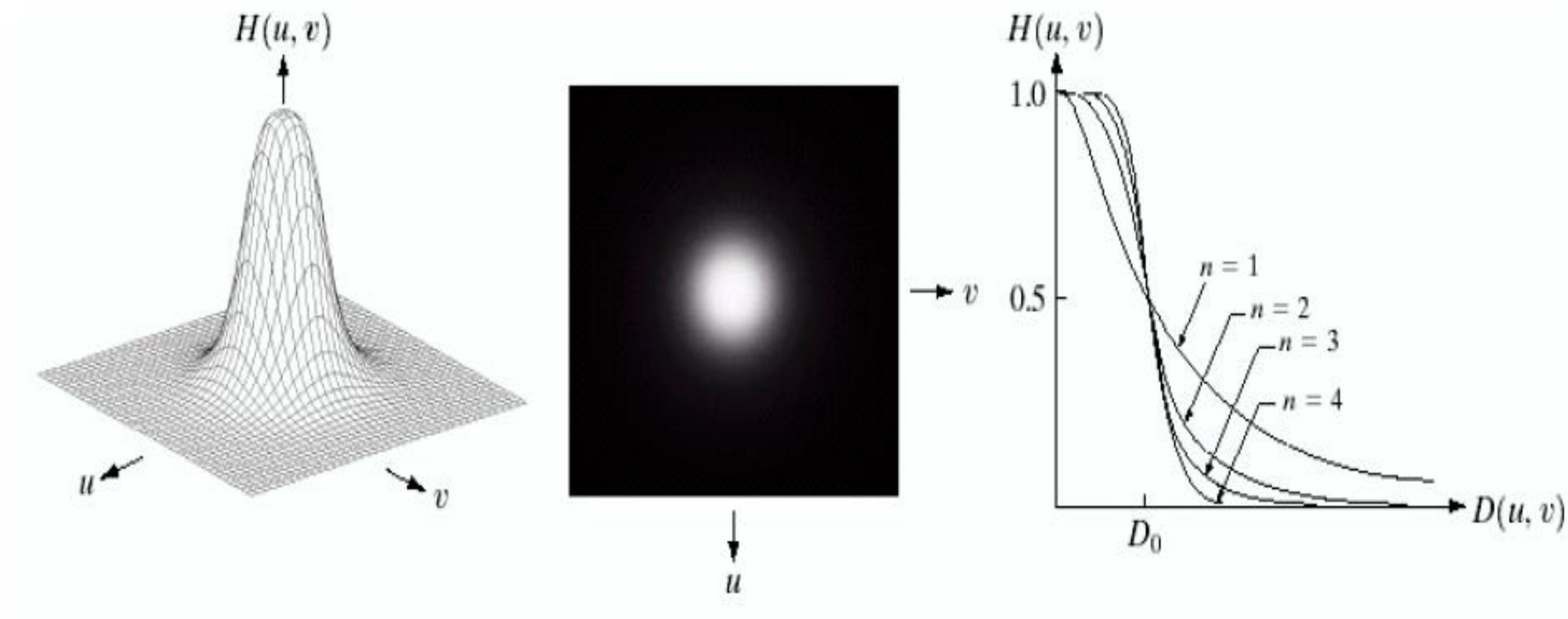
Procesamiento digital de Imágenes

Filtros



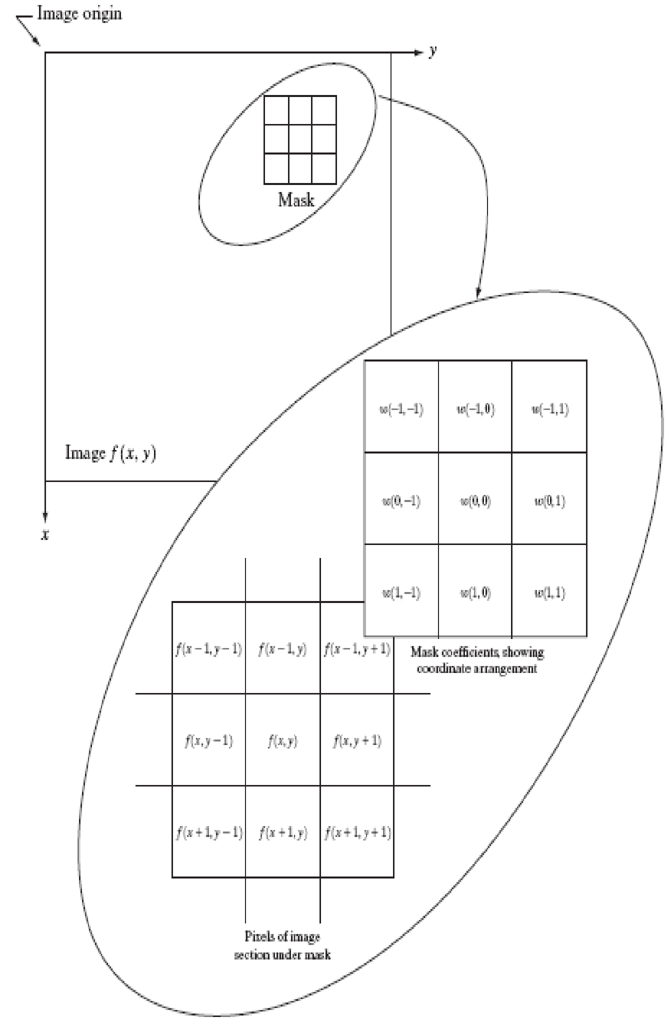
Filtros

Filtrado en el dominio espacial

$$
\begin{aligned}
R = {} & w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \cdots \\
& + w(0,0)f(x, y) + \cdots + w(1,0)f(x+1, y) + w(1,1)f(x+1, y+1),
\end{aligned}
$$

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

$\longleftarrow$ Filter
Mask
Kernel



Filtros

Filtrado en el dominio espacial

Una imagen f (x,y) de dimensiones MxN se puede filtrar con una mascara de dimensiones mxn aplicando la expresión:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$a = (m-1)/2$$

$$b = (n-1)/2$$

Mascara de convolucion

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

Para cubrir toda la imagen esta ecuación debe aplicarse para x=0….M-1 y=0….N-1

Esto no es otra cosa que la convolucion

Filtros

Filtrado en el dominio espacial

Las mascaras se pueden representar de diferentes formas por ejemplo para el caso de una mascara de 3x3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = w_1 z_1 + w_2 z_2 + \ldots w_9 z_9 = \sum_{i=1}^{9} w_i z_i .$$

Para cubrir toda la imagen esta ecuación debe aplicarse para x=0….M-1 y=0….N-1

Esto no es otra cosa que la convolucion

Filtros

Filtros Pasa Bajos

Estos filtros son usados para reducir el ruido en imágenes así como para eliminar pequeños detalles como paso previo al procesamiento de la imagen para extraer objetos de la misma

La salida de estos filtros es simplemente el promedio de los píxeles presentes en el vecindario de la mascara (filtros promediadores).

Como resultado se reducen las transiciones abruptas en los niveles de gris.

Dado que el ruido aleatorio tipicamente consiste en transiciones abruptas queda claro entonces que los filtros PB nos sirven para reducir el ruido.

Estos filtros tienen la desventaja de reducir los detalles que nos pueden interesar de ciertos objetos de la imagen

Filtros

Filtros Pasa Bajos

La primera mascara representa un filtro promediador (todos los coeficientes iguales. La segunda representa un filtro de promedios ponderados de diferente peso que decrece in forma inversa con la distancia.

$$
\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}
\qquad
\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}
$$

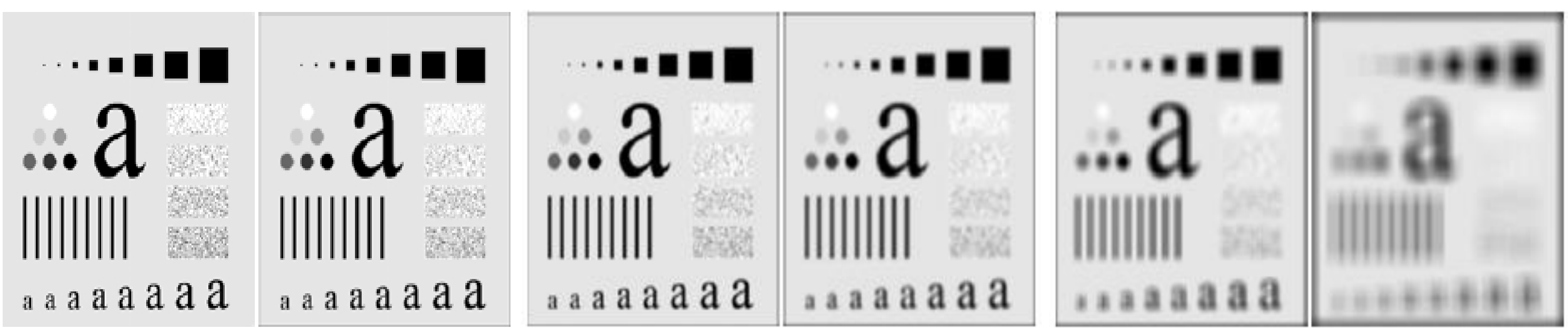
Filtrado en general:

$$
g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}
$$

Filtros

Filtros Pasa Bajos Ejemplos

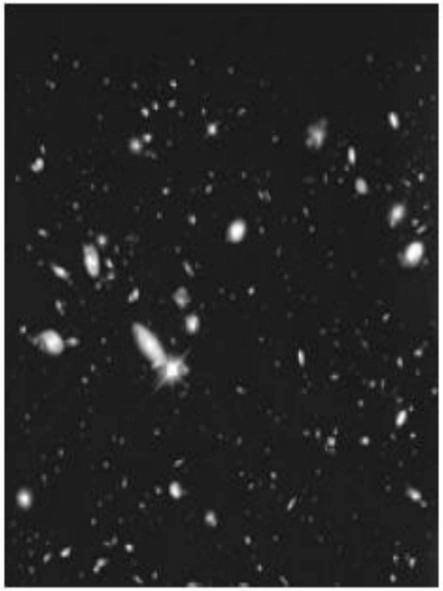
Imagen original 500 x 500 píxeles. Mascaras cuadradas n=3,5,9,15,25,35,45 y 55 píxeles. Los cuadrados en la parte superior son de tamaño 3,5,9,15,25,35,45 y 55 píxeles y sus bordes están separados en 25 píxeles. Las letras en la parte inferior van de 10 a 24 píxeles en incrementos de a dos. La letra grande es de 60 píxeles. Las barras verticales tienen 5 píxeles de ancho por 100 de alto con una separación de 20. El diámetro de los círculos es de 25 píxeles y sus bordes están separados en 15 píxeles. Su nivel de gris va del 0% al 100% negro en incrementos de a 20%. El color de fondo contiene un 10% de negro. Los rectángulos con ruido son de tamaño 50 x 120 píxeles.



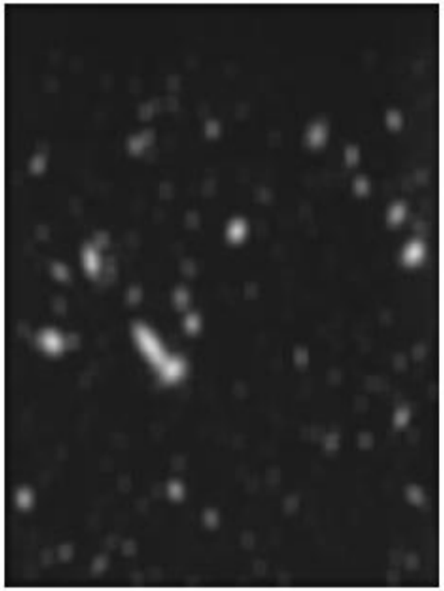
Filtros

Filtros Pasa Bajos Ejemplos

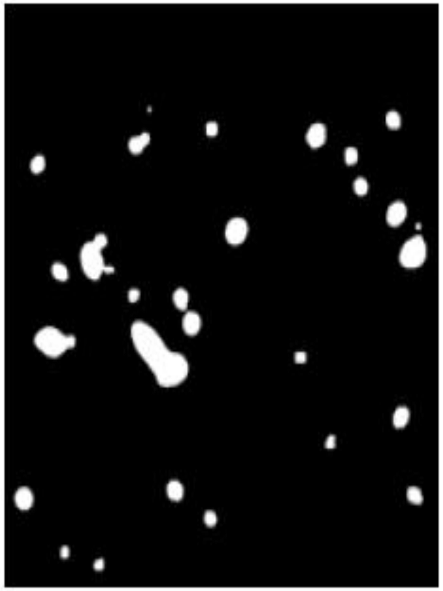
Procesamiento de una imagen para obtener los objetos mas representativos



Original



Low Pass 15x15



Threshold 25%

Filtros

Filtros No Lineales

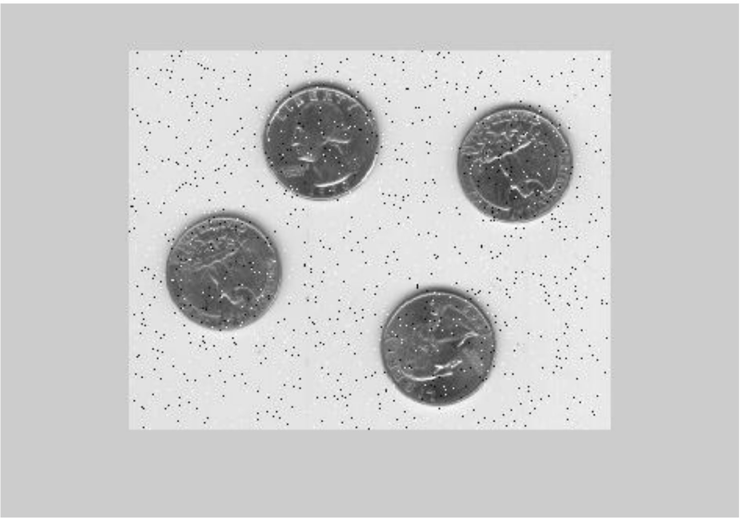
Filtro mediana

Se reemplaza el píxel original por la mediana formada por el y los del vecindario. Esto produce mejor resultado que un pasa bajos (menos blur). Este filtro fuerza a que píxeles con niveles de gris diferentes a los demás se asemejen mas a sus vecinos. Ejemplo:

```
I = imread('eight.tif');
imshow(I)                          % original
J = imnoise(I,'salt & pepper',0.02);
figure, imshow(J)                  %original  + ruido
K = filter2(fspecial('average',3),J)/255;
figure, imshow(K)                  %Filtro pasabajos
L = medfilt2(J,[3 3]);
figure, imshow(L)                  %Filtro mediana
```

Filtros

Filtros No Lineales

Filtros

Filtros pasa altos

(Sharpening spatial Filters)

Low-pass → Integrador

Hi-pass → Diferenciador

Enfatiza bordes y ruido

Bases:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

Filtros

Filtros pasa altos
(Sharpening spatial Filters)

Low-pass → Integrador

Hi-pass → Diferenciador

Enfatiza bordes y ruido

Bases:

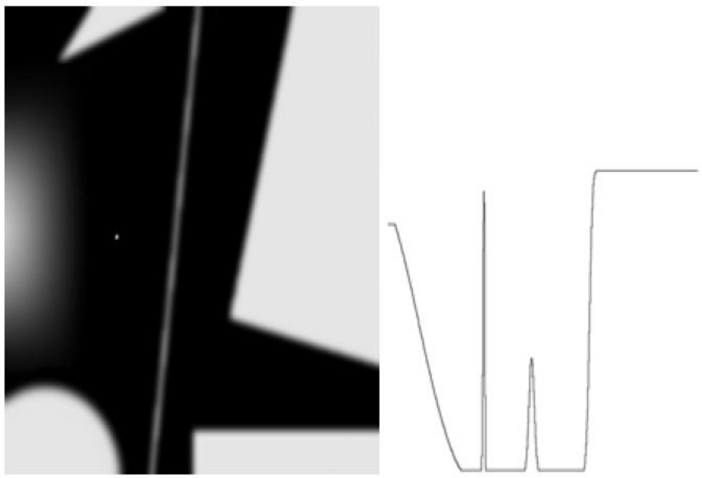
$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

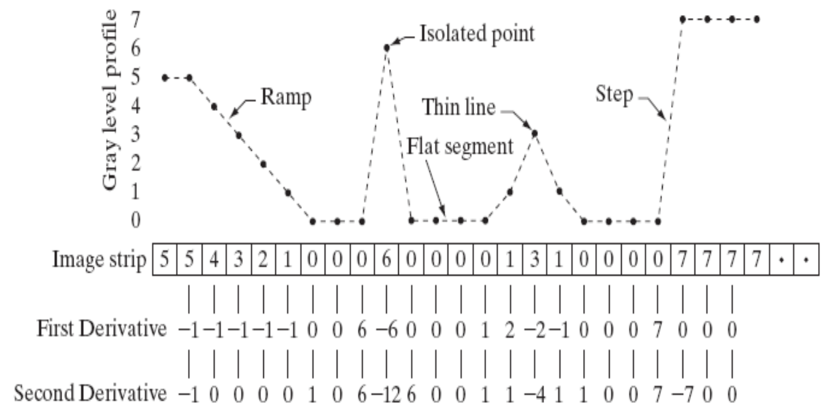
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

-1	1

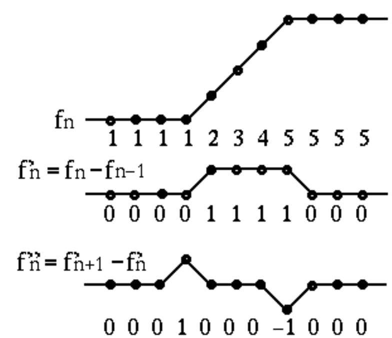
Filtros

Filtros pasa altos
(Sharpening spatial Filters)





Segunda derivada mas agresiva



Filtros

Filtros pasa altos
Laplaciano

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\right] - 4f(x, y).$$

0	1	0
1	−4	1
0	1	0

Filtros

Filtros pasa altos
Laplaciano

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\right] - 4f(x, y).$$

1	1	1
1	−8	1
1	1	1

Incluye diagonales

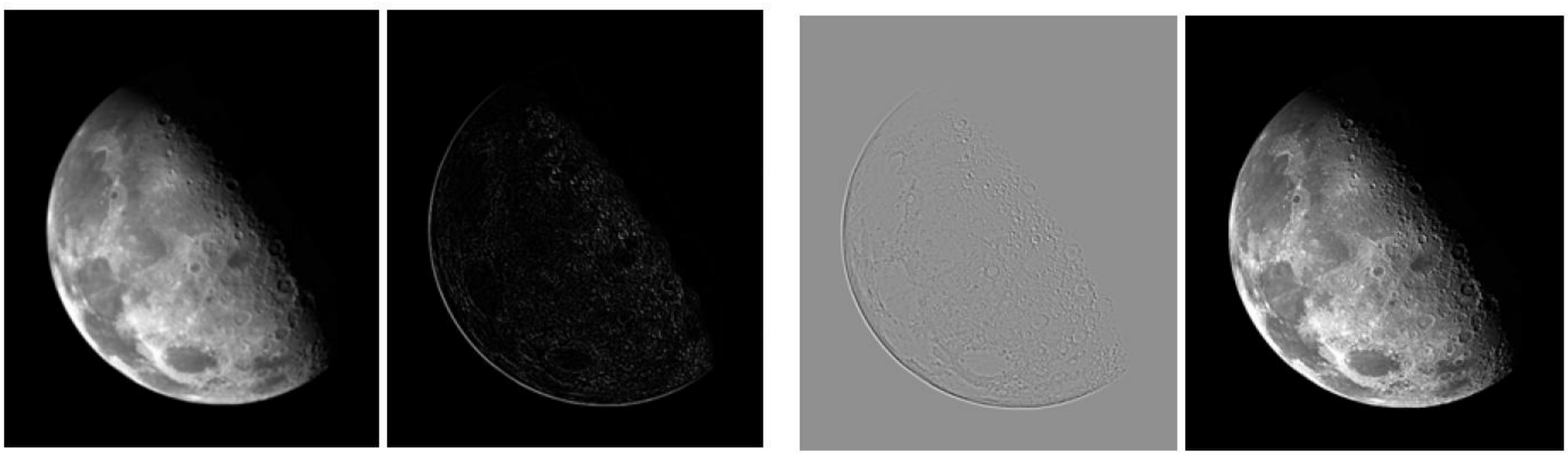
Filtros

Laplaciano

Recuperación del fondo sin perder la información de los bordes

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Original Lapaciano Cambio de escala Mas Fondo



Filtros

Laplaciano

Recuperación del fondo sin perder la información de los bordes

Original

Lapaciano + Fondo

Filtros

Laplaciano

Recuperación del fondo sin perder la información de los bordes

Simplificación (Lapaciano + Fondo)

$$\begin{aligned} g(x, y) &= f(x, y) - \big[f(x + 1, y) + f(x - 1, y) \\ &\quad + f(x, y + 1) + f(x, y - 1)\big] + 4f(x, y) \\ &= 5f(x, y) - \big[f(x + 1, y) + f(x - 1, y) \\ &\quad + f(x, y + 1) + f(x, y - 1)\big]. \end{aligned}$$

Kernel

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1

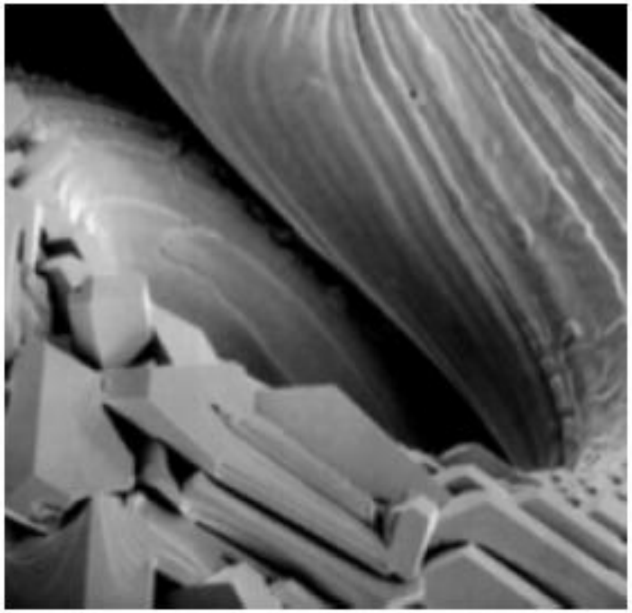
Kernel incluyendo las diagonales

Filtros

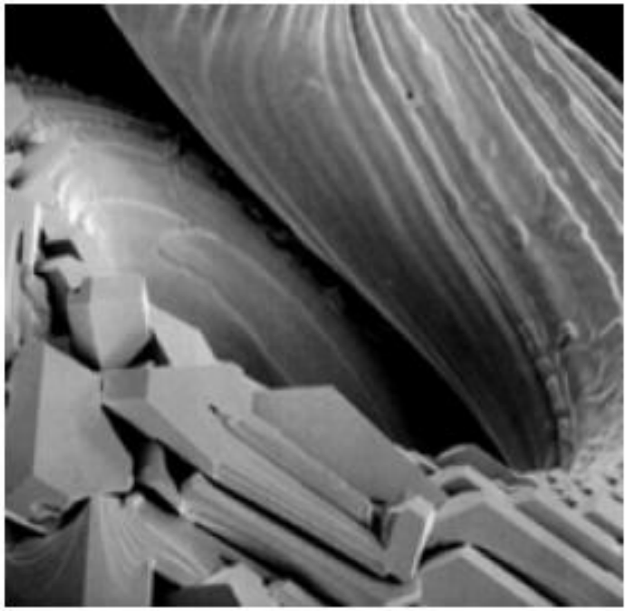
Laplaciano

Recuperación del fondo sin perder la información de los bordes

Original



Lapaciano + Fondo (Solo 90 grados)

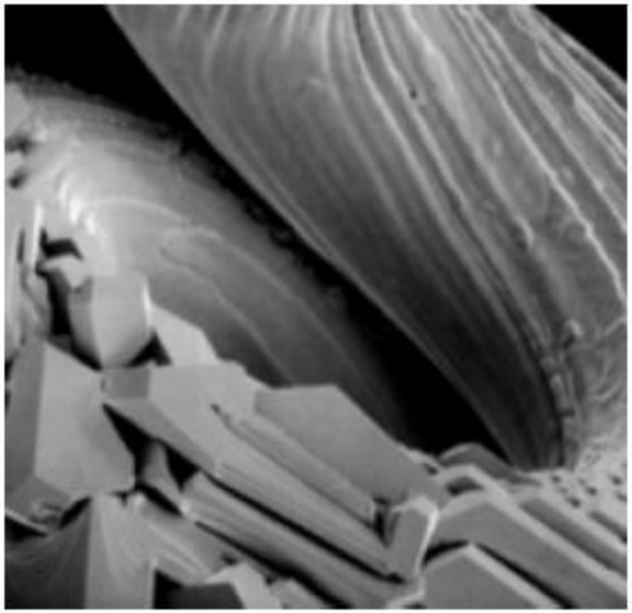


Filtros

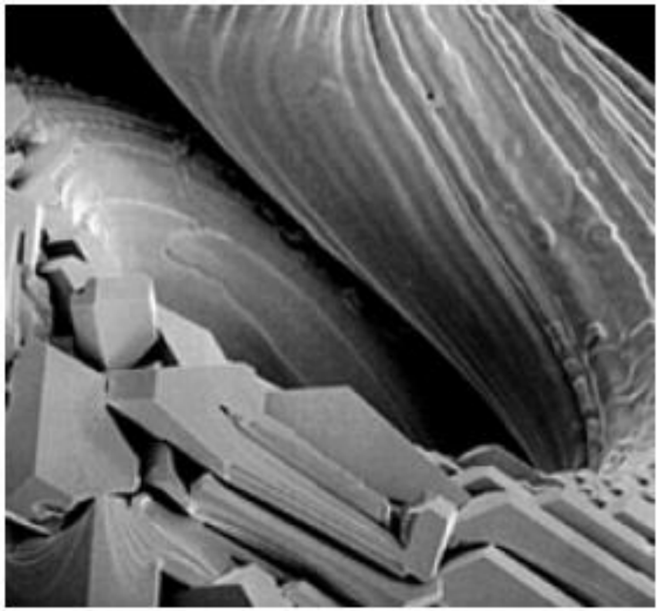
Laplaciano

Recuperación del fondo sin perder la información de los bordes

Original



Lapaciano + Fondo (+ diagonales)

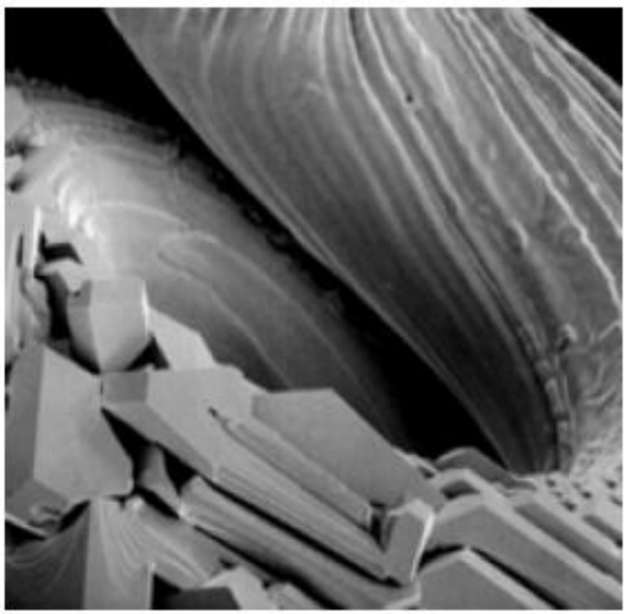


Filtros

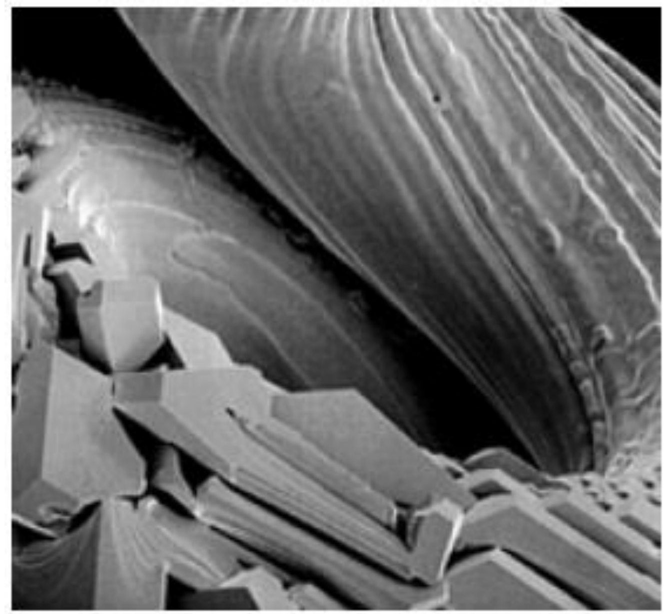
Laplaciano

Recuperación del fondo sin perder la información de los bordes

Lapaciano + Fondo (Solo 90 grados)



Lapaciano + Fondo (+ diagonales)



Filtros

Laplaciano

Mejoramiento mediante el uso de una imagen borrosa

En las épocas de revelado en un cuarto oscuro para obtener una imagen mas nítida se revelaba la superposición de la imagen original con una versión borrosa de la misma invertida (negativo).

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

Mejorada = Original - Blurred version

Filtros

Laplaciano

Mejoramiento mediante el uso de una imagen borrosa

Generalización- High Boost

$$f_{\mathrm{hb}}(x, y) = Af(x, y) - \bar{f}(x, y) \qquad A \geq 1.$$

Mejorada = A * Original - Blurred version

0	−1	0
−1	$A + 4$	−1
0	−1	0

−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

Filtros

Laplaciano

Mejoramiento mediante el uso de una imagen borrosa

Generalización- High Boost

“High Boost” es aplicado en el caso de imágenes muy oscuras variando el coeficiente A se mejora el nivel promedio de gris mejorando así el brillo de la imagen final. Nótese que si A=1 estamos en el caso estándar y a medida que aumentamos A el efecto de mejoramiento de los bordes se va reduciendo hasta el punto donde cuando A es muy grande volvemos a la imagen original

$$f_{\mathrm{hb}}(x, y) = Af(x, y) - \bar{f}(x, y)$$

Filtros

Laplaciano

Mejoramiento mediante el uso de una imagen borrosa

Generalización- High Boost

La ecuación de f_{hb} se puede poner en función de f_s:

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = (A - 1)f(x, y) + f(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = (A - 1)f(x, y) + f_s(x, y)$$

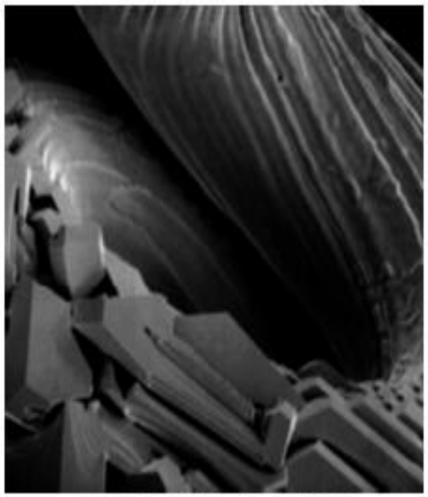
$$f_{hb} = \begin{cases} Af(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ Af(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

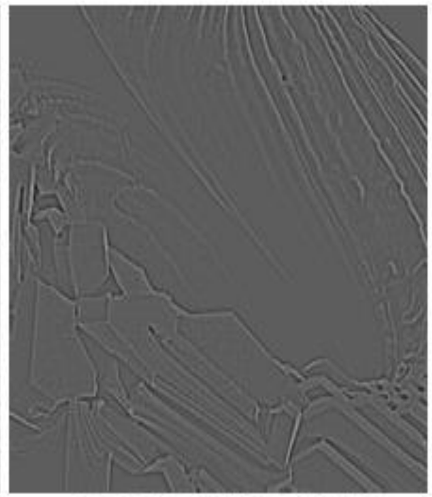
Filtros

Laplaciano

Generalización- High Boost -Ejemplo

Imagen original pero mas oscura





Laplaciano mejorado de la Imagen original con la siguiente mascara con A=0

-1	-1	-1
-1	$A + 8$	-1
-1	-1	-1

A=1

A=1.7

Filtros

Uso de la primera derivada para el mejoramiento de la imagen

Las componenprimeras derivadas en procesamiento de imágenes son implementadas usando el modulo del gradiente. Dadavvvvv una f(x,y) su gradiente en las coordenadas x,y esta dado por

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

El modulo de este vector esta dado por:

$$\begin{aligned} \nabla f &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

Filtros

Uso de la primera derivada para el mejoramiento de la imagen

Las componentes del vector gradiente son operadores lineales no asi el modulo (debido al cuadrado y la raíz). Por el otro lado las derivadas parciales no son isotropicas (invariantes a la rotación) si lo es el modulo.
El calculo del gradiente no es trivial desde el punto de vista computacional. Una posible aproximación es :

$$\nabla f \approx |G_x| + |G_y|$$

Donde se uso el modulo en lugar del cuadrado y la raíz. Esta ecuación preserva los cambios relativos en la escala de grises. En cambio la isotropía no es general solo se limita a algunos casos particulares. De e hecho las mascaras mas populares se limitan a múltiplos de 90 grados

Filtros

Uso de la primera derivada para el mejoramiento de la imagen

Como se ha hecho antes primero formularemos las versiones digitales de las ecuaciones precedentes y luego formulamos las mascaras apropiadas. Usando la notación previa para las mascaras definiremos las nuevas. Así por ejemplo para una mascara de 3x3 el punto central es z5 o sea f(x,y),z1 será f(x-1,y-1).

z_1	z_2	z_3
z_4	z_5	Z_6
z_7	z_8	z_9

La aproximación de la primera derivada en x será: $G_x = (z_8 - z_5)$

recordar que: $\dfrac{\partial f}{\partial x} = f(x+1) - f(x).$

De igual forma la primera derivada en y será: $G_y = (z_6 - z_5)$

Filtros

Uso de la primera derivada para el mejoramiento de la imagen

Recordando que:

$$\begin{aligned}\nabla f &= \mathrm{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}.\end{aligned}$$

Nos queda:

$$\nabla f = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}$$

Y si aproximamos con valores absolutos:

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Se puede implementar con las siguientes mascaras:

−1	0
0	1

0	−1
1	0

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Filtros

Uso de la primera derivada para el mejoramiento de la imagen

Mascaras de tamaño par no son deseables. Una aproximación de 3x3 (Sobel):

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|.$$

Gx y Gy:

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

El valor 2 permite cierta suavizar dando mas importancia al píxel central

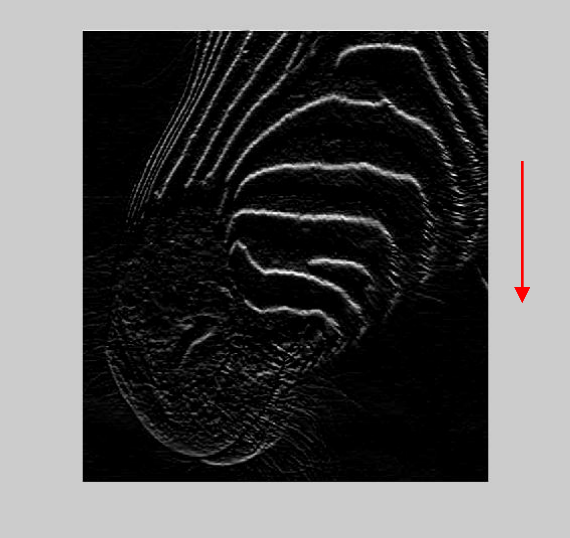
Matlab Gy = -fspecial('sobel') Gx= h’

Filtros

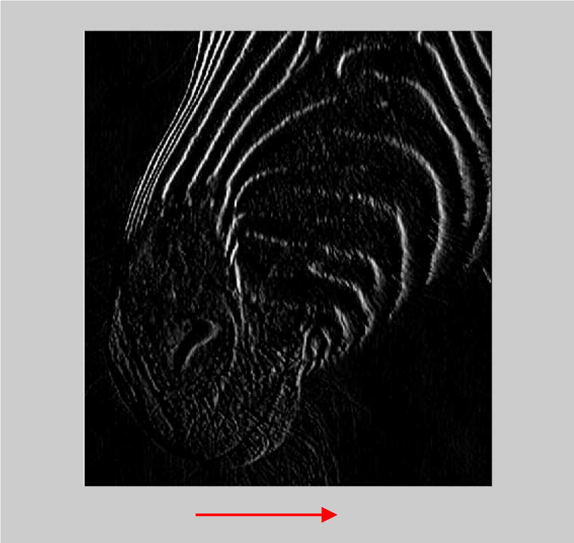
Sobel Ejemplo

Original

Gy

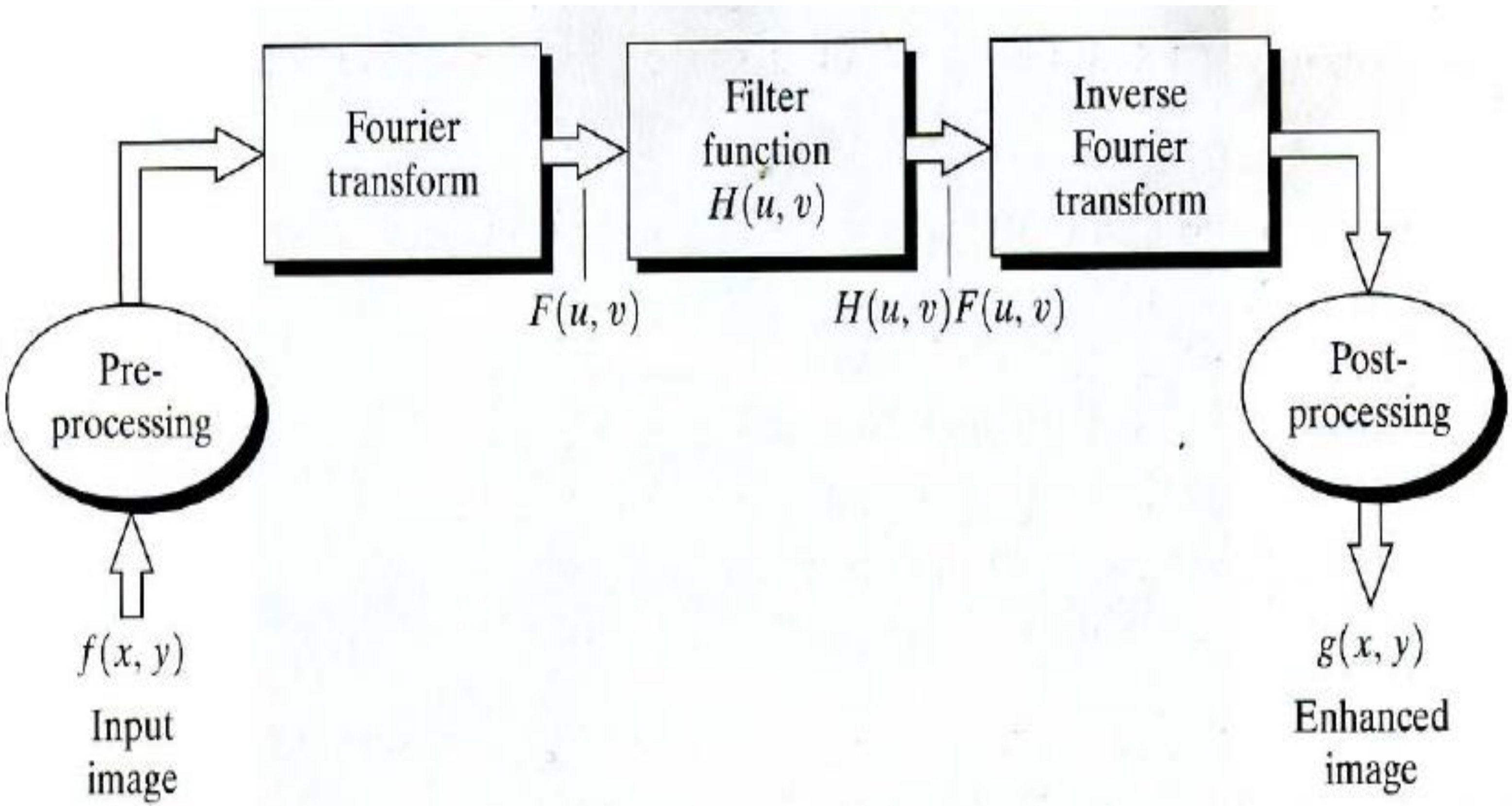


Gx



Filtros

Filtrado en el dominio de la frecuencia



Filtros

Filtrado en el dominio de la frecuencia

1- Multiplicar f(x,y) por $(-1)^{x+y}$ para centrar el espectro[nota1]

2- Hallar la DFT de (1) = F(u,v)

3- Multiplicar H(u,v) por F(u,v)

4- Encontrar la DFT inversa de (3)

5- Encontrar la parte real de (4)

6- Multiplicar (5) por $(-1)^{x+y}$

Nota 1: $\Im\left[f(x, y)(-1)^{x+y}\right] = F(u - M/2, v - N/2)$

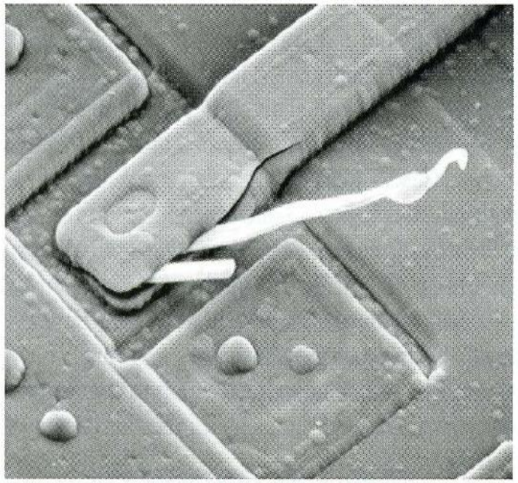
Filtros

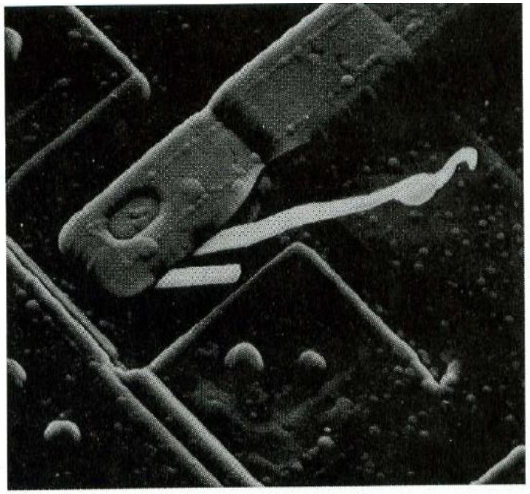
Filtrado en el dominio de la frecuencia

Ejemplo:

$$H(u,v) = \begin{cases} 0 & \text{if} \quad (u,v) = (M/2, N/2) \\ 1 & \text{otherwise.} \end{cases}$$

En este caso el filtro (Notch) elimina la componente de continua que representa el valor medio de gris en la imagen

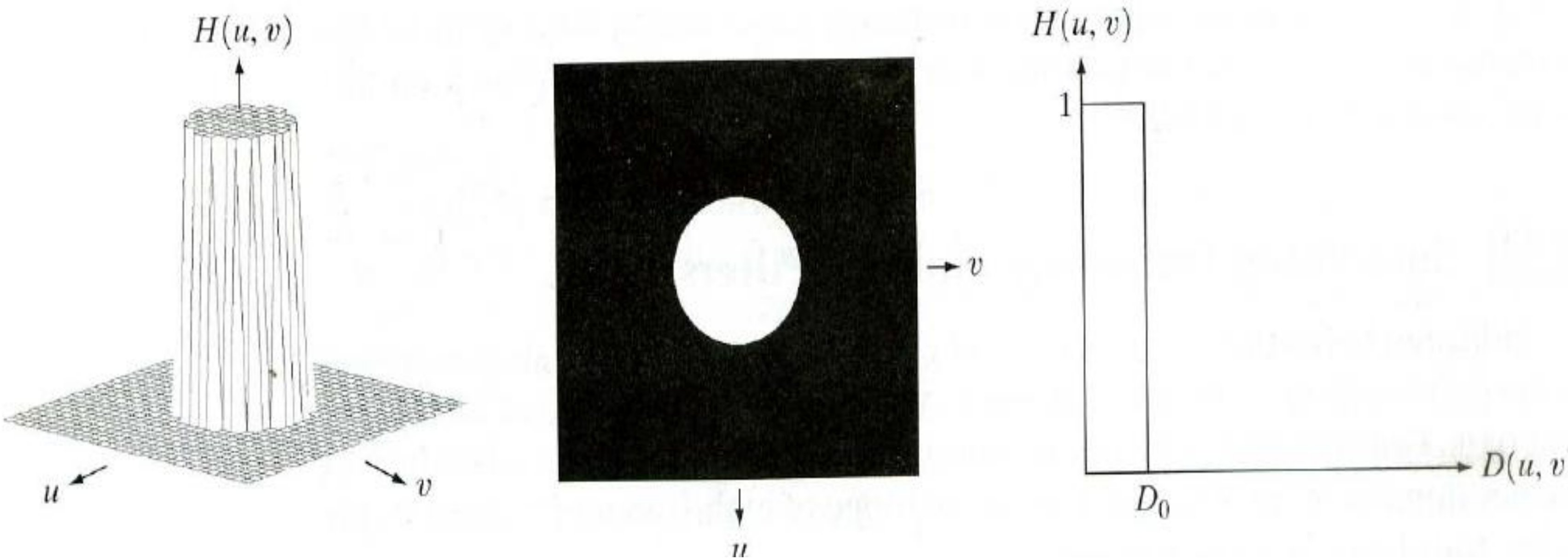




Filtros

Filtrado en el dominio de la frecuencia

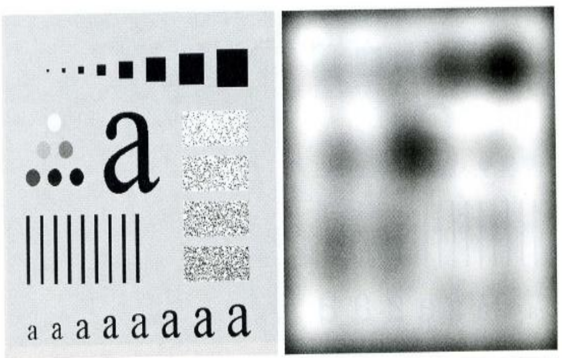
Filtro Pasabajos Ideal

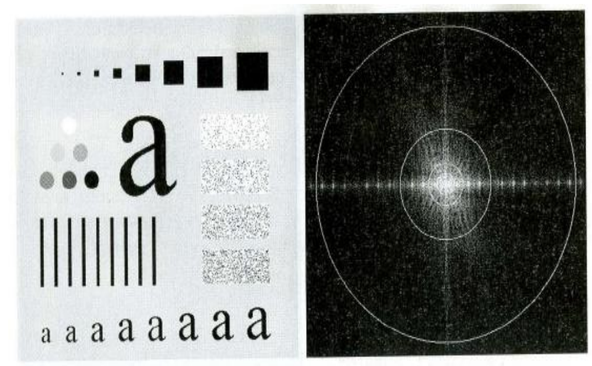


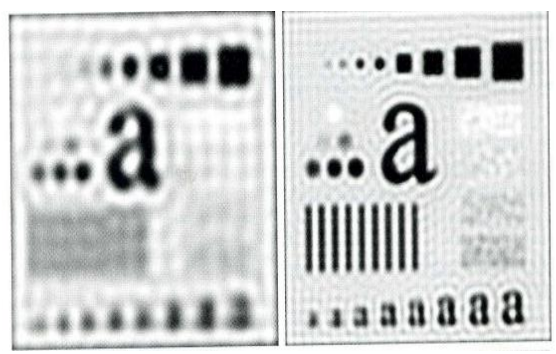
Filtros

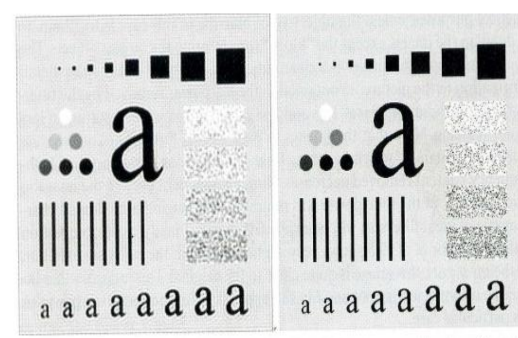
Filtrado en el dominio de la frecuencia

Filtro Pasabajos Ideal









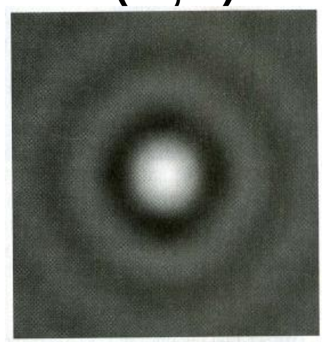
Filtros

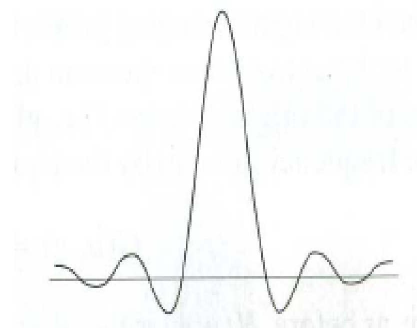
Filtrado en el dominio de la frecuencia

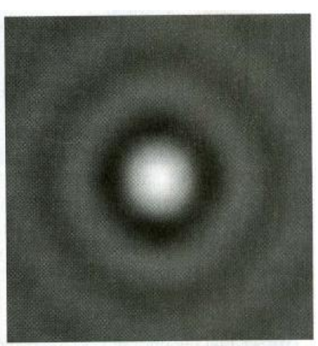
Blurr y Ringing

f(x,y)

F(u,v)

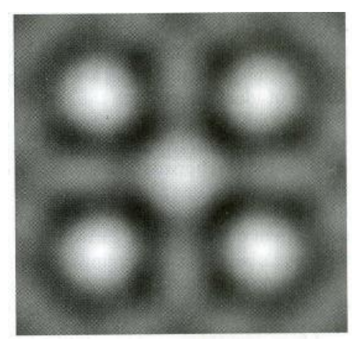






*

=

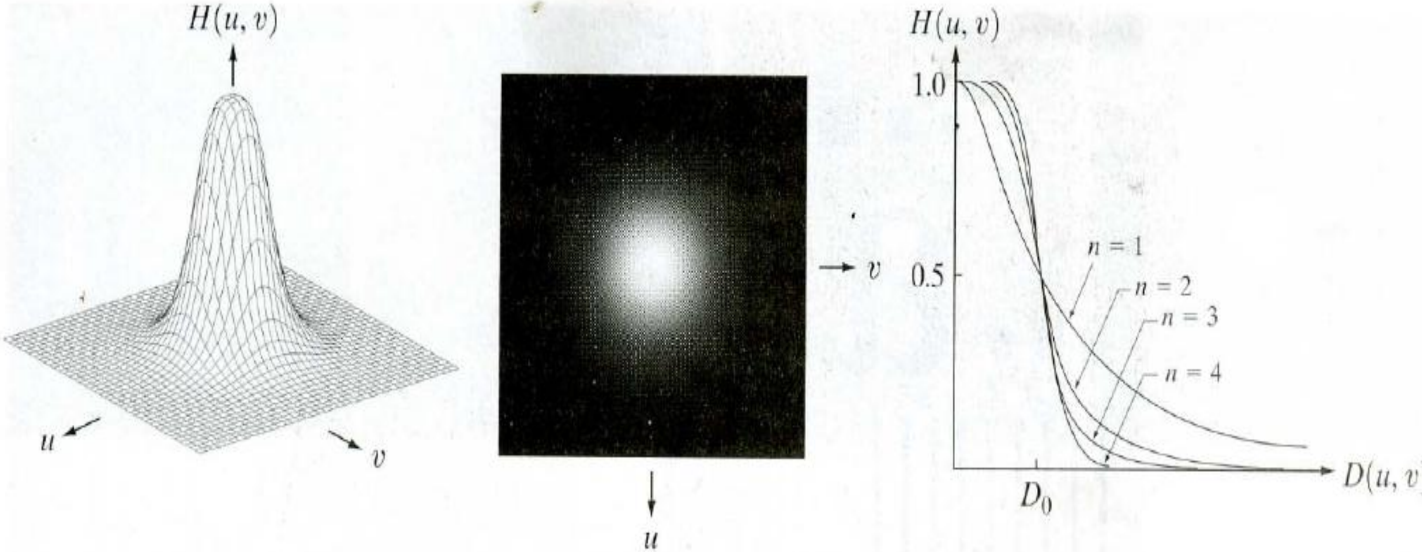


Filtros

Filtrado en el dominio de la frecuencia

Butterworth (poco ringing)

$H(u.v)=0.5\ H(u,v)_{max}$ cuando $D(u,v)=D_0$

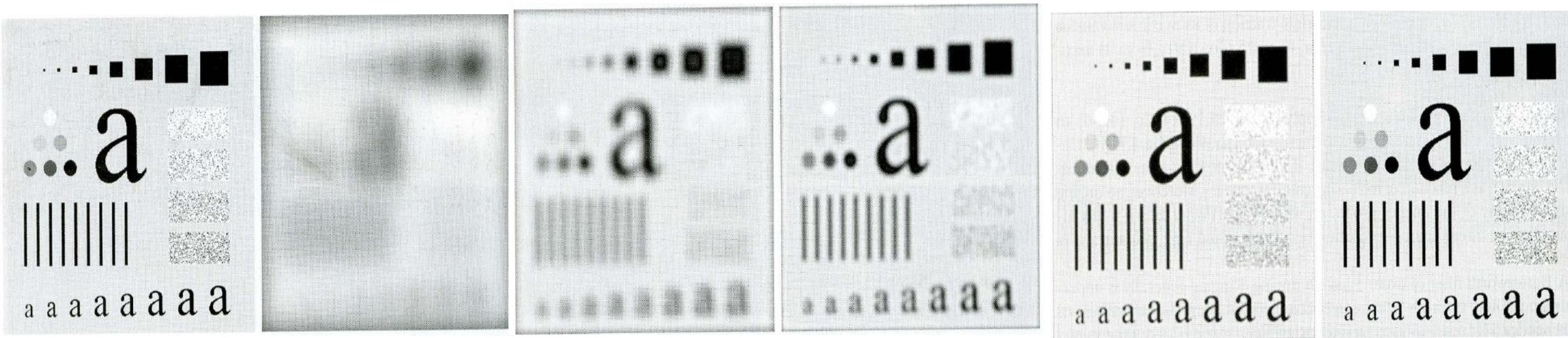


$$H(u,v) = \frac{1}{1 + \left[D(u,v)/D_0\right]^{2n}}$$

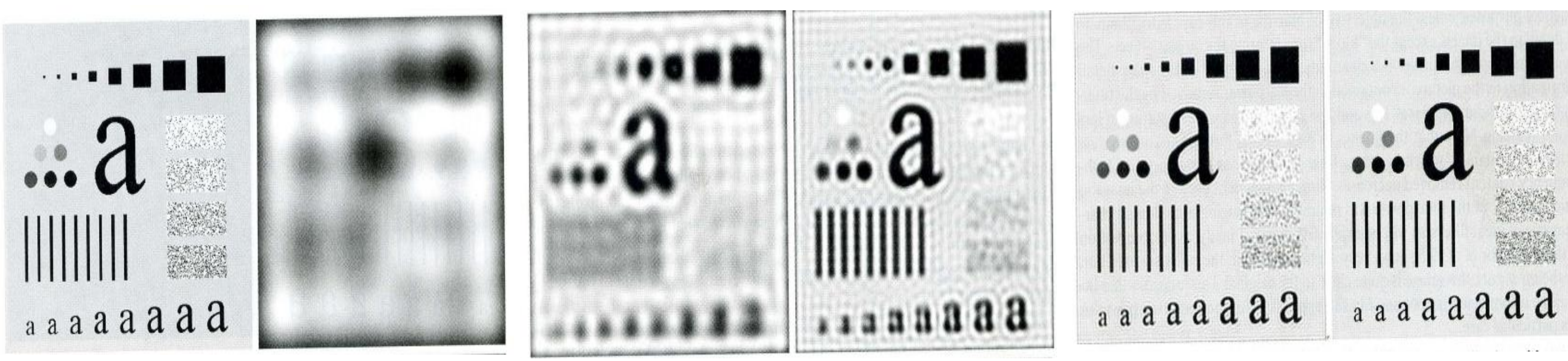
Filtros

Filtrado en el dominio de la frecuencia

Butterworth (poco ringing)   n=2



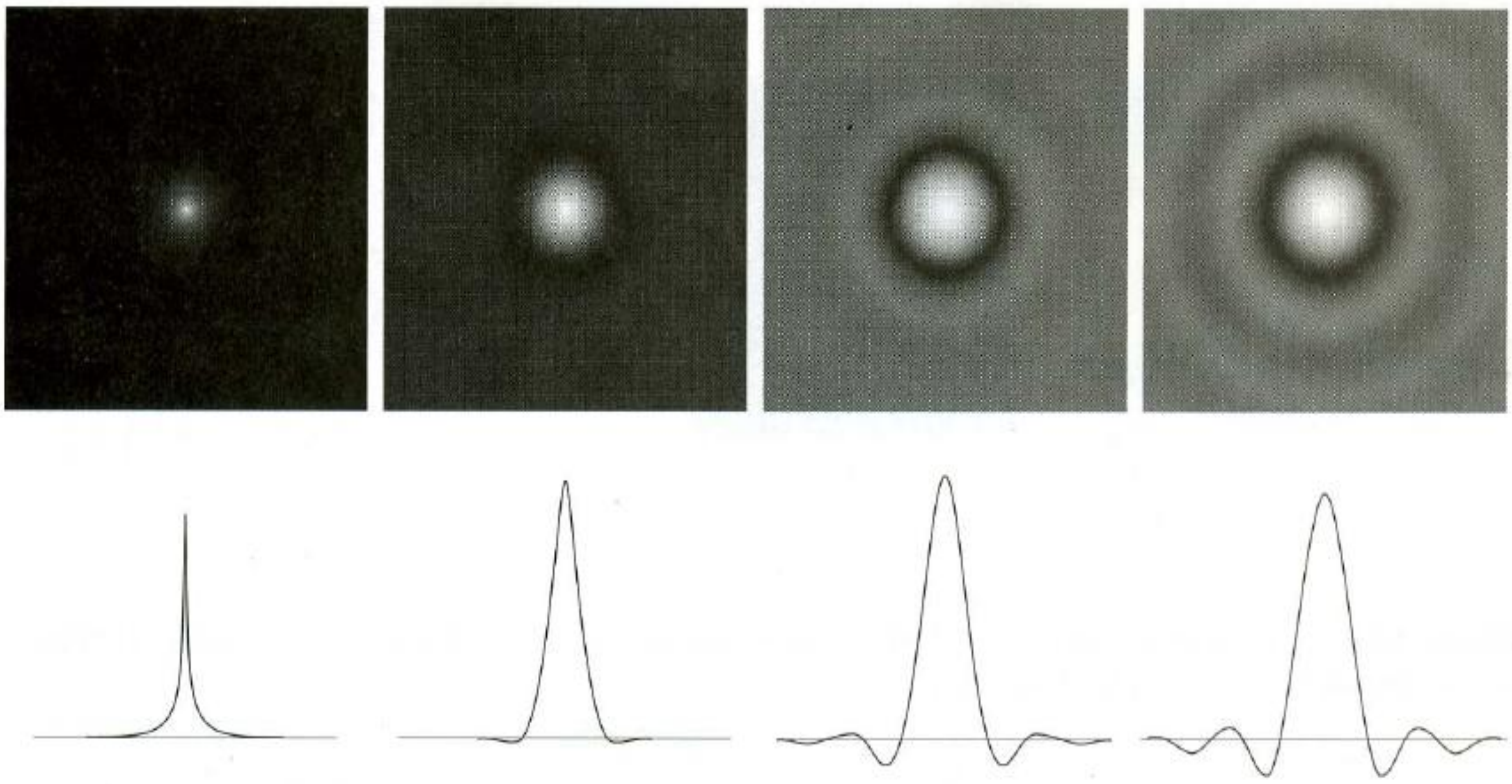
Ideal Low pass (Alto ringing)



Filtros

Filtrado en el dominio de la frecuencia

Butterworth N =1,2,5,20



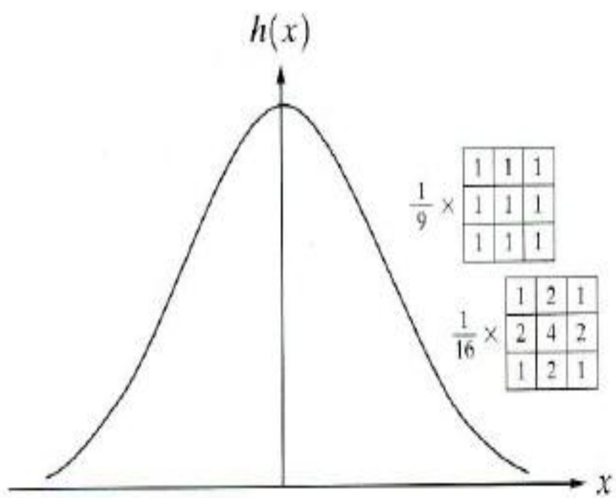
Filtros

Filtrado en el dominio de la frecuencia

Filtro Gausiano 1-D

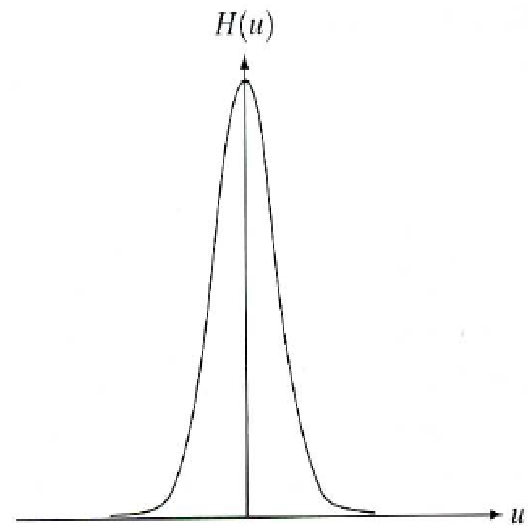
Espacio

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 x^2}.$$



Frecuencia

$$H(u) = A e^{-u^2/2\sigma^2}$$



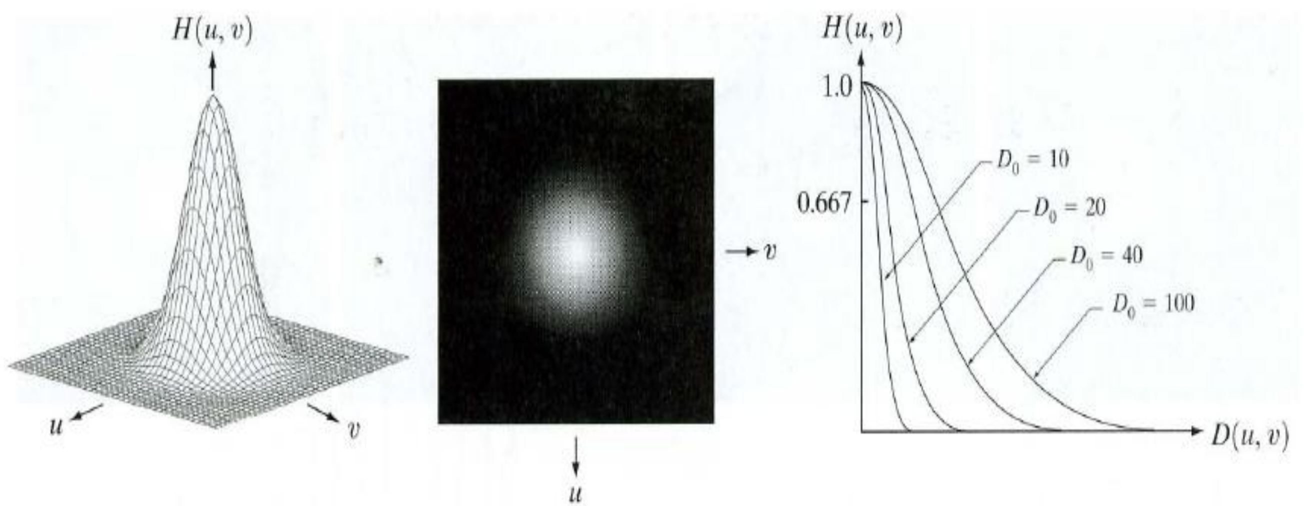
Filtros

Filtrado en el dominio de la frecuencia

Filtro Gausiano (No ringing)

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2} \qquad \sigma = D_0 \qquad H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

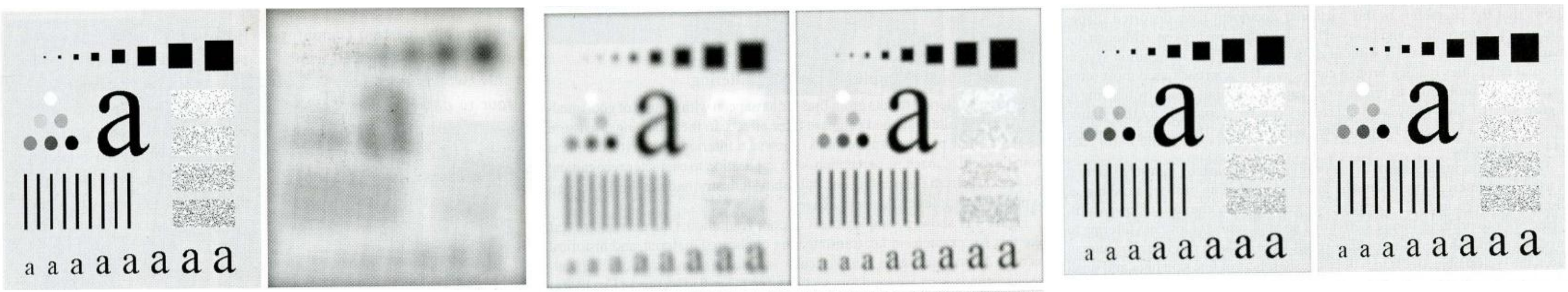
D_0 Frecuencia de corte $\qquad D(u,v)=D_0 \qquad$ Gain=0.607



Filtros

Filtrado en el dominio de la frecuencia

Filtro Gausiano (No ringing)

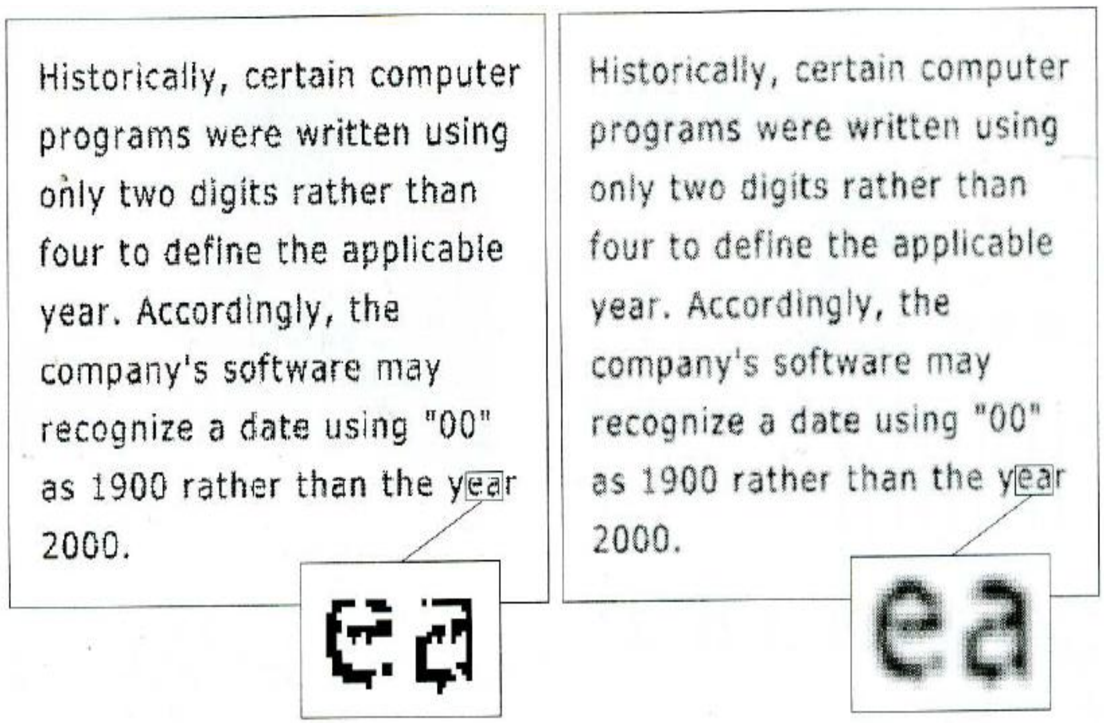


Radios = 5,15,30,80 y 230

Filtros

Filtrado en el dominio de la frecuencia

Filtro Gausiano Ejemplo



$D_0 = 80$

Filtros

Original

D_0 =100

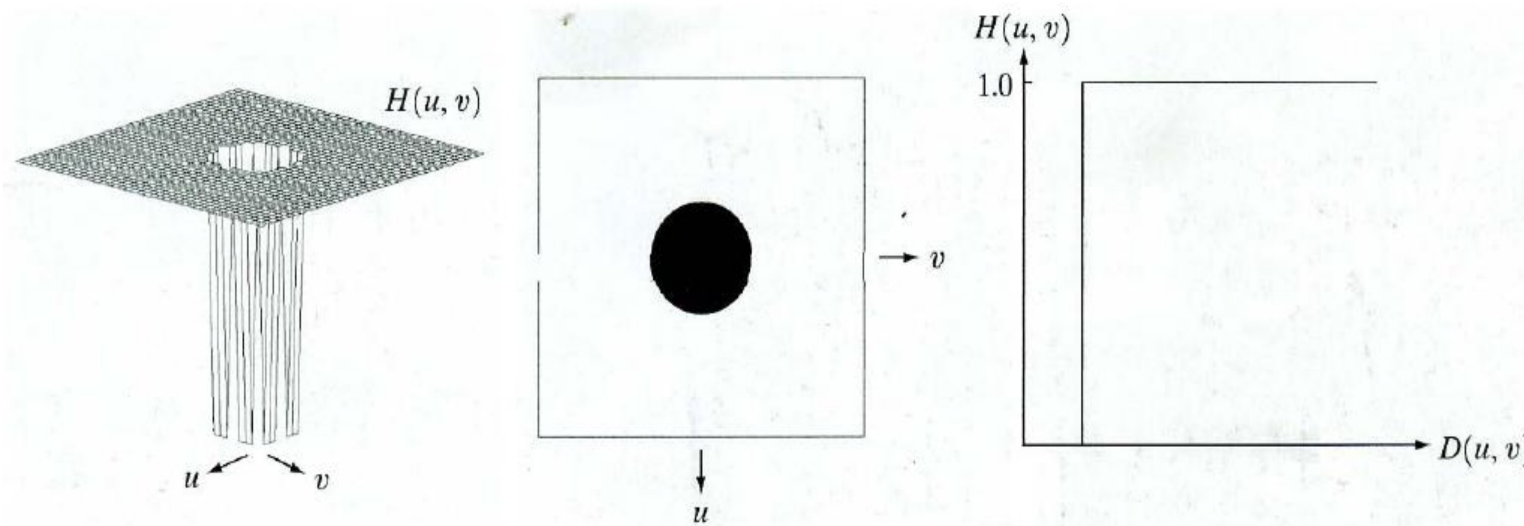
D_0 = 80

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos

$$H_{\mathrm{hp}}(u, v) = 1 - H_{\mathrm{lp}}(u, v)$$

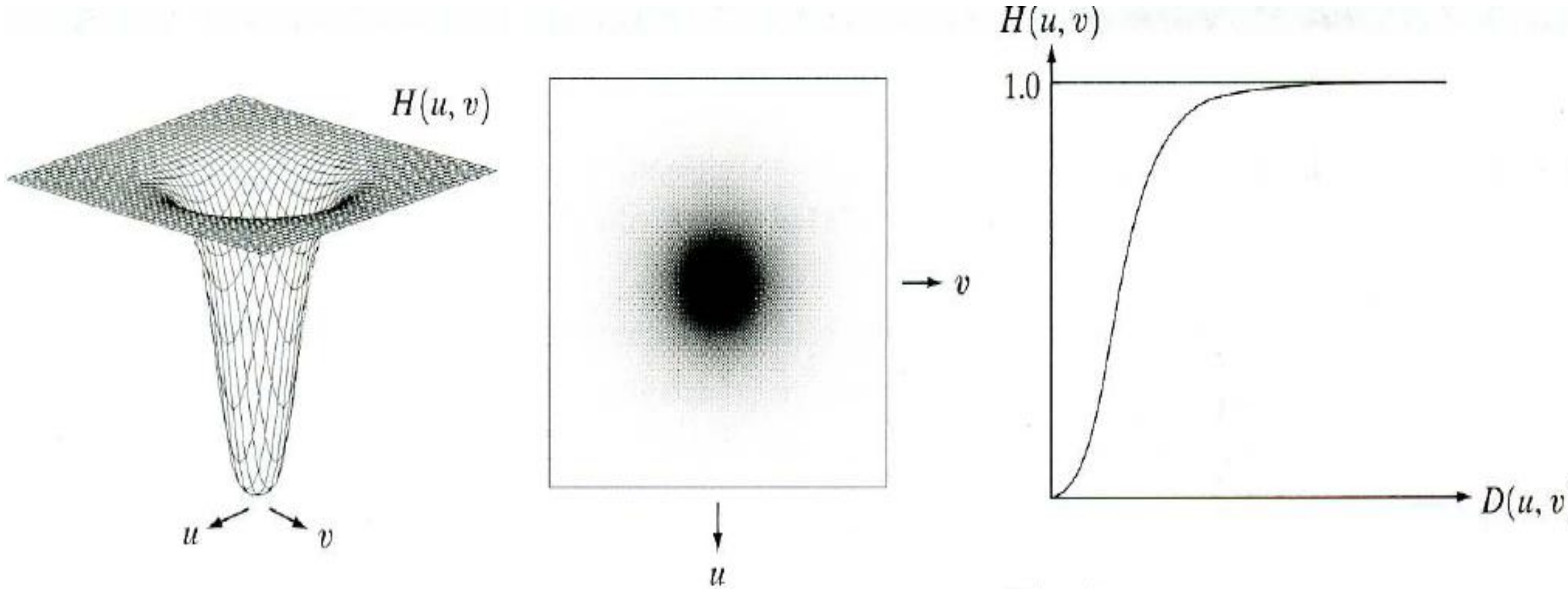


Ideal

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos

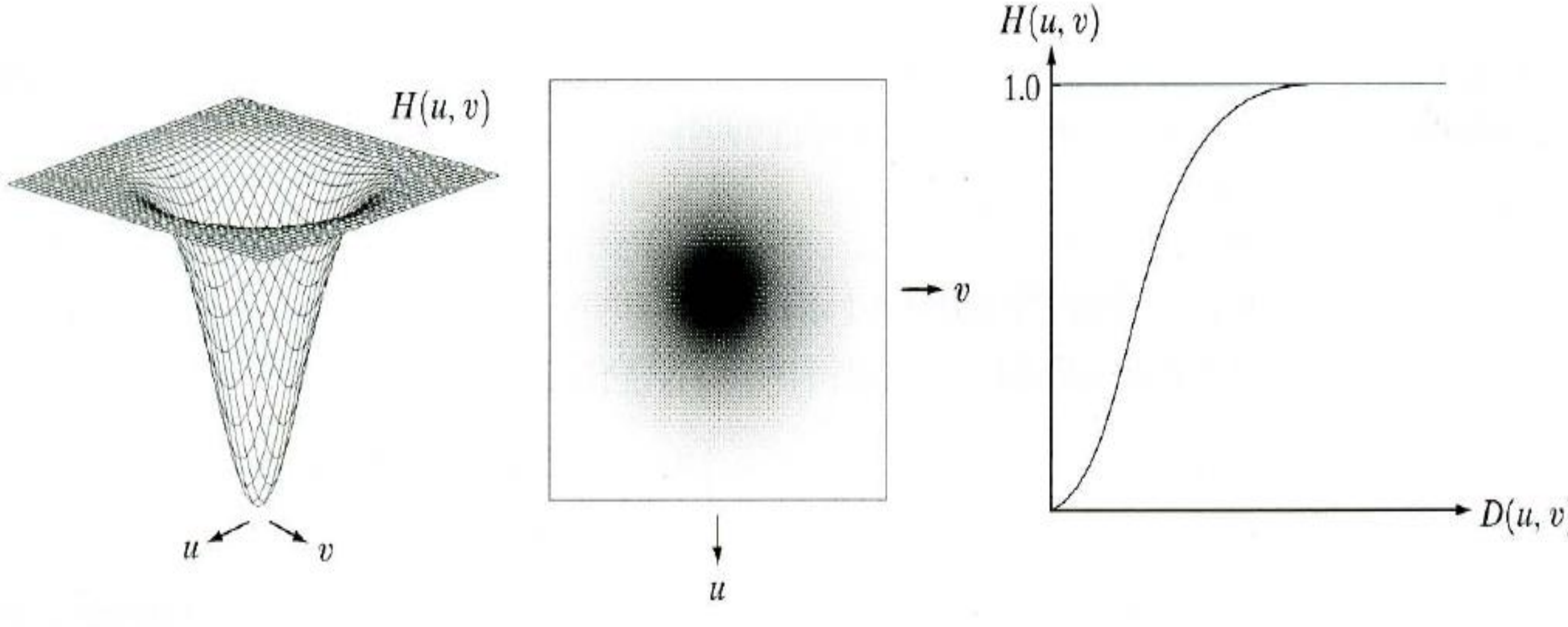


Butterworth

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos

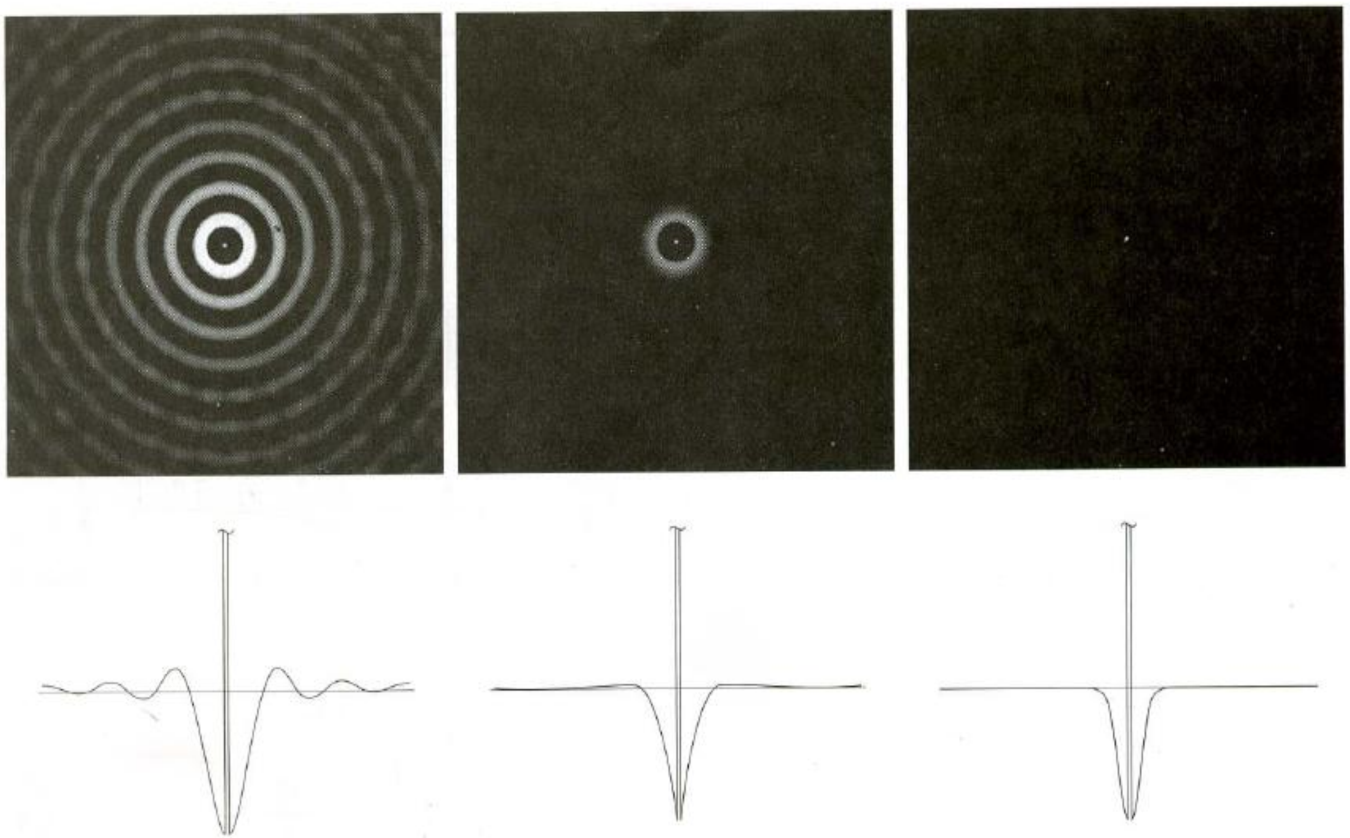


Gaussiano

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos



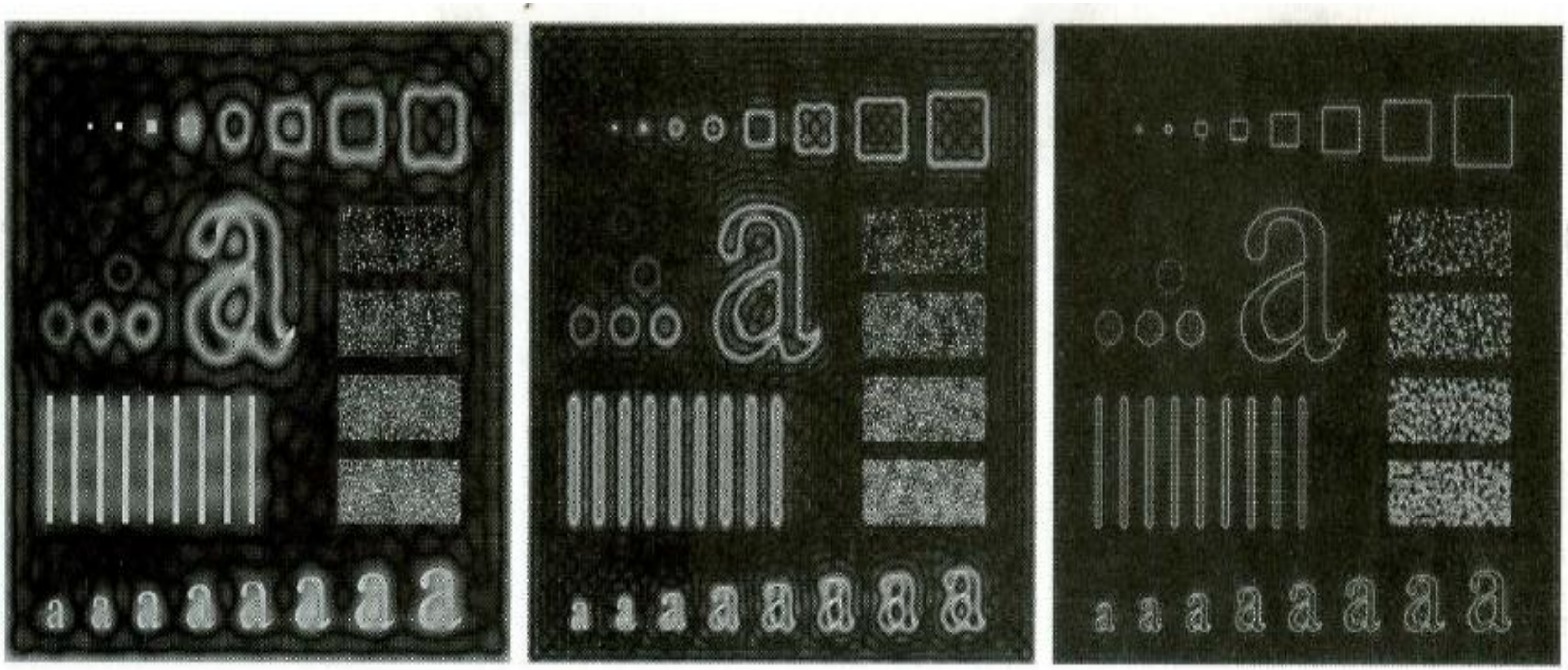
Ideal　　Butterworth　　Gaussiano

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos ideal

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

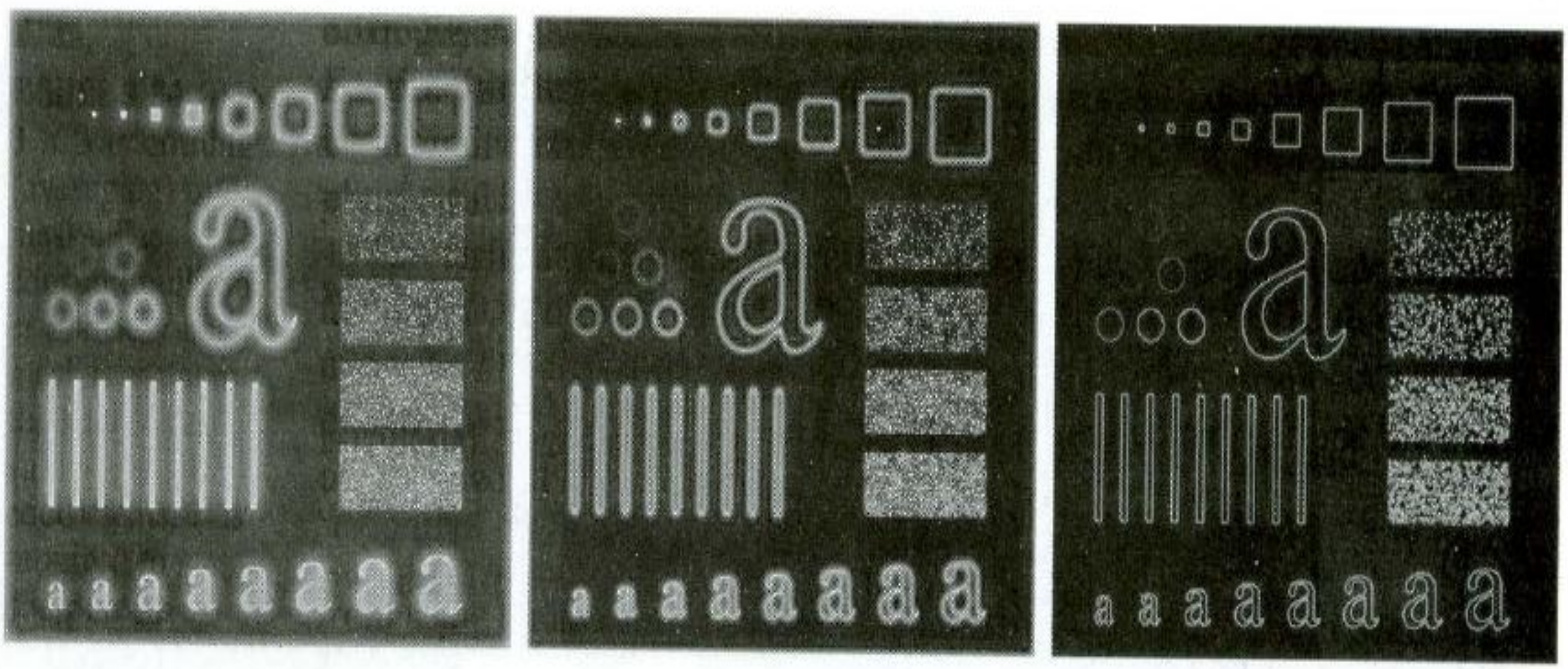


Do=15 Do=30 Do=80

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos Butterworth N=2

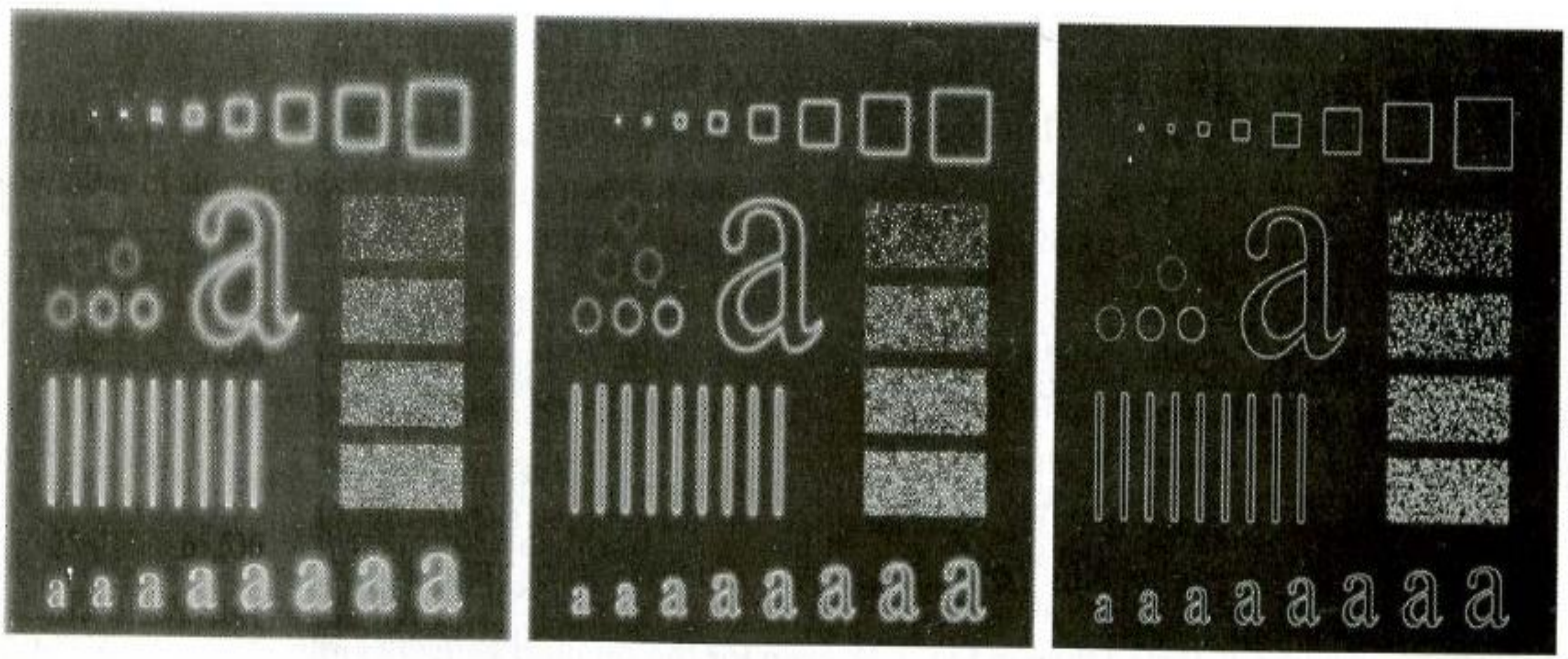


Do=15 Do=30 Do=80

Filtros

Filtrado en el dominio de la frecuencia

Filtros Pasa Altos Gausiano



Do=15 Do=30 Do=80

Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering

$$f(x, y) = i(x, y)r(x, y)$$

$$\Im\{f(x, y)\} \neq \Im\{i(x, y)\}\Im\{r(x, y)\}$$

Definimos:

$$\begin{aligned} z(x, y) &= \ln f(x, y) \\ &= \ln i(x, y) + \ln r(x, y). \end{aligned}$$

$$\begin{aligned} \Im\{z(x, y)\} &= \Im\{\ln f(x, y)\} \\ &= \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\} \end{aligned}$$

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering

$$\begin{aligned} S(u, v) &= H(u, v) Z(u, v) \\ &= H(u, v) F_i(u, v) + H(u, v) F_r(u, v) \end{aligned}$$

$$\begin{aligned} s(x, y) &= \Im^{-1}\{S(u, v)\} \\ &= \Im^{-1}\{H(u, v) F_i(u, v)\} + \Im^{-1}\{H(u, v) F_r(u, v)\} \end{aligned}$$

Definimos:

$$\left.\begin{aligned} i'(x, y) &= \Im^{-1}\{H(u, v) F_i(u, v)\} \\ r'(x, y) &= \Im^{-1}\{H(u, v) F_r(u, v)\} \end{aligned}\right\} s(x, y) = i'(x, y) + r'(x, y)$$

Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering

$$
\begin{aligned}
g(x, y) &= e^{s(x, y)} \\
&= e^{i'(x, y)} \cdot e^{r'(x, y)} \\
&= i_0(x, y) r_0(x, y)
\end{aligned}
$$

Imagen mejorada

Donde $i_0(x, y) = e^{i'(x, y)}$ $\quad r_0(x, y) = e^{r'(x, y)}$

$$
f(x, y) \Rightarrow \boxed{\ln} \Rightarrow \boxed{\text{DFT}} \Rightarrow \boxed{H(u, v)} \Rightarrow \boxed{(\text{DFT})^{-1}} \Rightarrow \boxed{\exp} \Rightarrow g(x, y)
$$

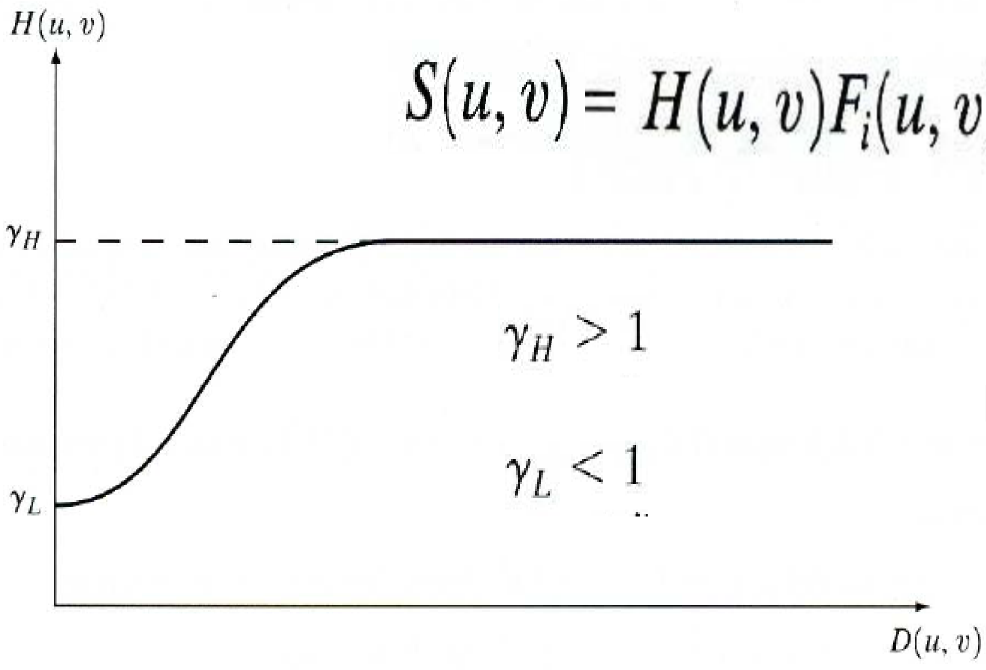
Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering

La iluminación esa caracterizada por variaciones lentas (bajas frecuencias) mientras que la reflectancia esta asociada a cambios abruptos (altas frecuencias) Del desarrollo previo vemos que podemos procesar en forma independiente la iluminación y la reflectancia si definimos H(u,v) de la siguiente forma:

$$S(u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v)$$



D(u,v): Distancia desde el origen

Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering: Ejemplo

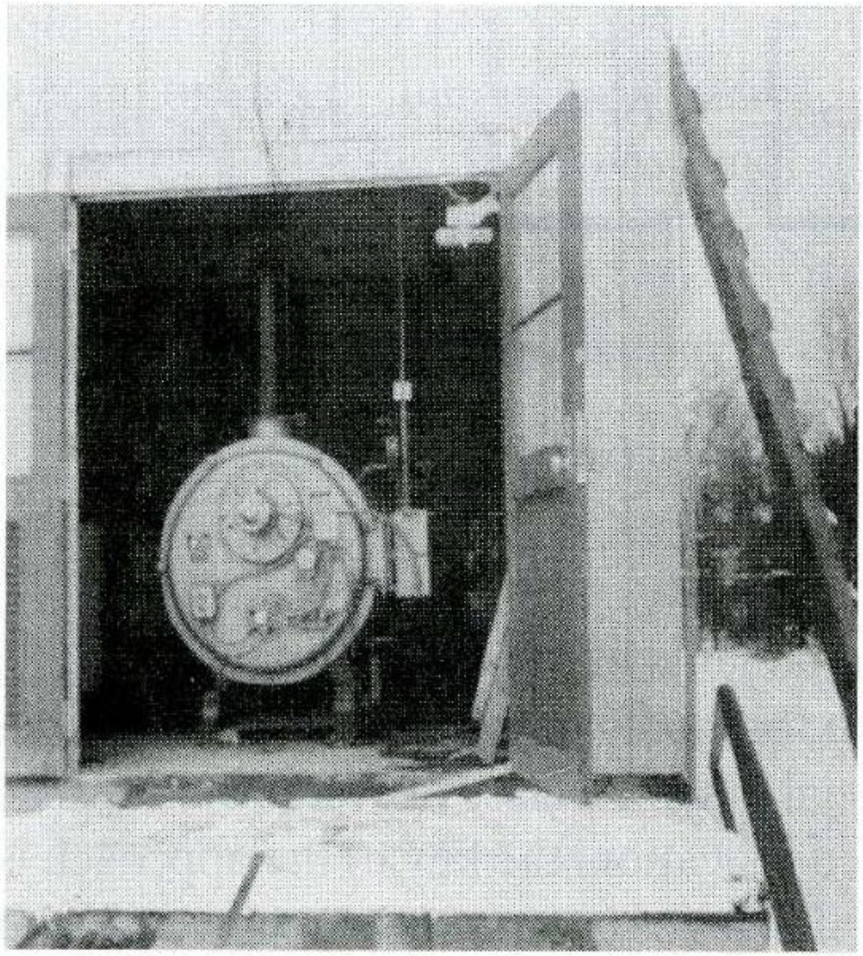
La curva previa puede aproximarse mediante cualquier forma básica de un filtro pasa altos por ejemplo usando la siguiente forma del filtro gausiano nos da:

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left(D^2(u,v)/D_0^2\right)}\right] + \gamma_L$$

Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering: Ejemplo



Filtros

Filtrado en el dominio de la frecuencia

Homomorphic Filtering: Ejemplo

Filtros

Filtrado en el dominio de la frecuencia

Filtros FIR

Los Filtros FIR tienen características que los hacen ideales para el filtrado de imágenes.

- Los Filtros FIR en 2D son extensiones naturales de los 1D
- Métodos de diseño son conocidos y confiables
- Se los puede diseñar de manera que tengan fase lineal lo que evitaría la distorsión.

Filtros

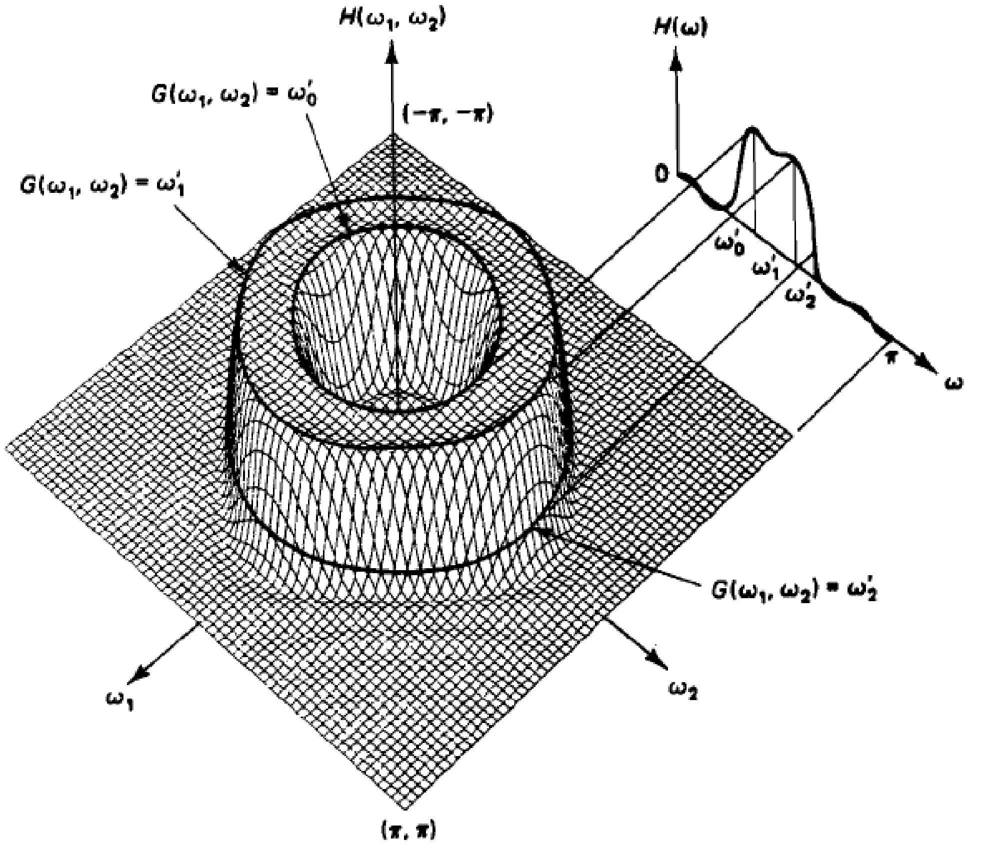
Filtrado en el dominio de la frecuencia

Transformación en frecuencia

Los Filtros FIR se pueden diseñar a partir de filtros FIR   1-D mediante una transformación denominada "**Frequency transformation Method** "

Ver Jae-Lim Cap 4 pag 218

$$H(\omega_1, \omega_2) = H(\omega)\big|_{\omega = G(\omega_1, \omega_2)}$$



Filtros

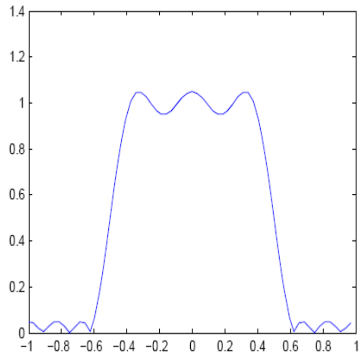
Filtrado en el dominio de la frecuencia

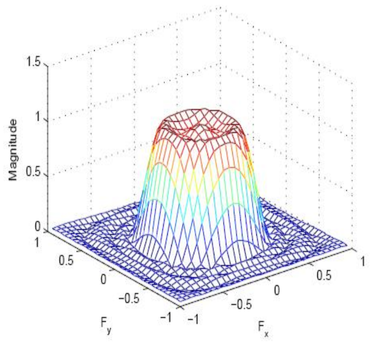
Transformación en frecuencia

Matlab implementa esta transformación mediante la función **ftrans2**

Ejemplo: Filtro Equiriple Parks-McClellan (Matlab Function remez o firpm)

```
b = remez(10,[0 0.4 0.6 1],[1 1 0 0]);
h = ftrans2(b);
[H,w] = freqz(b,1,64,'whole');
colormap(jet(64))
plot(w/pi-1,fftshift(abs(H)))
figure, freqz2(h,[32 32])
```



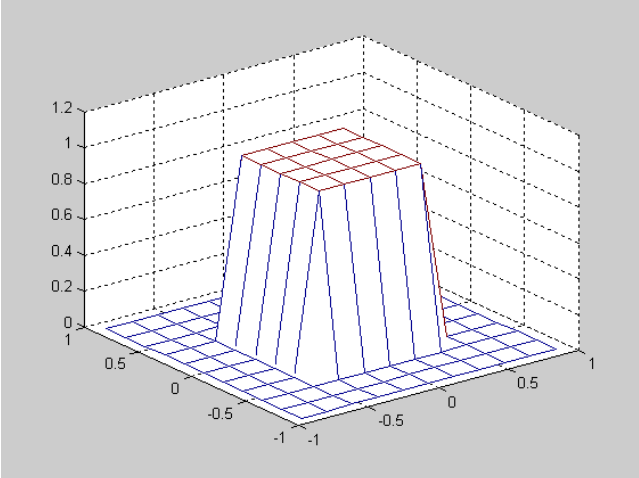


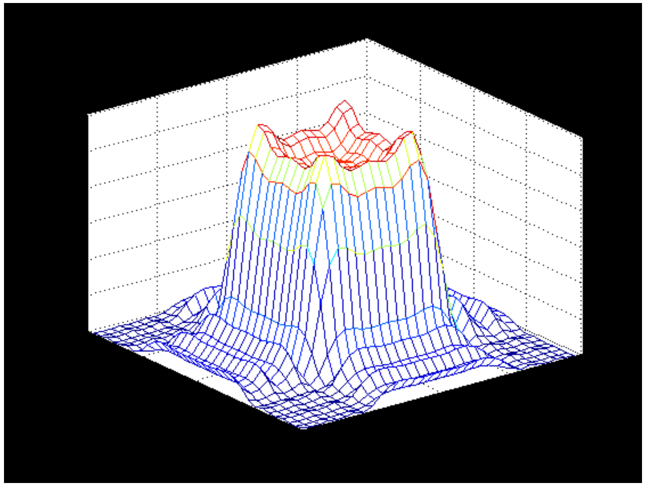
Filtros

Filtrado en el dominio de la frecuencia

Ejemplo: Muestreo en frecuencia (Matlab Función fsamp2)

```
Hd = zeros(11,11); Hd(4:8,4:8) = 1;
[f1,f2] = freqspace(11,'meshgrid');
mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
h = fsamp2(Hd);
figure, freqz2(h,[32 32]), axis([-1 1 -1 1 0 1.2])
```





Filtros

Filtrado en el dominio de la frecuencia

Ejemplo: Diseño por ventanas (Matlab Función fwind1)

```
Hd = zeros(11,11); Hd(4:8,4:8) = 1;
[f1,f2] = freqspace(11,'meshgrid');
mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
h = fwind1(Hd,hamming(11));
figure, freqz2(h,[32 32]), axis([-1 1 -1 1 0 1.2])
```