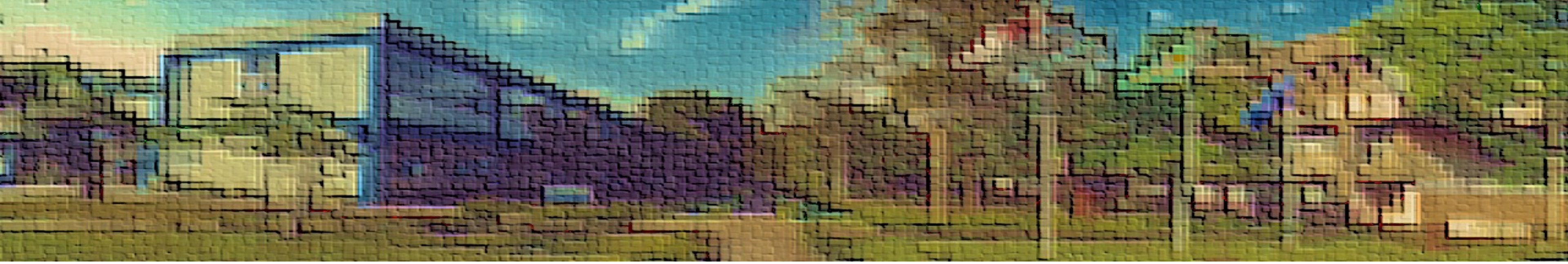


MAC317
Introdução ao Processamento de Sinais Digitais

Prof. Marcel P. Jackowski

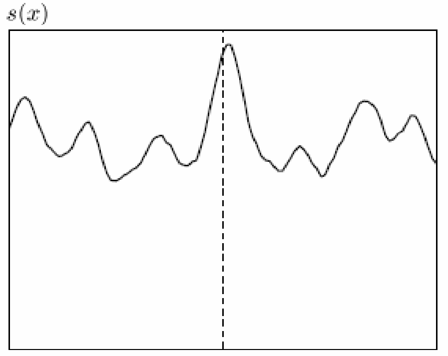
mjack@ime.usp.br

Aula #6: Transformada discreta de Fourier (Parte 2)

Fourier

Theory developed by Joseph Fourier (1768-1830)

The Fourier transform of a signal s(x) yields its *frequency spectrum S*(*k*)

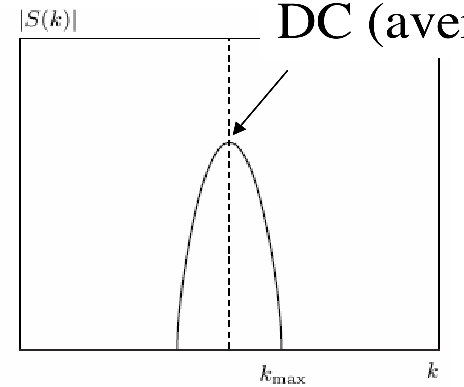


$s(x)$

forward transform

$$S(k) = F\{s(x)\} = \int_{-\infty}^{+\infty} s(x) e^{-2\pi i k x} dx$$

DC (average) term



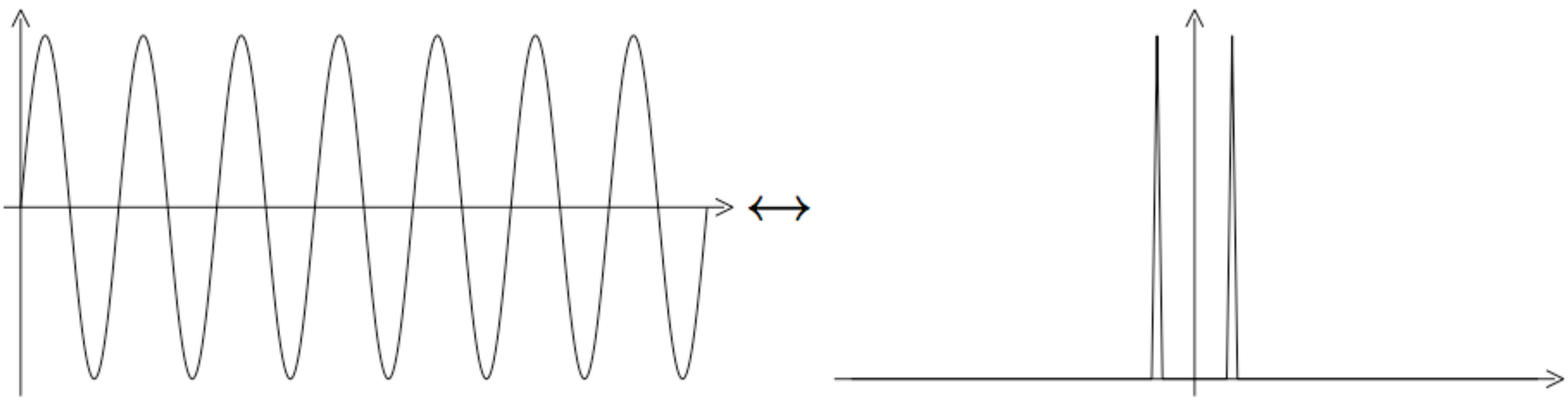
$S(k)$

inverse transform

$$s(x) = F^{-1}\{S(k)\} = \int_{-\infty}^{+\infty} S(k) e^{2\pi i k x} dk$$

Exemplo de sinal: seno

Single sine wave $f(x) = \sin(\omega x)$:



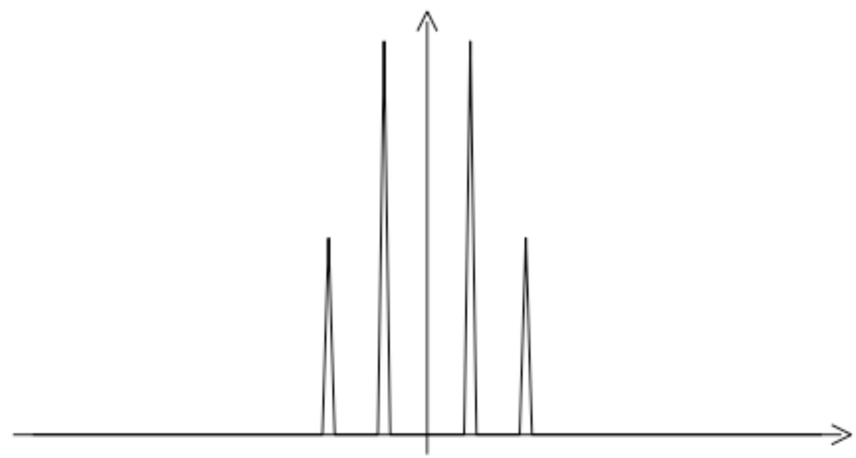
$f(x)$ $\quad$ $|F(\omega)|$

Exemplo de sinal senoidal

Sum of two sine waves

$f(x) = \sin(\omega x) + 0.5\sin(2\omega x)$:

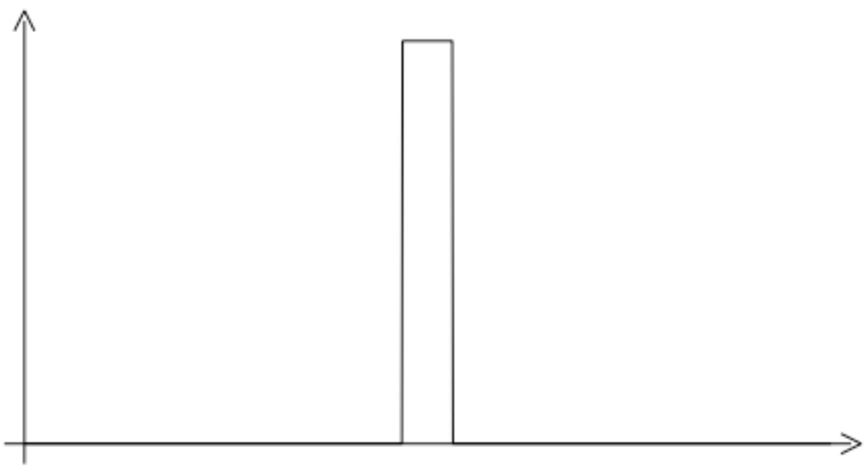
$\longleftrightarrow$



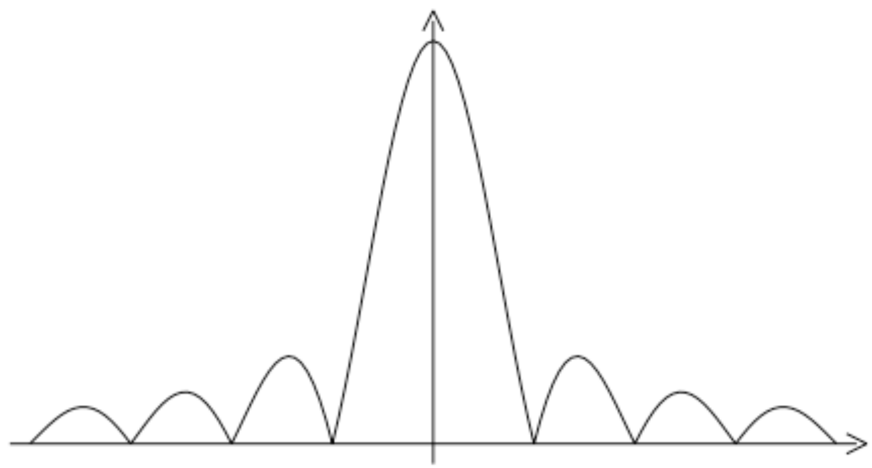
f(x) **|F(ω)|**

Função Rect

Box function:



$\leftrightarrow$

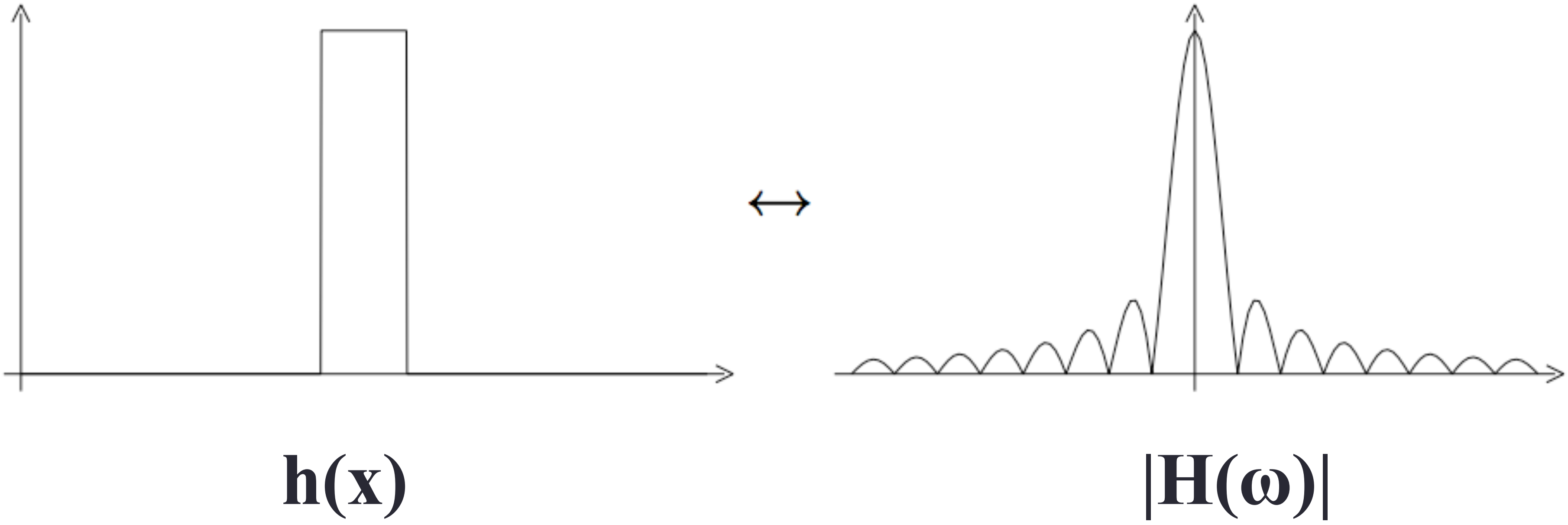


$h(x)$

$|H(\omega)|$

Função Rect

Wider box function:



Generalização

The Fourier transform generalizes to higher dimensions

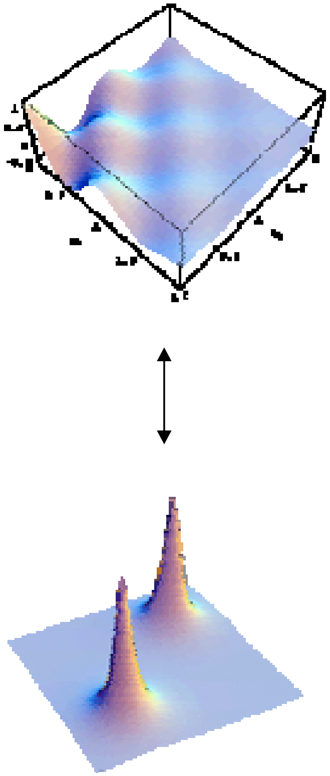
Consider the 2D case:

forward transform

$$S(k,l) = F\{s(x,y)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} s(x,y)e^{-2\pi i(kx+ly)}dxdy$$

inverse transform

$$s(x,y) = F^{-1}\{S(k,l)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} S(k,l)e^{2\pi i(kx+ly)}dkdl$$



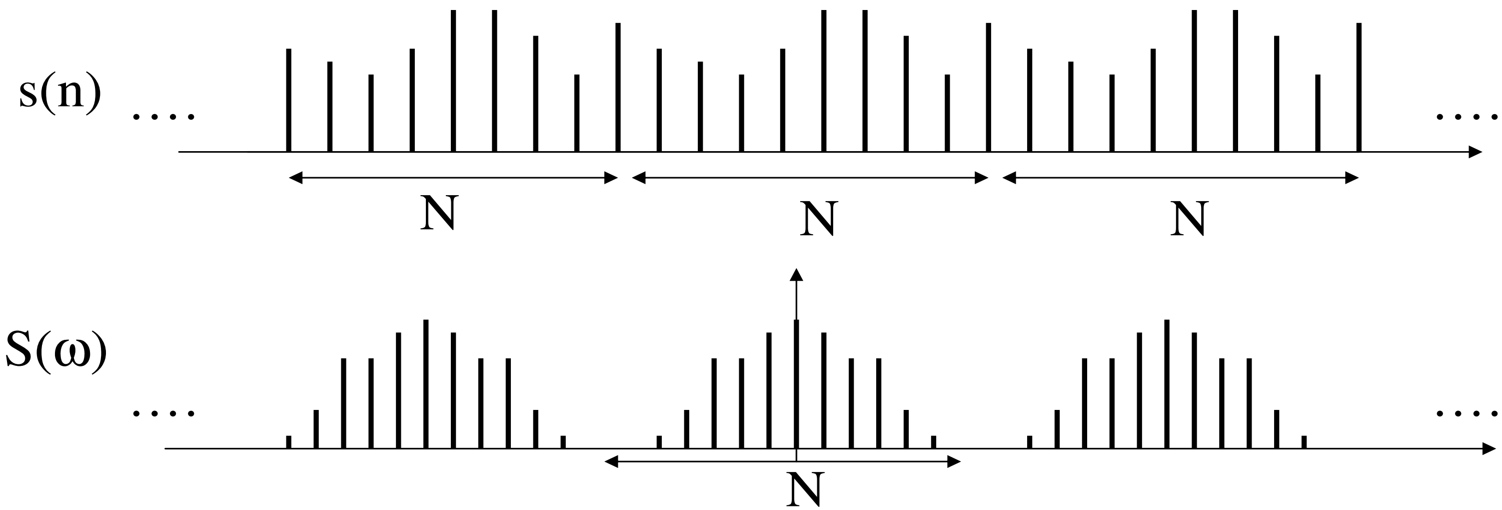
Transformada discreta

Discrete Fourier Transform (DFT)

- assumes that the signal is discrete and finite

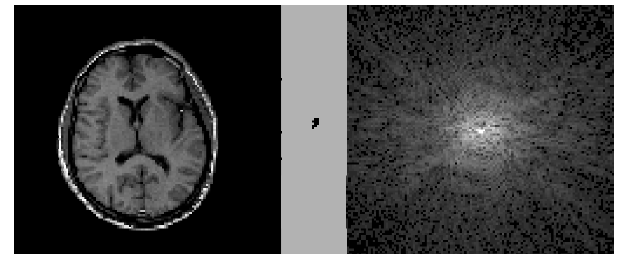
$$S(k) = \sum_{n=0}^{N-1} s(n) e^{\frac{-i2\pi kn}{N}} \qquad s(n) = \frac{1}{N} \sum_{n=0}^{N-1} S(k) e^{\frac{i2\pi kn}{N}}$$

- now we have only N samples, and we can calculate N frequencies
- the frequency spectrum is now discrete, and it is periodic in N



Exemplo

The 2D transform:



$$S(k,l)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}s(n,m)e^{\frac{-i2\pi(kn+lm)}{NM}}$$

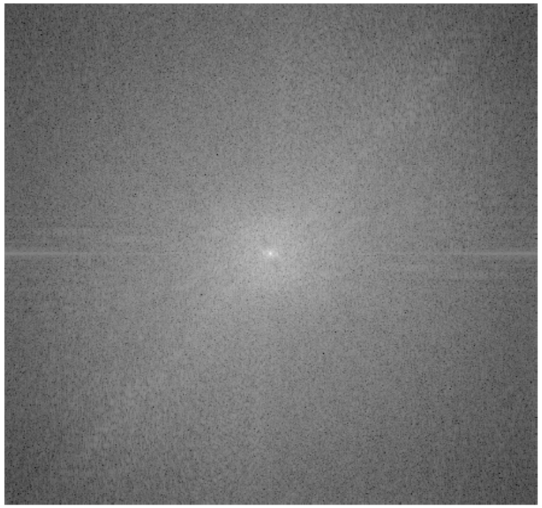
$$s(n,m)=\frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}S(k,l)e^{\frac{i2\pi(kn+lm)}{NM}}$$

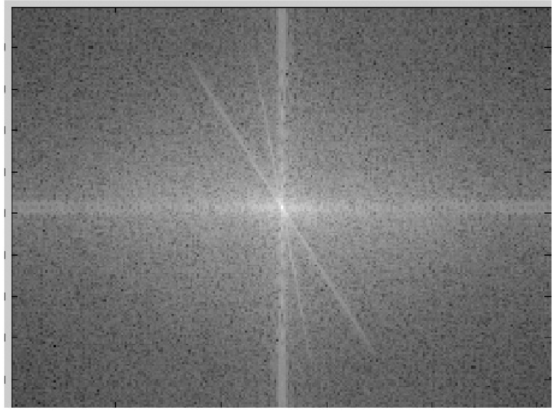
Separability:

$$S(k,l)=\frac{1}{NM}\sum_{m=0}^{M-1}e^{\frac{-i2\pi lm}{M}}P(k,m) \quad \text{where } P(k,m)=\sum_{n=0}^{N-1}s(n,m)e^{\frac{-i2\pi kn}{N}}$$

$$s(n,m)=\frac{1}{NM}\sum_{l=0}^{M-1}e^{\frac{-i2\pi lm}{M}}p(n,l) \quad \text{where } \mathrm{p}(n,l)=\sum_{k=0}^{N-1}S(n,m)e^{\frac{-i2\pi kn}{N}}$$

Exemplos





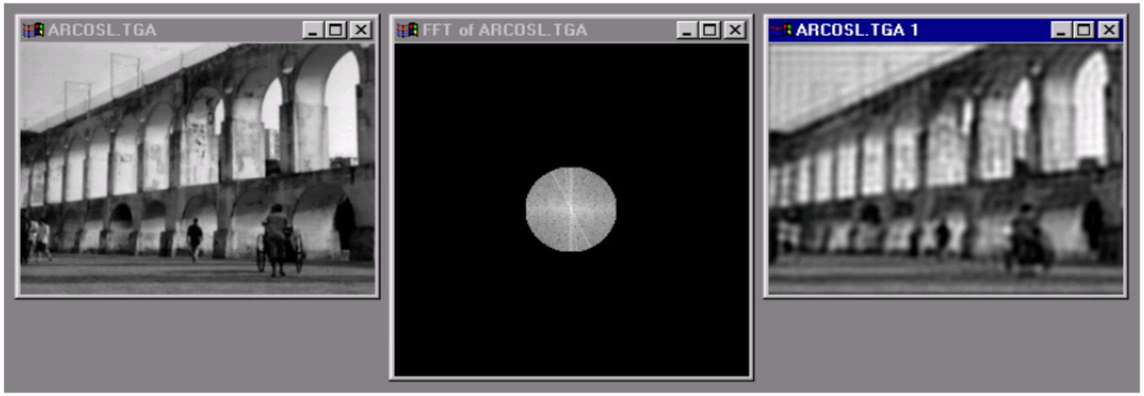
A. Efros, CMU

Modificações no espectro

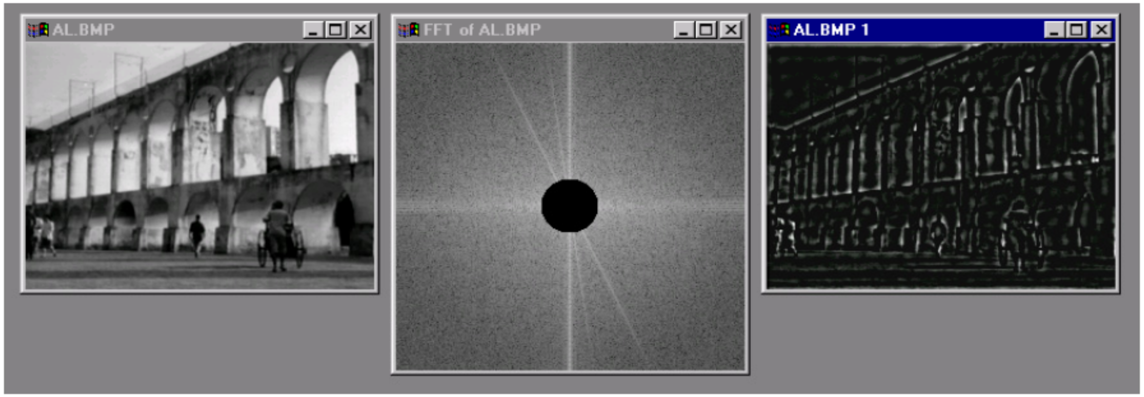
(a) Lower frequencies (close to origin) give overall structure

(b) Higher frequencies (periphery) give detail (sharp edges)

(a)

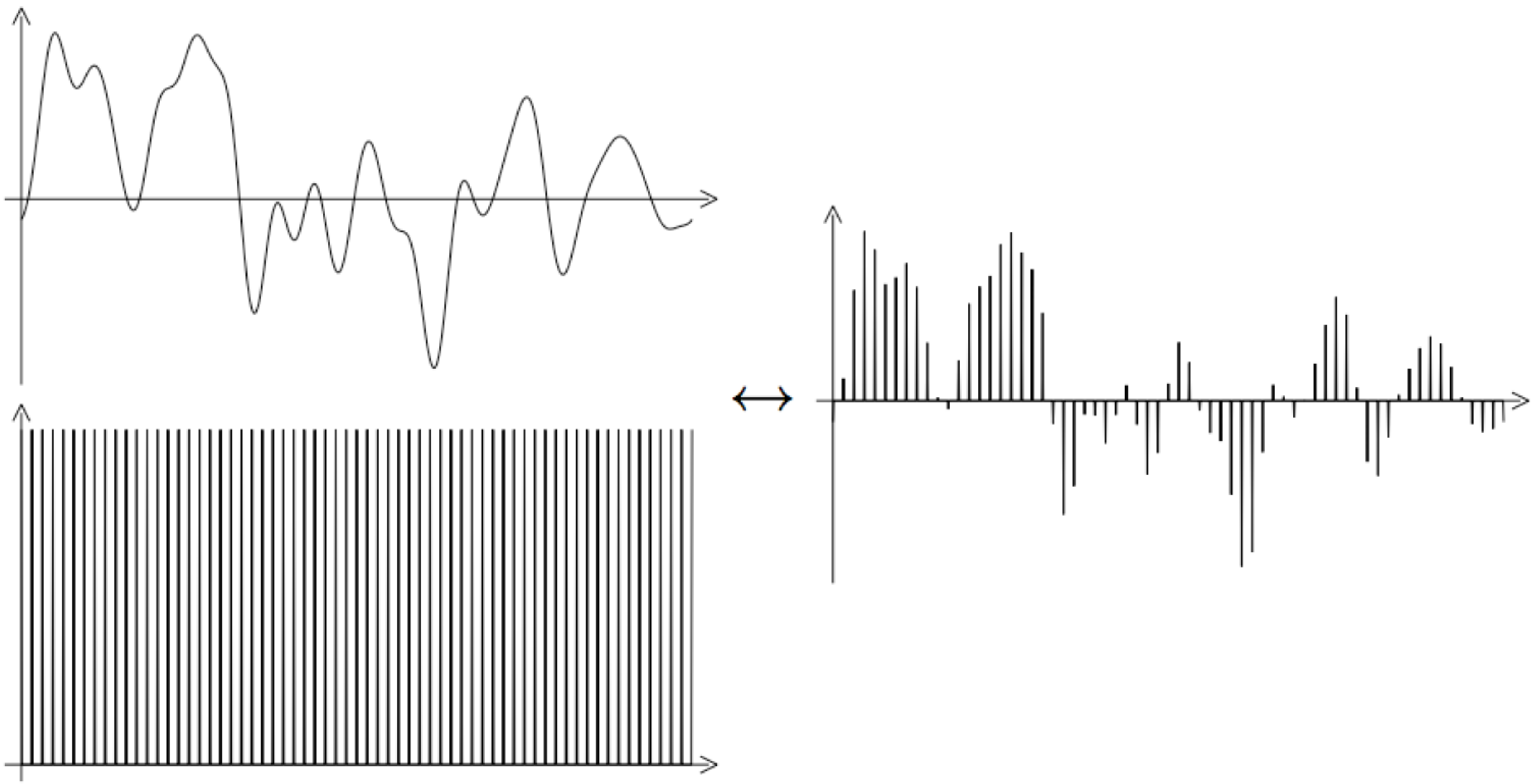


(b)



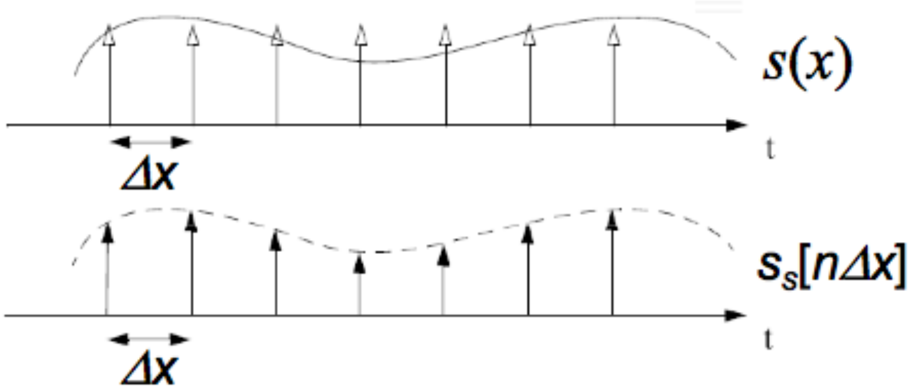
A. Efros, CMU

Amostragem



Filtro “pente”

- a continuous signal *s*(*x*) is measured at fixed instances spaced apart by an interval Δx
- the data points so obtained form a discrete signal $s_s[n\Delta x] = s_s(n\Delta x)$
- here, Δx is called the *sampling period (distance)*, and K = 1/Δx the *sampling frequency*



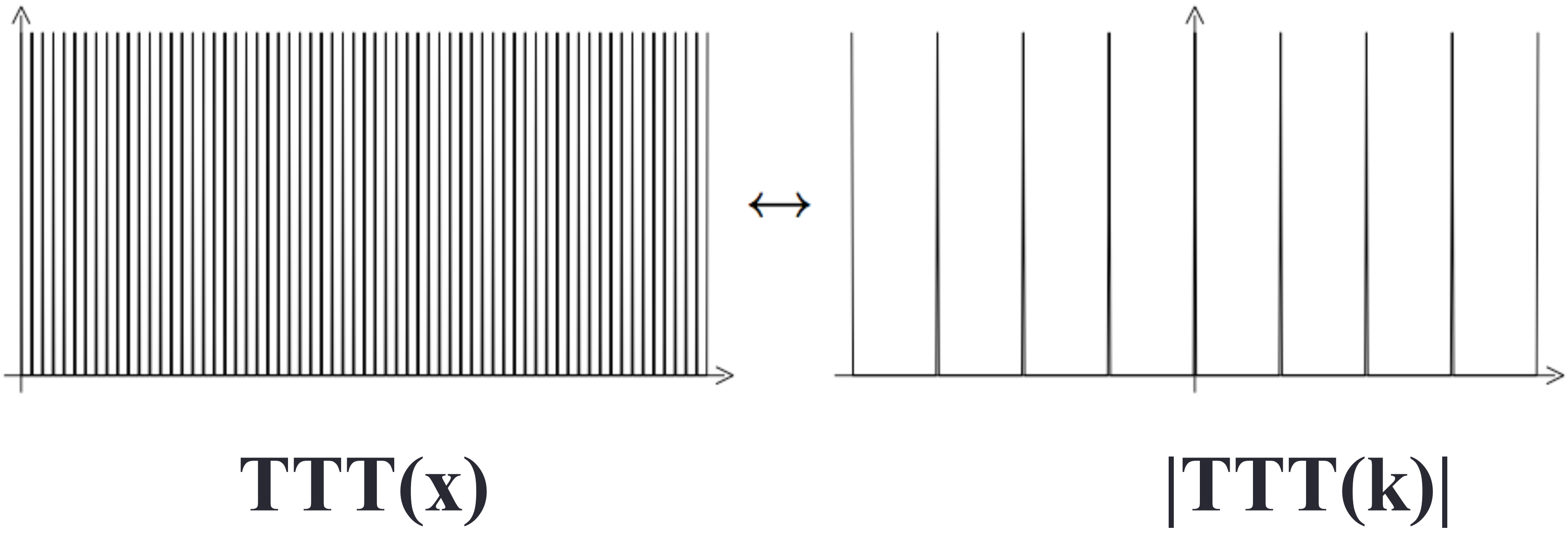
Sampling is the multiplication of the signal with an impulse train:

$$s_s(x) = s(x) \cdot \mathrm{TTT}(x)$$

$$\mathrm{TTT}(x) = \sum_{n=-\infty}^{+\infty} \delta(x - n\Delta x), \quad \mathrm{TTT}(x) \text{ is the comb function}$$

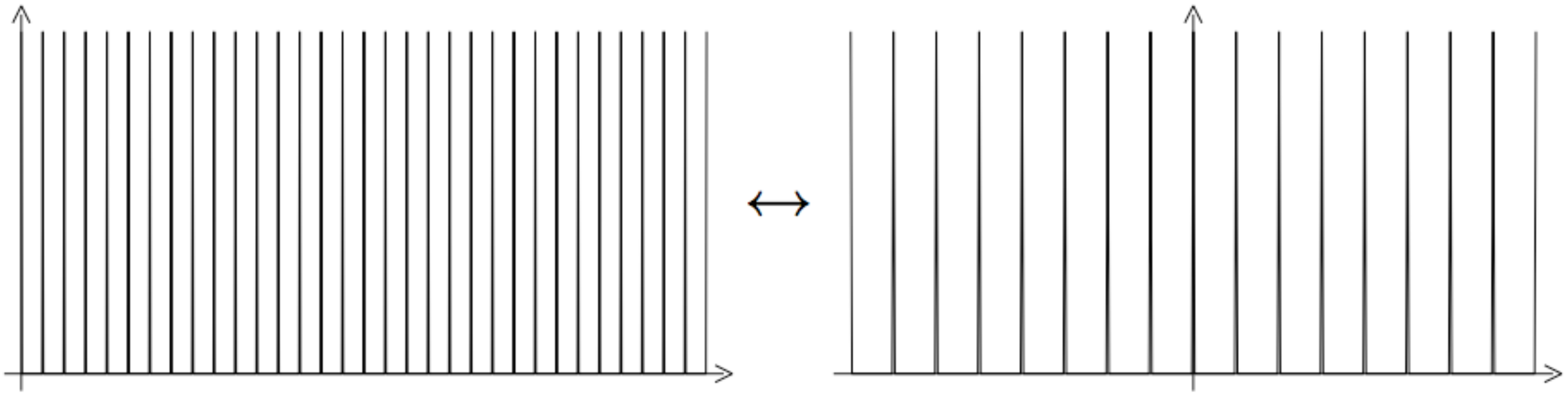
Filtro ”pente”

Comb function:



Filtro “pente”

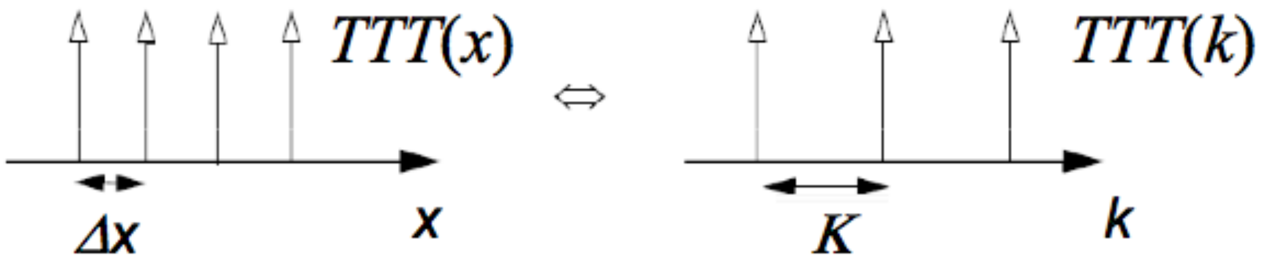
Wider comb function:



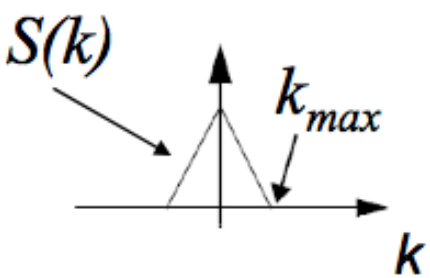
TTT(x) $\leftrightarrow$ |TTT(k)|

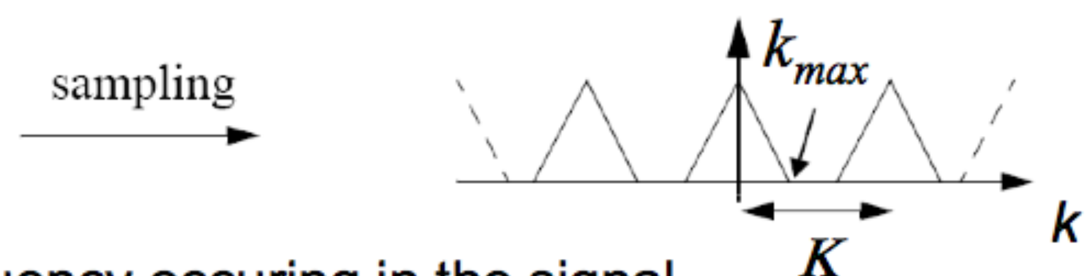
Amostragem no espaço Fourier

$$S_s(k) = S(k) * F\{\mathrm{TTT}(x)\}, \quad \text{where } F\{\mathrm{TTT}(x)\} = K \sum_{l=-\infty}^{+\infty} \delta(k - lK)$$



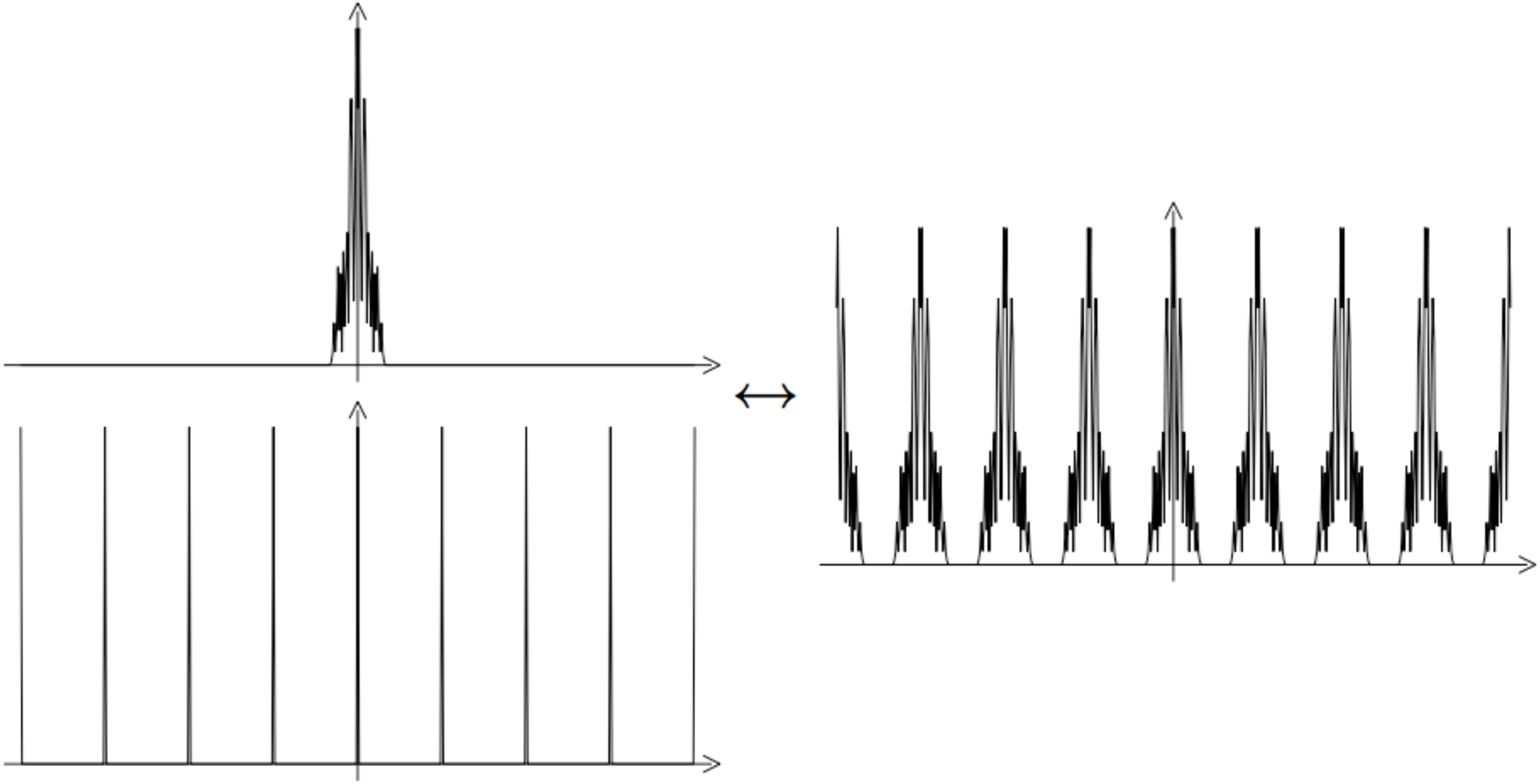
- the smaller Δx the wider K (recall the Fourier scaling theorem)
- sampling (the convolution of $TTT(k)$ and $S(k)$) replicates the signal spectrum $S(k)$ at integer multiples of sampling frequency K



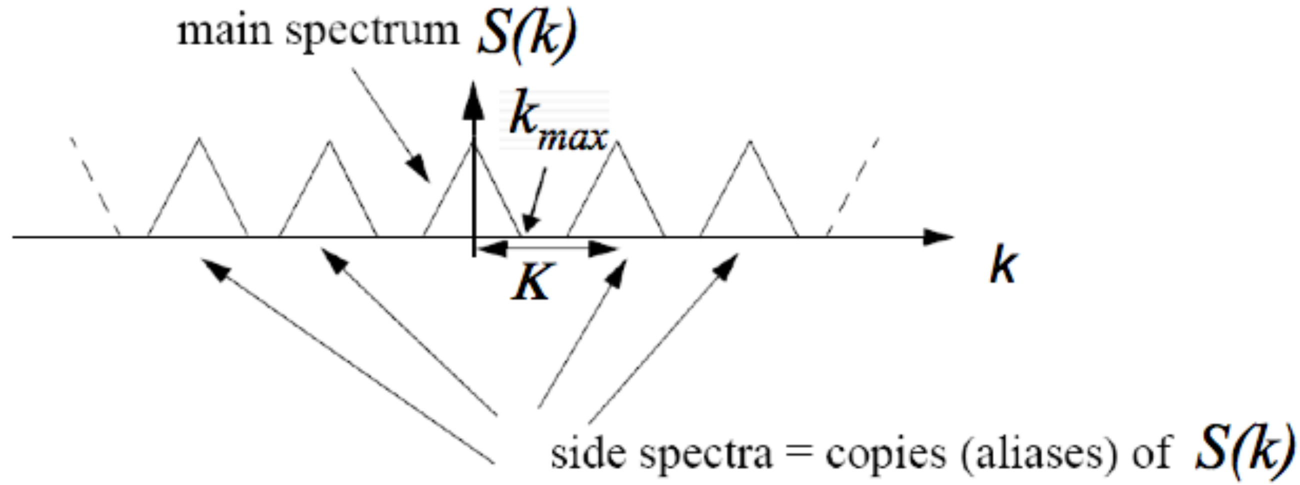


- k_{max} is maximum frequency occuring in the signal

Amostragem no espaço Fourier

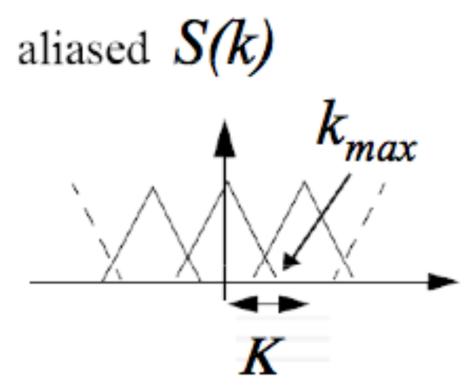


"Aliasing"



However, if we choose $K < 2\, k_{max}$ the aliases overlap and we get *aliasing*

- what does aliasing look like?



Frequência de Nyquist

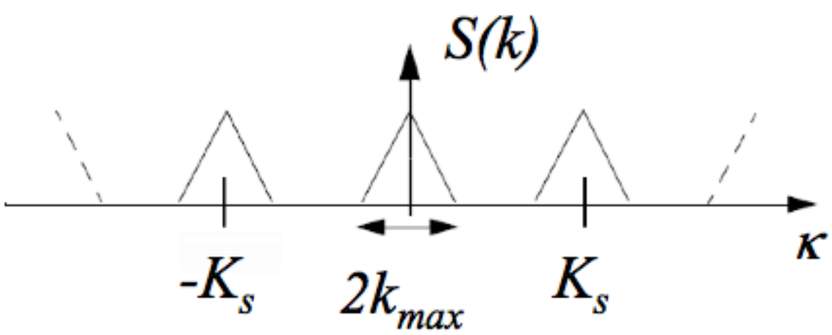
$$K > K_s = 2 \cdot k_{\max}, \quad K_s \text{ is the } \textit{Nyquist rate}$$

In other words:

- the samples only uniquely define the signal if:

$$S(k) = 0 \quad \forall |k| > k_{\max}$$

$$\frac{1}{\Delta x} > 2k_{\max} = K_s$$



- this assumes that the signal is band-limited ($S(k)$=0 above K_s

"Antialising"

Usually signals are not band-limited

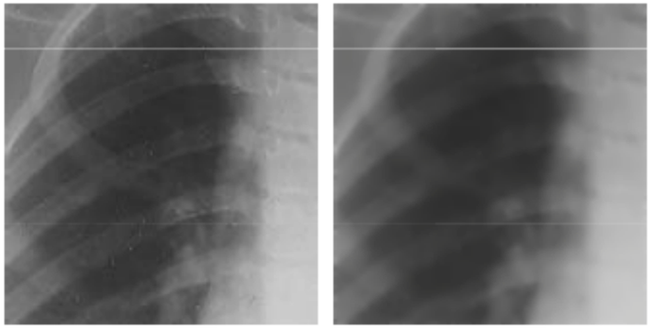
- recall the infinite spectrum of a sharp edge (for example: a bone)

To prevent the inevitable aliasing we must perform anti-aliasing before sampling the signal

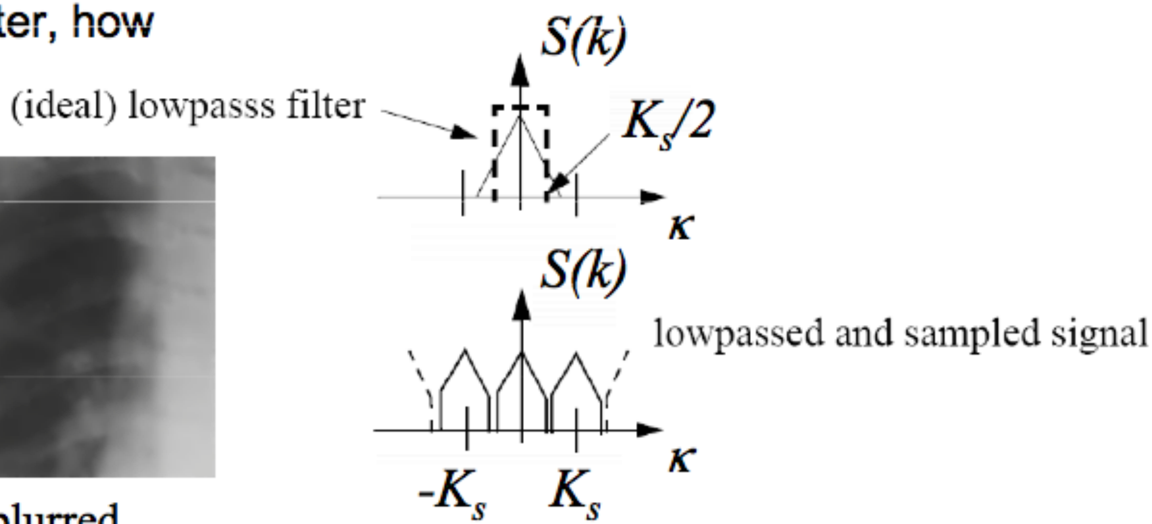
- for example: when digitizing a radiograph of a bone or a chest

Anti-aliasing is done by low-pass filtering (blurring)

- band-limit the signal *prior* to sampling
- we shall see later, how

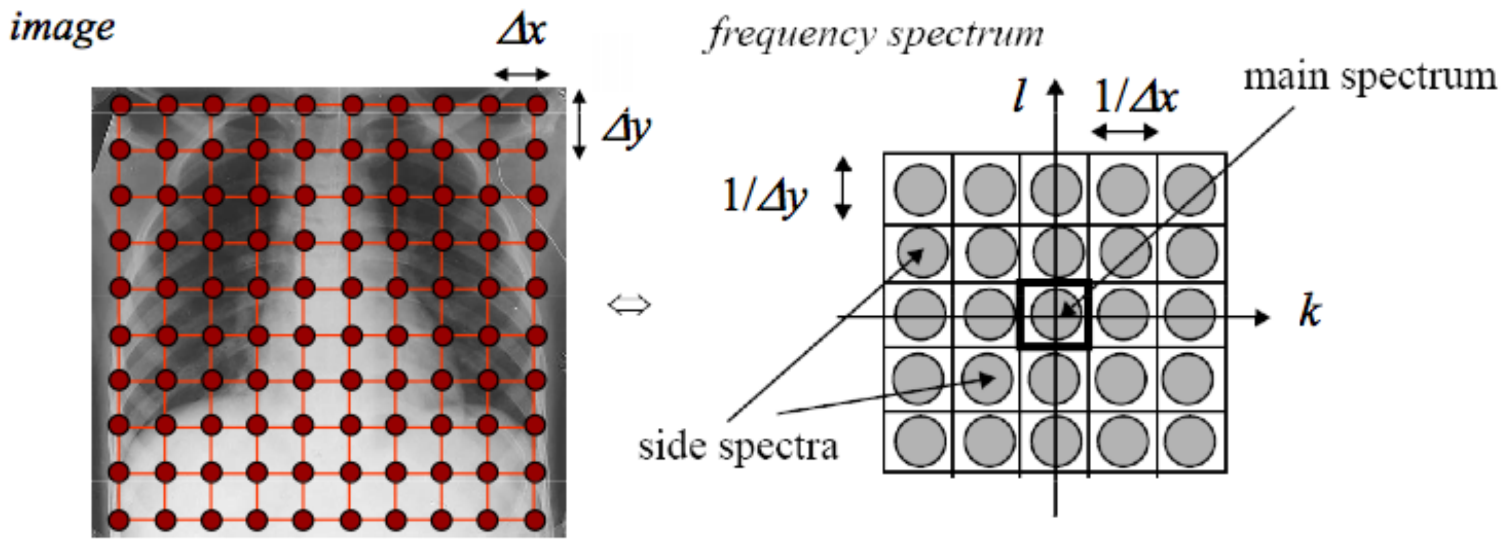


original blurred



Generalização

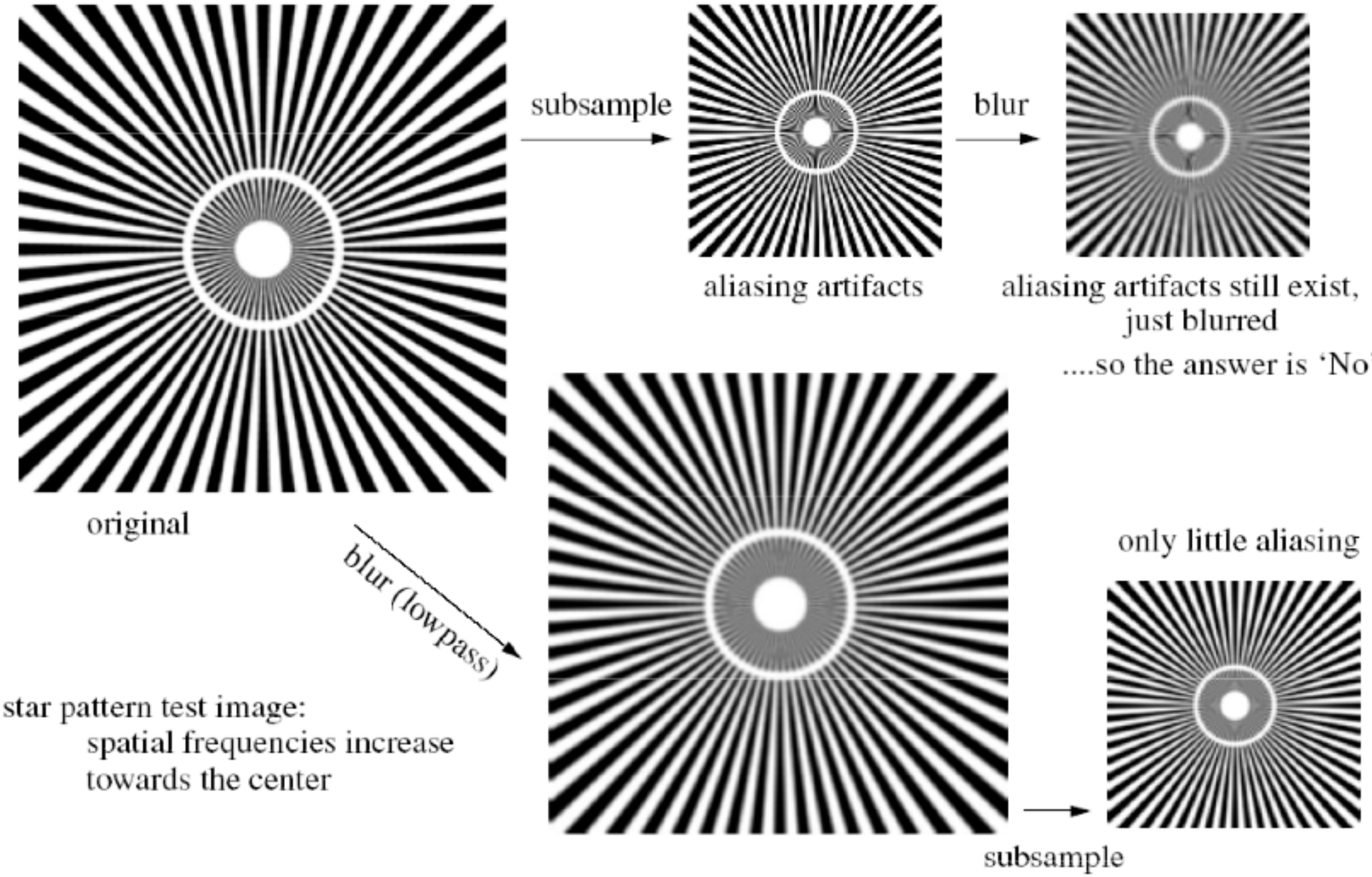
All of these concepts readily extend to higher dimensions



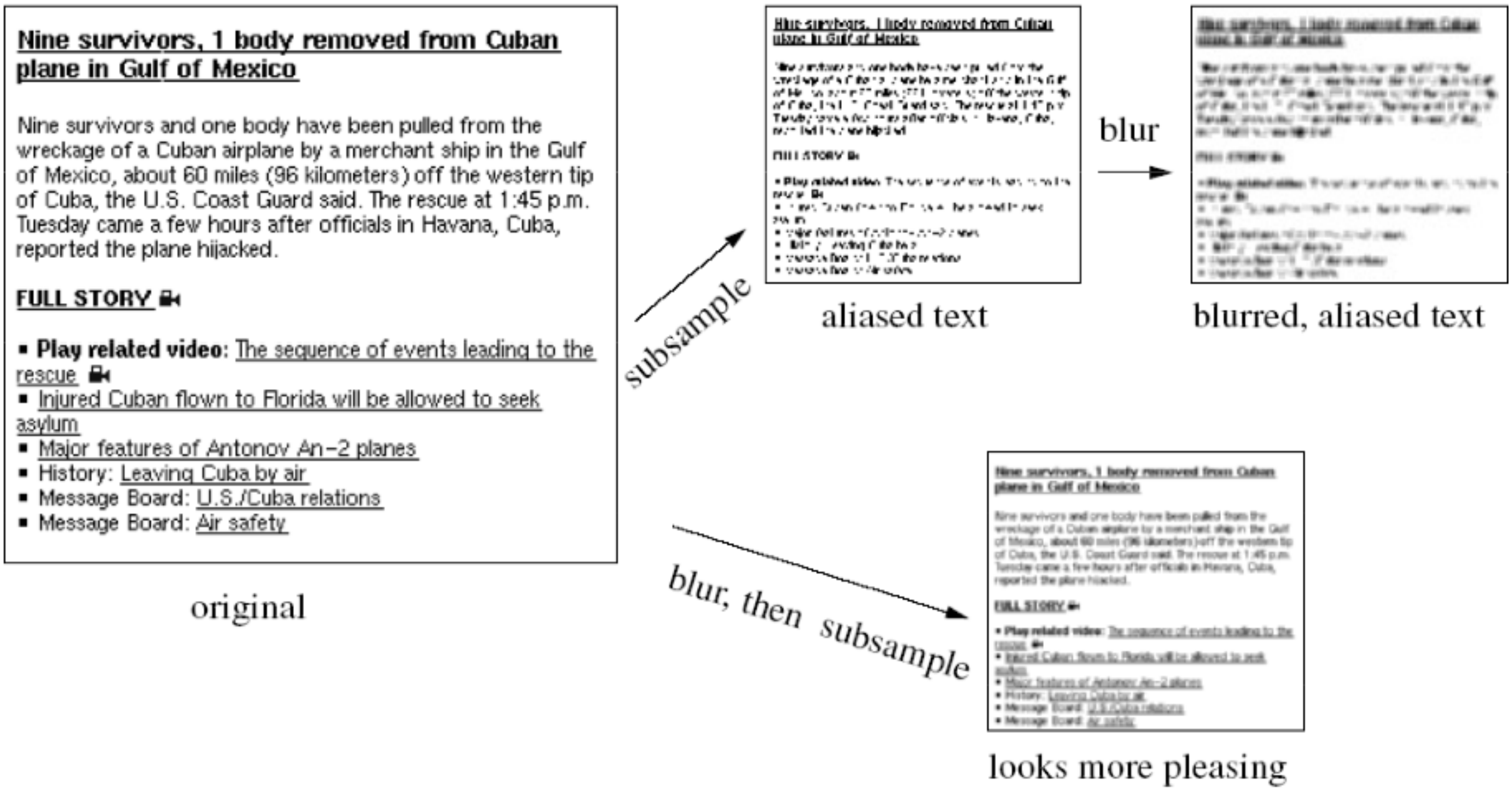
Main spectrum ($S(k,l)$) must fit into the center box to prevent overlap with side-spectra (and aliasing)

$$\frac{1}{\Delta x} > 2 \cdot k_{x\max} \qquad \frac{1}{\Delta y} > 2 \cdot k_{y\max}$$

Exemplo


subsample
blur
aliasing artifacts
aliasing artifacts still exist,
just blurred
....so the answer is 'No'
original
blur (lowpass)
only little aliasing
star pattern test image:
spatial frequencies increase
towards the center
subsample

Exemplo



We observe: Anti-aliasing (i.e., blurring, lowpassing) must be applied before sampling

Resposta de sistema

Now assume the input is a complex sinusoid with $Ae^{2\pi ikx}$ then:

for now, assume $\varphi=0$

$$s_0(x) = \int_{-\infty}^{+\infty} Ae^{2\pi ik(x-\xi)} h(\xi)d\xi$$

$$= Ae^{2\pi ikx} \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi)d\xi$$

$$= Ae^{2\pi ikx} H$$

H is called the *Fourier Transform* of *h*(*x*):

$$H = \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi)d\xi$$

- *H* is also often called the *transfer function* or *filter*

Observações

H scales, and maybe phase-shifts, the input sinusoid S_i

In essence, we have now two alternative representations:

- determine the effect of L on s_i by convolution with h: $s_i * h$
- determine the effect of L on s_i by multiplication with H: $S_i \cdot H$

$$s_i * h \leftrightarrow S_i \cdot H$$

Since convolution is expensive for wide h, the multiplication may be cheaper

- but we need to perform the Fourier transforms of s_i and h

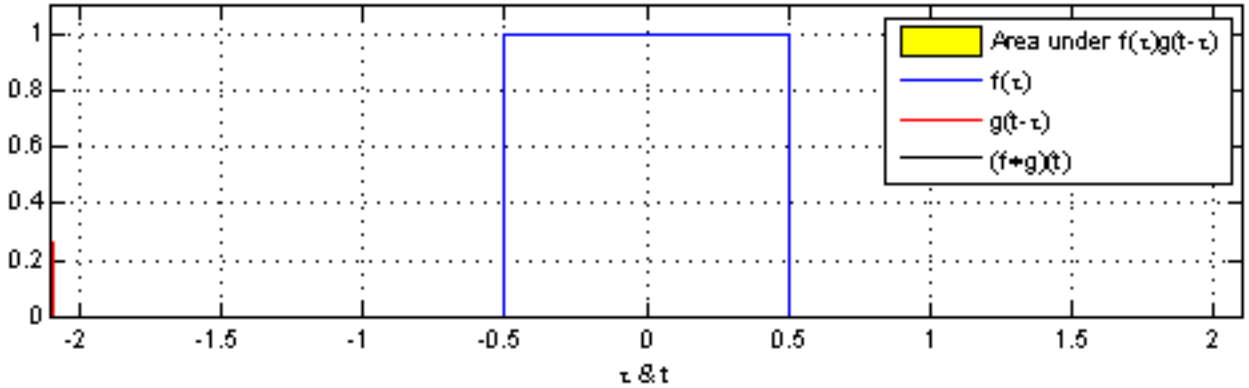
Convolução

The expression

$$s_o(x) = \int_{-\infty}^{+\infty} s_i(\xi) h(x-\xi) d\xi = s_i * h$$

is called *convolution*, defined as:

$$s_1(x) * s_2(x) = \int_{-\infty}^{+\infty} s_1(\xi) s_2(x-\xi) d\xi = \int_{-\infty}^{+\infty} s_1(x-\xi) s_2(\xi) d\xi$$



Sinais 2-D

$$s_1(x,y) * s_2(x,y)$$

$$= \iint_{-\infty}^{+\infty} s_1(x-\xi, y-\zeta)\, s_2(\xi,\zeta)\, \mathrm{d}\xi\, \mathrm{d}\zeta,$$

- Propriedades:
  - $s_1 * s_2 = s_2 * s_1$ (comutativa)
  - $(s_1 * s_2) * s3 = (s_1 * s_2) * s_3$ (associativa)
  - $s_1 * (s_2 + s_3) = s_1 * s_2 + s_1 * s_3$ (distributiva)

Resposta de sistema

Now assume the input is a complex sinusoid with $Ae^{2\pi ikx}$ then:

for now, assume $\varphi=0$

$$s_0(x) = \int_{-\infty}^{+\infty} Ae^{2\pi ik(x-\xi)} h(\xi) d\xi$$

$$= Ae^{2\pi ikx} \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi) d\xi$$

$$= Ae^{2\pi ikx} H$$

H is called the *Fourier Transform* of *h*(*x*):

$$H = \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi) d\xi$$

- *H* is also often called the *transfer function* or *filter*

Observações

H scales, and maybe phase-shifts, the input sinusoid S_i

In essence, we have now two alternative representations:

- determine the effect of *L* on s_i by convolution with *h*: $s_i * h$
- determine the effect of *L* on s_i by multiplication with *H*: $S_i \cdot H$

$$s_i * h \leftrightarrow S_i \cdot H$$

Since convolution is expensive for wide *h*, the multiplication may be cheaper

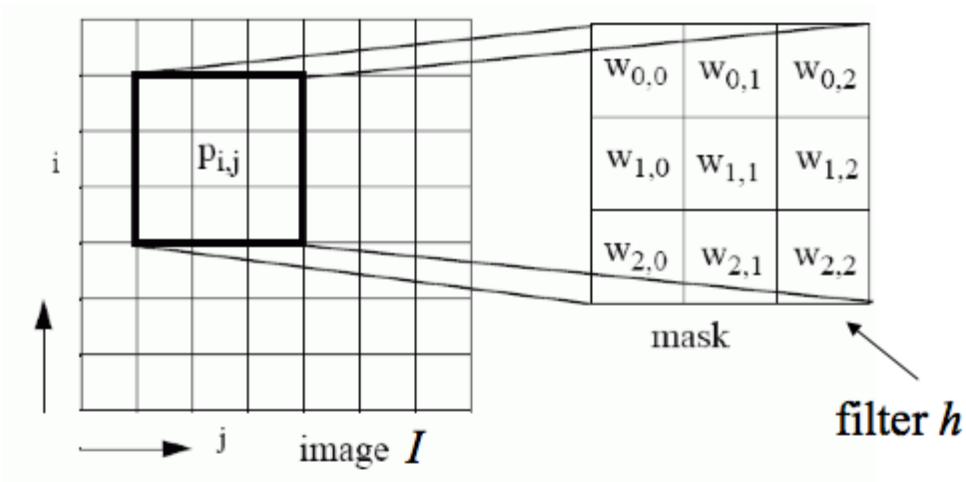
- but we need to perform the Fourier transforms of s_i and *h*

Filtragem espacial

We say *discrete filters* since they operate on a discretized signal, the image

- to implement discrete filters we use discrete convolution

$h * I$



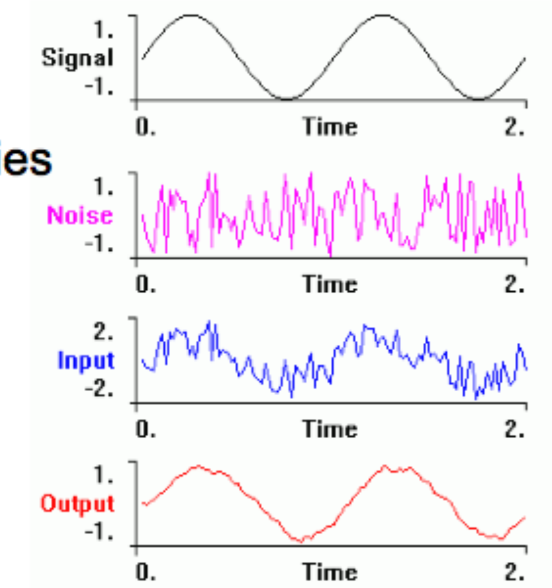
Procedure:

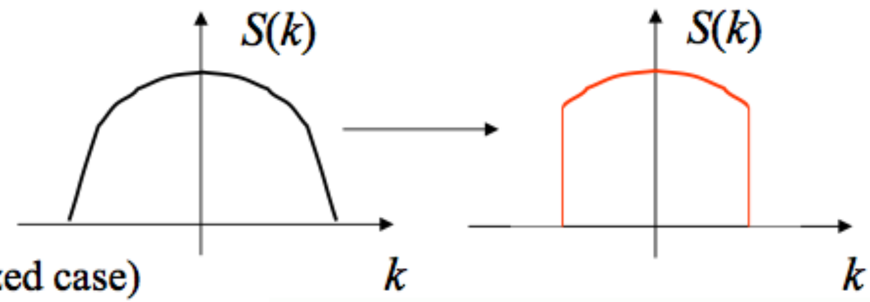
- place a weight matrix or *mask* at each pixel location p_{ij}
- this mask weighs the pixel's neighborhood and determines the output pixel's value
- important: do not replace the computed values into the original image, but write to an output image

Suavização

Smoothing (averaging):

- also called *low-passing:* keeps the low frequencies, but reduces the high frequencies
- removes noise and jagged edges
- but also blurs the signal

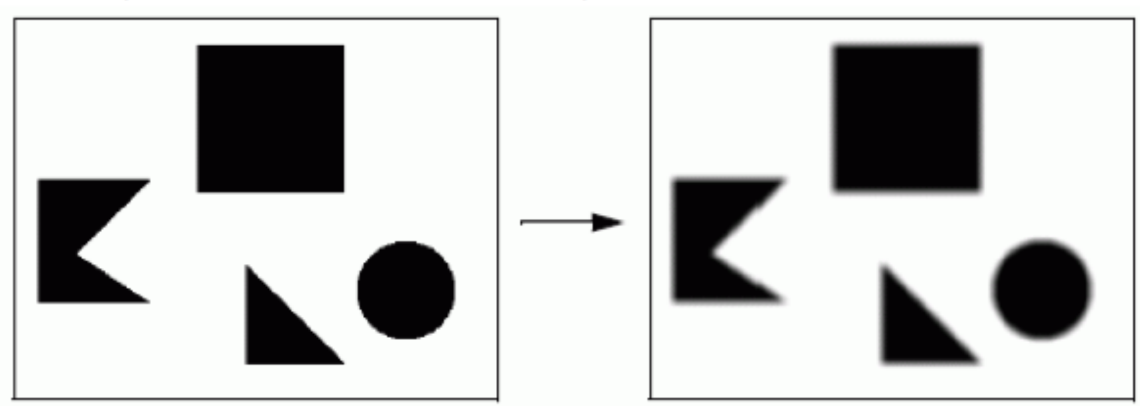




(idealized case)

1/9 ×

1	1	1
1	1	1
1	1	1



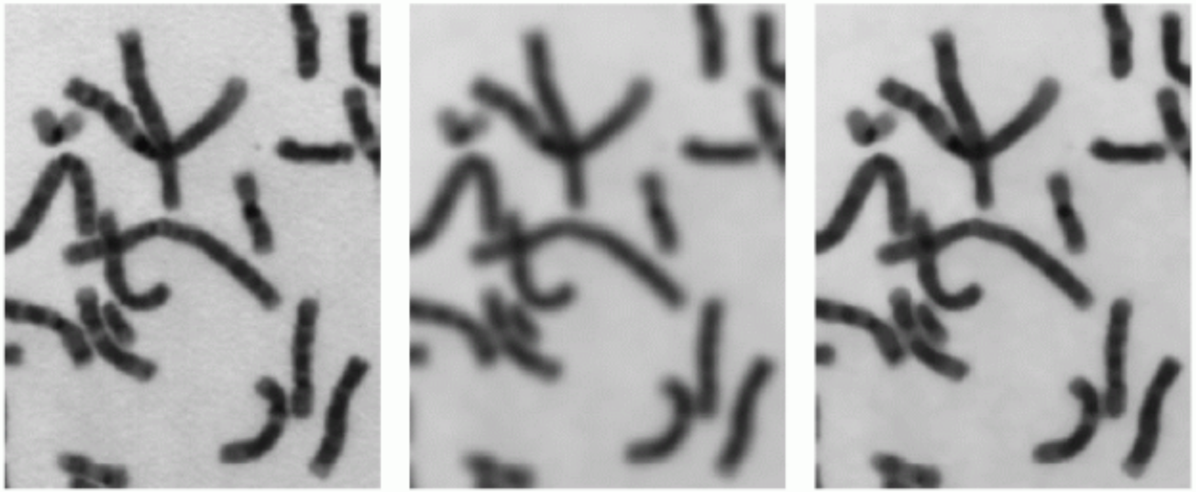
Mediana

A non-linear filter, best used to remove speckle noise

- a regular smoothing filter would blur the speckles (and the signal)
- the median filter will elimate the speckle and leave the signal as is

Procedure:

- convolve with a mask as usual
- but this time, for each mask position, sort the values under the mask
- pick the median and write to the output image
- the speckle pixel will be an outlier and not be selected as the median



original smoothed median filtered

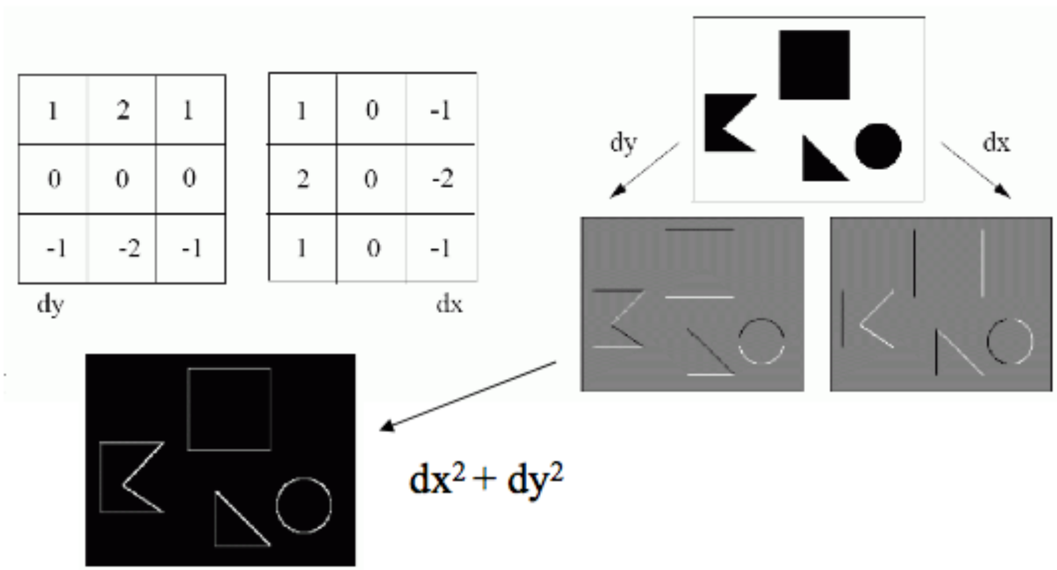
Filtro passa-alta

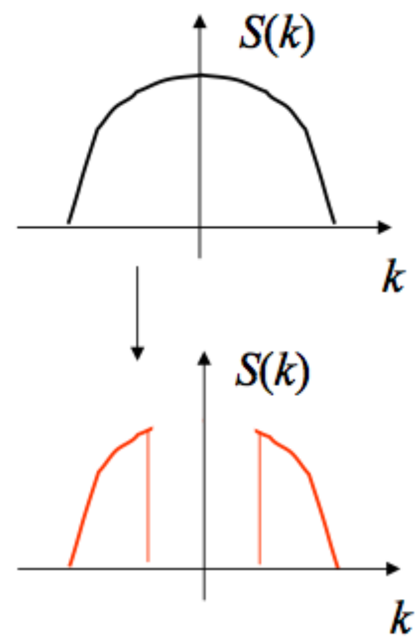
Edge detector / enhancer:

$$\nabla I = \nabla h * I \qquad \text{first derivative (gradient)}$$

$$\nabla^2 I = \nabla^2 h * I \qquad \text{second derivative (Laplacian)}$$

- also called *high-passing:* keeps the high frequencies, but reduces the low frequencies
- enhances edges and contrast
- but also enhances noise and jagged edges



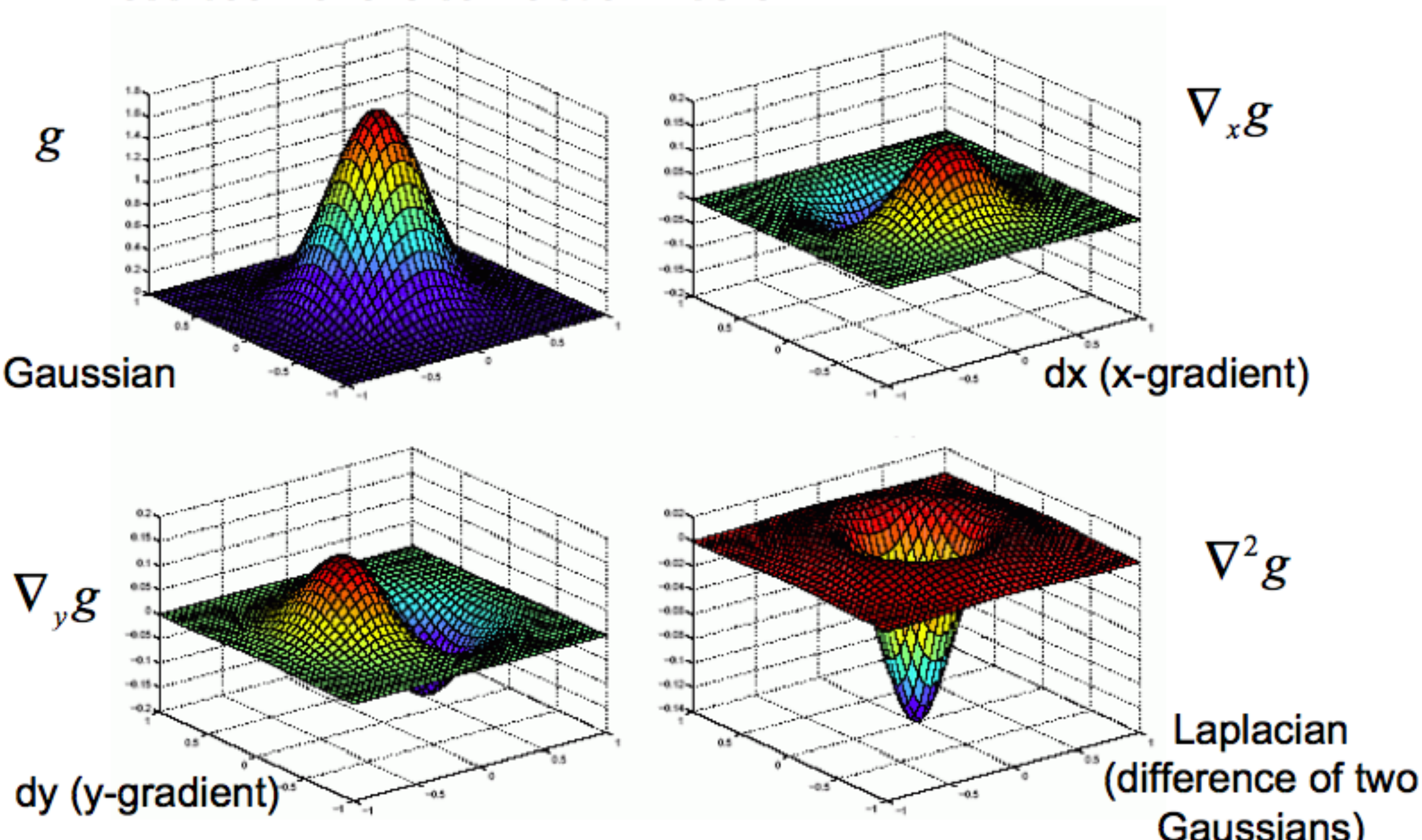


(idealized case)

Função Gaussiana

The Gaussian kernel is a popular filter function

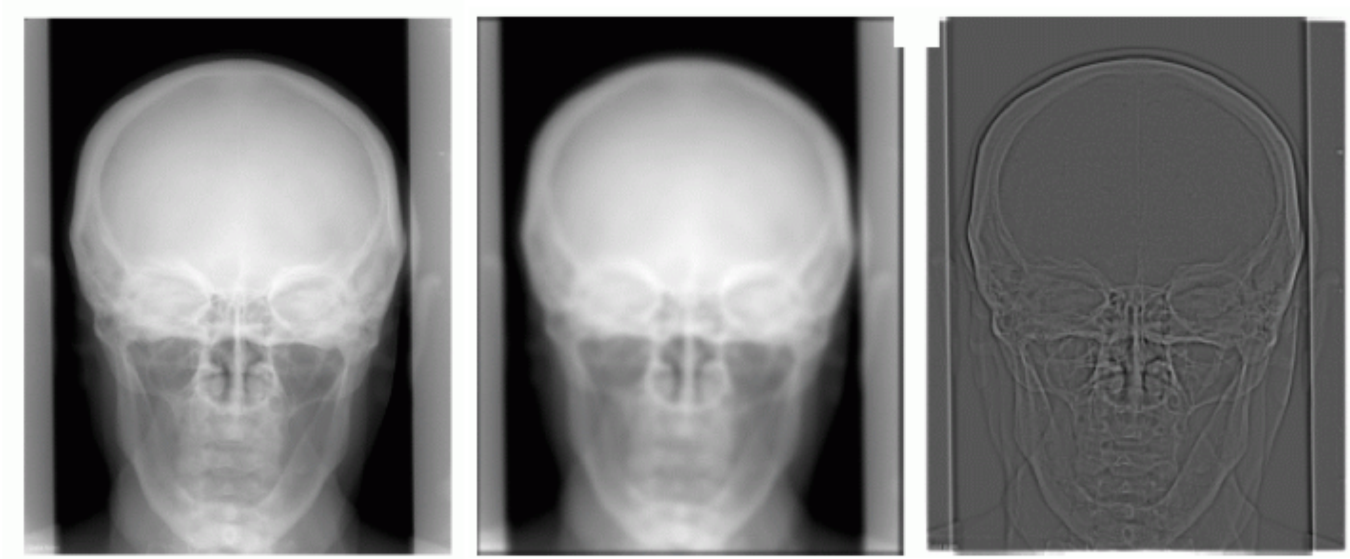
- see book for 3x3 convolution masks



Multiplas operações

Several useful effects can be achieved by subsequent filtering with different masks (kernels) and/or multi-image operations

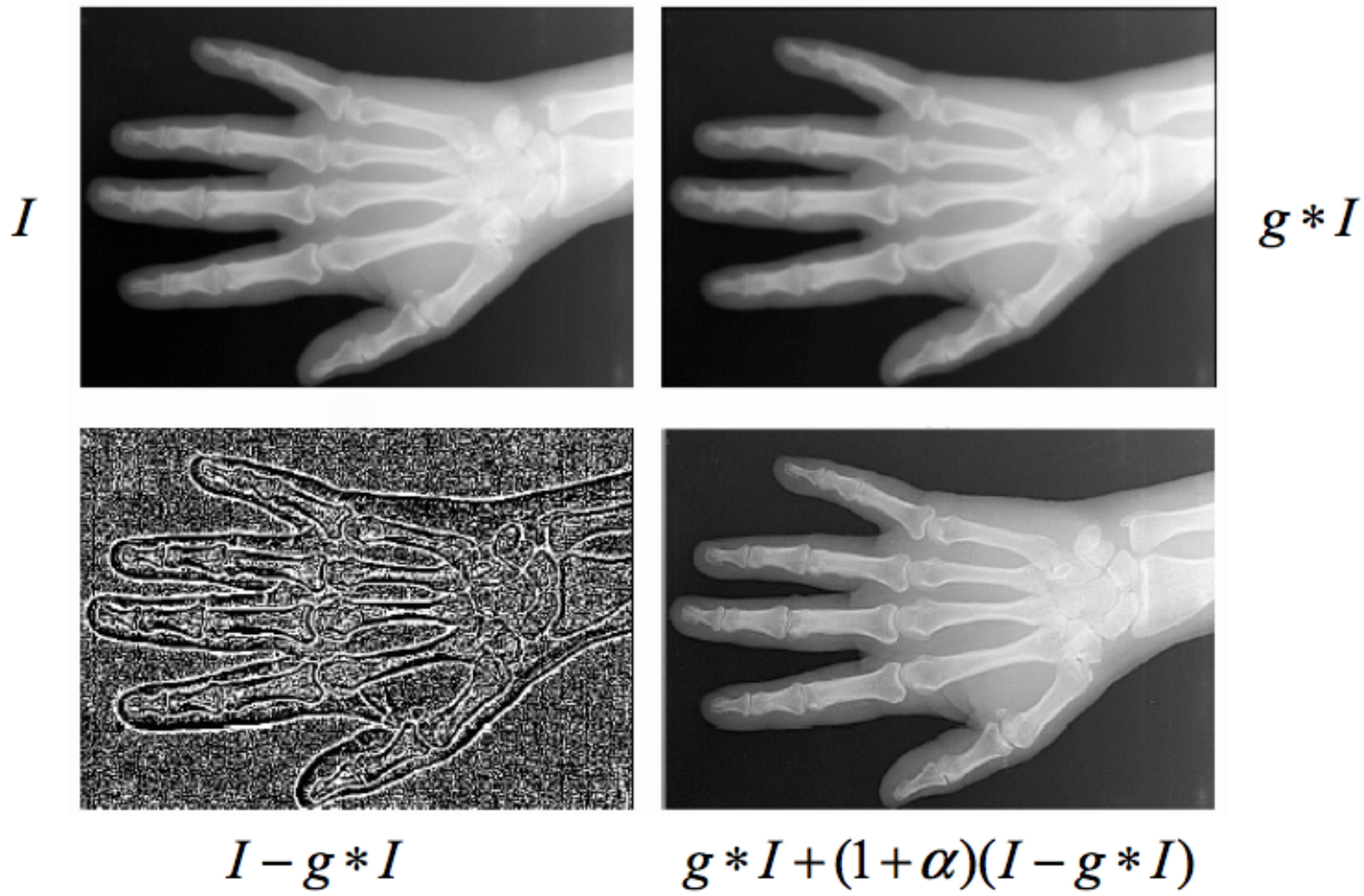
Subtracting a smoothed image from the original image leaves the edges (the high frequencies):



original I | smoothed $g * I$ | original - smoothed $I - g * I$

"Sharpening"

Places the enhanced edges on top of a smoothed original



Alterações locais e globais

Windowing enhances contrast only for a specific range of grey levels (not sensitive to edges)

- strong edges with already good contrast are further enhanced

Edge enhancement (such as sharp masking) only boosts features within a certain frequency band

- this frequency band is determined by filter size -- features outside that band are not enhanced (cannot see many scales at the same time)
- all grey value variations (within that band) are enhanced, even if they already had good contrast

original small filter: small detail large filter: large-scale variations

Tarefa de casa

- Exercício-programa 1