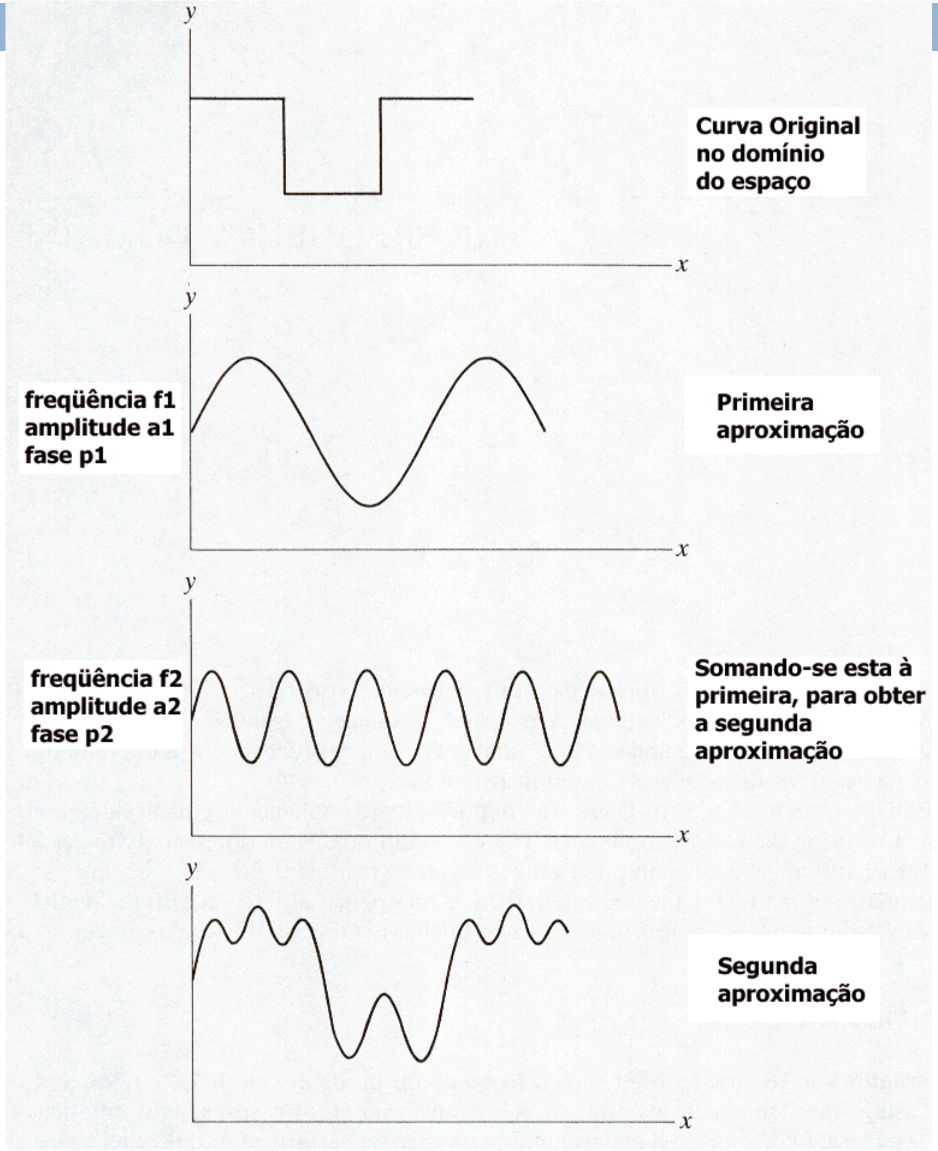
FILTRAGEM NO DOMÍNIO DA FREQUÊNCIA

Introdução

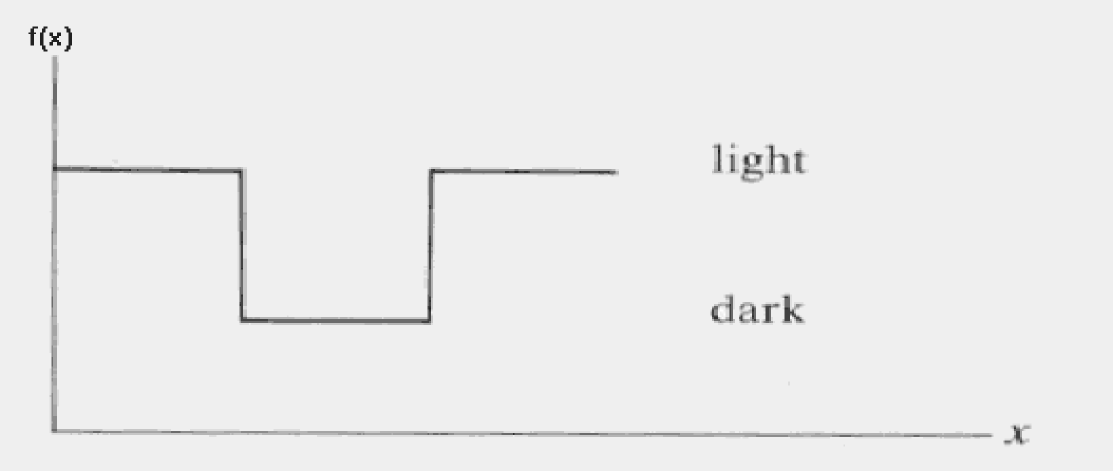
- Um sinal no domínio do espaço *(x,y)* pode ser aproximado através de uma soma de senos e cossenos com frequências *(f1, f2, f3, ...fn)* de amplitudes *(a1, a2, ... an)* e fases *(p1, p2, ...pn)*



Introdução

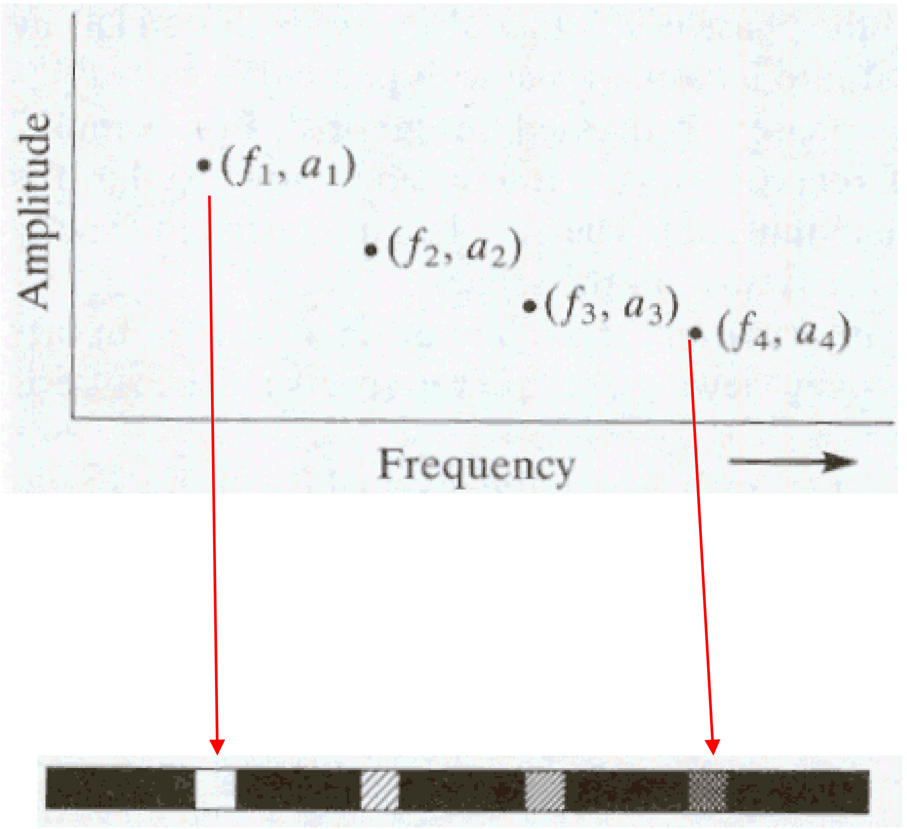
- Exemplo
  - Uma linha de uma imagem formada por uma sequência de pixels brancos e pretos

  - Pode ser representada no domínio do espaço como uma forma de onda



Introdução

- Exemplo
  - E no Domínio da Frequência pode ser representada por uma soma de senos e cossenos, através de suas frequências (f) e amplitudes (a)
  - Que podem ser colocadas no formato de uma Imagem como uma linha de amplitudes em escala de cinza



Introdução

- Exemplo
  - A imagem gerada através das amplitudes das frequências

  - É a Transformada no domínio da frequência da imagem original dada no domínio do espaço

  - Geralmente é possível aplicar sobre a imagem no domínio da frequência, uma Transformada Inversa, obtendo a imagem original

Transformada de Fourier

- Consiste em converter uma função em componentes senos e cossenos
  - Seja *f(x)* uma função contínua de uma variável real x, a Transformada de Fourier de *f(x)* é definida por

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\exp[-j2\pi ux]dx$$

  - A Transformada Inversa de Fourier é dada por

$$\Im^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u)\exp[j2\pi ux]du$$

  - Observe que estamos trabalhando com números complexos

Transformada de Fourier

- Usando-se a fórmula de Euler, o termo exponencial dentro da integral, pode ser colocado na forma

$$\exp[-j2\pi ux] = \cos(2\pi ux) - j\,\text{sen}(2\pi ux)$$

  - *F(u)* é uma soma infinita de senos e cossenos e que cada valor de (*u*) determina a frequência de seu correspondente par (seno-cosseno)
  - A variável (*u*) é denominada de *Variável de Frequência*
- A Transformada de Fourier de uma função *f(x)* real, é complexa:

$$F(u) = R(u) + jI(u)$$

Transformada de Fourier

- Propriedades
  - A Magnitude de *F(u)* é chamada de *Espectro de Fourier* de f(x)

$$|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2}$$

  - O *ângulo de fase* é dado por

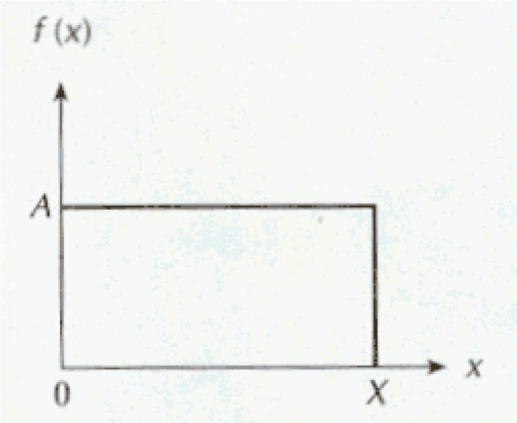
$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

  - O quadrado do espectro é chamado de *Espectro de Potência* de *f(x)* ou *Densidade Espectral*:

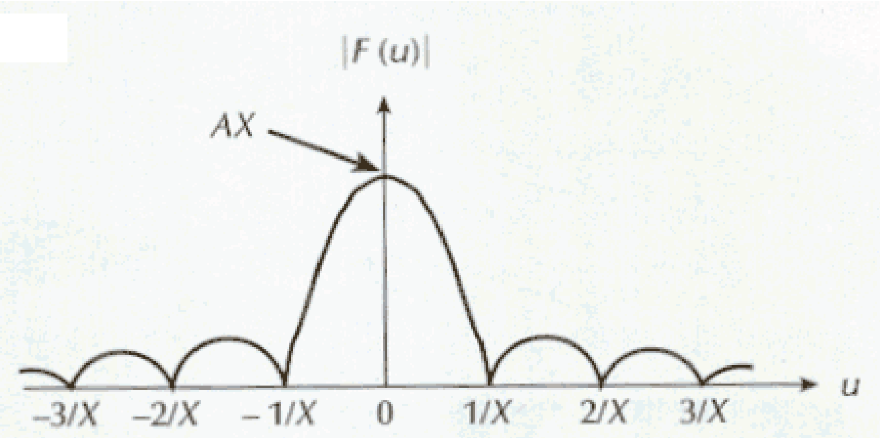
$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

Transformada de Fourier - Exemplo

- Função 1D *f*(*x*) no domínio do espaço



- Espectro de Fourier da função *f*(*x*)



Variando-se o valor de (*u*), obtém-se as infinitas amplitudes das frequências que constituem a função *f*(*x*).

Transformada de Fourier 2D

- Consiste em converter uma função bidimensional em componentes senos e cossenos
  - A Transformada de Fourier de uma função contínua *f(x,y)* é dada por

$$\Im\{f(x,y)\} = F(u,v) = \int\int_{-\infty}^{\infty} f(x,y)\exp[-j2\pi(ux+vy)dxdy$$

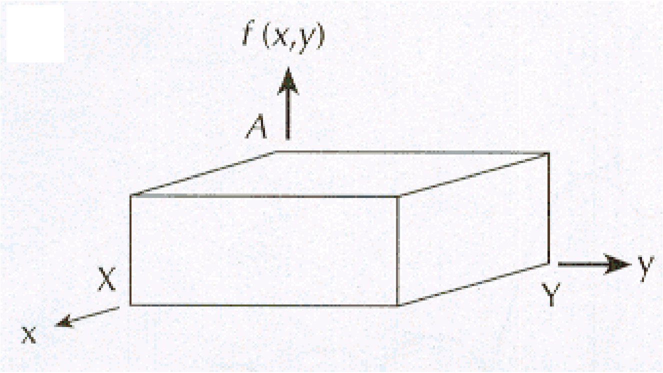
  - A Transformada Inversa é dada por

$$\Im\{F(u,v)\} = f(x,y) = \int\int_{-\infty}^{\infty} F(u,v)\exp[j2\pi(ux+vy)dudv$$

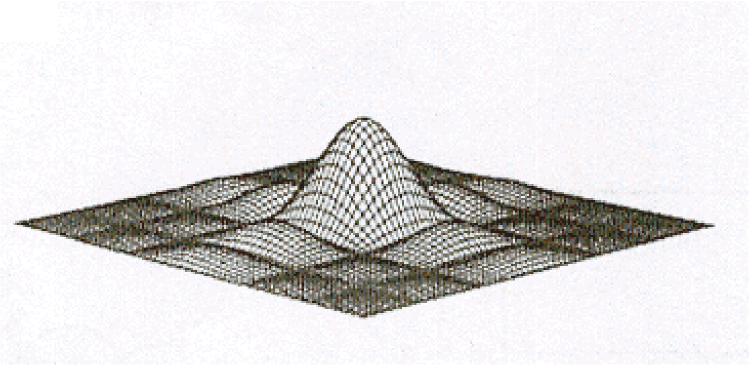
    - Sendo que (*u*) e (*v*) são as variáveis de frequência.

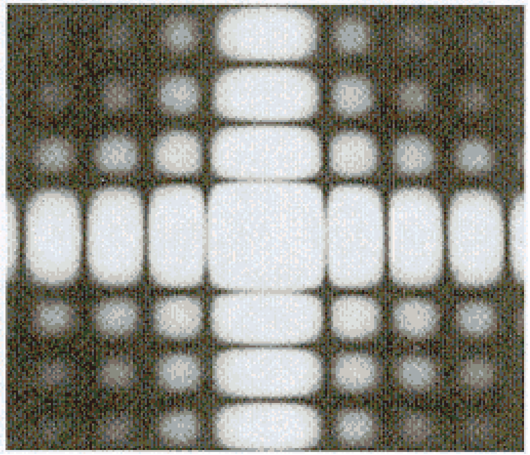
Transformada de Fourier 2D - Exemplo

- Função 1D *f(x,y)* no domínio do espaço



- Espectro de Fourier da função *f(x,y)*



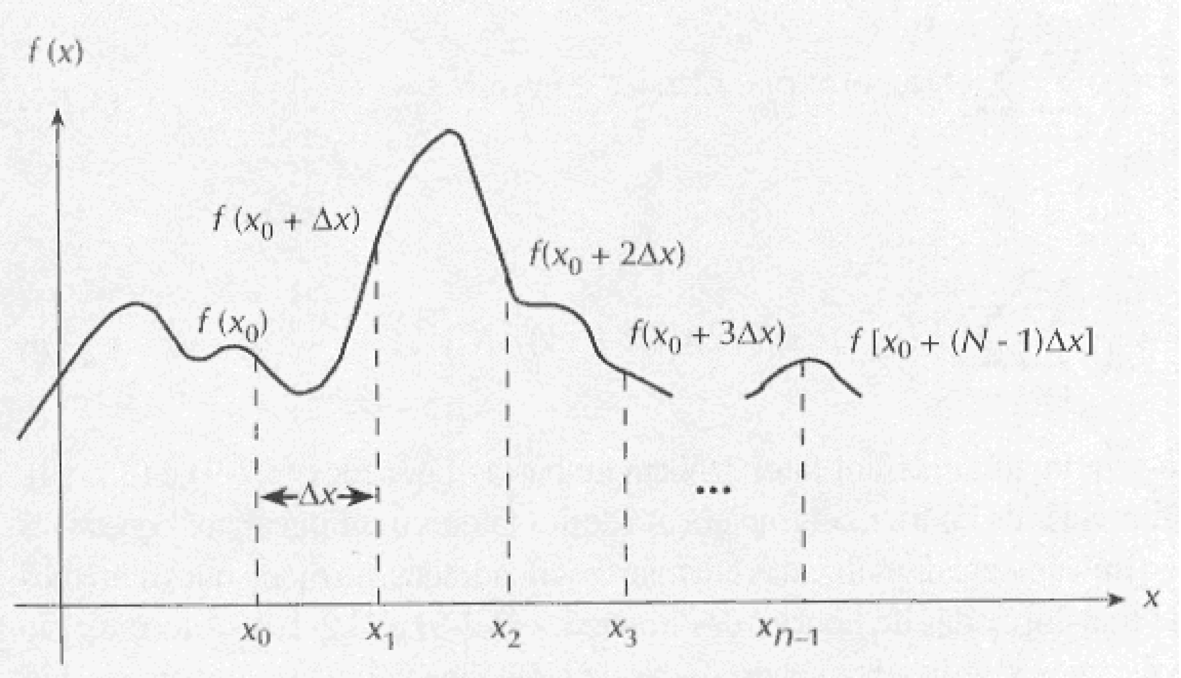


Transformada Discreta de Fourier - DFT

- Idéia
  - Discretizar uma função contínua *f(x)* numa sequência de *N* amostras separadas de unidades Δx

$$f(x) = f(x_0 + x\Delta x)$$

f(x) = {f(0), f(1), ..., f(N-1)}
Para *x = 0, 1, ..., N-1*



Transformada Discreta de Fourier - DFT

- O par de Transformadas Discretas de Fourier que se aplica a funções 1D amostradas é dado por
  - Transformada

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N]$$

$$\text{para} \quad u = 0,\ldots,N-1$$

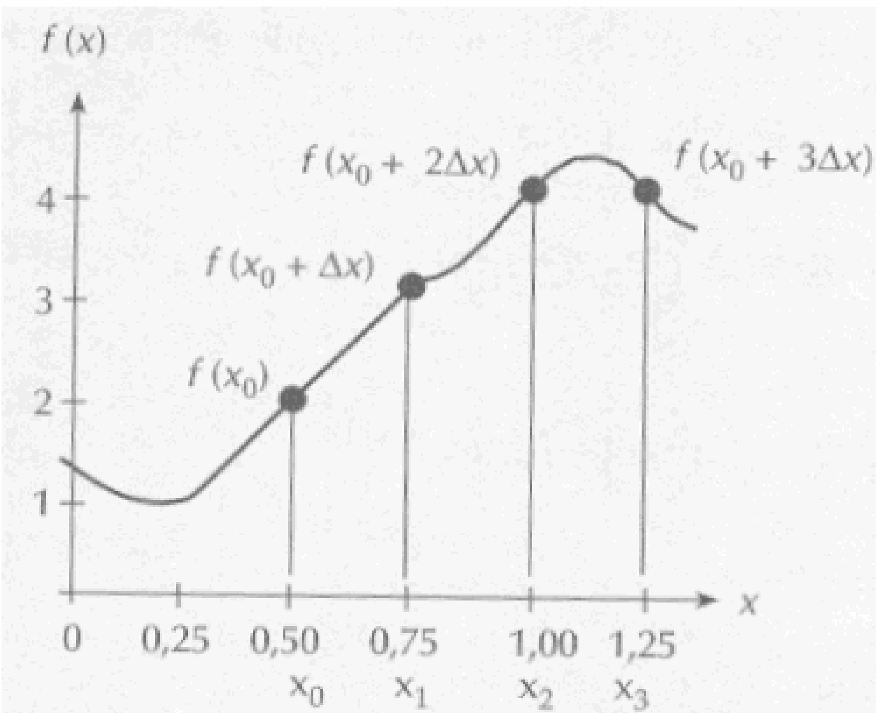
  - Inversa

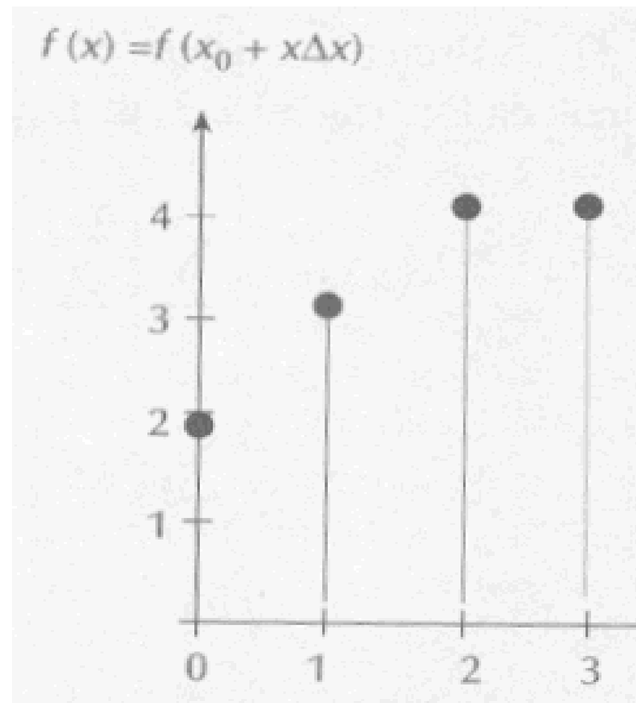
$$f(x) = \sum_{u=0}^{N-1} F(u)\exp[j2\pi ux/N]$$

$$\text{para} \quad x = 0,\ldots,N-1$$

Transformada Discreta de Fourier - DFT

- Exemplo: função amostrada





f(0) = 2
f(1) = 3
f(2) = 4
f(3) = 4

Transformada Discreta de Fourier - DFT

- Exemplo: cálculo da transformada

f(0) = 2
f(1) = 3
f(2) = 4
f(3) = 4

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N]$$

$$F(1) = \frac{1}{4}\sum_{x=0}^{3} f(x)\exp[-j2\pi x/4]$$

$$= \frac{1}{4}[2e^{0} + 3e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2}]$$

$$= \frac{1}{4}\left[2 + 3(\cos\frac{\pi}{2} - j\,\mathrm{sen}\frac{\pi}{2}) + 4(\cos\pi - j\,\mathrm{sen}\,\pi) + 4(\cos\frac{3\pi}{2} - j\,\mathrm{sen}\frac{3\pi}{2})\right]$$

$$= \frac{1}{4}[2 + 3(0 - j) + 4(-1 - j0) + 4(0 - j(-1))]$$

$$= \frac{1}{4}[2 - 3j - 4 + 4j]$$

$$= \frac{1}{4}(-2 + j)$$

$$F(0) = \frac{1}{4}\sum_{x=0}^{3} f(x)\exp[0]$$

$$= \frac{1}{4}[f(0) + f(1) + f(2) + f(3)]$$

$$= \frac{1}{4}[2 + 3 + 4 + 4] = 3{,}25$$

$$F(2) = -\frac{1}{4}$$

$$F(3) = -\frac{1}{4}[2 + j]$$

Transformada Discreta de Fourier - DFT

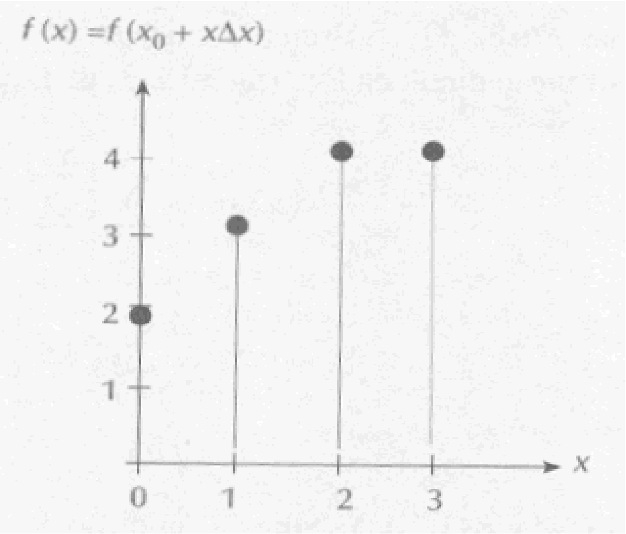
- Magnitude do espectro de Fourier

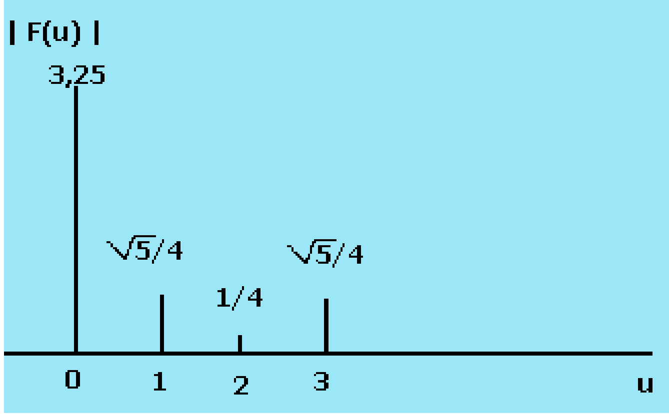
$$|F(0)| = 3{,}25$$

$$|F(1)| = \left[\left(\frac{2}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right]^{1/2} = \frac{\sqrt{5}}{4}$$

$$|F(2)| = \left[\left(\frac{1}{4}\right)^2\right]^{1/2} = \frac{1}{4}$$

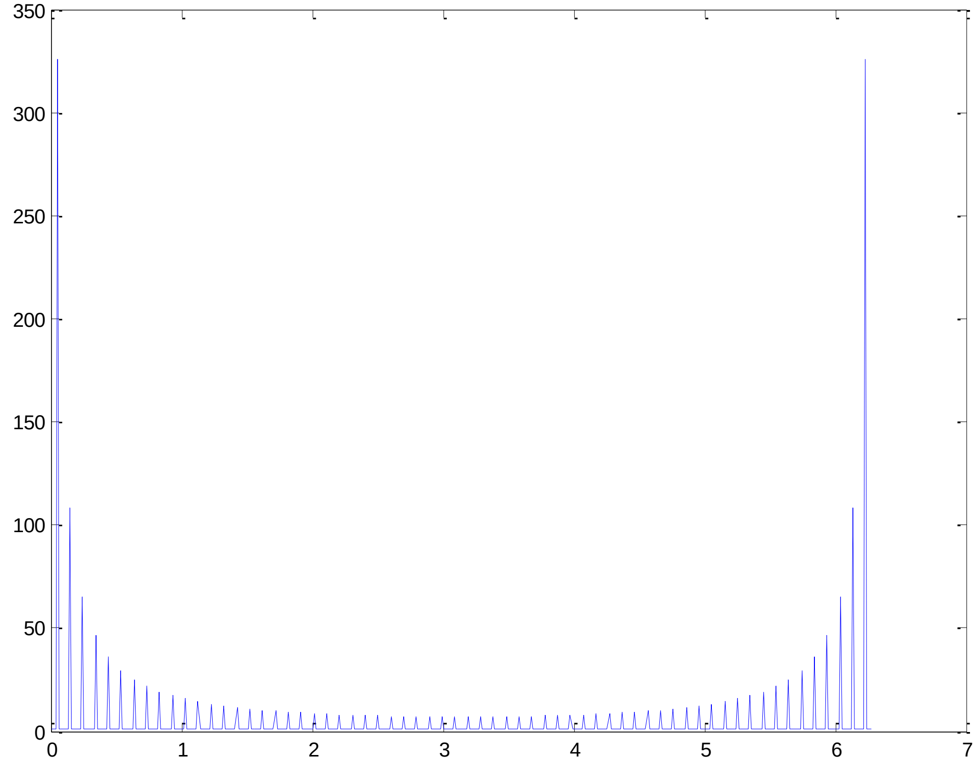
$$|F(3)| = \left[\left(\frac{2}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right]^{1/2} = \frac{\sqrt{5}}{4}$$





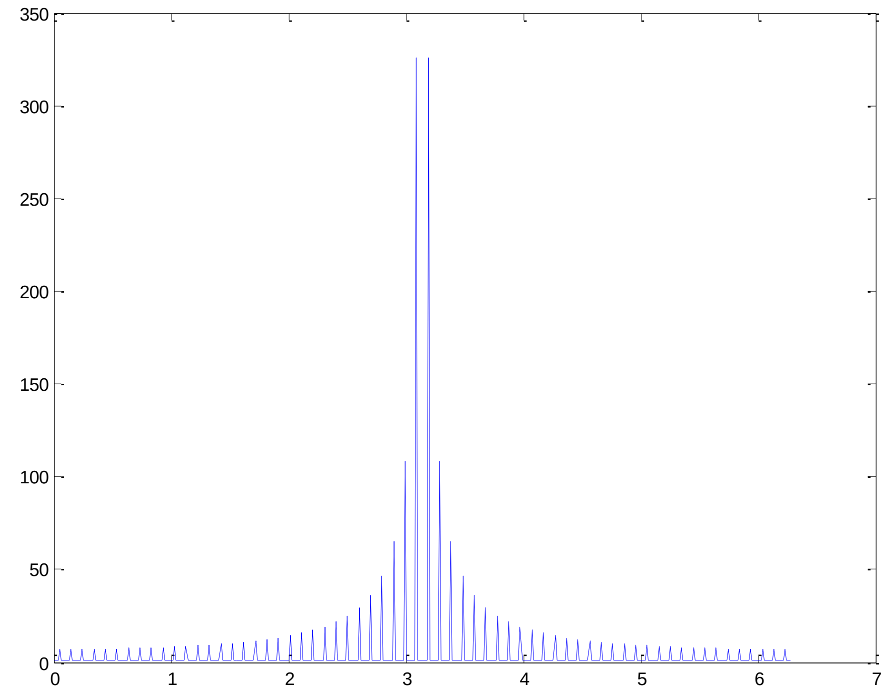
Transformada Discreta de Fourier - DFT

- Quando realizado a transformada discreta de uma onda quadrada obtemos um deslocamento



Transformada Discreta de Fourier - DFT

- A Transformada de Fourier é centralizada na origem, mas a Transformada **Discreta** de Fourier é centralizada em $N/2$
  - É necessário realizar um deslocamento para corrigir o resultado
    - Matlab: função **fftshift()**



Transformada Rápida de Fourier - FFT

- Problema com a DFT
  - O número de multiplicações e adições complexas necessárias para implementar a DFT-1D é proporcional a N^2

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N]$$

  - A decomposição adequada desta equação pode tornar o número de multiplicações e adições proporcional a $N\log_2 N$
    - Este procedimento é chamado de *Transformada Rápida de Fourier* (FFT)

Transformada Rápida de Fourier - FFT

- Comparação DFT x FFT
  - Muitas aplicações de processamento de sinais (ou imagens) em tempo real seriam impraticáveis utilizando a DFT
  - Vários algoritmos para FFT
    - Para alguns, somente podem ser considerados amostras onde *N* é uma potência de 2

N	N^2 (DFT)	$N\log_2 N$ (FFT)	Vantagem Computacional
2	4	2	2,00
4	16	8	2,00
8	64	24	2,67
16	256	64	4,00
32	1024	160	6,40
64	4096	384	10,67
128	16384	896	18,29
256	65536	2048	32,00
512	262144	4608	56,89
1024	1048576	10240	102,40
2048	4194304	22528	186,18
4096	16777216	49152	341,33
8192	67108864	106496	630,15

Transformada Discreta de Fourier 2D: DFT-2D

- O par de Transformadas Discretas de Fourier que se aplica a funções 2D amostradas é dado por
  - Transformada

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(\frac{ux}{M} + \frac{vy}{N})]$$

  - Inversa

$$\text{para } u = 0,\ldots, M-1 \text{ e } v = 0,\ldots, N-1$$

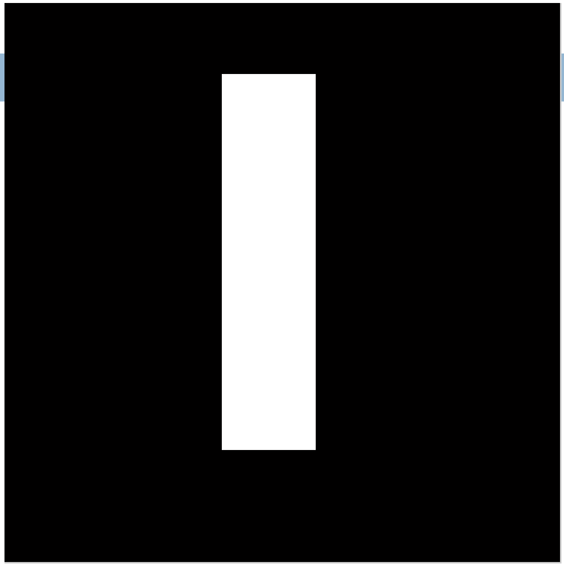
$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp[j2\pi(\frac{ux}{M} + \frac{vy}{N})]$$

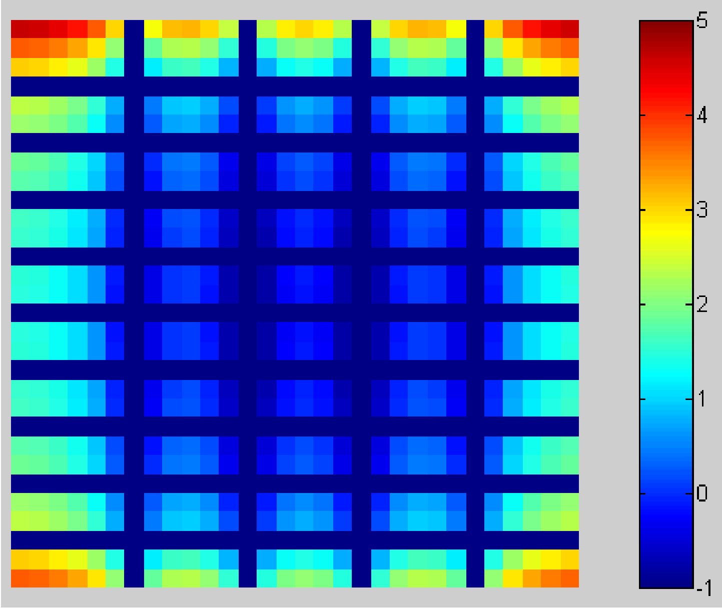
$$\text{para } x = 0,\ldots, M-1 \text{ e } y = 0,\ldots, N-1$$

Transformada Discreta de Fourier 2D: DFT-2D

- Exemplo (Matlab)

```
f = zeros(30,30);
f(5:24,13:17)=1;
imshow(f);

F =fft2(f);
F2 = log(abs(F));
imshow(F2,[-1, 5]);
colormap(jet); colorbar;
```

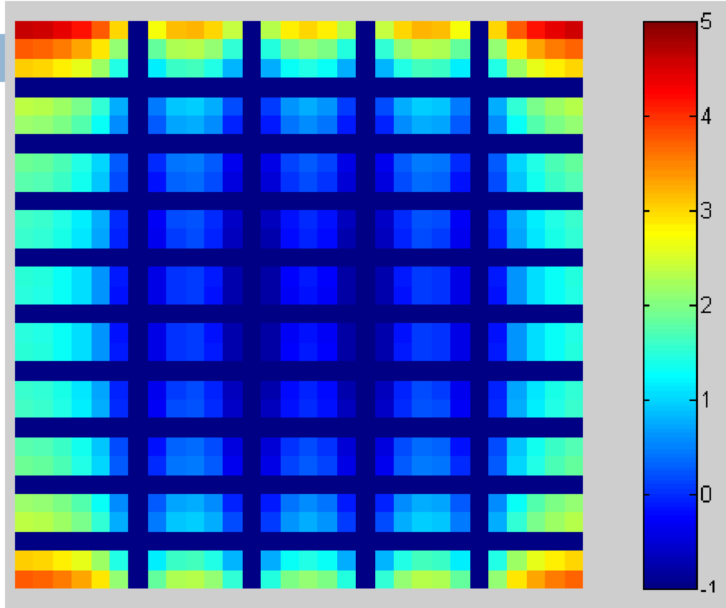


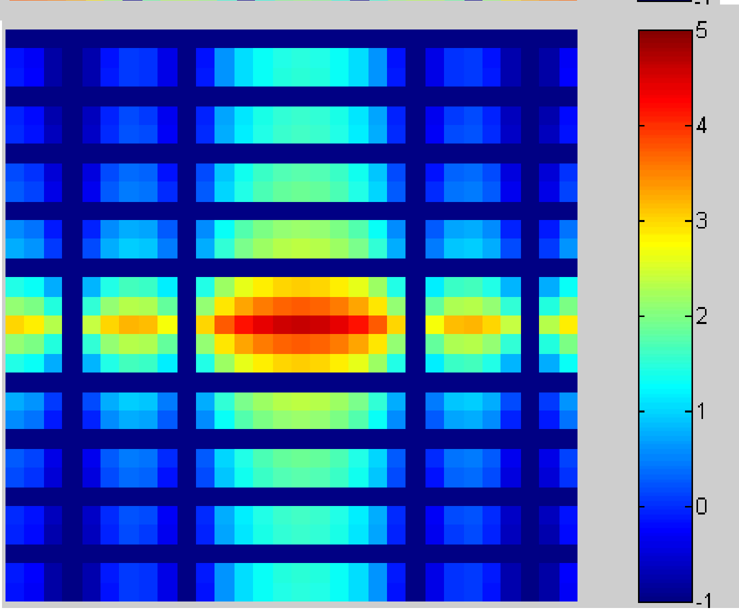


Transformada Discreta de Fourier 2D: DFT-2D

- Exemplo (Matlab)

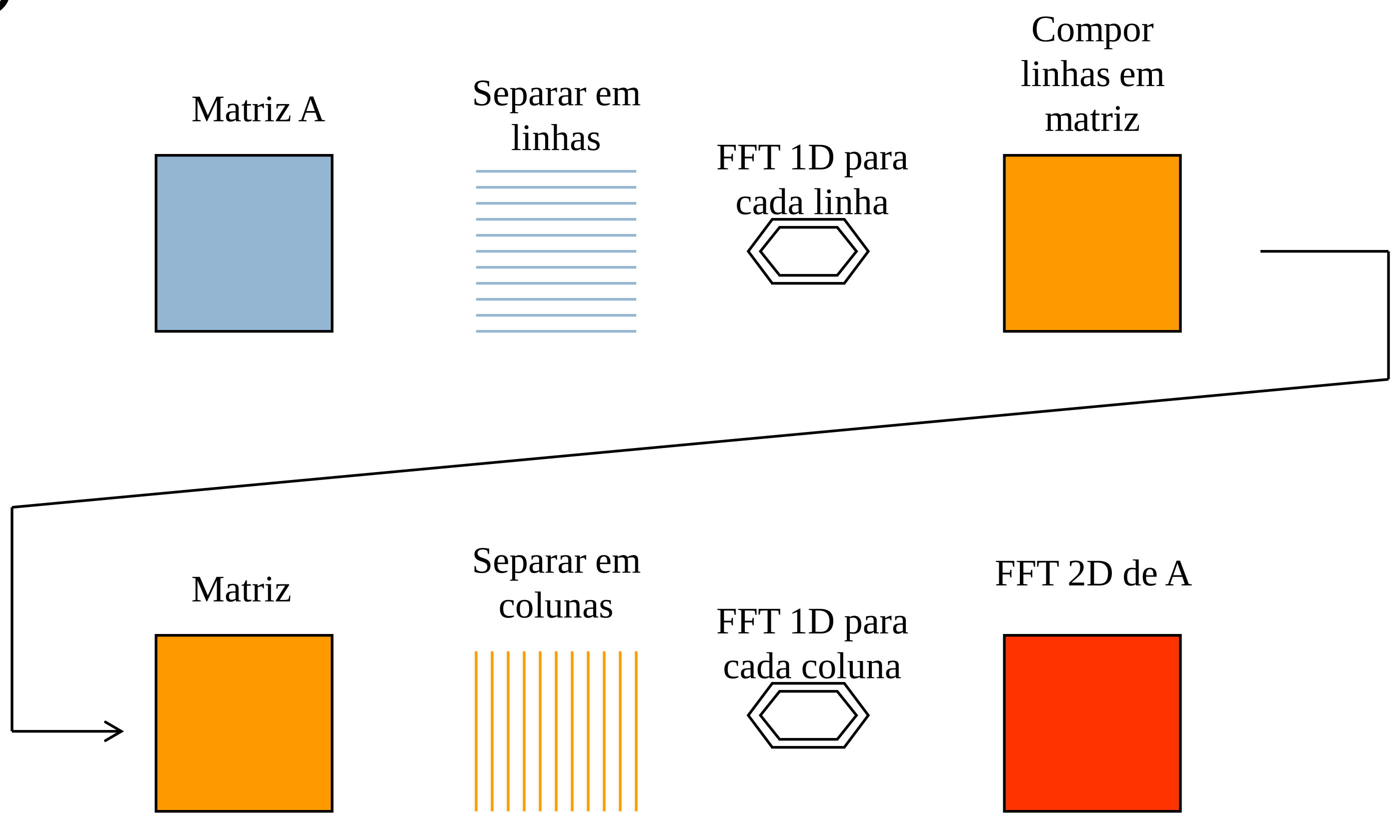
```
F2=fftshift(F);
F3=log(abs(F2));

imshow(F3,[-1, 5]);
colormap(jet); colorbar;
```





Separabilidade

- Permite calcular *F(u,v)* e *f(x,y)* em dois passos por aplicações sucessivas da Transformada de Fourier 1D



Filtragem no domínio da frequência

- Teorema da convolução
  - A convolução de uma máscara na imagem no espaço equivale no espectro a multiplicação da transformada da imagem pela transformada máscara

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

$$f(x,y)g(x,y) \Leftrightarrow F(u,v) * G(u,v)$$

Convolution in Space

Multiplication in Frequency

Filtragem no domínio da frequência

- Passo a passo do processo de filtragem

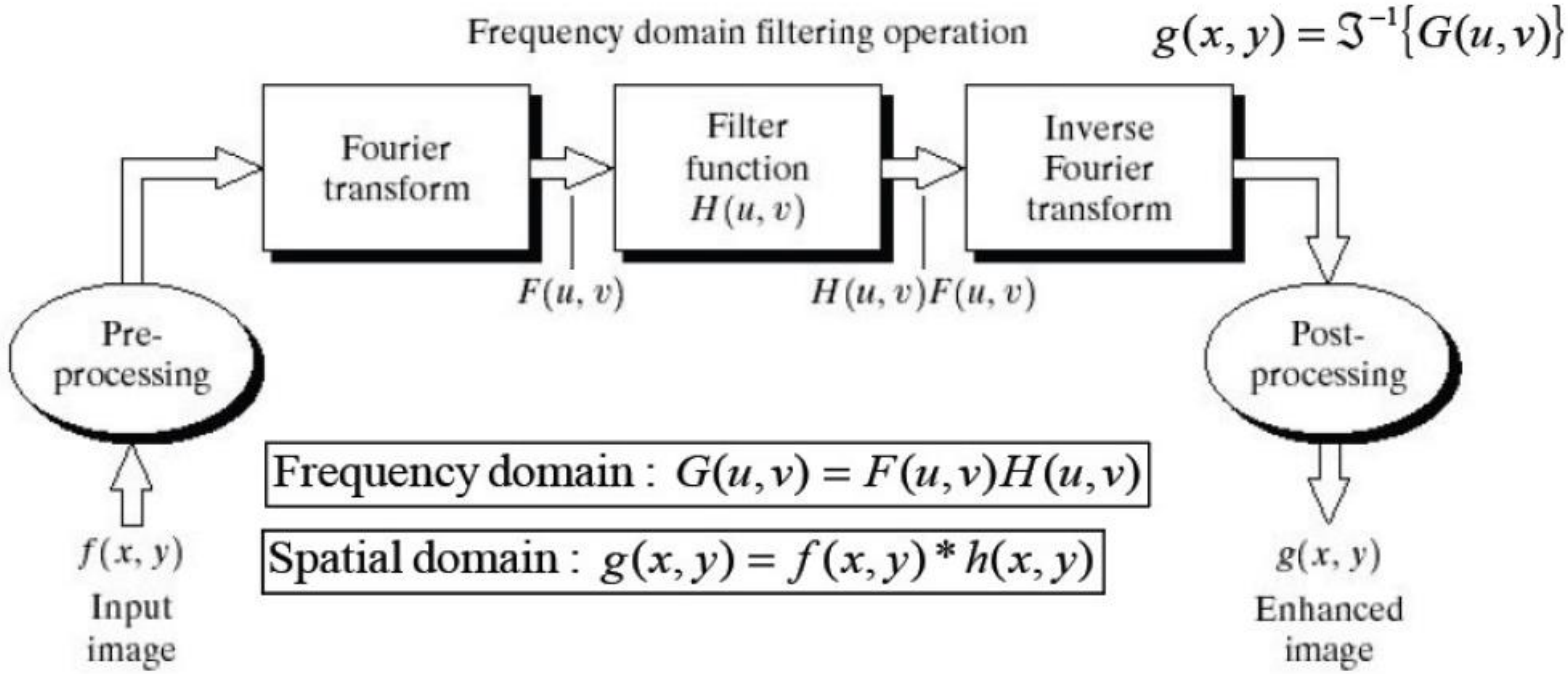
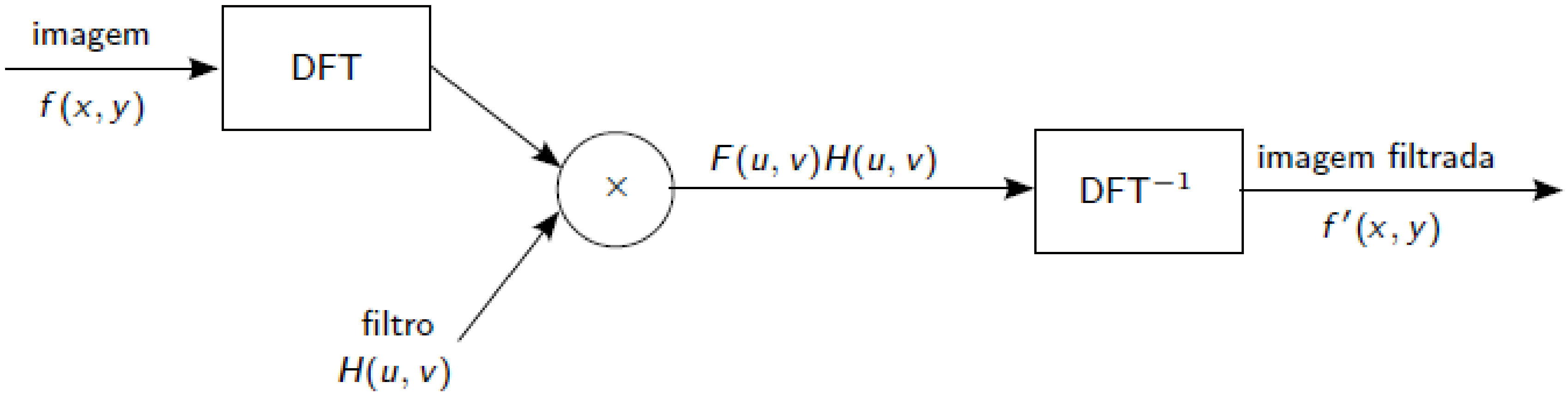


FIGURE 4.5 Basic steps for filtering in the frequency domain.

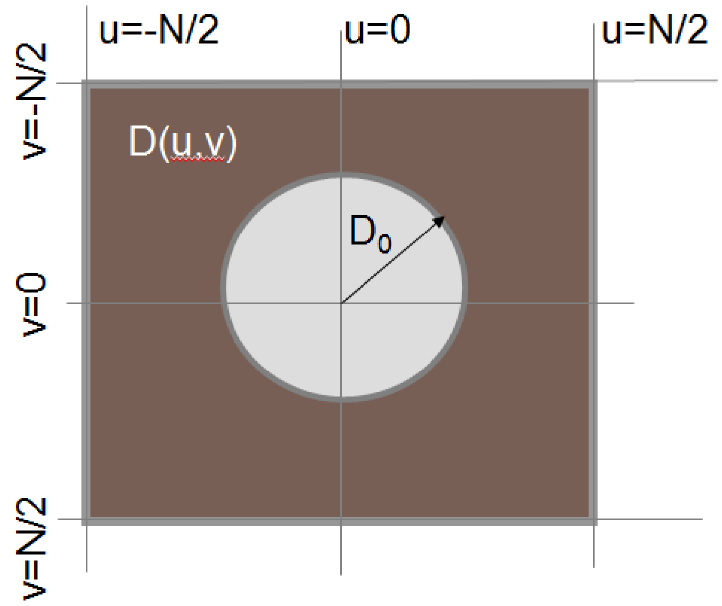
Filtragem no domínio da frequência

- Passo a passo do processo de filtragem



Filtragem no domínio da frequência

- Considerações importantes
  - Baixas frequências: localizam-se próximas do centro da imagem
    - Mudanças suaves nas intensidades da imagem
  - Altas frequências: localizam-se afastadas do centro da imagem
    - Bordas de objetos, mudanças bruscas nas intensidades

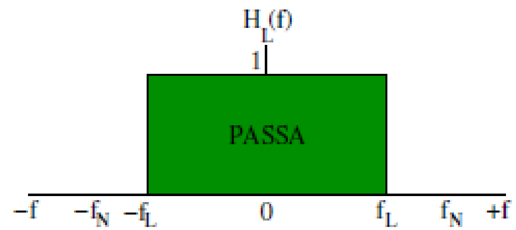


Filtragem no domínio da frequência

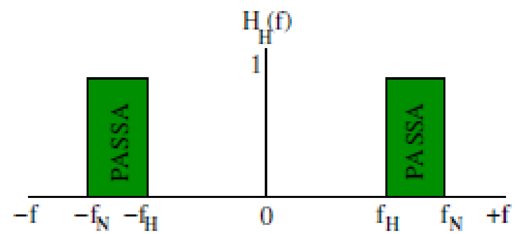
- Dado a posição das frequências no espectro, podemos criar filtros do tipo
  - Passa-baixa: deixa passar as baixas frequências da imagem (suavização)
  - Passa-alta: deixa passar as altas frequências da imagem (realce)

Filtragem no domínio da frequência

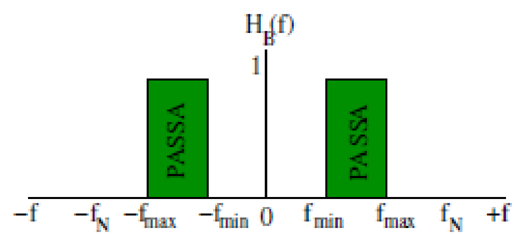
passa-baixa: $H_L(f) = \begin{cases} 1, & \text{para } -f_L \leq f \leq +f_L \\ 0, & \text{caso contrário} \end{cases}$



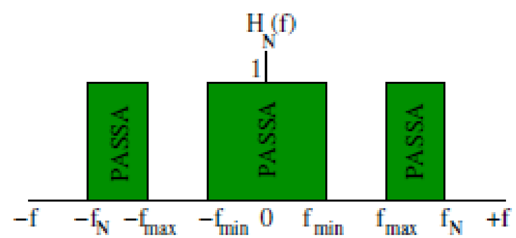
passa-alta: $H_H(f) = \begin{cases} 1, & \text{para } f_H \leq |f| \\ 0, & -f_H < f < +f_H \end{cases}$



passa-faixa: $H_B(f) = \begin{cases} 1, & \text{para } f_{\min} \leq |f| \leq f_{\max} \\ 0, & \text{caso contrário} \end{cases}$



rejeita-faixa: $H_N(f) = \begin{cases} 0, & \text{para } f_{\min} \leq |f| \leq f_{\max} \\ 1, & \text{caso contrário} \end{cases}$



Filtragem no domínio da frequência

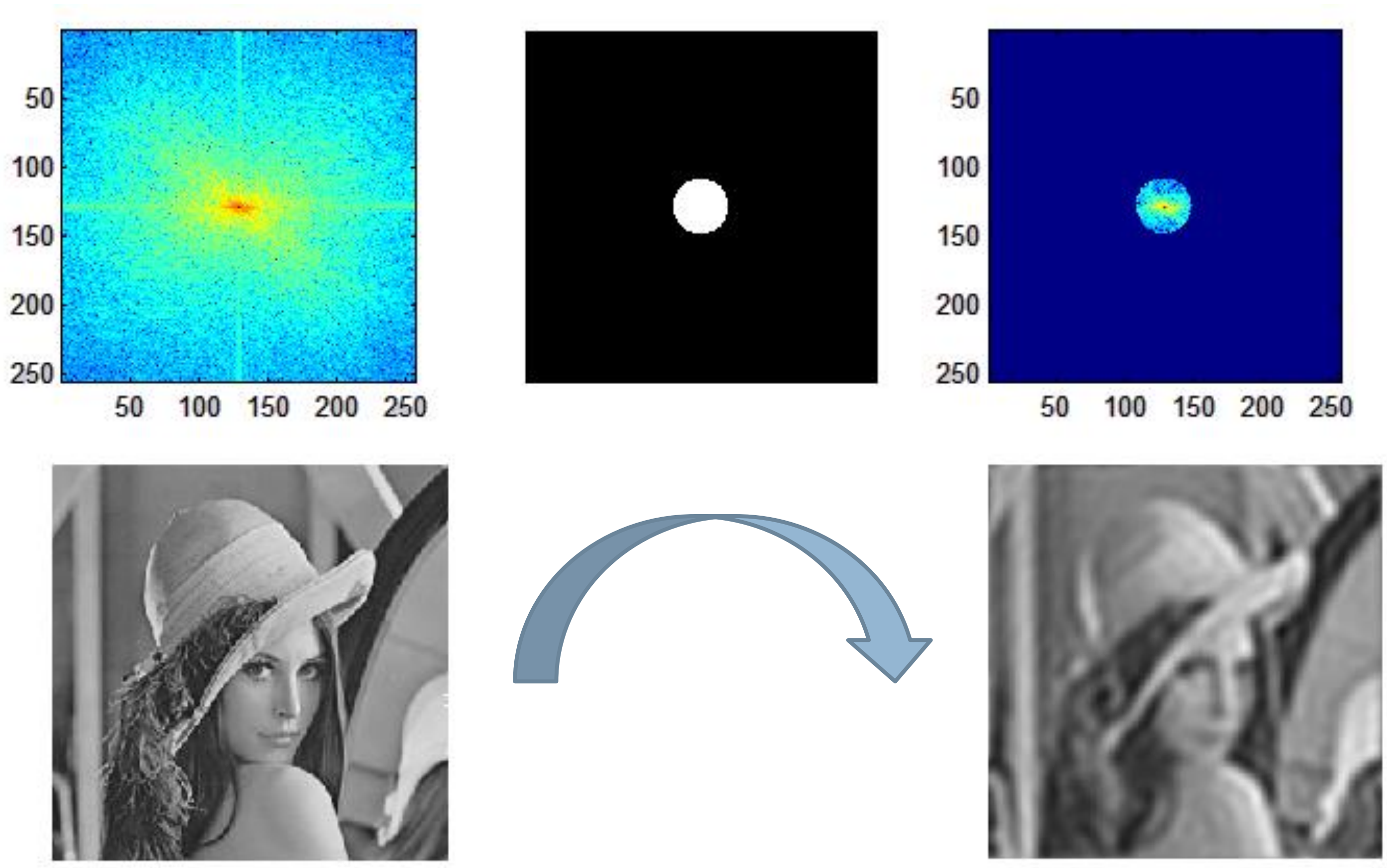
- Máscaras dos filtros ideais
  - No caso ideal teríamos filtros passa-baixas e passa-altas, respectivamente

$$L(u,v) = \left\{ \begin{array}{ll} 1, & \text{se } D(u,v) \leq D_l \\ 0, & \text{no c.c.} \end{array} \right\}$$

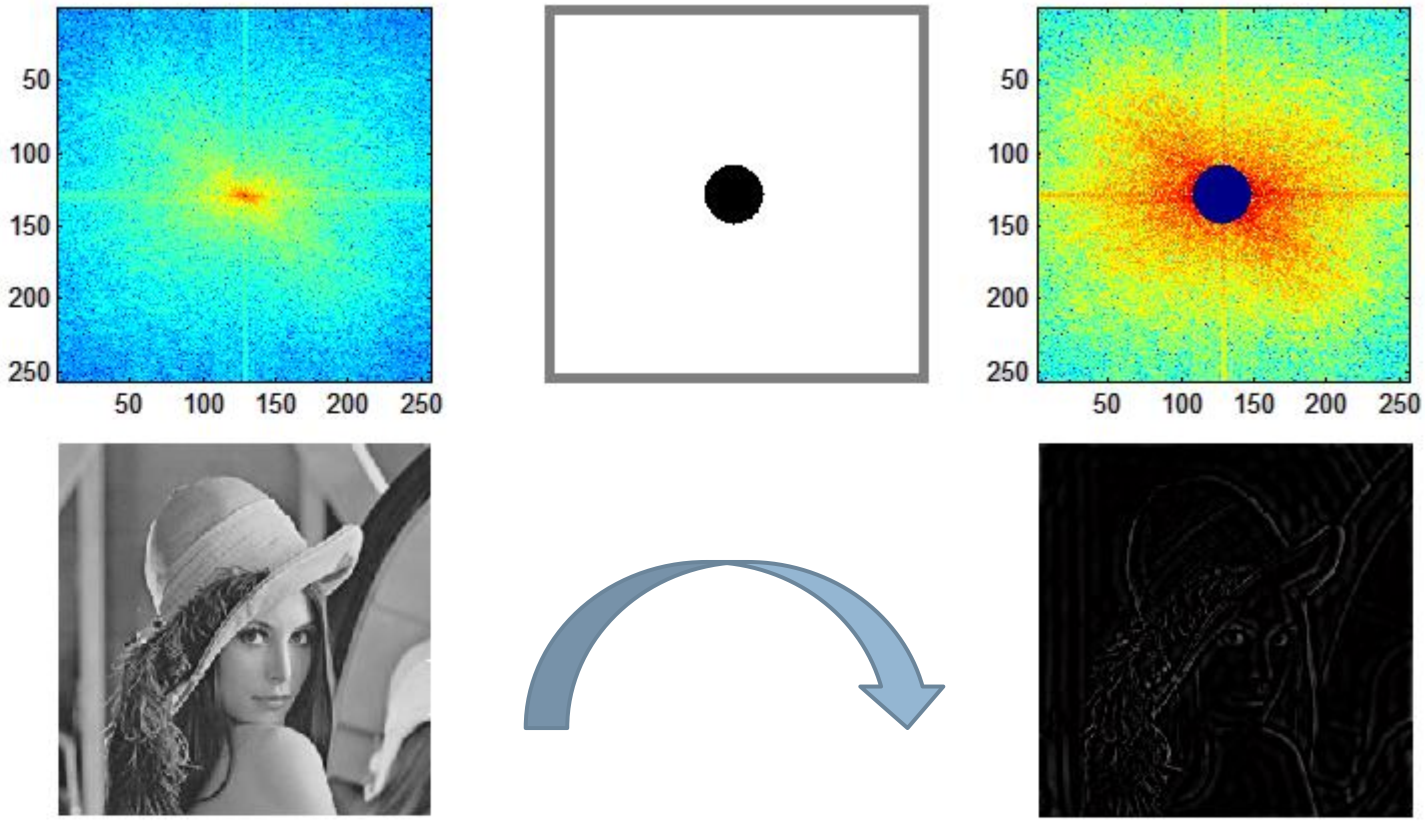
$$H(u,v) = \left\{ \begin{array}{ll} 1, & \text{se } D(u,v) \geq D_h \\ 0, & \text{no c.c.} \end{array} \right\}$$

  - $D_l > 0$ e $D_h > 0$ definem as **frequências de corte**
  - $D(u,v) = (u^2 + v^2)^{1/2}$

Filtro passa-baixa

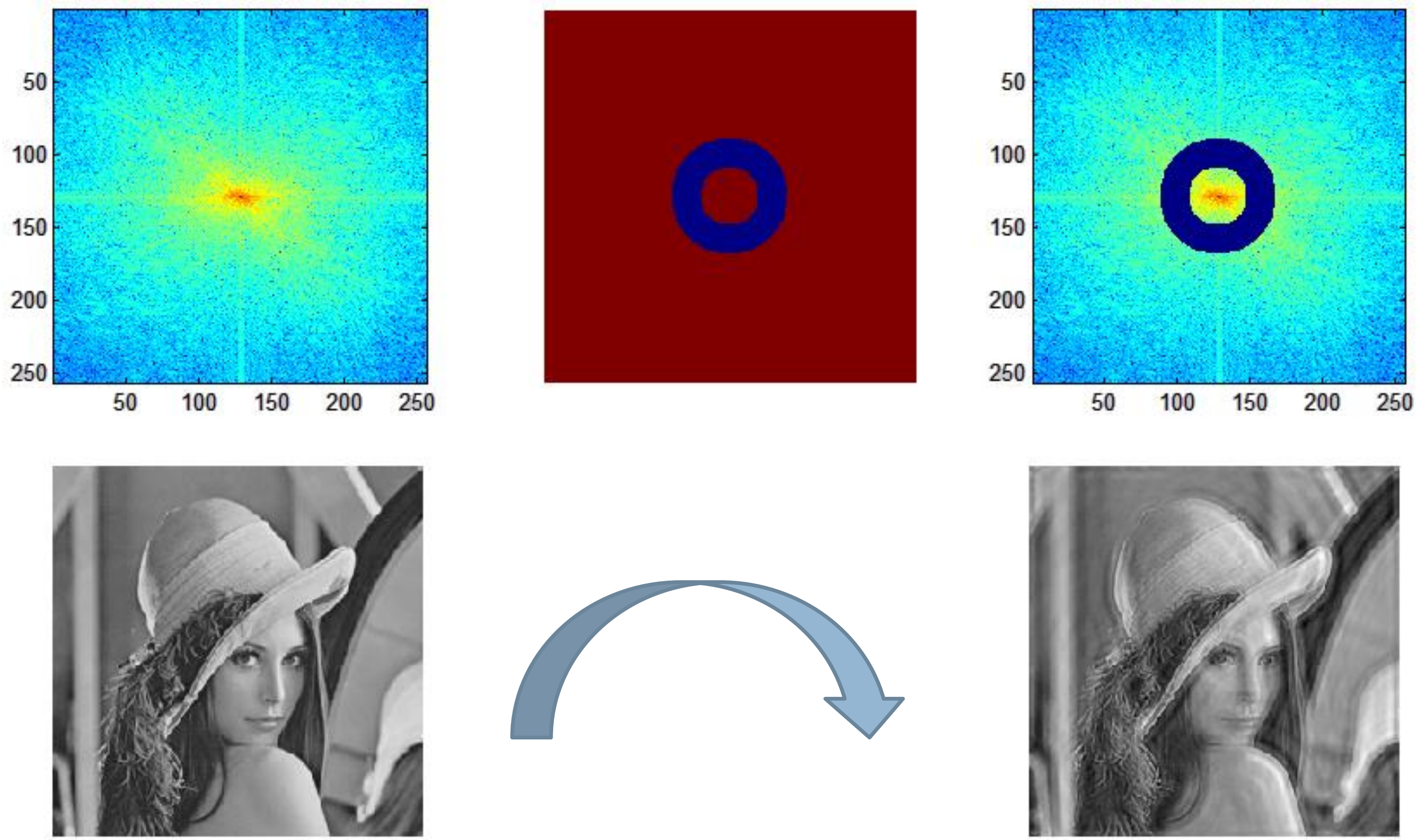


Filtro passa-alta



Filtro passa/rejeita-faixa

- Atual apenas numa faixa de frequências



Efeito oscilatório (*Ringing Problem*)

- Os filtros ideais possuem uma variação abrupta de valor na frequência resulta
  - Surgimento do efeito *ringing* (falsas bordas) no domínio espacial

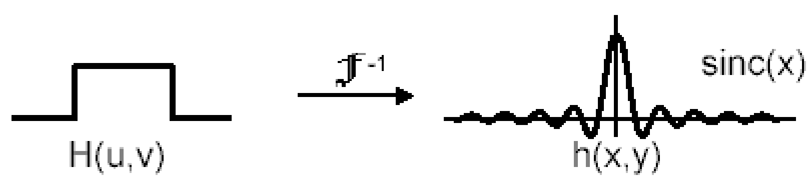
Efeito oscilatório (*Ringing Problem*)

The Ringing Problem

$$G(u,v) = F(u,v) \cdot H(u,v)$$

$$\Big\downarrow \text{ Convolution Theorm}$$

$$g(x,y) = f(x,y) * h(x,y)$$



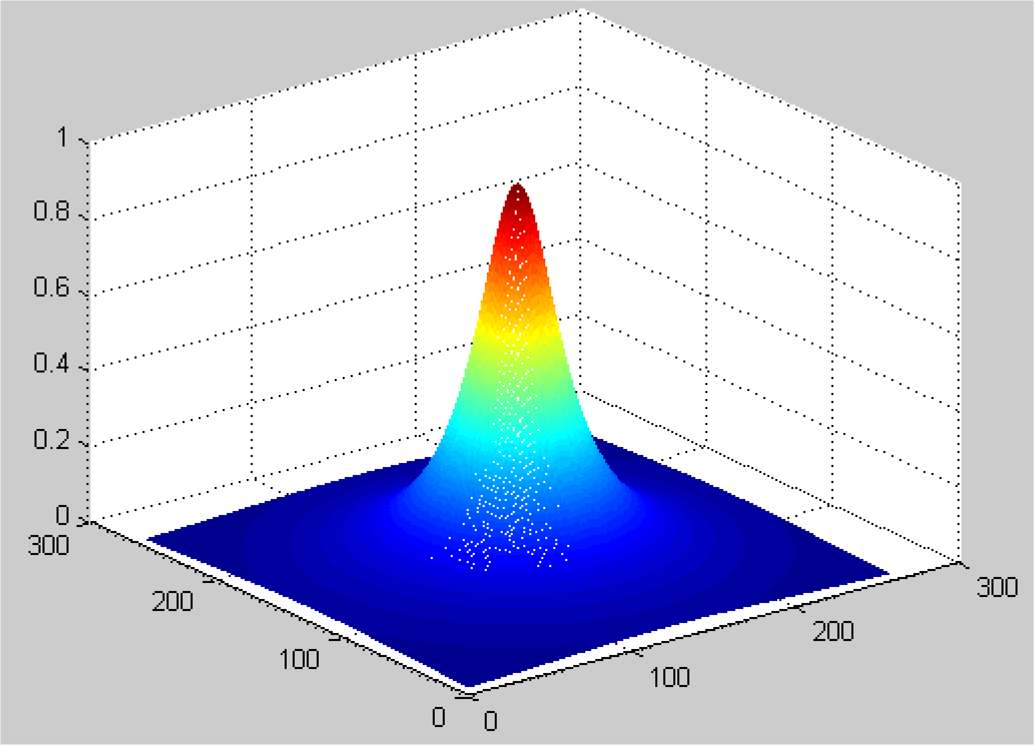
$\uparrow D_0 \longrightarrow \downarrow$ Ringing radius + blur

Efeito oscilatório (*Ringing Problem*)

- Solução
  - Usar filtros que possuem uma variação mais suave em torno das frequências de corte
- Exemplos
  - Filtro Butterworth
    - corte mais abrupto em relação ao Gaussiano
    - ainda apresenta ruído oscilatório
  - Filtro Gaussiano
    - corte suave - maior blur
    - não apresenta ruído oscilatório

Filtro Passa-Baixa Butterworth

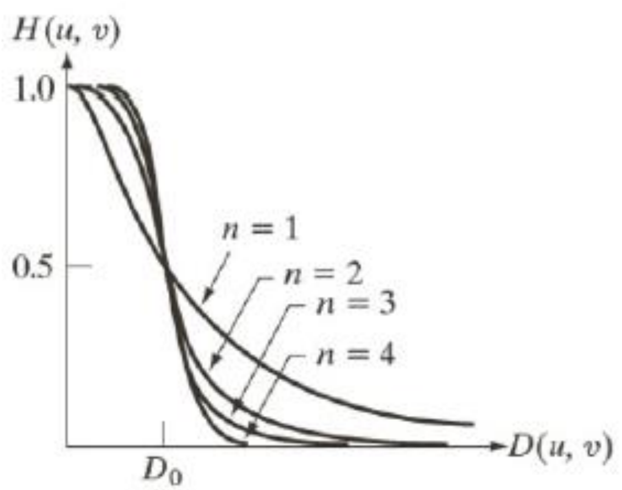
- Não apresenta uma descontinuidade abrupta
  - Não resulta em um corte bem definido



Filtro Passa-Baixa Butterworth

- É definido pela equação

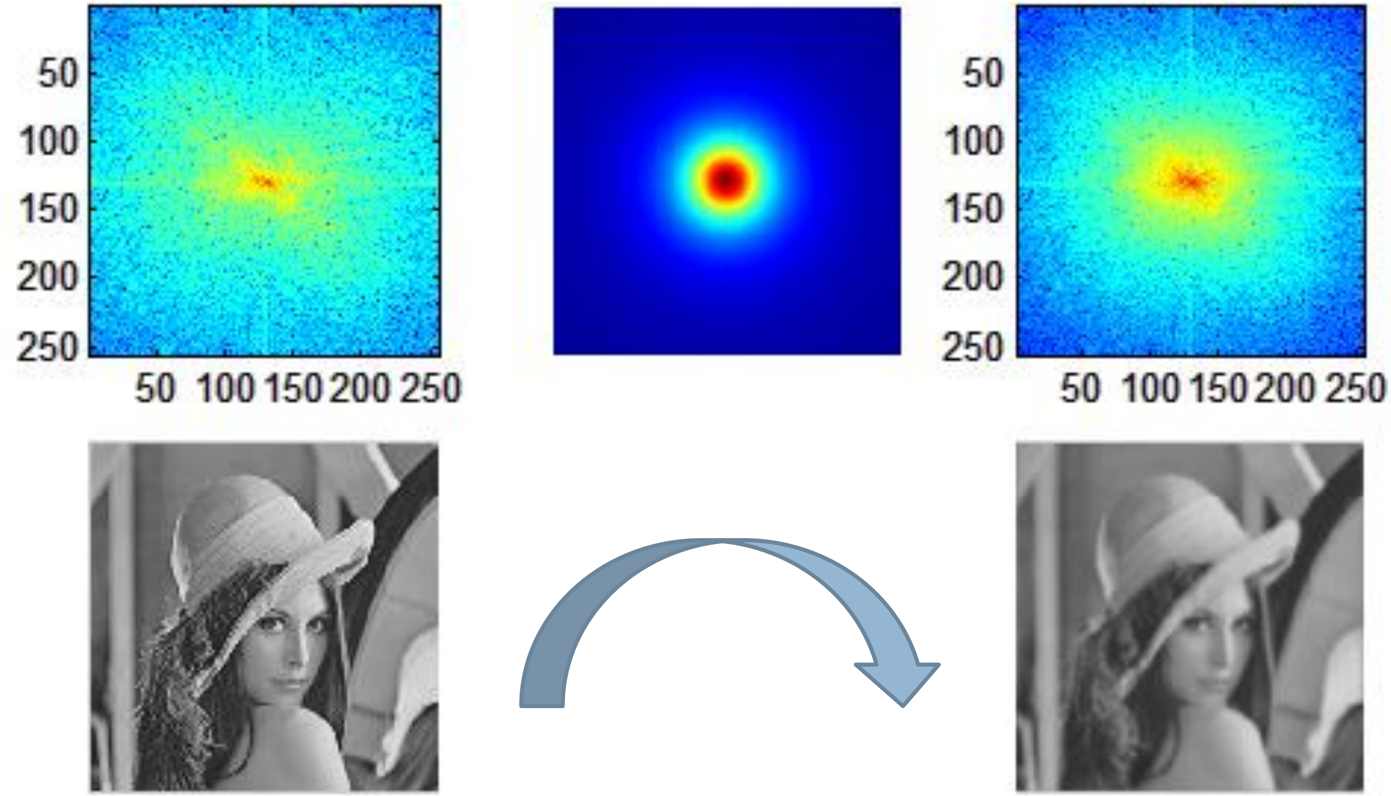
$$H(\mu,\nu) = \frac{1}{1 + [D(\mu,\nu)/D_0]^{2n}}$$



- onde
  - *D(μ,v)* é a distância entre um ponto *(μ,v)* no domínio da frequência e o centro da função de frequência
  - ***n*** é a ordem do filtro
  - D_0 é a frequência de corte (distância da origem)

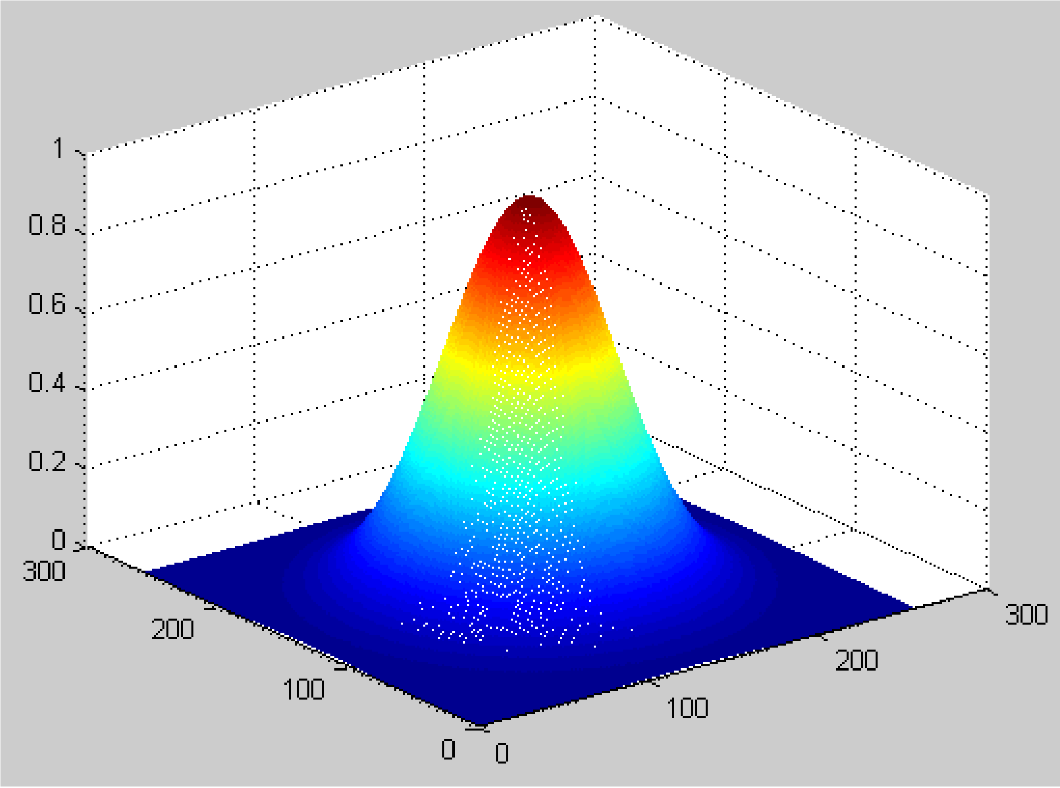
Filtro Passa-Baixa Butterworth

- O filtro passa-alta é obtido fazendo-se
  - $L(\mu,v) = 1 - H(\mu,v)$



Filtro Passa-Baixa Gaussiano

- Não apresenta efeito de *ringing*
  - Custo maior de calcular em relação ao *Butterworth*

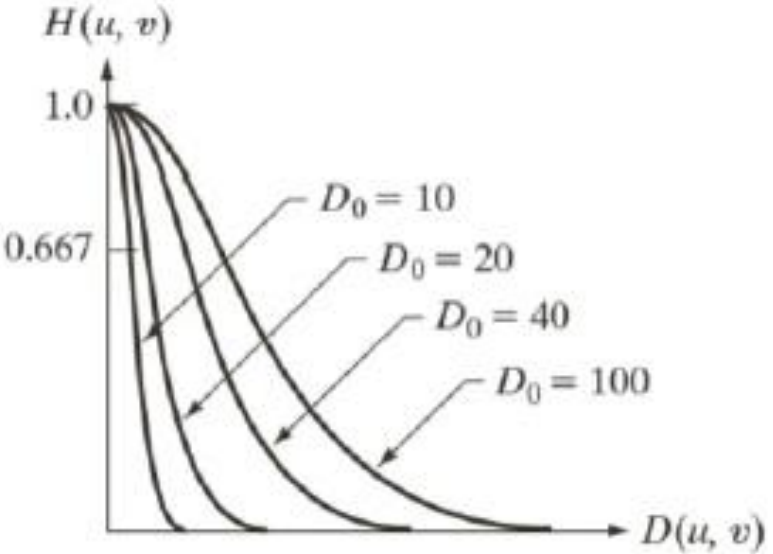


Filtro Passa-Baixa Gaussiano

- É definido pela equação

$$H(\mu,\nu) = e^{-D^2(\mu,\nu)/2D_0^2}$$

- onde
  - $D(\mu,v)$ é a distância entre um ponto (μ,v) no domínio da frequência e o centro da função de frequência
  - D_0 é a frequência de corte (distância da origem)



Filtro Passa-Baixa Gaussiano

- O filtro passa-alta é obtido fazendo-se
  - $L(\mu,v) = 1 - H(\mu,v)$