16/07/2021 ①

Teorema 3.2.3 (Base discreta de cosenos - IV)
El conjunto $\left\{ \left(C^{(4N)}_{2k+1}(n)\right)_{n=0}^{N-1} = \sqrt{\frac{2}{N}}\left(\cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})\right)_{n=0}^{N-1} \right\}_{k=0}^{N-1}$
es b.o.n. de S_N.

D/ $\|C^{(4N)}_{2k+1}\|^2 = \langle C^{(4N)}_{2k+1}, C^{(4N)}_{2k+1}\rangle = \frac{2}{N}\sum_{n=0}^{N-1}\left[\cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})\right]^2$
$= \frac{2}{N}\sum_{n=0}^{N-1}\frac{1}{2}\left[1+\cos\frac{\pi}{N}(2k+1)(n+\frac{1}{2})\right] = 1 + \frac{1}{N}\sum_{n=0}^{N-1}\cos\frac{\pi}{N}(2k+1)(n+\frac{1}{2})$

Ej 3.2.2 $1+0$ porque $0<2k+1 \leq 2(N-1)+1 = 2N-1$

$0\leq \ell < k \leq N-1$

$\langle C^{(4N)}_{2k+1}, C^{(4N)}_{2\ell+1}\rangle = \sum_{n=0}^{N-1}\sqrt{\frac{2}{N}}\left[\cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})\right]\left[\sqrt{\frac{2}{N}}\cos\frac{\pi}{N}(\ell+\frac{1}{2})(n+\frac{1}{2})\right]$
$= \frac{1}{N}\sum_{n=0}^{N-1}\cos\frac{\pi}{N}(k+\ell+1)(n+\frac{1}{2}) + \frac{1}{N}\sum_{n=0}^{N-1}\cos\frac{\pi}{N}(k-\ell)(n+\frac{1}{2})$

Ej. 3.2.2 $0+0$ porque $0<k+\ell+1 \leq N-1+N-1+1 = 2N-1$ y
$0<k-\ell\leq N-1$

Es base porque tiene N elementos y S_N tiene dimensión N ∎

Con respecto a la base del Teor. 3.2.3, si $f\in S_N$ tenemos

$$f(n) = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1}\langle f, C^{(4N)}_{2k+1}\rangle \cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2}) \quad (3.2.4)$$

Def 3.2.4. $f\in S_N$, su transformada discreta de coseno -IV es $\{\hat{f}_{IV}(k)\}_{k=0}^{N-1}$ donde si $0\leq k\leq N-1$ (DCT-IV)

$$\hat{f}_{IV}(k) = \langle f, C^{(4N)}_{2k+1}\rangle = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} f(n)\cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2}). \quad (3.2.5)$$

La fórmula (3.2.4) se escribe

$$f(n) = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1}\hat{f}_{IV}(k)\cos\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2}) \quad (3.2.6)$$

que es la IDCT-IV (Inversa)

Ejercicio 3.2.6 Escribir DCT-IV y su inversa en S_2 ($N=2$) en forma matricial

Si $\hat{f}_{IV}(0) = \left[f(0)\cos\frac{\pi}{8} + f(1)\cos\frac{3\pi}{8}\right]$

$\hat{f}_{IV}(1) = \left[f(0)\cos\frac{3\pi}{8} + f(1)\cos\frac{9\pi}{8}\right]$

$$\begin{pmatrix} \hat{f}_{IV}(0) \\ \hat{f}_{IV}(1) \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\frac{\pi}{8} & \operatorname{sen}\frac{\pi}{8} \\ \operatorname{sen}\frac{\pi}{8} & -\cos\frac{\pi}{8} \end{pmatrix}}_{C} \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}$$

$\cos^2\frac{\pi}{8} = \frac{1+\cos\frac{\pi}{4}}{2} = \frac{1+\frac{1}{\sqrt{2}}}{2}$

$= \frac{\sqrt{2}+1}{2\sqrt{2}} \Rightarrow \cos\frac{\pi}{8} = \sqrt{\frac{\sqrt{2}+1}{2\sqrt{2}}}$

$\cos\frac{3\pi}{8} = \operatorname{sen}\frac{\pi}{8}$

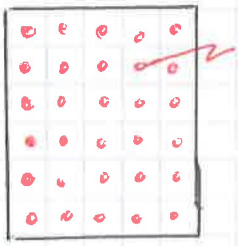
$= \sqrt{\frac{\sqrt{2}-1}{2\sqrt{2}}}$

$\cos\frac{9\pi}{8} = -\cos\frac{\pi}{8}$

$$C\,C^T = \begin{pmatrix} \cos\frac{\pi}{8} & \operatorname{sen}\frac{\pi}{8} \\ \operatorname{sen}\frac{\pi}{8} & -\cos\frac{\pi}{8} \end{pmatrix}\begin{pmatrix} \cos\pi/8 & \operatorname{sen}\pi/8 \\ \operatorname{sen}\pi/8 & -\cos\pi/8 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow C^{-1} = C^T$$

$$\begin{pmatrix} f(0) \\ f(1) \end{pmatrix} = \begin{pmatrix} \cos\frac{\pi}{8} & \operatorname{sen}\frac{\pi}{8} \\ \operatorname{sen}\frac{\pi}{8} & -\cos\frac{\pi}{8} \end{pmatrix}\begin{pmatrix} \hat{f}_{IV}(0) \\ \hat{f}_{IV}(1) \end{pmatrix}$$

3.2.3 Bases discretas para imágenes



Pixel (Picture element)

$f = \{f(n,m)\}_{n=0,\,m=0}^{N-1,\,M-1}$ imagen discretizada, de tamaño $N\times M$, es decir $f \in S_{N,M} \approx \mathbb{C}^{N\times M}$ que es de Hilbert con $\langle f, g\rangle = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(n,m)\overline{g(n,m)}$

Una base para $S_{N,M}$ se puede hacer multiplicando una base de S_N con otra de S_M

Proposición 3.2.5 Sea $\{e_k\}_{k=1}^{N-1}$ b.o.n. de S_N y $\{f_\ell\}_{\ell=1}^{M-1}$ b.o.n. de S_M. Para $0\leq k\leq N-1$ y $0\leq \ell\leq M-1$, definir

$$e_{k,\ell}(n,m) = e_k(n)\,f_\ell(m) \quad , \; 0\leq n\leq N-1,\; 0\leq m\leq M-1.$$

El conjunto $\{e_{k,\ell}\}_{k=0,\,\ell=0}^{N-1,\,M-1}$ es b.o.n de $S_{N,M}$.

D/ $\langle e_{k_1,\ell_1}, e_{k_2,\ell_2}\rangle = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} e_{k_1,\ell_1}(n,m)\,\overline{e_{k_2,\ell_2}(n,m)}$

$$= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} e_{k_1}(n) f_{\ell_1}(m)\, \overline{e_{k_2}(n)}\, \overline{f_{\ell_2}(m)} =$$

$$= \left(\sum_{n=0}^{N-1} e_{k_1}(n)\overline{e_{k_2}(n)}\right)\left(\sum_{m=0}^{M-1} f_{\ell_1}(m)\overline{f_{\ell_2}(m)}\right) = \langle e_{k_1}, e_{k_2}\rangle\langle f_{\ell_1}, f_{\ell_2}\rangle$$

$= \delta_{k_1,k_2}\,\delta_{\ell_1,\ell_2}$. Como hay $N\times M$ elementos y $S_{N,M}$ tiene dim $N\times M$, ya es base. ∎

Con respecto a la base $\{e_{k,\ell}\}_{k,\ell}$ de la Prop. 3.2.5

$$f(n,m) = \sum_{k=0}^{N-1} \langle f, e_{k,\ell}\rangle e_{k,\ell}(n,m) \qquad \begin{matrix} k=0,\dots,N-1 \\ \ell=0,\dots,M-1 \end{matrix}.$$

$\hat{f}(k,\ell) = \langle f, e_{k,\ell}\rangle$ es la transformada de Fourier 2-D con respecto a esta base:

$$\hat{f}(k,\ell) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(n,m)\,\overline{e_{k,\ell}(n,m)}$$

Por ejemplo, si $N=M=8$ y en S_N usamos DC-I base del Teor 3.2.1 e, d.

$$\left\{ \frac{\lambda_k}{2}\left(\cos\frac{k\pi}{8}\left(n+\tfrac{1}{2}\right)\right)_{n=0}^{7} \right\}, \quad \lambda_k = \begin{cases} 1/\sqrt{2} & \text{si } k=0 \\ 1 & \text{si } 1\le k\le 7 \end{cases}.$$

Base de $S_{N,N}$:

$$\left\{\left(\frac{\lambda_k}{2}\frac{\lambda_\ell}{2}\left[\cos\frac{k\pi}{8}\left(n+\tfrac{1}{2}\right)\right]\left[\cos\frac{\ell\pi}{8}\left(m+\tfrac{1}{2}\right)\right]\right)_{n=0,\,m=0}^{7,\;7}\right\}_{k=0,\,\ell=0}^{7,\;7}$$

La transformada 2D-DCT-I

$$\hat{f}_I(k,\ell) = \frac{\lambda_k\lambda_\ell}{4}\cdot\sum_{n=0}^{7}\sum_{m=0}^{7} f(n,m)\left[\cos\frac{k\pi}{8}\left(n+\tfrac{1}{2}\right)\right]\left[\cos\frac{\ell\pi}{8}\left(m+\tfrac{1}{2}\right)\right]$$

(3.2.6)

3.3. ALGORITMOS RÁPIDOS PARA CALCULAR DCT-I y DCT-IV

3.3.1. Algoritmo rápido para DCT-I

Vamos a probar que la DCT-I de $f \in S_N$ puede calcularse usando la DFT de una señal $\tilde{f}$ de tamaño $2N$. Cuando $N = 2^q$, $2N = 2^{q+1}$ y se puede usar FFT. Con FFT, el nº de operaciones es $\frac{3}{2}(2N)\log_2(2N) \approx N\log_2 N$, en lugar de los N^2 operaciones si DCT-I se calcula directamente.

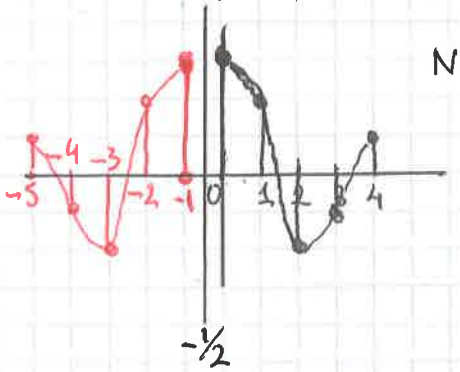
Recuerde (DCT-I, Def 3.2.2)

$$\hat{f}_I(k) = \lambda_k \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} f(n) \cos \frac{\pi k}{N}\left(n+\frac{1}{2}\right) \qquad (3.3.1)$$

donde

$$\lambda_k = \begin{cases} 1/\sqrt{2} & \text{si } k=0 \\ 1 & \text{si } 1 \le k \le N-1 \end{cases}$$

Extender $f = (f(n))_{n=1}^{N-1}$ a una señal $\tilde{f}$ de tamaño $2N$ que coincide con f en $0 \le n \le N-1$ y es simétrica respecto a $-\frac{1}{2}$:



$$\tilde{f}(n) = \begin{cases} f(n), & 0 \le n \le N-1 \\ f(-1-n), & -N \le n \le -1 \end{cases} \qquad (3.3.2)$$

Recuerde (DFT para señales de tamaño $2N$, Def 2.5.2)

$$\hat{\tilde{f}}(k) = \frac{1}{\sqrt{2N}} \sum_{n=-N}^{N-1} \tilde{f}(n)\, e^{-\frac{2\pi i k n}{2N}}, \quad k=-N,\dots,N-1 \qquad (3.3.3)$$

Prop 3.3.1 $f \in S_N$, $\hat{f}_I(k) = \lambda_k e^{-\frac{\pi i k}{2N}} \hat{\tilde{f}}(k)$, $k=0,\dots,N-1$ con $\tilde{f}$ como en (3.3.2).

"$\hat{f}_I$ puede calcularse usando DFT de $\tilde{f}$ y multiplicando por $\lambda_k e^{-\frac{\pi i k}{2N}}$"

D/ $\hat{\tilde{f}}(k) \overset{0\le k\le N-1}{\underset{(3.3.3)}{=\!=}} \frac{1}{\sqrt{2N}} \sum_{k=-N}^{N-1} \tilde{f}(n)\, e^{-\frac{\pi i k n}{N}}$

$$= \frac{1}{\sqrt{2N}} \sum_{n=-N}^{-1} f(-1-n)\, e^{-\frac{\pi i k n}{N}} + \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} f(n)\, e^{-\frac{\pi i k n}{N}}$$

$m=-1-n$

$$= \frac{1}{\sqrt{2N}} \sum_{m=N-1}^{0} f(m)\, e^{\frac{\pi i k(m+1)}{N}} + \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{-\frac{\pi i k n}{N}}$$

$m=n$

$$= \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{\frac{\pi i k n}{N}} \cdot e^{\frac{\pi i k}{N}} + \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{-\frac{\pi i k n}{N}}$$

$$= \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n) \left[e^{\frac{\pi i k n}{N}} e^{\pi i \frac{k}{N}} + e^{-\frac{\pi i k n}{N}} \right]$$

$$= \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{\frac{\pi i k}{2N}} \left[e^{\frac{\pi i k n}{N}} e^{\frac{\pi i k}{2N}} + e^{-\frac{\pi i k n}{N}} e^{-\frac{\pi i k}{2N}} \right]$$

$$= \frac{1}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{\frac{\pi i k}{2N}} \left[e^{\frac{\pi i k}{N}(n+\frac{1}{2})} + e^{-\frac{\pi i k}{N}(n+\frac{1}{2})} \right]$$

$$= \frac{2}{\sqrt{2N}} \sum_{n=0}^{N-1} f(n)\, e^{\frac{\pi i k}{2N}} \left(\cos \frac{\pi k}{N}(n+\tfrac{1}{2}) \right) \overset{(3.3.1)}{=\!=} \frac{1}{\lambda_k} e^{\frac{\pi i k}{2N}} \hat{f}_I(k)$$

Entonces

$$\hat{f}_I(k) = \lambda_k\, e^{-\frac{\pi i k}{2N}}\, \hat{\tilde{f}}(k).$$

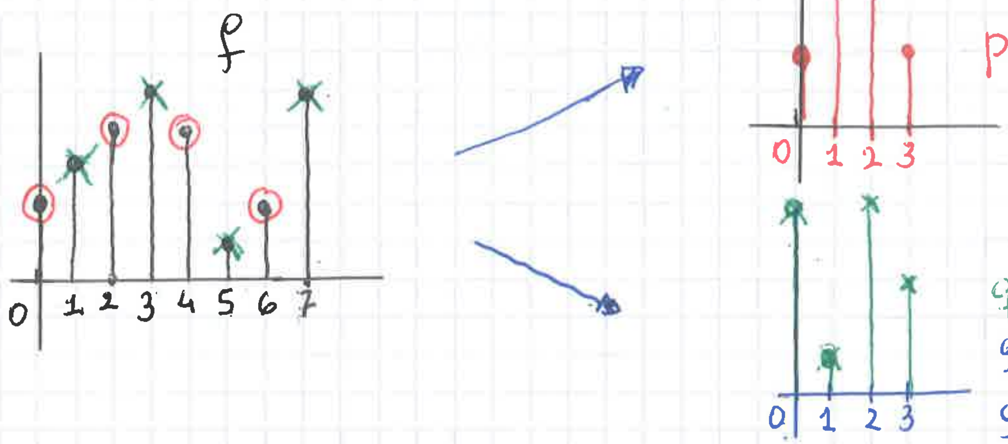
3.3.2. Algoritmo rápido para calcular DCT-IV

$N = 2^q$, $f = (f(n))_{n=0}^{N-1} \in S_N$. Separar la señal f en dos señales, cada una de tamaño $N/2 = 2^{q-1}$, una con los términos pares y la otra con los impares

$$p(n) = f(2n), \qquad q(n) = f(N-(2n+1))$$

$$n = 0, 1, \dots, \frac{N}{2}-1$$

$N=8=2^3$



$q(n)=f(8-(2n+1))$
$q(0)=f(8-1)=f(7)$
$q(1)=f(8-3)=f(5)$

Recuerde (DCT-IV, Def 2.3.4), $f\in S_N$

$$\hat{f}_{IV}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} f(n)\cos\frac{\pi}{N}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right),\quad 0\leq k\leq N-1$$

Recuerde (DFT para tamaño $N/2$, Def 2.5.2) para $g\in S_{N/2}$

$$\hat{g}(k)=\frac{1}{\sqrt{N/2}}\sum_{n=0}^{\frac{N}{2}-1} g(n)\, e^{-\frac{2\pi i k n}{N/2}},\quad k=0,\dots,\frac{N}{2}-1.$$

Ejercicio 3.3.1. Sea $f=(f(0),f(1))=(1,-1)\in S_2$

(a) Calcular $\hat{f}_{IV}(0)$ y $\hat{f}_{IV}(1)$

(b) Escribir la señal $g(0)=[p(0)+i\,q(0)]\,e^{-\frac{\pi i}{8}}$ y calcular $\hat{g}(0)$

(c) Relacionar los resultados anteriores

S/ (a) $\hat{f}_{IV}(0)=1\left[f(0)\cos\frac{\pi}{8}+f(1)\cos\frac{3\pi}{8}\right]=\cos\frac{\pi}{8}-\operatorname{sen}\frac{\pi}{8}$

$\hat{f}_{IV}(1)=1\left[f(0)\cos\frac{3\pi}{8}+f(1)\cos\frac{9\pi}{8}\right]=\operatorname{sen}\frac{\pi}{8}+1\cos\frac{\pi}{8}$

(b) $p(0)=f(0)=1,\quad q(0)=f(2-(2\cdot 0+1))=f(1)=-1$

$g(0)=(1-i)\,e^{-\frac{\pi i}{8}}$

$\hat{g}(0)=1\,g(0)=(1-i)\,e^{-\frac{\pi i}{8}}=(1-i)\left(\cos\frac{\pi}{8}-i\operatorname{sen}\frac{\pi}{8}\right)$

$=\left(\cos\frac{\pi}{8}-\operatorname{sen}\frac{\pi}{8}\right)+i\left(-\cos\frac{\pi}{8}-\operatorname{sen}\frac{\pi}{8}\right)$

(c) $\hat{f}_{IV}(0)=\operatorname{Real}\hat{g}(0)$, $\hat{f}_{IV}(1)=-\operatorname{Img}\hat{g}(0)$.

Resultado: Se puede probar que si $f \in S_N$ con N par y $g(n) = [p(n) + i\, q(n)]\, e^{-\frac{\pi i}{N}(n+\frac{1}{4})}$, $n = 0, 1, \dots, \frac{N}{2}-1$

$$\hat{f}_{IV}(2k) = \text{Real}\left\{ e^{-\frac{\pi i k}{N}} \hat{g}(n) \right\}, \quad k = 0, 1, \dots \frac{N}{2}-1$$

y

$$\hat{f}_{IV}(2k+1) = -\text{Img}\left\{ e^{-\frac{\pi i k}{N}} \hat{g}(n) \right\}, \quad k = 0, 1, \dots, \frac{N}{2}-1$$

3.4 EL FORMATO JPEG PARA IMÁGENES

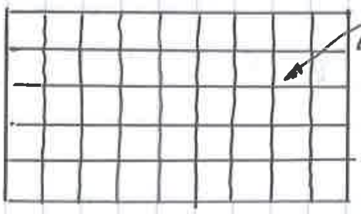
JPEG = Joint Photographic Expert Group

(Estandart para tratar e intercambiar imágenes)

JPG 80 – Algoritmo que se comercializó en 1990

JPG 2000 – Algoritmo que se comercializó en 2000

3.4.1 Representación del color



pixel (picture element)

A cada pixel hay que asignarle un valor que represente su color

Fotografía en escala de grises: los tonos de grises se representan con números enteros desde 0 hasta 255

0 = Negro (Black) $\qquad$ 255 = Blanco (White)

$255 = 2^8 - 1$ $\qquad$ Base 2: $a_n a_{n-1} \dots a_1 a_0\,(2) = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2 + a_0$ $\qquad a_g = 0, 1$

1bit $\qquad$ 1Byte

$69 = \underbrace{01000101}_{\text{1Byte}}\,(2)$ $\qquad$ $255 = 11111111\,(2)$

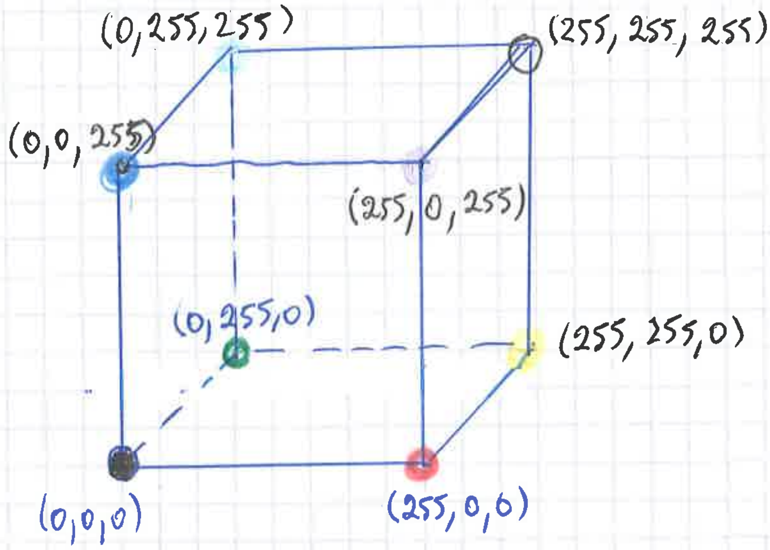
$1 = 00000001\,(2)$

Colores - Sistema RGB (Red, Green, Blue)

A cada color primario se le pueden asignar de 0 a 255 valores

(R, G, B)	
(255, 0, 0)	Red
(0, 255, 0)	Green
(0, 0, 255)	Azul
(0, 0, 0)	Black

(R, G, B)	
(255, 255, 0)	Yellow
(0, 255, 255)	Cyan
(255, 0, 255)	Magenta
(255, 255, 255)	White



Los colores de la diagonal (a, a, a) con $0 \leq a \leq 255$ son las escalas de grises

3.4.2 Transformación del color

$$(R, G, B) \longrightarrow (Y, Cb, Cr)$$

(Brillo, Saturación del azul, Saturación del rojo)

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.493 \\ 0.493 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix}$$

Salen números con decimales, que se redondean

$16 \leq Y \leq 220$, $16 \leq Cb \leq 255$, $16 \leq Cr \leq 255$

3.4.3 Down sampling (Submuestreo)

Y es la mas importante y no se hace downsampling)

Solo se hace downsampling en las coordenadas Cb y Cr: solo se consideran la mitad de los valores

Original de $N \times N$ pixeles : $N^2 \times N^2 \times N^2 = N^6$ datos

Downsampling : $N^2 \times \left(\frac{N}{2}\right)^2 \times \left(\frac{N}{2}\right)^2 = \frac{N^6}{16}$ datos

3.4.4. Transformada de coseno discreta (JPG 80)

$N \times N$, $N = 2^b$. Se divide la imagen en bloques de 8×8 para ser procesados independientemente. Para $f \in S_{N,N}$ hemos calculado $\{\hat{f}_I(k,l)\}_{k=0,\, l=0}^{7,\, 7}$.

El coeficiente $\hat{f}(0,0)$ es

$$\hat{f}_I(0,0) = \frac{1}{8} \sum_{n=0}^{7} \sum_{m=0}^{7} f(n,m)$$ (Coef DC - Direct component)

- Antes de calcular DCT - I, se resta -128 a todos los componentes para que ~~salgan~~ se obtengan nºs entre [-128, 128]

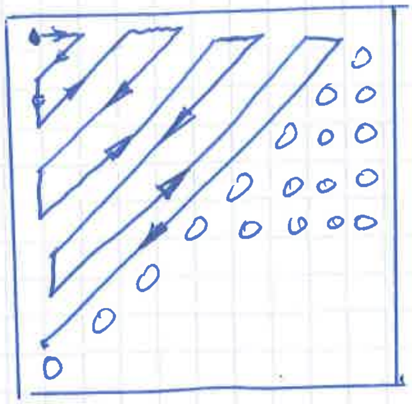
3.4.5. Cuantización

$$\begin{bmatrix} 16 & \cdots & 16 & \\ & & \vdots & \\ & & 24 & \\ & & & 90 \end{bmatrix}$$

Cada uno de los 64 números $\{\hat{f}_I(k,l)\}_{k=0,\, l=0}^{7,\, 7}$ se dividen por un número dado en la matriz de cuantización de JPEG y se redondea

$$-\frac{415}{16} = -25'93 \sim -26$$

3.4.6. Codificación



Se ordenan los 64 elementos en zig-zag

Algoritmo de Huffman