCT de la imagen en $K \times K$ subimágenes, pero ahora con $K = 16$ (es decir en 256 subimágenes). Calcule el % de energía de la subimagen que menos energía contiene. Sitúe un umbral T justo en el doble de este valor y anule las subimágenes de la DCT cuyo valor sea inferior a este umbral. Represente la DCT así modificada e indique cuántas subimágenes han sido anuladas:

Obtenga la transformada inversa de la DCT modificada y observe a tamaño real la imagen que obtiene. Si no aprecia diferencia con respecto a la original repita el proceso anterior aumentando el valor del umbral T hasta que aprecie diferencia con respecto a la imagen original. Indique cuál es el valor del umbral $T = T_{apreciable}$ en esta situación:

$T_{apreciable}$	
------------------	--

A continuación siga aumentando el valor del umbral hasta que juzgue que la pérdida de calidad de la imagen obtenida es claramente visible en condiciones normales (por ejemplo observando ambas imágenes a tamaño real a unos 60cm del monitor). Indique cuál es el valor de este nuevo umbral $T = T_{aceptable}$:

$T_{aceptable}$	
-----------------	--

Obtenga el % de píxeles de la DCT que ha anulado al llegar a $T_{apreciable}$ y el que ha anulado hasta llegar a $T_{aceptable}$. Comente las posibles aplicaciones de los resultados obtenidos:

3 Transformada de Haar

3.1 Ejercicio 5: Cálculo de la transformada de Haar

El objetivo de este ejercicio es implementar la Transformada Discreta de Haar (DHT) de una imagen. La DHT esta basada en las funciones de Haar, familias de N funciones (orden N) continuas distintas:

$$h_k(x), k \in \{0, \dots, N-1\}, N = 2^n \quad (6.3)$$

La construcción de la función k -ésima se basa en la definición intermedia de dos valores, p y q (asociados a cada k), que toman N valores. A partir de estos índices, las k funciones se definen según la expresión:

$$h_k(x) = h_{pq}(x) = \begin{cases} \frac{2^{p/2}}{\sqrt{N}}, & \frac{q-1}{2^p} \leq x < \frac{q-0.5}{2^p} \\ \frac{-2^{p/2}}{\sqrt{N}} & \frac{q-0.5}{2^p} \leq x < \frac{q}{2^p} \\ 0 & \text{resto} \end{cases}, x \in [0,1] \quad (6.4)$$

Para realizar este ejercicio desarrolle una función MatLab (denomínela `matriz_haar`) que tome como parámetro de entrada el valor n mínimo tal que $N = 2^n$ donde N se corresponde con las dimensiones de la imagen $N \times N$. Esta función devolverá el núcleo de la transformación W . La función deberá tener en cuenta los siguientes aspectos:

- Primeramente debe obtener los valores p y q que permiten calcular las funciones de Haar. Utilice el siguiente código para obtener dichos valores:

```
>> N=2^n;  
>> p=zeros(1,2);  
>> q=[0,1];  
>> for(i=1:n-1)  
>>     p=[ p i*ones(1,2^i)];  
>>     q=[ q 1:2^i];  
>> end  
>> k=2.^p+q-1; % k=0:(N-1)
```

- Las funciones h_k se calculan mediante la expresión 6.4. Utilice el siguiente código:

```
>> z=[0:1/N:1-1/N];  
>> hk=zeros(N,N);  
>> hk(k(1)+1,:)=1;  
>> for(i=2:N)  
>>     z1=(q(i)-1)/(2^p(i));  
>>     z2=(q(i)-0.5)/(2^p(i));  
>>     z3=q(i)/(2^p(i));  
>>     for(j=1:N)  
>>         if(z(j)>=z1 & z(j)<z2) hk(k(i)+1,j)=2^(p(i)/2);  
>>         elseif(z(j)>=z2 & z(j)<z3) hk(k(i)+1,j)=-2^(p(i)/2);  
>>         else hk(k(i)+1,j)=0;  
>>         end  
>>     end  
>> end  
>> hk=(1/sqrt(N))*hk;
```

- Finalmente calcule la transpuesta de la matriz resultante h_k para obtener la matriz de transformación \mathbf{W} que ha de devolver la función `matriz_haar`.

Utilice el núcleo calculado con la función `matriz_haar` y las expresiones generales anteriormente descritas para calcular la DHT de la imagen `edificio2.bmp`². Recuerde que antes de calcular la transformada deberá convertir la imagen a tipo `double` (no utilice `im2double` sino sencillamente `double`).

Una vez obtenida la transformada aplique sobre ella la operación inversa para obtener de nuevo la imagen original. Finalmente, visualice en distintas figuras la imagen original, la imagen reconstruida y la Transformada Haar resultante de aplicar la matriz \mathbf{W} . Debido al rango de valores que presenta la Transformada Haar de una imagen, en vez de representar el módulo normalizado resulta más conveniente representar su logaritmo³:

```
>> ima_transf=log(abs(ima_transf)+0.1);  
>> ima_max=max(max(ima_transf)); ima_min=min(min(ima_transf));  
>> imshow(ima_transf,[ima_min ima_max/256],'InitialMagnification',100);
```

Para comprobar que el resultado es correcto, calcule la energía de la diferencia entre la imagen inicial (convertida a `double`) y la imagen obtenida tras aplicar la transformada directa e inversa. Observe que la energía obtenida es despreciable.

² Descárguese esta imagen de la página *web* de la asignatura.

³ Observe que en vez de calcular directamente el logaritmo del módulo de la Transformada, se le suma una unidad para evitar que el resultado tome valores negativos.

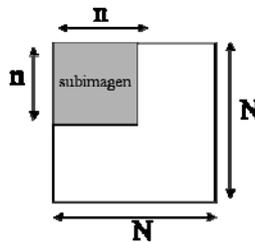
3.2 Ejercicio 6: efecto del muestreo de la transformada Haar.

El objetivo de este ejercicio es apreciar la pérdida de calidad que se produce sobre una imagen al ir eliminando (haciendo nulas) componentes de su transformada DHT.

En primer lugar obtenga la DHT de la imagen `edificio2.bmp` y representela. Para ello utilice el código MatLab descrito en el apartado anterior.

Posteriormente, modifique la DHT calculada seleccionando una submatriz (o subimagen) de ella (anulando el resto de los coeficientes de la DHT). Después obtenga la transformada inversa de la DHT modificada y observe a tamaño real la imagen que obtiene. Para seleccionar una submatriz (o subimagen) de la DHT, deberá tener en cuenta

- Las subimágenes de la DHT se seleccionaran a partir de la esquina superior izquierda (índice (1,1)) de la imagen para garantizar una reconstrucción adecuada.



- El tamaño de la imagen seleccionada ha de ser potencia de dos para que la subimagen seleccionada contenga secciones completas de la DHT.

Obtenga la transformada inversa de la DHT modificada para los valores de n iguales a $N/8, N/4, N/2$ y N y observe a tamaño real la imagen reconstruida que obtiene. Comente las diferencias observadas entre las distintas imágenes, y las posibles aplicaciones de los resultados obtenidos: