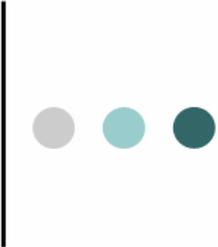
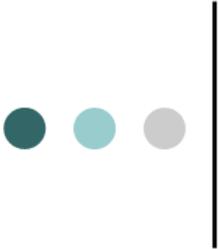


ELO 313 –Procesamiento Digital de Señales con Aplicaciones



# Conceptos Básicos para el Procesamiento Digital de Imágenes

**Matías Zañartu, Ph.D.**

Departamento de Electrónica

Universidad Técnica Federico Santa María



# Aplicaciones del Procesamiento Digital de Imágenes

- Cámaras Digitales
- Scanners Digitales
- Internet y Aplicaciones multimedia
- Televisión Digital
- Aplicaciones Biomédicas
- Astronomía
- Sensado Remoto
- Automatización y Seguridad
- Aplicaciones Industriales

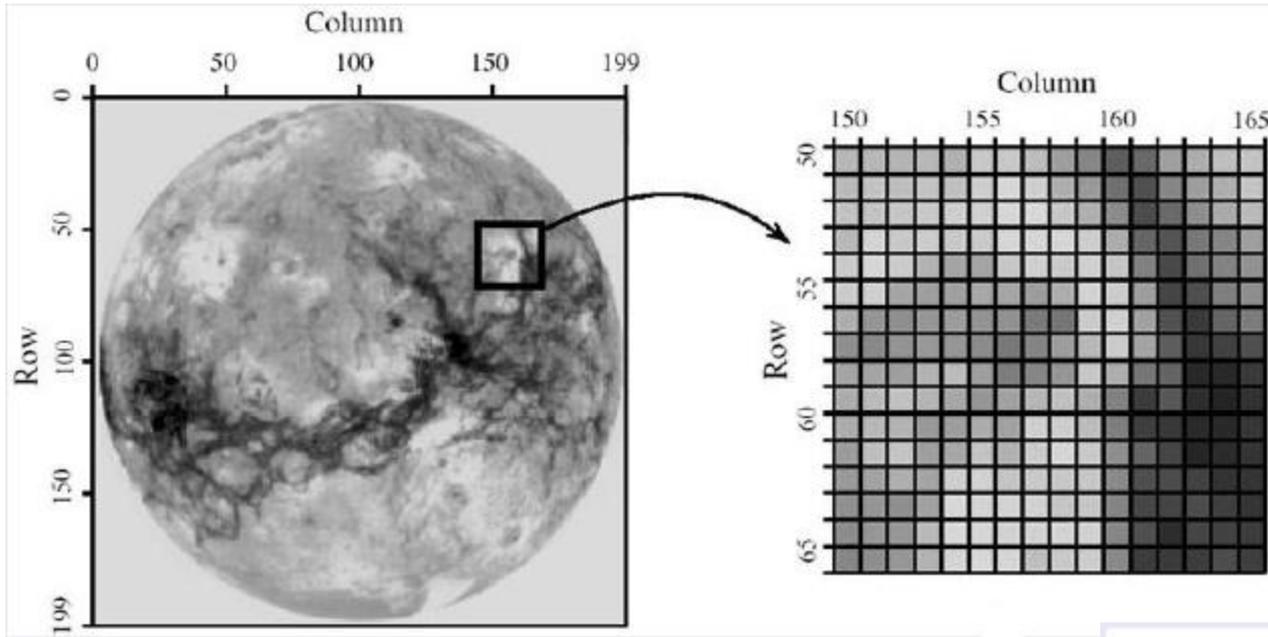


# Conceptos Generales

- **Imagen:** Representación óptica de objetos iluminados por una fuente de luz. También información representada en 2D
- Digitalmente, es una función de variables espaciales discretas
- Imágenes monocromáticas  $f(i, j)$  representa la intensidad de la luz en la coordenada  $(i, j)$  donde  $i = 1, \dots, M$  y  $j = 1, \dots, N$

$$f(i, j) = \begin{bmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, N) \\ f(2, 1) & f(2, 2) & \cdots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix}$$

- Cada elemento de la matriz de  $M \times N$  es un pixel

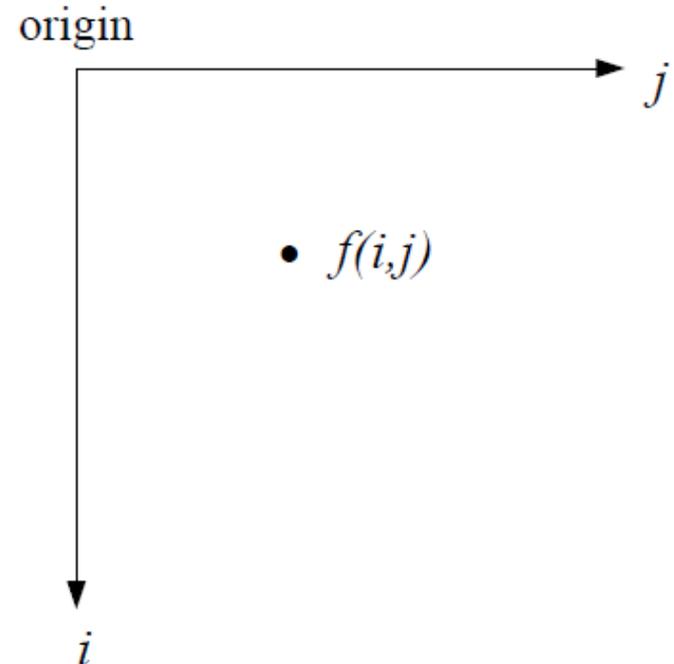


Representación de imágenes monocromáticas

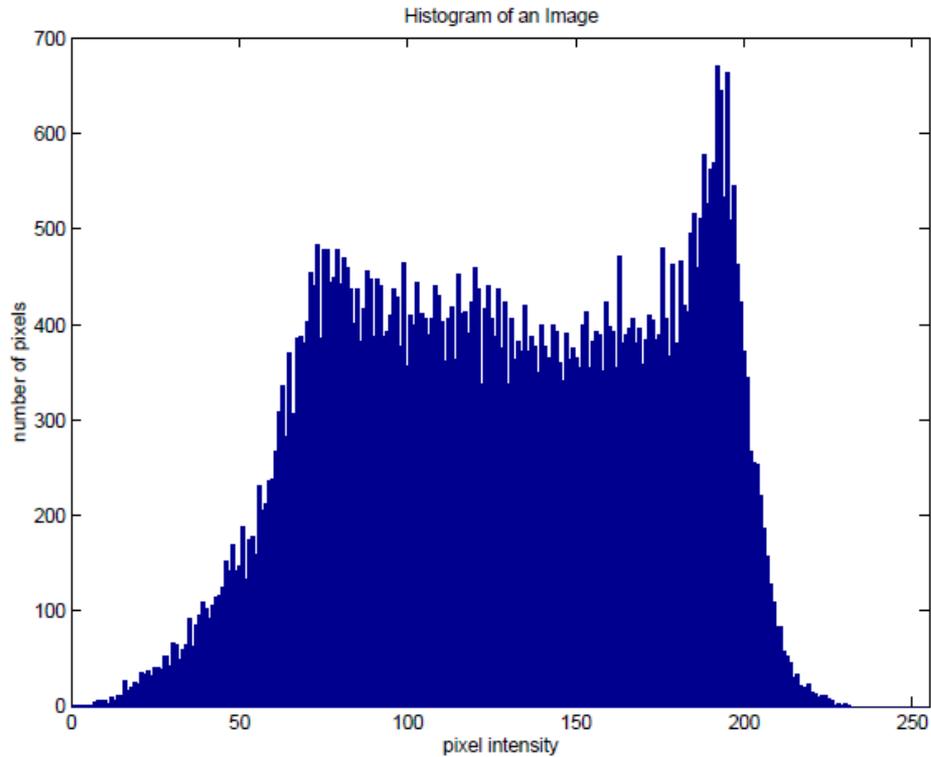
		Column															
		150				155				160				165			
50	183	183	181	184	177	200	200	189	159	135	94	105	160	174	191	196	
	186	195	190	195	191	205	216	206	174	153	112	80	134	157	174	196	
	194	196	198	201	206	209	215	216	199	175	140	77	106	142	170	186	
	184	212	200	204	201	202	214	214	214	205	173	102	84	120	134	159	
55	202	215	203	179	165	165	199	207	202	208	197	129	73	112	131	146	
	203	208	166	159	160	168	166	157	174	211	204	158	69	79	127	143	
	174	149	143	151	156	148	146	123	118	203	208	162	81	58	101	125	
	143	137	147	153	150	140	121	133	137	184	203	164	94	56	66	80	
60	164	165	159	179	188	159	126	134	150	199	174	119	100	41	41	58	
	173	187	193	181	167	151	162	182	192	175	129	60	88	47	37	50	
	172	184	179	153	158	172	163	207	205	188	127	63	56	43	42	55	
	156	191	196	159	167	195	178	203	214	201	143	101	69	38	44	52	
65	154	163	175	165	207	211	197	201	201	199	138	79	76	67	51	53	
	144	150	143	162	215	212	211	209	197	198	133	71	69	77	63	53	
	140	151	150	185	215	214	210	210	211	209	135	80	45	69	66	60	
	135	143	151	179	213	216	214	191	201	205	138	61	59	61	77	63	

# Conceptos Generales

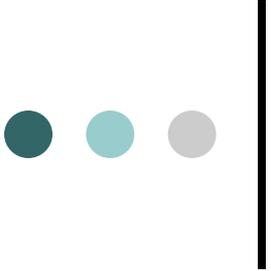
- En imágenes, el origen se considera en la esquina superior izquierda
- En la respuesta a impulso de filtros (PSF), el origen se considera en el centro
- Los pixeles toman valores enteros en un rango  $0 \leq f(i,j) \leq L_{max}$ , donde 0 representa mínima energía (negro) y  $L_{max}$  máxima energía (blanco)
- Imágenes monocromáticas son de 8 bits  $\rightarrow [0,255]$



# Histograma de una imagen



- Cantidad de píxeles en una imagen (eje y) que toman una cierta intensidad (eje x)

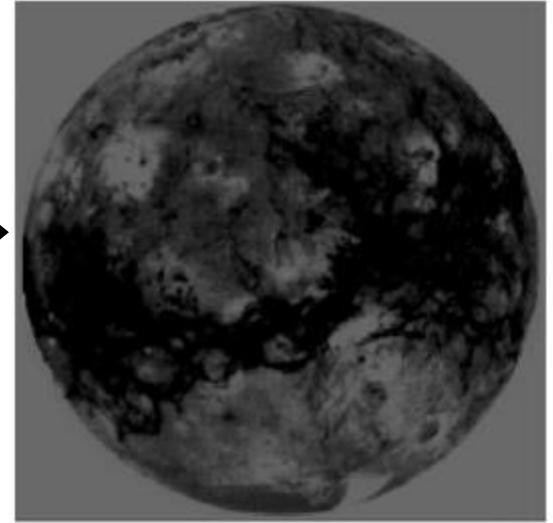
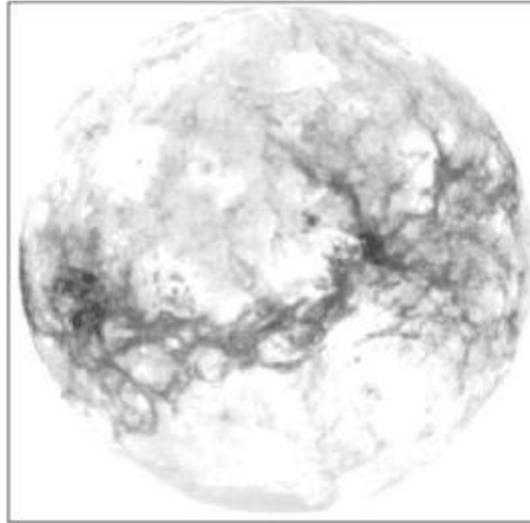


# Propiedades Básicas

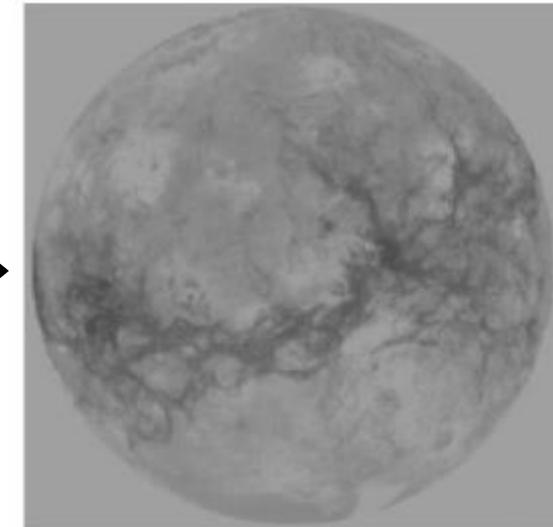
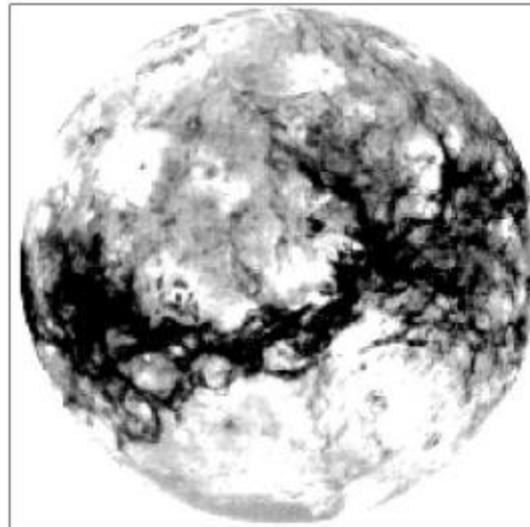
- **Brillo:** Intensidad total de la imagen
- Cambios de brillo → sumar o restar una constante a cada pixel (desplazar el histograma hacia la izquierda o derecha)
- **Contraste:** Diferencia de intensidad en la imagen
- Cambios de contraste → cambiar la relación entre la intensidad de los pixeles (compresión o expansión del histograma)
- Los pixeles con valores intermedios se truncan a un entero, y si se salen por abajo o por arriba, toman los valores 0 y 255 respectivamente

# Brillo y Contraste

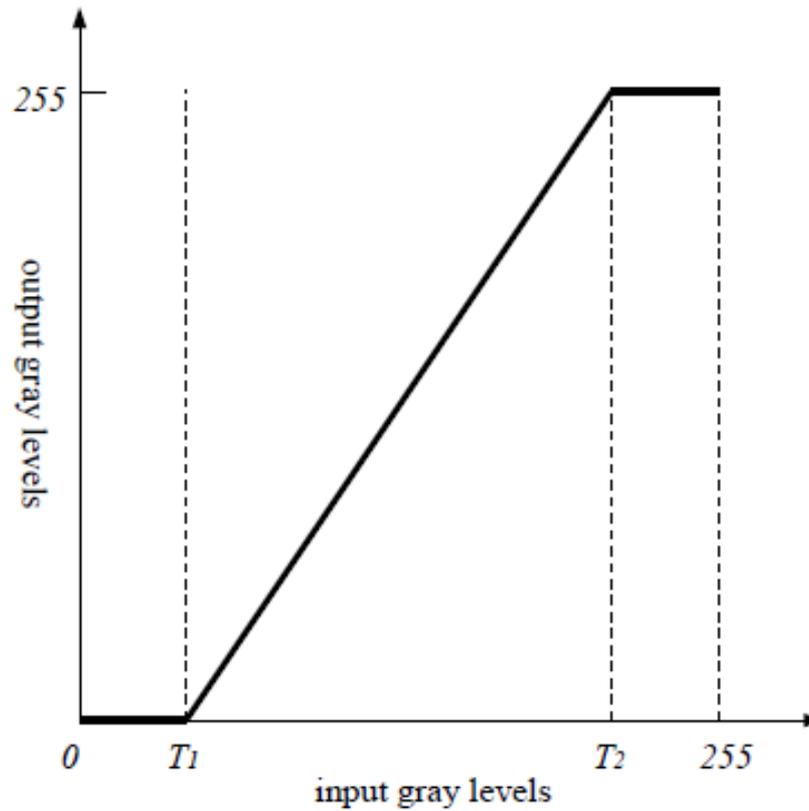
Cambios de  
Brillo



Cambios de  
Contraste



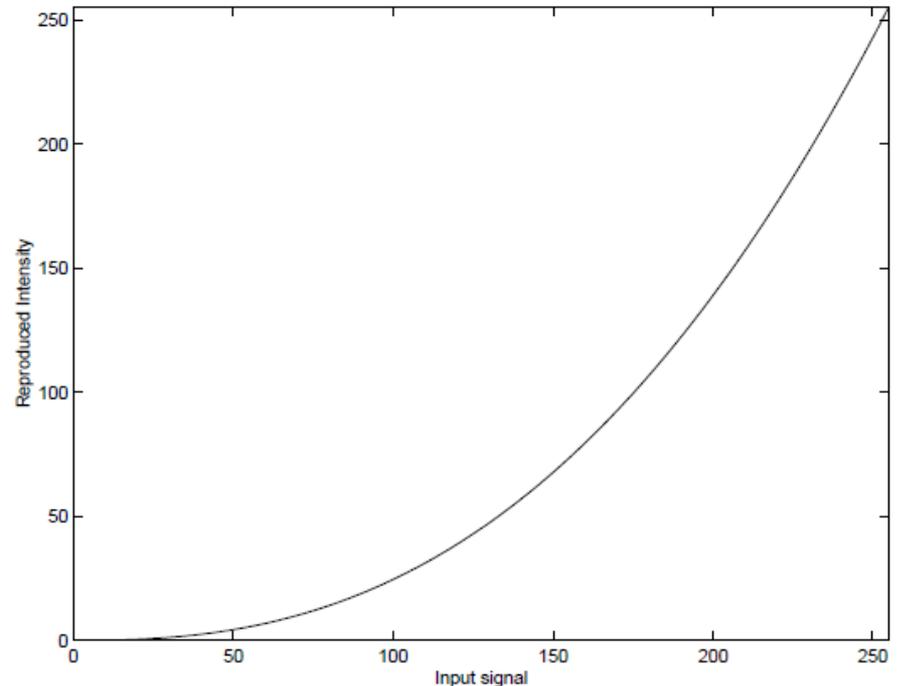
# Transformaciones Lineales de Brillo y Contraste



# Corrección Gamma

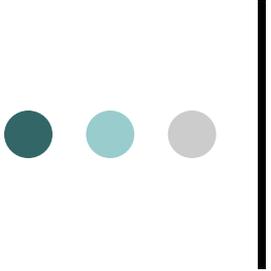
- Existen transformaciones no-lineales de la intensidad de los píxeles que afectan el brillo y contraste
- **Corrección Gamma:** compensa las propiedades asociadas a la percepción
- Optimiza la distribución de intensidades para una mejor percepción de la imagen

$$y = 255\left(\frac{x}{255}\right)^\gamma$$



# Corrección Gamma

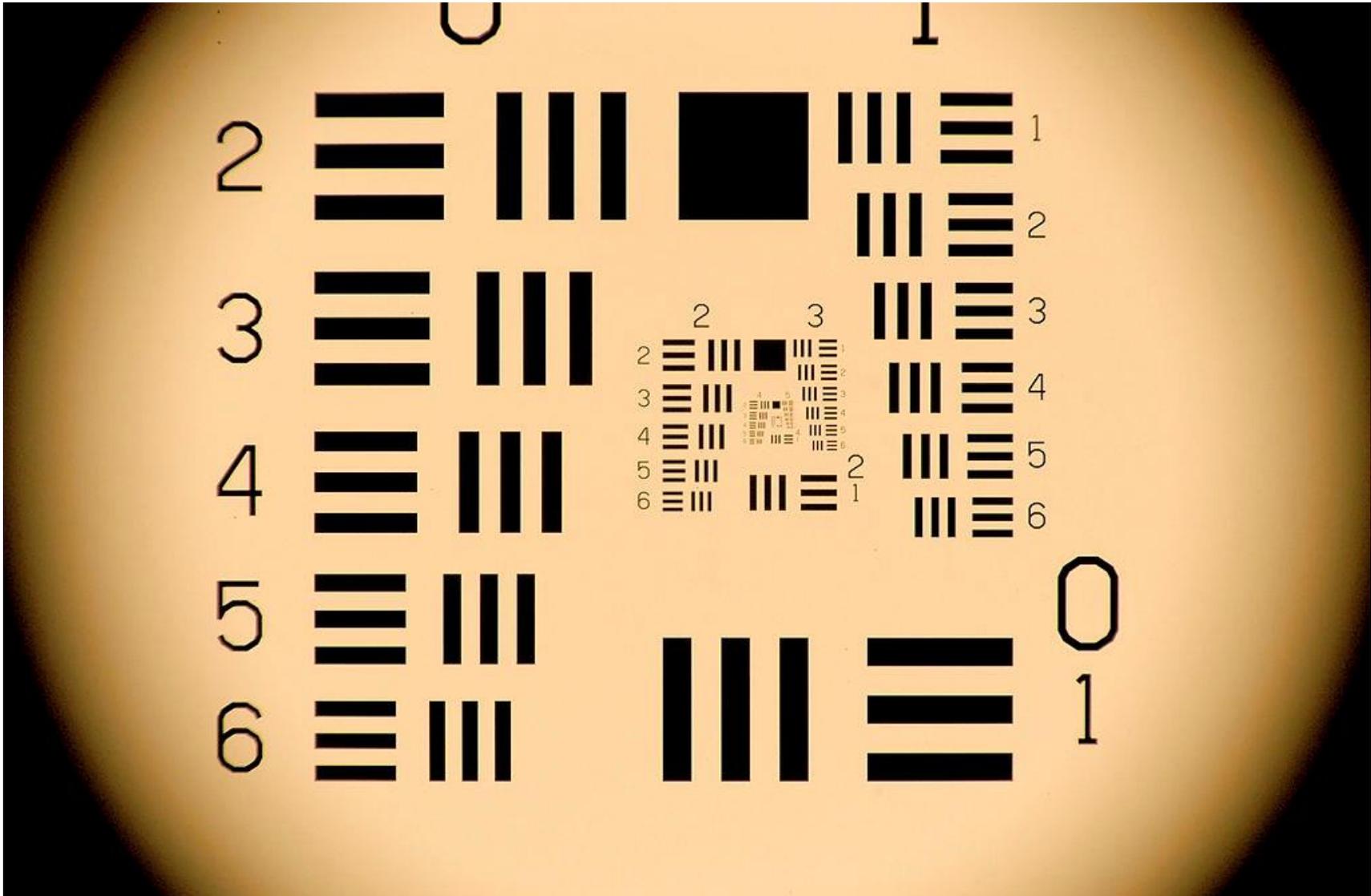




# Imágenes Monocromáticas

- De interés para este laboratorio
- Cada pixel representa la intensidad total en 1 byte de 8 bits
- El diferencial de brillo entre niveles (0.39%) es menor que la sensibilidad del ojo humano
- Mejoras en calidad se obtiene aumentando la **resolución**
  - más pixeles no necesariamente implica mejor resolución
  - lo importante es la cantidad de pixeles independientes
  - Se mide con tests para resolver espacialmente objetos
- Resoluciones típicas (pix count): 720×480 (DVD), 1024×768 (PC), 1920×1080 (HDTV), 2048×1536 (iPad 3), 10000×7000 (IMAX)

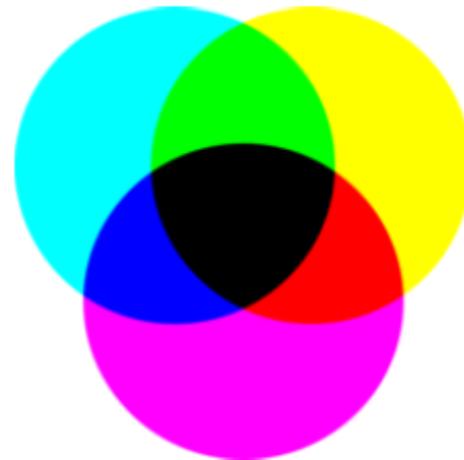
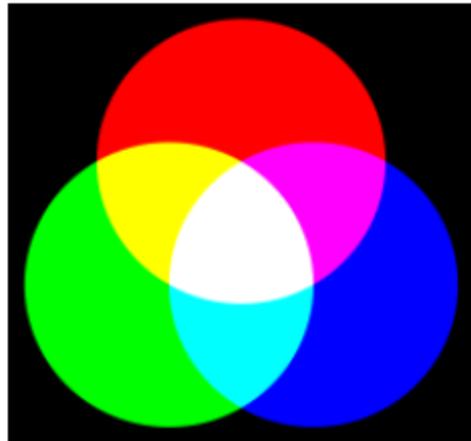
# Test de resolución espacial



# Imágenes en Color

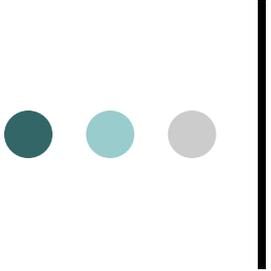
- Imágenes usan un byte de 8 bits para representar cada base que permitan descomponer colores, por ejemplo:
  - RGB: Rojo, verde, azul (3 bytes de 8 bits = 24 bits).  
Compone colores en forma aditiva
  - CMYK: Cyan, magenta, amarillo, negro (4 bytes de 8 bits = 32 bits).  
Compone colores en forma sustractiva

RGB→



←CMYK





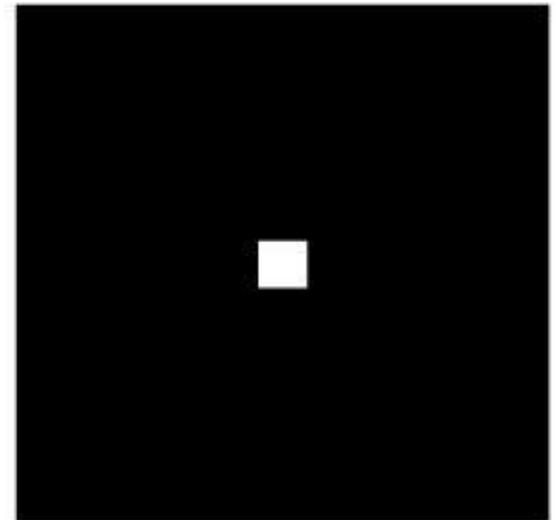
# Filtrado Digital de Imágenes

- Aplicaciones:
  - Smoothing (pasa-bajos), Sharpening (pasa-altos)
  - Reducción de ruido, detección de bordes
  - Corrección de iluminación, desenfoque, movimiento
- Filtros involucran ambas direcciones (x,y)
  - Respuesta a impulso de un filtro: point spread function (PSF)
  - Bordes pueden ser libres (0) o tomar valores de otros extremos
- Filtrado se realiza en frecuencia por su eficiencia

# Impulso (delta de Dirac) en 2D

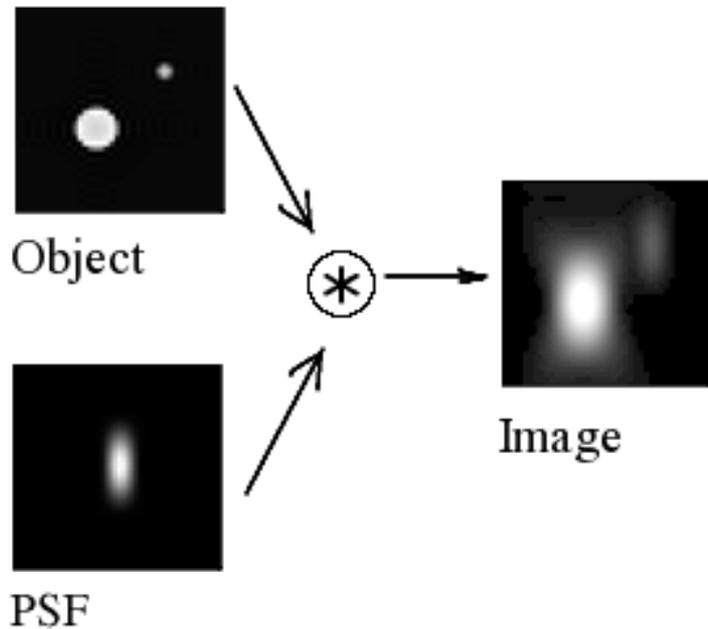
255 ×

5	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
-1	0	0	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	0	0
-4	0	0	0	0	0	0	0	0	0	0	0
-5	0	0	0	0	0	0	0	0	0	0	0
	-5	-4	-3	-2	-1	0	1	2	3	4	5



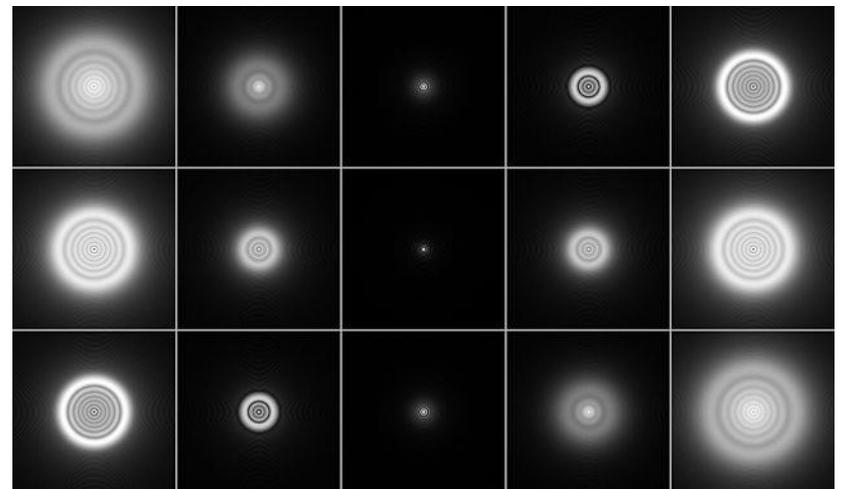
Impulso Dirac  
 $h(i, j) = \delta(i)\delta(j)$

Respuesta a impulso en 2D:  $h(i, j)$   
Función de dispersión (PSF) de un filtro

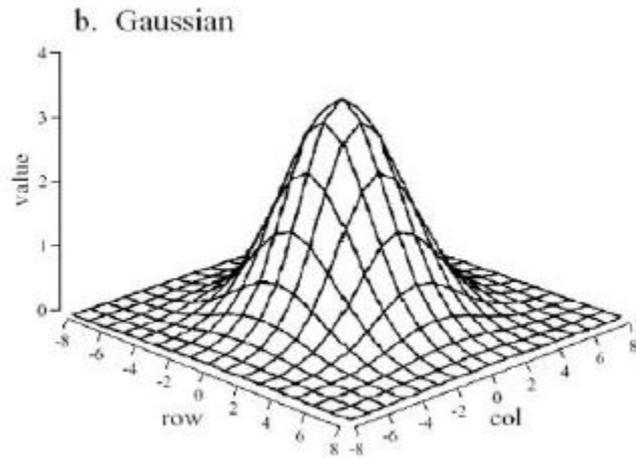


Efecto de una PSF en  
una imagen

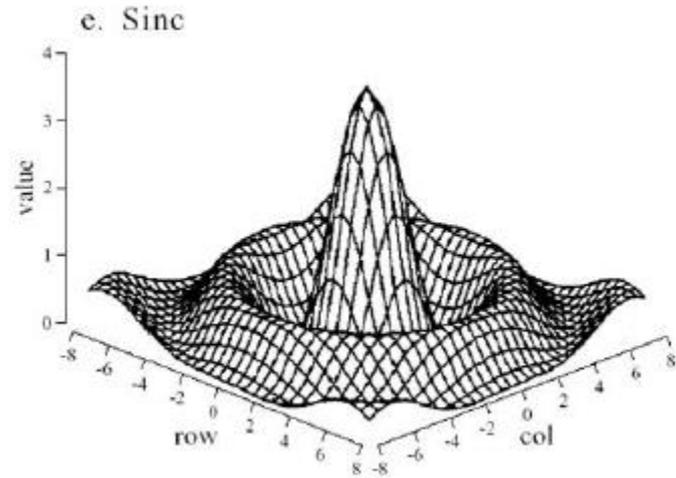
Ejemplos de PSF  
típicas



# Ejemplos de PSF de filtros graficadas en mesh

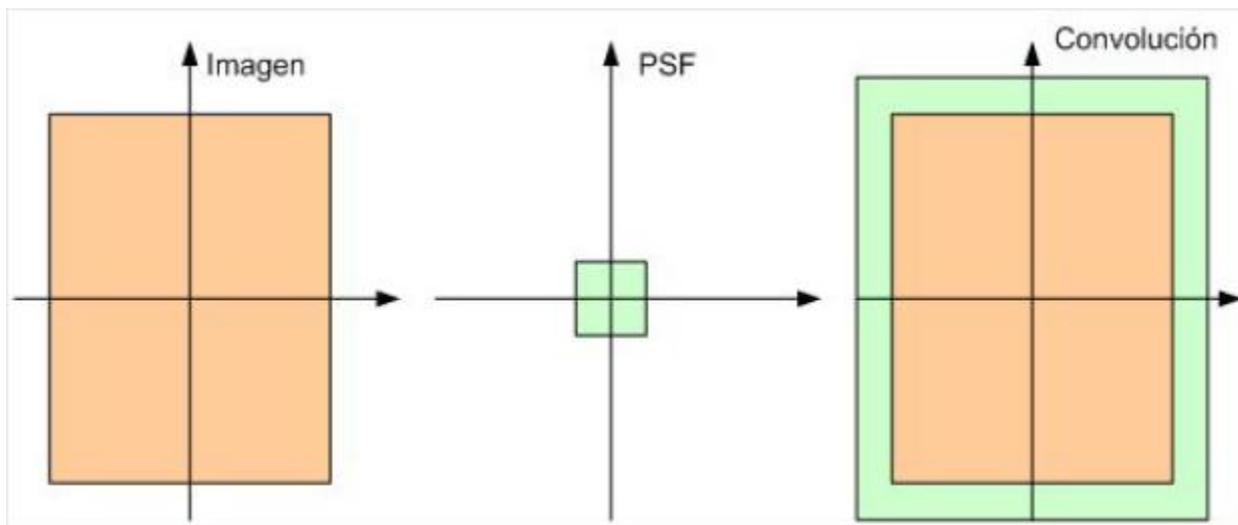


PSF Gaussiana



PSF Sinc

Respuesta a impulso en 2D:  $h(i, j)$   
Función de dispersión (PSF) de un filtro



$$M \times M$$

$$N \times N$$

$$M+N-1 \times M+N-1$$

Se descartan pixeles que están fuera  
de la frontera de la imagen original

## Ecuaciones de diferencia en 2D

- Filtros lineales FIR

$$y(m, n) = \sum_{k=-N}^N \sum_{l=-N}^N h(k, l)x(m - k, n - l)$$

- Filtros lineales IIR

$$\begin{aligned} y(m, n) = & \sum_{k=-N}^N \sum_{l=-N}^N b(k, l)x(m - k, n - l) \\ & + \sum_{k=-P}^P \sum_{l=1}^P a(k, l)y(m - k, n - l) \\ & + \sum_{k=1}^P a(k, 0)y(m - k, n) \end{aligned}$$

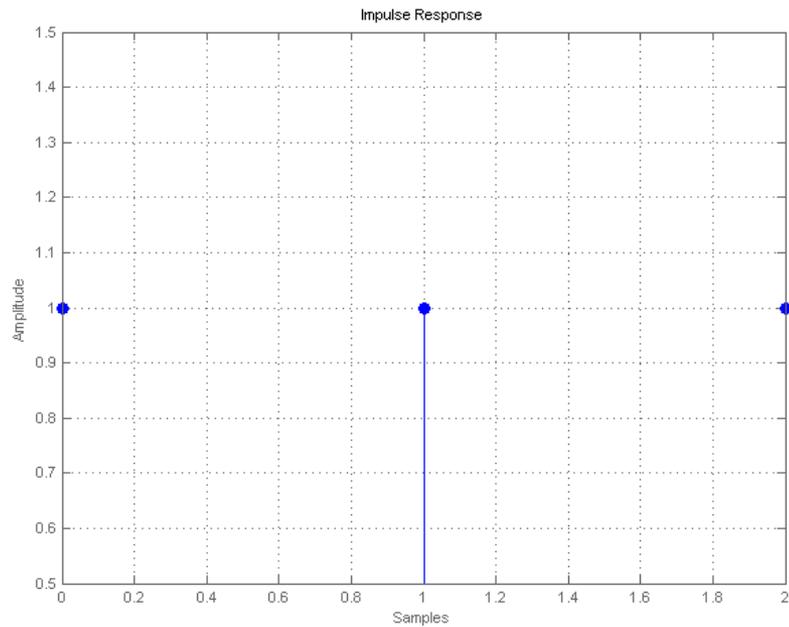
## Ecuaciones de diferencia en 2D

- Los filtros lineales en los cuales una PSD está bien definida se pueden implementar como una convolución en 2D (denotado **\*\***)
- En la práctica esto se realiza sólo para filtros FIR. Para filtros IIR se utiliza la ecuación de diferencia
- En MATLAB el comando **filter2** realiza esta convolución en dos dimensiones

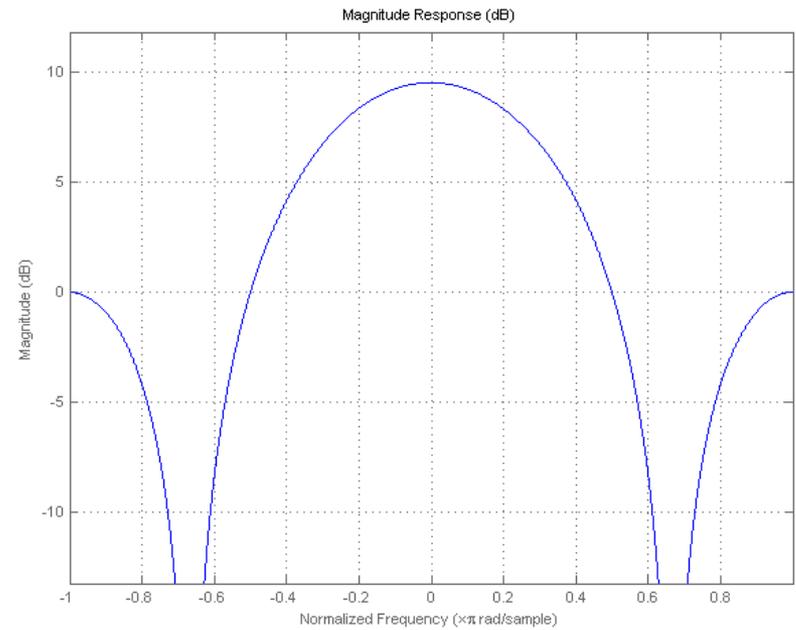
$$\begin{aligned}g(i, j) &= f(i, j) ** h(i, j) \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(k, l)h(i - k, j - l)\end{aligned}$$

# Filtros pasa-bajos en 1D

Respuesta a impulso



Magnitud en frecuencia



# Filtros pasa-bajos lineales

$$\frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

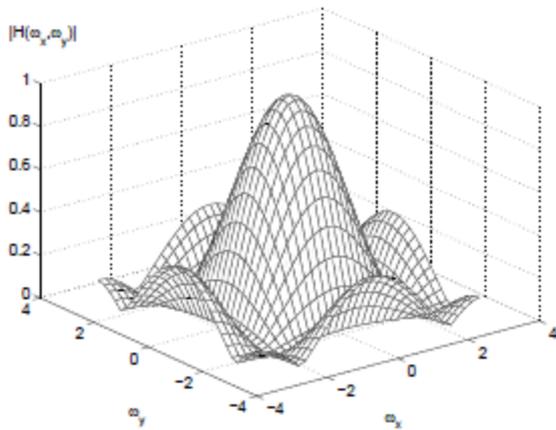
(a)

$$\frac{1}{10} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

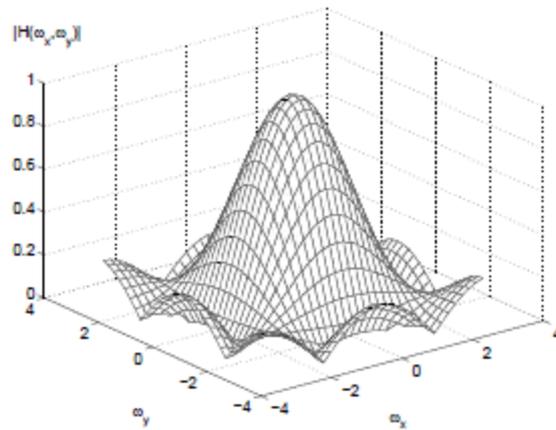
(b)

$$\frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

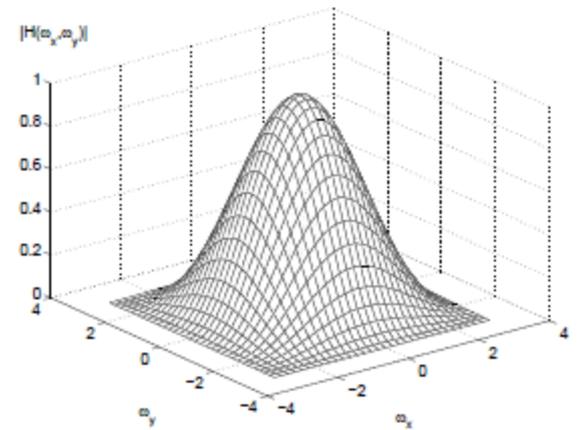
(c)



(a)



(b)



(c)

# Filtros pasa-bajos



(a)



(b)



(c)

## Filtros pasa-bajos no-lineales

- Un filtro no-lineal común es la mediana, que resulta de gran utilidad para eliminar ciertos tipos de ruidos

$$g(i, j) = \text{median}\{f(i - k, j - l)\}, (k, l) \in W$$



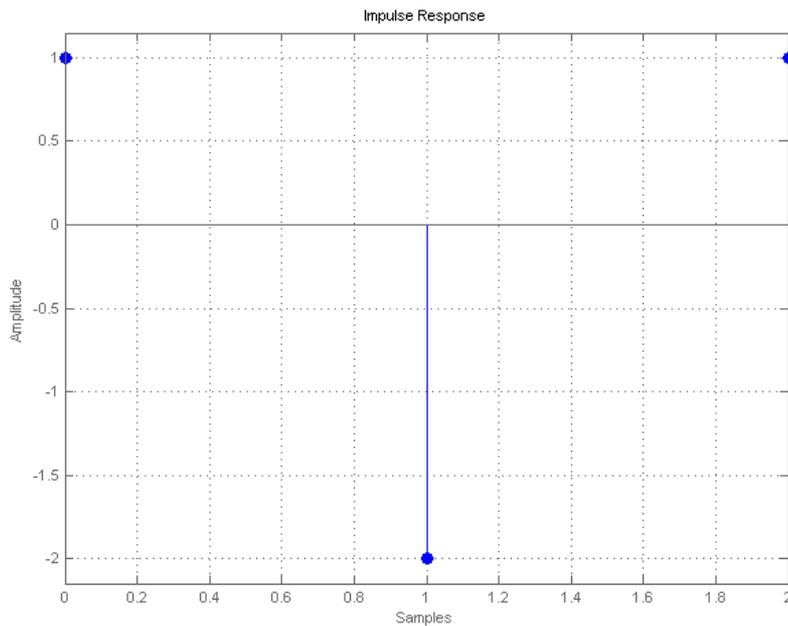
(a)



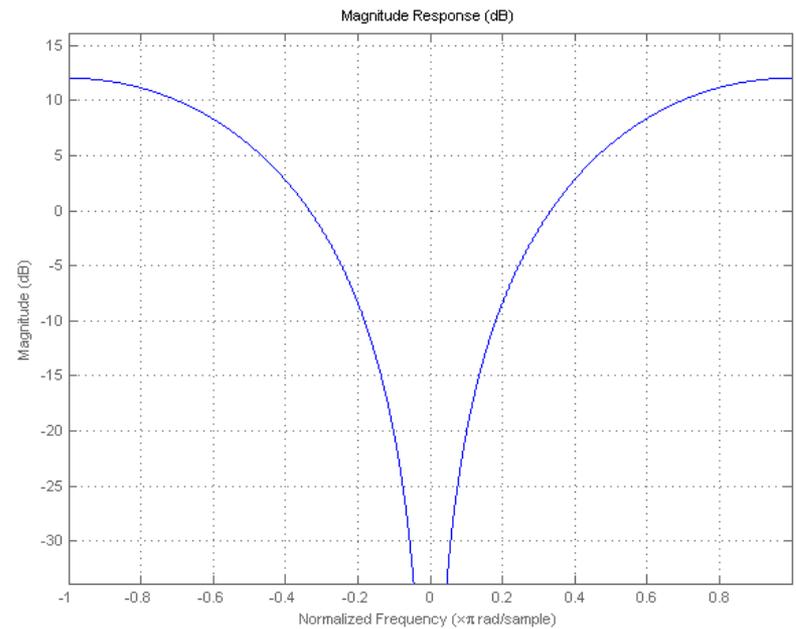
(b)

# Filtros pasa-altos en 1D

Respuesta a impulso



Magnitud en frecuencia



# Filtros pasa-altos lineales

0	1	0
1	-4	1
0	1	0

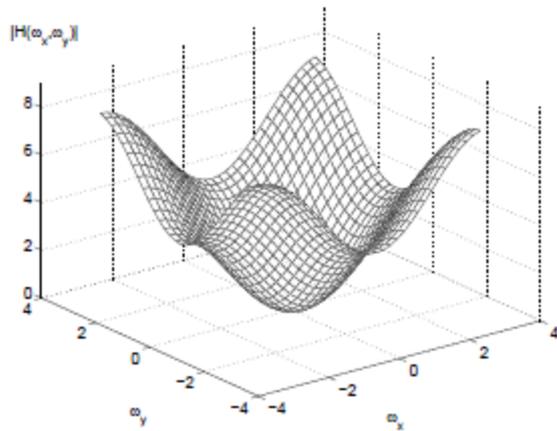
(a)

1	1	1
1	-8	1
1	1	1

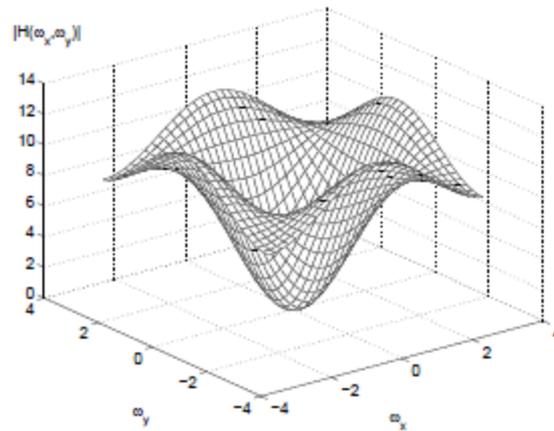
(b)

-1	2	-1
2	-4	2
-1	2	-1

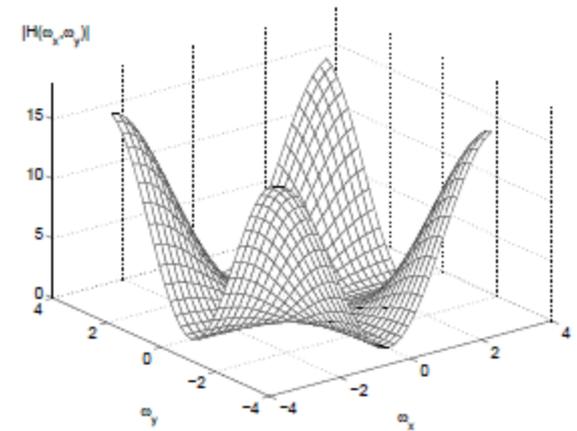
(c)



(a)



(b)



(c)

## Filtros pasa-altos lineales

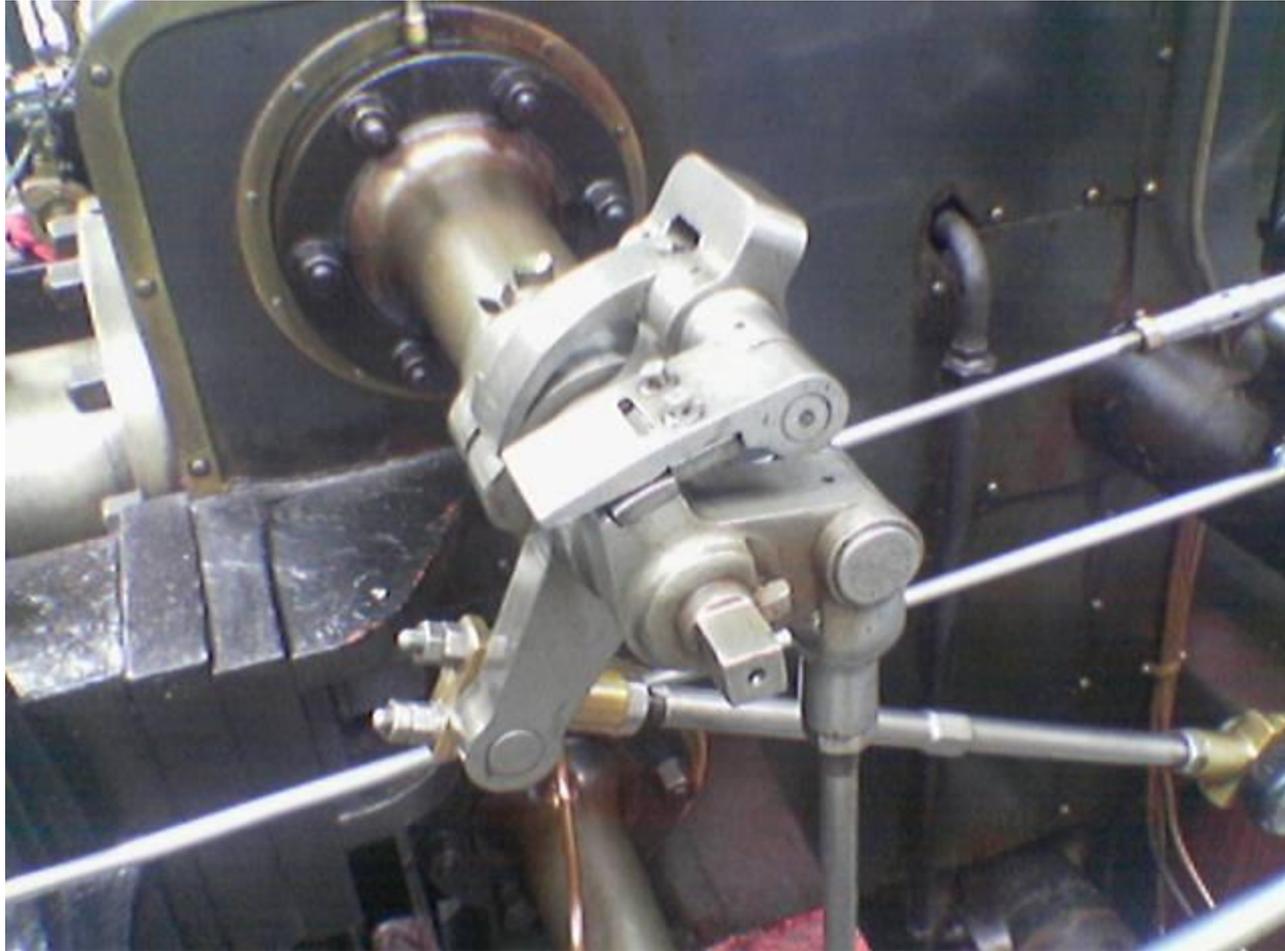


(a)

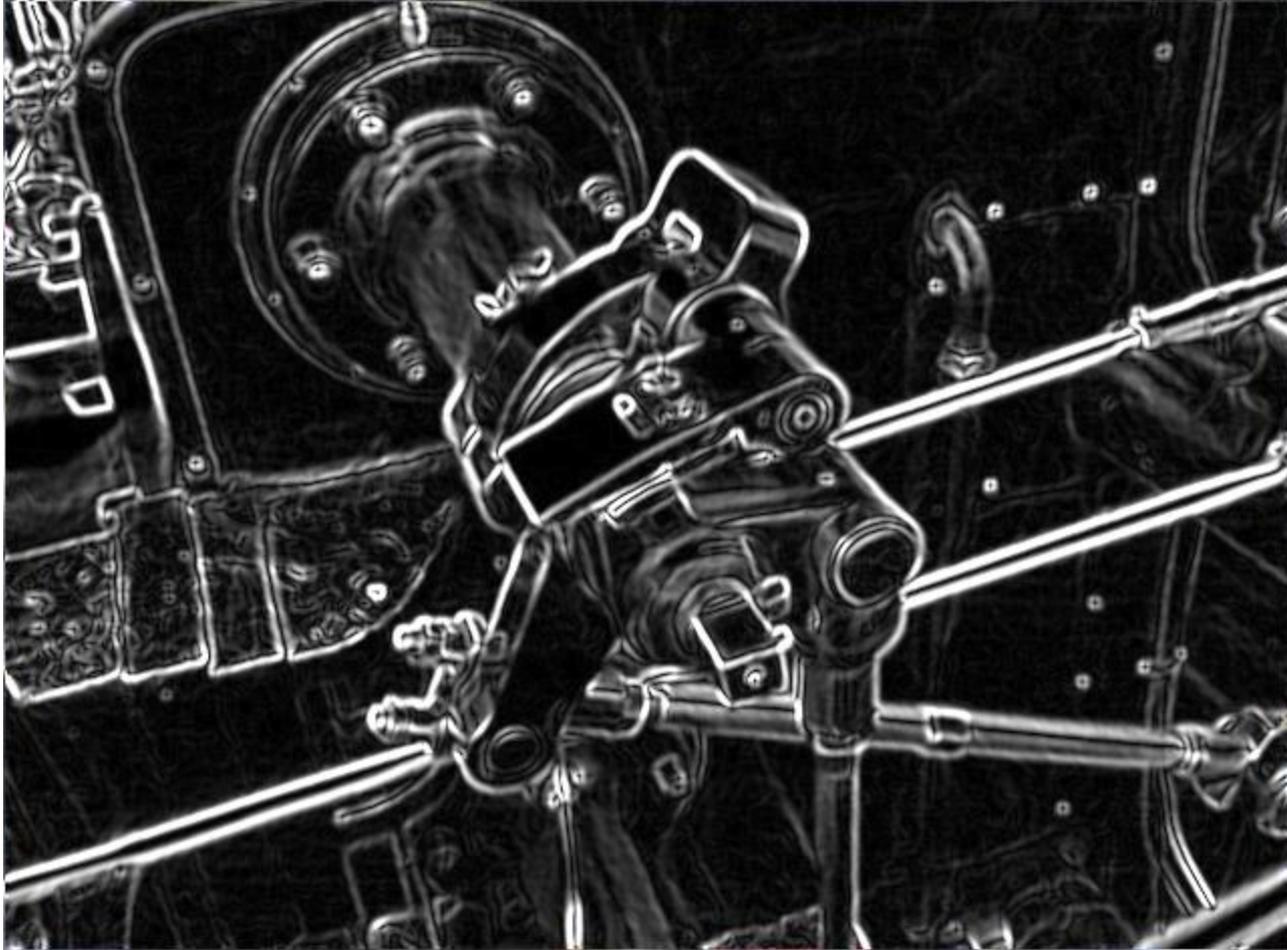


(b)

# Filtros pasa-altos como operadores de gradiente



# Filtros pasa-altos como operadores de gradiente



# Operador de gradiente Sobel

- Permite estimar magnitud y fase del gradiente
- Combina dos etapas de filtrado
- Usa kernels 3x3 transpuestos para filtrar basados en una combinación de filtros pasa-alto y pasa-bajo:

$$\text{Kernel 1: } \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} [1 \ 0 \ -1] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} [1 \ -1] * [1 \ 1]$$

$$\text{Kernel 2: } \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} [1 \ 2 \ 1] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} [1 \ 1] * [1 \ 1]$$

# Operador de gradiente Sobel

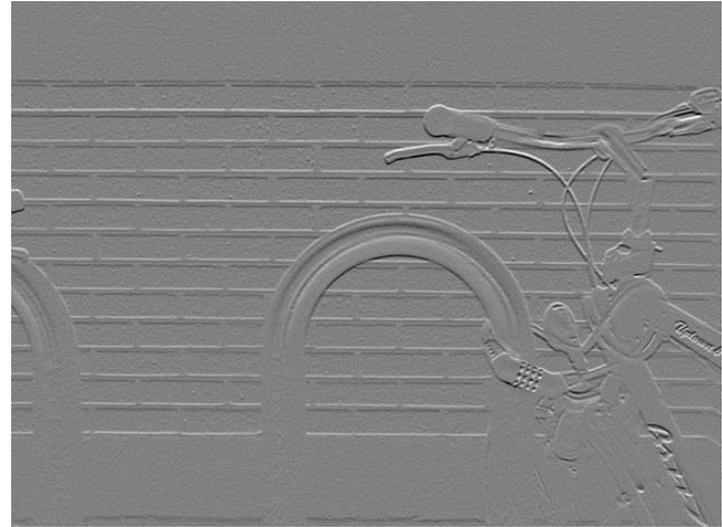
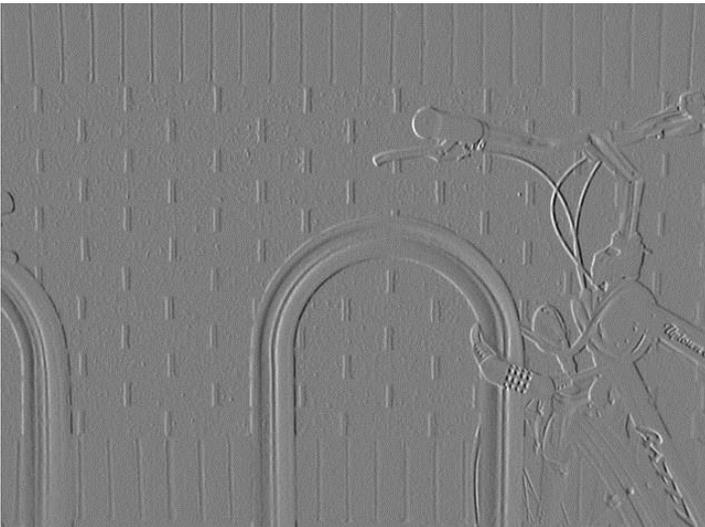
- La magnitud ( $\mathbf{G}$ ) y la fase ( $\Theta$ ) se obtienen a partir de salidas de cada filtro ( $\mathbf{G}_x$ ,  $\mathbf{G}_y$ )

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

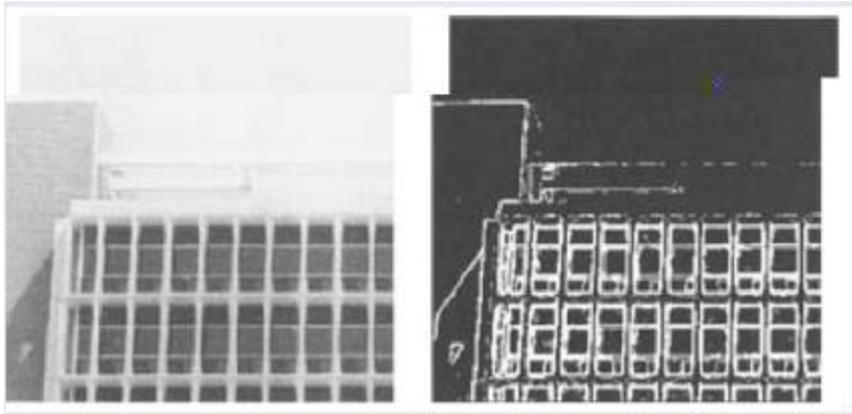
$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad \Theta = \text{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$

- El símbolo  $*$  representa convolución en 2D
- Generalmente se trabaja con la magnitud del gradiente y la imagen se normaliza entre  $[0,1]$

# Operador de gradiente Sobel

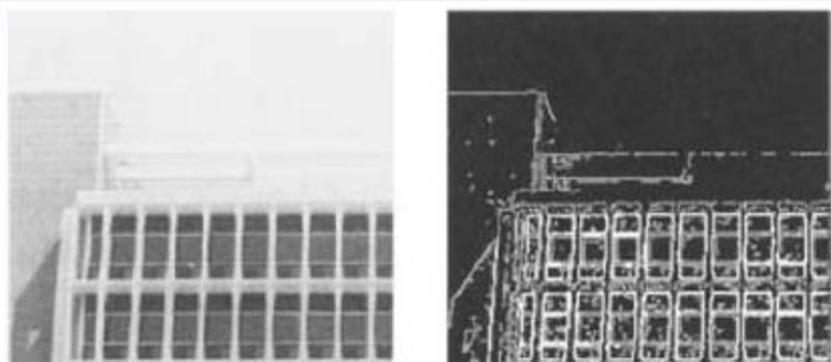


# Otros operadores de gradiente



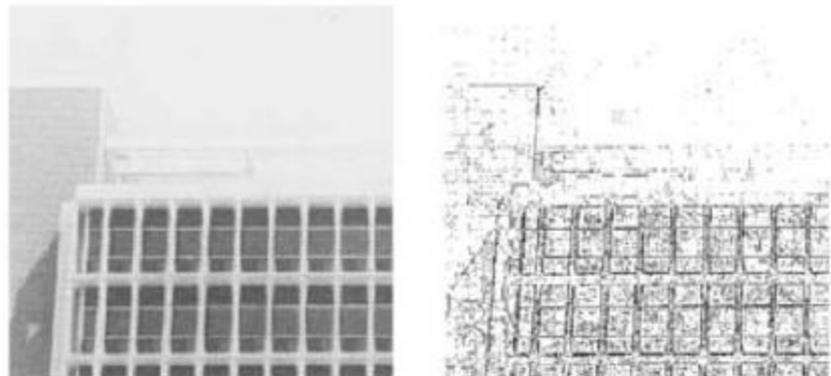
Sobel  $3 \times 3$ ,  
Kernel:

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



Prewitt  $3 \times 3$ ,  
Kernel:

1	1	1
0	0	0
-1	-1	-1



Laplaciano  $3 \times 3$ ,  
Kernel:

0	0	0
1	-2	1
0	0	0



## Transformada Z: Extensión a 2D

- 1-D Z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$
$$x(n) = \frac{1}{2\pi j} \int_{C \in ROC} X(z)z^{n-1}dz$$

- 2-D Z-transform

$$F(z_1, z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n)z_1^{-m}z_2^{-n}$$
$$f(m, n) = \frac{1}{(2\pi j)^2} \int_{C^2 \in ROC} F(z_1, z_2)z_1^{m-1}z_2^{n-1}dz_1dz_2$$

## DTFT: Extensión a 2D

- In 1-D, the DTFT is the 1-D Z-transform evaluated on the unit circle.

$$F(e^{j\omega}) = F(z)|_{z=e^{j\omega}}$$

- In 2-D the DSFT is the 2-D Z transform evaluated on the unit sphere.

$$F(e^{j\mu}, e^{j\nu}) = F(z_1, z_2)|_{z_1=e^{j\mu}, z_2=e^{j\nu}}$$

## Transformada Z y DTFT en 2D

- Filtros FIR

$$y(m, n) = \sum_{k=-N}^N \sum_{l=-N}^N h(k, l)x(m - k, n - l)$$

$$H(z_1, z_2) = \sum_{k=-N}^N \sum_{l=-N}^N h(k, l)z_1^{-k}z_2^{-l}$$

$$H(e^{j\mu}, e^{j\nu}) = \sum_{k=-N}^N \sum_{l=-N}^N h(k, l)e^{-j(k\mu + l\nu)}$$

# Transformada Z en 2D

- Filtros IIR

$$\begin{aligned}y(m, n) = & \sum_{k=-N}^N \sum_{l=-N}^N b(k, l)x(m - k, n - l) \\ & + \sum_{k=-P}^P \sum_{l=1}^P a(k, l)y(m - k, n - l) \\ & + \sum_{k=1}^P a(k, 0)y(m - k, n)\end{aligned}$$

$$\begin{aligned}H(z_1, z_2) = & \frac{\sum_{k=-N}^N \sum_{l=-N}^N b(k, l)z_1^{-k}z_2^{-l}}{1 - \sum_{k=-P}^P \sum_{l=1}^P a(k, l)z_1^{-k}z_2^{-l} - \sum_{k=1}^P a(k, 0)z_1^{-k}}\end{aligned}$$

$$\begin{aligned}H(e^{j\mu}, e^{j\nu}) = & \frac{\sum_{k=-N}^N \sum_{l=-N}^N b(k, l)e^{-j(k\mu+l\nu)}}{1 - \sum_{k=-P}^P \sum_{l=1}^P a(k, l)e^{-j(k\mu+l\nu)} - \sum_{k=1}^P a(k, 0)e^{-j(k\mu)}}\end{aligned}$$



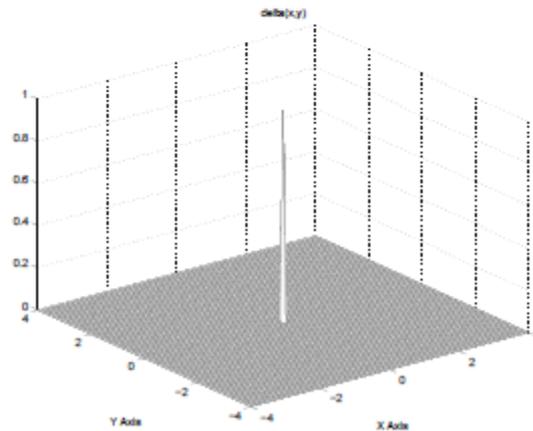
# Análisis Espectral de Imágenes



## DTFT de un impulso en 2D

- 2-D delta function:

$$\begin{aligned} CSFT \{ \delta(x, y) \} &= CSFT \{ \delta(x) \delta(y) \} \\ &= CTFT \{ \delta(x) \} \cdot CTFT \{ \delta(y) \} \\ &= 1 \cdot 1 = 1 \end{aligned}$$

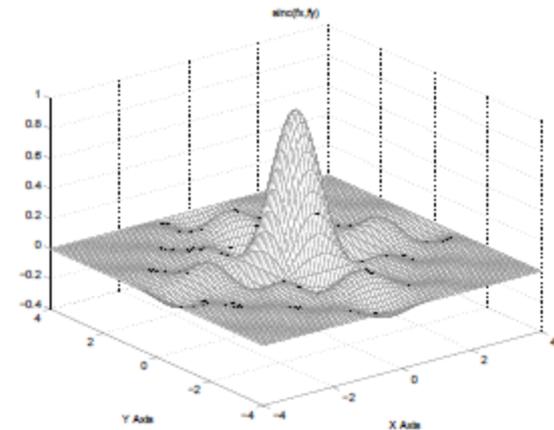
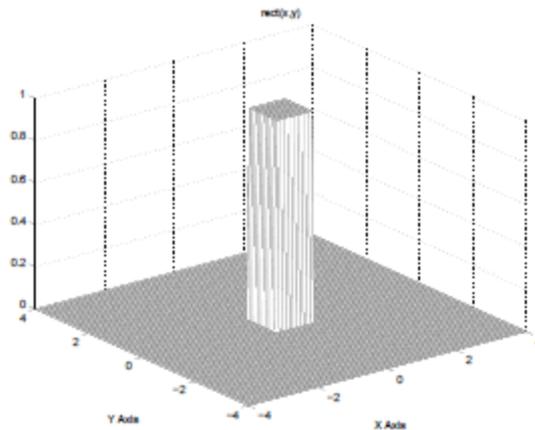


1

## DTFT de rect en 2D

- 2-D rect function:

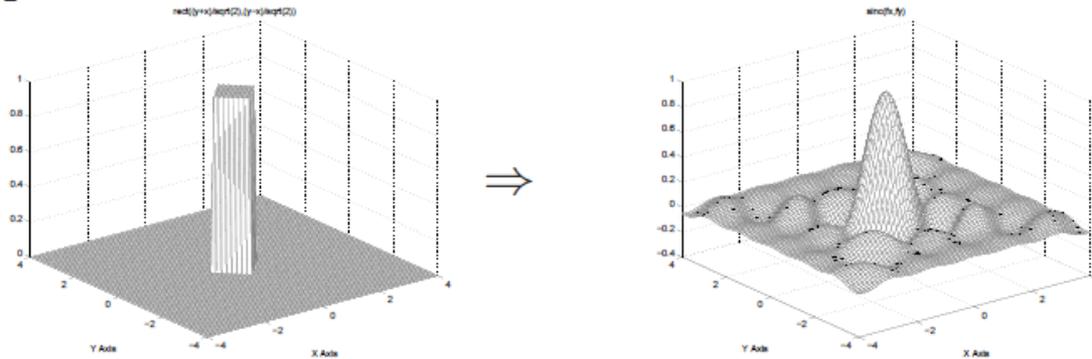
$$\begin{aligned} CSFT \{ \text{rect}(x, y) \} &= CSFT \{ \text{rect}(x) \text{rect}(y) \} \\ &= CTFT \{ \text{rect}(x) \} \cdot CTFT \{ \text{rect}(y) \} \\ &= \text{sinc}(u) \text{sinc}(v) \\ &= \text{sinc}(u, v) \end{aligned}$$



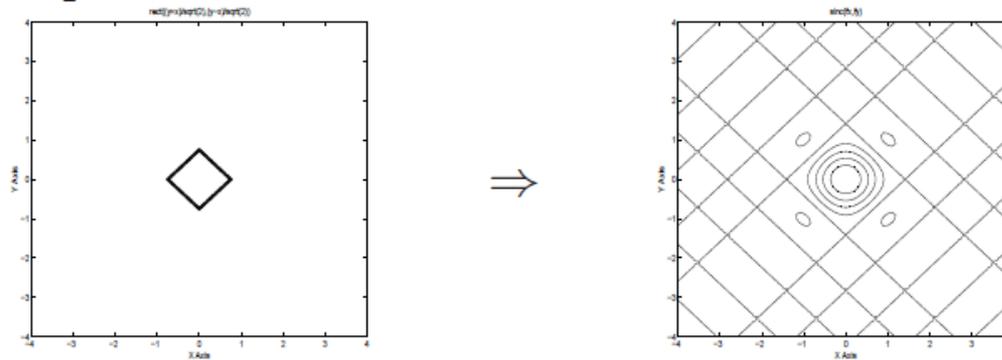
# DTFT de rect rotado en 2D

## Rotated 2-D Rect and Sinc Transform Pairs

- Mesh plot

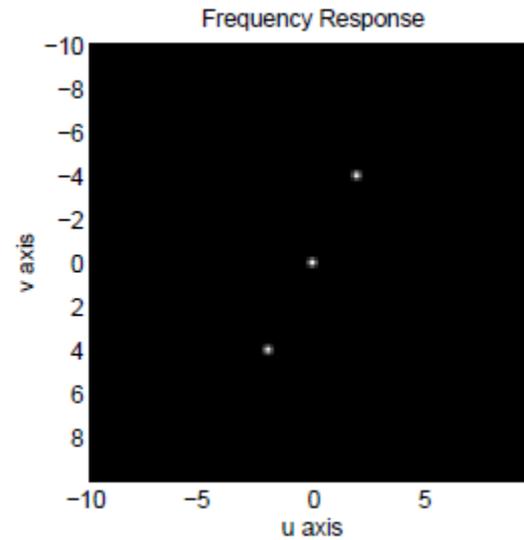
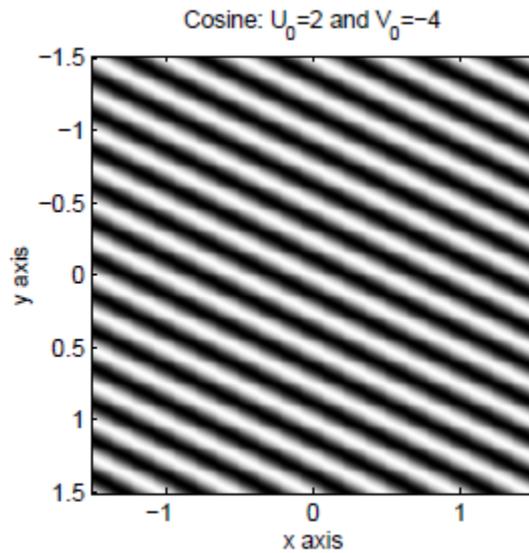


- Contour plot



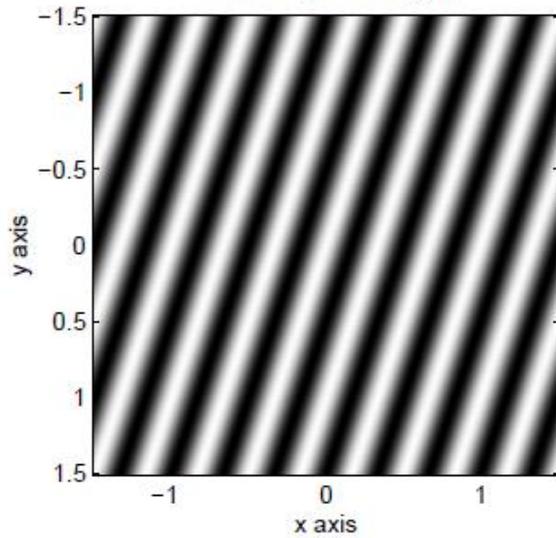
## DTFT de coseno en 2D

$$f(x, y) = \cos(u_0 x + v_0 y) + 0.5$$

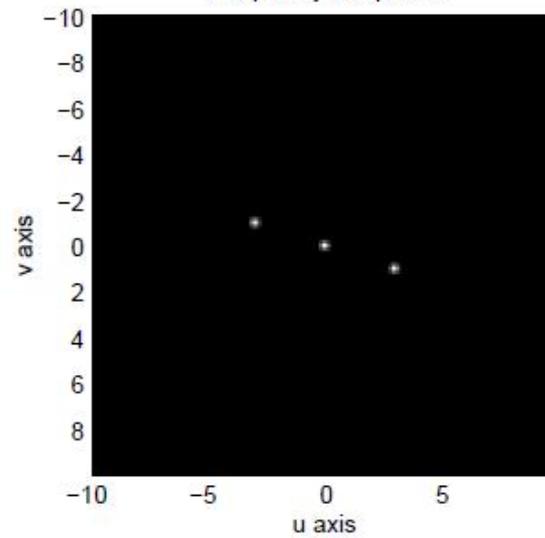


# DTFT de coseno en 2D

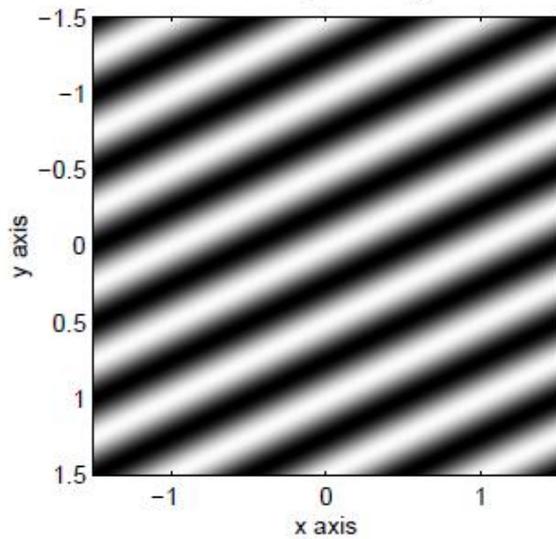
Cosine:  $U_0=3$  and  $V_0=1$



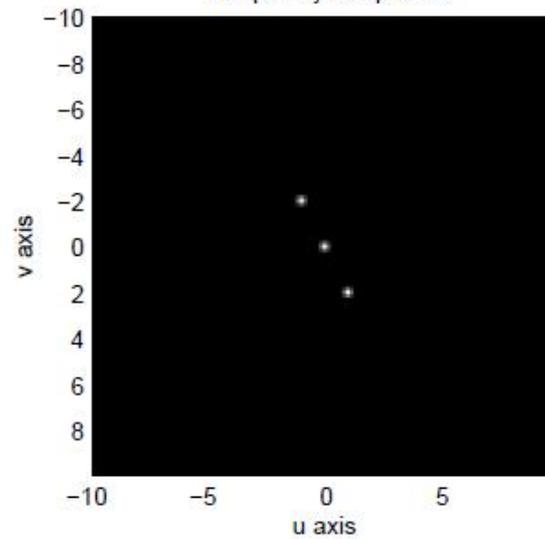
Frequency Response



Cosine:  $U_0=1$  and  $V_0=2$

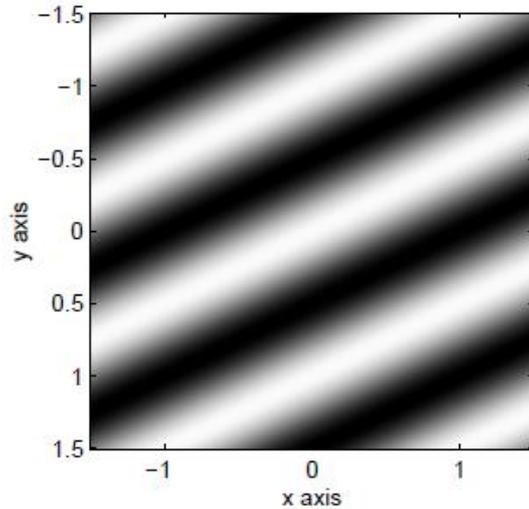


Frequency Response

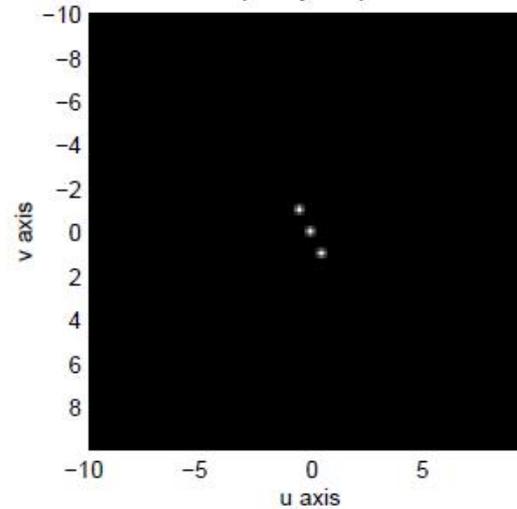


# DTFT de coseno en 2D

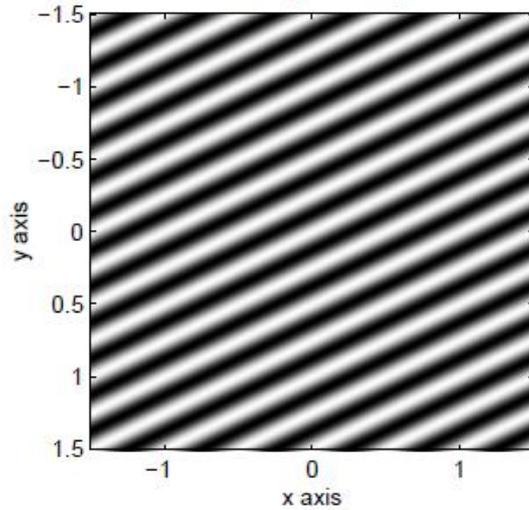
Cosine:  $U_0=0.5$  and  $V_0=1$



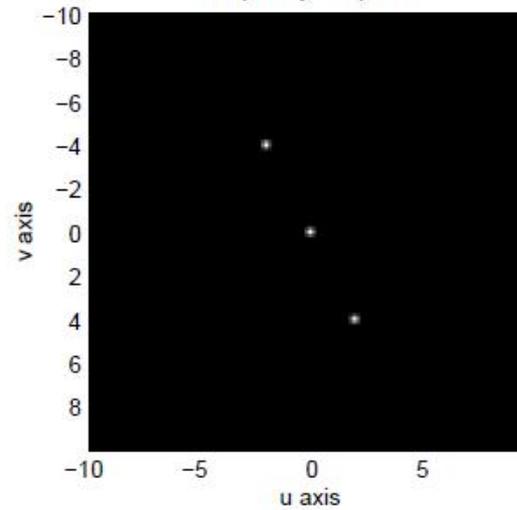
Frequency Response



Cosine:  $U_0=2$  and  $V_0=4$



Frequency Response



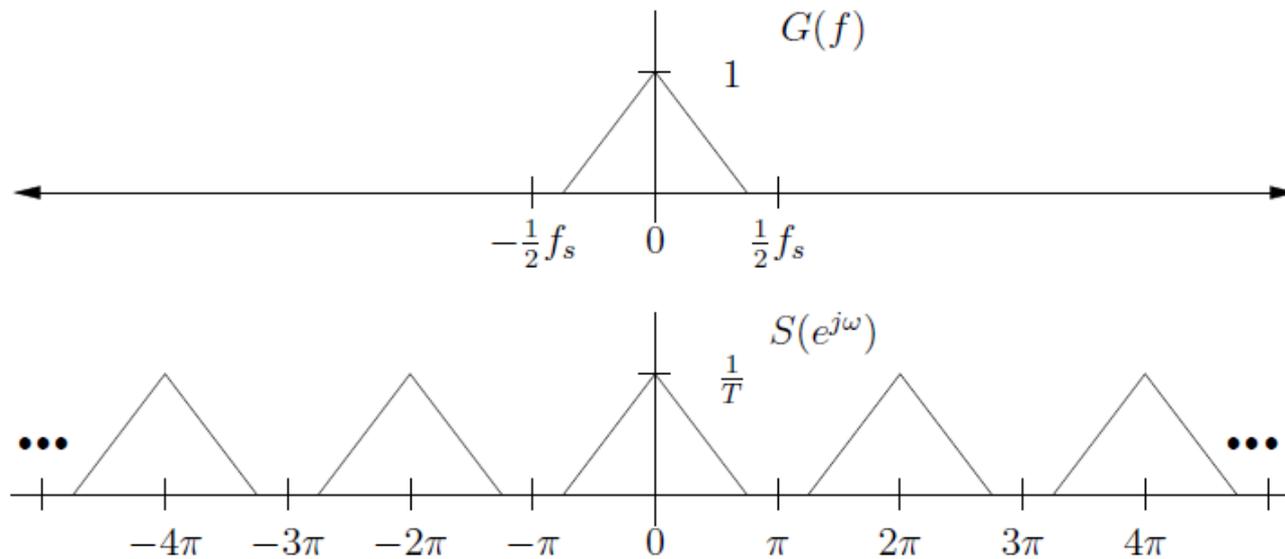


# Muestreo y Aliasing en Imágenes



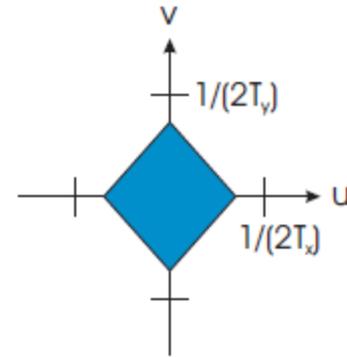
# Muestreo y periodicidad de la DTFT en 1D

## 1-D Sampling

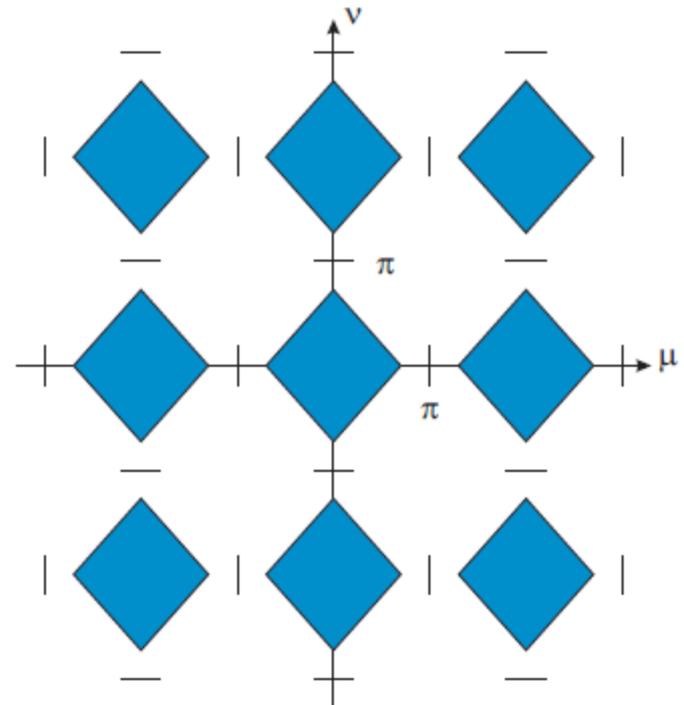


Muestreo SIN  
 efectos de aliasing y  
 SIN distorsión

$G(u, v)$  - Spectrum of continuous space image.

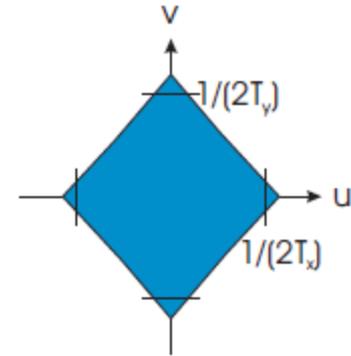


$S(e^{j\mu}, e^{j\nu})$  - Spectrum of sampled image.

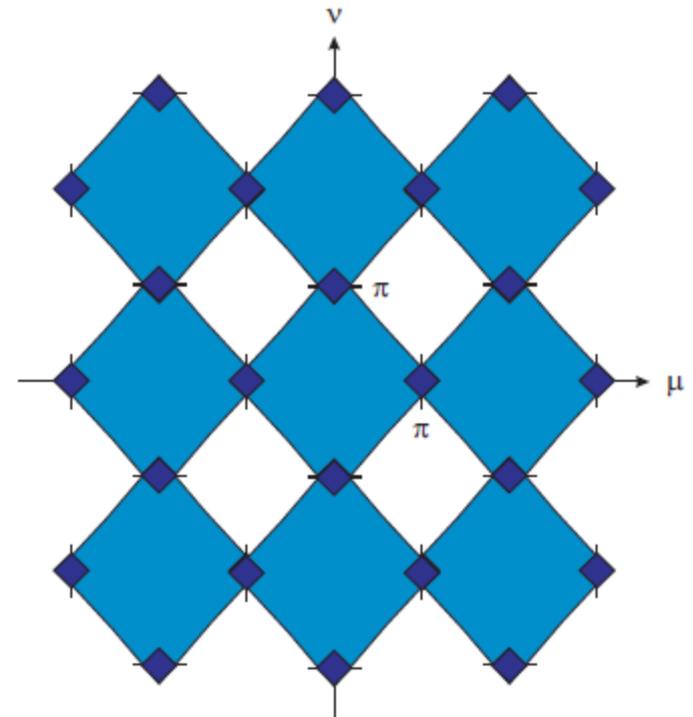


Muestreo CON  
efectos de aliasing y  
CON distorsión

$G(u, v)$  - Spectrum of continuous space image.

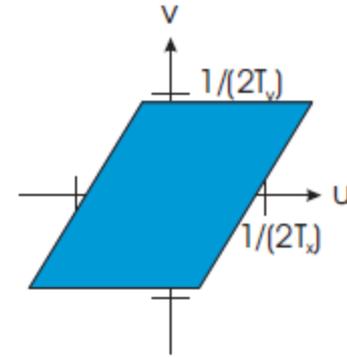


$S(e^{j\mu}, e^{j\nu})$  - Spectrum of sampled image.

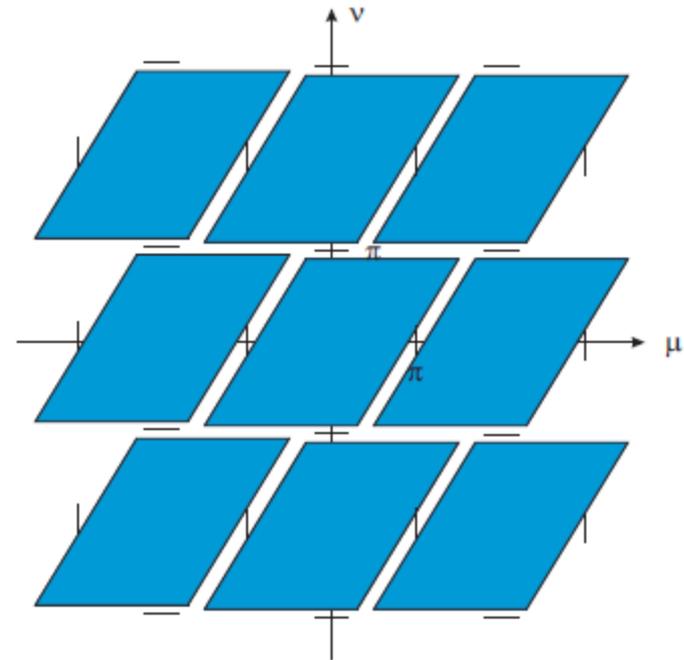


Muestreo CON  
efectos de aliasing y  
SIN distorsión

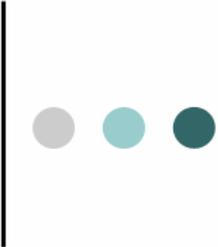
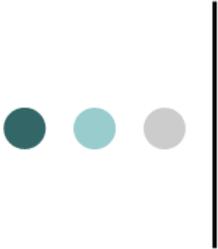
$G(u, v)$  - Spectrum of continuous space image.



$S(e^{j\mu}, e^{j\nu})$  - Spectrum of sampled image.



ELO 313 –Procesamiento Digital de Señales con Aplicaciones



# Conceptos Básicos para el Procesamiento Digital de Imágenes

**Matías Zañartu, Ph.D.**

Departamento de Electrónica

Universidad Técnica Federico Santa María