

Procesamiento Digital de Imágenes

Filtrado en el Dominio Frecuencial

Proc. Digital de Imágenes

1

Transformada Discreta de Fourier 2D

Sea

$$f(x, y), \quad x = 0, \dots, M-1, \quad y = 0, \dots, N-1$$

la imagen original de dimensiones $M \times N$. La transformada discreta de Fourier 2D de f se define como

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad (1)$$

donde $u = 0, 1, 2, \dots, M-1$, $v = 0, 1, 2, \dots, N-1$ son las variables frecuencia. La región rectangular definida por $u = 0, 1, 2, \dots, M-1$, $v = 0, 1, 2, \dots, N-1$ se conoce como **rectángulo de frecuencia**, que claramente es de las mismas dimensiones que la imagen.

Proc. Digital de Imágenes

2

La **Transformada Discreta de Fourier Inversa**, viene dada por

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (2)$$

$$x = 0, \dots, M - 1, \quad y = 0, \dots, N - 1$$

Es decir, dada la transformada $F(u, v)$ se puede recuperar la imagen mediante la transformada inversa. A $F(u, v)$ se los suele denominar **coeficientes de Fourier** en la expansión (2).

Aún si la señal $f(x, y)$ es real, su transformada es compleja. Para visualizar la transformada se suele graficar su módulo $|F(u, v)|$ (espectro de módulo) como una imagen, o equivalentemente su espectro de densidad de energía $|F(u, v)|^2$

- Notar de la expresión (1) que la DFT es periódica en las dos direcciones u y v con períodos M y N , respectivamente.
- Notar, de (2), que ésta también es una propiedad de la DFT Inversa.

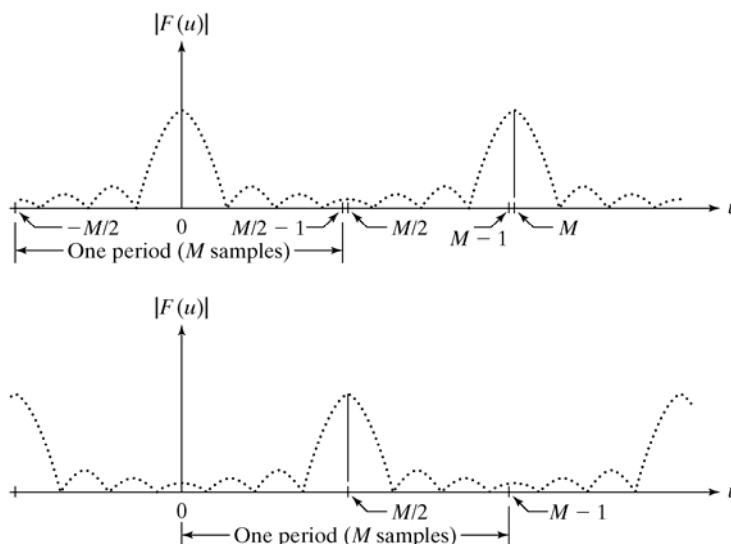
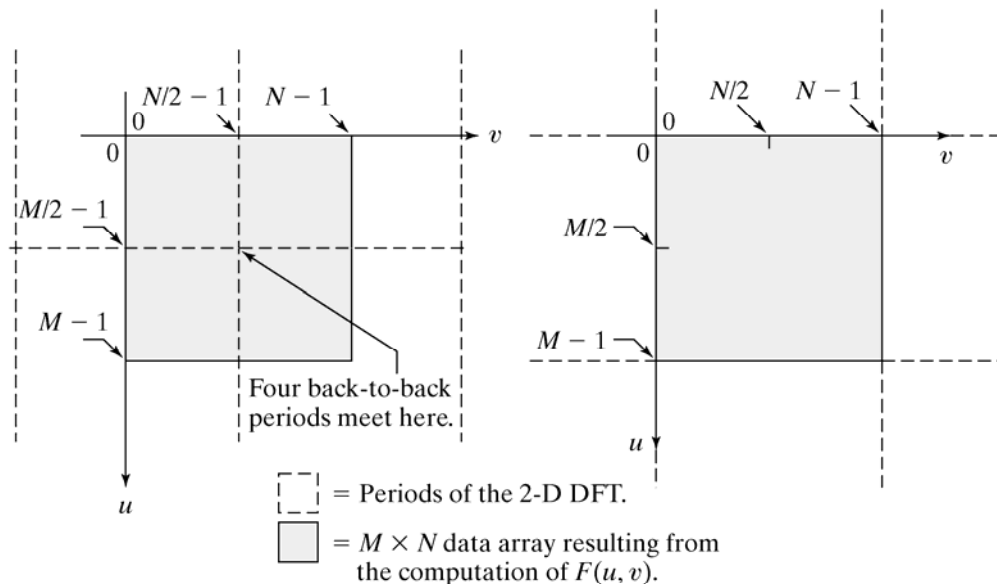


FIGURE 4.1
 (a) Fourier spectrum showing back-to-back half periods in the interval $[0, M - 1]$.
 (b) Centered spectrum in the same interval, obtained by multiplying $f(x)$ by $(-1)^x$ prior to computing the Fourier transform.



a b

FIGURE 4.2 (a) $M \times N$ Fourier spectrum (shaded), showing four back-to-back quarter periods contained in the spectrum data. (b) Spectrum obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ prior to computing the Fourier transform. Only one period is shown shaded because this is the data that would be obtained by an implementation of the equation for $F(u, v)$.

Cómputo de la 2D-DFT en Matlab

El comando

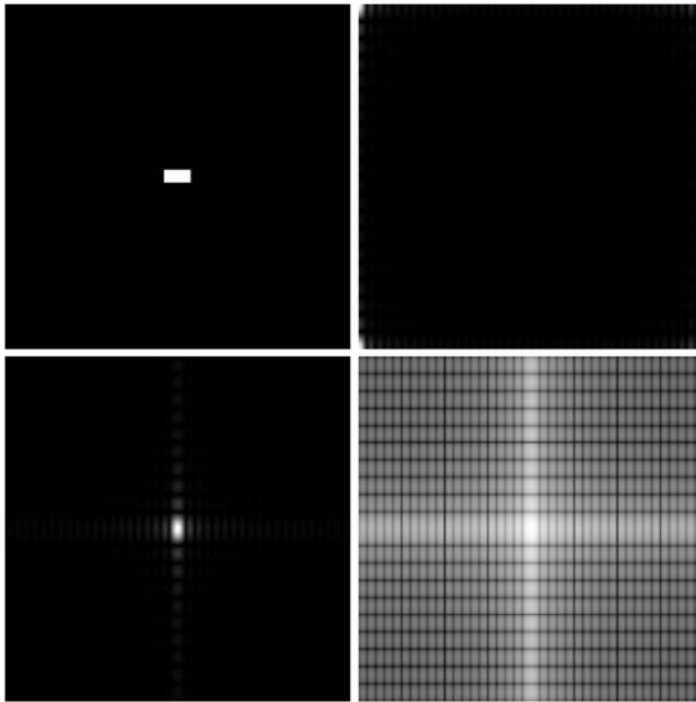
```
>> F= fft2(f);
```

devuelve la 2D-DFT de la imagen f (de dimensiones $M \times N$), en un arreglo de dimensiones $M \times N$, con los datos ubicados como en Fig. 4.2(a).

Cuando la DFT es usada para filtrado es necesario completar la imagen original con ceros (*zero padding*). En este caso, la sintáxis es:

```
>> F= fft2(f, P, Q);
```

que resulta en una matriz (compleja) de dimensiones $P \times Q$.



a b
c d

FIGURE 4.3

(a) A simple image.
(b) Fourier spectrum.
(c) Centered spectrum.
(d) Spectrum visually enhanced by a log transformation.

```
>>F=fft2(f);
>>S=abs(F);
>>imshow(S,[])
>>Fc=fftshift(F);
>>imshow(abs(Fc),[])
>>S2=log(1+abs(Fc));
>>imshow(S2,[])
```

La imagen original f puede recuperarse a partir de la DFT con el comando

```
>>f=real(iff2(F));
```



En teoría, si la imagen original es real entonces $\text{iff2}(F)$ debería ser real. Debido a errores de cómputo, en la práctica $\text{iff2}(F)$ resulta compleja con parte imaginaria muy pequeña, por lo que debe tomarse la parte real.

Filtrado en el Dominio Frecuencial

Vimos que en el dominio espacial el filtrado se obtiene convolucionando la imagen $f(x,y)$ con la máscara $h(x,y)$, es decir:

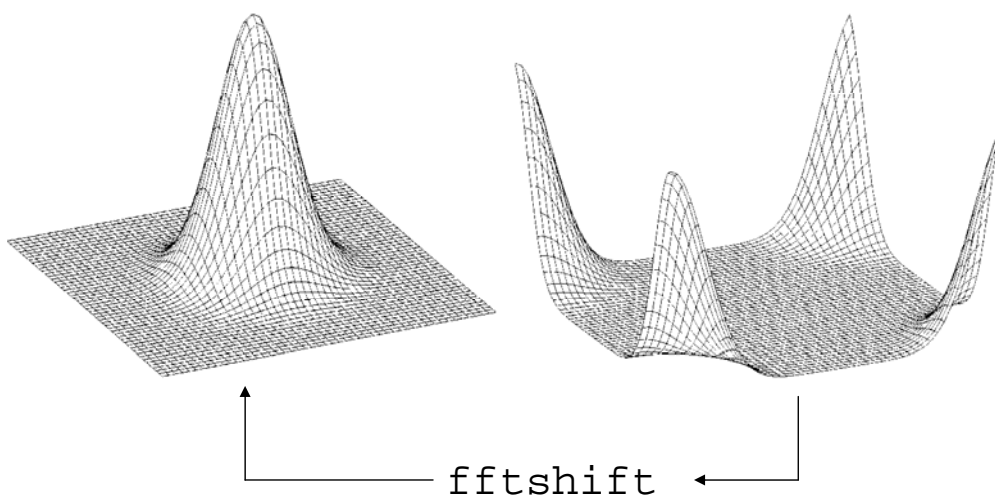
$$f_{filt}(x, y) = f(x, y) * h(x, y)$$

Por el Teorema de Convolución, en el dominio transformado resulta

$$F_{filt}(u, v) = F(u, v)H(u, v)$$

donde $H(u,v)$ se denomina **respuesta en frecuencia del filtro**.

Respuesta en frecuencia típica de un filtro pasabajo



a b

FIGURE 4.4 Transfer functions of (a) a centered lowpass filter, and (b) the format used for DFT filtering. Note that these are frequency domain filters.

Ejemplo de filtrado con y sin padding

Filtrado sin padding

```
>>[M,N]=size(f);
>>F=fft2(f);
>>sig=10;
>>H=lpfilter('gaussian',M,N,sig);
>>G=H.*F;
>>g=real(ifft2(G));
>>imshow(g,[])
```

Filtrado con padding

```
>>PQ=paddedsize(size(f));
>>Fp=fft2(f,PQ(1),PQ(2));
>>sig=10;
>>Hp=lpfilter('gaussian',PQ(1),PQ(2),2*sig);
>>Gp=Hp.*Fp;
>>gp=real(ifft2(Gp));
>>gpc=gp(1:size(f,1),1:size(f,2));
>>imshow(gpc,[])
```

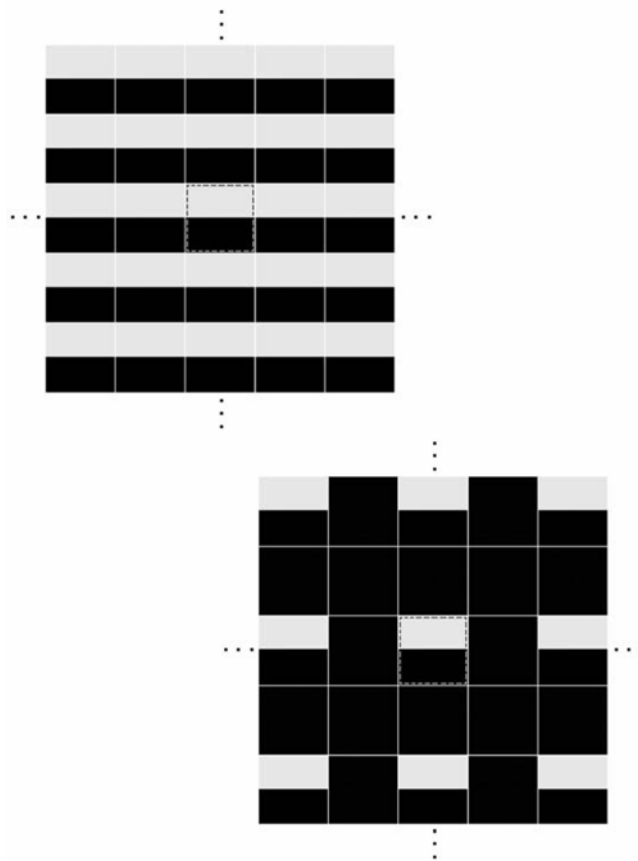
Proc. Digital de Imágenes

11



a b c

FIGURE 4.5 (a) A simple image of size 256×256 . (b) Image lowpass-filtered in the frequency domain without padding. (c) Image lowpass-filtered in the frequency domain with padding. Compare the light portion of the vertical edges in (b) and (c).



a
b

FIGURE 4.6
(a) Implied, infinite periodic sequence of the image in Fig. 4.5(a). The dashed region represents the data processed by `fft2`. (b) The same periodic sequence after padding with 0s. The thin white lines in both images are shown for convenience in viewing; they are not part of the data.



FIGURE 4.7 Full padded image resulting from `ifft2` after filtering. This image is of size 512×512 pixels.

Pasos básicos de filtrado en el dominio DFT

1. Calcular los parámetros de *padding*

```
>>PQ=paddedsize(size(f));
```

2. Computar la Transformada de Fourier con *padding*

```
>>F=fft2(f,PQ(1),PQ(2));
```

3. Generar la función de filtro H , de dimensiones $PQ(1) \times PQ(2)$

4. Multiplicar la transformada por el filtro

```
>>G=H.*F;
```

5. Obtener la parte real de la IFFT de G

```
>>g=real(iff2(G));
```

6. Recortar (*crop*) el rectángulo superior izquierdo

```
>>g=g(1:size(f,1),1:size(f,2));
```

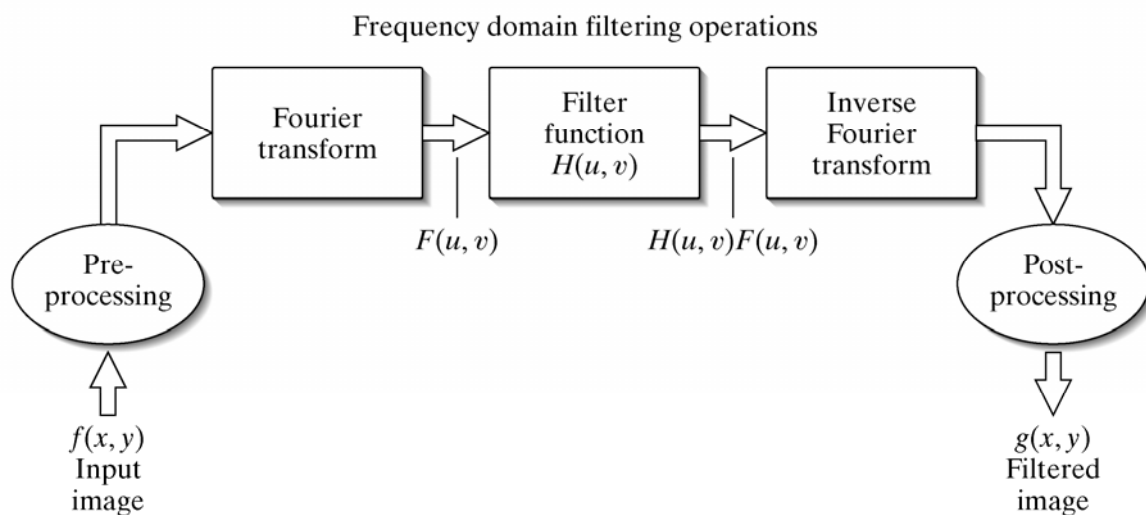


FIGURE 4.8
Basic steps for filtering in the frequency domain.

Obtención de filtros en el dominio DFT a partir de filtros espaciales

Función Matlab: `freqz2`

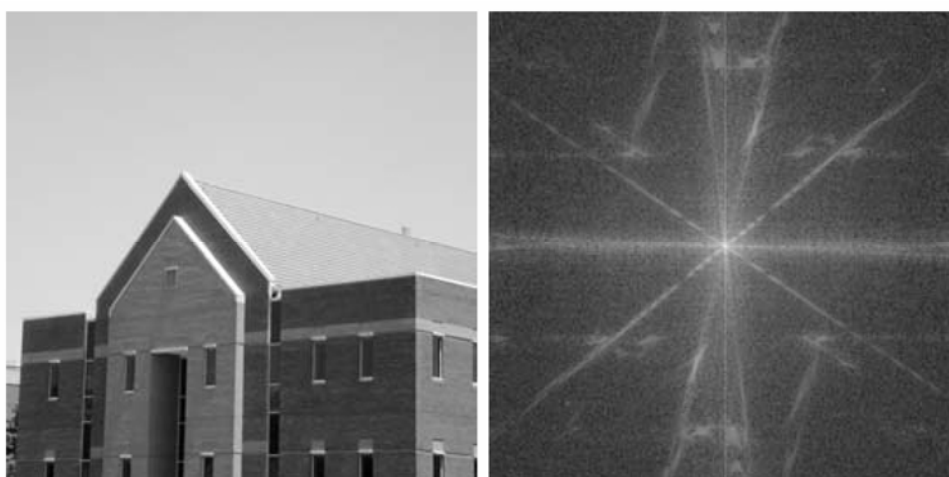
```
>>H=freqz2(h,R,C);
```

devuelve el filtro en el dominio frecuencial H correspondiente al filtro en el dominio espacial h , siendo R y C el número de filas y el número de columnas de H , respectivamente.

Ejemplo: Consideremos la imagen de Fig. 4.9(a). El espectro puede calcularse como

```
>>F=fft2(f);  
>>S=fftshift(log(1+abs(F)));  
>>S=gscale(S);  
>>imshow(S)
```

→ Fig. 4.9(b)



a b

FIGURE 4.9

(a) A gray-scale image. (b) Its Fourier spectrum.

Generemos ahora un filtro de Sobel con el comando `fspecial`

```
>>h=fspecial('sobel');
```

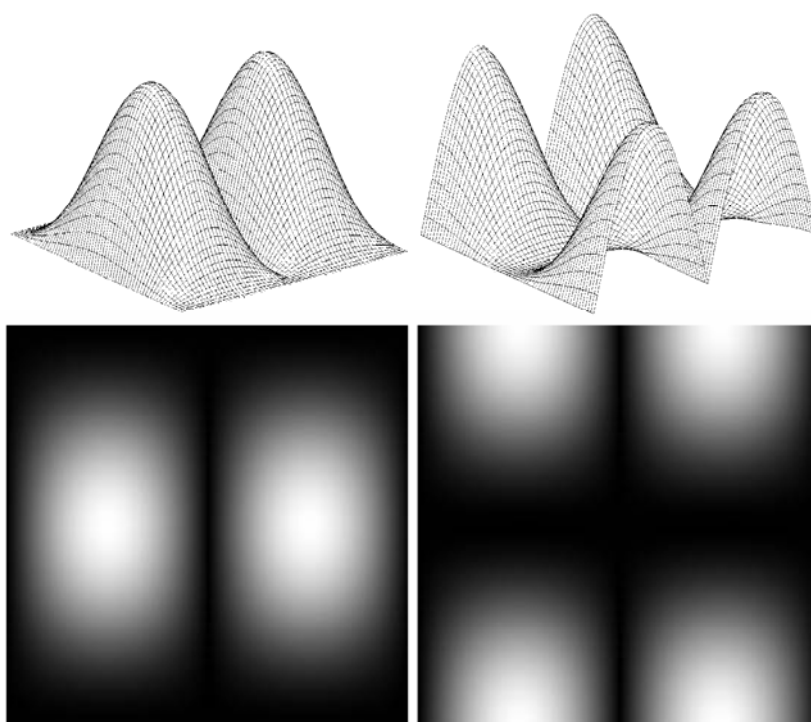
```
>>freqz2(h)
```

→ Fig. 4.10(a)

```

>>PQ=paddedsize(size(f));
>>H=freqz2(h,PQ(1),PQ(2));
>>H1=ifftshift(H)           → Fig. 4.10(b)
>>imshow(abs(H),[])         → Fig. 4.10(c)
>>figure, imshow(abs(H1),[]) → Fig. 4.10(d)

```



a b
c d

FIGURE 4.10
(a) Absolute value of the frequency domain filter corresponding to a vertical Sobel mask. (b) The same filter after processing with function `fftshift`. Figures (c) and (d) are the filters in (a) and (b) shown as images.

La imagen filtrada en el dominio espacial resulta:

```
>>gs=imfilter(double(f),h);
```

Por otra parte, la imagen filtrada en el dominio frecuencial resulta:

```
>>gf=dftfilt(f,H1);
```

```
>>figure, imshow(gs,[])
```

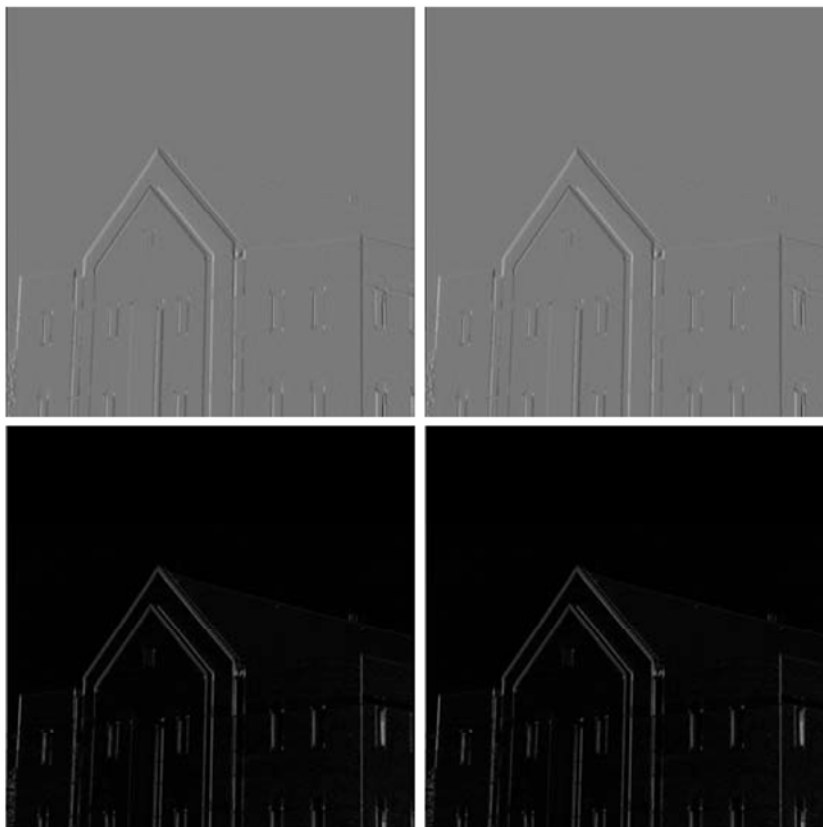
```
>>figure, imshow(gf,[])
```

```
>>figure, imshow(abs(gs),[])
```

```
>>figure, imshow(abs(gf),[])
```

```
>>figure, imshow(abs(gs)>0.2*abs(max(gs(:))))
```

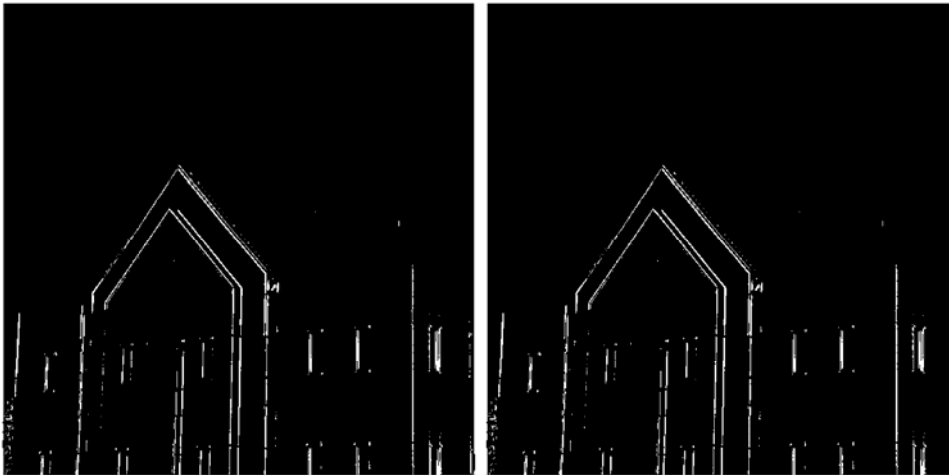
```
>>figure, imshow(abs(gf)>0.2*abs(max(gf(:))))
```



a b
c d

FIGURE 4.11

(a) Result of filtering Fig. 4.9(a) in the spatial domain with a vertical Sobel mask. (b) Result obtained in the frequency domain using the filter shown in Fig. 4.10(b). Figures (c) and (d) are the absolute values of (a) and (b), respectively.



a b

FIGURE 4.12
Thresholded versions of Figs. 4.11(c) and (d), respectively, to show the principal edges more clearly.

```
>>d=abs ( gs-gf ) ;
```

```
>>max ( max ( d ) )
```

```
>>min ( min ( d ) )
```



Diferencia mínima entre las imágenes

Generación de filtros directamente en el dominio DFT

Nos concentramos en filtros circularmente simétricos que se definen como distancias desde el punto en el dominio frecuencial (u,v) al origen de la transformada.

Filtros Pasa Bajo en el dominio frecuencial

Un **Filtro Pasa Bajo Ideal (ILPF)** está definido como

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

Donde $D(u,v)$ es la distancia del punto (u,v) al centro del filtro, y D_0 es una constante.(frecuencia de corte).

Un **Filtro Pasa Bajo de Butterworth (BLPF)** de orden n con un frecuencia de corte a una distancia D_0 del origen tiene una respuesta en frecuencia de la forma

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

A diferencia de un ILPF, la respuesta en frecuencia de un BLPF no tiene una discontinuidad en D_0 . La frecuencia de corte usualmente se define como los puntos donde $H(u, v) = 0.5$, y $D(u, v) = D_0$.

Un **Filtro Pasa Bajo Gaussiano (GLPF)** tiene una respuesta en frecuencia definida como

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

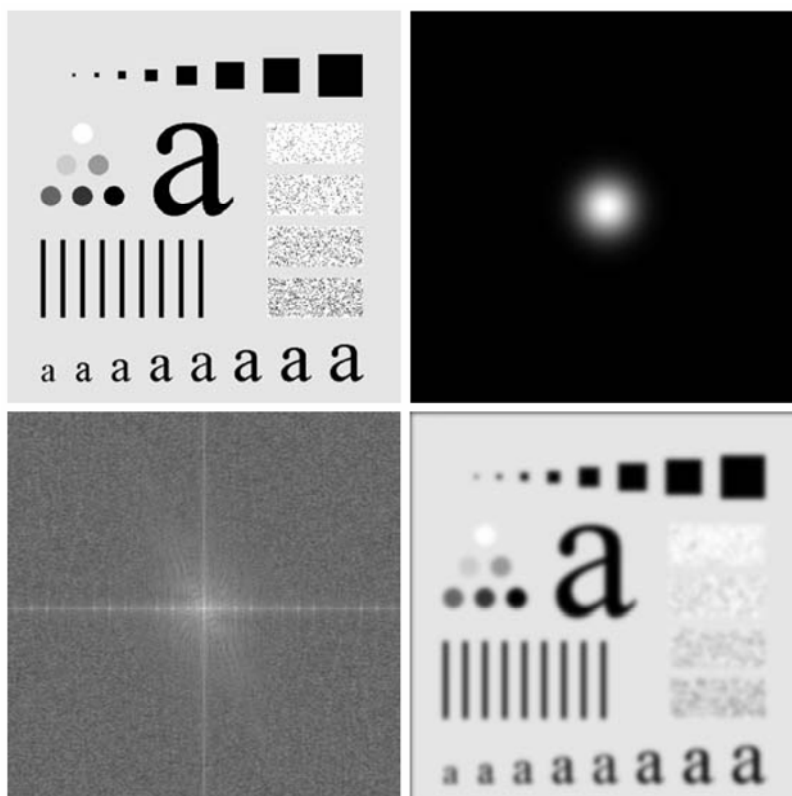
donde σ es el desvío estándar. Haciendo $\sigma = D_0$ se obtiene la expresión de la respuesta en frecuencia en función de la frecuencia de corte D_0

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

Cuando $D(u, v) = D_0$ la ganancia cae a 0,607.

Ejemplo: Aplicar un GLPF a la imagen 500 x 500. Se adopta D_0 igual al 5 % del ancho de la imagen con padding.

```
>> PQ = paddedsize(size(f));
>> [U,V] = dftuv(PQ(1),PQ(2));
>> D0 = 0.05*PQ(2);
>> F = fft2(f,PQ(1),PQ(2));
>> H = exp(-(U.^2+V.^2)/(2*(D0^2)));
>> g = dftfilt(f,H);
>> figure, imshow(fftshift(H),[])
>> figure, imshow(log(1+abs(fftshift(F))),[])
>> figure, imshow(g,[])
```

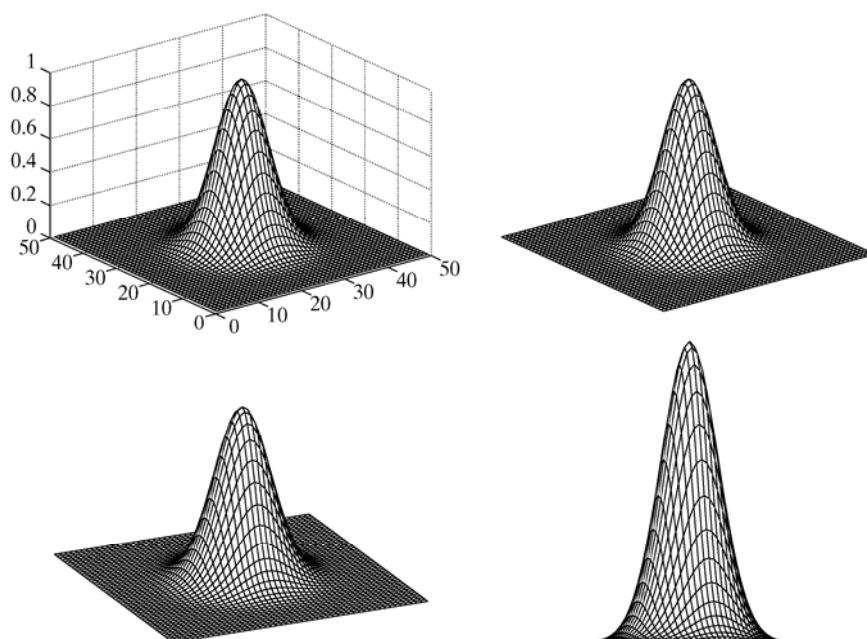


a b
c d

FIGURE 4.13 Lowpass filtering. (a) Original image. (b) Gaussian lowpass filter shown as an image. (c) Spectrum of (a). (d) Processed image.

Ejemplo: Ploteo de la respuesta en frecuencia de los filtros

```
>> H = fftshift(lpfilter('gaussian',500,500,50));  
>> mesh(H(1:10:500,1:10:500))  
>> axis([0 50 0 50 0 1])  
>> colormap([0 0 0])  
>> axis off  
>> grid off  
>> view(-25,30)  
>> view(-25, 0)
```



a b
c d

FIGURE 4.15
(a) A plot obtained using function mesh. (b) Axes and grid removed. (c) A different perspective view obtained using function view. (d) Another view obtained using the same function.

Filtros Pasa Alto en el dominio frecuencial

Dada la respuesta en frecuencia $H_{lp}(u, v)$ de un filtro pasa bajo, la función transferencia del filtro pasa alto complementario de obtiene

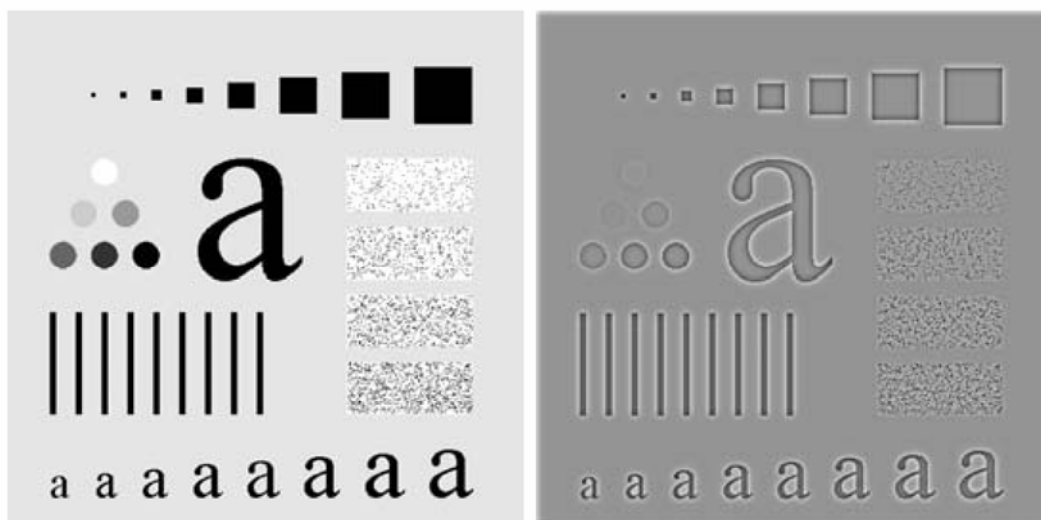
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Ejemplo:

```
>> PQ = paddedsize(size(f));  
>> D0 = 0.05*PQ(1);  
>> H = hpfilter('gaussian', PQ(1),PQ(2),D0);  
>> g = dftfilt(f,H);  
>> figure, imshow(g, [])
```

Proc. Digital de Imágenes

31



a b

FIGURE 4.18

(a) Original image.
(b) Result of Gaussian highpass filtering.

Filtros de Enfasis de Alta Frecuencia

La respuesta en frecuencia es de la forma

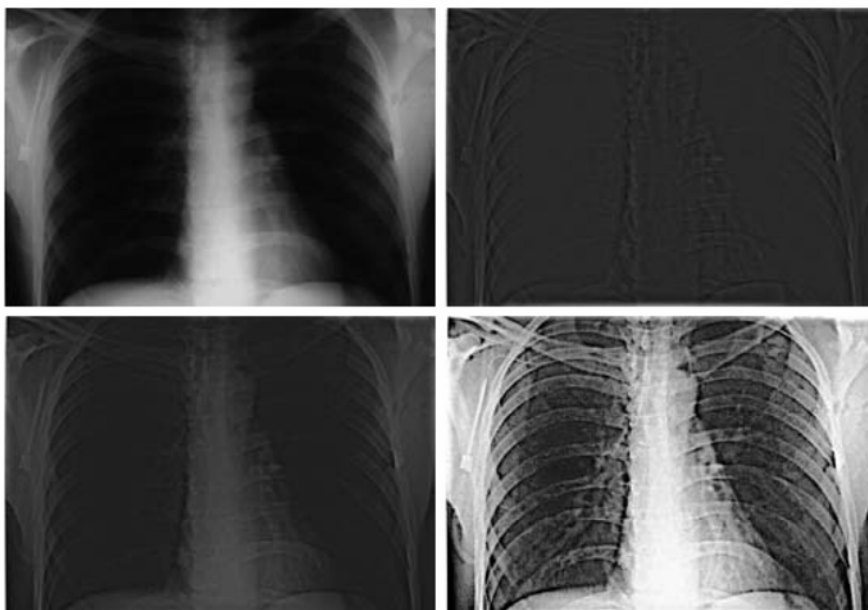
$$H_{hfe}(u, v) = a + bH_{hp}(u, v)$$

donde a es el offset, y b es un factor de escala.

Ejemplo: Imagen de radiografía de tórax

```
>> PQ = paddedsize(size(f));  
>> D0 = 0.05*PQ(1);  
>> HBW = hpfilter('btw',PQ(1),PQ(2),D0,1);  
>> H = 0.5 + 2*HBW;  
>> gbw = dftfilt(f, HBW);  
>> gbw = gscale(gbw);
```

```
>> ghf = dftfilt(f,H);  
>> ghf = gscale(ghf);  
>> ghe = histeq(ghf, 256);
```



a b
c d

FIGURE 4.19 High-frequency emphasis filtering. (a) Original image. (b) Highpass filtering result. (c) High-frequency emphasis result. (d) Image (c) after histogram equalization. (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)