Tema 12. Compresión y calidad de imagen y vídeo

Procesamiento de Señales, Sonido e Imágenes Digitales

Carlos Quemada Mayoral

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

La Universidad en Internet

04/03/2021

Índice

12.1. Redundancia en la imagen

- Los **métodos de compresión** de imagen se dividen en dos grupos:

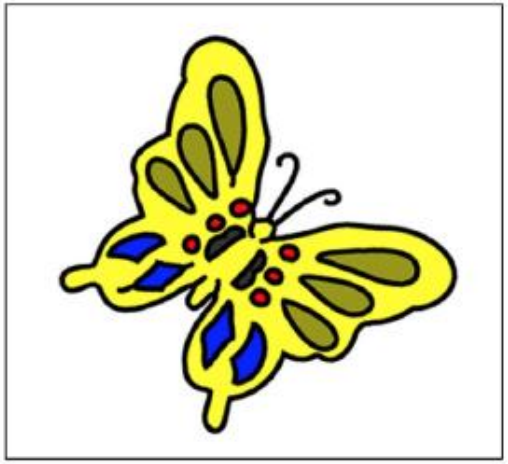
    - **Sin perdidas** (**lossless**): Cuando la descompresión de imagen recupera la imagen original sin cambios.
    - **Compresión con pérdidas** (**lossy**). Cuando se tolera un pequeño grado de deterioro visual en la imagen descomprimida.

- La **compresión sin pérdidas** suele utilizarse en **imágenes sintéticas** donde es fácil percibir el deterioro de la imagen.

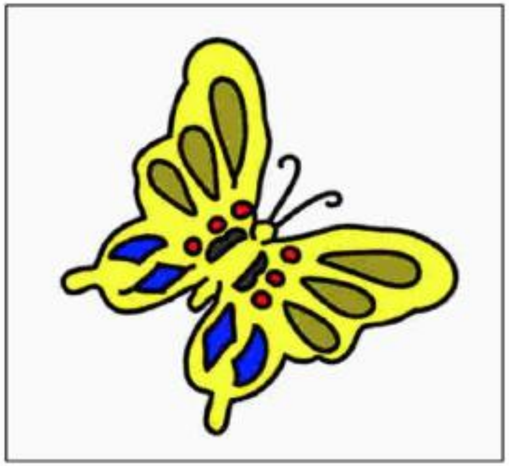
- La **compresión con pérdidas** se utiliza para **imágenes fotográficas** donde el ojo humano no es capaz de percibir el deterioro de la imagen, a excepción de las fotografías médicas en las que no se tolera pérdida de información.

- A continuación se presentan dos imágenes comprimidas con pérdidas, una sintética y otra fotográfica. En la imagen sintética sí se percibe un deterioro de imagen, mientras que en la imagen fotográfica el deterioro no es perceptible.

12.1. Redundancia en la imagen



(a) Imagen sintética original



(b) Imagen sintética comprimida

(c) Imagen fotográfica original

(d) Imagen fotográfica comprimida

12.1. Redundancia en la imagen

- **Medidas de compresión**.

- **Ratio de compresión CR (Compression Ratio)**:

$$CR = \frac{s}{s_c}$$

s ≡ tamaño en bytes de la imagen original

s_c ≡ tamaño en bytes de la imagen comprimida

- En general, la información se puede comprimir si existe redundancia, la cual se define como:

$$R = 1 - \frac{1}{CR}$$

- Si no hay compresión (CR=1) $\Rightarrow$ R=0.

- Si hay alta tasa de compresión ($CR \to \infty$) $\Rightarrow$ hay mucha redundancia ($R \to 1$)

- Esto quiere decir que $CR \in [1,\infty)$ y $R \in [0,1)$

12.1. Redundancia en la imagen

- **Compresión de imágenes con Octave**.

- El comando **imwrite** permite comprimir una imagen usando el parámetro ‘quality’ que establece la calidad de compresión JPEG de una imagen. Este parámetro debe darse en el rango 0-100. Ejemplo:
  - I = imread('logo_unir.png'); Mete la imagen en I
  - imwrite(I, 'logo_unir_50.jpg', 'quality', 50); Convierte a JPEG y comprime con quality=50
  - imwrite(I, 'logo_unir_10.jpg', 'quality', 10); Convierte a JPEG y comprime con quality=10
  - imwrite(I, 'logo_unir_0.jpg', 'quality', 0); Convierte a JPEG y comprime con quality=0

Nivel calidad JPEG	Tamaño fichero	Ratio de compresión (CR)
100	34733	34733/34733 = 1
50	12447	34733/12447 = 2.79
25	6834	34733/6834 = 5.08
0	4759	34733/4759 = 7.3

12.1. Redundancia en la imagen

- **Compresión de imágenes con Octave**.

Mínima compresión

(a) Imagen original

(b) Compresión a 50% de calidad

(c) Compresión a 25% de calidad

(d) Compresión a 0% de calidad

Máxima compresión

12.2. Tipos de redundancias en la imagen

- Las técnicas de compresión aprovechan tres tipos de redundancias:
- Redundancia en la **codificación**.
- Redundancia entre **píxeles**.
- Redundancia **psicovisual**.

12.2. Tipos de redundancias en la imagen

- **Redundancia en la codificación**. Pueden usarse dos tipos de técnicas de compresión, **con longitud de código fija y variable**:

- Por ejemplo, en una imagen en escala de grises si tuviéramos solo dos niveles, blanco y negro, bastaría con usar un código de un bit por píxel para representar la imagen (en vez de 8 bits). 1 para blanco y 0 para negro.

- A veces se usan algoritmos de compresión donde la **longitud** de la palabra del código varía **en función de la probabilidad de aparición** de cada color o cada nivel de escala de grises. El algoritmo más conocido de este tipo es el de Huffman, el cual asigna el mayor número de bits a los niveles menos probables de la imagen y viceversa. De esta forma, el tamaño de la imagen en bits se reduce.

12.2. Tipos de redundancias en la imagen

- **Redundancia entre píxeles**. Se produce porque píxeles cercanos tienden a contener valores o niveles cercanos.

- La redundancia entre píxeles se puede explotar de varias formas:

  - **Diferenciación**. En vez de almacenar el valor de cada píxel, almacenamos la diferencia entre un píxel y el anterior. El valor de las diferencias suele ser más pequeño que el valor de cada píxel individual pudiendo reducir el número de bits de codificación para representar la imagen. Además, guardando el valor del primer píxel es posible recuperar el valor exacto de cada uno sumando las diferencias almacenadas. Mirar ejemplo en apuntes.

  - **Predicción**. Podemos predecir el valor de un píxel en función de los valores adyacentes anteriores y después almacenar la diferencia entre el valor predicho y el muestreado.

12.2. Tipos de redundancias en la imagen

- **Redundancia psicovisual**.

- Conocer cuándo el ojo humano no tiene suficiente agudeza para percibir diferencias en una imagen digital permite comprimir una imagen con la misma calidad.

- Conocer la información más difícil de percibir por el ojo humano permite crear sistemas que comprimen todavía más la imagen, a cambio de pequeñas perdidas de calidad.

- Los codificadores que aprovechan la redundancia **psicovisual** son por naturaleza lossy, es decir, el algoritmo de descompresión no recupera exactamente la misma información.

12.3. La DCT (*Discrete Cosine Transform*)

- Permite comprimir una señal de tiempo discreto en un conjunto reducido de coeficientes.

- Herramienta fundamental del estándar de compresión de vídeo **MPEG** y del estándar de compresión de imágenes **JPEG**.

- Además **MP3** (MPEG-1 Layer 3) también utiliza una versión modificada de la DCT para comprimir la señal de audio.

12.3. La DCT (*Discrete Cosine Transform*)

- En la **DFT** los **coeficientes** X[m] son en general **complejos**, incluso si la secuencia x[n] es real.

- La **DCT** es una transformada **real** pura. Tanto la señal como sus coeficientes se representan **solo** con **números reales**.

- Existen varios tipos de DCT (DCT-1, DCT-2, ..., DCT-8). Aquí vamos a estudiar solo la DCT-2 por la propiedad de compresión que presenta.

ecuación de análisis $\Rightarrow X[m] = DCT\{x[n]\} = \beta[m] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi}{2N} m\right) \quad \text{con} \quad m = 0, 1, .., N-1$

ecuación de síntesis $\Rightarrow x[n] = IDCT\{X[m]\} = \sum_{m=0}^{N-1} \beta[m] X[m] \cos\left(\frac{(2n+1)\pi}{2N} m\right) \quad \text{con} \quad n = 0, 1, .., N-1$

Donde β[m] $\Rightarrow \beta[m] = \begin{cases} \sqrt{\frac{1}{N}}, & m = 0 \\ \sqrt{\frac{2}{N}}, & m \neq 0 \end{cases}$

Los primeros coeficientes de X[m] constituyen las bajas frecuencias (m<<) y los últimos, las altas (m>>).

12.3. La DCT (*Discrete Cosine Transform*)

- **Ejemplo**. Calcular los coeficientes de la DCT de: $x[n] = \{1, -2, 1, 3\}$

$$X[m] = \beta[m] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi}{2N} m\right)$$

$$= \beta[m]\left[\cos\left(\frac{\pi m}{8}\right) - 2\cos\left(\frac{3\pi m}{8}\right) + \cos\left(\frac{5\pi m}{8}\right) + 3\cos\left(\frac{7\pi m}{8}\right)\right]$$

- Evaluando en m = 0, ..., m = N-1 tenemos:

$$X[m] = \begin{cases} 3/2, & m = 0 \\ -3/\sqrt{2}, & m = 1 \\ 5/2, & m = 2 \\ \sqrt{2}, & m = 3 \end{cases}$$

- Veremos si se cumple la relación de Parseval: $\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{m=0}^{N-1} |X[m]|^2$
- Energía en el dominio del tiempo: $E_x = \sum_{n=0}^{N-1} |x[n]|^2 = 1 + 4 + 1 + 9 = 15$
- En el dominio de la frecuencia: $E_X = \sum_{m=0}^{N-1} |X[m]|^2 = \frac{1}{4}\left[\left(\frac{3}{2}\right)^2 + \left(\frac{-3}{\sqrt{2}}\right)^2 + \left(\frac{5}{2}\right)^2 + \left(\sqrt{2}\right)^2\right] = 15$

12.3. La DCT (*Discrete Cosine Transform*)

- **Cálculo con Octave**.

- Podemos usar el comando **dct(x)** para calcular la DCT de una secuencia y el comando **idct(X)** para obtener la DCT inversa.

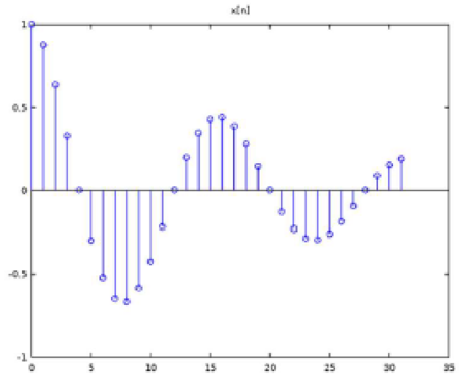
- x = [1 -2 1 3];
- X = dct(x)
- X = 1.5000 -2.1184 2.5000 1.4186

12.3. La DCT (*Discrete Cosine Transform*)

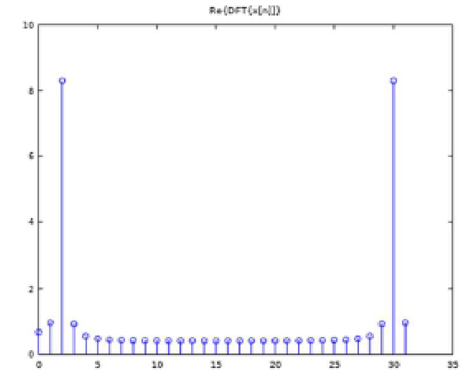
- **Compactación de coeficientes**. La DCT se usa en aplicaciones de compresión por su propiedad de concentrar la energía en los primeros coeficientes. Ejemplo. Sea la señal $\Rightarrow$ $x[n] = 0.95^n \cos\left(\frac{\pi}{8}n\right) \quad \text{con} \quad N = 32$

- **La mayoría de la energía se concentra en los primeros 7 coeficientes.**

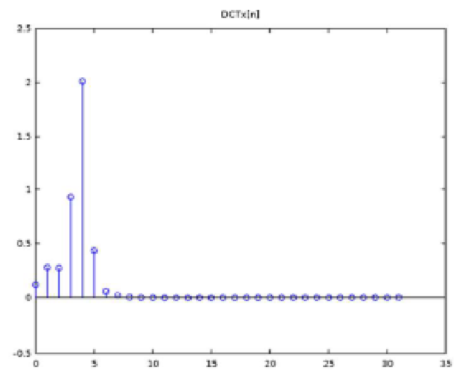
- **Con la DFT es necesario almacenar 32 coeficientes (16 reales y 16 complejos) ya que la señal es real y los coeficiente de m=16-31 son simétricos conjugados.**



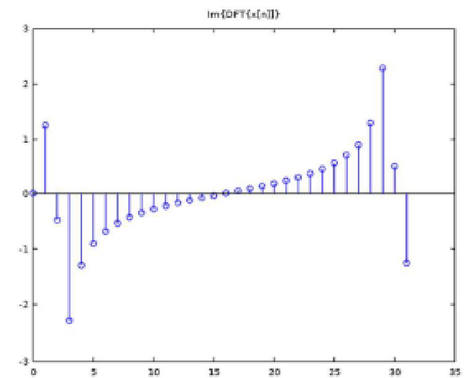
(a) Señal $x[n]$



(c) Coeficientes $\mathrm{Re}\{\mathrm{DFT}\{x[n]\}\}$



(b) Coeficientes $\mathrm{DCT}\{x[n]\}$



(d) Coeficientes $\mathrm{Im}\{\mathrm{DFT}\{x[n]\}\}$

12.3. La DCT (*Discrete Cosine Transform*)

- **Mecanismo de compresión**. El algoritmo de compresión suele implicar:

- **Normalización**. Habitualmente la señal x[n] de audio o imagen corresponde a valores enteros (típicamente de -128-127), mientras que los coeficientes son valores reales con cualquier rango. Para normalizar debemos encontrar el valor máximo y mínimo de los coeficientes y escalarlos al rango -128-127.

- **Redondeo**. Los coeficientes normalizados son valores reales. Los primeros coeficientes se redondean al entero más cercano y los siguientes coeficientes se redondean a cero.

- **Partición**. Los datos se suelen particionar en secuencias de N elementos y la DCT se aplica a cada secuencia de forma individual.

12.3. La DCT (*Discrete Cosine Transform*)

- **Ejemplo** ⇒ x[n] = [12,10,15,10,15,13];

- x = [12,10,15,10,15,13];
- X = dct(x)
- X = 30.61862 -1.85177 0.00000 -0.40825 0.00000 4.68020

- No es necesario normalizar ya que todos los valores se encuentran en el rango -128-127.

- Xr = int8(X) // Redondeamos a entero de 8 bits (-128-127)
- Xr = 31 -2 0 0 0 5
- x2 = idct(double(Xr)) // Calculamos la DCT inversa
- x2 = 12.2875 9.7980 15.1452 10.1662 15.5134 13.0239
- int8(x2) Redondeamos a entero de 8 bits (-128-127)
- ans = 12 10 15 10 16 13

- Vemos que solo ha cambiado el penúltimo coeficiente en una unidad.
- Si hubiéramos almacenado solo los dos primeros coeficientes de Xr (31 y -2) el resultado hubiera cambiado ligeramente ⇒ ans = 12 12 12 13 13 14

12.4. Compresión de imágenes con la DCT

- La DCT es especialmente útil para compresión *lossy* cuando existen correlaciones (redundancia) entre valores contiguos.

- Hemos visto que las imágenes presentan alto nivel de correlación espacial entre píxeles cercanos en 2D. Por esto, es necesario extender la DCT a 2D:

- Dada una matriz de píxeles con M filas y N columnas x[m,n], obtenemos una matriz de coeficientes X[p,q] con M filas y N columnas aplicando la siguiente ecuación de análisis de la DCT en 2D:

$$X[p,q] = DCT2\{x[m,n]\}$$

$$= \beta[p]\beta[q] \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] \cos\left(\frac{(2m+1)\pi}{2M} p\right) \cos\left(\frac{(2n+1)\pi}{2N} q\right)$$

$$p = 0, 1, \dots, M-1 \qquad \beta[p] = \begin{cases} \sqrt{\frac{1}{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & p \neq 0 \end{cases} \qquad \beta[q] = \begin{cases} \sqrt{\frac{1}{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & q \neq 0 \end{cases}$$

$$q = 0, 1, \dots, N-1$$

12.4. Compresión de imágenes con la DCT

- **Ecuación de síntesis**.

$$x[m,n] = IDCT2\{X[p,q]\}$$

$$= \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \beta[p]\beta[q] \cos\left(\frac{(2m+1)\pi}{2M}p\right) \cos\left(\frac{(2n+1)\pi}{2N}q\right)$$

Con:

$$m = 0, 1, \ldots, M-1$$

$$n = 0, 1, \ldots, N-1$$

12.4. Compresión de imágenes con la DCT

- **Algoritmo de la DCT 2D**.

- Habitualmente la imagen se divide en matrices de 8x8 píxeles (bloques). Si la última fila o columna no son divisibles entre 8 se duplican estas filas o columnas el número de veces necesarias para hacerlas divisibles.

- Cada bloque de píxeles se pasa por la ecuación de análisis para obtener un bloque de coeficientes.

- Dado que el bloque de píxeles contiene números reales se redondean al entero más cercano.

- Para deshacer esta operación se aplica la operación de **síntesis** de la **DCT** en 2D a cada bloque de coeficientes redondeados.

12.4. Compresión de imágenes con la DCT

- **Cálculo con Octave**.
- X = dct2(x); // DCT en 2D aplicada a una matriz x.
- x = idct2(X); // Ecuación de síntesis aplicada a la matriz de coeficientes X.

12.4. Compresión de imágenes con la DCT

- **Cálculo con Octave**. **Ejemplo**: Dada la matriz de 8x8 píxeles, obtener sus coeficientes de la DCT 2D.

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
159	155	169	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Valores de escala de grises en el rango 0-255

-128

11	16	21	25	27	27	27	27
16	23	25	28	31	28	28	28
31	27	41	35	30	28	28	28
31	33	34	32	32	31	31	31
31	32	33	34	34	27	27	27
33	33	33	33	32	29	29	29
34	34	33	35	34	29	29	29
34	34	33	33	35	30	30	30

Restamos a cada valor 128 para dejarlos en -128-127

X = dct2(x)

237.87	1.41	-11.22	-5.44	2.13	-0.48	-0.63	2.96
-20.82	-15.56	-5.56	-3.34	-2.86	0.87	2.07	0.10
-12.17	-10.59	-2.04	1.66	0.20	-1.59	-1.69	-0.95
-10.20	-5.30	-0.97	1.78	0.90	-1.74	-2.93	-1.93
-2.87	-3.28	0.61	1.79	-0.13	-1.86	-1.47	-0.36
2.37	0.47	1.86	-0.41	-0.78	1.81	1.61	-0.54
1.66	2.83	0.81	-1.77	-0.49	3.30	3.79	1.37
0.05	4.43	-2.75	-2.12	1.87	2.62	1.88	1.47

Cuantización

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Matriz de cuantización más usada con compresión DCT

237.87/16 = 14.86 ⇒ 15
-20.82/12 = -1.73 ⇒ -2
…
…

Con el objetivo de hacer el mayor número de ceros en la matriz de píxeles y por ende codificar con menor número de bits, la matriz se divide por la matriz de cuantización y después cada término se redondea al entero más cercano

12.4. Compresión de imágenes con la DCT

- **Descompresión.**

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

15x16 = 240
-2x12 = -24
…
…

Se multiplica por la matriz cuantización

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
-14	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Matriz de cuantización más usada con compresión DCT

IDCT

14,03	15,88	18,87	21,97	24,29	25,47	25,76	25,69
20,71	22,22	24,61	26,91	28,37	28,76	28,44	28,04
28,97	29,99	31,46	32,57	32,73	31,93	30,69	29,80
33,57	34,17	34,85	34,94	34,00	32,17	30,15	28,83
33,62	34,05	34,41	34,07	32,67	30,42	28,08	26,59
32,10	32,62	33,17	33,08	31,94	29,94	27,78	26,39
31,69	32,46	33,47	33,97	33,49	32,08	30,38	29,24
32,24	33,21	34,58	35,55	35,58	34,64	33,31	32,36

Redondeo

14	16	19	22	24	25	26	26
21	22	25	27	28	29	28	28
29	30	31	33	33	32	31	30
34	34	35	35	34	32	30	29
34	34	34	34	33	30	28	27
32	33	33	33	32	30	28	26
32	32	33	34	33	32	30	29
32	33	35	36	36	35	33	32

+128

142	144	147	150	152	153	154	154
149	150	153	155	156	157	156	156
157	158	159	161	161	160	159	158
162	162	163	163	162	160	158	157
162	162	162	162	161	158	156	155
160	161	161	161	160	158	156	154
160	160	161	162	161	160	158	157
160	161	163	164	164	163	161	160

12.5. Compresión de la redundancia temporal

- La redundancia espacial se produce porque píxeles cercanos en la imagen tienden a tener valores similares.

- En el caso del vídeo, además de redundancia espacial dentro del *frame*, se produce una redundancia temporal debida a que *frames* contiguos en el tiempo tienden a ser similares.

- La redundancia temporal puede ser eliminada mediante una codificación diferencial, almacenando la diferencia entre *frames* consecutivos similares.

(a) Frame n (b) Frame n+1 (c) Frame diferencia

- Suponiendo que en escala de grises tenemos 256 niveles (0 blanco y 255 negro), el *frame* diferencia presenta valores más bajos que los de n o n+1 por lo que el número de bits para codificarlos puede reducirse.

Ejercicio 1

- Calcular la DCT de la secuencia x = [10, 8, 10, 12]

- Aplicamos la ecuación de análisis

$$X[m] = \beta[m] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi}{2N} m\right) \qquad \beta[m] = \begin{cases} \sqrt{\frac{1}{N}} = \frac{1}{2}, & m = 0 \\ \sqrt{\frac{2}{N}} = \frac{1}{\sqrt{2}}, & m \neq 0 \end{cases}$$

$$X[m] = \beta[m]\left[10\cos\left(\frac{\pi m}{8}\right) + 8\cos\left(\frac{3\pi m}{8}\right) + 10\cos\left(\frac{5\pi m}{8}\right) + 12\cos\left(\frac{7\pi m}{8}\right)\right]$$

$$X[0] = \frac{1}{2}\left[10\cos\left(\frac{\pi 0}{8}\right) + 8\cos\left(\frac{3\pi 0}{8}\right) + 10\cos\left(\frac{5\pi 0}{8}\right) + 12\cos\left(\frac{7\pi 0}{8}\right)\right] = 20$$

$$X[1] = \frac{1}{\sqrt{2}}\left[10\cos\left(\frac{\pi}{8}\right) + 8\cos\left(\frac{3\pi}{8}\right) + 10\cos\left(\frac{5\pi}{8}\right) + 12\cos\left(\frac{7\pi}{8}\right)\right] = -1.84$$

$$X[2] = \frac{1}{\sqrt{2}}\left[10\cos\left(\frac{2\pi}{8}\right) + 8\cos\left(\frac{6\pi}{8}\right) + 10\cos\left(\frac{10\pi}{8}\right) + 12\cos\left(\frac{14\pi}{8}\right)\right] = 2$$

$$X[3] = \frac{1}{\sqrt{2}}\left[10\cos\left(\frac{3\pi}{8}\right) + 8\cos\left(\frac{9\pi}{8}\right) + 10\cos\left(\frac{15\pi}{8}\right) + 12\cos\left(\frac{21\pi}{8}\right)\right] = 0.76$$

Ejercicio 2

- Calcular la DCT inversa del ejemplo anterior

- Aplicamos la ecuación de síntesis

$X[m]$ = [20 -1.84 2 0.76]

$$x[n] = \sum_{m=0}^{N-1} \beta[m] X[m] \cos\left(\frac{(2n+1)\pi}{2N} m\right)$$

$$\beta[m] = \begin{cases} \sqrt{\frac{1}{N}} = \frac{1}{2}, & m = 0 \\ \sqrt{\frac{2}{N}} = \frac{1}{\sqrt{2}}, & m \neq 0 \end{cases}$$

$$x[n] = \beta[0]X[0]\cos\left(\frac{(2n+1)\pi}{8}0\right) + \beta[1]X[1]\cos\left(\frac{(2n+1)\pi}{8}\right)$$

$$+ \beta[2]X[2]\cos\left(\frac{(2n+1)\pi}{8}2\right) + \beta[3]X[3]\cos\left(\frac{(2n+1)\pi}{8}3\right)$$

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

unir

www.unir.net