



Recuperación de Información Multimedia

Dominio de Frecuencias

CC5213 – Recuperación de Información Multimedia

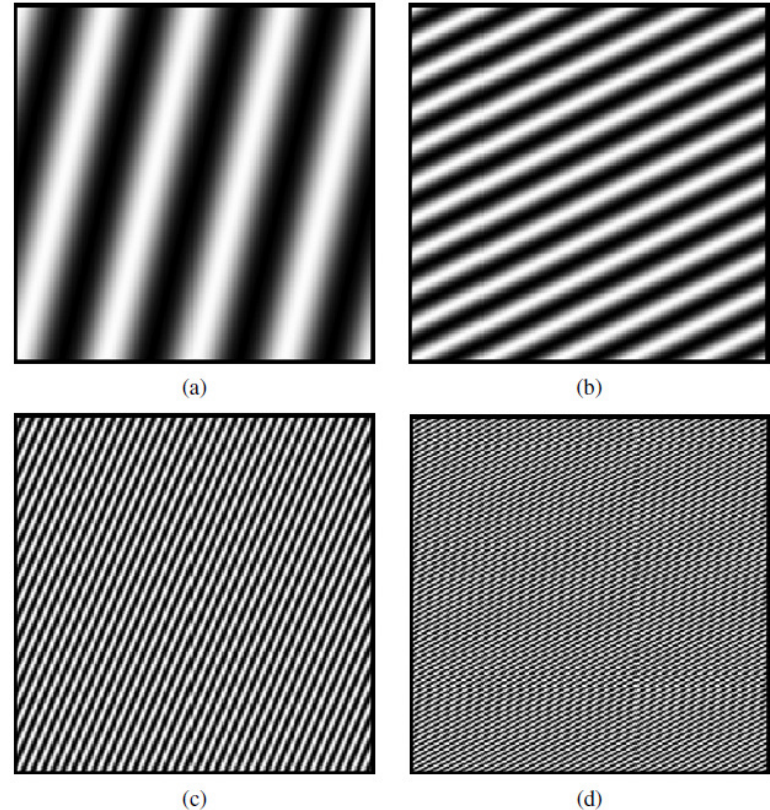
Departamento de Ciencias de la Computación

Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2020

Dominio de Frecuencias

- Una imagen se puede considerar como la unión de muchas funciones sinusoidales
- Parámetros (u, v) corresponden a la frecuencia vertical y horizontal



(a) $u = 1, v = 4$; (b) $u = 10, v = 5$; (c) $u = 15, v = 35$;

Dominio de Frecuencias

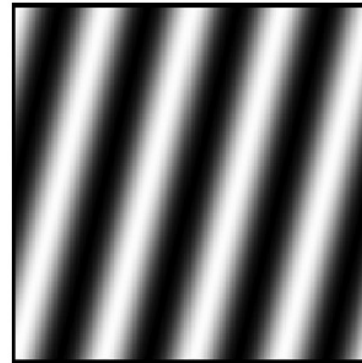
- Con el par (u, v) se puede calcular

- Dirección:

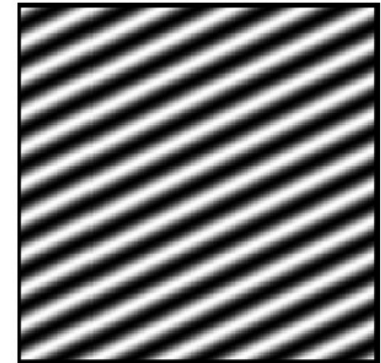
$$\theta = \arctan\left(\frac{v}{u}\right)$$

- Frecuencia radial:

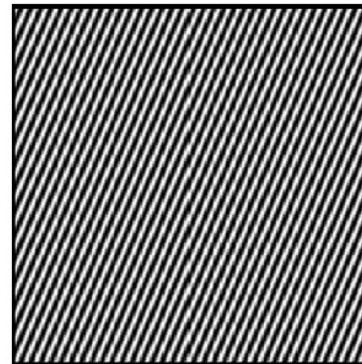
$$\Omega = \sqrt{u^2 + v^2}$$



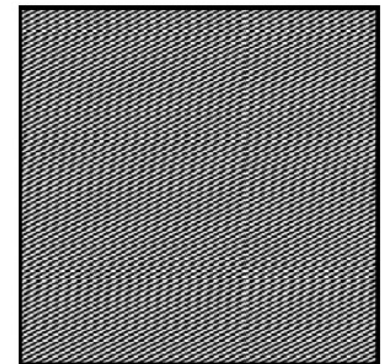
(a)



(b)



(c)



(d)



Transformada de Fourier 2D

FT 2D (“forward”):

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

FT 2D Inversa:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(ux+vy)} dx dy$$

Los valores $f(x,y)$ y $F(u,v)$ son complejos

$$e^{i\theta} = \cos \theta + i \sin \theta$$



Transf. Fourier Discreta (DFT) 2D

- Dada una imagen $I(x, y)$ de $M \times N$ pixeles

DFT 2D (“forward”):

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) e^{-i2\pi(\frac{u}{M}x + \frac{v}{N}y)}$$

DFT 2D Inversa:

$$I(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(\frac{u}{M}x + \frac{v}{N}y)}$$

Donde u y v son ciclos por imagen

El factor de escala ($1/MN$) y la polaridad (signo de la exponencial) se pueden ubicar tanto en la DFT forward como en la inversa.



Propiedades DFT

- La DFT tiene periodicidad M y N :

$$F(u, v) = F(u + kM, v + lN) \quad \forall k, l \in \mathbb{Z}$$

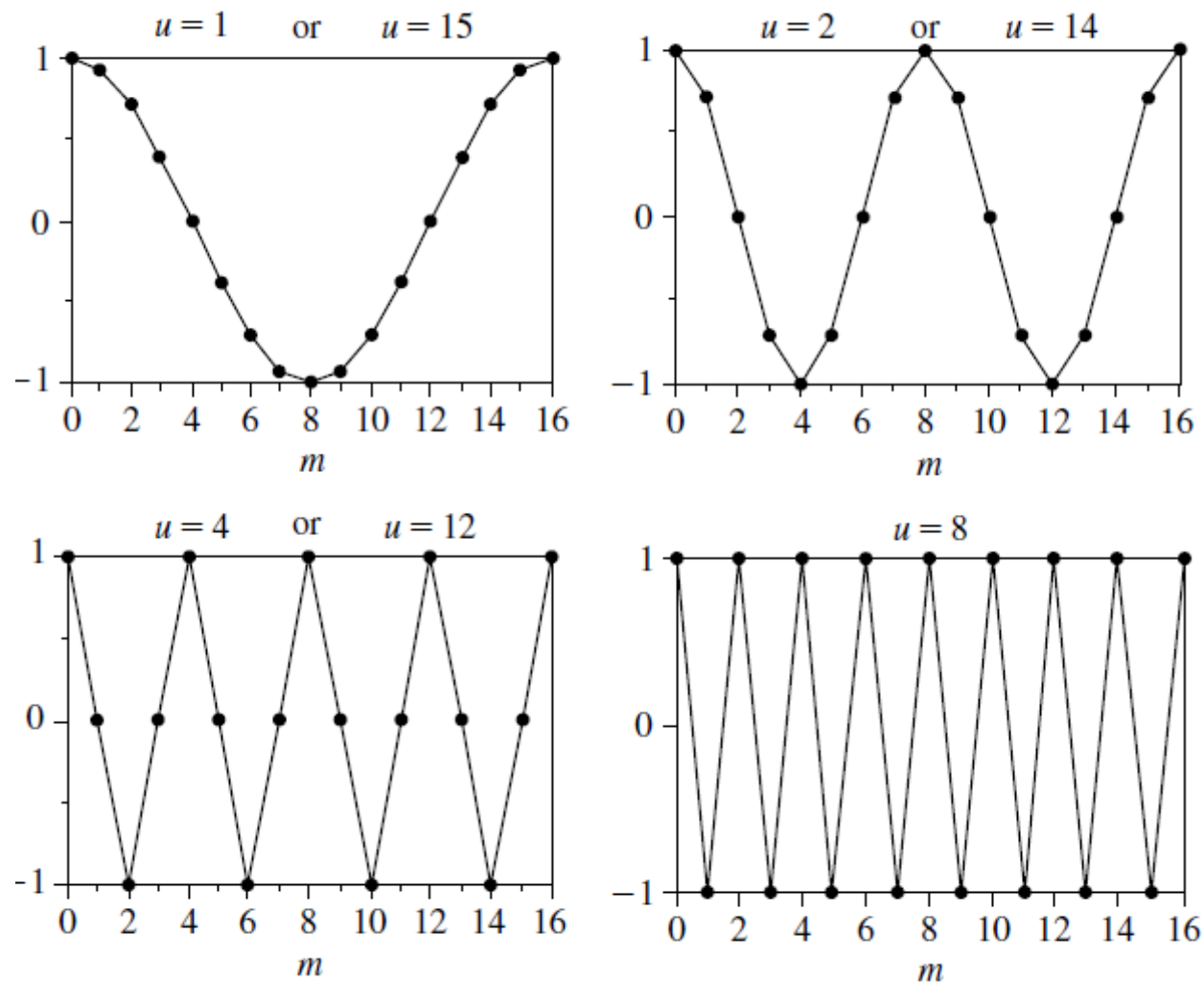
- Si $I(x, y)$ son números reales (caso común) existe simetría de los conjugados

$$F(u, v) = F^*(M - u, N - v)$$

- Implica simetría de las magnitudes:

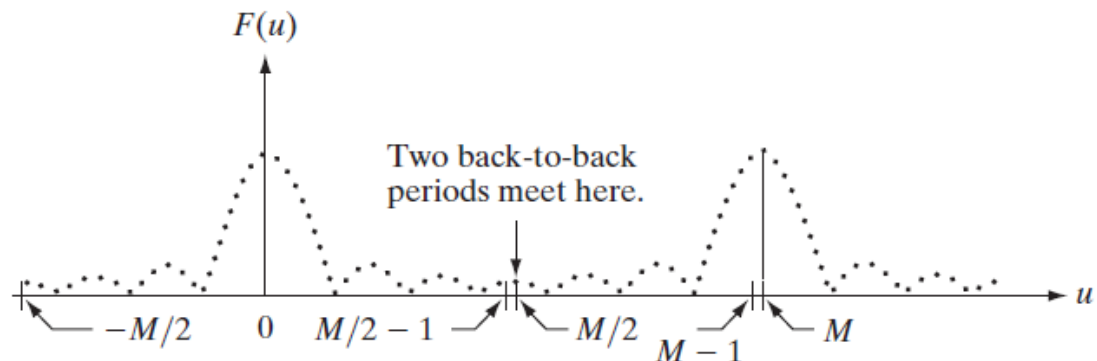
$$|F(u, v)| = |F(M - u, N - v)|$$

Efecto de la Discretización



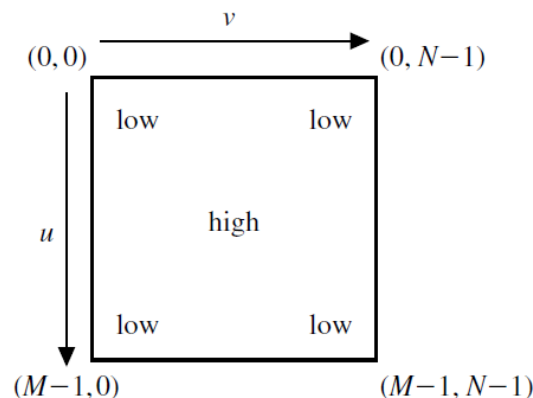
Discretizando una Señal

- Sampling Theorem (Nyquist-Shannon): para digitalizar una señal continua se deben obtener ejemplos (samples) a una frecuencia mayor al doble de la máxima frecuencia en la señal
- Para señales 1D:
 - Para medir la frecuencia f se deben tener más de $2f$ samples
 - Si se tienen n samples basta considerar hasta la frecuencia $n/2$

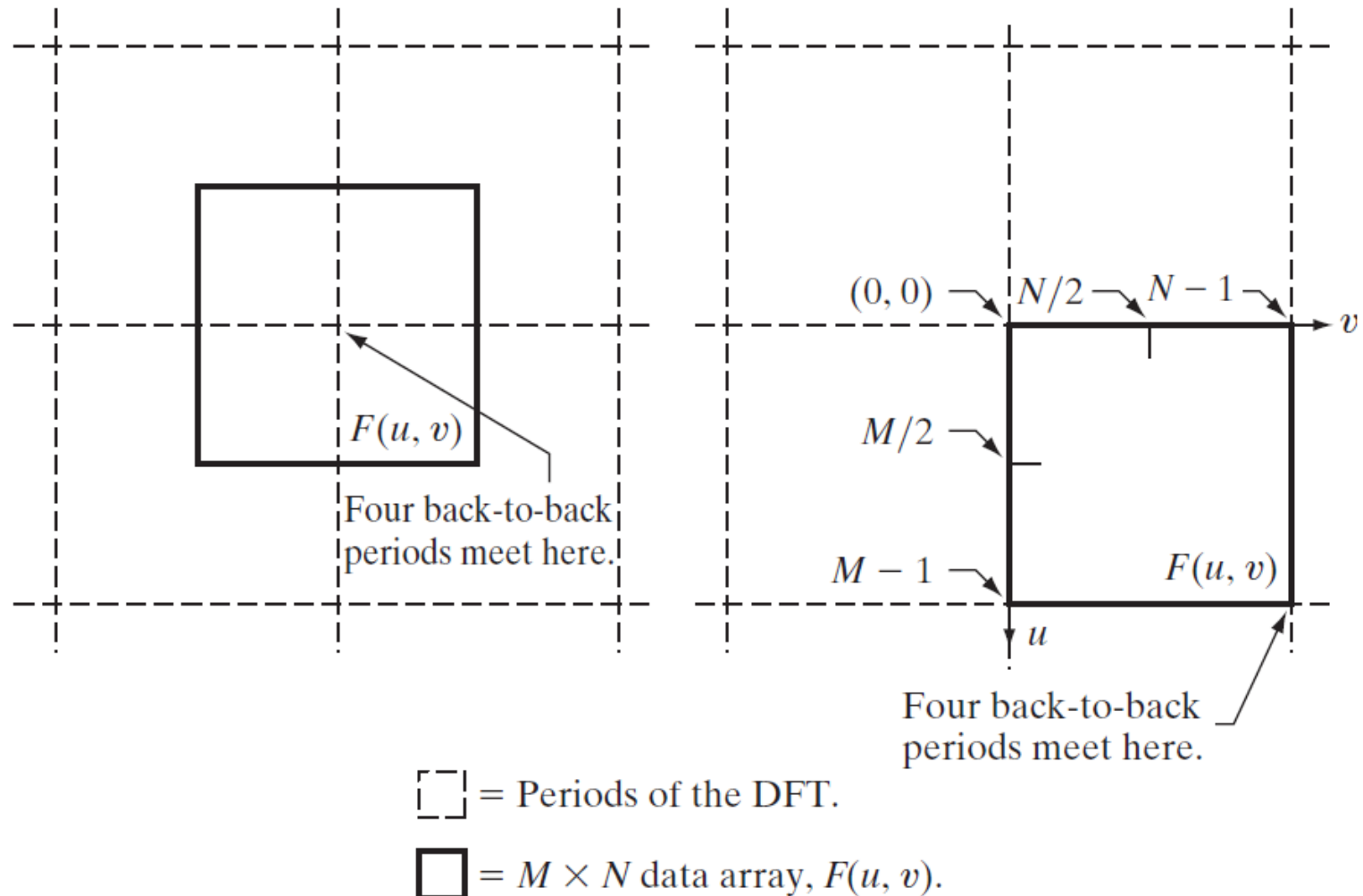


Efecto de la Discretización 2D

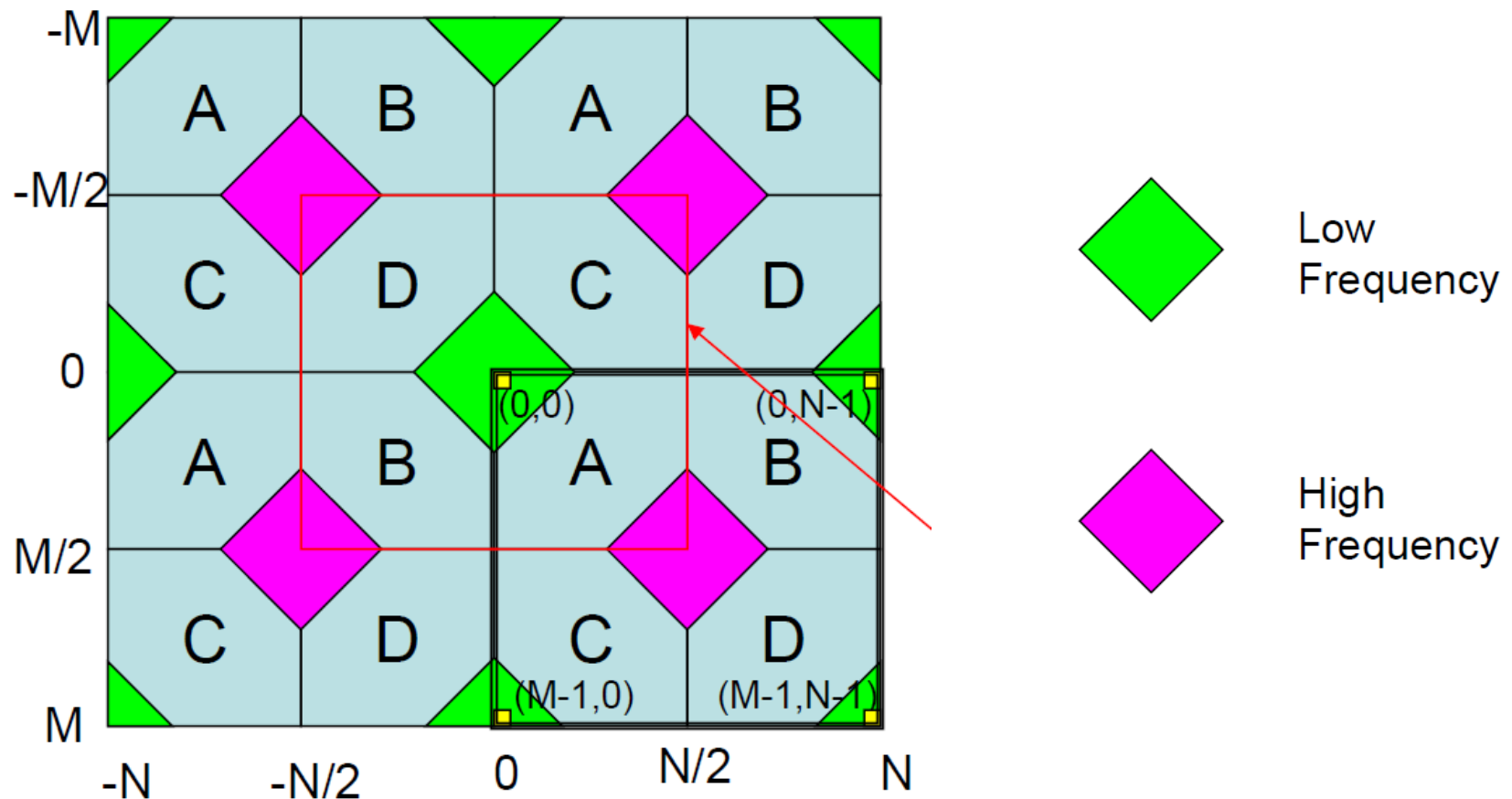
- Para imágenes con $M \times N$ samples se consideran hasta las frecuencias (M, N) :
 - Las frecuencias más altas se encuentran en torno a $(M/2, N/2)$
 - Las magnitudes tienen simetría con respecto al origen (ver siguiente slide)



Simetría DFT en 2D



Simetría DFT en 2D





Visualización de la DFT

- Mostrar la magnitud de los valores complejos
- Usualmente el origen se mueve al centro, ya sea copiando los valores o mediante:

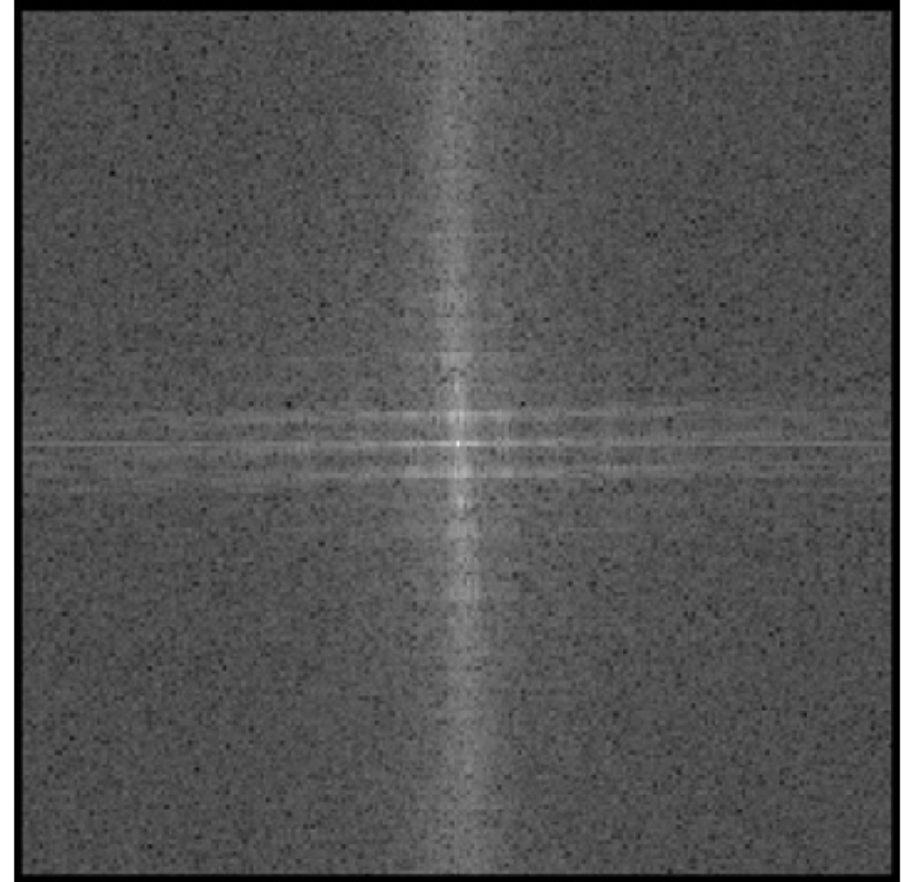
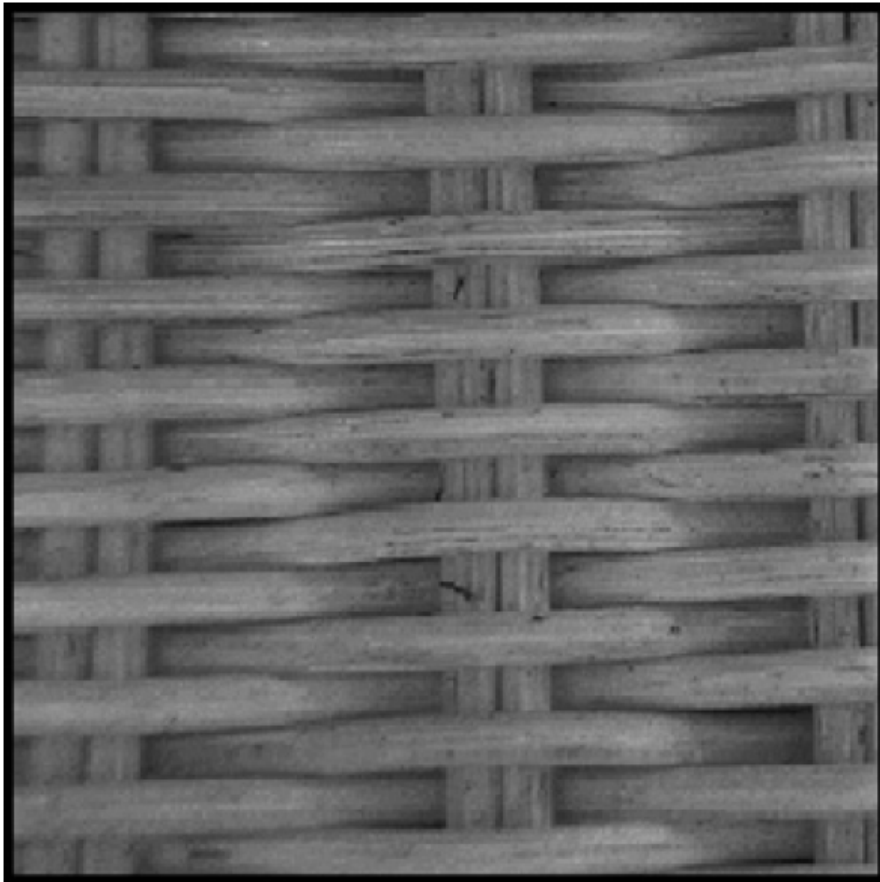
$$DFT[(-1)^{x+y} f(x, y)] = F(u - M/2, v - N/2)$$

$$\boxed{(-1)^{m+n} = e^{j\pi(m+n)}}$$

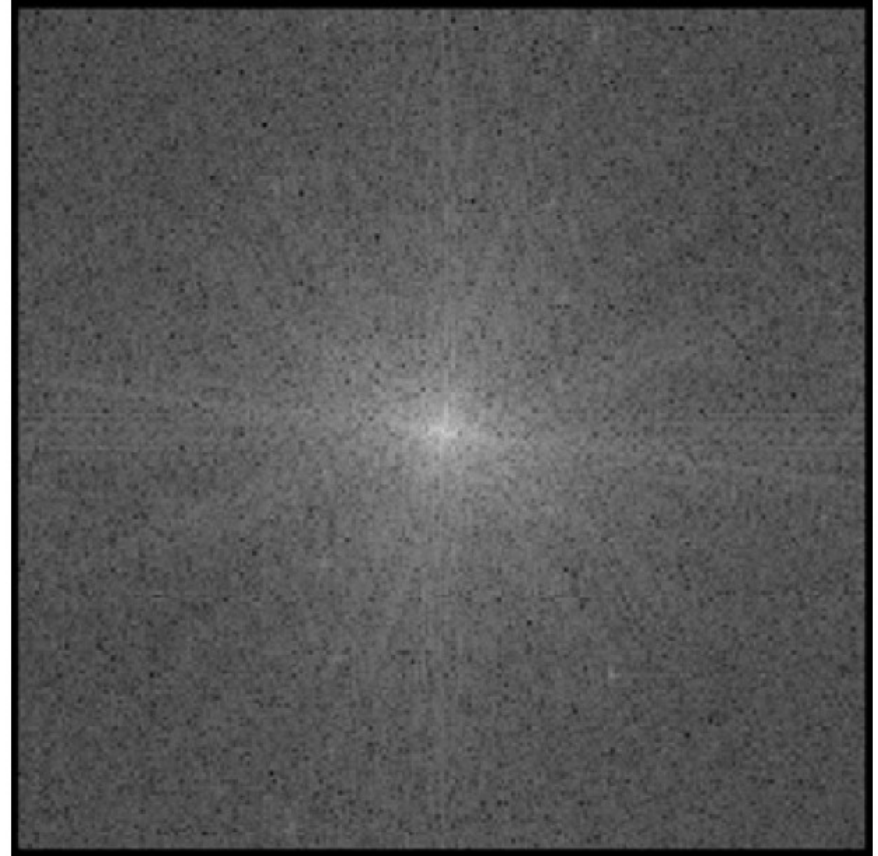
- Las frecuencias bajas usualmente son mucho más grandes que las altas → visualizar el logaritmo de las magnitudes:

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

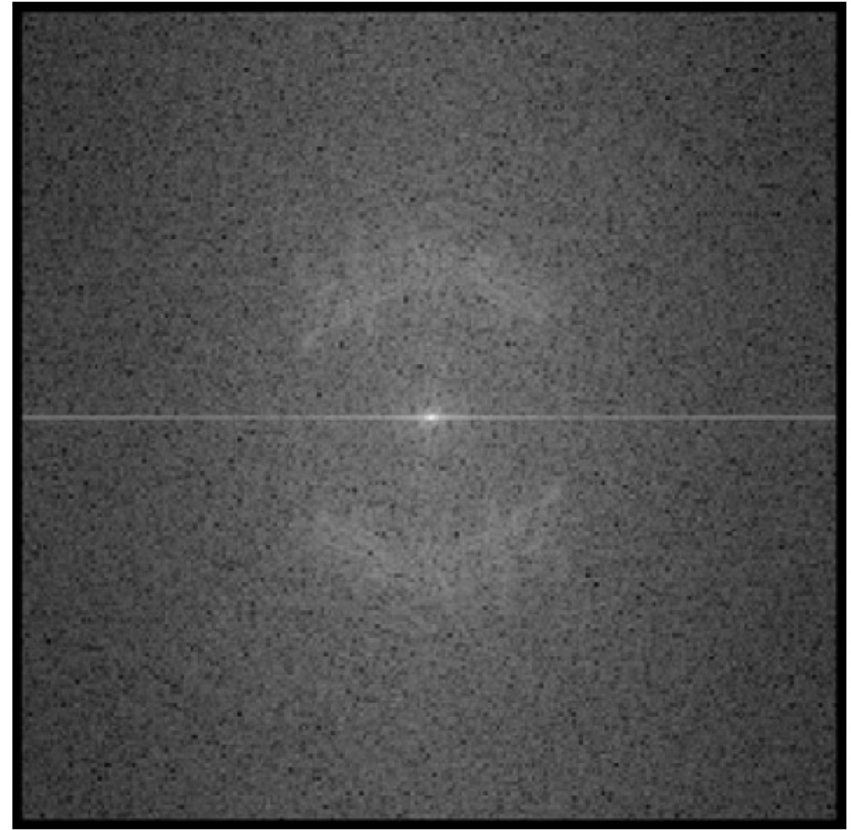
$$\log_2 (1 + |F(u, v)|)$$



- Patrones repetitivos aumentan los peaks

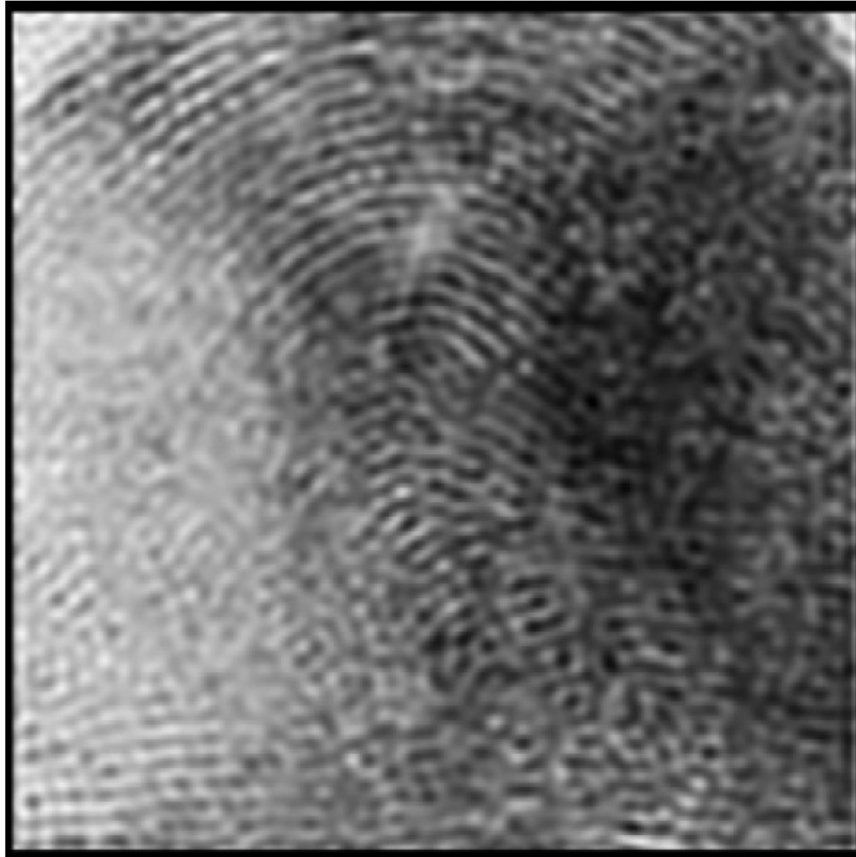
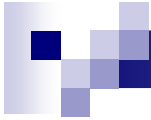


- Colores suaves aumentan las frecuencias bajas
- Bordes aumentan las intensidades medias a altas

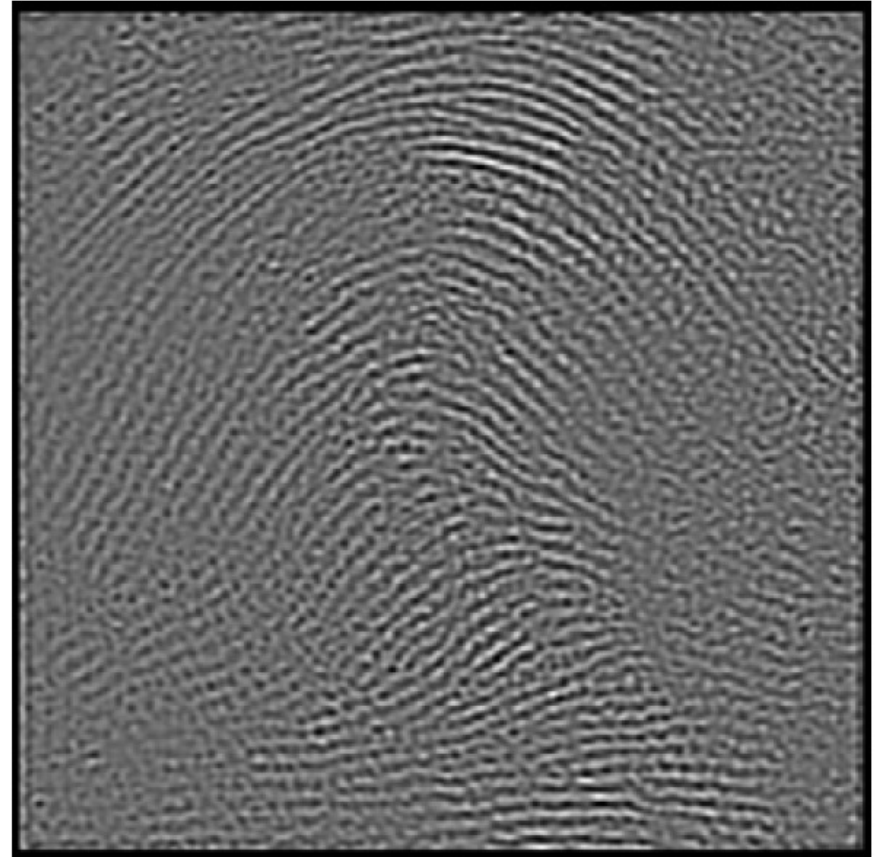


- Las líneas de la huella aumentan las frecuencias medias

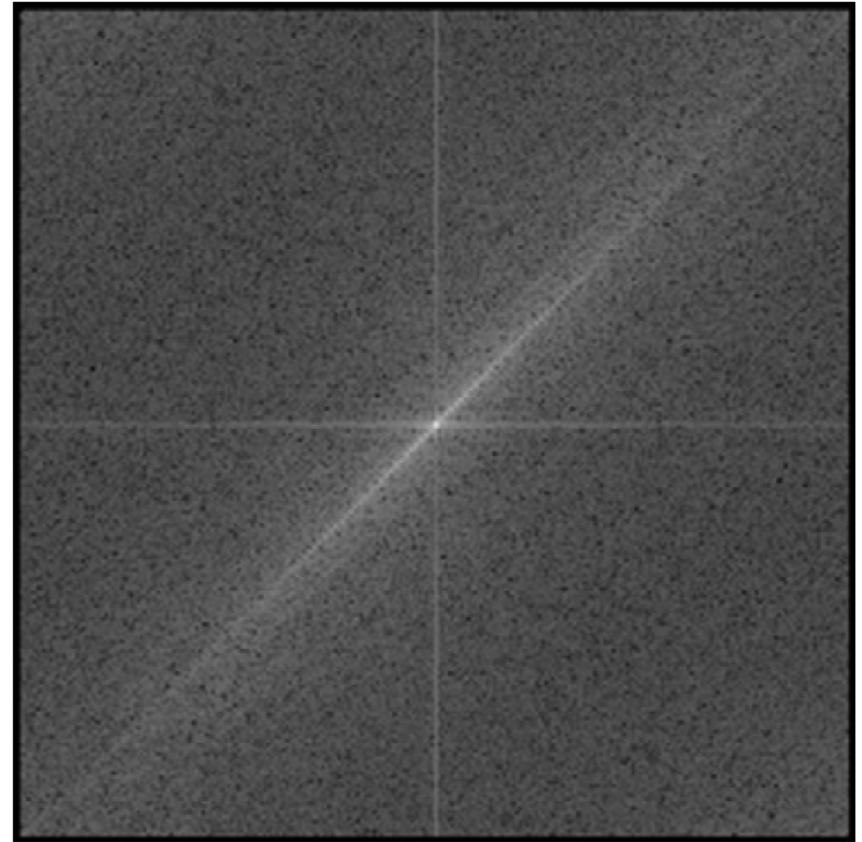
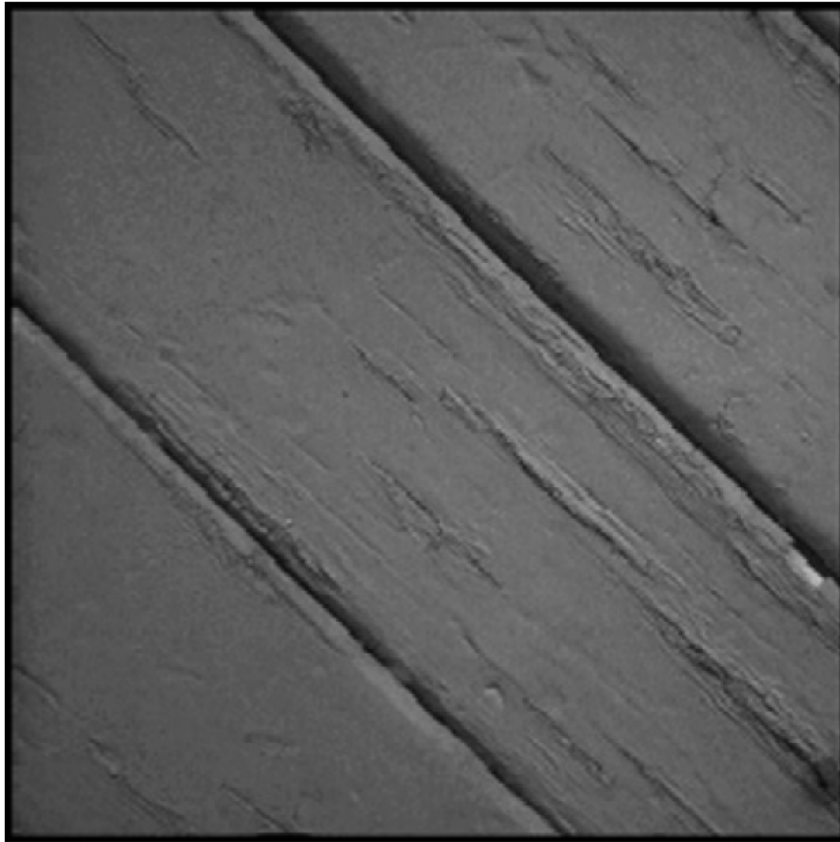
Ver Bovik, cap 5.



Seleccionando sólo
frecuencias bajas

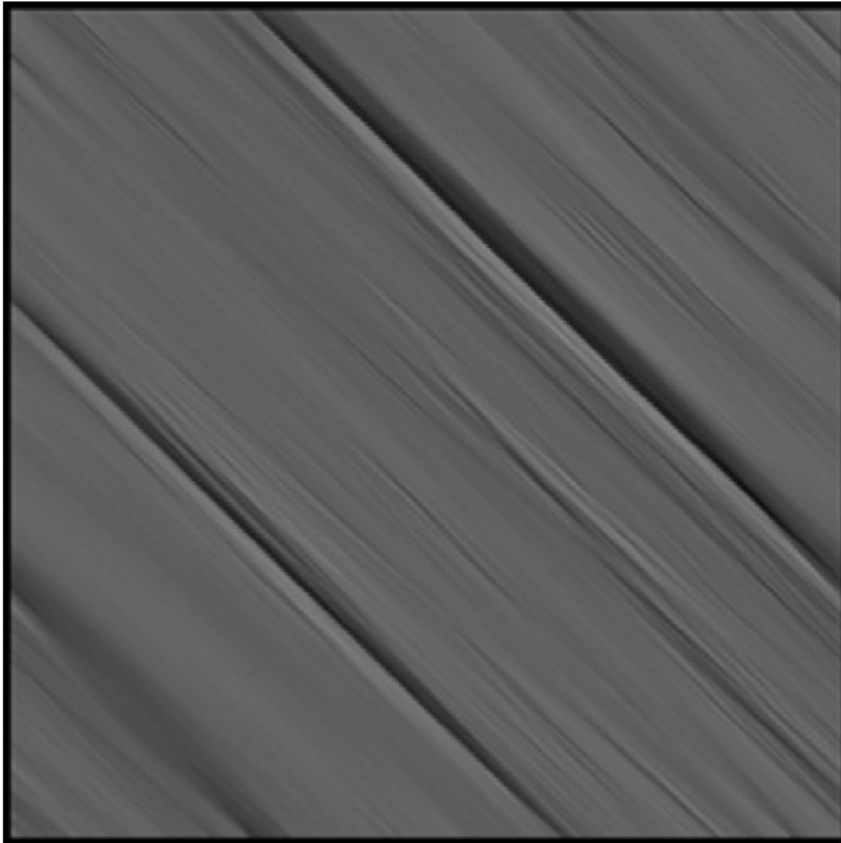


Seleccionando sólo
frecuencias medias

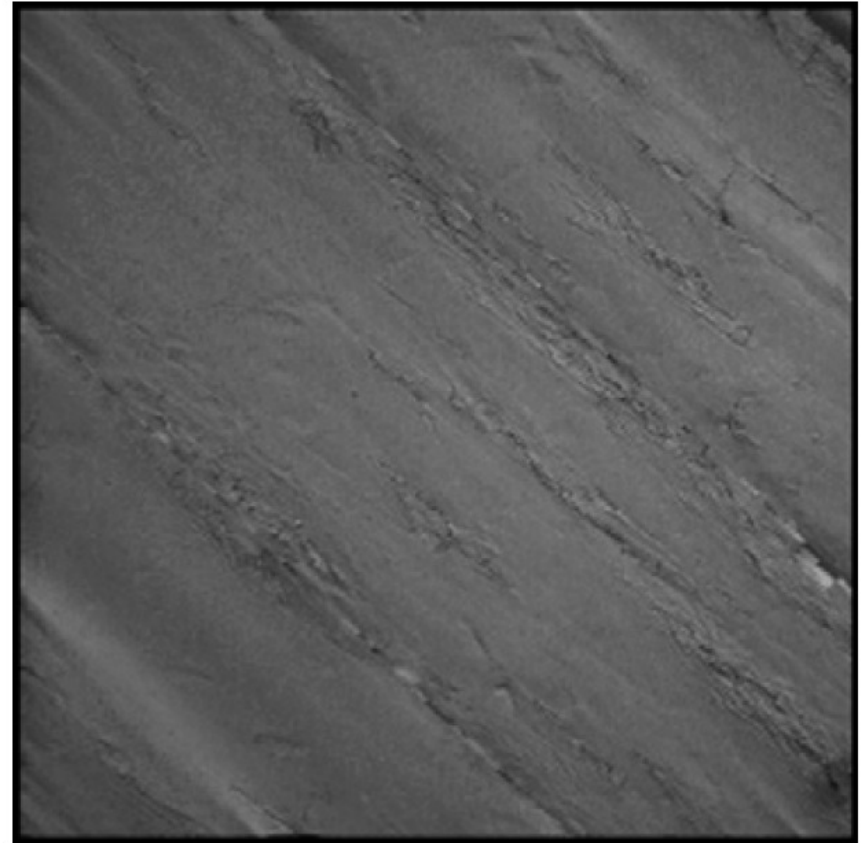


- Fuerte componente direccional entorno a los 45° y a los 225°

Ver Bovik, cap 5.



Seleccionando sólo entre
 40° a 50° y 220° a 230°



Seleccionando sólo entre
 50° a 220° y -130° a 40°



Fast Fourier Transform (FFT)

- Algoritmo naive requiere $O(n^2)$ operaciones para calcular n términos
 - n es el número de píxeles de la imagen
- Algoritmo FFT requiere $O(n \log n)$ operaciones para calcular los mismos valores
 - También se usa para la transformada inversa
 - Algoritmo “divide and conquer”. Se obtiene la mayor performance cuando el tamaño es potencia de 2



Teorema de Convolución

- La FT de la convolución entre una imagen y un kernel es lo mismo que el producto (multiplicación) de la FT de la imagen y la FT del kernel
 - La convolución entre una imagen y un kernel se puede calcular usando el dominio de frecuencias

$$\begin{aligned} f(x, y) * k(x, y) &= g(x, y) \\ F(u, v) \cdot K(u, v) &= G(u, v) \end{aligned}$$



Teorema de Convolución

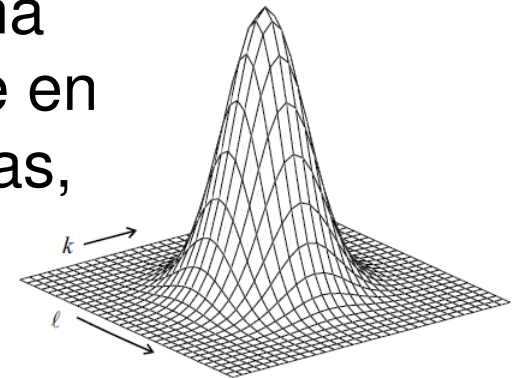
- Para una imagen de $(W \times H) = n$ pixeles y un kernel de $(A \times B) = m$ pixeles
 - La implementación “tradicional” de la convolución requiere $O(n \cdot m)$ operaciones
- Usando el dominio de frecuencias la convolución requiere de 4 pasos:
 - FFT imagen, FFT kernel (escalado a n), multiplicación punto a punto, FFT inversa
 - $O(n \log n) + O(n \log n) + O(n) + O(n \log n) \rightarrow \text{total: } O(n \log n)$.
- Para filtros grandes el costo $O(n \log n)$ puede ser mejor que $O(n \cdot m)$

Teorema de Convolución

■ Ejemplo:

- Convolución de una imagen con un kernel gaussiano es un Blur (ver capítulo 1)
- La FT de una gaussiana es una gaussiana
- Teorema de Convolución implica que una convolución con una gaussiana consiste en llevar la imagen al dominio de frecuencias, ubicar una gaussiana en el centro y multiplicar punto a punto

→ Blur gaussiano es lo mismo que reducir el valor de las frecuencias altas, i.e., seleccionar las frecuencias bajas (filtro “pasa bajos”)





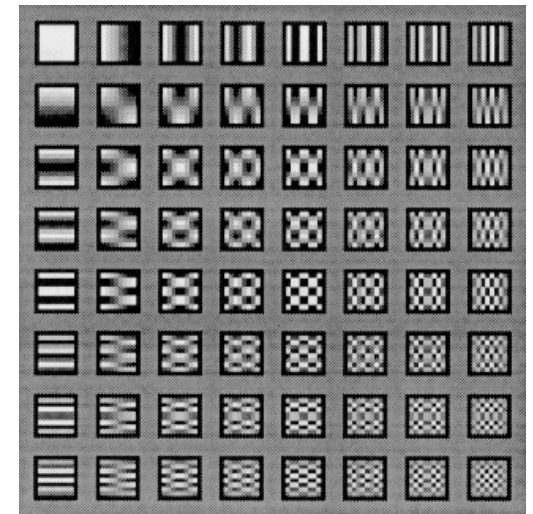
Transformada Coseno

- La Transformada Coseno es similar a la Transformada de Fourier, pero sólo usando cosenos
 - Los coeficientes son reales (en vez de complejos)
 - Requiere el doble de coeficientes que la Transformada de Fourier
- Para imágenes, interesa la Transformada Discreta Coseno (DCT)

Transformada Discreta Coseno (DCT)

- Un caso común de uso de la DCT es transformar ventanas de 8x8 pixeles
- Dada una imagen de 8x8, la DCT encuentra los pesos para ponderar cada una de las 64 bases y reconstruir la imagen
 - 8x8 pixeles se convierten en 8x8 pesos
 - Los pesos se usan para aproximar o redondear la imagen original
- Ver ejemplo en:

http://en.wikipedia.org/wiki/Discrete_cosine_transform



bases

Múltiples Definiciones de DCT

■ Por ejemplo:

$$c_k = \sum_{n=0}^{N-1} \sqrt{n = \begin{cases} \frac{1}{N} & \text{if } n=0 \\ \frac{2}{N} & \text{else} \end{cases}} \cdot x_n \cdot \cos\left(-\pi \frac{(2k+1)n}{N}\right)$$

$$\alpha(u, v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \cos\left[(2x+1)\frac{\pi u}{2M}\right] \cos\left[(2y+1)\frac{\pi v}{2M}\right] I(x, y)$$

$$\text{with } \alpha(u, v) = \frac{2}{\sqrt{NM}} C(u)C(v) \text{ and } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{otherwise} \end{cases}$$



DCT para la codificación JPEG

Zona 8x8 en imagen

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

DCT

915.6	451.3	25.6	-12.6	16.1	-12.3	7.9	-7.3
216.8	19.8	-228.2	-25.7	23.0	-0.1	6.4	2.0
-2.0	-77.4	-23.8	102.9	45.2	-23.7	-4.4	-5.1
30.1	2.4	19.5	28.6	-51.1	-32.5	12.3	4.5
5.1	-22.1	-2.2	-1.9	-17.4	20.8	23.2	-14.5
-0.4	-0.8	7.5	6.2	-9.6	5.7	-9.5	-19.9
5.3	-5.3	-2.4	-2.4	-3.5	-2.1	10.0	11.0
0.9	0.7	-7.7	9.3	2.7	-5.4	-6.7	2.5

Tabla de cuantización

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

Resultado de cuantizar

57	41	2	0	0	0	0	0
18	1	-16	-1	0	0	0	0
0	-5	-1	4	1	0	0	0
2	0	0	0	-1	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

Muchos ceros → buena compresión lossless al usar RLE.



Resultado de descomprimir

181	185	196	208	203	159	86	27
191	189	197	203	178	118	58	25
192	193	197	185	136	72	36	33
184	199	195	151	90	48	38	43
185	207	185	110	52	43	49	44
201	198	151	74	32	40	48	38
213	161	92	47	32	35	41	45
216	122	43	32	39	32	36	58

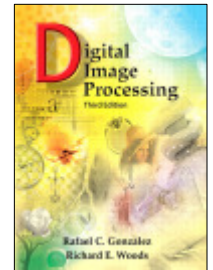
Original

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

Bibliografía

- **Digital Image Processing.**
González et al. 2008

- cap 4 (Fourier)



- **The Essential Guide To Image Processing.** Bovik. 2009.

- cap 5 (Fourier)

- pág. 425 (DCT, compresión JPG)

