

Análise de Imagens - 2018

# **Operações Globais em Imagens :** **Transformações Geométricas e Transformadas**

## Transformações Geométricas

Levam o tom do *pixels* na posição  $(x_o, y_o)$  da *imagem origem*, para outra posição  $(x_d, y_d)$  do espaço em uma *imagem destino*

Aplicações:

matching de impressões digitais reconhecimento de padrões flexíveis

# Transformações Geométricas

o *pixels* muda a posição  $(x_0, y_0)$  na *imagem origem*,  
Mas é considerado o mesmo pois seu tom fica inalterado

**Translação, Rotação e Escala**

**Espelhamento ou reflexão**

**Deformações e *Morphing***

# Translação, Rotação e Escala



$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(a)

(b)

Coordenadas homogêneas

Exemplo de translação da imagem. (a) Imagem Original. (b) Imagem Transladada.

## Escala

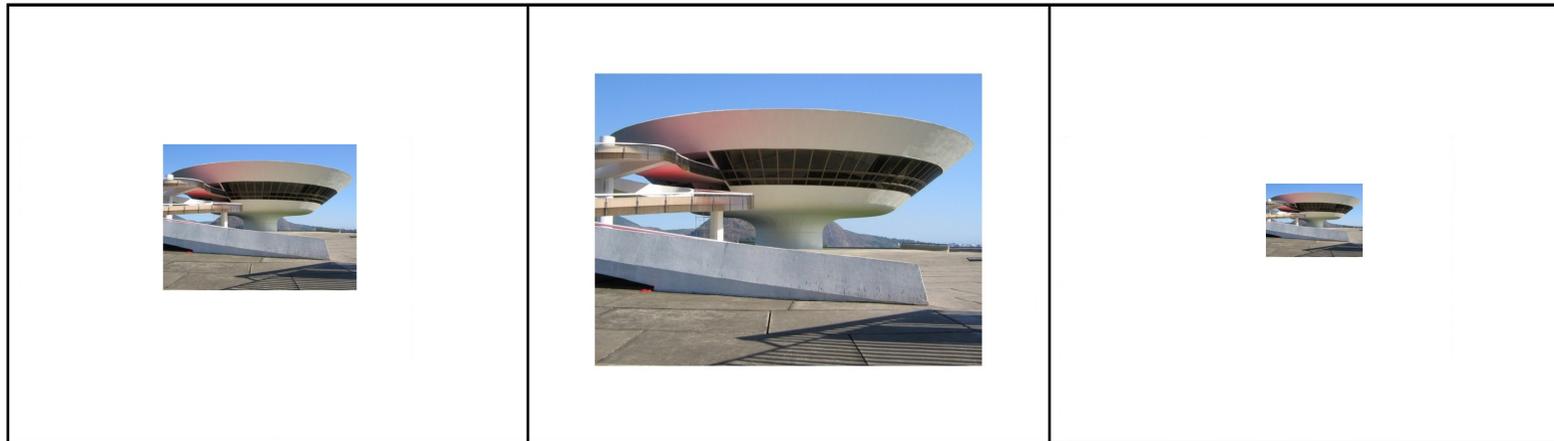
$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

Coordenadas  
homogeneas

(a)

(b)

(c)

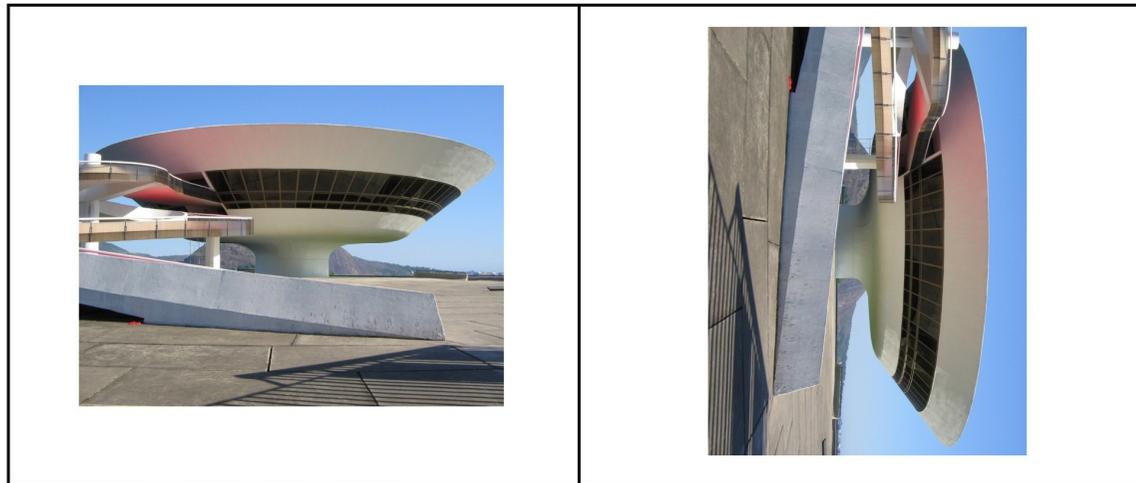


Exemplo de ampliação e redução da imagem. (a) Imagem Original. (b) Imagem Ampliada 2 vezes. (c) Imagem Reduzida pela metade

## Rotação

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta & 0 \\ \operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(c. homogêneas)

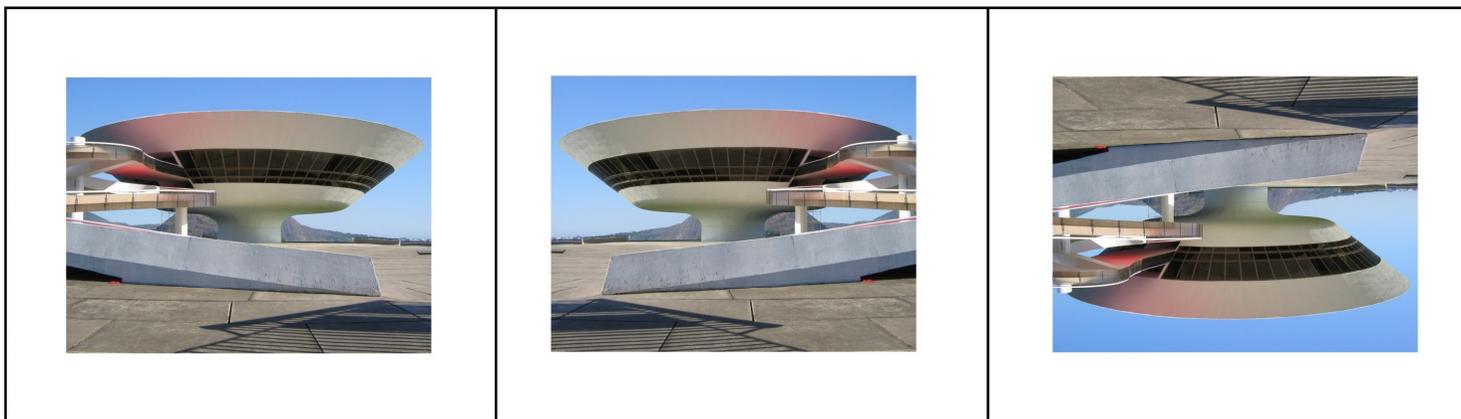


Exemplo de Rotação de  $90^\circ$  no sentido horário

## Espelhamento ou reflexão

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$



(a)

(b)

(c)

Exemplo de espelhamento. (a) Imagem Original. (b) Flip Horizontal.  
(c) Flip Vertical.

## Rotação em duas passadas:

$$x_d = x_o \cos \theta - y_o \operatorname{sen} \theta$$

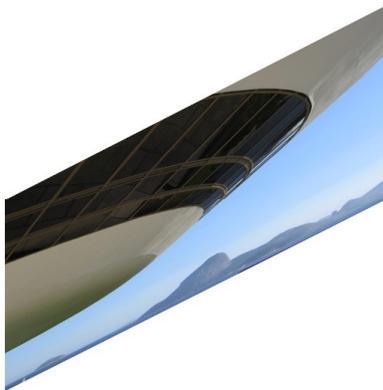
$$y_d = x_o \operatorname{sen} \theta + y_o \cos \theta$$

$$y_d = x_d \operatorname{tg} \theta + y_o \cos \theta$$

$$x_d = x_o \cos \theta - y_d \operatorname{tg} \theta$$



(a)



(b)



(c)

Rotação de 45 graus da imagem. (a) Imagem original. (b) 1ª passada.  
(c) Resultado final (2ª passada).

# Manipulações básicas para **matching** em reconhecimento de padrões flexíveis

- São geralmente feitas fazendo o casamento de 3 pontos básicos nas 2 imagem existentes a serem “casadas” .
- Assim se acha uma *transformação afim entre as duas imagens*.
- *Ou quais os valores das operações elementares anteriores que transformam uma imagem na outra.*

## Deformações x *Movimentos de corpo rígido*

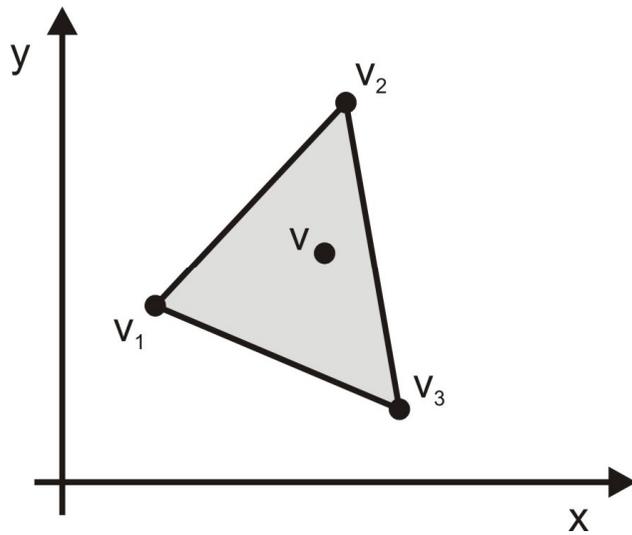
Geralmente se considera que a imagem está envolvida por segmentos de reta com seus respectivos vértices e se altera **a forma** dos objetos a partir de mudanças de posição dos segmentos envolventes.

O mais simples é envolver por 2 ***box*** ou triângulos.

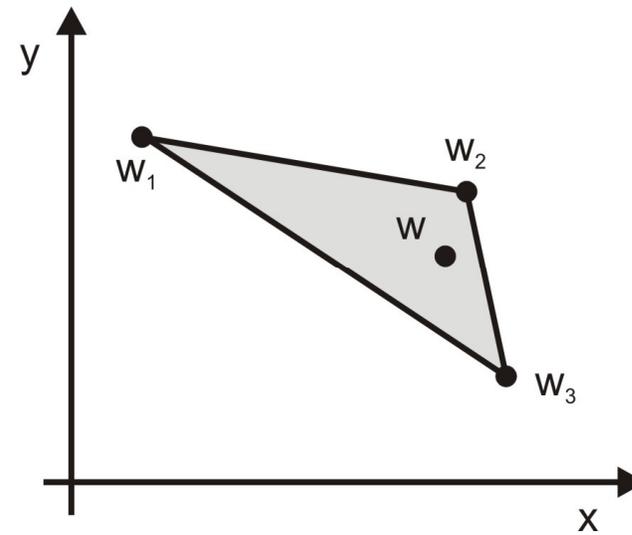
Se no processo de **deformação não houver alteração da distancia entre qualquer ponto**, o processo é chamado de transformação de *corpo rígido*.

***Caso contrario, se houver, é chamado de uma Deformação.***

## Deformações genéricas: como calcular

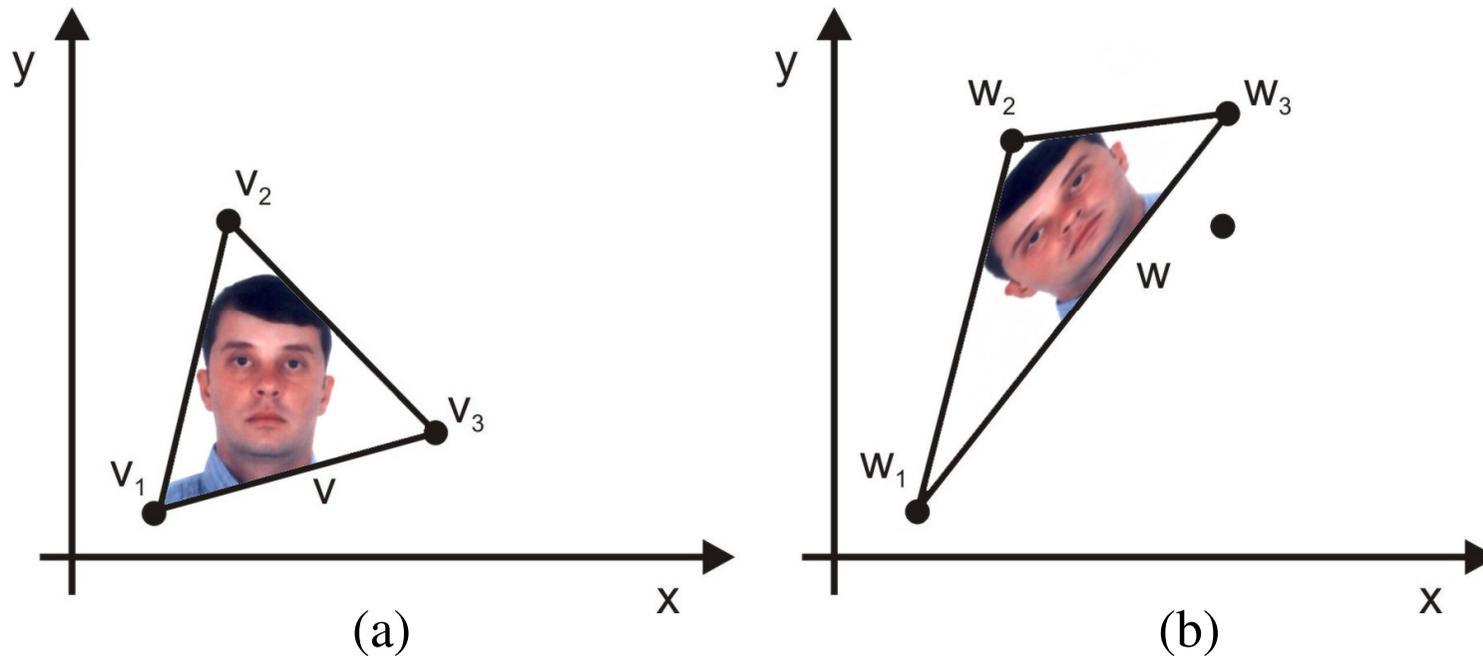


(a)



(b)

Deformação simples de uma região triangular com pontos de vértice não-colineares. (a)  $v = c_1v_1 + c_2v_2 + c_3v_3$ . (b)  $w = c_1w_1 + c_2w_2 + c_3w_3$



Triângulo inicial com uma imagem. (a) Imagem original:  $v = c_1v_1 + c_2v_2 + c_3v_3$ . (b) Imagem transformada:  $w = c_1w_1 + c_2w_2 + c_3w_3$ .

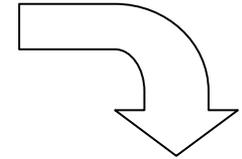
6 valores conhecidos: As coordenadas horizontais e verticais dos pontos:  $w_1, w_2, w_3$  e  $v_1, v_2, v_3$

6 valores desconhecidos:  $c_1$  a  $c_3$  nas direções x e y

Com os 6 valores conhecidos:  $w_1, w_2, w_3$  e  $v_1, v_2, v_3$  monta-se um sistema de 6 equações

$$w_i = M v_i + b.$$

$$M = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} \quad b = \begin{vmatrix} b_1 \\ b_2 \end{vmatrix}$$



$$w_{1x} = m_{11} v_{1x} + m_{12} v_{1y} + b_1$$

$$w_{1y} = m_{21} v_{1x} + m_{22} v_{1y} + b_2$$

Para se obter as 6 incognitas

$$w_{3x} = m_{11} v_{3x} + m_{12} v_{3y} + b_1$$

$$w_{3y} = m_{21} v_{3x} + m_{22} v_{3y} + b_2$$

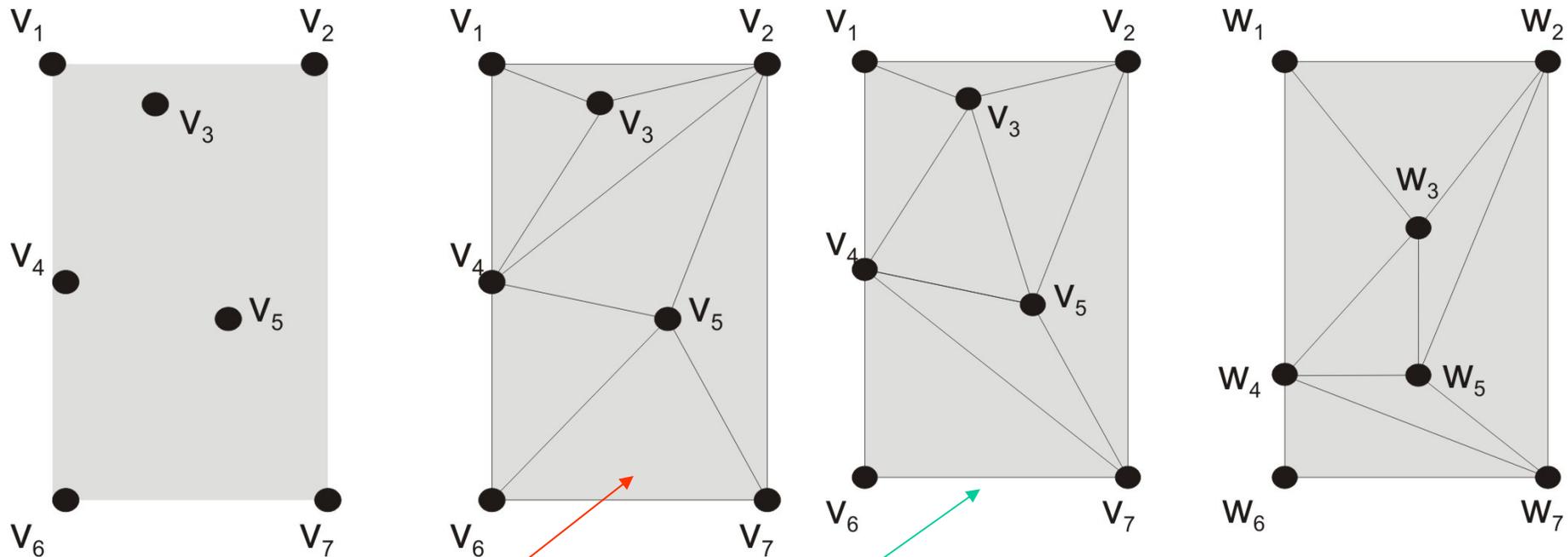
$$w_{2x} = m_{11} v_{2x} + m_{12} v_{2y} + b_1$$

$$w_{2y} = m_{21} v_{2x} + m_{22} v_{2y} + b_2$$

Resolvendo o sistema de 6 equações a 6 incognitas obtém-se a transformação que leva as **deformações** de cada triângulo da **imagem inicial** na imagem final;

E isso é feito para todos os pixels do interior dos triângulos.

Pode-se repartir uma imagem em várias regiões triangulares e deformar cada região de uma maneira diferente, mas similaridades de “triangularização” são importantes:



Fazer inadequadas ou adequadas Triangulações depende das imagens a serem casadas.

Isso se chama fazer uma “malha” ou um “grid” nos objetos.

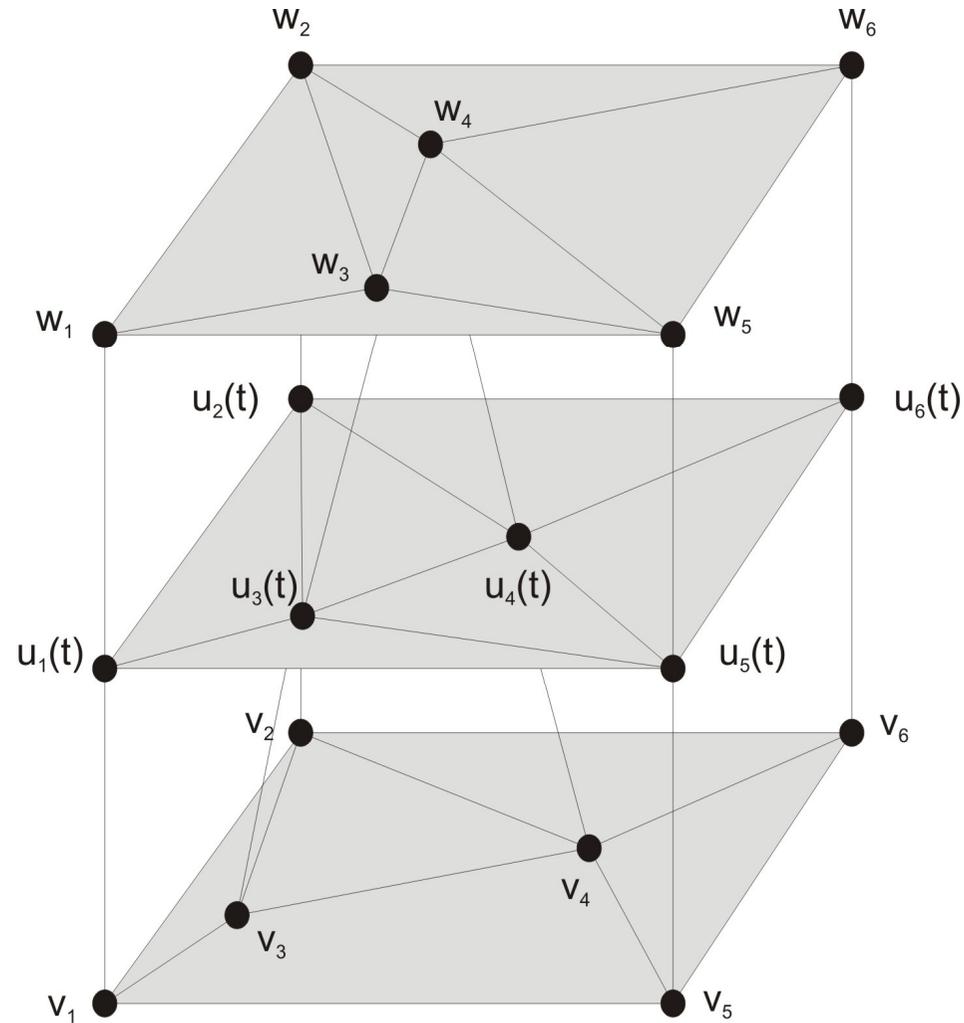
## Deformações Dependentes do Tempo

É um conjunto de deformações geradas quando os pontos de vértice da imagem inicial são **movidos continuamente ao longo do tempo** desde suas posições originais até posições finais especificadas.

Ao longo de diversos frames de um video, por exemplo.

$$u_i(t) = (1 - t)v_i + t w_i$$

Considera-se essas quando globais se forem feitas em toda a imagem que está envolvida pela Malha.



Triangulação dependente do tempo de uma região

Considera-se essas quando globais se forem feitas em toda a imagem que está envolvida pela Malha.

# Medição do Desempenho

Comparação entre imagem original e transformada.

Critérios de fidelidade: se a alteração causou perda de informação visual.

Podem ser:

quantitativos

ou

Subjetivos (qualitativos) .

# Critérios de comparação objetivos

## Funções de avaliação:

- Erro Total.
- Raiz Quadrada do Quadrado da Média dos Erros (Root Mean Square Error -  $e_{rms}$ ).
- Relação Sinal Ruído (*Signal To Noise Ratio* -  $SNR_{ms}$  SNR).
- Relação Sinal Ruído de Pico (*Peak Signal to Noise Ratio* - PSNR).

Sendo  $F(x, y)$  a imagem original e  $G(x, y)$  a imagem modificada, tem-se:

### **Erro Total ou absoluto:**

$$e_t = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |G(x, y) - F(x, y)|$$

### **Raiz Quadrada do Quadrado da Média dos Erros:**

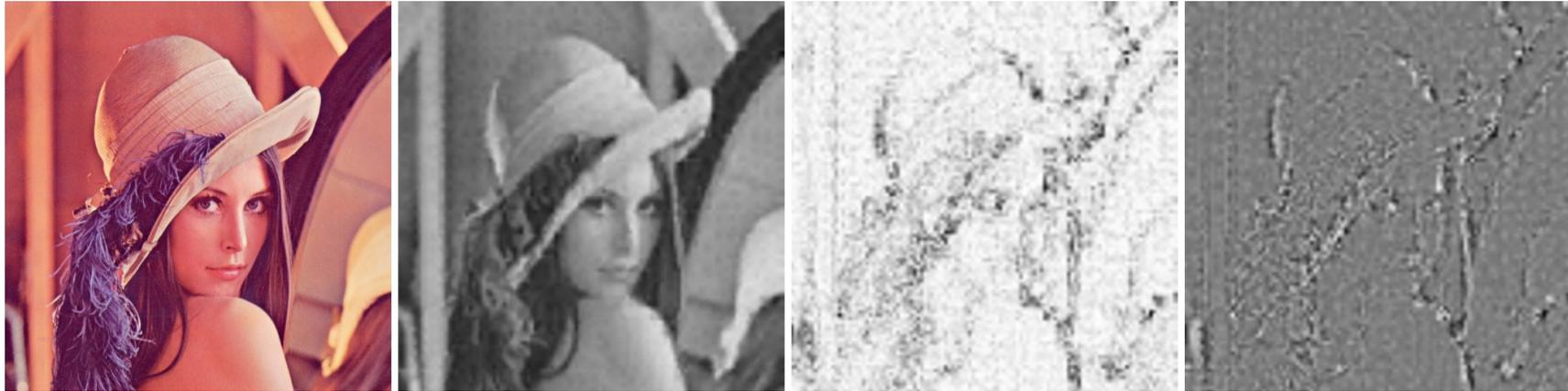
$$e_{rms} = \sqrt{\left[ \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [G(x, y) - F(x, y)]^2 \right]}$$

## Razão ou Relação Sinal Ruído:

$$SNR_{rms} = \sqrt{\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} e(x, y)^2}} = \sqrt{\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [G(x, y) - F(x, y)]^2}}$$

## Relação Sinal Ruído de Pico:

$$PSNR = 20 \log_{10} \left( \frac{2^n - 1}{e_{rms}} \right)$$



**(a) imagem Lena original; (b) imagem comprimida e reconstruída usando compressão fractal; (c) imagem de diferença absoluta ampliada e (d) imagem de diferença relativa ampliada**

**Erro rms = 9,7622**

**SNR rms = 10,4823**

**PSNR (dB) = 28,3398**

# Critérios subjetivos

Notas de avaliação subjetiva.

<b>Valor</b>	<b>Avaliação</b>	<b>Descrição</b>
1	Excelente	Extrema alta qualidade.
2	Muito Bem	Alta qualidade, visualização agradável, a interferência não é desagradável.
3	Passável	Qualidade aceitável. A interferência não é desagradável.
4	Marginal	Qualidade pobre. Você desejaria melhorá-la. A interferência é de alguma forma desagradável.
5	Inferior	Muito pobre, mas ainda assim você poderia vê-la. Desagradável interferência presente.
6	Não Usável	Muito ruim, que não você não conseguiria ver.

# **Voltando a considerar Operações Globais em uma banda de uma imagem multi bandas:**

**As** Operações Globais **são chamadas**  
**de Transformadas quando**  
**mudam o espaço de referencia**  
**das imagens .**

# Operações Globais : Transformadas

Cada *pixel* da imagem resultante depende de um processamento realizado em **todos os *pixels* da imagem original**.

As diversas Transformadas mudam a forma de representação do espaço, usualmente uma representação 2D espacial para outras formas como as:

Transformadas Fourier,

Transformadas Cosenos (usada para codificação);

Transformadas Gabor,

Transformadas Wavelet,

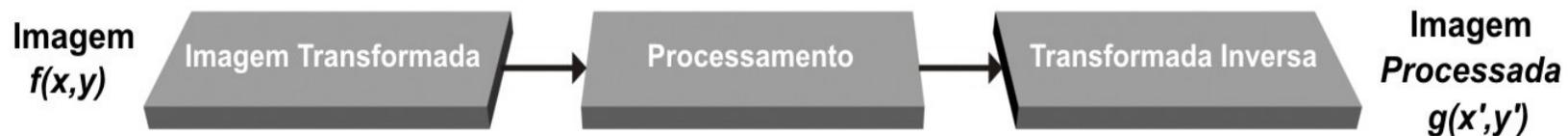
Transformadas Hough,

Transformadas Fourier-Mellin,

Transformada por Funções iterativas (IFS) ou fractal, Etc..

# Transformadas para o Domínio da Frequência

- 1- A imagem é **transformada do domínio espacial** para o da **frequência** (transformada de Fourier).
- 2- Operações (por exemplo filtragem) são realizadas nessa imagem.
- 3- Realiza-se o processo inverso, a imagem no domínio da frequência **volta para o domínio espacial**.



Esquema de processamento no domínio da frequência usando a transformação de imagens

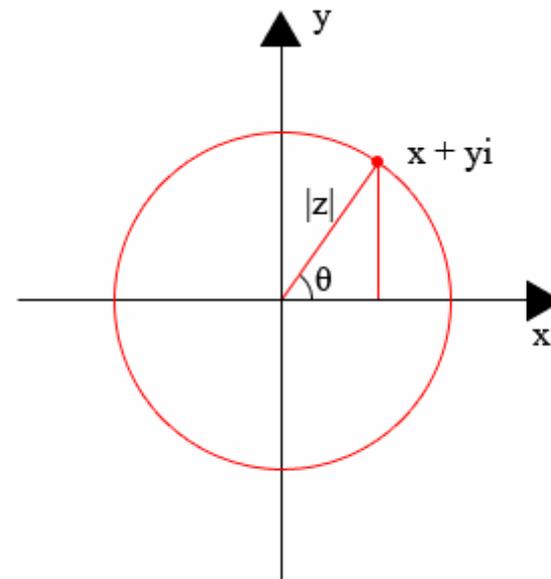
# Números Complexos

- São os elementos do conjunto  $\mathbf{C}$ , uma extensão do conjunto dos  $\mathbf{R}$ , onde existe um elemento que representa a raiz quadrada de -1 (chamado imaginário)
- Cada número complexo  $C$  pode ser representado na forma:  $a + b i$
- onde  $a$  e  $b$  são números reais conhecidos como *parte real* e *parte imaginária* de  $C$  e  $i$  é o imaginário (raiz quadrada de menos 1)

$$i = j = \sqrt{-1}$$

# Plano complexo

- Também chamado de **plano de Argand-Gauss**
- É uma representação do **conjunto dos números complexos,  $\mathbb{C}$** .
- Da mesma forma como a cada ponto da reta está associado um  $\mathbb{R}$ , o **plano complexo** associa o ponto  $(x,y)$  ao número complexo  $x + iy$ .



# Transformada de Fourier



**Jean-Baptiste Joseph Fourier** foi matemático e físico francês, iniciou estudos de decomposição de funções em séries trigonométricas.

A TF ou FT deve esse nome em sua homenagem.

Em **processamento de sinais** usa-se a notação  $j$  para o imaginário  $i$

A **transformada de Fourier**  $F(u)$ , de uma função contínua  $f(x)$  de uma variável real  $x$  pode ser definida como:

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi u x] dx \quad \text{onde } j = \sqrt{-1}$$

A partir de  $F(u)$ , pode-se obter  $f(x)$  através da transformada **inversa** de Fourier:

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

Essas duas equações são chamadas de **par de transformada de Fourier** e podem existir se ambas forem integráveis e se  $f(x)$  for contínua.

A transformada de Fourier de uma função  $f(x)$  é uma função **complexa**, *i.e.* tem parte **real** e **imaginária**:

$$F(u) = R(u) + j I(u) \quad j = \sqrt{-1}$$

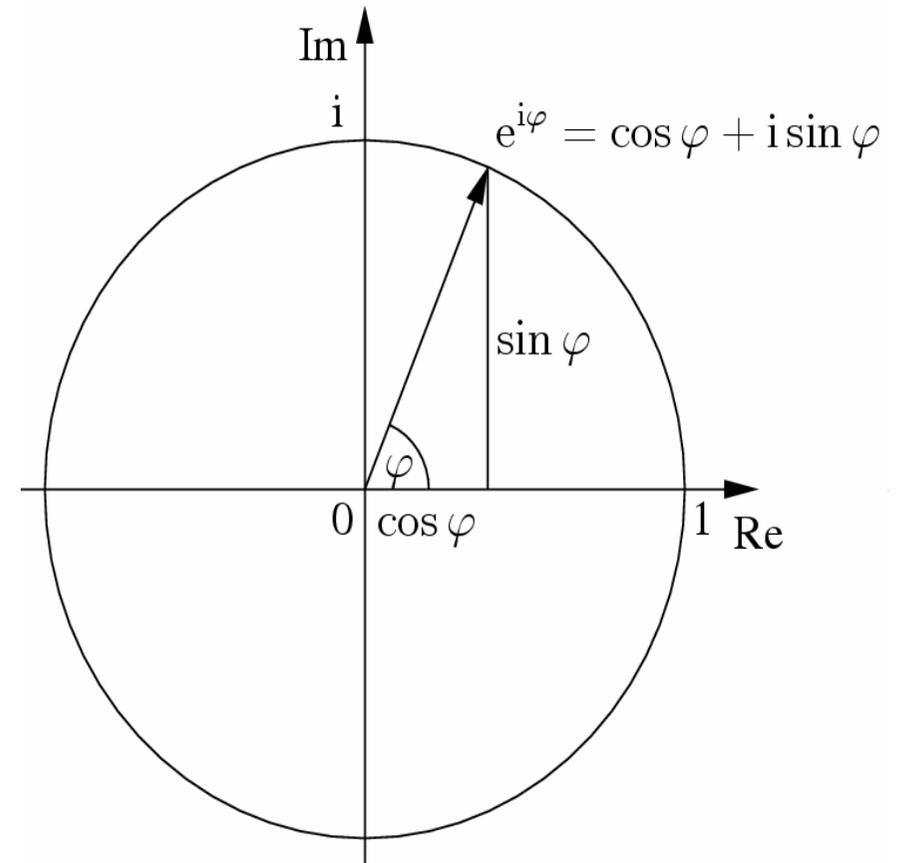
Como outras funções complexas pode ser **escrita** na forma também na **forma exponencial**:

$$F(u) = |F(u)| e^{j\theta(u)} = |F(u)| \exp[j\theta(u)]$$

# Fórmula de Euler

- mostra a relação entre a função exp (o numero  $e$ ) senos e co-senos:

$$e^{ix} = \cos(x) + i \sin(x)$$



# Algumas transformadas de Fourier

$$\delta(x) = 0 \text{ se } x \neq 0$$

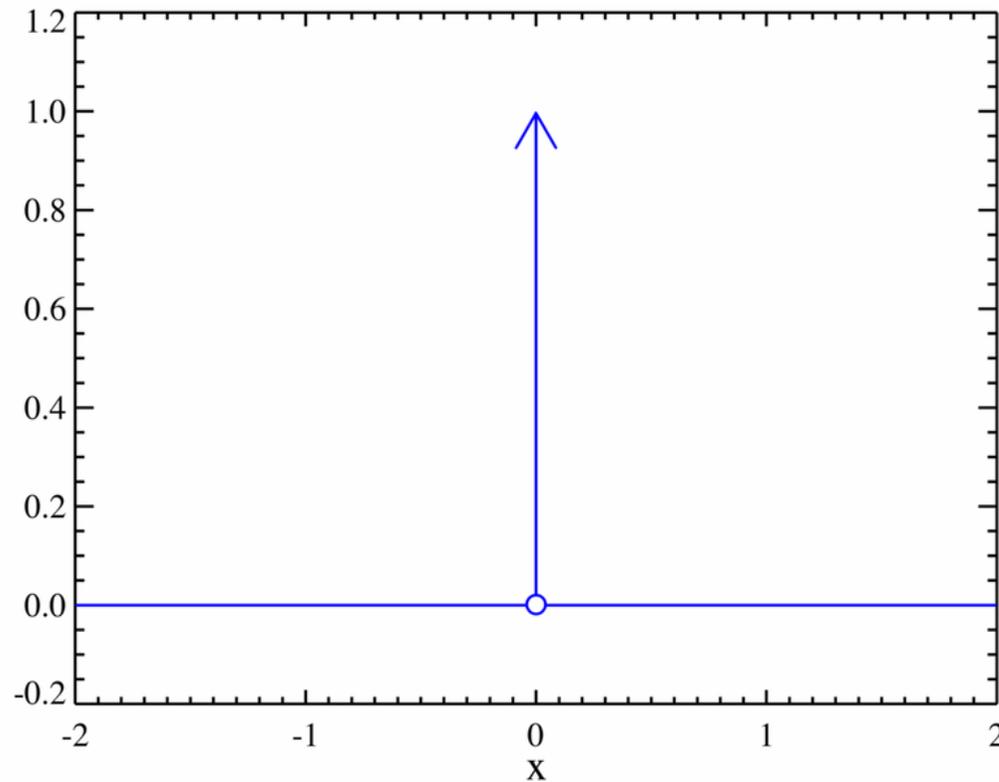
$$\delta(0) = \infty$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

Delta de Dirac

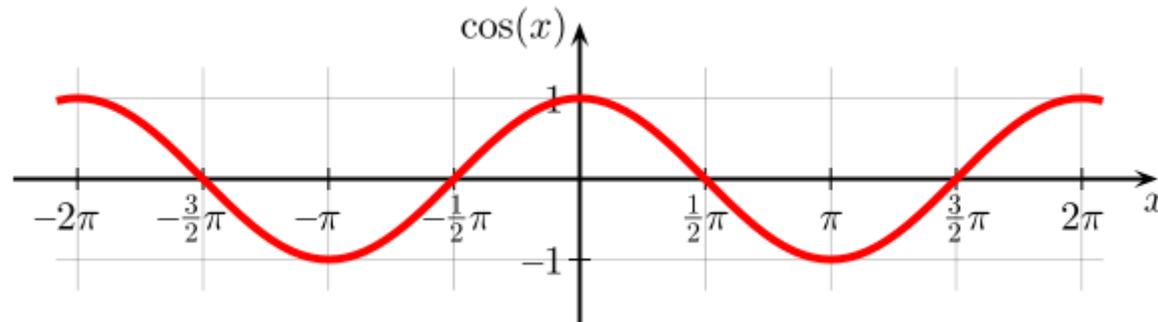
(introduzida por Paul Dirac)

$f(x)$	$F(u)$
$\delta(x)$	1



$$f(x) = \delta(x)$$

# Algumas transformadas de Fourier



$f(x)$	$F(u)$
$\cos(u_0 x)$	$\pi[\delta(u - u_0) + \delta(u + u_0)]$

Função co-seno

# Convolução

- É uma operação que vem de processamento de sinais e sistemas lineares

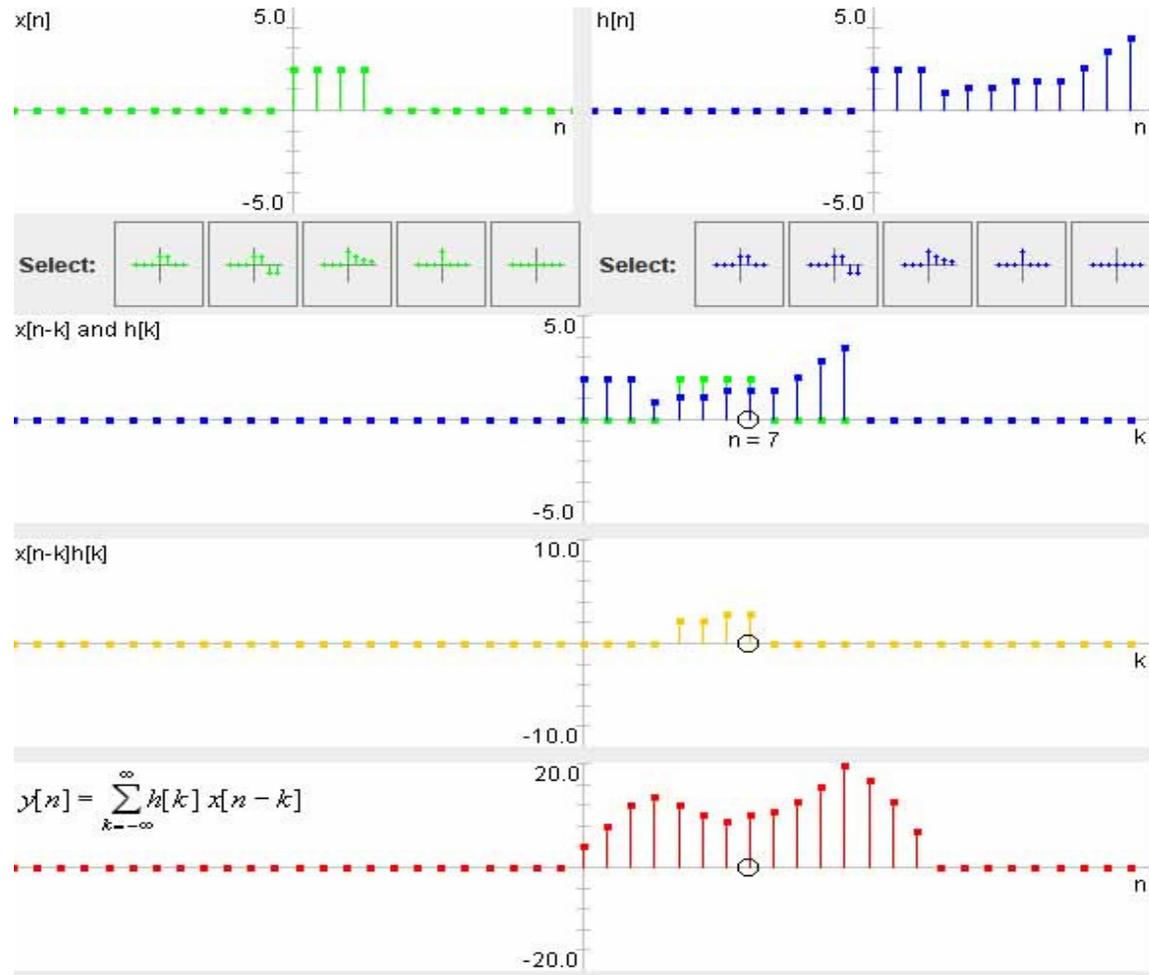
# Convolução em funções discretas

- O comportamento de um sistema linear, em tempo discreto e invariante no tempo com sinal de entrada  $x[n]$  e sinal de saída  $y[n]$  é descrito pela soma da convolução.

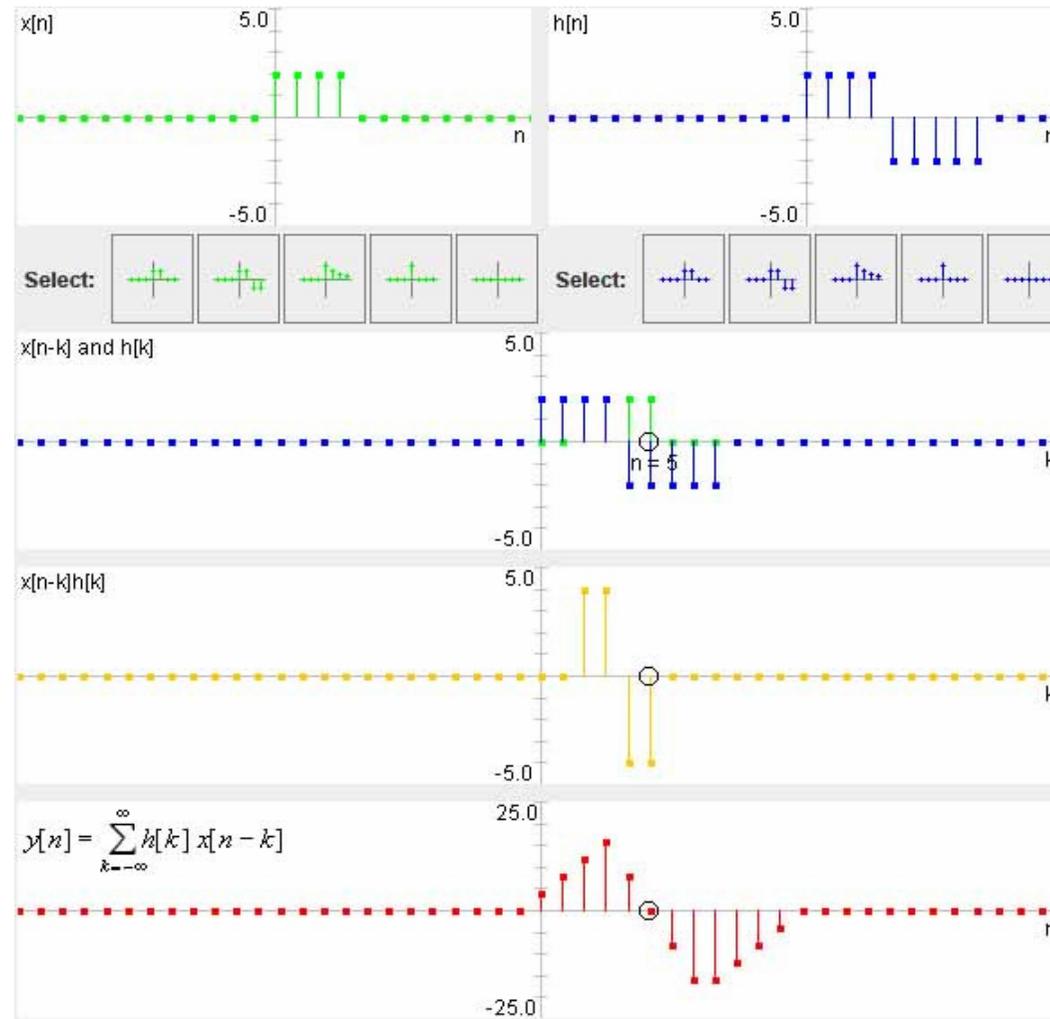
$$y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k]$$

- O sinal  $h(t)$ , assumido conhecido, é a resposta do sistema a uma entrada de impulso unitário.
- Para calcular a saída  $y(t)$  a um especificado  $t$ , primeiro o integrando  $h(v) x(t-v)$  é *computado como uma função de  $v$* . A integração relacionada a  $v$  é feita, resultando em  $y(t)$ .
- Primeiro, plota  $h(v)$  e o  $x(t-v)$  *deslocado no eixo  $v$ , onde  $t$  é fixo*. Segundo, multiplica os dois sinais e computa a área de sinal da função resultante de  $v$  para obter a plotagem  $y(t)$ . Essas operações podem ser repetidas para cada valor de  $t$  de interesse.

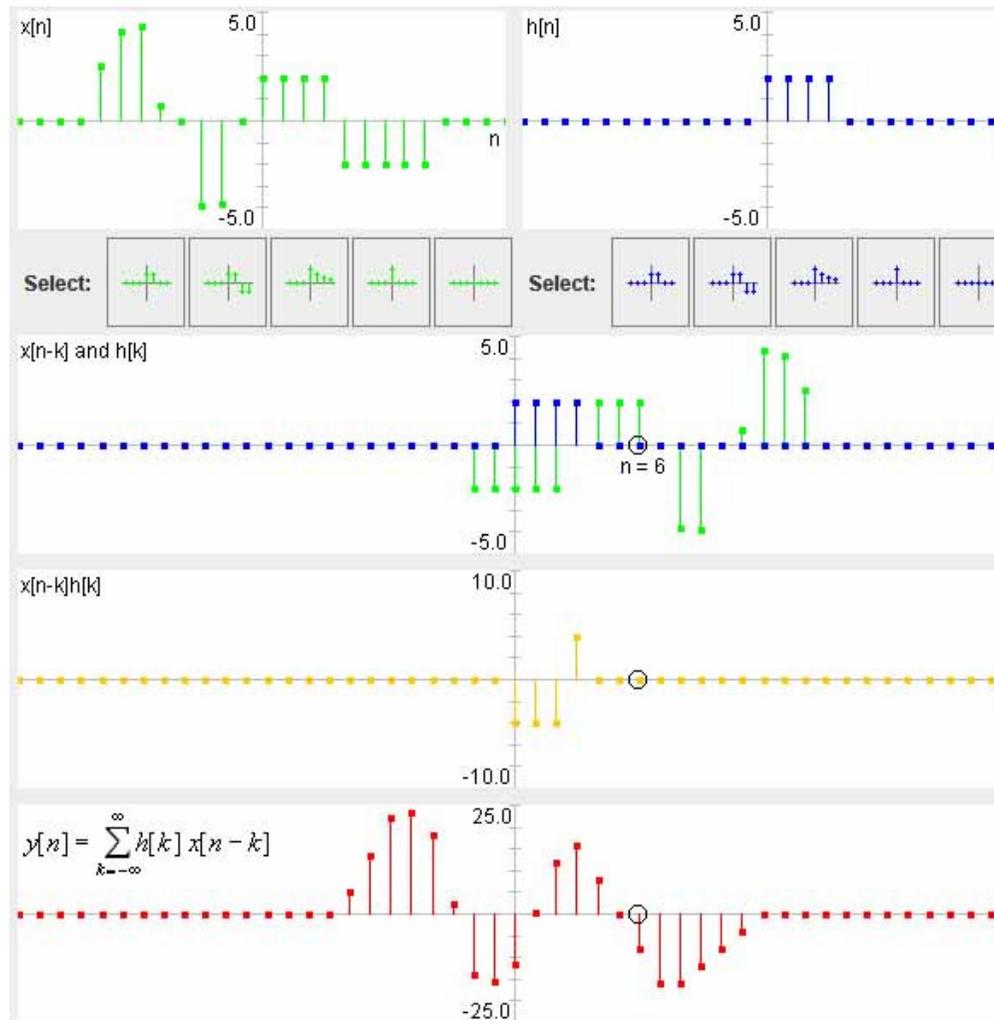
# Convolução em funções discretas - exemplo 1



# Convolução em funções discretas - exemplo 2



# Convolução em funções discretas - exemplo 3

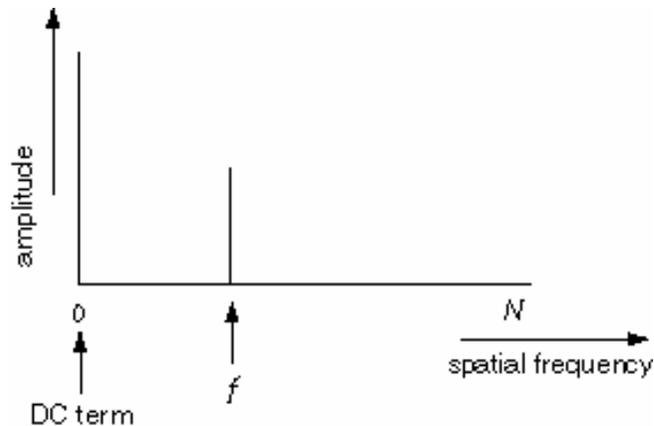


# Transformada de Fourier

- A magnitude da sinusóide corresponde a seu contraste, ou a diferença entre os picos mais escuros e mais claros da imagem. Uma magnitude negativa representa um contraste reverso, i.e. O claro se torna escuro e vice-versa. A fase representa como a onda é deslocada com relação a origem, representando o quanto a sinusóide está deslocada para a esquerda ou direita.
- Uma transformada Fourier codifica uma série completa de sinusóides através de uma faixa de frequências espaciais a partir do zero (i.e. sem modulação, brilho médio da imagem completa) durante todo o caminho até a "frequencia de nyquist", i.e. A frequência espacial de maior brilho que pode ser codificada na imagem digital, a qual está relacionada a resolução, ou tamanho dos pixels. A transformada Fourier codifica todas as frequências espaciais presentes em uma imagem simultaneamente, como segue. Um sinal contendo apenas uma frequência espacial única de frequência  $f$  é plotado como um pico único no ponto  $f$  ao longo do eixo de frequência espacial, a altura do pico correspondente a amplitude, ou contraste daquele sinal sinusoidal.

# Transformada de Fourier

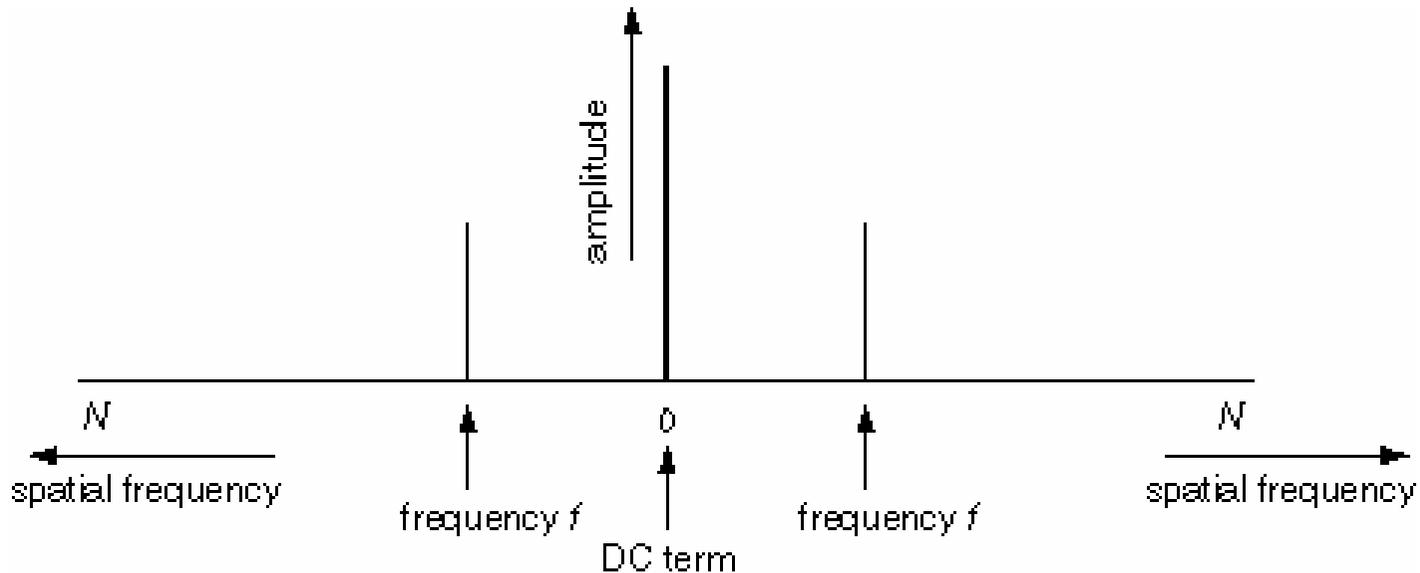
- O “termo DC” correspondente a frequência zero, representa o brilho médio ao longo de toda a imagem. Um termo zero DC significaria uma imagem com brilho médio de zero, o que poderia significar a sinusóide alternada entre valores positivos e negativos na imagem de brilho. Como não existe brilho negativo, todas as imagens reais tem um termo DC positivo.



- A transformada Fourier também plota uma imagem espelho da frequência espacial através da origem, com frequência espacial aumentando em ambas as direções a partir da origem. Essas duas plotagens são sempre reflexões da imagem em espelho uma da outra, com picos idênticos em  $f$  e em  $-f$  como mostrado a seguir.

# Transformada de Fourier

- O que foi mostrado anteriormente é a transformada Fourier de uma única linha de varredura da imagem sinusoidal, que é um sinal unidimensional. Uma transformada Fourier completa 2D desenvolve uma transformada 1D em cada linha de varredura ou linha da imagem, e outra transformada 1-D em cada coluna da imagem, produzindo uma transformada Fourier 2-D do mesmo tamanho da imagem original.



Transformada de Fourier **bidimensional**:

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

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

Tem-se o **espectro de Fourier**, o **ângulo de fase** e o **espectro da potência bidimensionais**:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

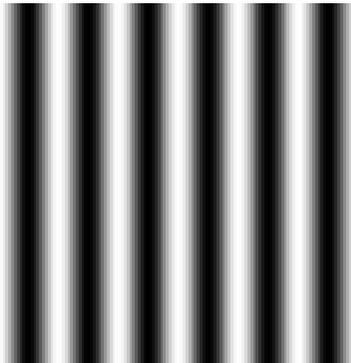
$$\phi(u, v) = \tan^{-1} [I(u, v) / R(u, v)]$$

$$P(u, v) = R^2(u, v) + I^2(u, v)$$

# Transformada de Fourier

- A imagem abaixo mostra uma imagem com intensidade variando de forma senoidal, e sua transformada de Fourier em duas dimensões, apresentada aqui como uma imagem. Cada pixel da imagem da T. Fourier é um valor da frequência espacial, a magnitude daquele valor é codificado pelo brilho do pixel. Nesse caso, tem um pixel bem no centro - esse é o termo DC, ladeado por dois pixels nos dois lados do centro, que codifica o padrão sinusoidal. Quanto mais intenso os picos na imagem da T. Fourier, maior o contraste na imagem original. Como tem apenas um componente nessa imagem, todos os outros valores na imagem da T. Fourier são zero, mostrados em preto.

- **Imagem com intensidade variando de forma senoidal**



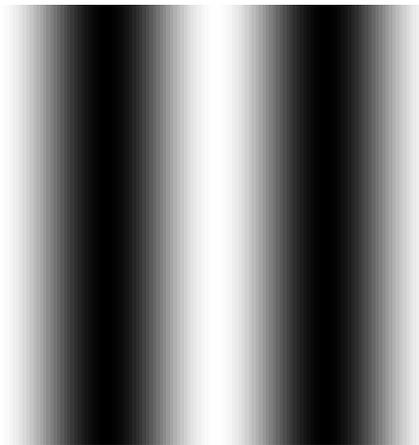
- **Transformada de Fourier**



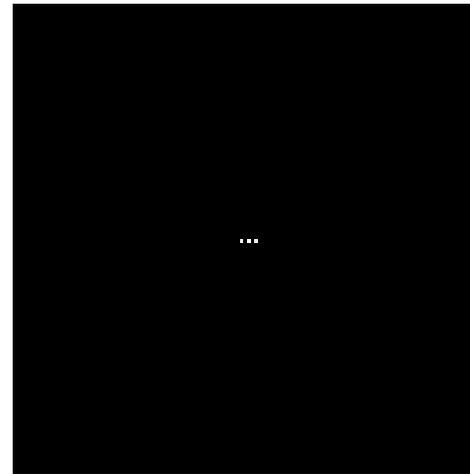
## Transformada de Fourier

- Aqui tem-se outra imagem com intensidade variando de forma senoidal, dessa vez com uma frequência espacial menor, junto com sua transformada de Fourier de 2D mostrando três picos como antes. Mas dessa vez os picos representando a senóide estão mais perto ao termo DC central, indicando uma frequência espacial menor.

**Imagem de brilho**



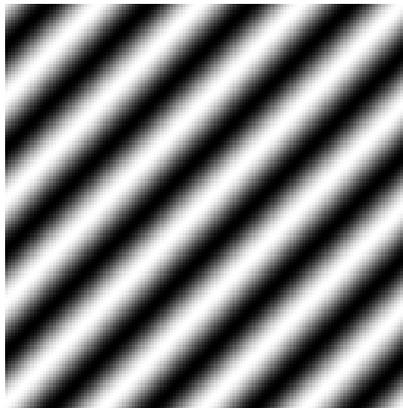
**Transformada Fourier**



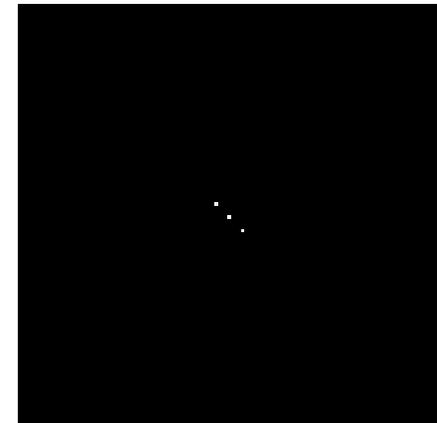
## Transformada de Fourier

- A imagem da T. de Fourier codifica exatamente a mesma informação da imagem original, exceto por expressar não mais espacialmente. A transformada inversa de Fourier da imagem T. de Fourier produz uma réplica exata pixel-a-pixel da imagem original.
- A orientação da sinusóide se correlaciona com a orientação dos picos na T. de Fourier relativa ao ponto DC central. Nesse caso um padrão sinusoidal cria um par inclinado de picos na imagem Fourier.

Imagem de brilho



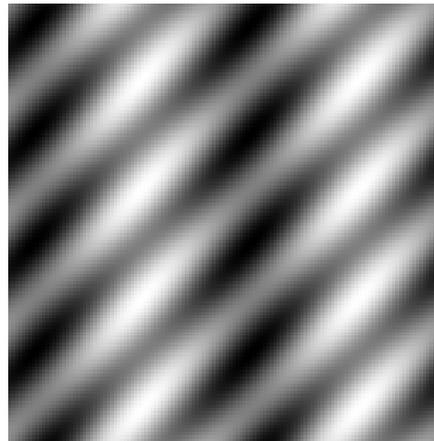
Transformada Fourier



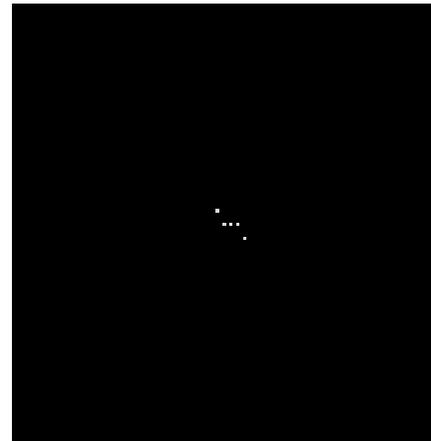
# Transformada de Fourier

- Coeficientes diferentes Fourier combinam adaptativamente para produzirem padrões de combinação. Por exemplo, a imagem sinusoidal mostrada abaixo é computada como a soma das sinusóides inclinadas mostrada acima, e a sinusóide vertical de frequência espacial inferior mostrada acima desta.

Imagem de brilho



Transformada Fourier

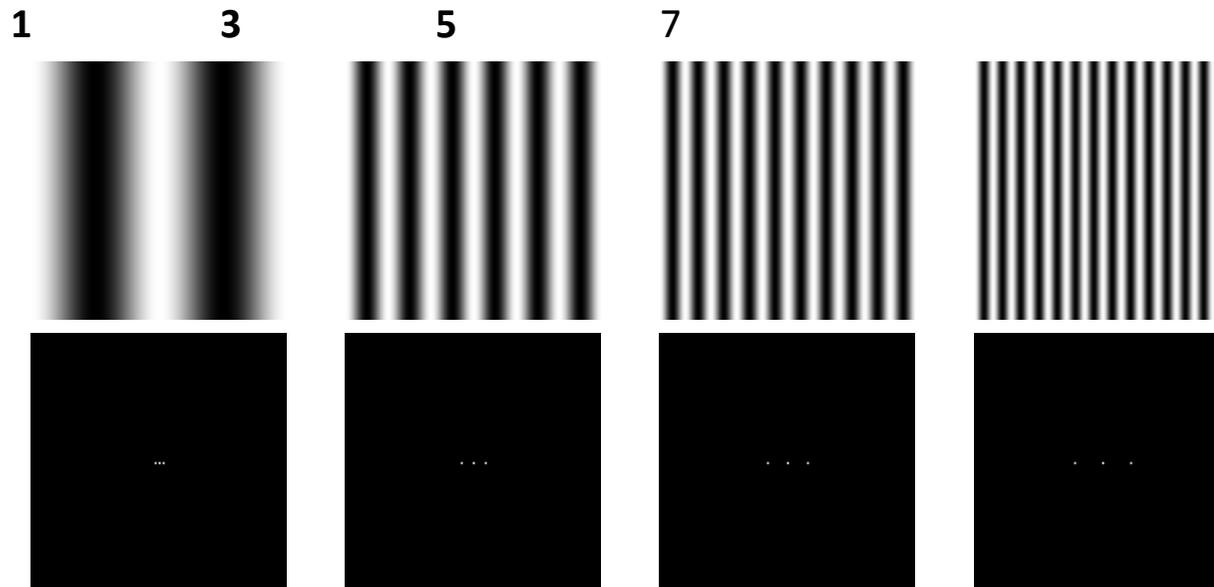


- **As imagens de brilho e as de Fourier são completamente intercambiáveis, porque elas contém exatamente a mesma informação.** A imagem de brilho combinada mostrada acima poderia ter sido produzida por uma adição pixel a pixel de duas imagens de brilho, ou por uma adição pixel a pixel das transformadas de Fourier correspondentes, seguida por uma transformada inversa para retornar ao domínio do brilho. Nas duas formas o resultado seria exatamente idêntico.

# Transformada de Fourier

## Harmônicas superiores e efeito “Ringing” (repique)

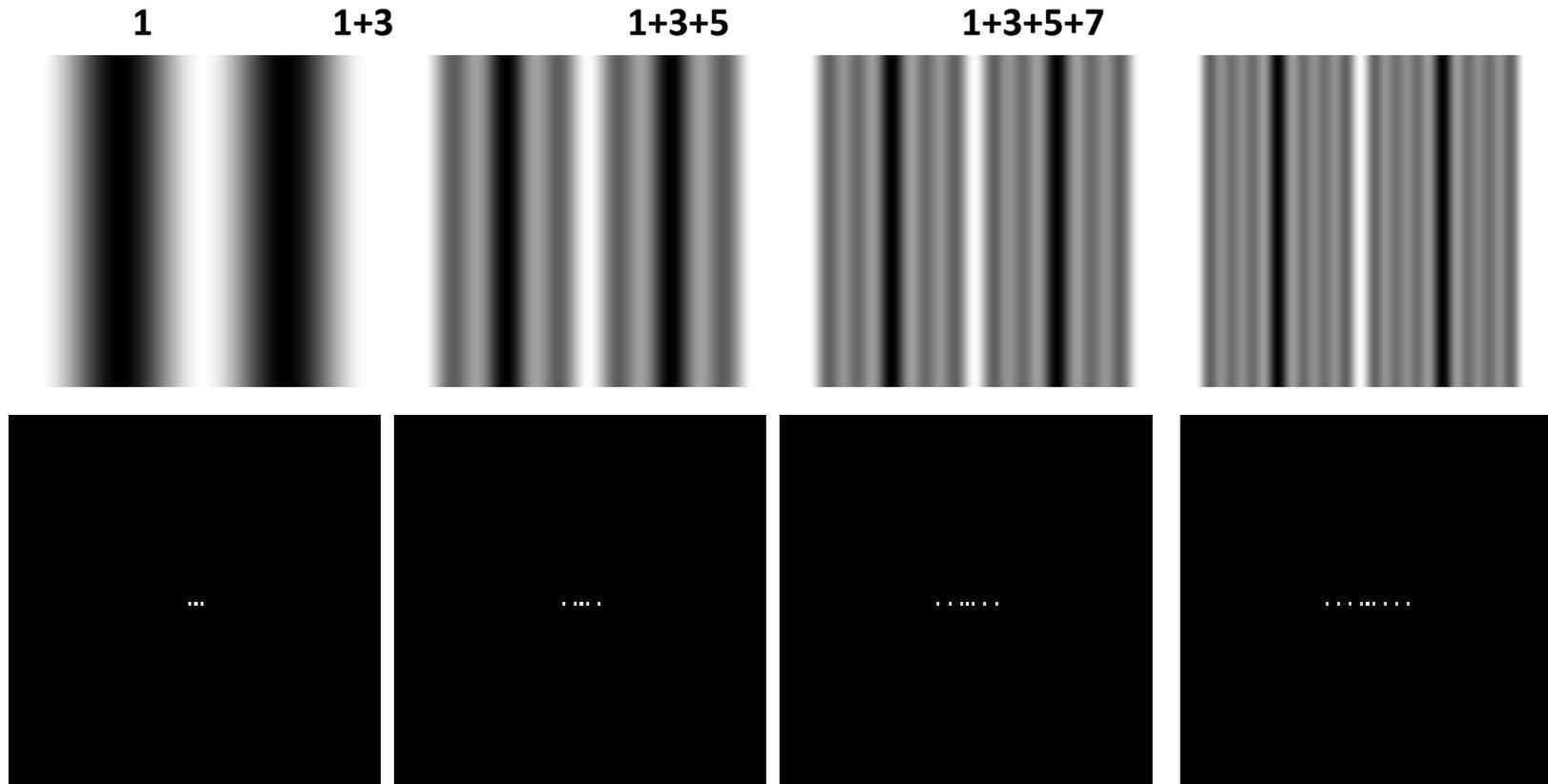
- O conjunto base para a transformada de Fourier é a função senoidal suave, que é otimizada para expressar formas arredondadas suaves. Mas a transformada de Fourier pode na verdade representar qualquer forma, até mesmo formas retilíneas duras com extremidades agudas, que são as mais difíceis de expressar em código Fourier, porque eles precisam muitos termos de ordens muito superiores, ou harmônicas maiores.
- A figura abaixo mostra quatro imagens de brilho de frequência espacial 1, 3, 5, e 7. A primeira, de frequência 1, é a fundamental, e as outras são harmônicas superiores àquela fundamental, porque elas são múltiplos inteiros da frequência fundamental. Essas são na realidade “harmônicas ímpares” na fundamental, e cada uma exibe uma banda vertical de brilho através do centro da imagem. A transformada de Fourier para cada um desses padrões é mostrado abaixo.



# Análise de Fourier

## Harmônicas superiores e efeito “Ringing”

- A tabela abaixo mostra o resultado de adicionar progressivamente harmônicas maiores à fundamental. Note como a banda vertical central se torna mais aguda e mais forte com cada harmônica superior adicional, enquanto o fundo cai em direção a um campo escuro uniforme. Note também como as harmônicas superiores produzem picos nas imagens Fourier que se espalham na mais longe da fundamental, definindo um padrão periódico no espaço da frequência.

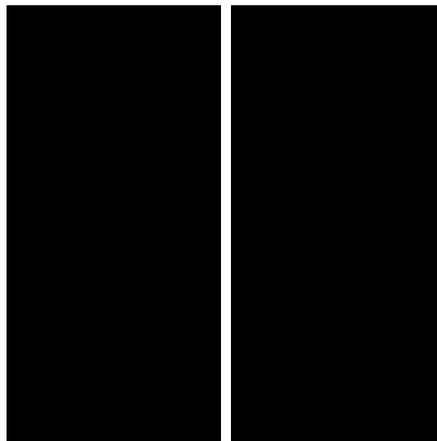


# Transformada de Fourier

## Harmônicas superiores e efeito “Ringing”

- A imagem abaixo mostra o que aconteceria se esse processo fosse continuado até a frequência de Nyquist – iria produzir uma listra vertical na imagem de brilho, com extremidades agudas, i.e. Uma “onda quadrada” em brilho ao longo da dimensão x. A transformada Fourier dessa imagem exibe um série "infinita" de harmônicas ou termos de ordens superiores, embora isso não vá até o infinito devido a resolução finita da imagem original. Isso é como a transformada de Fourier codifica imagens tipo onda quadrada aguda como a soma de uma série de sinusóides suaves.

Imagem de brilho



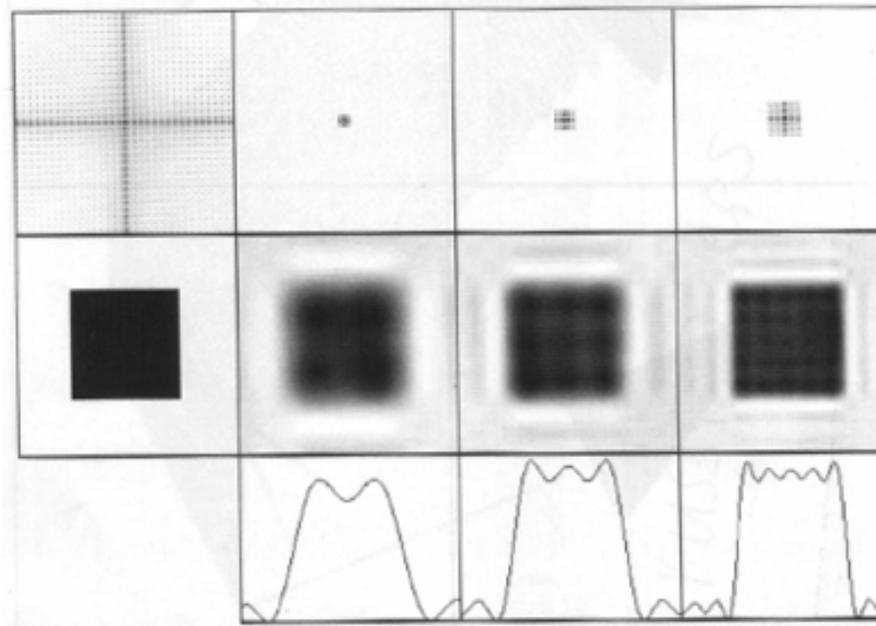
Transformada Fourier



# Transformada de Fourier

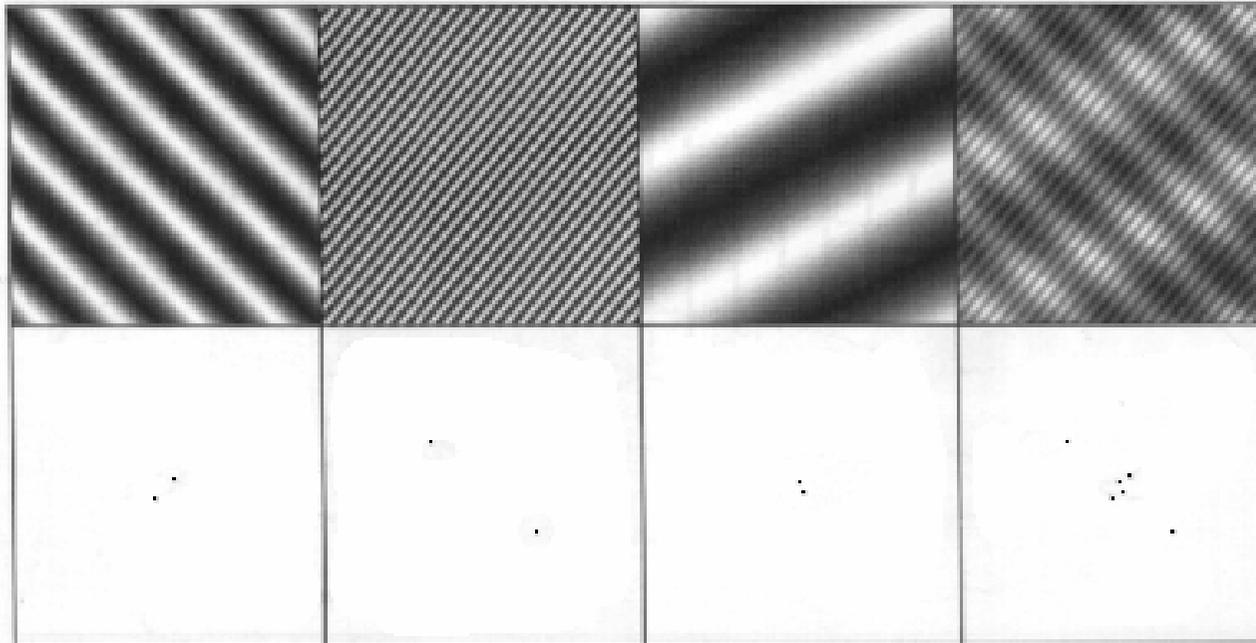
## Operação de Filtragem

Imagens no domínio espacial estão na linha do meio, e seus espaços de frequência são mostrados na linha superior. A linha inferior mostra a variação de brilho da linha horizontal através do centro de uma imagem.



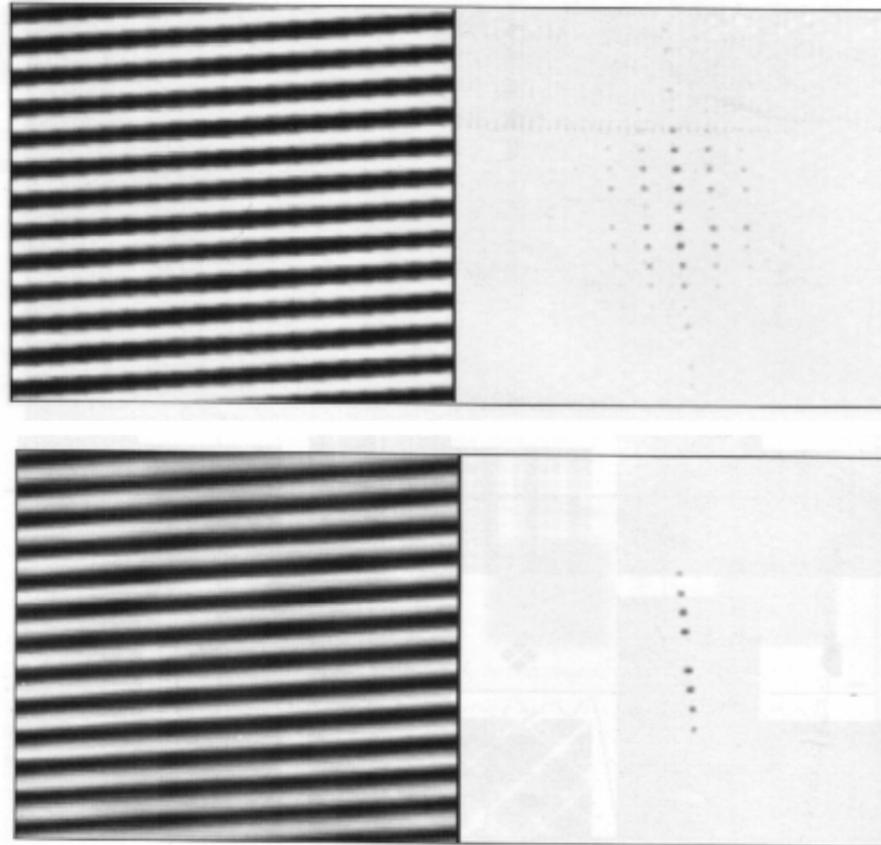
# Transformada de Fourier

Imagens com variações sinusoidais em brilho: as três primeiras são representadas por dois pontos. A posição e orientação daqueles pontos tem a ver com o que a imagem original se parece. A quarta imagem é a soma das três primeiras.



## Transformada de Fourier

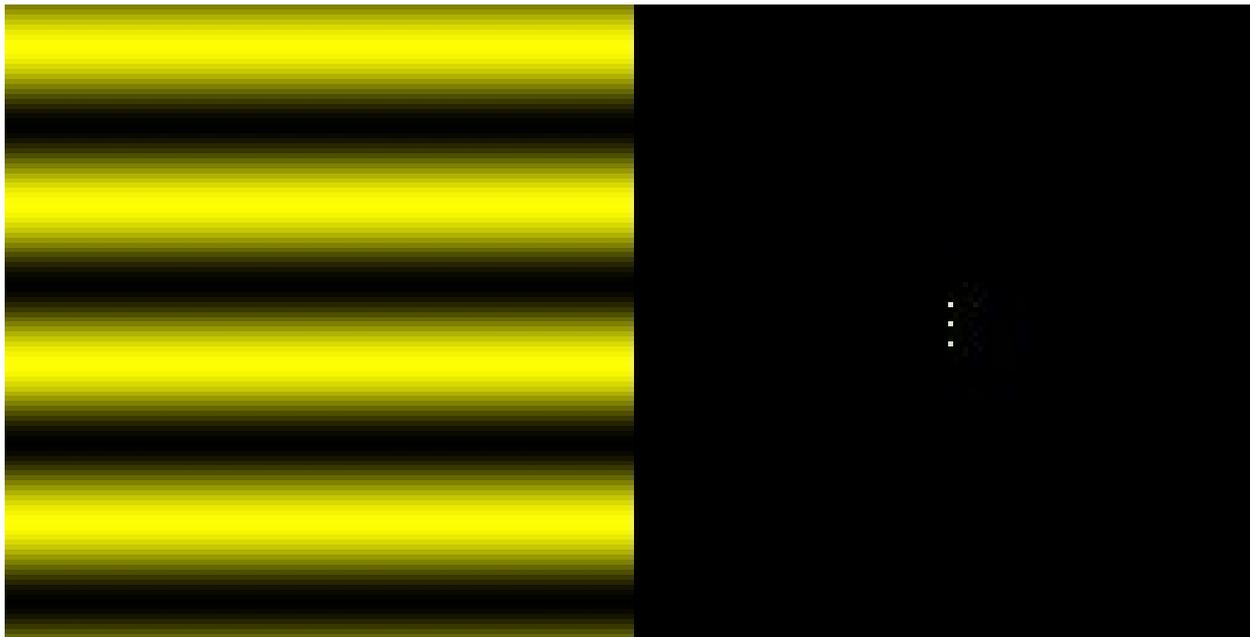
imagem superior: Linhas com zaggy edges, e componentes em frequência. Abaixo, removendo alguns componentes de frequência resulta em linhas mais suaves.



## Transformada de Fourier

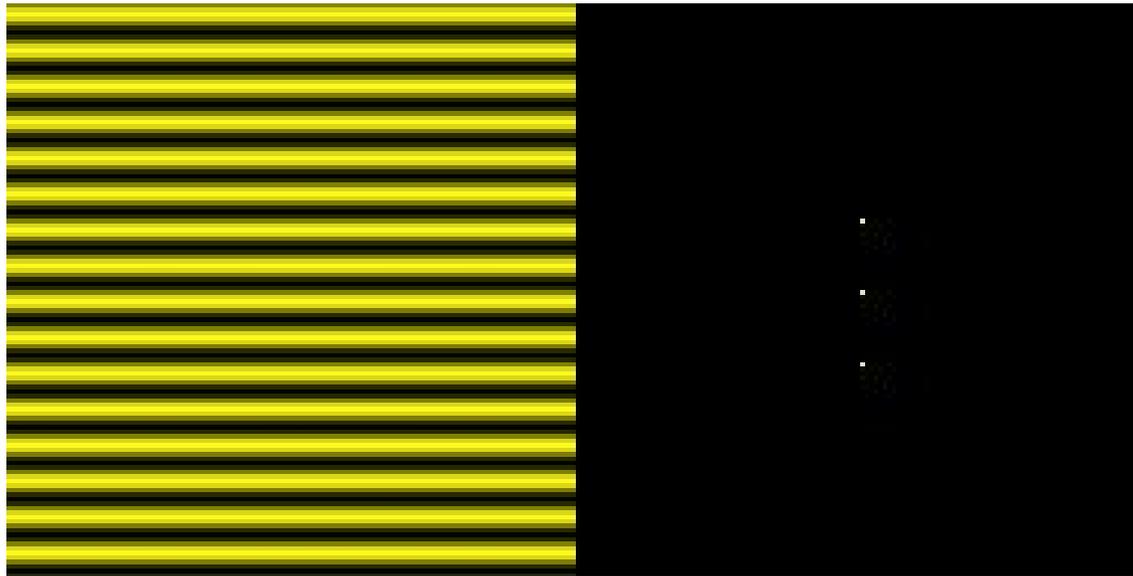
Função senoidal com um componente DC. O ponto central é o componente DC e os dois outros representam a frequência da função senoidal.

Não há pixels na direção  $x$ , porque a imagem é a mesma em qualquer lugar nessa direção .



## Transformada de Fourier

Função senoidal com uma frequência superior usada para gerar a imagem, os dois pontos estão mais separados da origem para representar uma maior frequência. De acordo com a propriedade de escala da transformada Fourier, os contrastes da imagem e seu espectro se tornam mais largos.



# *Discrete Fourier Transform* (DFT)

- O número de frequências corresponde ao número de pixels do domínio espacial, ou seja, a imagem do domínio espacial e Fourier são do mesmo tamanho

$$F(k, l) = \frac{1}{N^2} \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f(a, b) e^{-i2\pi(\frac{ka}{N} + \frac{lb}{N})}$$

$F(0,0)$  representa o componente DC da imagem

- que corresponde à média de brilho e  $F(N-1, N-1)$  representa a maior frequência.
- Ao descrever uma função periódica no domínio da frequência, a componente DC, ou coeficiente DC é o valor médio da onda .

# A transformada de Fourier inversa é dada por

$$f(a, b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{i2\pi(\frac{ka}{N} + \frac{lb}{N})}$$

- a Transformada de Fourier é separável :

$$F(k, l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k, b) e^{-i2\pi \frac{lb}{N}} \quad P(k, b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a, b) e^{-i2\pi \frac{ka}{N}}$$

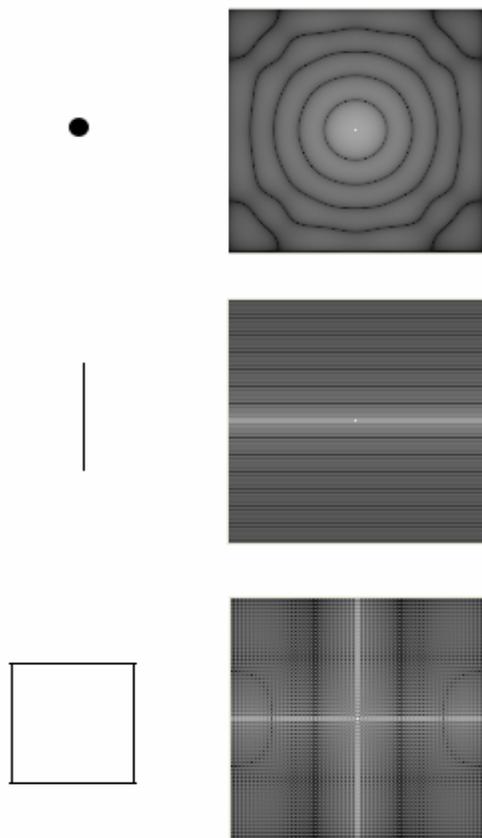
# Complexidade computacional e tamanho

- Expressando a TF 2D em termos de 2 séries unidimensionais diminui o número de cálculos necessários.
- a DFT unidimensional tem complexidade  $N^2$ . Isso pode ser reduzido

$$N \log_2 N$$

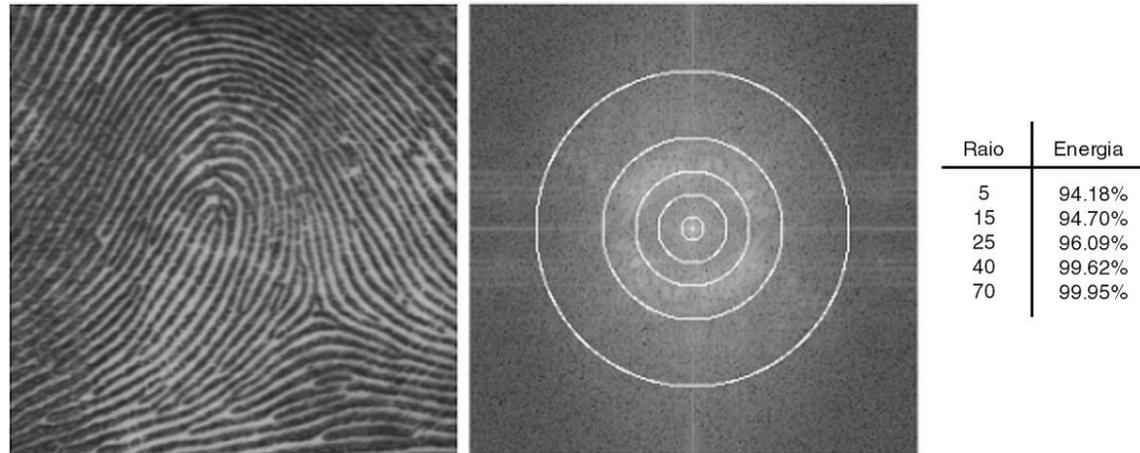
- se empregar a Transformada Rápida de Fourier (FFT) para calcular as DFTs unidimensional.
- Esta é uma melhoria significativa, em especial para imagens grandes. Existem várias formas da FFT e a maioria restringe o tamanho da imagem de entrada que pode ser transformado, para
- onde  $n$  é um inteiro.  $N = 2^n$
- Os detalhes matemáticos são bem descritos na literatura

# Transformada de Fourier 2D



Algumas imagens representadas como  
funções bidimensionais e seus  
espectros de Fourier.

A maior parte da informação de uma imagem normal se concentra em baixas frequencias



Imagens e seu espectro de Fourier, os círculos são falsamente incluídos para se ter uma idéia em que frequência se concentram

# Processamento de imagens no domínio de Fourier

1- A imagem  $I(x,y)$  é transformada para o **domínio de Fourier** usando transformada **discreta**: DFT.

2- A imagem no domínio de Fourier é representada por  $F(u,v)$  e é **convoluída** com o **filtro**  $H(u,v)$ .

3- Depois do produto  $F(u,v) H(u,v)$  é aplicada a **inversa da transformada** de Fourier para retornar ao **domínio espacial**, onde se tem a imagem processada  $I'(x,y)$ .

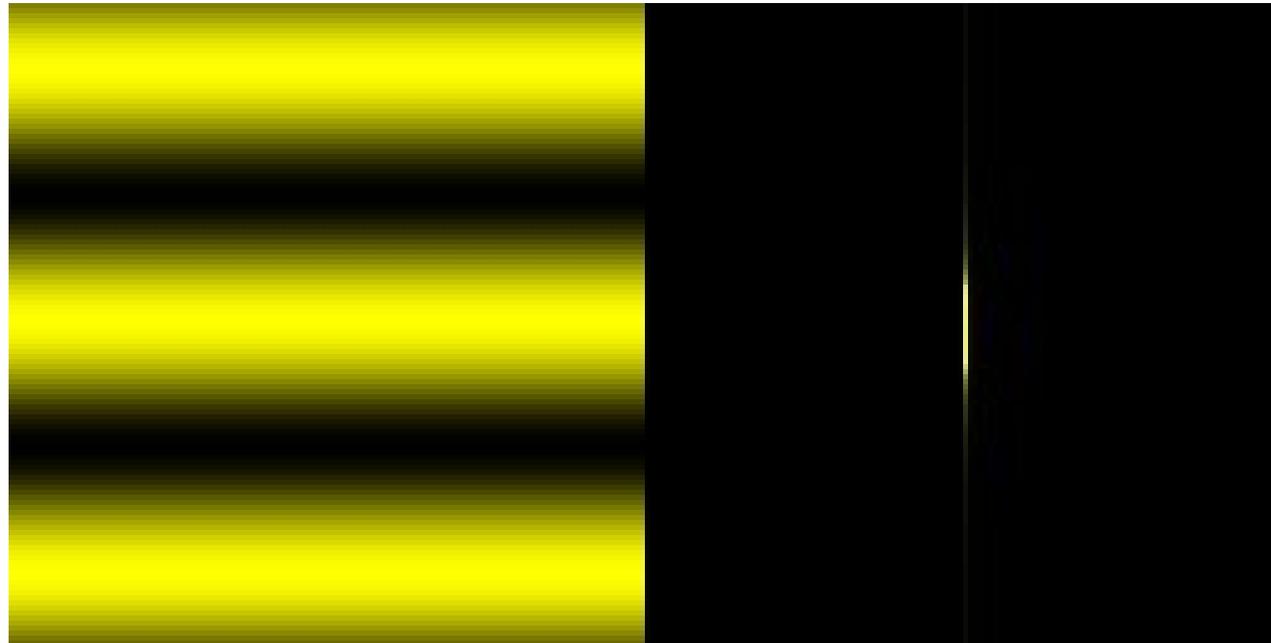
# Transformada de Fourier

- A teoria de Fourier diz que qualquer sinal ou imagens, pode ser expresso como uma soma de uma série de sinóides. No caso de imagens, essas são variações senoidais de intensidade. Por exemplo, o padrão **senoidais** mostrado nas figuras abaixo pode ser capturado em apenas um termo Fourier que codifica:
  1. a frequência espacial,
  2. a magnitude (positiva ou negativa),
  - 3: a fase.
- Esses três valores capturam toda a informação na imagem **senoidais**. A frequência espacial é a frequência através do espaço (o eixo x nesse caso) com a qual modula o brilho.
- A imagem abaixo à direita mostra outra **senoide** com uma frequência espacial mais alta.



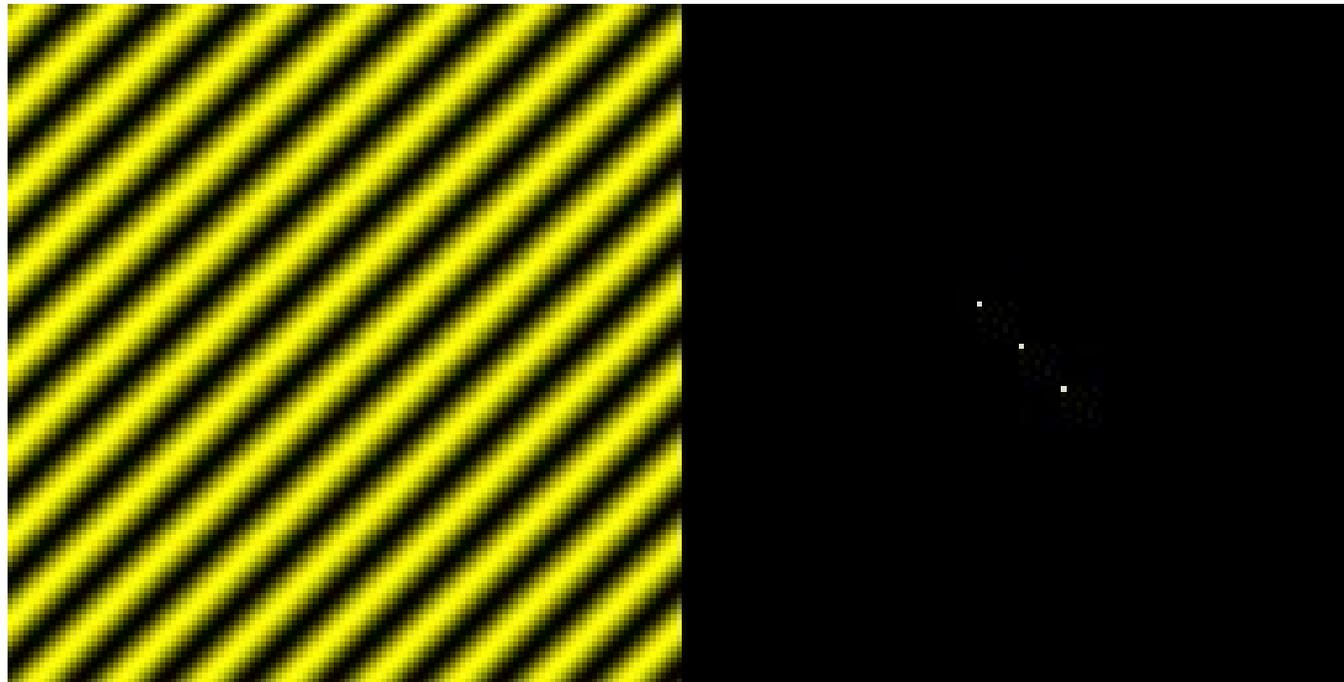
## Transformada de Fourier

se colocar duas imagens senoidais uma em cima da outra, você não haverá mais um seno na direção  $y$ . A imagem também não é mais considerada um seno verdadeiro pelo espectro e ela não contém apenas dois pontos, mas uma linha.



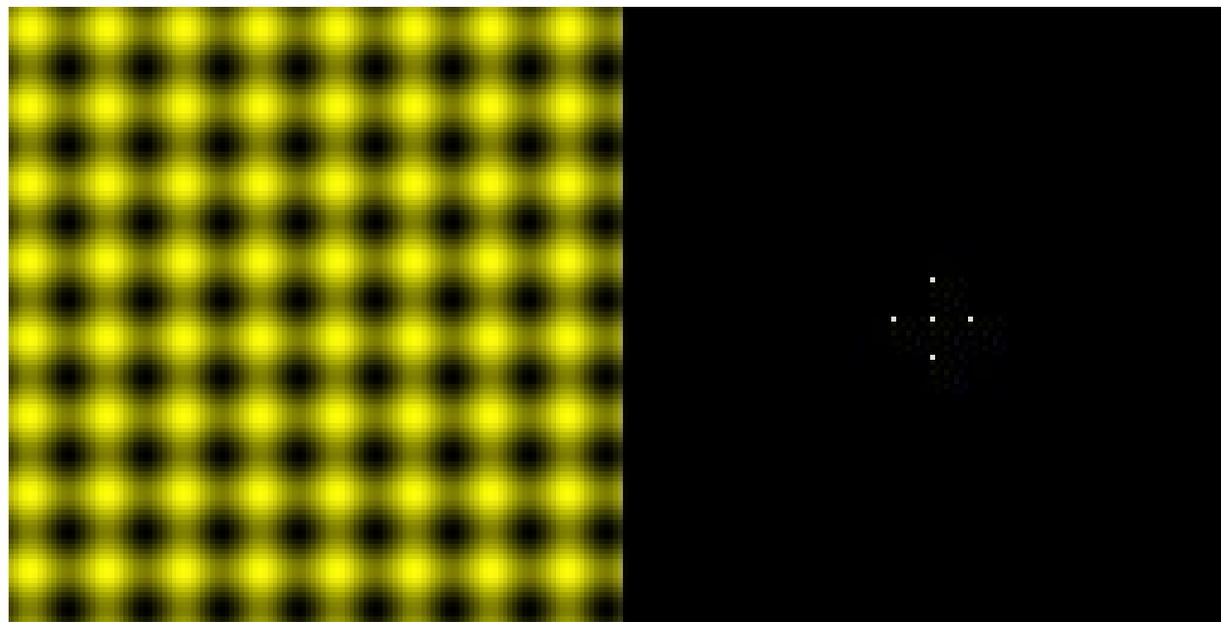
## Transformada de Fourier

Uma das propriedades da transformada Fourier 2D é que se rotacionar a imagem, o espectro irá rotacionar na mesma direção.

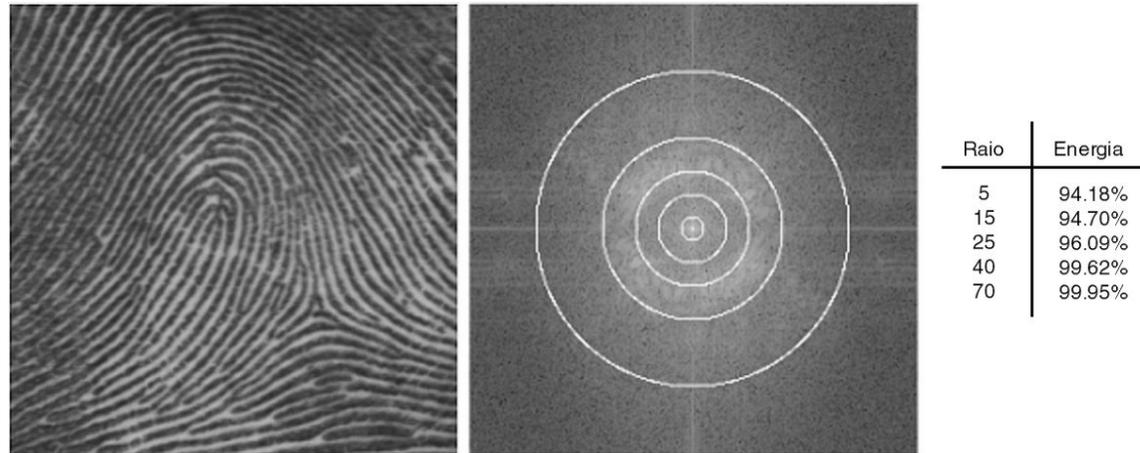


## Transformada de Fourier

A imagem seguinte é a soma de duas funções seno, em direções contrárias (em INTENSIDADES DIFERENTES DE PRETO E AMARELO).



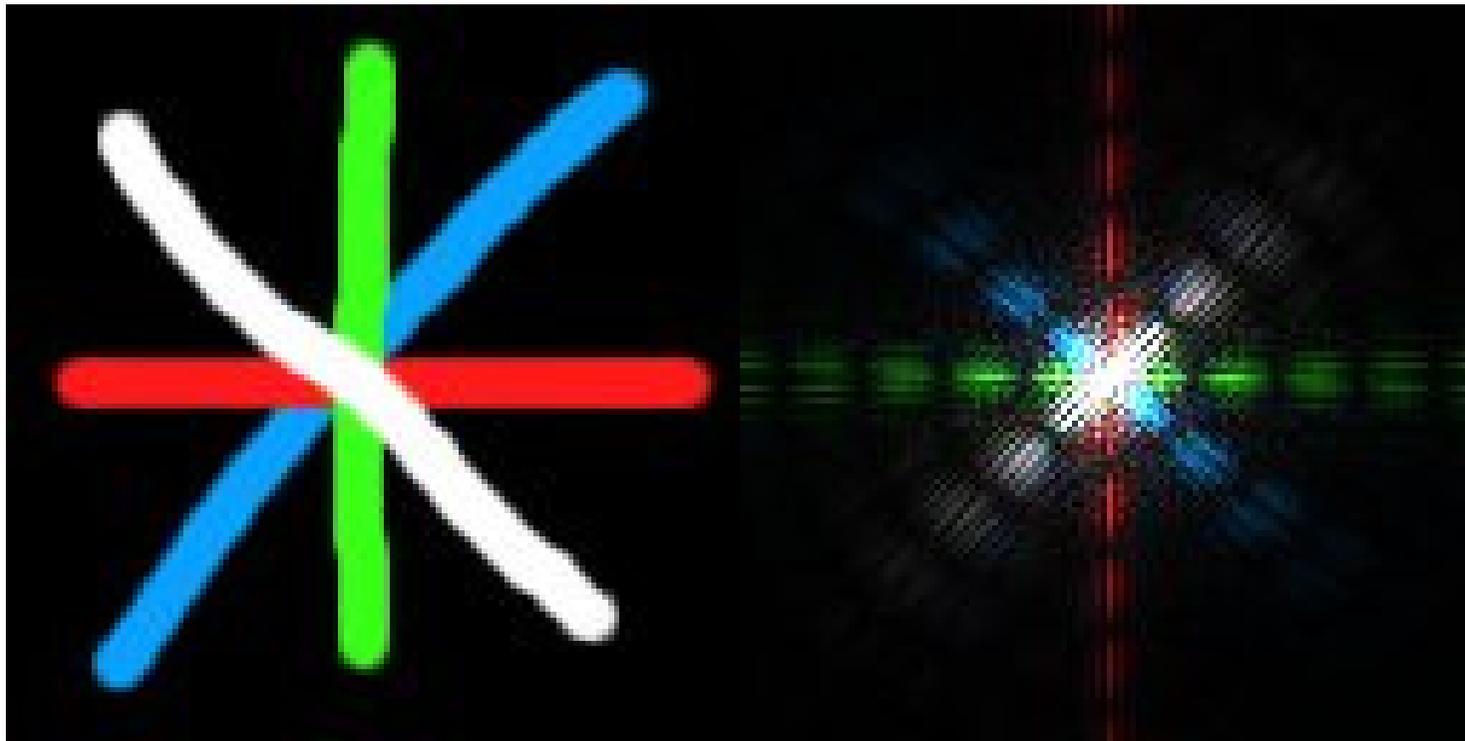
A maior parte da informação de uma imagem normal se concentra em baixas frequencias



Imagens e seu espectro de Fourier, os círculos são falsamente incluídos para se ter uma idéia em que frequência se concentram

## Transformada de Fourier

Como linhas em uma imagem geralmente geram linhas perpendiculares no espectro.



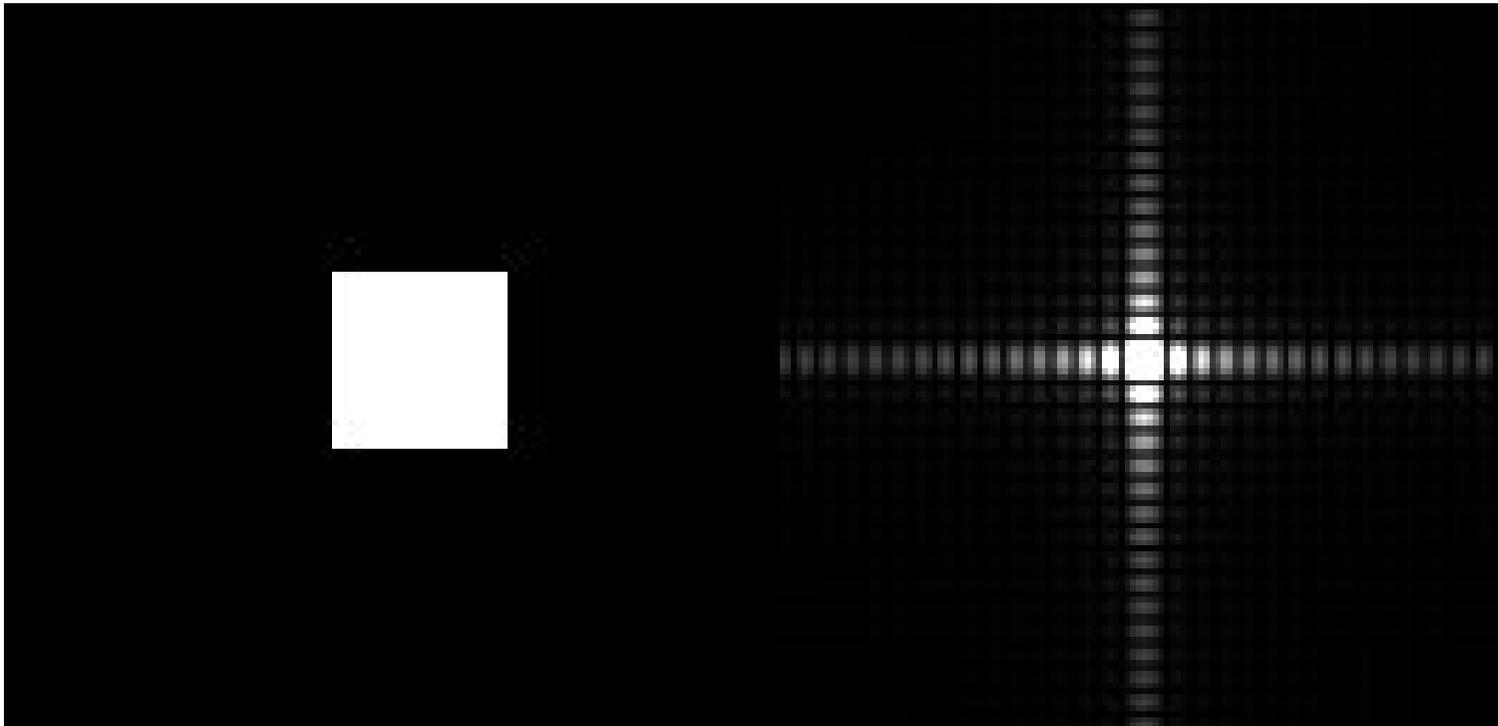
## Transformada de Fourier

As linhas inclinadas no espectro são devido a transição aguda do céu para a montanha.



## Transformada de Fourier

A FFT de uma função retangular é uma função seno 2D:

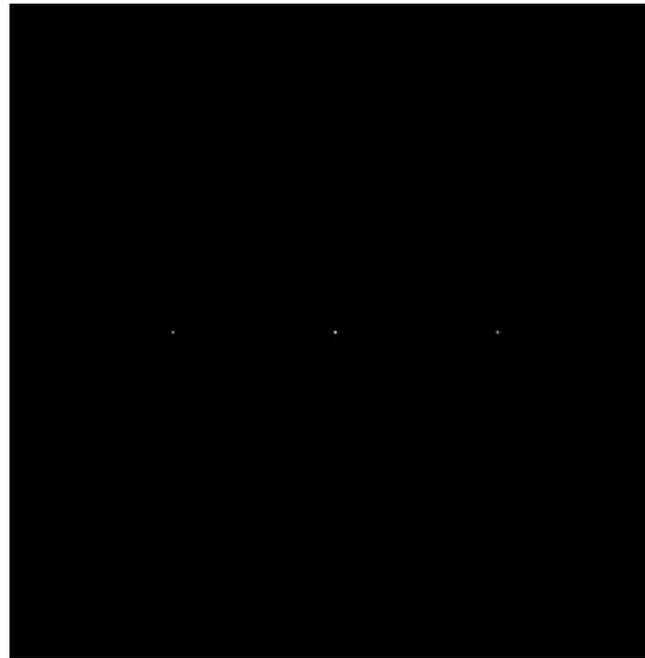
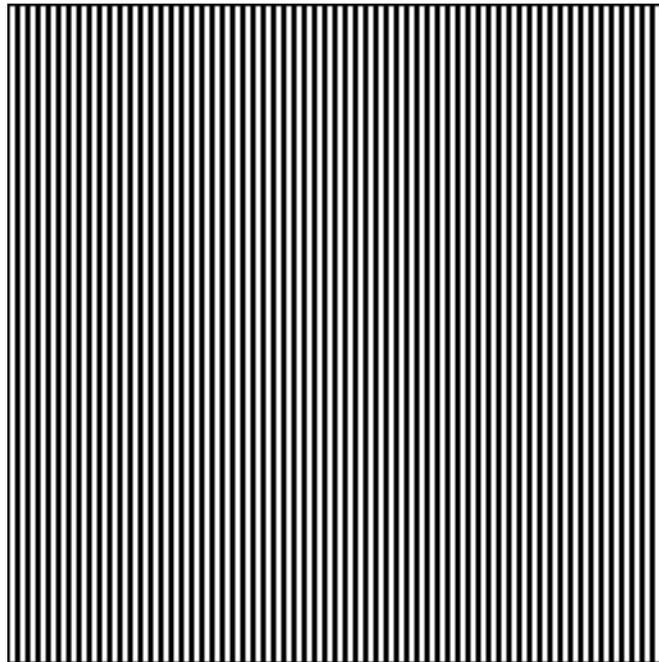


## Transformada de Fourier

TF de um textura sem mudanças abruptas na lateral horizontal e vertical, logo não há linhas horizontais e verticais no espectro.



# Transformada de Fourier 2D



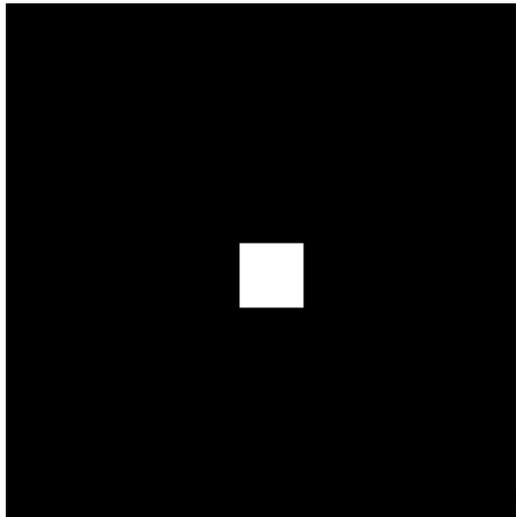
Exemplo de  
imagem  
representada  
como  
funções  
bidimensionais  
e seus  
espectros de  
Fourier.

Imagem com listras verticais com 2 pixels de largura. A magnitude da transformada de Fourier desta imagem é mostrada ao lado e tem a DC ou  $F(0,0)$  e outros dois pontos correspondentes à frequência das listras na imagem original

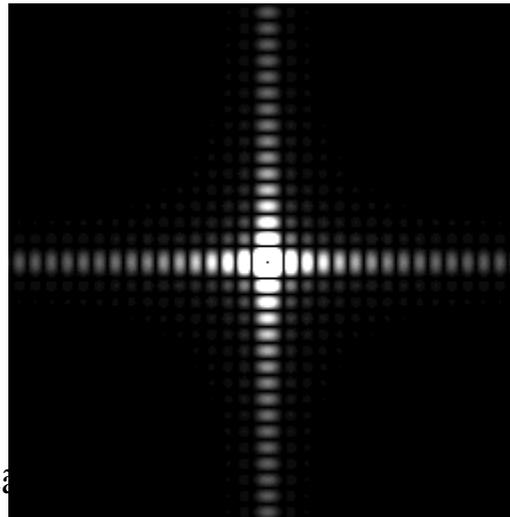
# Transformada de Fourier 2D

Exemplo de imagem representada como funções bidimensionais e seus espectros de Fourier.

Pulso quadrado



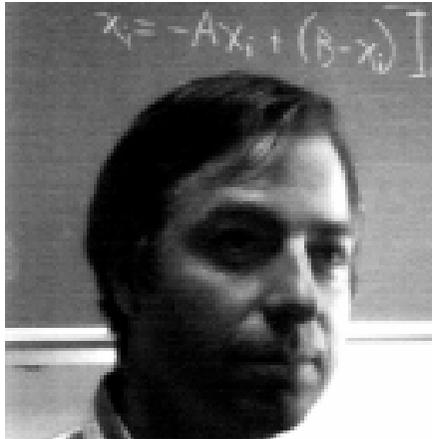
e sua transformada



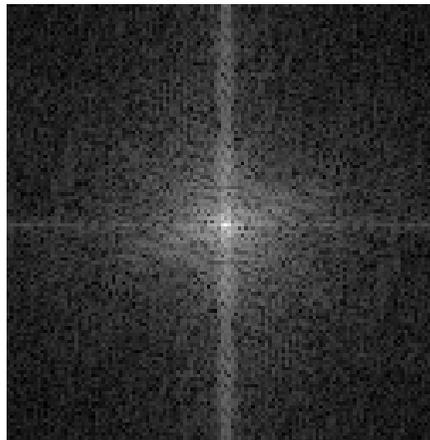
# Transformada de Fourier e Filtragem

- A transformada Fourier pode ser usada para filtragem ajustando o conteúdo em frequência de uma imagem.
- Na imagem abaixo, faz-se uma transformada de Fourier, e logo após uma transformada inversa para reconstruir a imagem original.
- Essa imagem reconstruída é idêntica, pixel-por-pixel, da imagem original.

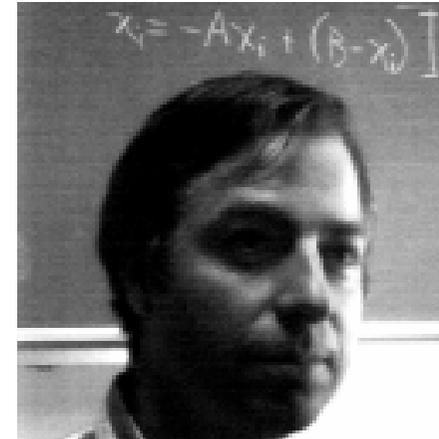
**Imagem e intensidade**



**Transformada Fourier**



**Transformada Inversa**



# Filtragem no domínio da Freqüência

---

- Transformada de Fourier

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

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

- Transformada Discreta de Fourier (DFT)

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

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

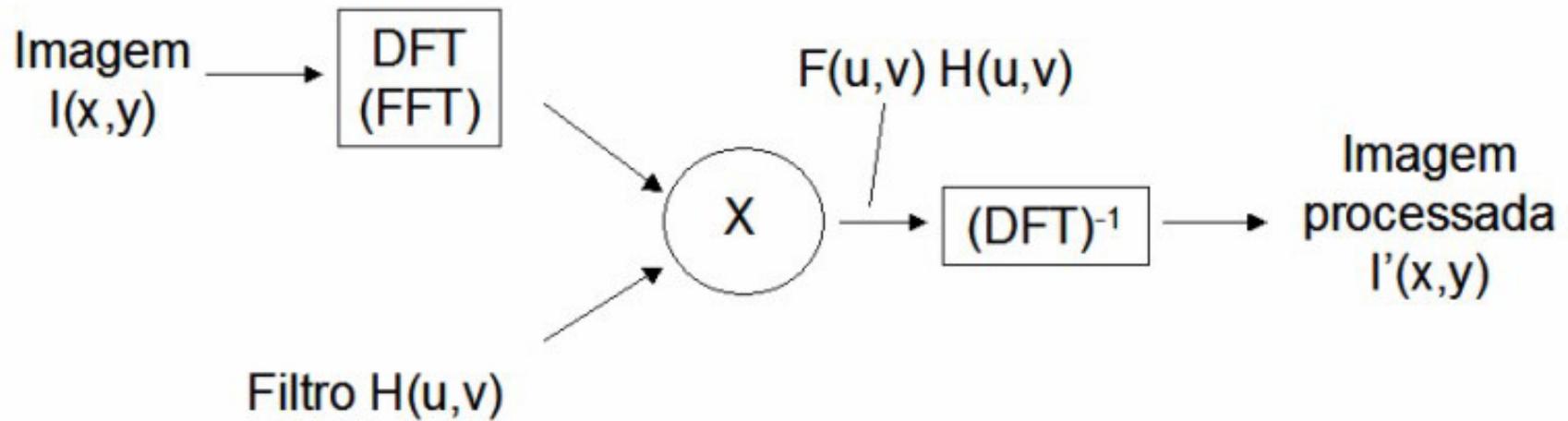
- Teorema da convolução

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

$$f(x, y) \cdot h(x, y) = F(u, v) * H(u, v)$$

$$g(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$$

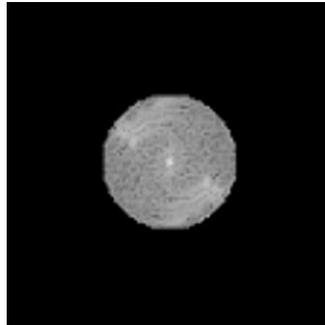
- faz um espelhamento da função  $h(x, y)$  sobre a origem;
- desloca-se esta função  $h(x, y)$  com respeito à  $f(x, y)$  pelo incremento dos valores de  $(m, n)$ ;
  - e calcula-se a soma dos produtos sobre todos os valores de  $m$  e  $n$ , para cada deslocamento. Os deslocamentos  $(m, n)$  são incrementos inteiros que param quando as funções não mais se sobrepõem.



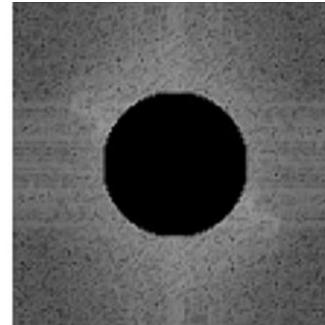
Esquema ilustrando os passos da filtragem no domínio de Fourier

# Tipos de filtro quanto a frequencia:

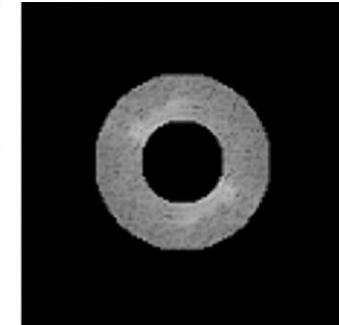
- Passa baixa,
- Passa alta e
- Passa faixa



(a)



(b)



(c)

# Filtragem Passa Baixa

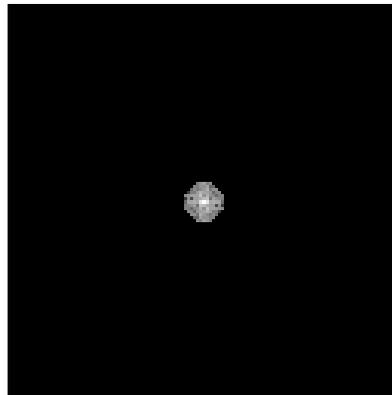
- Utilizando um **filtro passa baixa** obtém-se uma imagem **mais suavizada**.
- Os **detalhes finos** (ex: bordas, lados e outras transições abruptas de nível de cinza) da imagem correspondem a **altas frequências**.
- Pode-se ter uma perda de detalhes que são os componentes de altas frequências.

# Transformada de Fourier

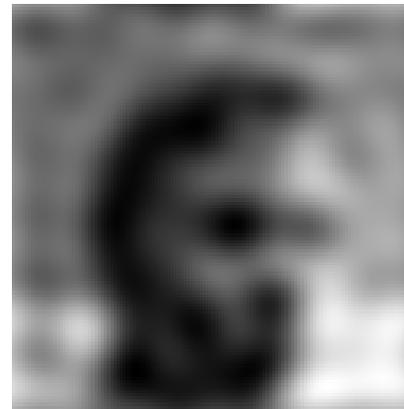
## Operação de Filtragem passa baixa

- A imagem transformada é manipulada para ajustar seu conteúdo de frequência espacial, e então desenvolver uma transformada inversa para produzir a imagem filtrada de Fourier. O filtro passa baixa permite que apenas as componentes de frequência espacial baixa passem, cortando as frequências espaciais altas. Os componentes de baixa frequência são encontrados próximos ao ponto DC central, define-se um raio ao redor do ponto DC, e zera cada ponto na imagem Fourier que está além daquele raio. **A transformada de filtragem passa baixa é idêntica à porção central da transformada Fourier, com o resto da imagem Fourier ajustada para zero.** Uma transformada inversa Fourier aplicada a essa imagem filtrada em passa baixa produz a imagem transformada inversa mostrada abaixo:

**Filtragem passa baixa**



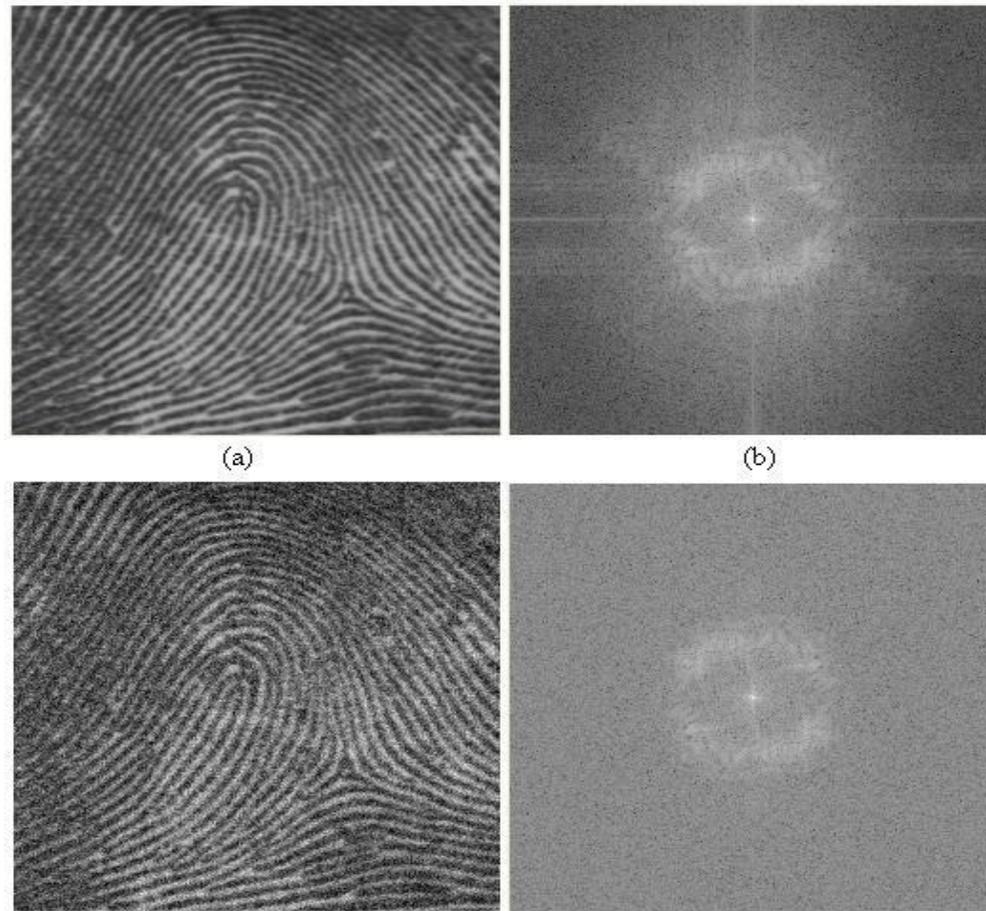
**Transformada Inversa**



- Observa-se que a imagem filtrada em passa baixa é borrada, preservando as regiões suaves amplas de baixa frequência de escuro e brilho, mas perdendo os contornos acentuados e bordas crisp.**

# Filtragem Passa Baixa

- Na filtragem **passa baixa**, os componentes de baixa frequência da transformada de Fourier **não são alterados**, enquanto os de **alta frequência são removidos**.
- Isto faz com que as partes **constantes** da imagem sejam enfatizados.

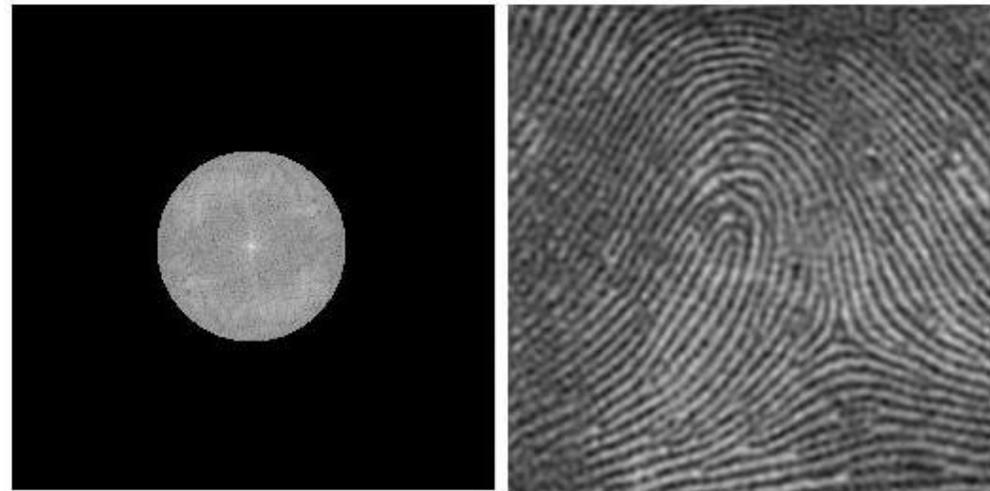


Comparação da imagem e do seu espectro de Fourier **depois** e **antes** de um filtro passa baixa.

**Filtro passa baixa ideal:**

$$H(u, v) = 1 \text{ se } u^2 + v^2 < r^2$$

$$H(u, v) = 0 \text{ se } u^2 + v^2 \geq r^2$$



(a)

(b)

Resultado da filtragem passa baixa

# Filtragem Passa Alta

Na filtragem **passa alta**, os componentes de alta frequência da transformada de Fourier **não são alterados**, enquanto os de baixa frequência são removidos.

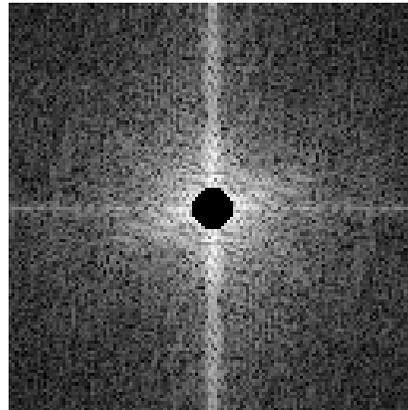
Isto faz com que os detalhes finos da imagem sejam enfatizados.

# Transformada de Fourier

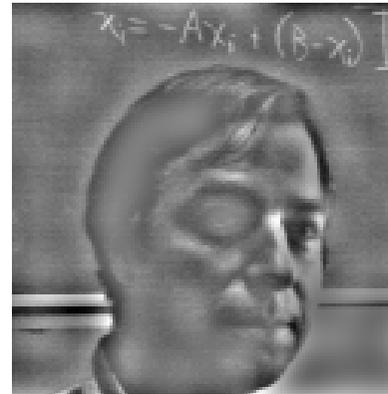
## Operação de Filtragem Passa Alta

- Usa-se o mesmo limiar de frequência espacial para definir um raio na imagem Fourier. Todos os componentes de frequência espacial que caem dentro desse raio são eliminados, preservando apenas os componentes de frequência espacial superiores. Após desempenhar a transformada inversa nessa imagem observa-se o efeito da filtragem passa alta, a qual é para preservar todas as bordas agudas crisp da original, mas ela perde as regiões mais amplas de escuro e brilho.

Filtragem passa alta



Transformada Inversa

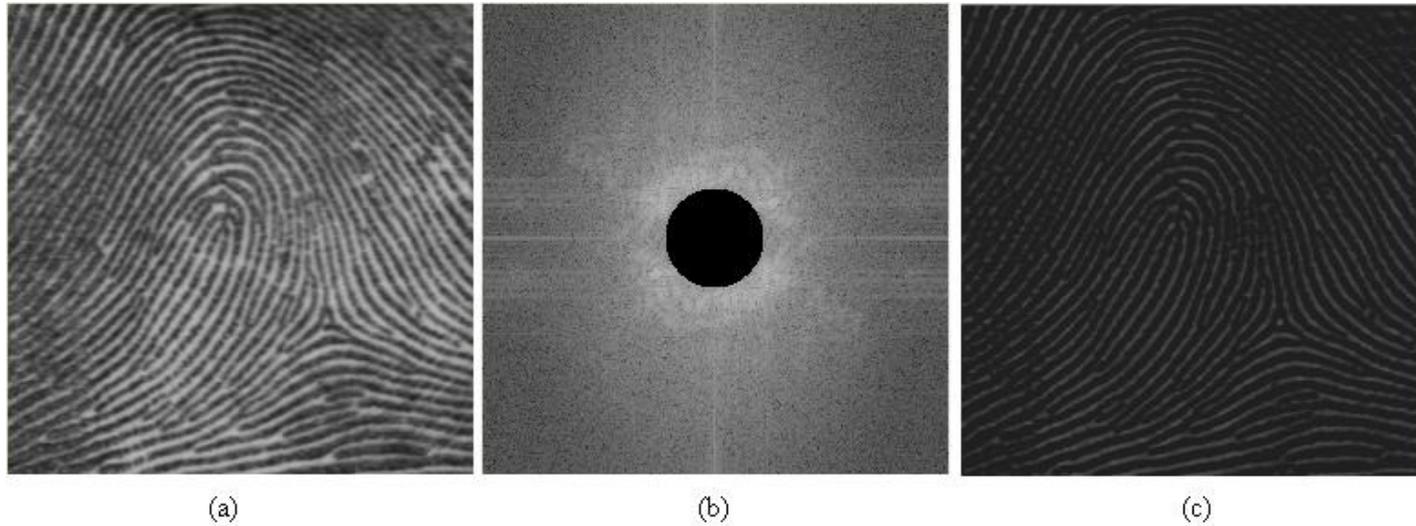


- Se a imagem da transformada inversa filtrada passa baixa é adicionada pixel-por-pixel à imagem da transformada inversa de passa alta, isso iria restaurar exatamente a imagem original sem filtragem. Essas imagens são complementares, portanto, cada uma codifica a informação que está faltando da outra.

## Filtro passa alta ideal:

$$H(u,v) = 0 \text{ se } u^2 + v^2 < r^2$$

$$H(u,v) = 1 \text{ se } u^2 + v^2 \geq r^2$$



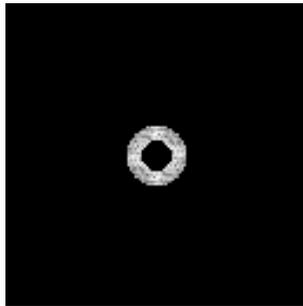
Resultado da filtragem passa alta.

# Transformada de Fourier

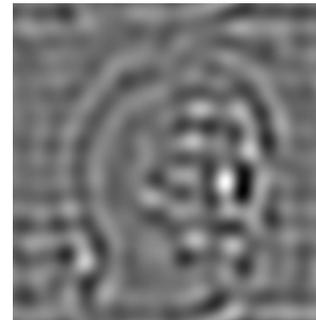
## Operação de Filtragem Passa Faixa

- Preserva apenas aquelas frequências espaciais que caem dentro de uma banda, maior que um cut-off baixo, mas inferior a um cut-off superior.

**Filtragem passa faixa**

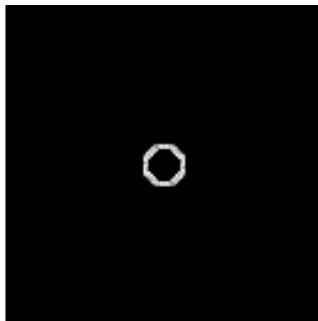


**Transformada Inversa**

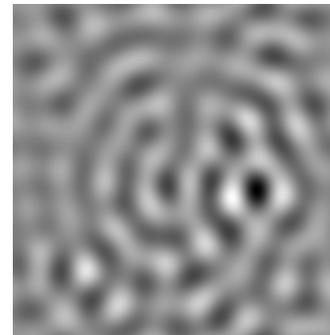


- A figura abaixo mostra uma faixa mais estreita de frequências espaciais.

**Filtragem passa faixa**



**Transformada Inversa**

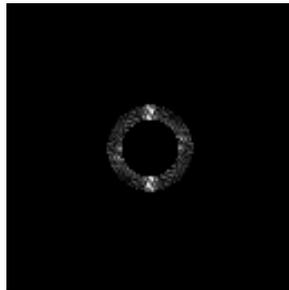


# Transformada de Fourier

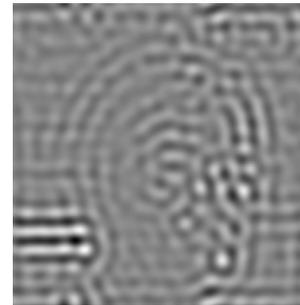
## Operação de Filtragem

- A próxima simulação mostra uma filtragem passa faixa sobre uma banda de frequência espacial superior

Filtragem passa faixa

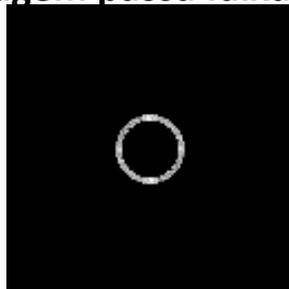


Transformada Inversa

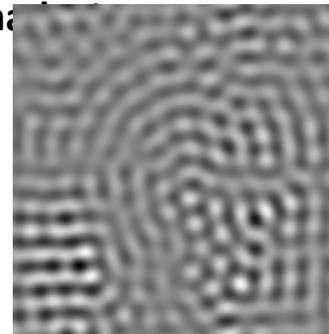


- E abaixo a mesma imagem da superior usando uma banda de frequência espacial mais estreita.

Filtragem passa faixa



Transformada Inversa



# Outros filtros no domínio de frequência

## Filtros pontuais

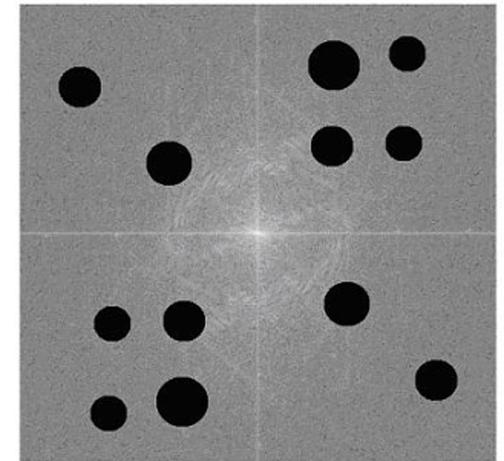
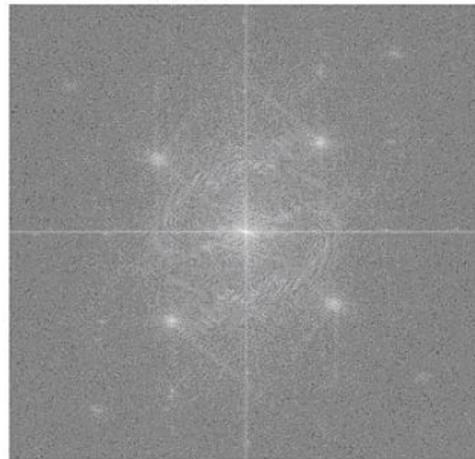


Imagem e seu espectro de Fourier.

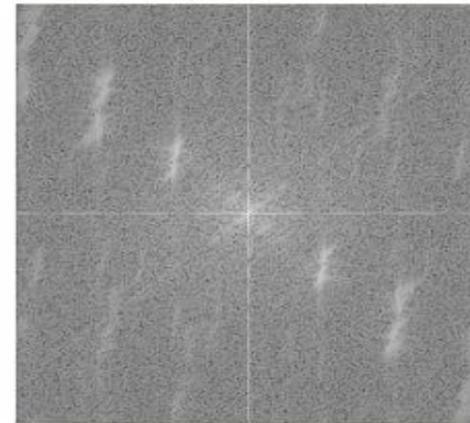
Resultado da filtragem utilizando filtro circular não centrado na origem.

## Outros filtros no domínio de frequência

### Filtros *fan* ou setor circular



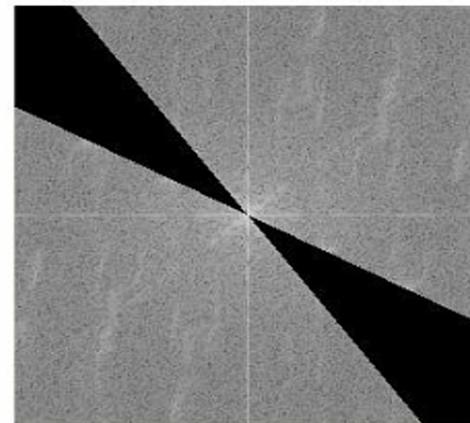
(a)



(b)



(c)



(d)

Resultado da filtragem utilizando filtro setor angular.



**Mas para algumas coisas precisaremos da filtragem em frequência por exemplo: retirar ruído com frequência como no exemplo acima definida**

# Outros filtros no domínio de frequência

## Filtros pontuais

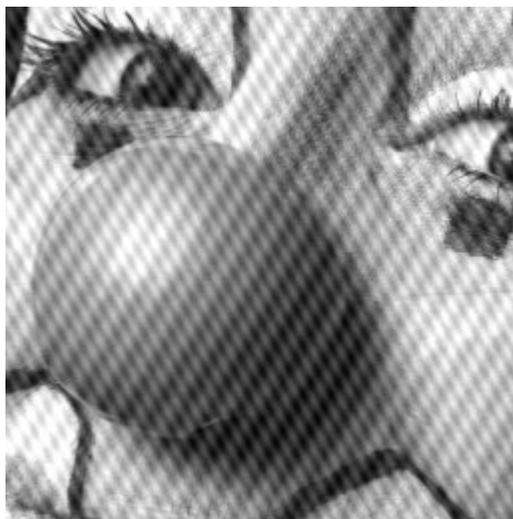
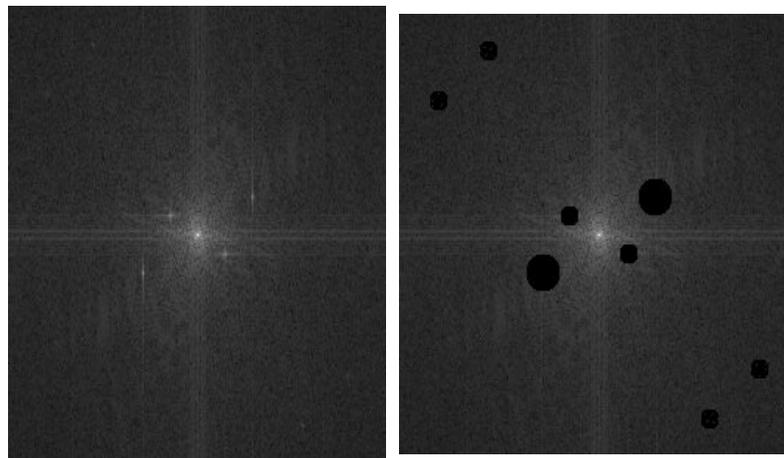


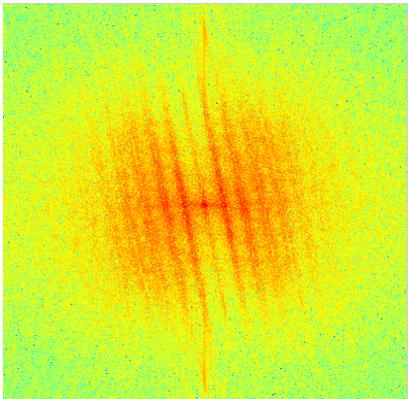
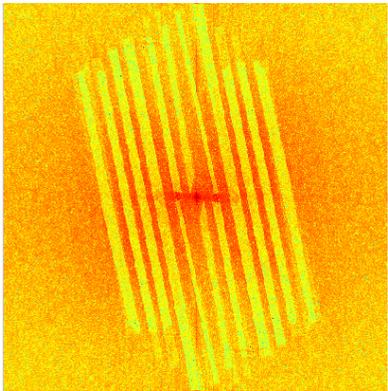
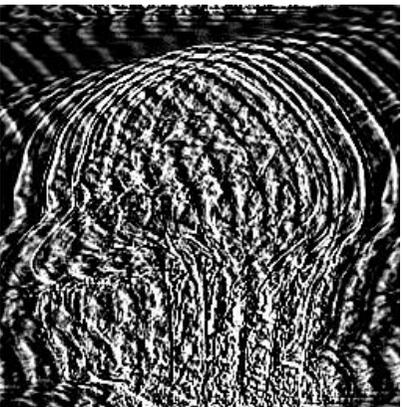
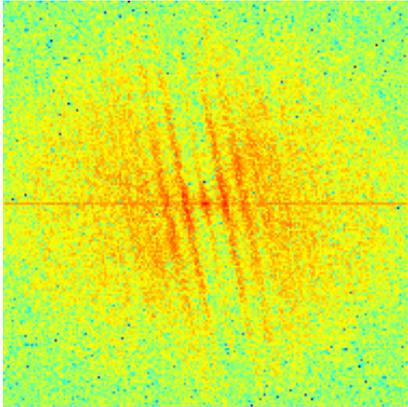
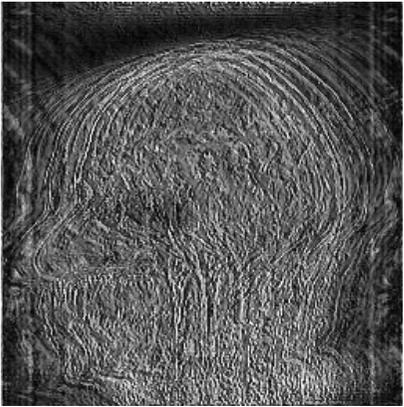
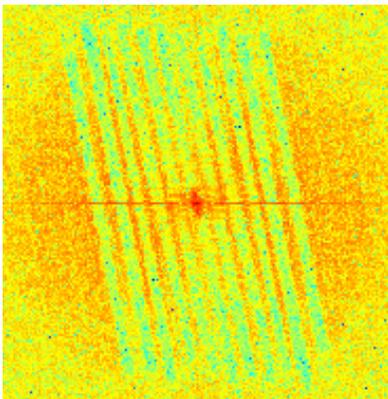
Imagem com moirés em 2 direções obtidas de [2]

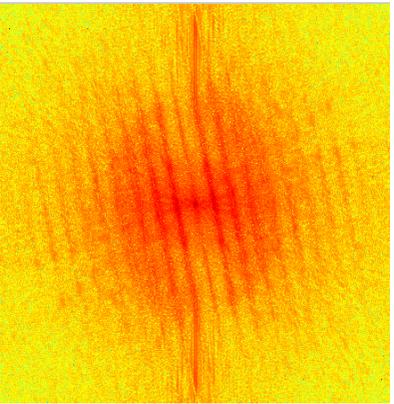
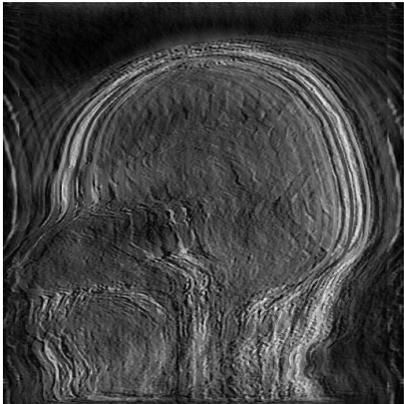
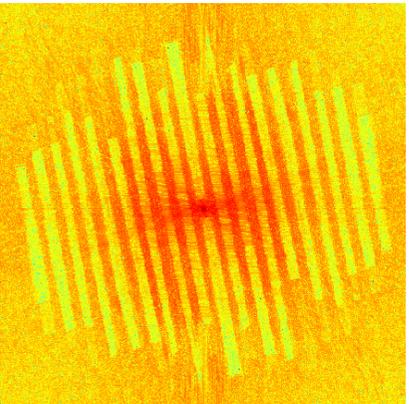
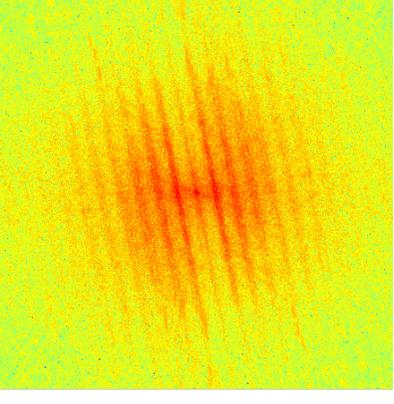
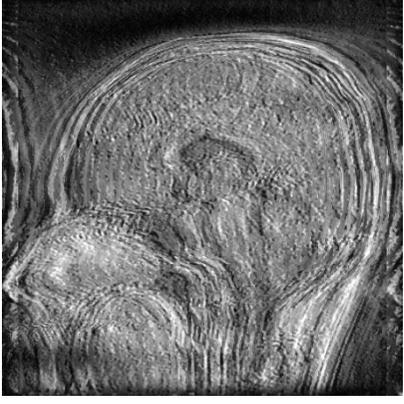
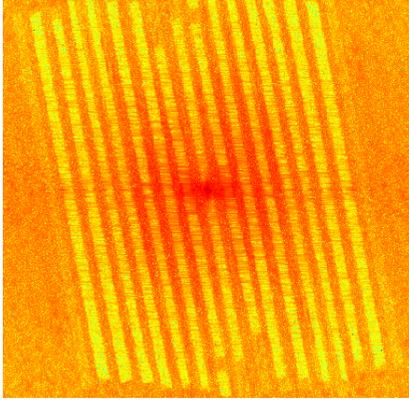


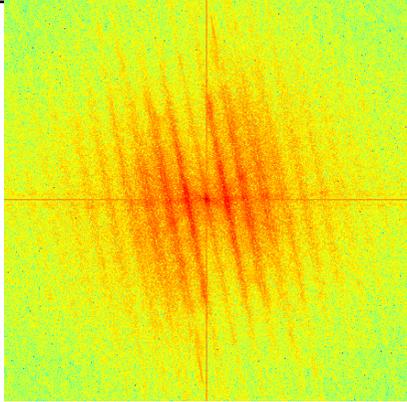
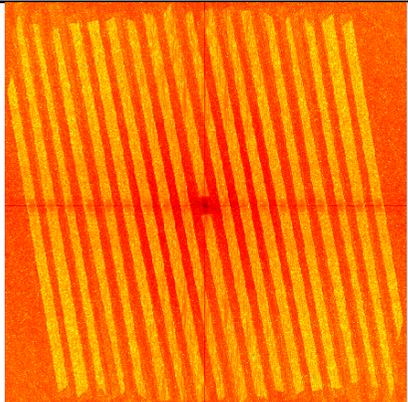
No seu espectro de Fourier aparecem **pontos de maior intensidade**, cuja distância em relação ao centro é proporcional à sua frequência e com inclinação perpendicular a inclinação das mesmas no espaço. **Subtraindo** esses pontos da imagem



Resultado da filtragem utilizando . Repare que melhorou muito

IMG	IMG. ORIGINAL	ESPECTRO DE FOURIER	IMG. FILTRADA	
P1-092				
P1-122				

IMG	IMG. ORIGINAL	ESPECTRO DE FOURIER	IMG. FILTRADA	
P1-152				
P2-082				

IM G	IMG. ORIGINAL	ESPECTRO DE FOURIER	IMG. FILTRADA	
P2- 102				

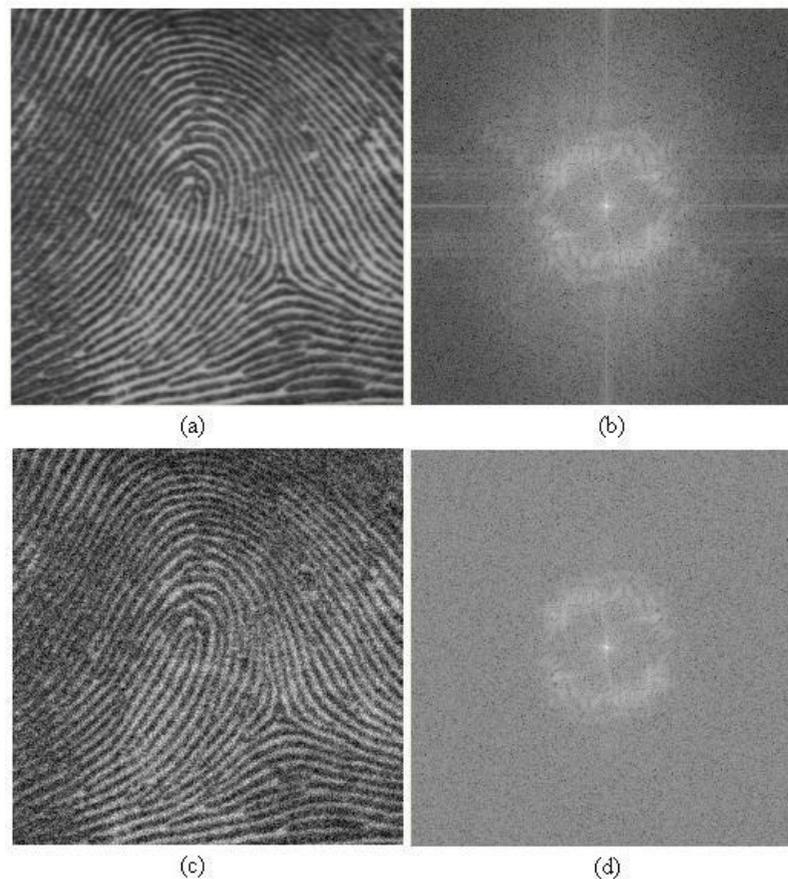
As 5 filtragem anteriores foram feitas por José Ramón González Montero para o curso de AI de 2015

### Ferramentas usadas

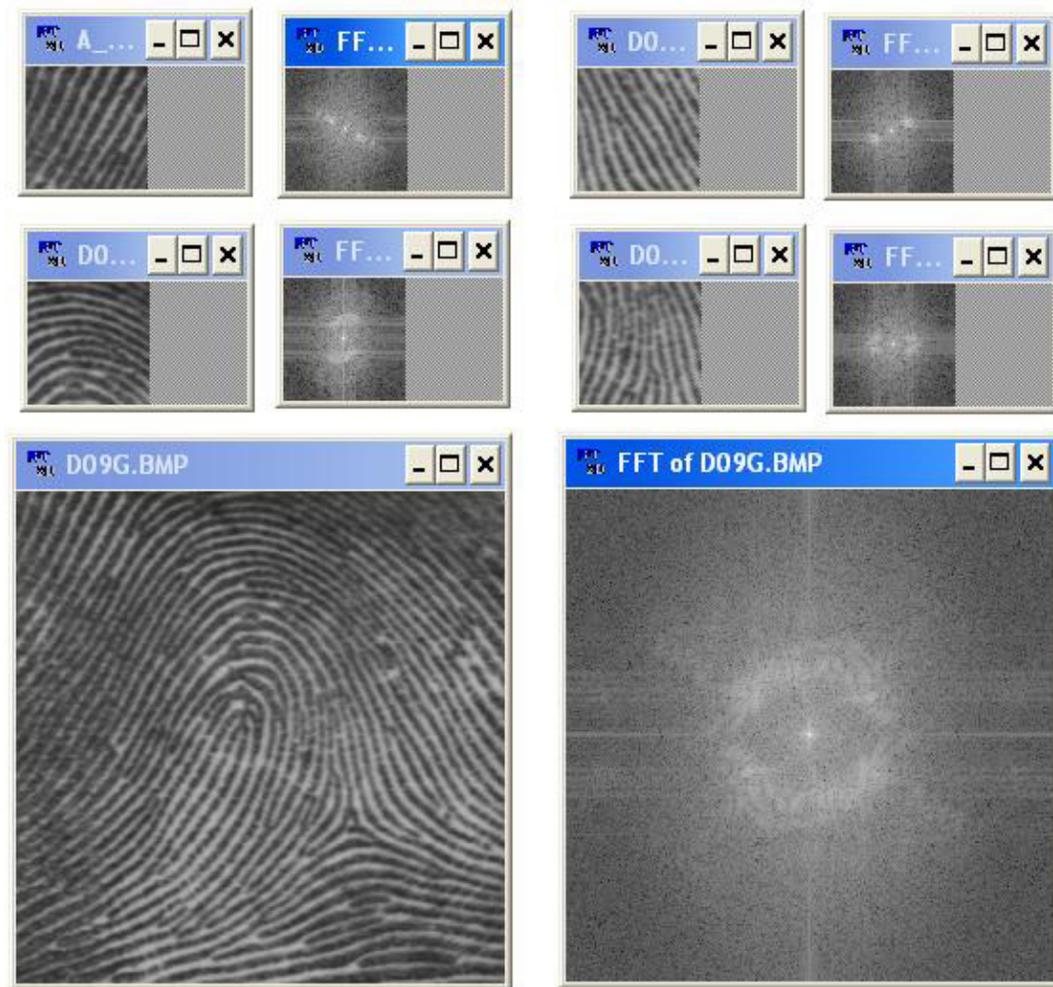
- Adobe Photoshop;
- Plug-in para FFT e IFFT para Photoshop (Disponível em: <http://www.3d4x.ch/Swift's-Reality/FFT-Photoshop-plugin-by-Alex-Chirokov/16,35>)
- Matlab R2009b (Image Processing Toolbox)

## Caracterizando elementos das Imagens pelo seu espectro de Fourier

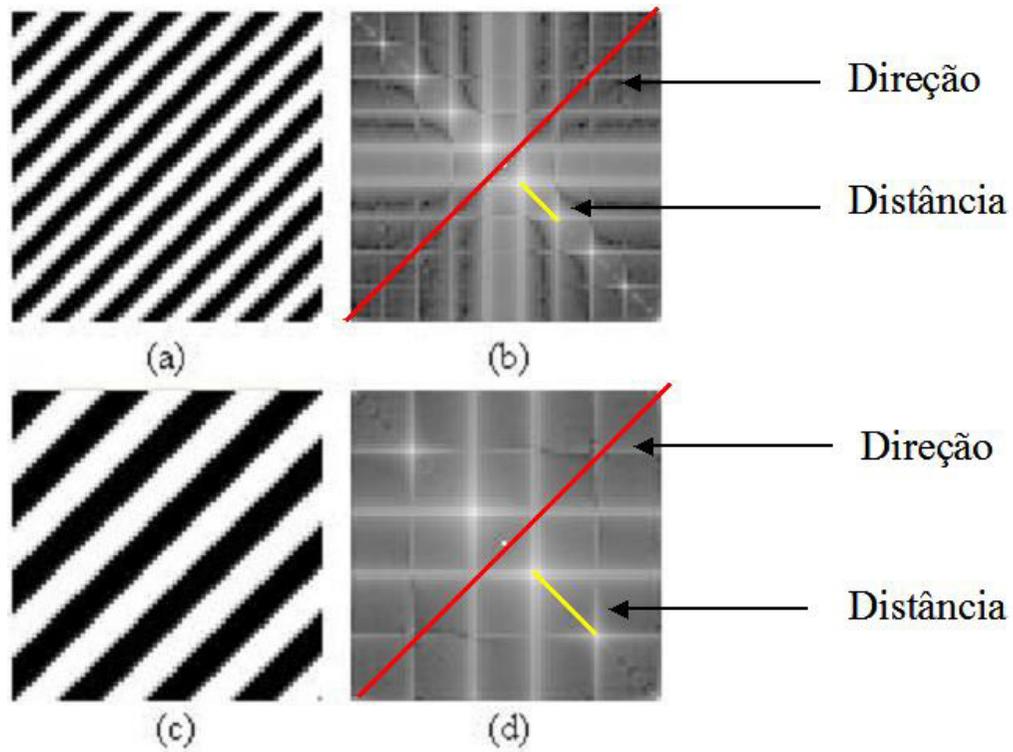
Observa-se no espectro de Fourier de uma impressão digital um **acúmulo de energia** em torno de um **anel**. Isso é devido ao fato das cristas se comportarem como **senóides**, apresentando frequências bem definidas.



Nos espectros de Fourier, de partes desta imagem, aparecem dois picos de intensidade simétricos, em relação à origem.

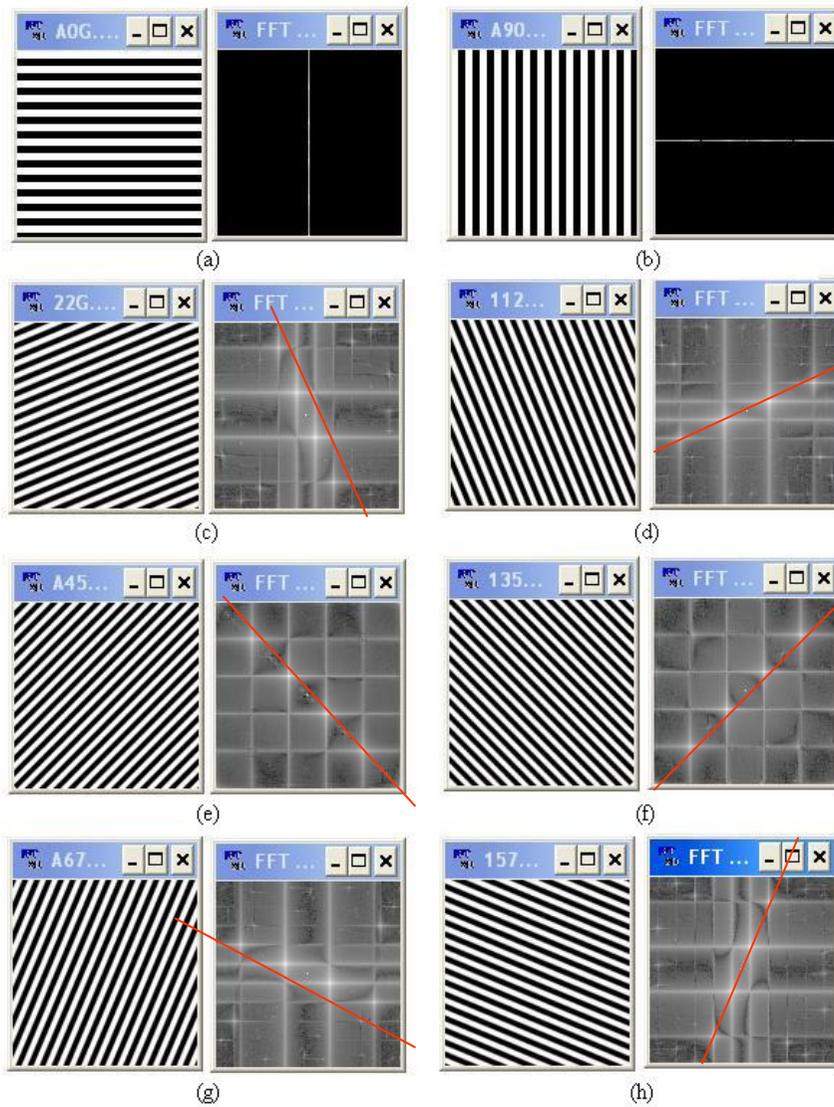


Fragmentos de uma impressão digital e seus espectros de Fourier.



De acordo com a localização desses picos têm-se: a distância e a direção das cristas na região.

Imagens sintéticas representando um fragmento de impressão digital e seus espectros de Fourier.



Imagens sintéticas com listras inclinadas e seus espectros de Fourier. Veja a relação entre os ângulos da imagem e sua TF

# Transformada de Fourier

## Conclusão

A teoria de Fourier é baseada na idéia que qualquer função pode ser decomposta de senos e cossenos de diferentes frequências.

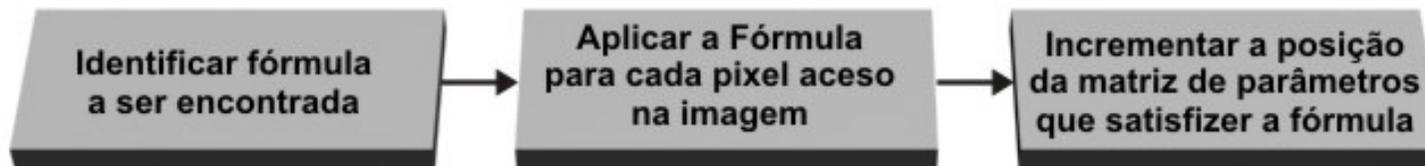
Em computação visual, imagens no domínio espacial podem ser transformadas no domínio da frequência em processamento de imagem, porque algumas operações e medidas são melhores se feitas no espaço da frequência ao invés do domínio espacial.

A implementação da transformada de Fourier é chamada transformada discreta de Fourier (DFT), e outros algoritmos como Transformada rápida de Fourier (FFT) também são desenvolvidas para reduzir a complexidade do DFT. Uma das técnicas que envolvem transformada de Fourier são descritores de Fourier. Eles descrevem as bordas de uma forma, e armazena várias propriedades que são úteis em várias aplicações.

Essas simulações demonstram a informação da imagem em uma maneira distribuída holística que permite manipulação da representação Fourier que codifica o conteúdo da informação global da imagem por manipulações espaciais da imagem transformada.

# Transformada de Hough

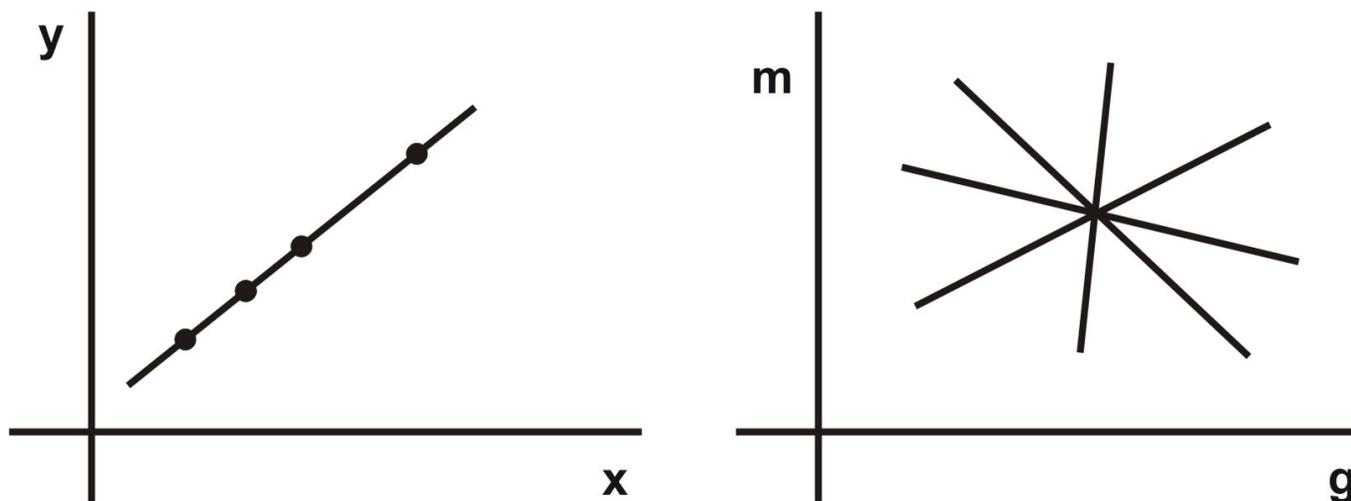
**Transformar** a imagem do espaço digital  $(x,y)$  para uma representação na forma dos parâmetros descritos pela curva que se deseja encontrar na imagem



Etapas da aplicação da transformada de Hough para qualquer forma geométrica.

Retas:

$$y = mx + g$$



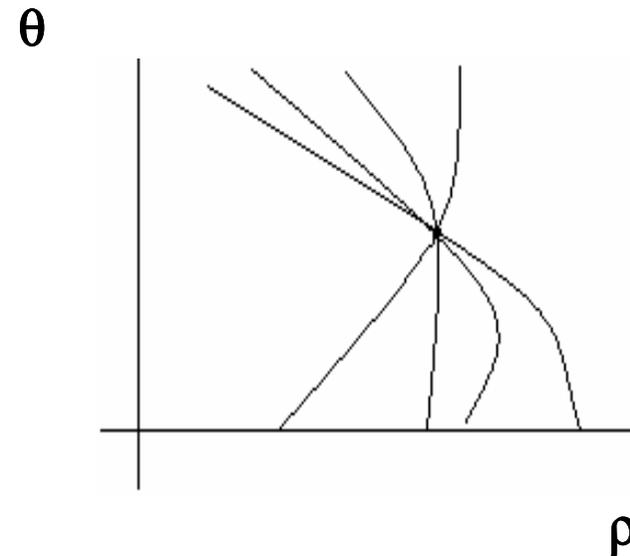
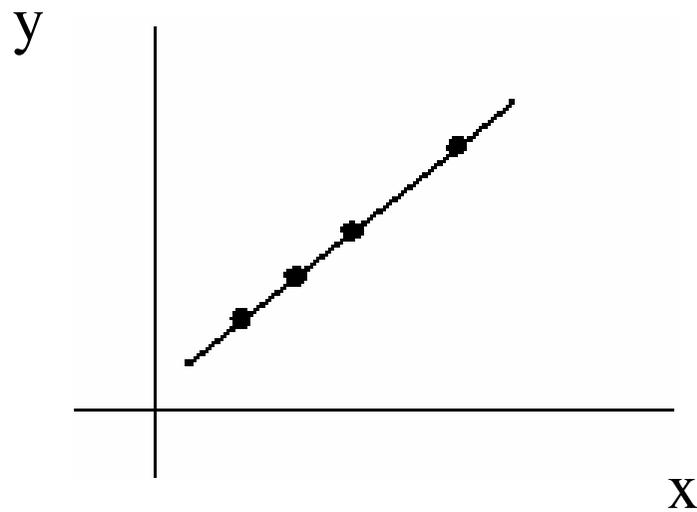
espaço (x,y)

espaço de parâmetros(m,g)

Cada ponto no espaço da imagem transforma-se em uma reta no espaço de parâmetro:  $g = -mx + y$ .

Retas – forma polar:

$$\rho = x \cos \theta + y \sin \theta$$



Cada ponto  $P(x,y)$  no espaço da imagem, corresponde a uma senóide  $S(\rho,\theta)$  no espaço de parâmetros.

## Discretização da matriz acumuladora

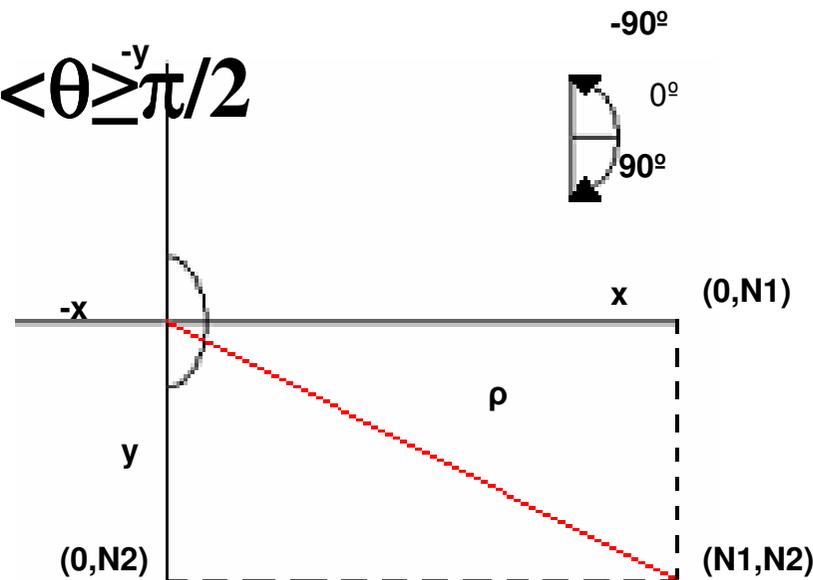
- Para determinar o tamanho e os limites da matriz acumuladora, devem-se observar todos os parâmetros que compõem o espaço dos parâmetros.
- No caso das retas, é necessário ter a informação dos limites angulares de  $\theta$  e dos valores limite de  $\rho$ .
- Para uma imagem descrita entre os limites  $(0,0)$  e  $(N_1, N_2)$ , o parâmetro  $\rho$  se apresenta em um intervalo de  $-\sqrt{N_1^2 + N_2^2}$  a  $\sqrt{N_1^2 + N_2^2}$

# Detecção de retas

## Discretização da matriz acumuladora:

- $\rho \rightarrow -\sqrt{N1^2+N2^2} \text{ à } \sqrt{N1^2+N2^2}$

- $\theta \rightarrow -\pi/2 < \theta \leq \pi/2$



# Transformada de Hough

1- Forma: **reta**

Fórmula:  $y = ax + b$

ou

$b = -ax + y$

2- Aplicação

$x=1$   $y=1$   $a=0,1$   $b=0,9$

$x=1$   $y=1$   $a=0,2$   $b=0,8$

$x=1$   $y=1$   $a=0,3$   $b=0,7$

$x=2$   $y=1$   $a=0,1$   $b=0,8$

$x=2$   $y=1$   $a=0,2$   $b=0,6$

$x=2$   $y=1$   $a=0,3$   $b=0,4$

3- Matriz acumuladora bidimensional

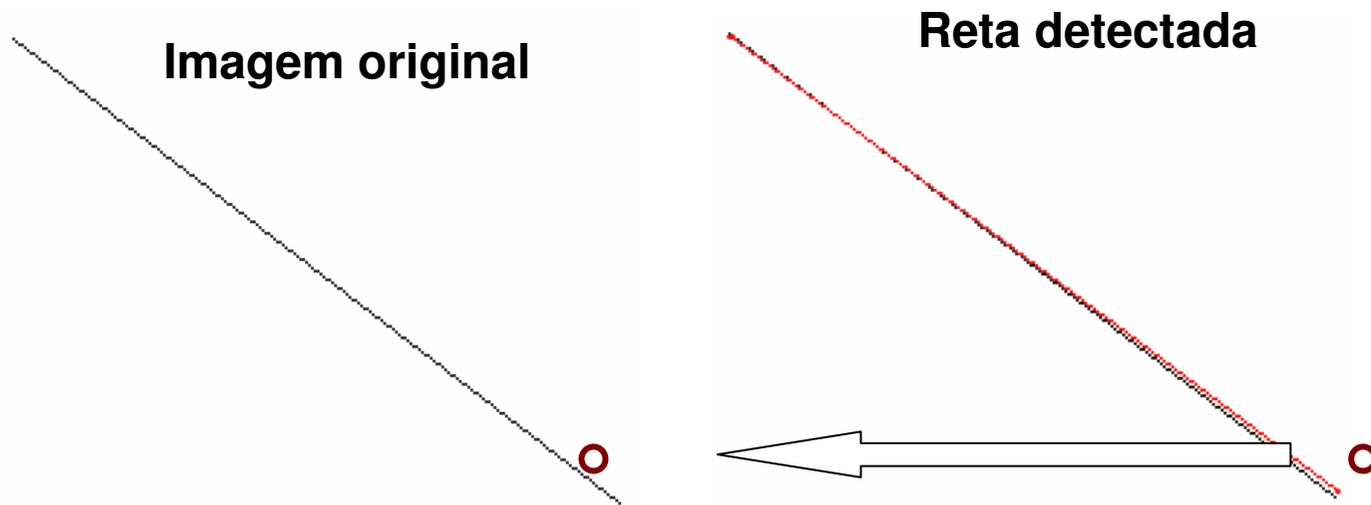
	0,1	a		0,5
0				30
		120		
			60	
		6		
1	4		8	

Reta eleita:  $y = 0,1x + 0,2$

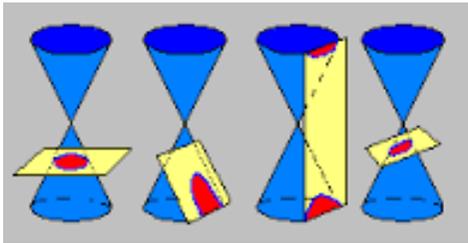
# Detecção de retas

Para identificação dos pontos inicial e final da reta é necessário fazer uma comparação entre imagem original e imagem detectada.

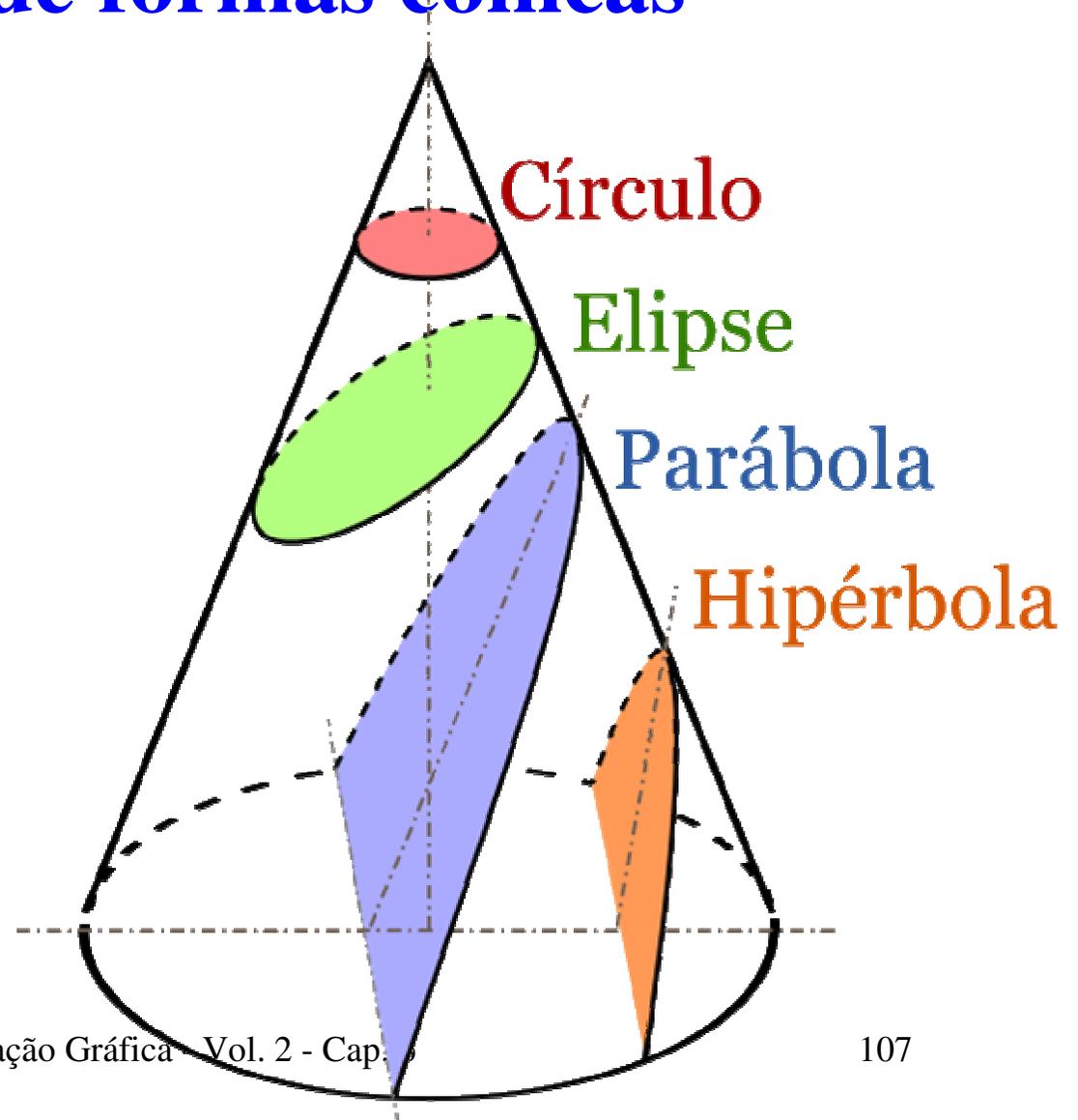
No caso de retas inclinadas, a tolerância a erros ajudará a encontrar os pontos limite reais.



# Detecção de formas cônicas



As cônicas são curvas obtidas pela interseção de um plano com um cone circular.



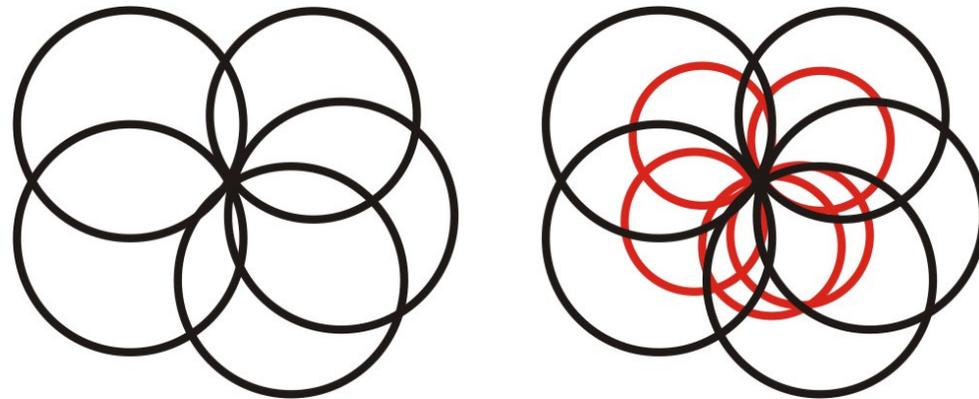
# Transformada de Hough para formas circulares

Formas circulares:

$$x = x_0 + \rho \cos \theta$$

$$y = y_0 + \rho \sin \theta$$

(6.67)



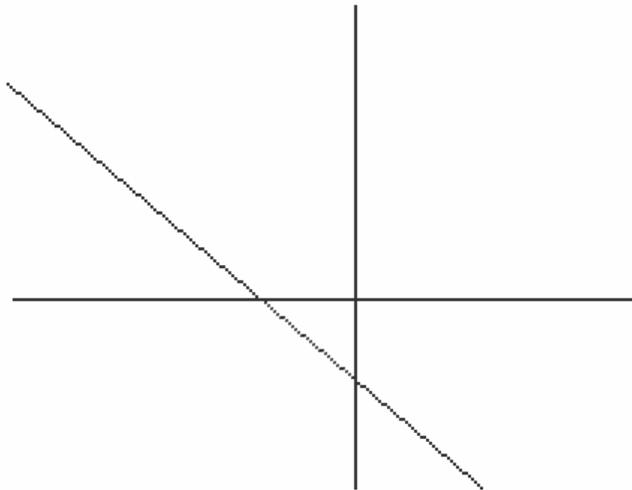
Representação gráfica da transformada de Hough para círculos.

# Detecção de retas

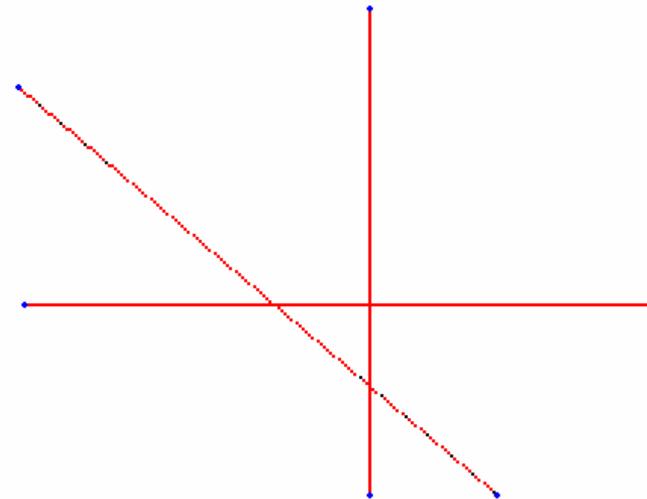
## TESTES

As imagens testadas são do tipo . Raw.

**Imagem original**



**Pontos finais e iniciais identificados**



$\theta$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
-40°	18	(26,0)	(161,158)	0,031s
0°	97	(98,2)	(98,210)	
90°	115	(0,116)	(161,116)	

# Detecção de retas

## TESTES

Imagem original



Pontos finais e iniciais identificados

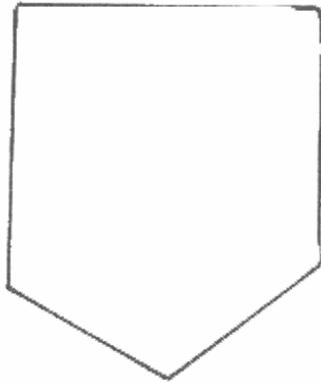


$\theta$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
0	26	(26,40)	(26,154)	0,063s
0	116	(116,40)	(116,154)	
90	40	(26,40)	(116,40)	
90	154	(26,154)	(116,154)	

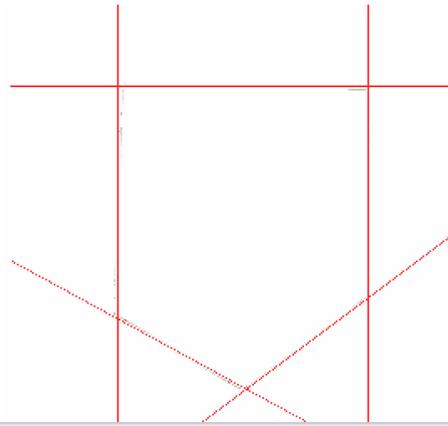
# Detecção de retas

## TESTES

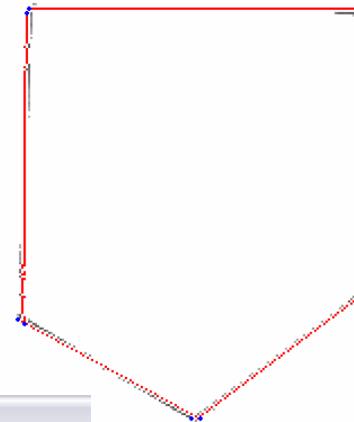
Molde original (bolso)



Retas identificadas



Pontos finais e iniciais identificados



```
Sistema de Detecção de Formas
Abrir Hough
molde111.raw.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
-28 135 188 61 232 142
0 49 50 66 50 212
36 269 176 215 232 138
90 64 52 65 190 64
90 213 51 213 178 213
```

# Detecção de formas circulares

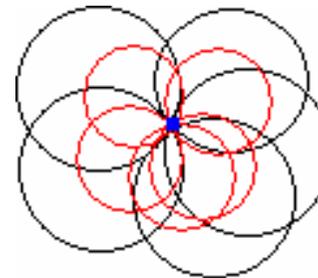
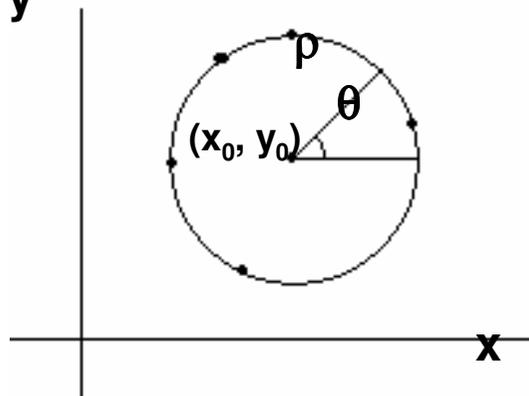
- Equação geral:  $(x - x_0)^2 + (y - y_0)^2 = r^2$
- Coordenadas polares:

$$x_0 = x - \rho \cos\theta$$

$$y_0 = y - \rho \sin\theta$$

**Espaço da imagem**      **A execução da técnica, pixel a pixel.**

onde  $x_0$  e  $y_0$  são as coordenadas do centro do círculo.



# Detecção de formas circulares

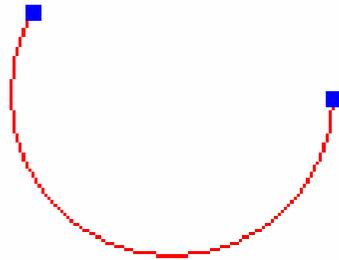
- A matriz acumuladora terá três dimensões para armazenar  $x_0$ ,  $y_0$  e  $\rho$ .

## **Discretização da matriz acumuladora:**

- $x_0 \rightarrow 0$  à  $N_i$  (coordenada máxima de  $x$  na imagem)
- $y_0 \rightarrow 0$  à  $N_j$  (coordenada máxima de  $y$  na imagem)
- $\rho \rightarrow$  raio mínimo ao raio máximo (considerando um intervalo de valores)

# Detecção de formas circulares

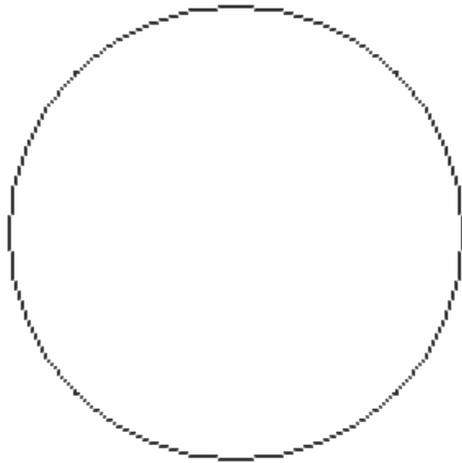
- Para detecção de arco de círculo, durante a execução da transformada de Hough inversa, ou seja, da extração dos parâmetros do círculo, a variável booleana **dentro** gerencia quando o pixel pertence ou não ao arco de círculo.



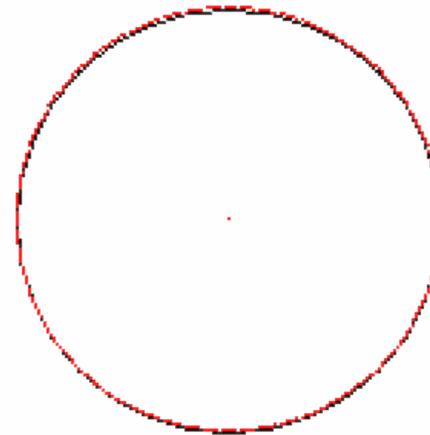
# Detecção de formas circulares

## TESTES

Imagem original



Forma detectada

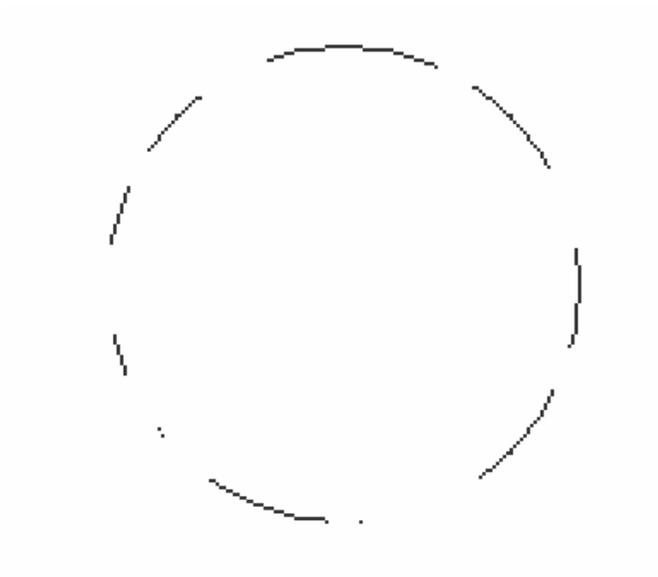


<b>(x0,y0)</b>	<b><math>\rho</math></b>	<b>Ponto inicial</b>	<b>Ponto final</b>	<b>Tempo de execução</b>
(95,102)	69	(0,0)	(0,0)	0,094s

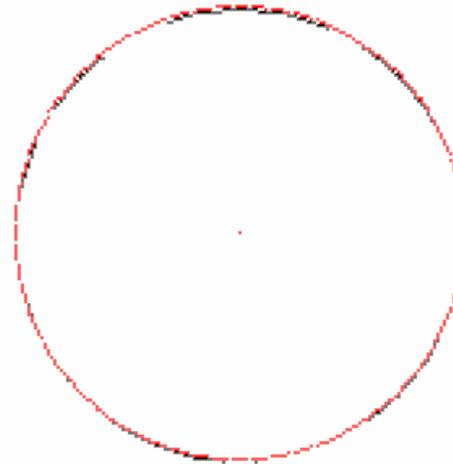
# Detecção de formas circulares

## TESTES

Imagem original



Forma detectada



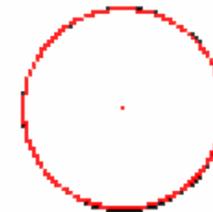
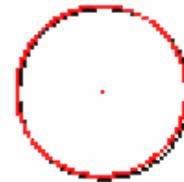
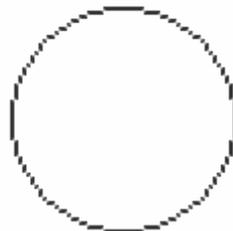
$(x_0, y_0)$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
(95,102)	69	-	-	0,062s

# Detecção de formas circulares

## TESTES

Imagem original

Círculos detectados



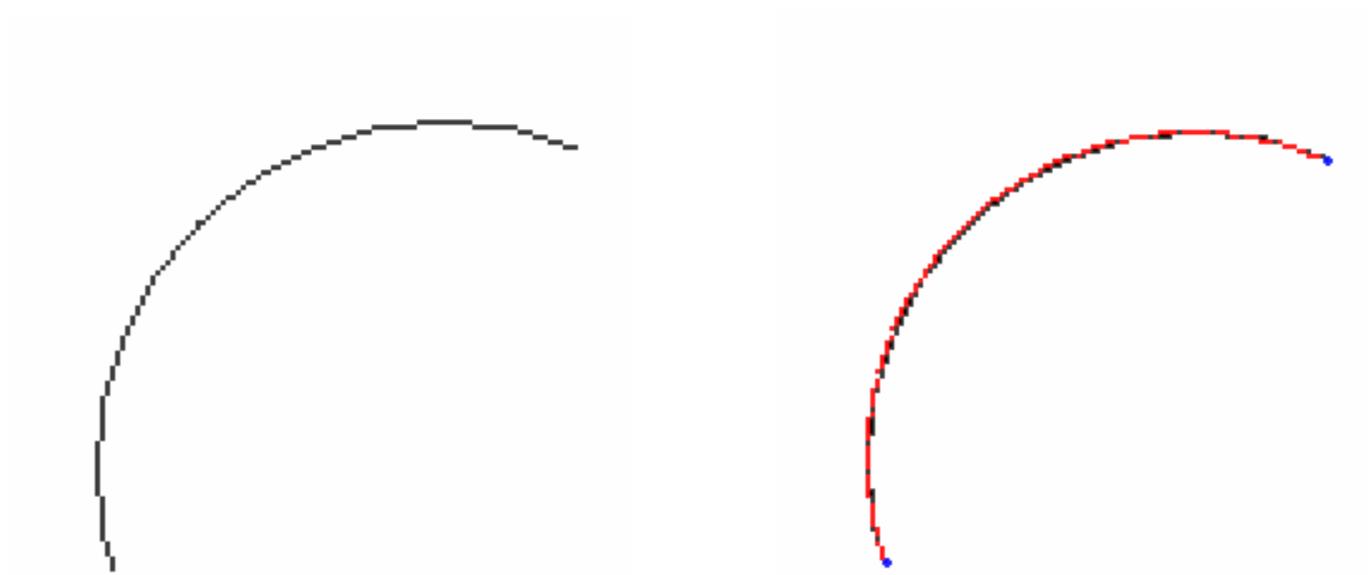
$(x_0, y_0)$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
(24,144)	11	(0,0)	(0,0)	1,157s
(39,37)	23	(0,0)	(0,0)	
(107,129)	27	(0,0)	(0,0)	

# Detecção de formas circulares

## TESTES

Imagem original

Forma detectada

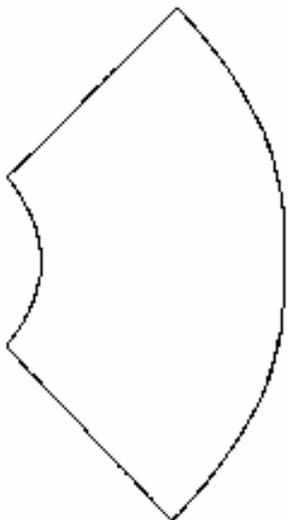


$(x_0, y_0)$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
(95,101)	68	(117,37)	(33,129)	0,47 s

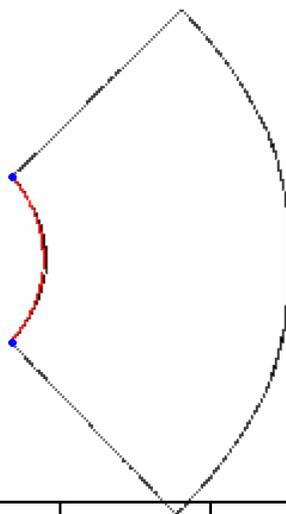
# Detecção de formas circulares

## TESTES

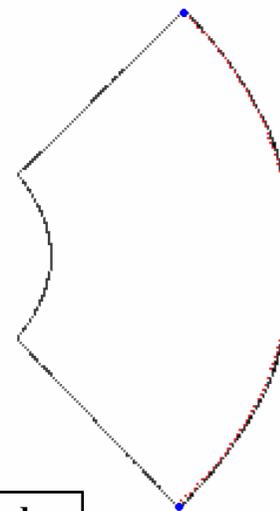
Molde original (saia)



Primeiro arco círculo detectado



Segundo arco de círculo detectado



$(x_0, y_0)$	$\rho$	Ponto inicial	Ponto final	Tempo de execução
(109,32)	51	(74,70)	(144,69)	0,10 s
(109,32)	140	(5,140)	(216,136)	0,11s

# Detecção de elipses

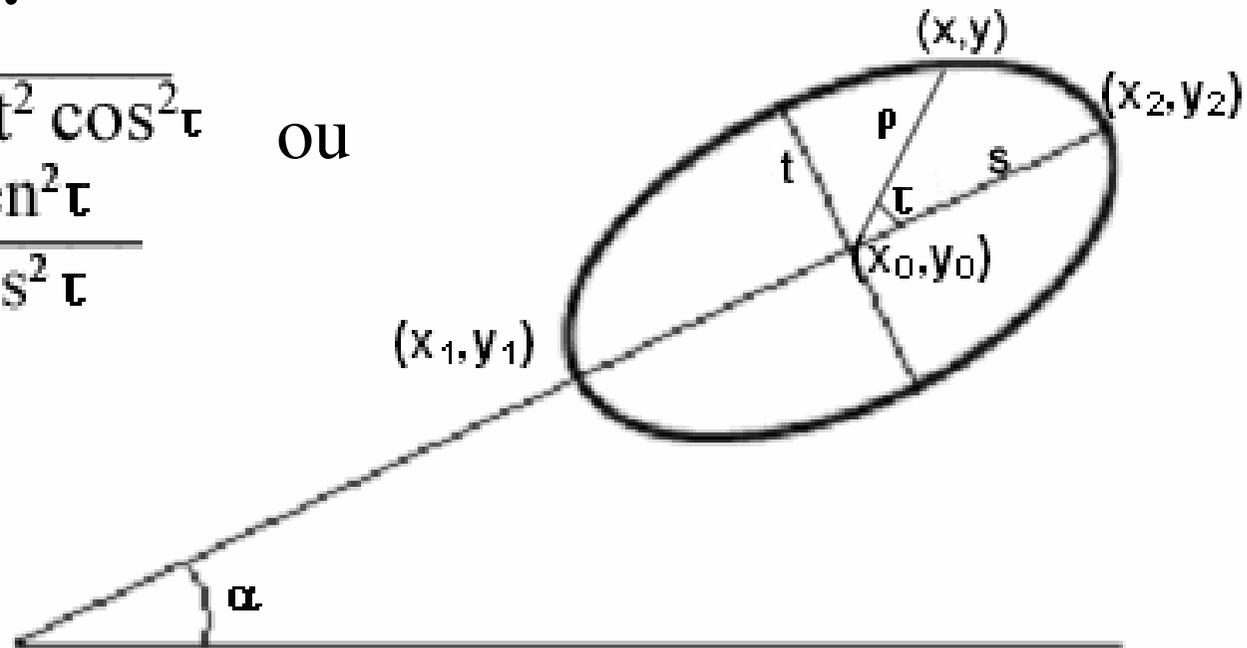
- Equação geral:  $\frac{x^2}{s^2} + \frac{y^2}{t^2} = 1$

- Forma polar:

$$\rho^2 = \frac{s^2 t^2}{s^2 \sin^2 \tau + t^2 \cos^2 \tau}$$

ou

$$t^2 = \frac{s^2 \rho^2 \sin^2 \tau}{s^2 - \rho^2 \cos^2 \tau}$$



# Detecção de elipses

- É suposto um tamanho para o **eixo maior** e para cada dois pontos candidatos a limite do eixo maior  $(x_1, y_1)$  e  $(x_2, y_2)$ , são calculados os parâmetros da elipse candidata:

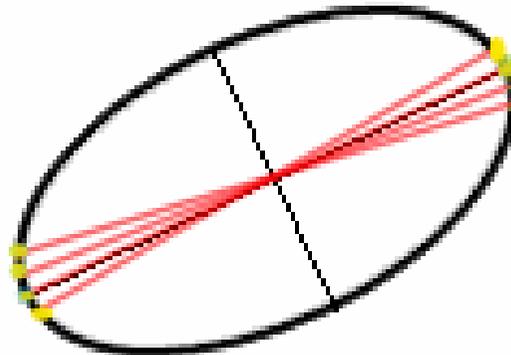
$$x_0 = \frac{x_1 + x_2}{2}$$

$$y_0 = \frac{y_1 + y_2}{2}$$

$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\alpha = \text{atan} \left[ \frac{(y_2 - y_1)}{(x_2 - x_1)} \right]$$

A execução da técnica, pixel a pixel.



# Detecção de elipses

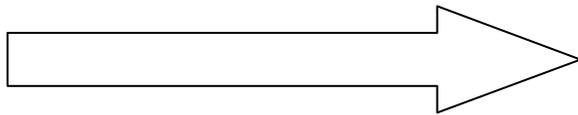
## Discretização da matriz acumuladora:

- $x_0 \rightarrow 1$  à  $N_i$  (coordenada máxima de  $x$  na imagem)
- $y_0 \rightarrow 1$  a  $N_j$  (coordenada máxima de  $y$  na imagem)
- $\alpha \rightarrow 0$  à 360 graus
- $2s \rightarrow 2$  à  $\sqrt{N_i^2 + N_j^2}$
- $2t \rightarrow 1$  à  $\sqrt{N_i^2 + N_j^2} - 1$  e  $2t < 2s$

# Detecção de elipses

Matriz acumuladora

t
20
10
4



Matriz armazenadora

s	t	x0	x0	$\alpha$	v
32	21	18	77	89	30
31	21	14	85	89	20
35	22	15	67	79	10
31	23	23	78	60	60

# Detecção de elipses

## TESTES

Imagem original



Forma detectada



<b>(x0,y0)</b>	<b>s</b>	<b>t</b>	<b><math>\alpha</math></b>	<b>Tempo de execução</b>
(70,94)	62	30	0	0,094s

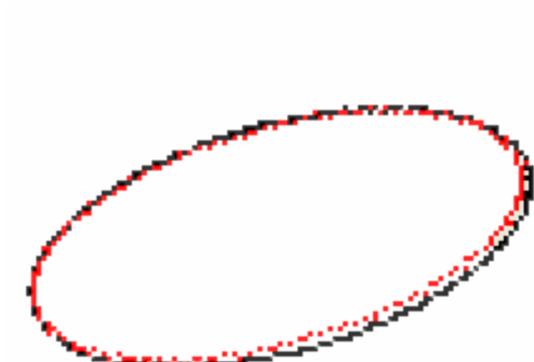
# Detecção de elipses

## TESTES

Imagem original



Forma detectada



$(x_0, y_0)$	s	t	$\alpha$	Tempo de execução
(82,92)	51	21	16	0,078s

# Detecção de elipses

## TESTES

Imagem original



Forma detectada



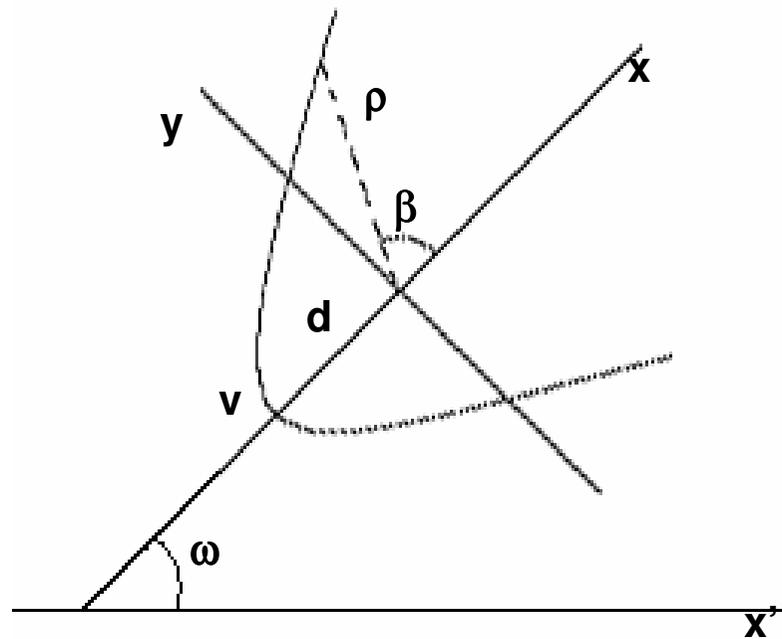
(x0,y0)	s	t	$\alpha$	Tempo de execução
(70,94)	62	30	0	0,047s

# Detecção de Parábolas

Equação geral:  $(x-x_0)^2 = 4d(y-y_0)$

Equação na forma polar:

$$\rho = \frac{2d}{1 - \cos \beta}$$

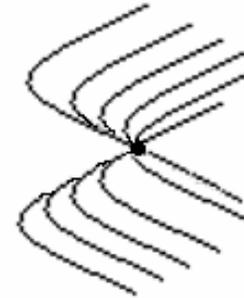


# Detecção de parábolas

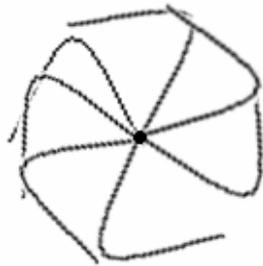
**Imagem original**



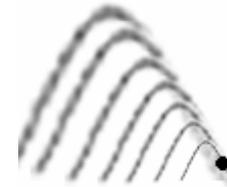
**Variação de foco**



**Variação de inclinação**



**Variação de tamanho**



# Detecção parábolas

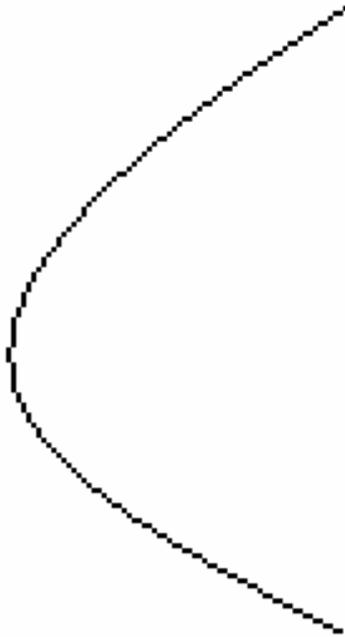
## Discretização da matriz acumuladora:

- $x_0 \rightarrow 0$  à  $N_i$  (coordenada máxima de  $\mathbf{x}$  na imagem)
- $y_0 \rightarrow 0$  à  $N_j$  (coordenada máxima de  $\mathbf{y}$  na imagem)
- $\mathbf{d} \rightarrow$  distância focal mínima à distância focal máxima
- $\omega \rightarrow 0$  a 360 graus

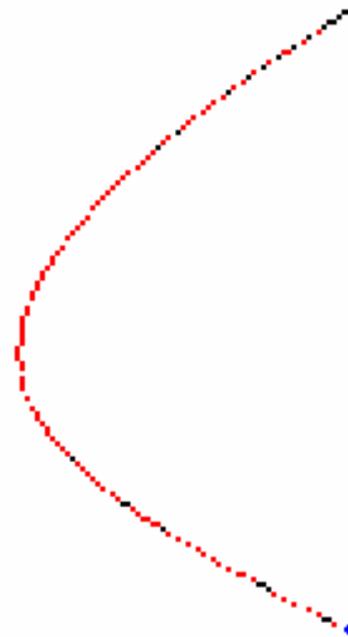
# Detecção parábolas

## TESTES

Imagem original



Forma detectada



$(x_f, y_f)$	d	Ponto inicial	Ponto final	$\alpha$	Tempo de execução
(64,76)	13	(130,118)	(7,119)	353	6,062s

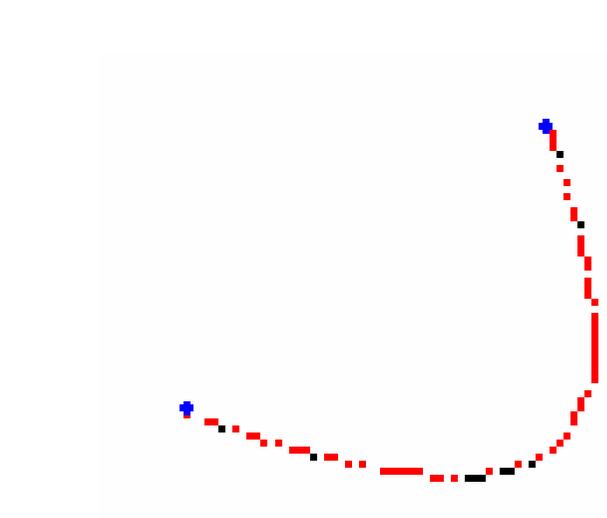
# Detecção parábolas

## TESTES

Imagem original



Forma detectada

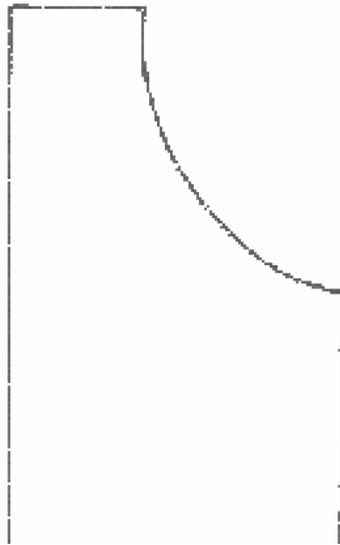


$(x_f, y_f)$	d	Ponto inicial	Ponto final	$\alpha$	Tempo de execução
(67, 62)	7	(22,69)	(62,18)	223	4, 453s

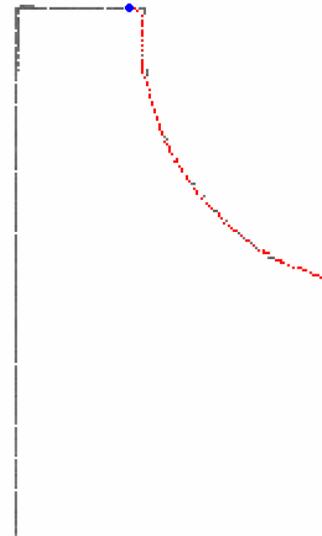
# Detecção parábolas

## TESTES

Imagem original



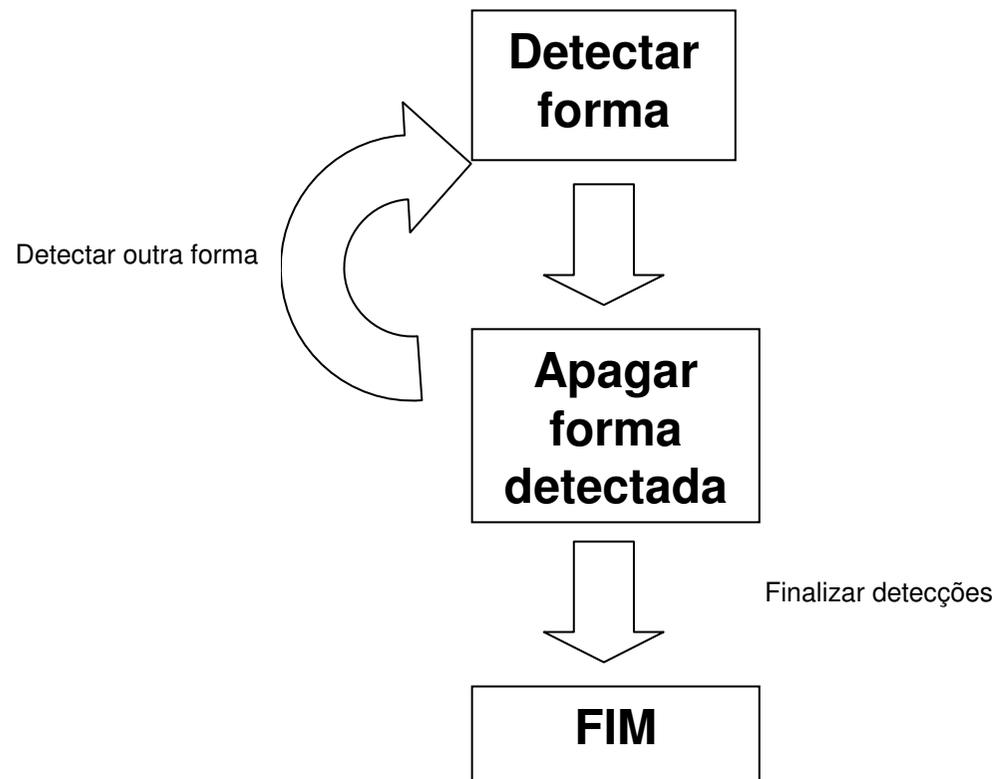
Forma detectada



$(x_f, y_f)$	d	Ponto inicial	Ponto final	$\alpha$	Tempo de execução
(135, 84)	51	(129,144)	(19,62)	323	21,73s

# Detecção de formas

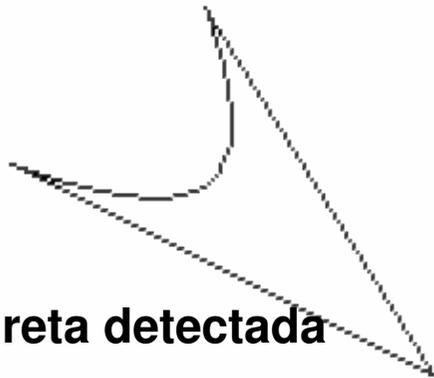
Esquema de detecção de várias formas diferentes.



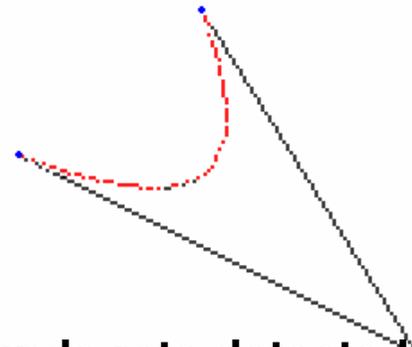
# Detecção de formas

Figura constituída de duas retas e uma parábola.

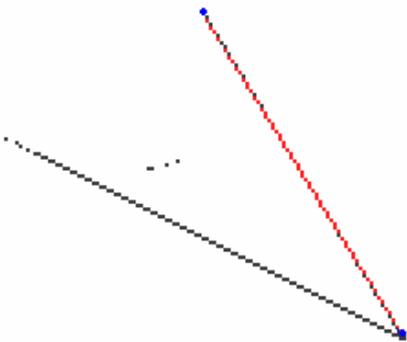
Imagem original



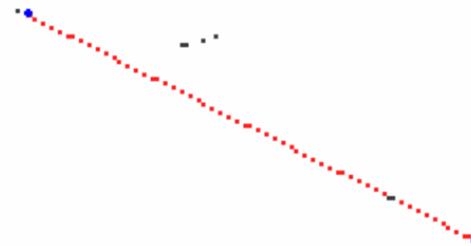
Parábola detectada



Primeira reta detectada



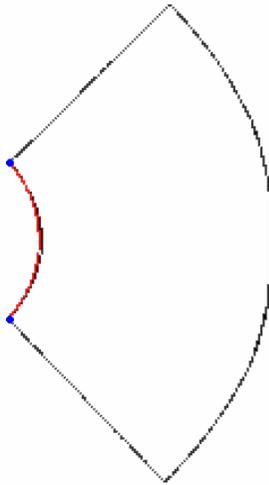
Segunda reta detectada



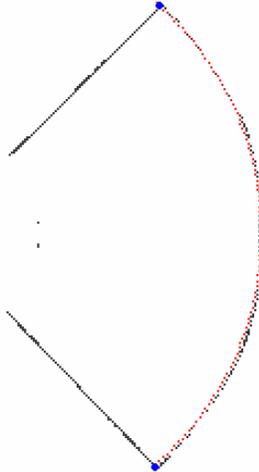
# Detecção de formas

Molde constituído de duas retas e dois arcos de círculo.

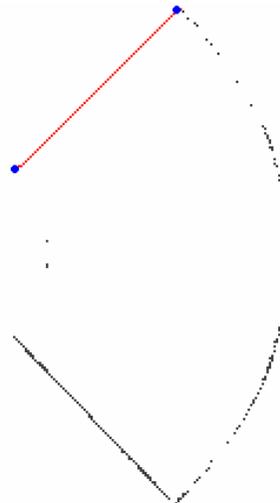
**Primeiro arco  
de círculo**



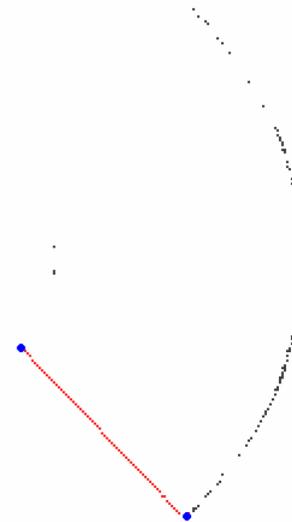
**Segundo arco  
de círculo**



**Primeira reta**



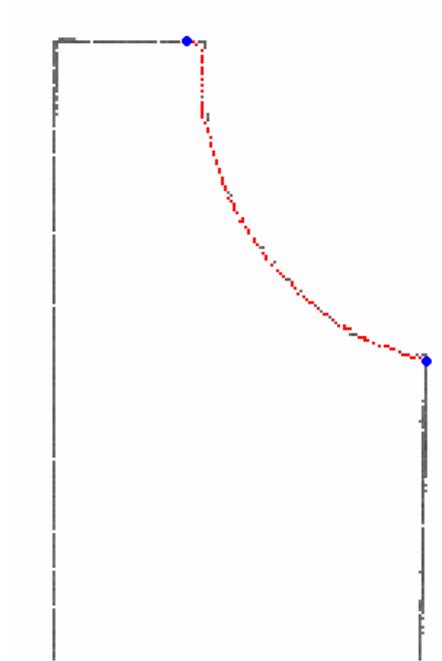
**Segunda reta**



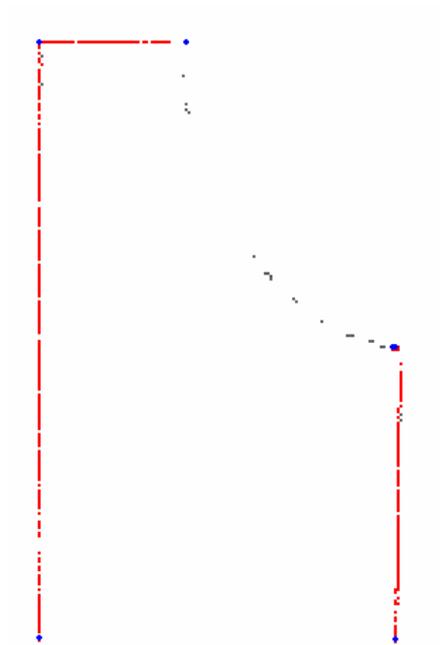
# Detecção de formas

Molde constituído de três retas e uma parábola.

Parábola detectada



Três retas detectadas



# Outras Aplicações

## Ensaio de Dureza

- A detecção de círculos por transformada de Hough também é útil na aplicação de ensaios de dureza .
- Os ensaios de dureza [BEL2003] são feitos por aparelhos chamados durômetros que verificam o quanto um dado material é resistente.
- O ensaio de dureza Brinell consiste em comprimir lentamente uma esfera de aço temperado de diâmetro  $D$ , sobre uma superfície plana, polida e limpa de um metal, produzindo uma calota esférica de diâmetro  $D'$ .
- O ensaio de dureza Vickers consiste em comprimir uma pirâmide de diamante de base quadrada e ângulo entre faces de  $136^\circ$ .

Filtros passa alta utilizados

Brinell

-1	-1	-1
-1	12	-1
-1	-1	-1

Vickers

-1	-1	-1
-1	10	-1
-1	-1	-1

# Outras Aplicações

## Ensaio de dureza Brinell

Imagem original

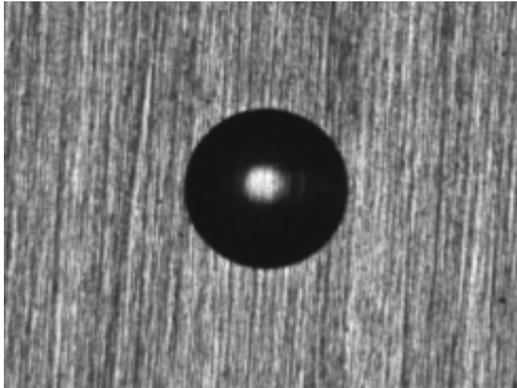


Imagem após filtragem

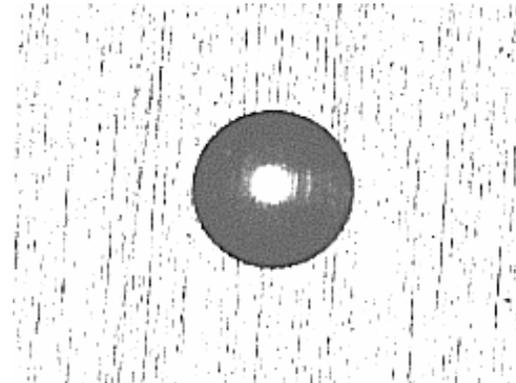
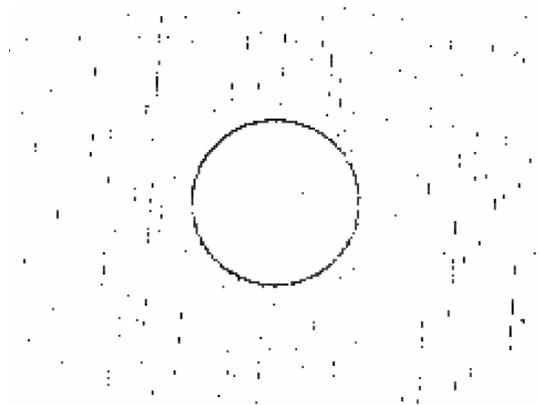
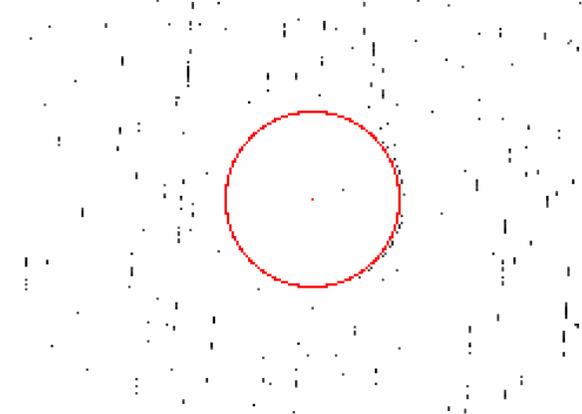


Imagem após aplicação de limiar 80



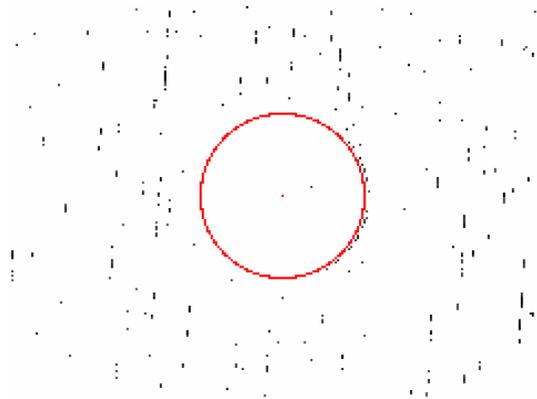
Círculo detectado



# Outras Aplicações

## Ensaio de dureza Brinell

### Círculo detectado



Intervalo considerado: 37,4 a 40,0 pixels

<b>(x0,y0)</b>	<b><math>\rho</math></b>	<b>raio real</b>	<b>erro</b>	<b>Tempo de execução</b>
(88,123)	74,8pixels	75 pixels	0,07cm	5,14 s

# Outras Aplicações

## Ensaio de dureza Vickers

Imagem original

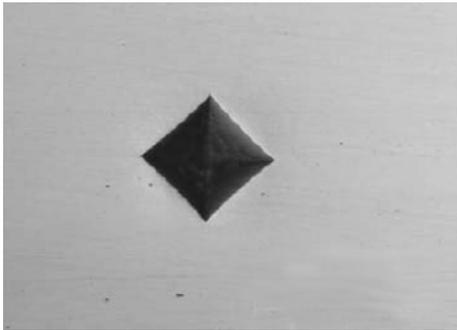


Imagem após filtragem

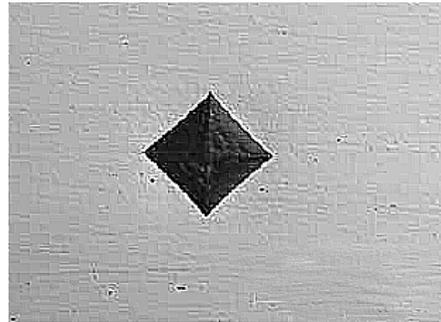
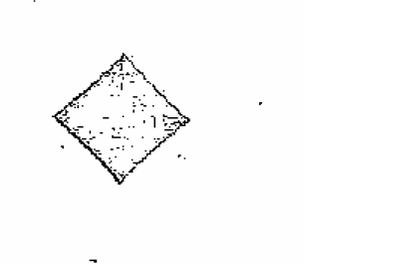
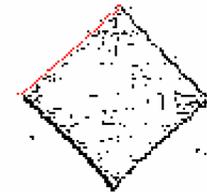
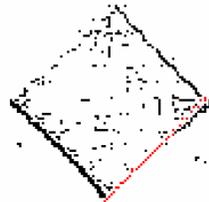
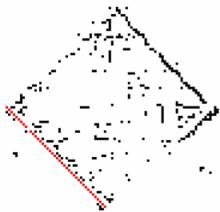


Imagem após limiar de 0



Detecção de cada reta presente na imagem



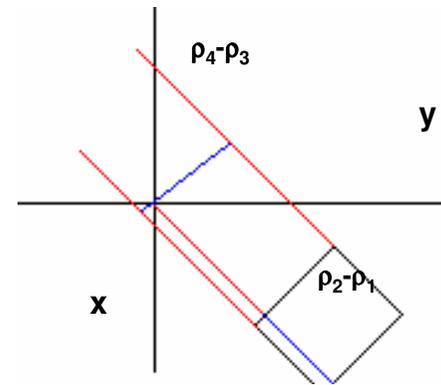
# Outras Aplicações

## Ensaio de dureza Vickers

Arquivo gerado pela execução do teste

```
teste_dureza10.raw.txt - Blo  
Arquivo Editar Formatar Exibir Aju  
40 143 66 144 106 95  
40 202 108 186 206 71  
-46 1 105 97 150 140  
-46 -58 64 142 108 185
```

Esquema de cálculo da aresta através do parâmetro  $\rho$ .



$\rho_2-\rho_1$	$\rho_4-\rho_3$	Diagonal	Diagonal real	erro
59 pixels	59 pixels	83,4 pixels	86 pixels	0,08cm

## Controle de Qualidade

- A pesquisa com base em controle de qualidade é questão fundamental para o provimento de regularidade e uniformidade de um produto. A indústria farmacêutica é uma das que preza pela qualidade de seus medicamentos, pois um comprimido a ser enviado ao mercado, não pode estar com tamanho fora do padrão, imperfeito ou danificado.
- A indicação de pontos limite de um arco de círculo pode indicar falha, quando se trata de comprimidos com forma circular. [MAC2004]

# Outras Aplicações

Imagem original de pílulas de forma circular, onde uma apresenta falha



Imagem pré-processada

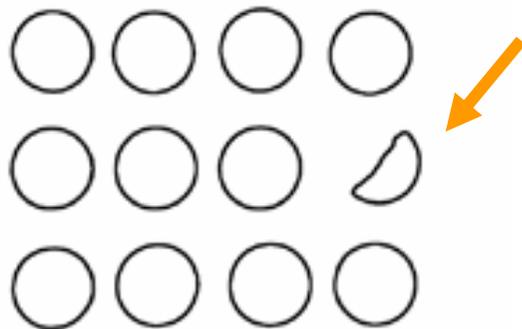
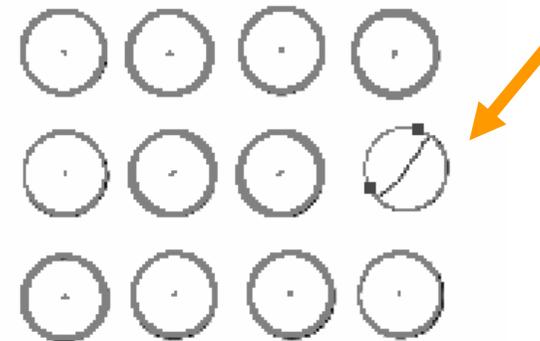


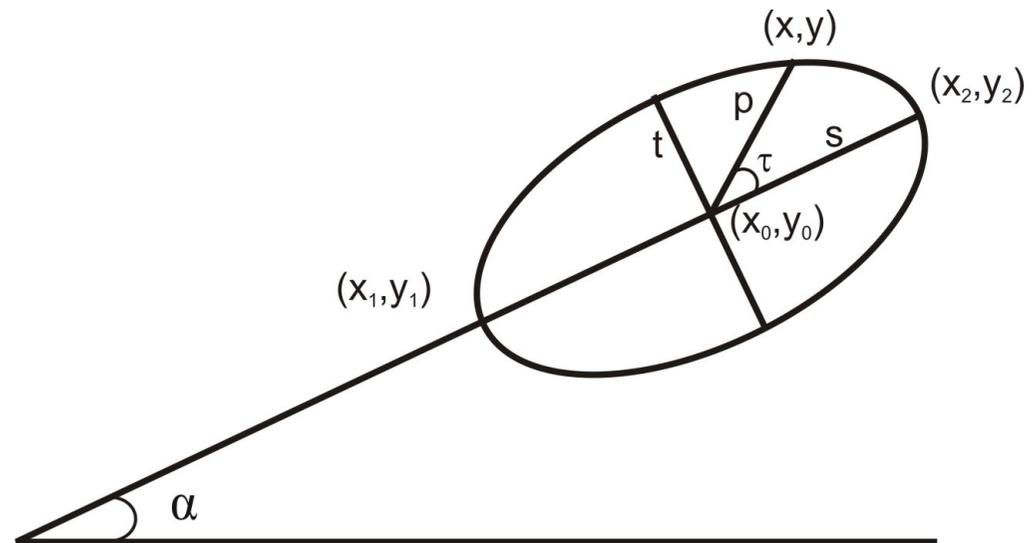
Imagem detectada. Os dois pontos no início e fim do fragmento, indicam falha.



## Transformada de Hough para elipses

Elipses:

$$\rho^2 = \frac{s^2 t^2}{s^2 \sin^2 \tau + t^2 \cos^2 \tau}$$

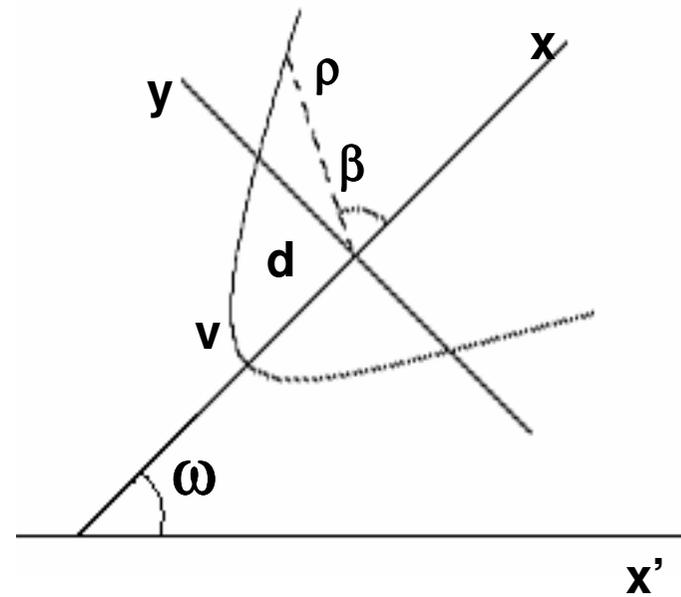


**Elipse inclinada com seus cinco parâmetros e coordenadas inicial e final de seu maior eixo.**

# Transformada de Hough para parábolas

Parábolas:

$$\rho = \frac{2d}{1 - \cos \beta}$$



Parábola rotacionada.

No exemplo de mamografia, a TH pode ser usada para retirar o músculo peitoral



(a)

(b)

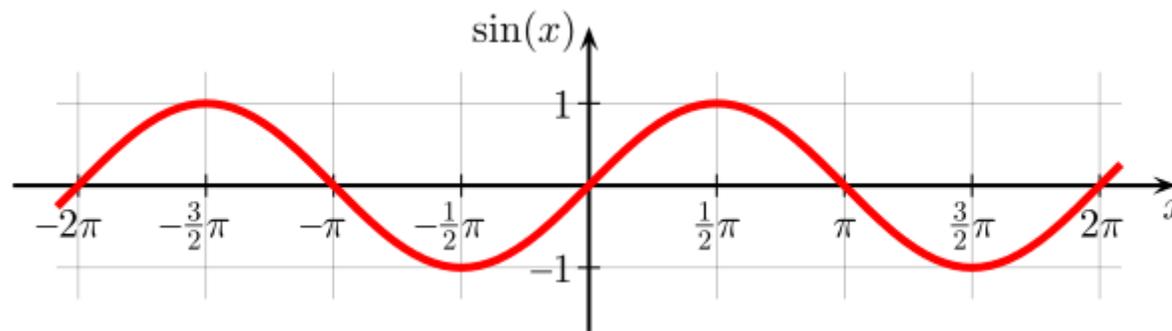
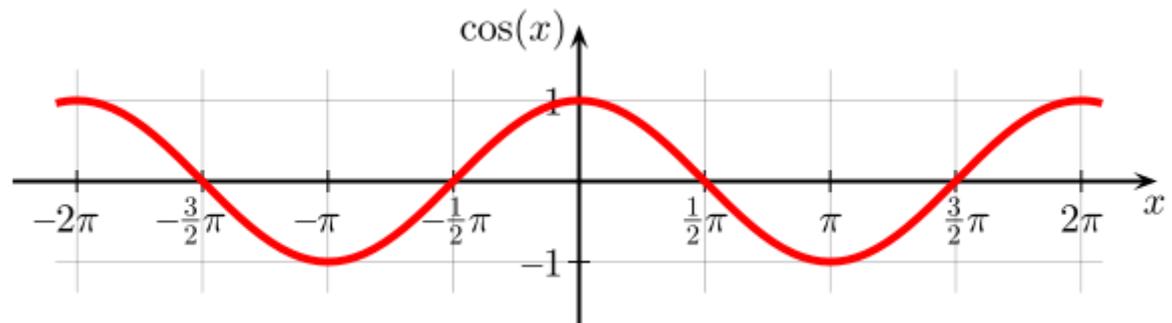
(c)

(d)

(e)

Remoção do músculo peitoral. a) Imagem MLO. b) Área do músculo localizada. (c) Bordas de mesma direção que o músculo peitoral localizadas pelo filtro de Canny e filtradas por erosão morfológica. d) Reta detectada pela transformada de Hough; e) Imagem sem o músculo peitoral.

# CARACTERISTICAS DE SENOS E COSENOS



# Transformada Discreta do Co-seno (DCT)

Transforma discreta de co-senos em 2-D :

$$T[i, j] = c(i, j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I[y, x] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N}$$

Coeficientes  $c(i, j)$ :

$$c(i, j) = \sqrt{\frac{2}{N}} \quad , \text{ para } i \text{ e } j \neq 0 \text{ e}$$

$$c(i, j) = \sqrt{\frac{1}{N}} \quad , \text{ para } i \text{ e } j = 0$$

Transformada Inversa IDCT 2-D:

$$I[y, x] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c(i, j) T[i, j] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N}$$

Essa compressão é usada no formato JPEG padrão com valor de  $N$  igual a 8.

# Transformada de *Wavelets*

Perpectiva Histórica

Análise de *Wavelet*

Transformada de *Wavelet* Contínua

Transformada de *Wavelet* Discreta

Semelhanças entre Transformada de Fourier e *Wavelet*

Diferenças entre Transformada de Fourier e Transformada de *Wavelet*

*Wavelets* Unidimensional

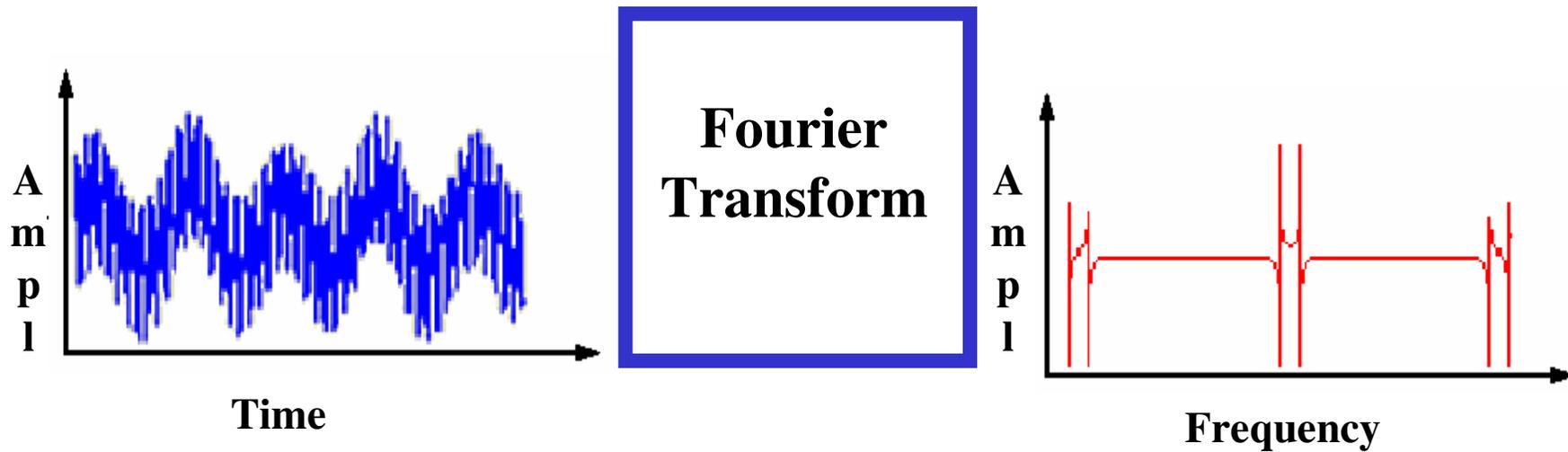
*Wavelet* Bidimensional

Aproximações e Detalhes

Banco de Filtros

# Introduction

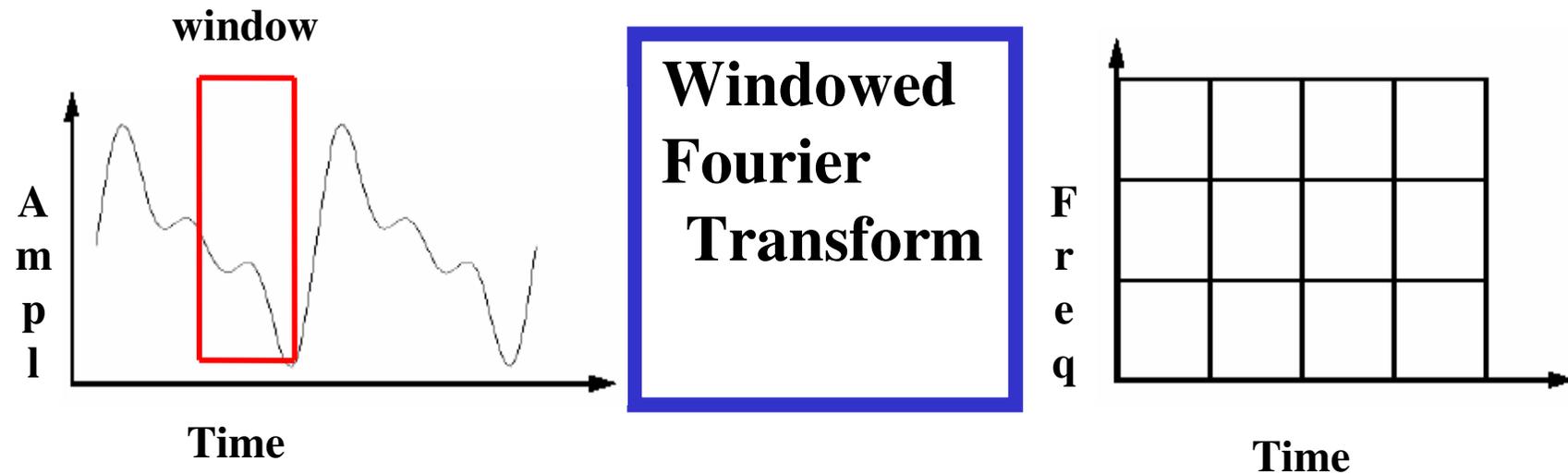
## Fourier Transform



Joseph Fourier (1807)

# Introduction

## Windowed Fourier T.



Dennis Gabor (1946)

# Transformada de *Wavelet* Discreta

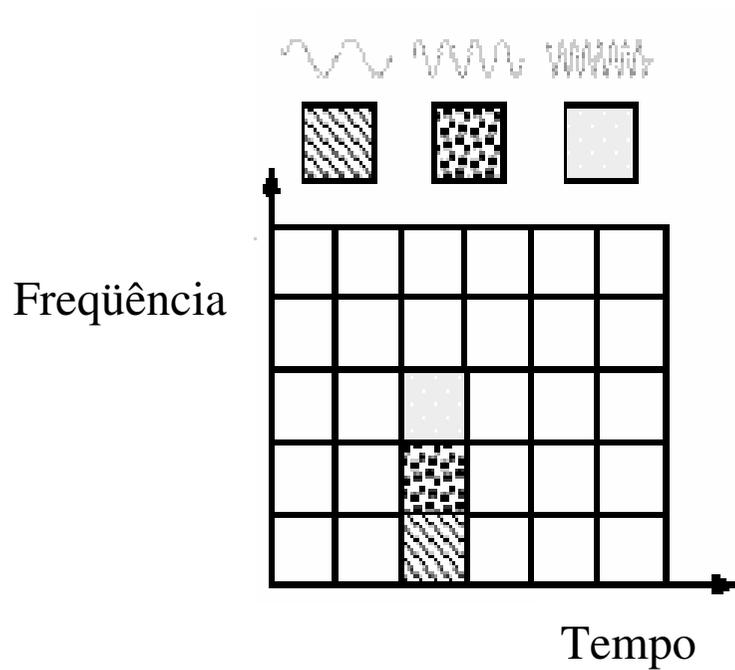
$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right), \quad a = 2^j, b = k 2^j, \quad (j,k) \in \mathbb{Z}^2$$

# Semelhanças entre Transformada de Fourier e *Wavelet*

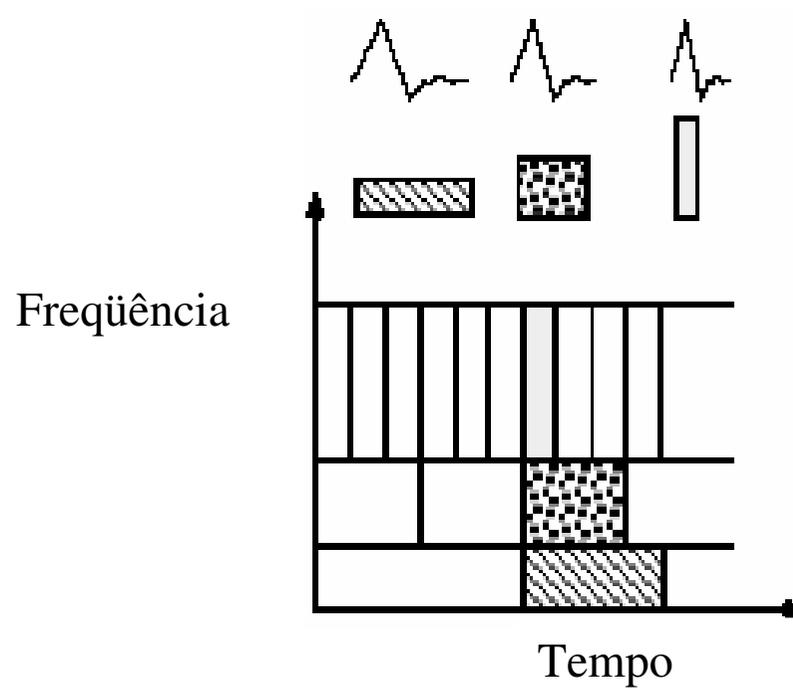
- São ambas operações lineares que geram uma estrutura de dados que contém segmentos de vários comprimentos.
- As transformadas matriciais inversas da FFT e da DWT são as transpostas das originais.
- As funções base são localizadas no domínio da frequência tornando-as ferramentas matemáticas poderosas na análise espectral de potência.

# Diferenças entre Transformada de Fourier e Transformada de *Wavelet*

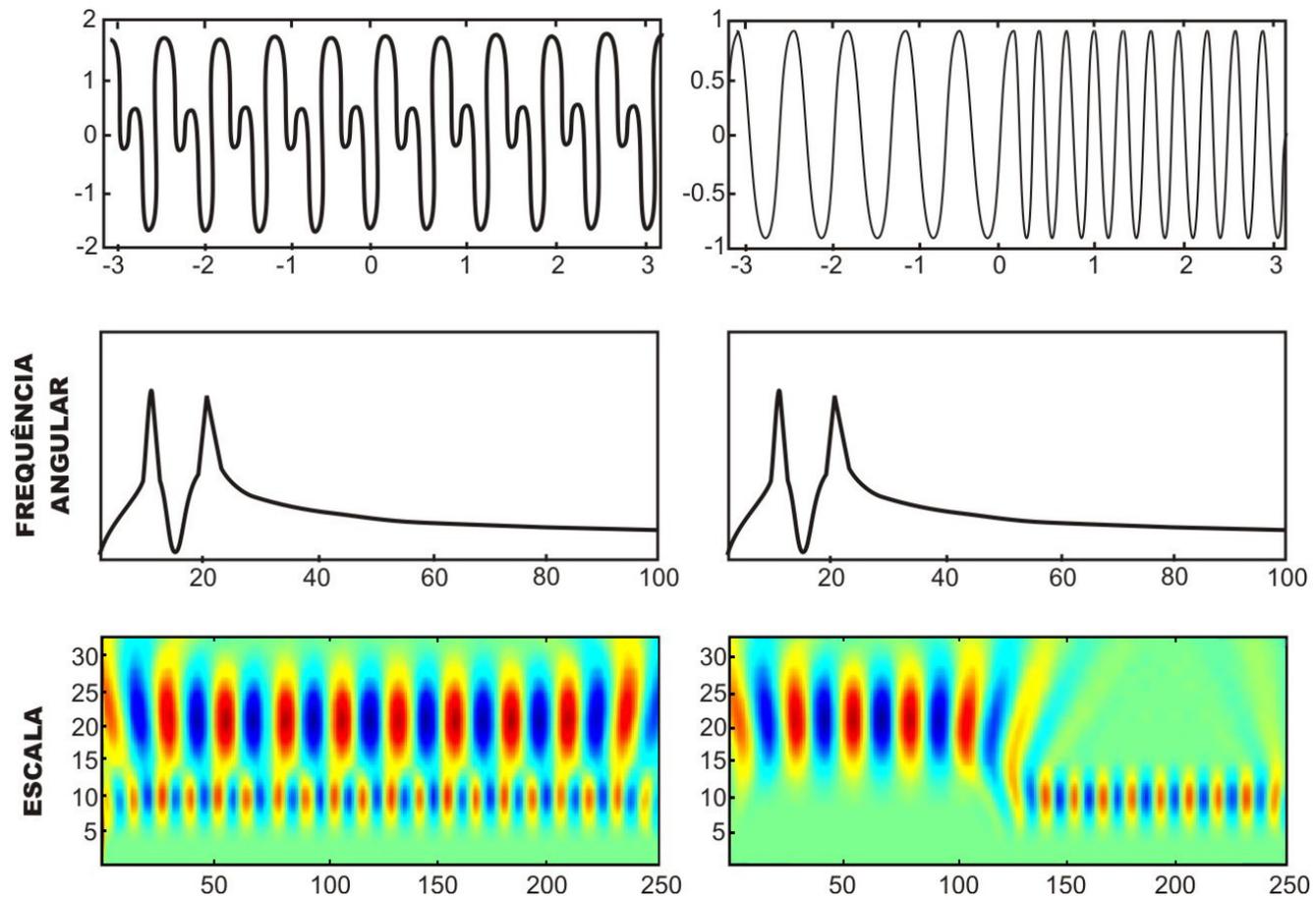
- As funções individuais *wavelet* são *localizadas no espaço*. Já as funções de Fourier, seno e co-seno não são.
- Esta característica junto com a localização em frequência das *wavelets*, fazem muitos operadores e funções usarem *wavelets esparsas* quando transformados para o domínio de *wavelet*.



**Funções base de Fourier  
descritas no plano Tempo x  
Frequência.**



**Função base *wavelet* de Daubechies  
descritas no plano Tempo x  
Frequência.**



**Comparação entre Transformada de Fourier e Transformada de *Wavelet*.**

# ***Wavelets Unidimensional***

Transformada *Wavelet* de Haar Unidimensional  
Funções bases de *Wavelet* de Haar  
Unidimensional  
Ortogonalidade  
Compressão

# Transformada *Wavelet* de Haar Unidimensional

Exemplo:

1- Suponha uma seqüência de uma dimensão com uma resolução de quatro *pixels*, tendo valores: [ 9 7 3 5 ].

2- Calcule primeiro a média dos valores em pares, obtendo os novos valores em resolução da imagem:[ 8 4 ].

3- Armazene alguns *coeficientes de detalhes*, que capturam a informação perdida. No exemplo: 1.

4- Repetindo este processo recursivamente até a decomposição tem-se a tabela:

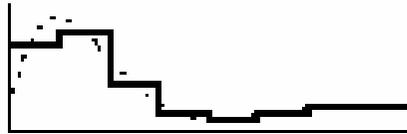
Decomposição em coeficientes de aproximação e detalhes.

Resolução	Média / Valores	Coeficientes de Detalhes
4	[ 9 7 3 5 ]	
2	[ 8 4 ]	[ 1 -1 ]
1	[ 6 ]	[ 2 ]

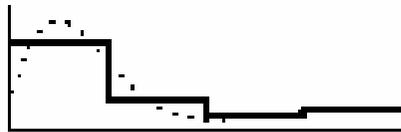
Para a base de Haar unidimensional, a transformada de *wavelet* da imagem original de quatro *pixels* é dada por:[ 6 2 1 -1 ]



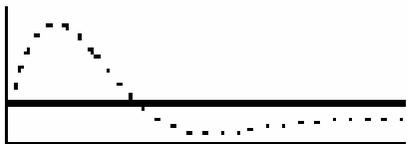
Aproximação  $V^4$



Aproximação  $V^3$



Aproximação  $V^2$

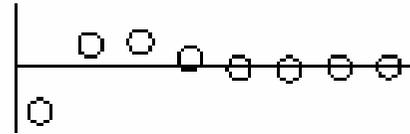


Aproximação  $V^1$

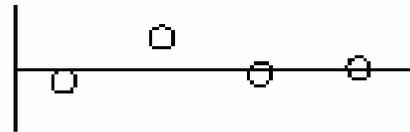


Aproximação  $V^0$

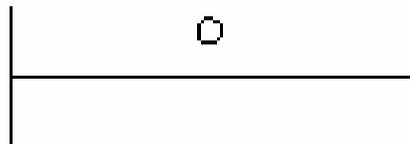
Seqüência de aproximação e coeficientes de detalhes.



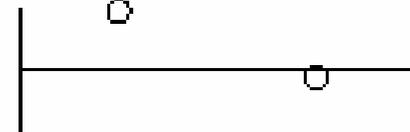
Coeficientes de detalhes  $W^3$



Coeficientes de detalhes  $W^2$



Coeficientes de detalhes  $W^1$



Coeficientes de detalhes  $W^0$

# Funções bases de *Wavelet* de Haar Unidimensional

*Wavelets* são coleções de funções  $\psi_i^j(x)$  linearmente independente que geram o espaço  $W^j$ .

Estas funções de base têm as seguintes propriedades:

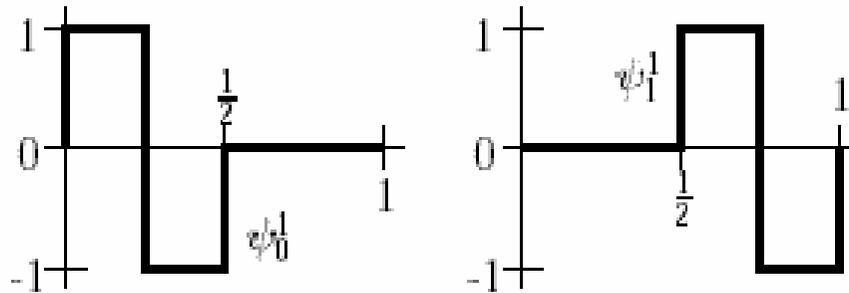
1. As *wavelets*  $\psi_i^j$  bases de  $W^j$ , juntamente com as funções base  $\phi_i^j$  de  $V^j$  formam a base para  $V^{j+1}$ .
2. Toda a função base  $\psi_i^j$  de  $W^j$  são ortogonais a todas as bases  $\phi_i^j$  de  $V^j$  sob um certo produto interno escolhido.

As *wavelets* que correspondem à base quadrada são conhecidas como *wavelets* de Haar e são dadas por:

$$\psi_i^j(x) := \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1$$

Onde:

$$\psi(x) := \begin{cases} 1 & \text{se } 0 \leq x < 1/2 \\ -1 & \text{se } 1/2 \leq x < 1 \\ 0 & \text{se } x < 0 \text{ ou } x \geq 1 \end{cases}$$



As *wavelets* de Haar para  $W^1$ .

Expressando a imagem original unidimensional como uma combinação linear das funções de base quadrada em  $V^2$ ; pode-se escrever:

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x)$$

$$\begin{aligned}
 \mathcal{I}(x) &= 9 \times \text{[rect]} \\
 &+ 7 \times \text{[rect]} \\
 &+ 3 \times \text{[rect]} \\
 &+ 5 \times \text{[rect]}
 \end{aligned}$$

Descrevendo a imagem por  $V^2$

É possível reescrever a expressão para  $\mathcal{I}(x)$  em termos das funções base em  $V^l$  e  $W^l$  e , usando coeficientes de média:

$$\mathcal{I}(x) = c_I^0 \phi_I^0(x) + c_I^I \phi_I^I(x) + \alpha_I^0 \mathcal{M}_I^0(x) + \alpha_I^I \mathcal{M}_I^I(x)$$

$$\begin{aligned}
&= 8 \times \text{[Step function 1]} \\
&+ 4 \times \text{[Step function 2]} \\
&+ 1 \times \text{[Step function 3]} \\
&+ -1 \times \text{[Step function 4]}
\end{aligned}$$

Descrevendo a imagem por média e detalhes

Pode-se reescrever  $I(x)$  como uma soma de funções de base em  $V^0$ ,  $W^0$  e  $W^1$ :

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x)$$

$$\begin{aligned}
 &= 6 \times \text{[rectangle]} \\
 &+ 2 \times \text{[rectangle]} \\
 &+ 1 \times \text{[rectangle]} \\
 &+ -1 \times \text{[rectangle]}
 \end{aligned}$$

Descrevendo a imagem como média total e detalhes

# Ortogonalidade

- Uma base ortogonal é aquela que todas as funções base, isto é  $\phi_0^0, \psi_0^0, \psi_0^1, \psi_1^1, \dots$ , são ortogonais entre si.
- A base de Haar possui propriedade de ortogonalidade.

# Normalidade

Uma função base  $u(x)$  é normalizada se  $\langle u | u \rangle = 1$

Pode-se normalizar a base de Haar substituindo as equações por:

$$\phi_i^j(x) := 2^{j/2} \phi(2^j x - i), \quad i = 0, \dots, 2^j - 1$$

$$\psi_i^j(x) := 2^{j/2} \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1$$

No exemplo, os coeficientes não normalizados [6 2 1 -1] se

tornam:  $\begin{bmatrix} 6 & 2 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$

# Compressão

O objetivo da compressão é expressar um conjunto inicial de dados usando outro conjunto menor, com ou sem perda de informação.

Suponha a imagem  $f(x)$  expressa pela soma de funções base :

$$f(x) = \sum_{i=1}^m c_i u_i(x)$$

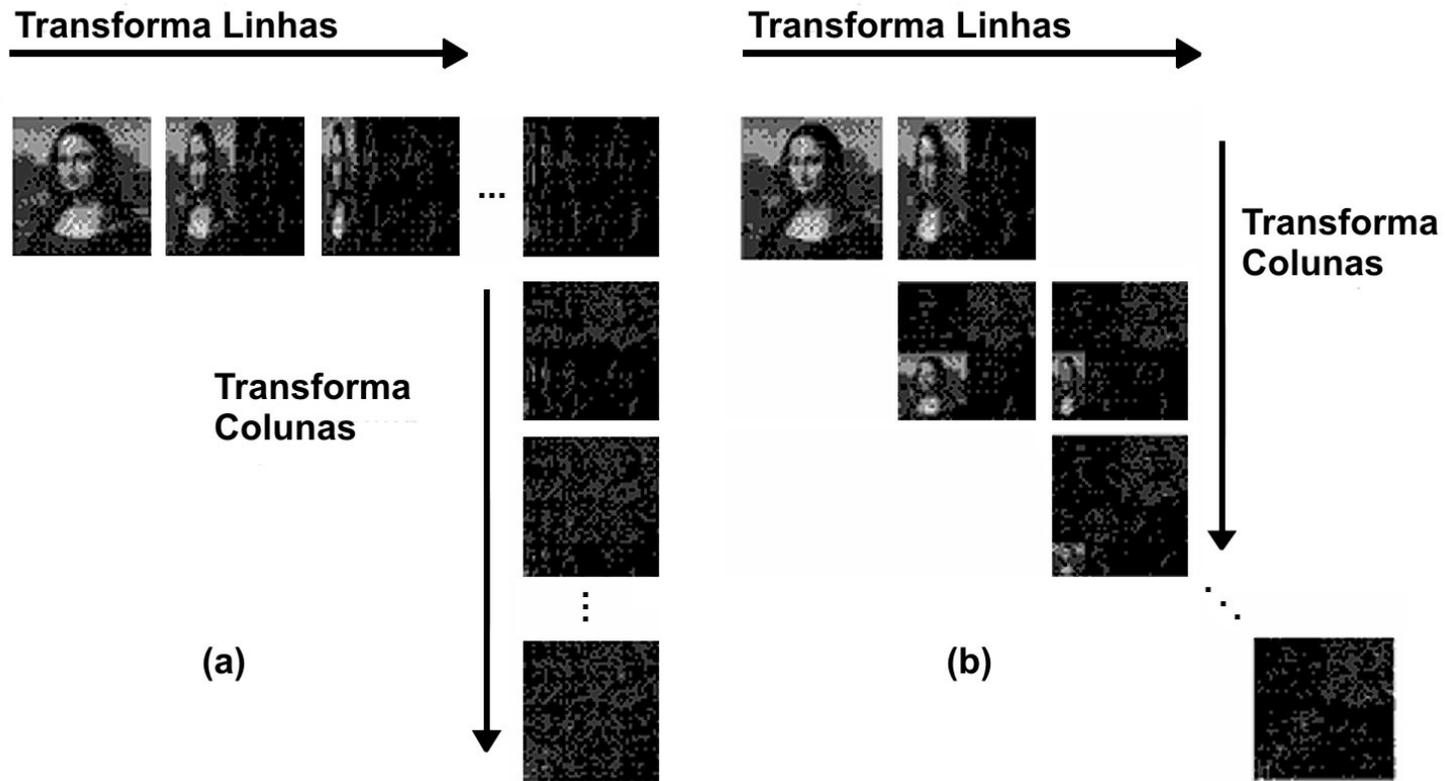
O conjunto de dados neste caso consiste de coeficientes  $c_i$ . Procura-se uma função que aproxima  $f(x)$ , mas com menos coeficientes:

$$\tilde{f}(x) = \sum_{i=1}^{\tilde{m}} \tilde{c}_i \tilde{u}_i(x) \cong f(x)$$

# ***Wavelet* Bidimensional**

Para entender a compressão de imagem, descrevem-se as funções escalar e de *wavelets* que formam as bases de *wavelet* bidimensionais.

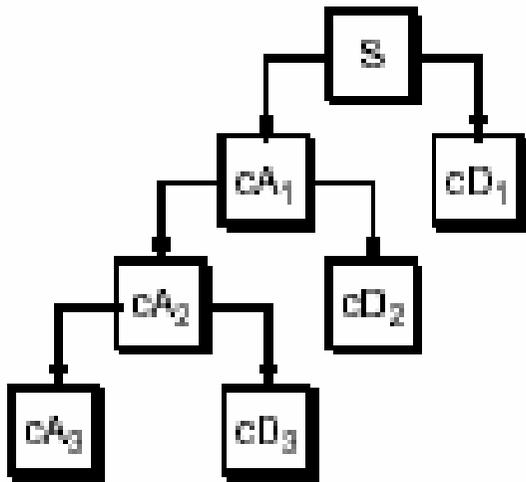
# Transformada de *Wavelet* de Haar bidimensional



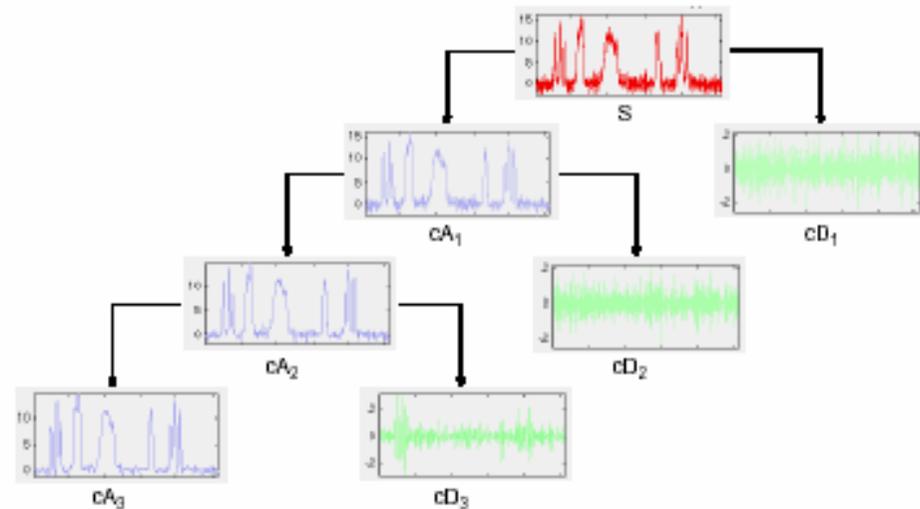
(a) Decomposição padrão,

(b) Decomposição não padrão.

# Aproximações e Detalhes



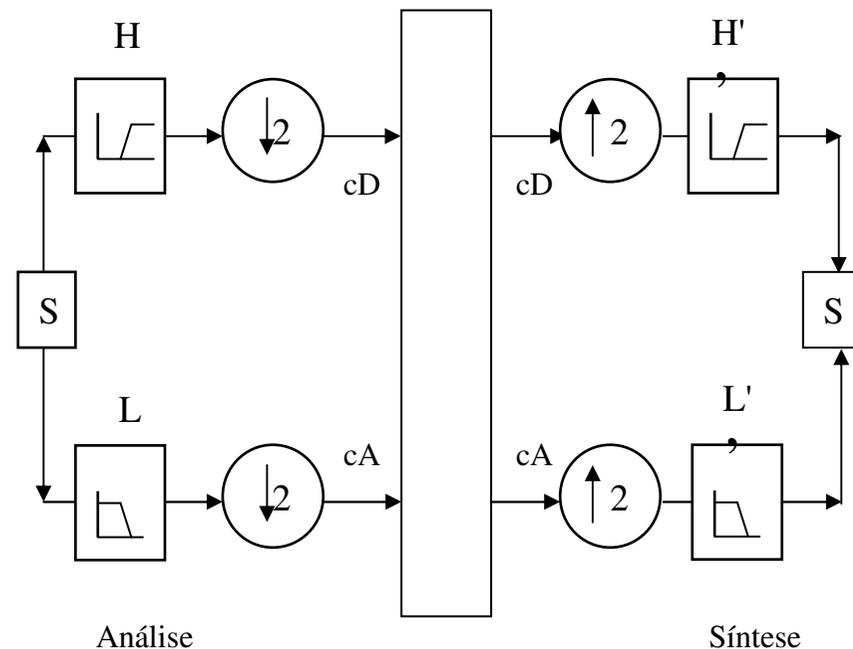
Árvore de Decomposição  
*Wavelet*



Árvore de Decomposição *Wavelet* de um  
sinal

# Banco de Filtros

A forma de implementação eficiente do algoritmo de compressão por *wavelet* é por intermédio de um Banco de Filtros em Quadratura Conjugado.



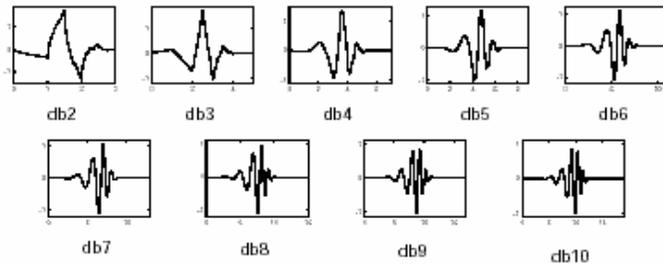
Banco de Filtros

# wavelet families : 36 variations

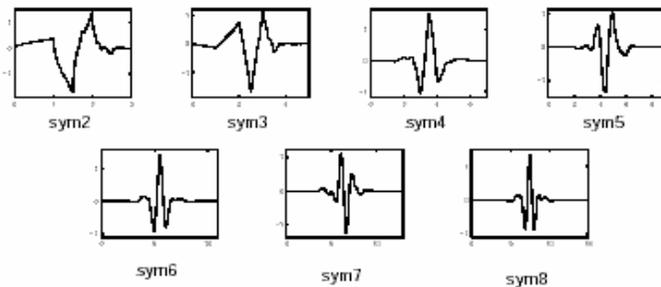


Haar

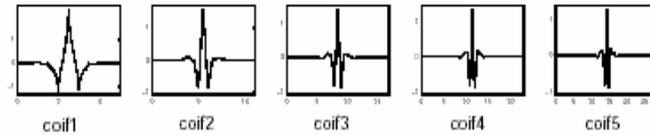
## Daubechies



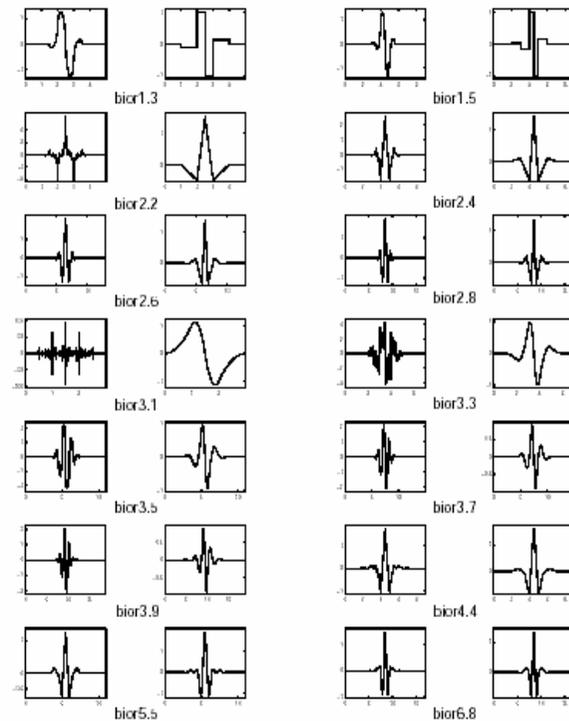
## Symlets



## Coiflets



## Biortogonal



# Transformada de Gabor

- A transformada de Fourier janelada ("windowed Fourier transform") ou transformada de Fourier a curto-termo ("short-time Fourier transform") - janela de observação sobre o sinal:

$$U(b, f) = \int_{-\infty}^{\infty} \rho^*(t-b) u(t) e^{-i2\pi f t} dt$$

em que  $p(t)$  é a função que cumpre o papel de janela, a qual desliza sobre a função  $u(t)$  .

# Transformada de Gabor

A **transformada de Gabor**: transformada janelada de Fourier com uma janela gaussiana.

Restrição:  $p(t)$  deve ser bem localizada simultaneamente nos domínios do tempo e da frequência.

Propriedade de localização no tempo da transformada  $U(b,f)$ .

A transformada de Gabor em termos das transformadas de Fourier do sinal e da janela:

A transformada de Gabor também realiza uma análise local no domínio da frequência em termos de magnitude.

$$U(b, f) = e^{-i2\pi fb} \int_{-\infty}^{\infty} G^*(v - f) U(v) e^{i2\pi vb} dv$$

## Transformada de Gabor

- A estrutura local da textura, nos vários canais utilizados, é descrita pela freqüência e pela orientação, que são definidas pela função sinusoidal no filtro de Gabor.
- O envelope Gaussiano define as resoluções espaciais com que as estruturas de textura são caracterizadas.
- A forma geral da Função de Gabor bidimensional é dada através da equação:

$$g(x, y, \sigma_x, \sigma_y, k_x, k_y) = A \cdot \exp \left\{ -\frac{1}{2} \left[ \left( \frac{x}{\sigma_x} \right)^2 + \left( \frac{y}{\sigma_y} \right)^2 \right] + 2\pi i (k_x x + k_y y) \right\}$$

onde  $A$  é um fator de escala e  $(\sigma_x, \sigma_y)$  definem a extensão espacial da função de Gabor. A orientação da grade sinusoidal é fornecida por  $|k| = \sqrt{k_x^2 + k_y^2}$

A freqüência da onda senoidal é dada por  $k = (k_x, k_y)$ , onde  $k_x, k_y$  são suas componentes ao longo dos eixos  $x$  e  $y$ , respectivamente.

# Transformada de Gabor

- Filtro linear bi-dimensional e não variante ao deslocamento.
- Pode ser entendido como o produto de uma função gaussiana, simétrica em relação à origem e uma função cossenoidal.

## Aplicações:

- Segmentação de imagens
- Reconhecimento de faces
- Reconhecimento de assinaturas
- Melhoria e identificação de impressões digitais

# Transformada de Gabor

As transformadas tempo-freqüência surgiram como uma alternativa à análise de Fourier que incorporasse a possibilidade de realização de uma análise local do sinal.

**Limitações da transformada de Fourier:** não pode tratar eventos transitórios de maneira satisfatória,

$$U(f) = F\{u(t)\} = \int_{-\infty}^{\infty} u(t) e^{-i2\pi f t} dt$$

A transformada baseia-se na integração de todo o sinal para o cálculo de cada freqüência.

Eventos que venham a ocorrer no sinal em momentos distintos e mesmo bastante remotos contribuem de maneira global para a transformada.

## Filtros de Gabor

Pode-se formar o núcleo :

$$h_f(t) = p^*(t) \exp^{-i2\pi f t}$$

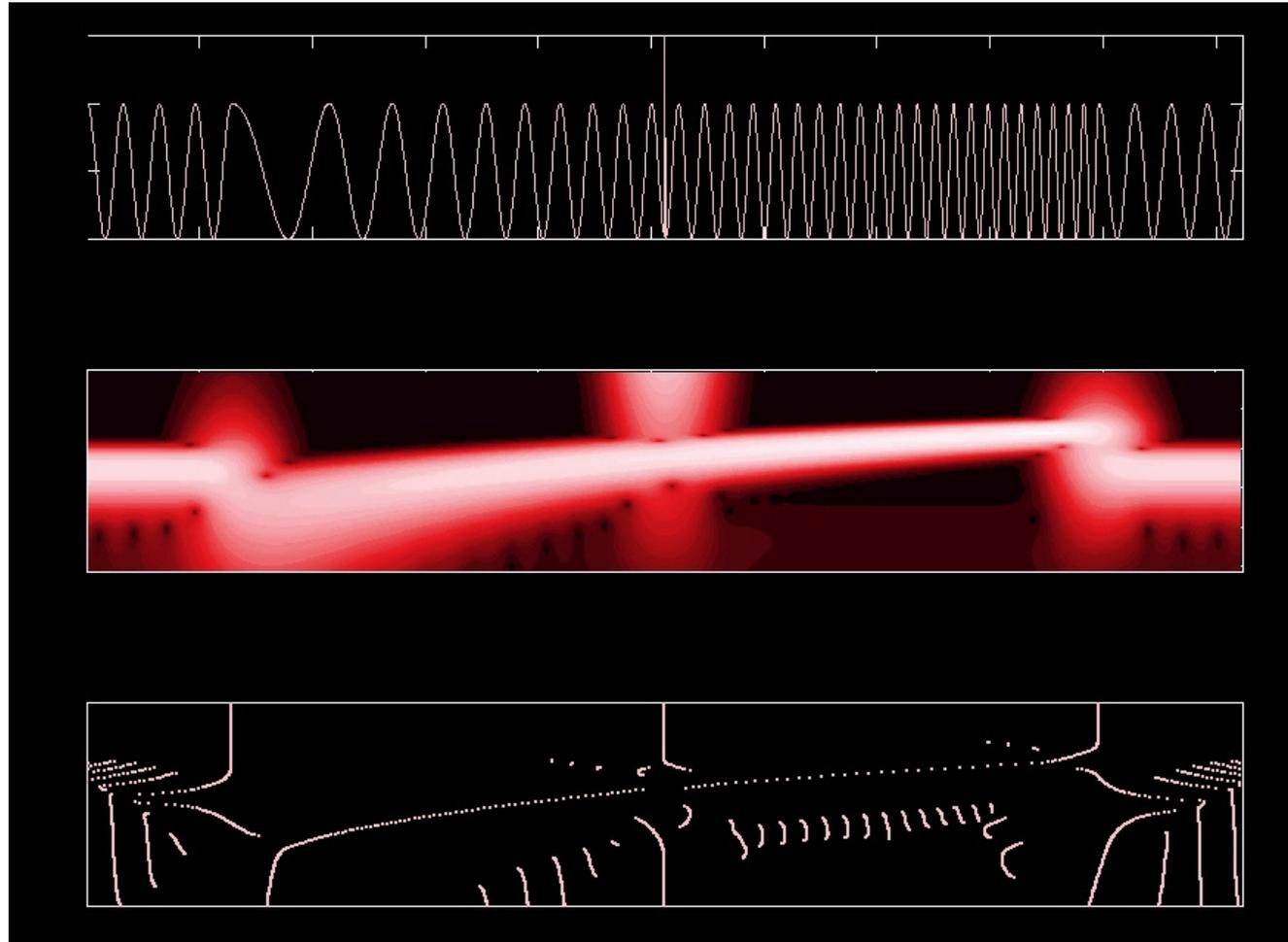
para cada valor fixo de  $f$ .

Janela  $p(t)$  modulada por uma exponencial complexa sintonizada na frequência  $f$ .

$$U(b, f) = \int_{-\infty}^{\infty} h_f(t-b) u(t) dt$$

A transformada  $U(b, f)$  pode ser vista como sendo o resultado da convolução de  $u(t)$  com um núcleo bem localizado em torno da frequência  $f$ .

# Filtro de Gabor



# Filtro de Gabor \*

- Filtro linear bi-dimensional e não variante ao deslocamento.
- Pode ser entendido como o produto de uma função Gaussiana, simétrica em relação à origem e uma função exponencial com parâmetro complexo (i.e. cossenoidal ou senoidal).

( lembre da Fórmula de Euler)

\* Dennis Gabor (Budapeste, 5 de Junho de 1900 — Londres, 9 de Fevereiro de 1979), foi Nobel de Física em 1971, e professor da BUTE (Budapest University of Technology and Economics)

# Filtro de Gabor

## Aplicações:

- Melhoria de imagens direcionais
- Segmentação de imagens
- Reconhecimento de faces
- Reconhecimento de assinaturas
- Identificação de impressões digitais

## Fórmula geral:

$$G(x, y, f, \theta, \sigma) = \exp\left[-\frac{1}{2}\left(\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right)\right] \cdot \exp[2.\pi.j.f(x_{\theta})]$$

Onde:

$$x_{\theta} = x \cos \theta + y \sin \theta \quad y_{\theta} = -x \sin \theta + y \cos \theta$$

e  $x, y$  são as coordenadas espaciais da imagem,  $j = \sqrt{-1}$

## Parâmetros:

- 1)  $f$  é a frequência da onda no plano senoidal;
- 2)  $\theta$  é a orientação do filtro na região  $x, y$  em consideração;
- 3)  $\sigma_x$  e  $\sigma_y$  é o desvio padrão da função Gaussiana ao longo dos eixos  $x$  e  $y$ , respectivamente.

( lembre da Fórmula de Euler)

Este filtro pode ser decomposto em componentes reais e imaginários:

$$G_{real}(x, y, f, \theta, \sigma) = \exp\left\{-\frac{1}{2} \cdot \left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right]\right\} \cdot \cos(2 \cdot \pi \cdot f \cdot x_{\theta}) \quad (5.25)$$

$$G_{imag}(x, y, f, \theta, \sigma) = \exp\left\{-\frac{1}{2} \cdot \left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right]\right\} \cdot \text{sen}(2 \cdot \pi \cdot f \cdot x_{\theta}) \quad (5.26)$$

sendo  $G_{complexo} = G_{real} + j G_{imag}$

$$G_{real}(x, y, f, \theta, \sigma) = W(x, y, \theta, \sigma) \cdot V(x, f)$$

então é possível visualizar a função gaussiana:

$$W(x, y, \theta, \sigma) = \exp\left\{-\frac{1}{2} \cdot \left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right]\right\}$$

e a função cossenoidal:

$$V(y, f) = \cos(2 \cdot \pi \cdot f \cdot x_{\theta})$$

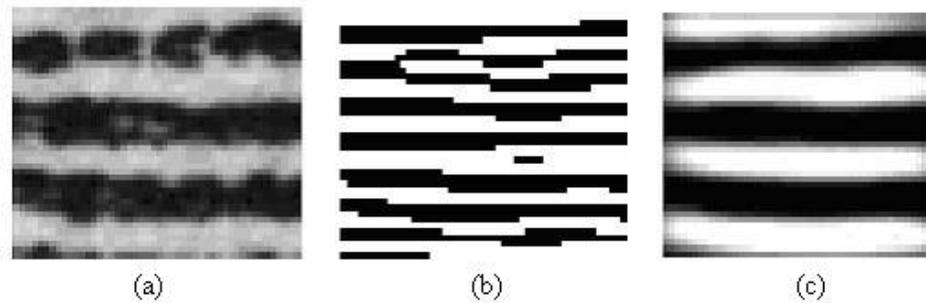
Como a função:

$$G_{real}(x, y, f, \theta, \sigma)$$

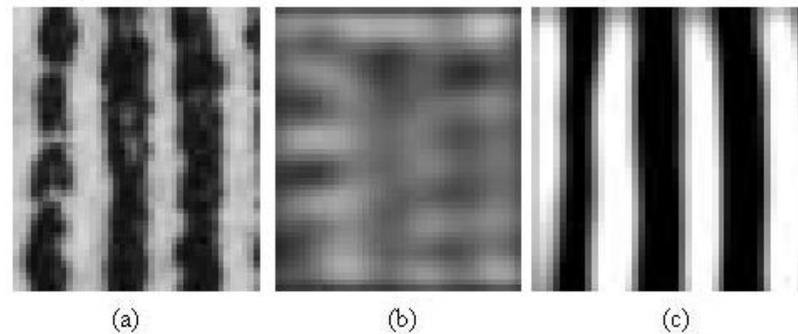
é obtida pelo produto de uma gaussiana por uma cossenoidal de frequência  $f$ , então pode ser representada no domínio da frequência e sua transformada de Fourier pode ser obtida pela convolução da transformada de Fourier dessas duas funções.

O resultado dessa convolução é um **filtro passa banda**, que realça as senóides com frequências em torno de  $f$ , suprimindo seus ruídos.

Uma das dificuldades para utilização do filtro de Gabor é a escolha ou obtenção de seus parâmetros.



Resultado de uma imagem de impressão digital filtrada por um filtro de Gabor com o **parâmetro  $f$  incorreto** e com **parâmetro  $f$  correto**.



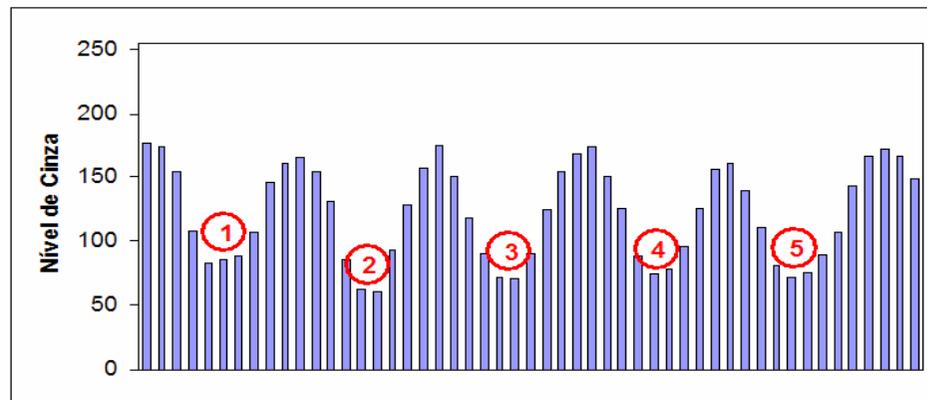
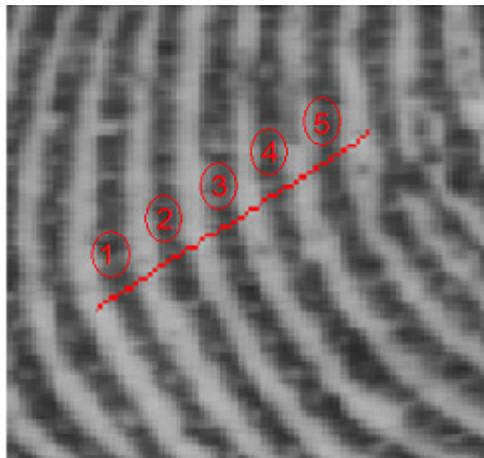
Resultado de uma imagem de impressão digital filtrada por um filtro de Gabor com o **parâmetro  $\theta$  incorreto** e com **parâmetro correto**.

# Número de linhas entre dois pontos

Original Image

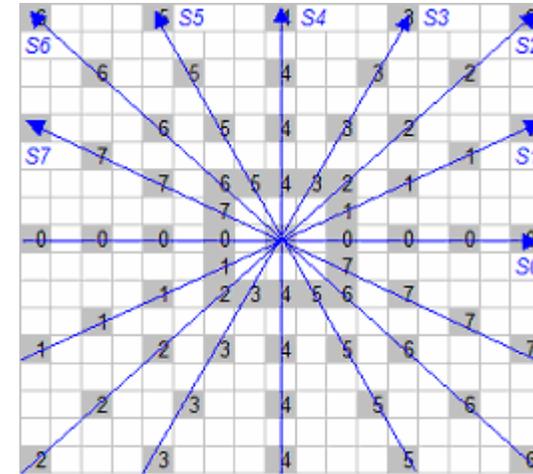


Posição Inicial:	<input type="text" value="149"/>	<input type="text" value="71"/>	<input type="button" value="Contar Linhas"/>	Número de Cristas: 5
Posição Final:	<input type="text" value="110"/>	<input type="text" value="127"/>	<input type="button" value="Limpar"/>	Número de Vales: 6

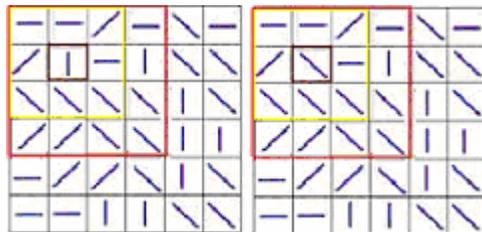




Original and directional image overlapped



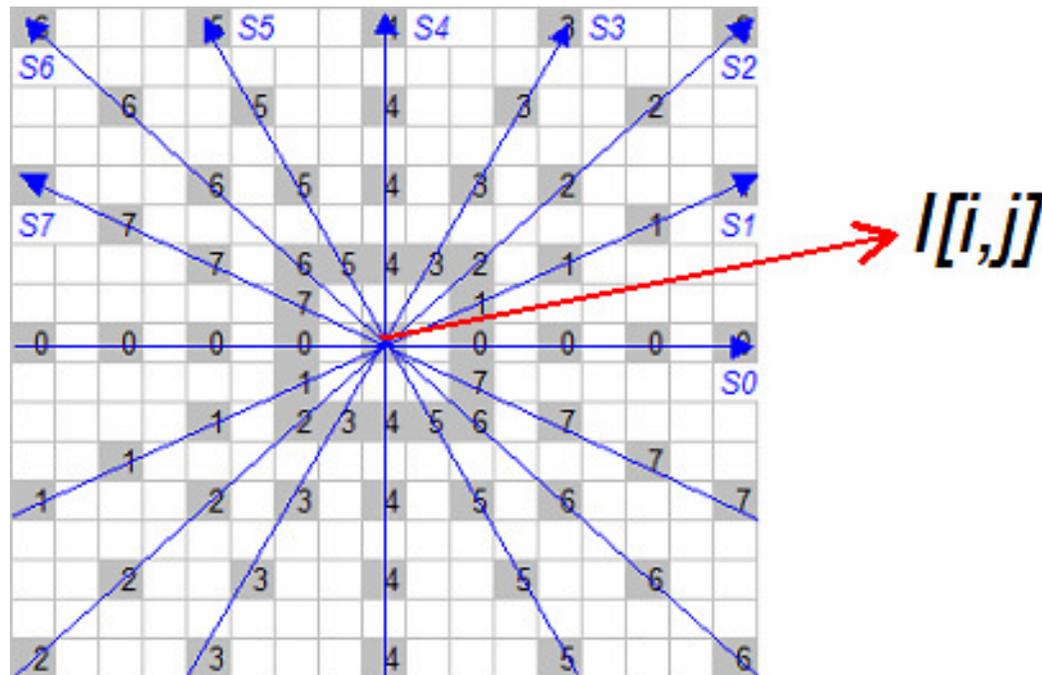
9x9 mask used to sum lines.



**O parâmetro  $\theta$  correto pode ser estimado por filtros direcionais [3]**

# Imagem direcional

- Estima a direção para cada bloco  $9 \times 9$  ou  $17 \times 17$  pixels, através de uma máscara com 8 direções.
- É calculada a soma das diferenças dos níveis de cinza entre o pixel  $[i,j]$  e os 8 ou 16 pixels vizinhos.



# Imagem direcional

- Para calcular a direção em um bloco  $9 \times 9$  ou  $17 \times 17$  pixels, a máscara é centrada no pixel  $[i, j]$ ;
- Os valores dos níveis de cinza nas 8 direções (identificados pelos números de 0 a 7) são somados para obter-se as somas  $S_0$  a  $S_7$

$$S_l = \sum_{K=-n}^n I(i - mK, j + m'K) - I(i, j)$$

- onde  $m = |\min(l, 2, 8 - l)|$
- $m' = 0$  SE  $l=4$ ,  $m' = -2$  SE  $l=7$
- Outros casos  $m' = |\min(4-l, 2)| \times (4-l)/|4-l|$
- Máscara  $9 \times 9$ ,  $n=2$  e  $K=1$
- Máscara  $17 \times 17$ ,  $n=4$  e  $K=2$

# Imagem direcional

---

$$S0 = \sum_{K=-2}^2 I(i, j+2K) - I(i, j)$$

$$S4 = \sum_{K=-2}^2 I(i+2K, j) - I(i, j)$$

$$S1 = \sum_{K=-2}^2 I(i+K, j-2K) - I(i, j)$$

$$S5 = \sum_{K=-2}^2 I(i+2K, j+K) - I(i, j)$$

$$S2 = \sum_{K=-2}^2 I(i+2K, j-2K) - I(i, j)$$

$$S6 = \sum_{K=-2}^2 I(i+2K, j+2K) - I(i, j)$$

$$S3 = \sum_{K=-2}^2 I(i+2K, j-K) - I(i, j)$$

$$S7 = \sum_{K=-2}^2 I(i+K, j+2K) - I(i, j)$$

# Imagem direcional

---

- $p, q$  são inteiros entre 0 e 7;
- $S_p = \text{mínimo } S_l (l=0\dots 7)$ ;
- $S_q = \text{máximo } S_l (l=0\dots 7)$ ;
- $S_l$  é a soma de  $S_0$  até  $S_7$ ;
- Se o pixel do centro tem o valor  $C$ , então a direção é determinada por:

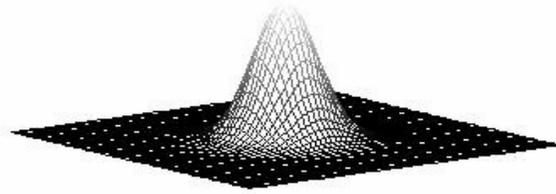
- $d = p, \text{ SE } (2nC + S_p + S_q) < \frac{3}{8} \sum_{i=0}^7 S_l$

- Outros casos  $d = q$ .

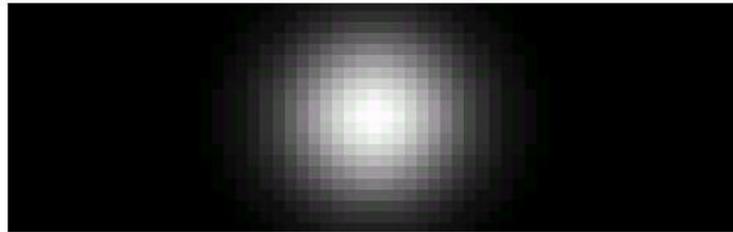
# Filtros de Gabor

- São conjunto de funções sinusoidais complexas, bidimensionais, moduladas por uma função Gaussiana também bidimensional e que
- apresentam certas propriedades extremamente úteis para fins de classificação de imagens.
- São criados bancos de filtros de Gabor para extrair as feições texturais presentes numa imagem.
- Adequados à análise textural, no sentido em que minimizam a incerteza conjunta, nos domínios espacial e no de freqüências espaciais, permitindo, desta forma, selecionar feições cujas características ficam determinadas por freqüências espaciais diretamente no domínio espacial, sem a necessidade de desenvolver operações no domínio de Fourier ou das freqüências.

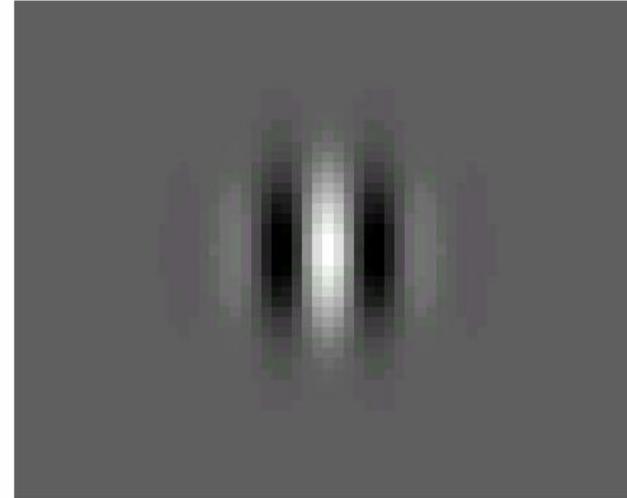
# Filtragem de Gabor em Imagens



Gaussiana



Gaussiana Modulada



Exponencial Complexa (parte real)

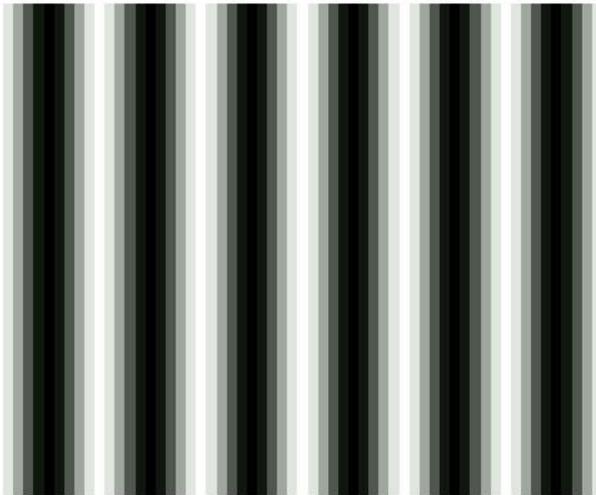


Imagem de Entrada



# Filtragem de Gabor em Imagens

Imagem Filtrada (parte real)

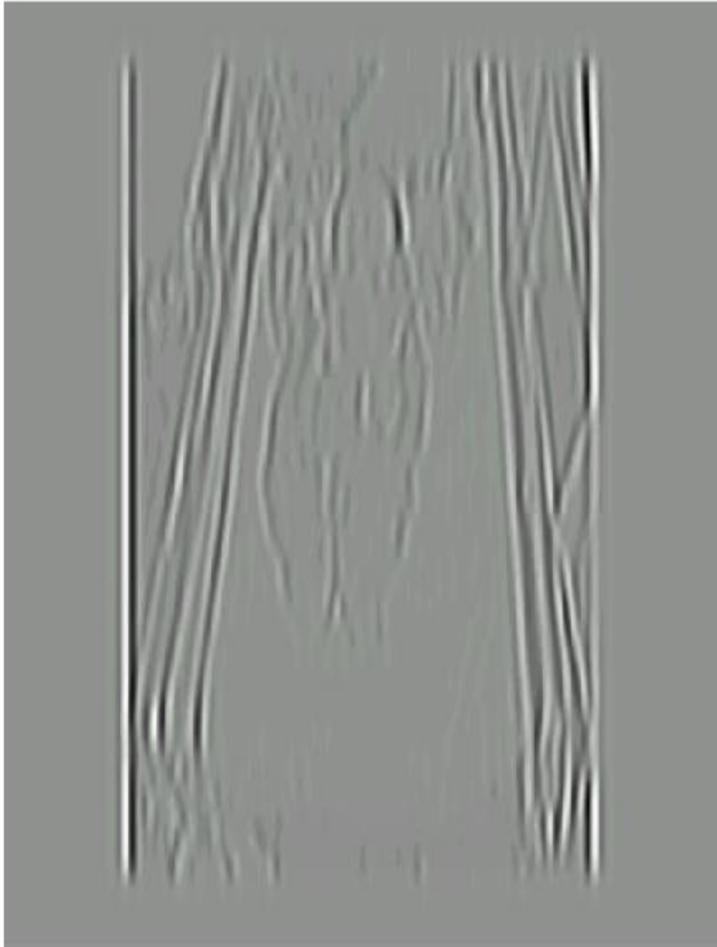
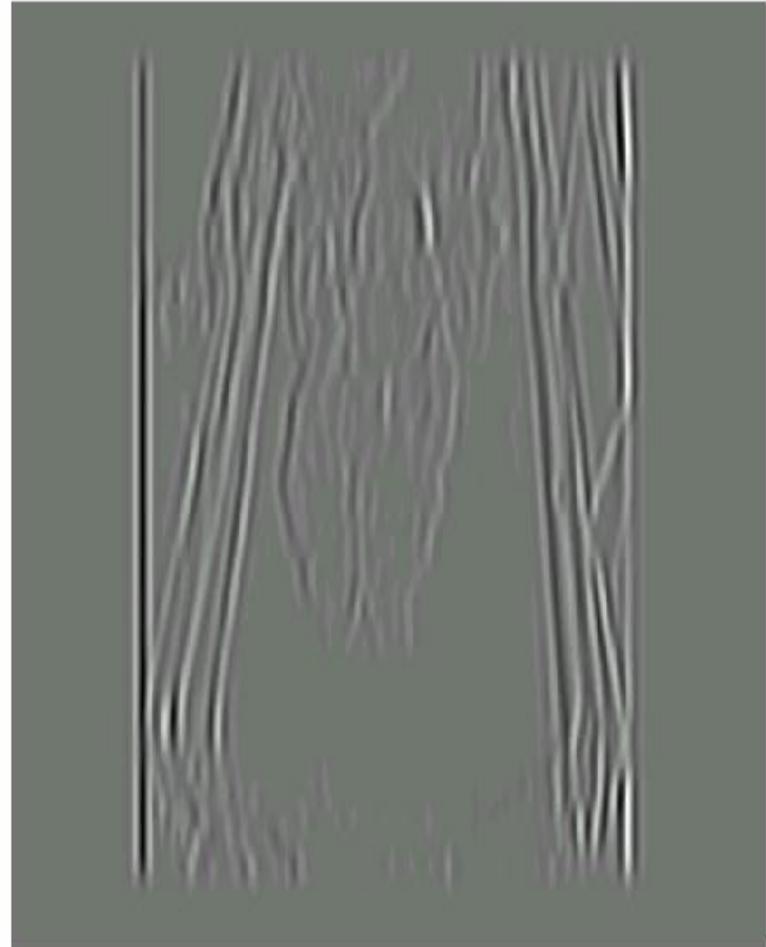


Imagem Filtrada (parte imaginária)



# Filtragem de Gabor em Imagens

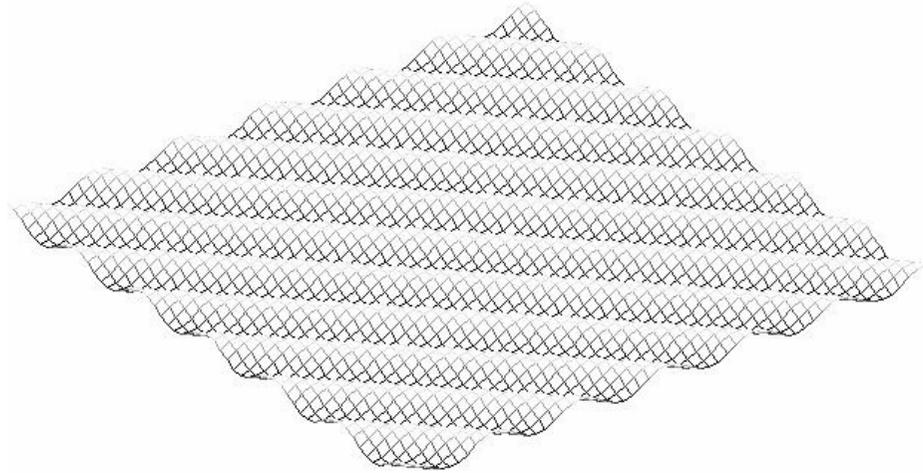


Imagem Filtrada (parte real)



# Filtro de Gabor

Exemplos de saída de amostra de filtros de Gabor com diferentes escalas e diferentes orientações:

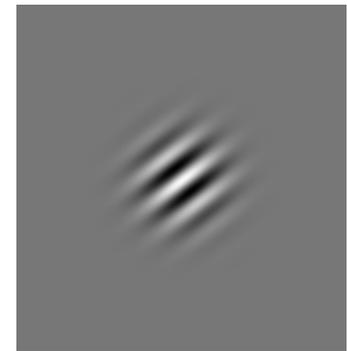
$0^{\circ}$ , escala 1

$36^{\circ}$ , escala 2

$72^{\circ}$ , escala 3

$108^{\circ}$ , escala 4

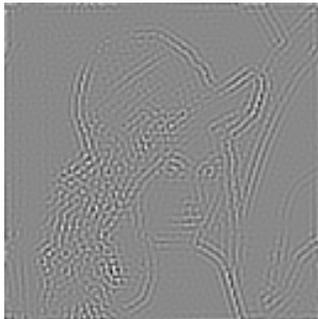
$144^{\circ}$ , escala 5



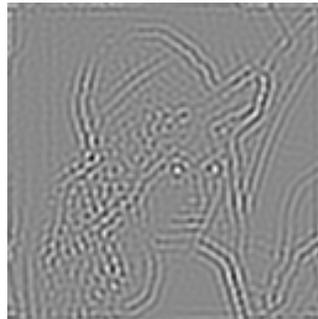
# Filtro de Gabor

Exemplo de geração de imagens em escalas diferentes:

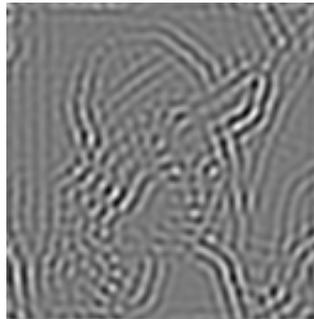
escala 1



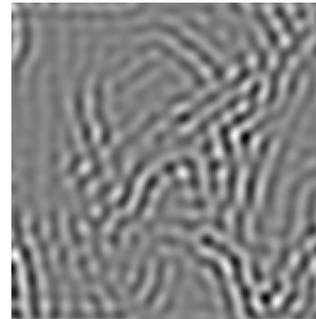
escala 2



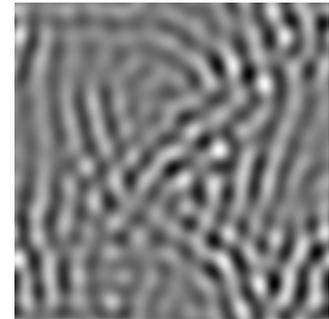
escala 3



escala 4

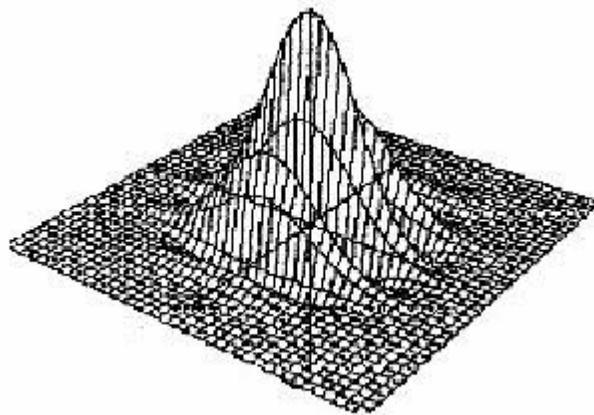


escala 5

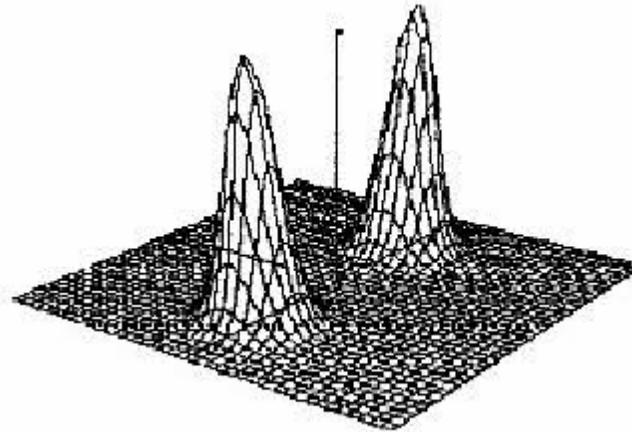


# Filtro de Gabor

Componente real do filtro de Gabor bidimensional no domínio espacial (a); resposta de freqüência do filtro definido em a (b).



(a)



(b)

# Filtro de Gabor

- A imagem processada por um filtro de Gabor é obtida através da convolução da imagem com a função de Gabor, que pode ser implementada tanto no domínio das frequências como no domínio espacial. Após o processo de convolução obtém-se uma imagem filtrada formada por valores imaginários. A imagem assim formada, por conter uma componente imaginária, não pode ser diretamente representada. Uma alternativa consiste na representação através de valores absolutos ou o quadrado deste, conhecido como energia. A energia de cada pixel é dada por:

$$f(x,y) = |I(x,y) * g(x,y)|^2$$

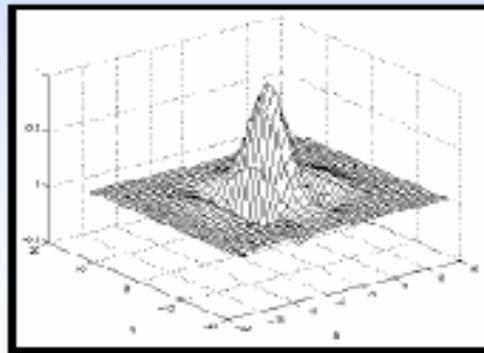
onde \* significa convolução,  $I(x,y)$  representa a imagem original e  $g(x,y)$  o filtro de Gabor, tomado com os parâmetros  $k_x, k_y, s_x, s_y$ .

# Filtro de Gabor

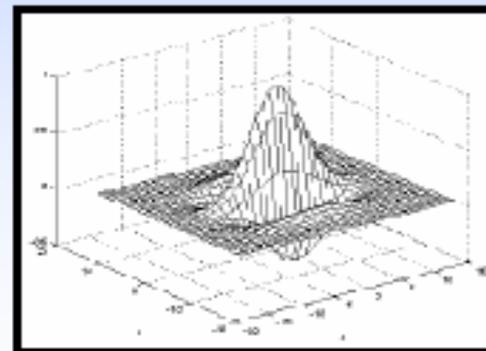
## Even-symmetric Gabor Filters

$$G_{\theta, f}(x, y) = \exp\left\{-\frac{1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\} \cos(2\pi f x'),$$
$$x' = x \sin\theta + y \cos\theta,$$
$$y' = x \cos\theta - y \sin\theta,$$

where  $f$  is the frequency of the sinusoidal plane wave at an angle  $\theta$  with the  $x$ -axis, and  $\delta_x$  and  $\delta_y$  are the standard deviations of the Gaussian envelope along the  $x$  and  $y$  axes, respectively.



0° orientation filter

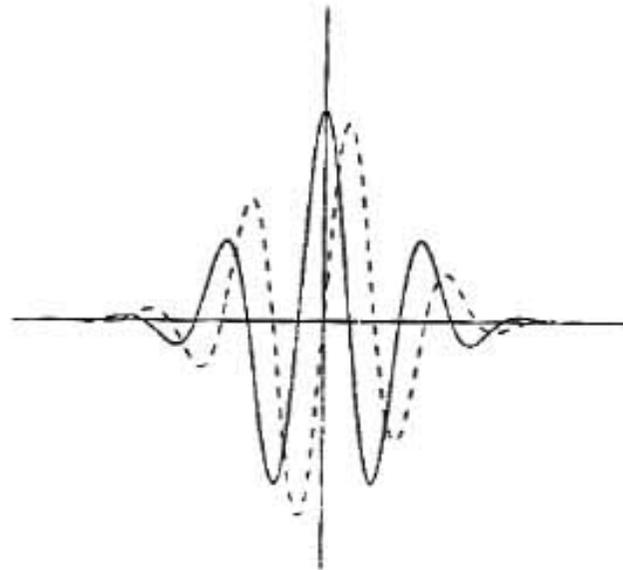


45° orientation filter

## Filtro de Gabor

As funções de *Gabor* utilizam três parâmetros: (i) localização espacial; (ii) orientação; (iii) frequência. São formadas por sinusóides envolvidas por uma função gaussiana. A expressão de Gabor é mostrada a seguir onde  $\alpha x$  é a constante da envolvente gaussiana que determina a largura da função no espaço,  $x_0$  é o centro da envolvente gaussiana que determina a localização da função no espaço,  $\beta$  é a frequência de oscilação e  $\gamma$  é a fase da oscilação

$$g_{1D}(x) = e^{-\alpha x^2(x-x_0)^2} \cdot [\cos(2\pi\beta(x-x_0) + \gamma) + i\text{sen}(2\pi\beta(x-x_0) + \gamma)]$$



# Filtro de Gabor

## Aplicação - Autenticação por textura orientada

usa a relação entre as rugas para definir a autenticação de impressões digitais. Localmente, as rugas são paralelas. Esta configuração só é perturbada pontualmente pela presença de minúcias, as quais interrompem o andamento paralelo das rugas e podem alterar ligeiramente a orientação local. Analisando a impressão digital obtendo informação local porém sem perder informação global, extraem-se características para medir a orientação, espaçamento entre rugas e a existência de perturbações originadas pelas minúcias em cada bloco. Estas características são obtidas através de um banco de filtros, onde cada filtro é previamente ajustado para medir características numa determinada direção.

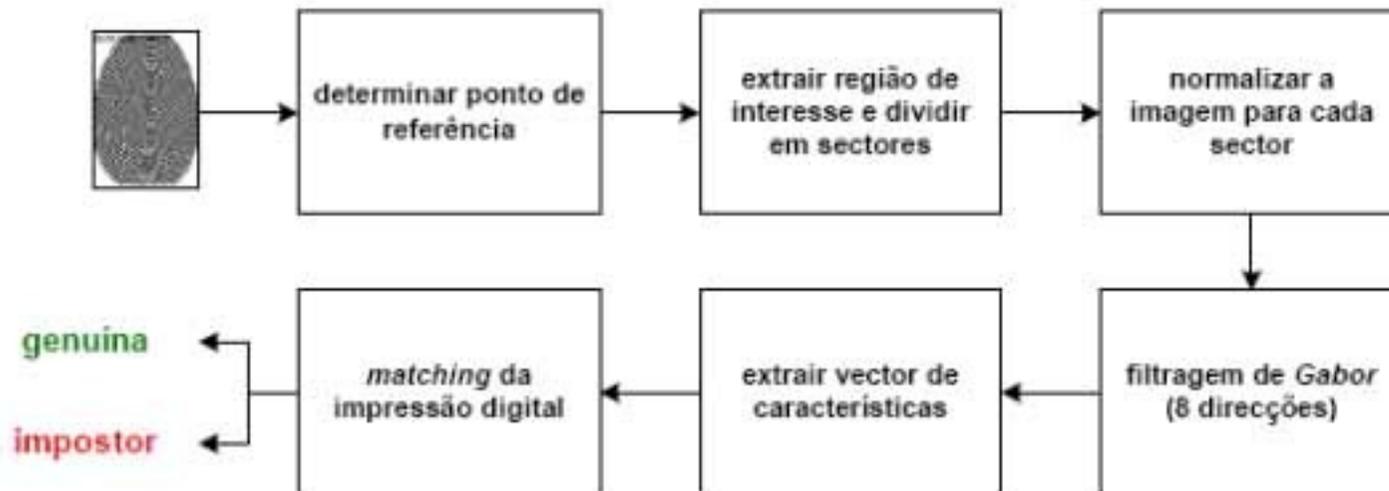
- Ao montar o banco de filtros, define-se que cada filtro mede a textura das rugas de acordo com uma direção, gerando um vetor de características que representa a orientação e o espaçamento delas (os valores deste dependem também do número de minúcias existentes em cada bloco). O filtro de *Gabor* tem esse papel porque realiza filtragem sintonizada e orientada segundo uma direção.

## Filtro de Gabor

- **Extração de características**

A extração de características acontece de seguinte maneira: efetuar filtragem com um conjunto de filtros de *Gabor*; dividir a imagem numa série de blocos; em cada bloco calcular uma medida estatística.

- A imagem abaixo mostra o processo necessário para que haja a autenticação por textura orientada:



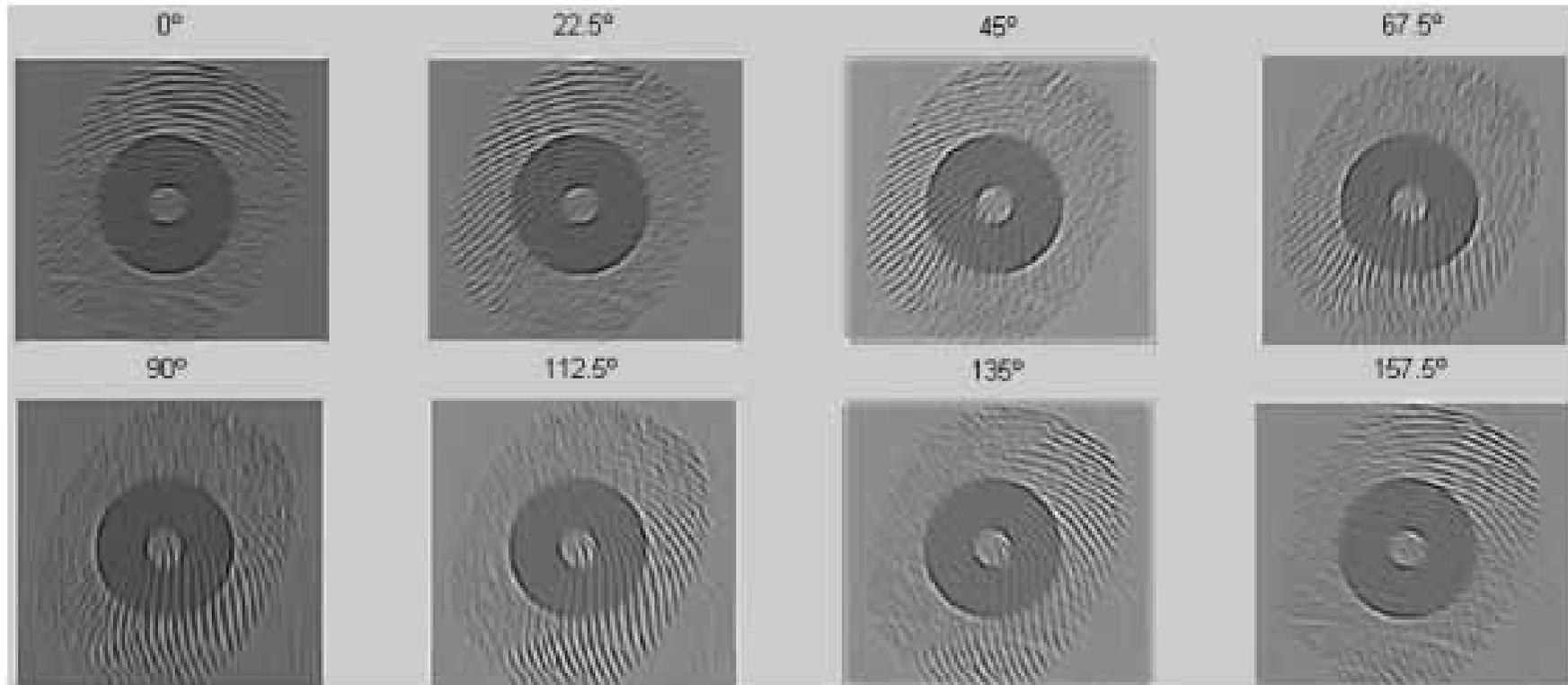
## Filtro de Gabor

- A normalização tem como objetivo corrigir diferenças de intensidade que possam existir entre zonas distintas (estas diferenças podem surgir devido a variações da pressão exercida pelo dedo no sensor, por exemplo)



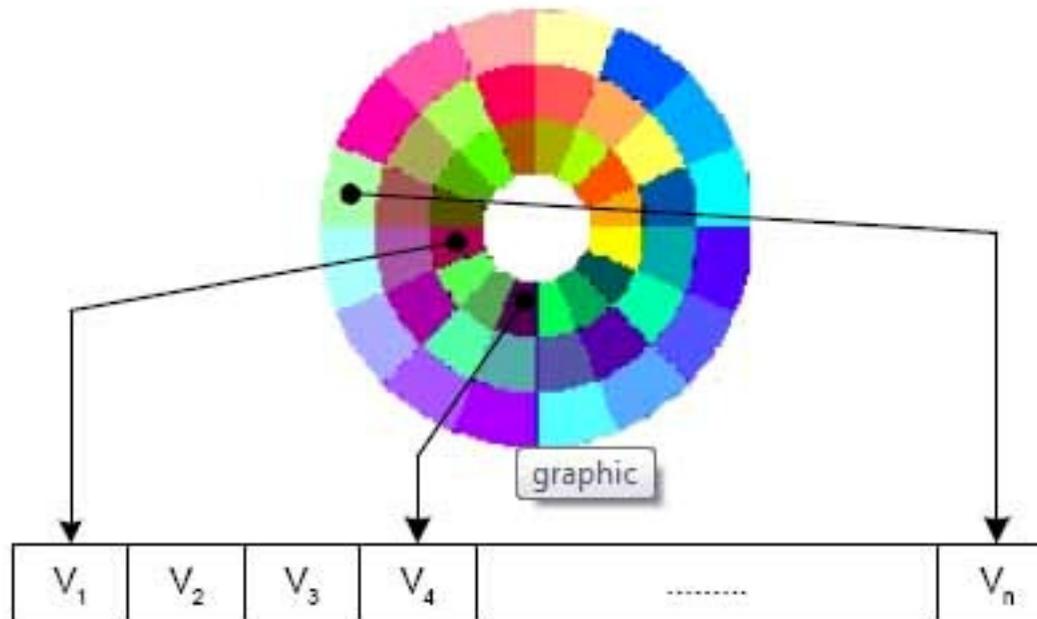
## Filtro de Gabor

- Depois do processo de normalização inicia-se a fase de filtragem da imagem, através de um banco de oito filtros de *Gabor* sintonizados à mesma frequência com diferentes orientações.



## Filtro de Gabor

- Após a operação de filtragem, é calculado o vetor de características da impressão digital. O conteúdo de cada entrada do vetor de características é dado pela média do desvio em relação à média do nível de intensidade em cada setor. Para cada setor: (1) obtém-se a média do nível de intensidade; (2) determina-se o desvio de todos os pontos pertencentes ao sector em relação à sua média; (3) obtém-se a média desse desvio.
- A Figura abaixo mostra como o vetor de características é formado.

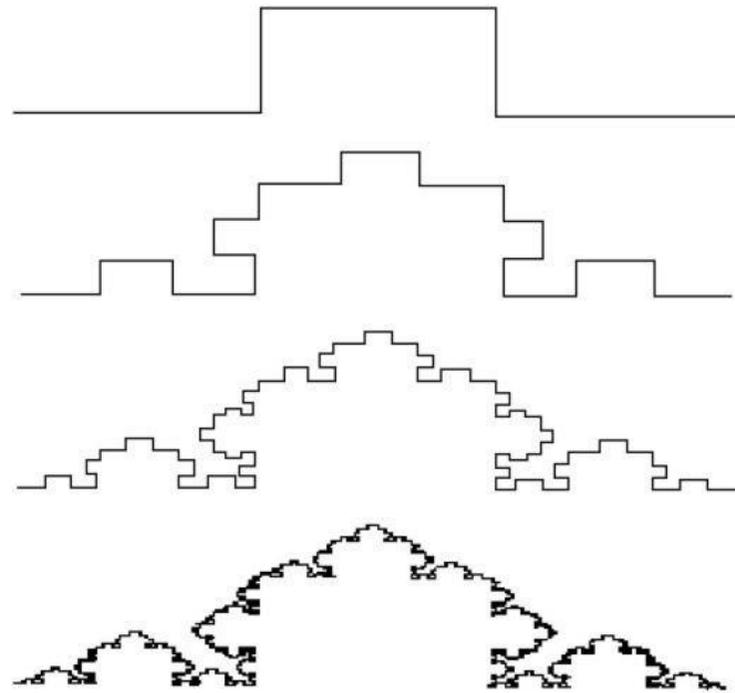


# Filtro de Gabor

## Vantagens:

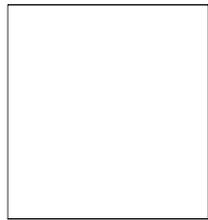
- 1- simulam o sistema visual humano,
- 2- são dependentes da direção,
- 3- são localizáveis tanto no espaço como na frequência,
- 4- atendem ao critério de igualdade no princípio da incerteza,
- 5- usado com uma orientação local correta e frequência local correta podem remover ruídos enquanto preservam os sulcos, arestas (ridges) reais.

# Transformada Fractal

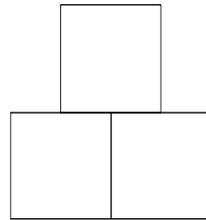


Curva de Koch

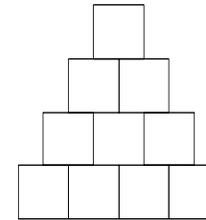
# Comprimindo Imagens com a Geometria Fractal



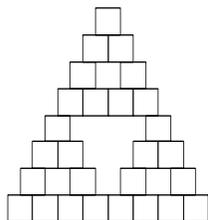
1ª iteração



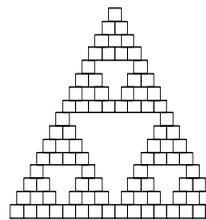
2ª iteração



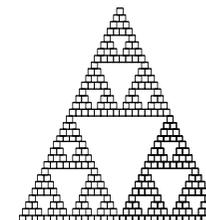
3ª iteração



4ª iteração



5ª iteração



6ª iteração

**Triângulo de Sierpinski**

DF do Triângulo de Sierpinski:

$$DF = \log 3 / \log 2 = 1,5849625007211561814537389439478\dots \quad (8.39)$$

# Teorema do Mapeamento de Contração e Teorema da Colagem

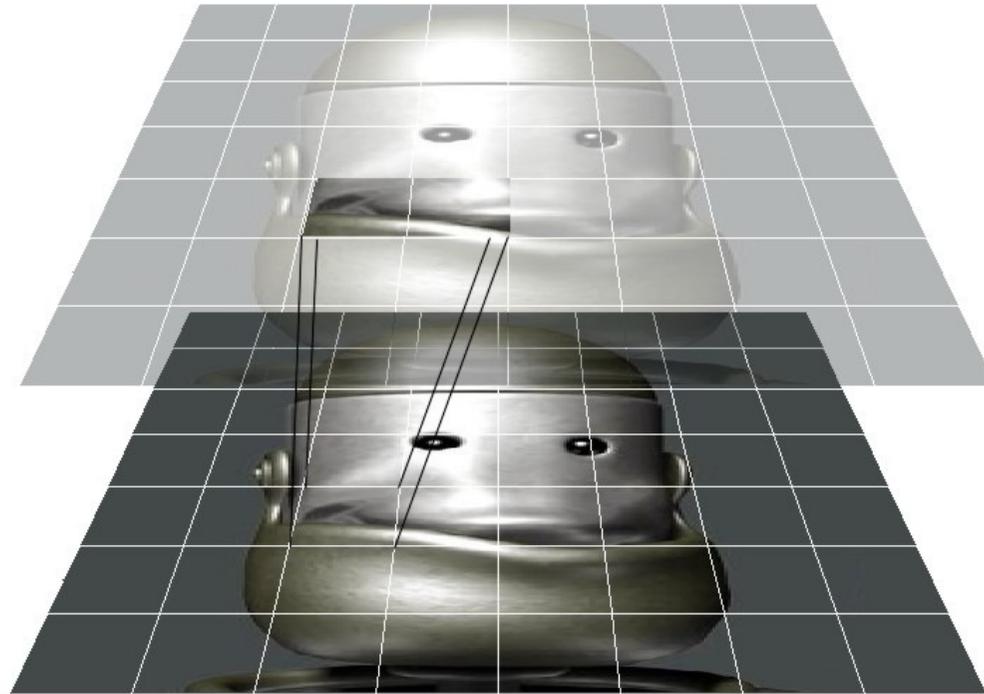
Partindo de qualquer conjunto inicial (imagem inicial) será encontrado o mesmo atrator para um determinado Sistema de Funções Iterativas (SFI).

**Pelo Teorema da Colagem** o código IFS de uma imagem pode ser obtido cobrindo-se essa imagem com cópias reduzidas dela mesma, e determinando as transformações que levam a imagem original em suas cópias.

# Determinando o SFI de imagens automaticamente

**Sistemas de Funções de Iteração Particionados SFIP** que buscam a auto-similaridade entre partes maiores e partes menores da imagem.

- As imagens são vistas como uma colagem de partes auto similares que podem ser mapeadas entre si.
- Como partes menores tomam-se blocos quadrados de  $n \times n$  *pixels*, chamados de blocos imagem ou molde, e como partes maiores blocos com o dobro das dimensões do menor ( $2n \times 2n$ ), chamados de blocos domínio.

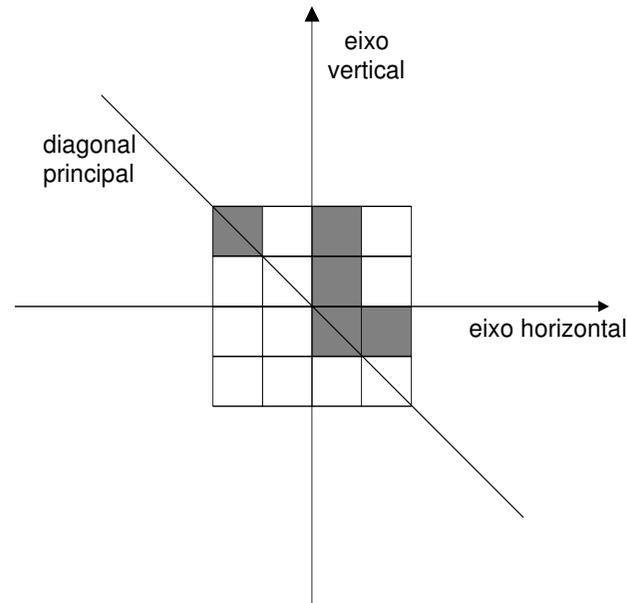


### **Formação de um par de blocos domínio-molde ótimo.**

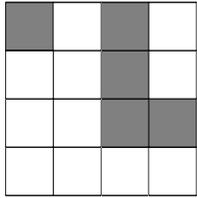
A imagem deve ser entendida como um objeto tridimensional, o valor da tonalidade de um pixel ( $z$ ) é tratado como sendo a terceira dimensão espacial. Os blocos tornam-se cubos.

# Considerações sobre a simetria do bloco domínio

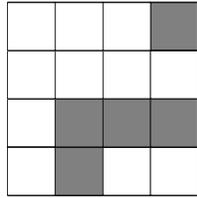
Para cada bloco domínio tem-se oito simetrias que serão testadas em relação ao bloco molde corrente.



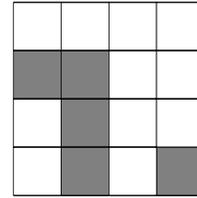
**Eixos de Reflexão considerados nas operações de simetria.**



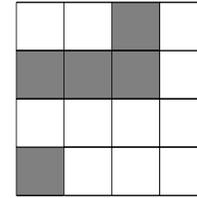
Identidade  
(0,0,0)



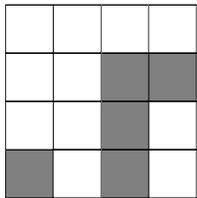
Rotacionada de 90°  
(1,1,0)



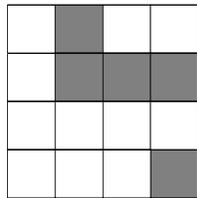
Rotacionada de 180°  
(0,1,1)



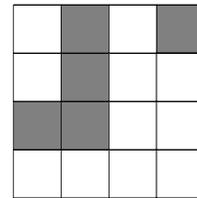
Rotacionada de 270°  
(1,0,1)



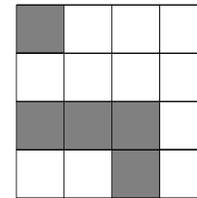
Ref. horizontal  
imagem identidade  
(0,0,1)



Ref. horizontal  
da imagem rotacionada  
de 90° (1,1,1)



Ref. Vertical da  
imagem identidade  
(0,1,0)

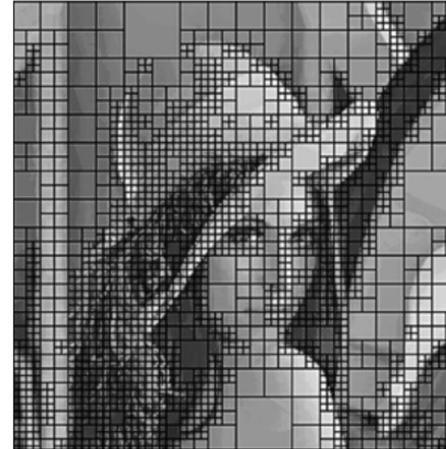


Ref. na diagonal da  
principal (i,j = j,i)  
(1,0,0)

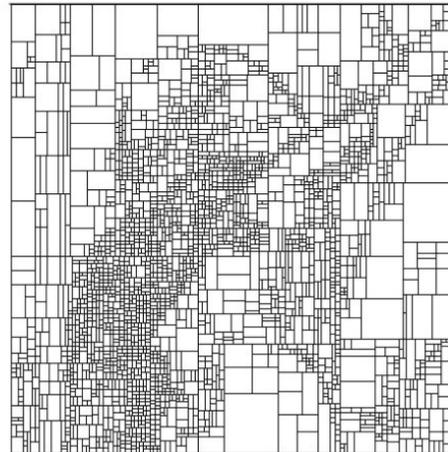
Bloco molde e suas 8 possíveis simetrias.



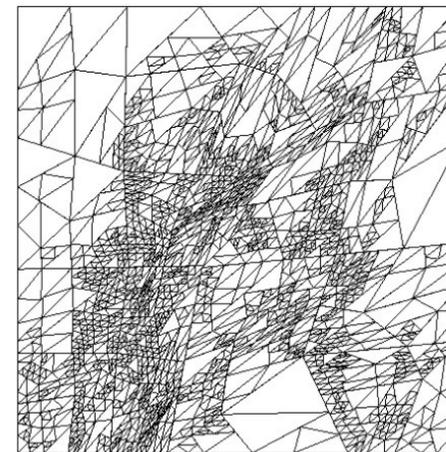
A- ORIGINAL



B- QUADTREE



C- HV

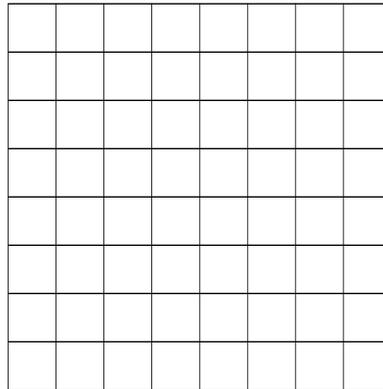


D- TRIÂNGULO

**Diferentes formas de *particionamento* de uma imagem.**

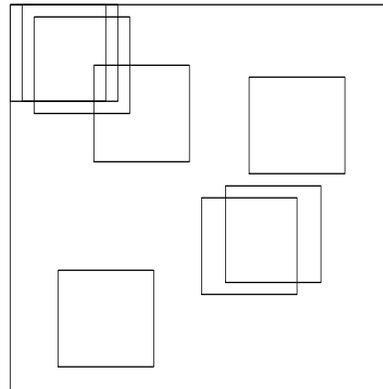
# **Etapas da Compressão Fractal Automática**

**Primeira etapa: Definição dos blocos molde**



**Malha de blocos molde de uma imagem.**

## Segunda etapa: Definição dos blocos domínio



Alguns blocos domínio da figura anterior.

## Terceira etapa: Redução dos blocos domínio

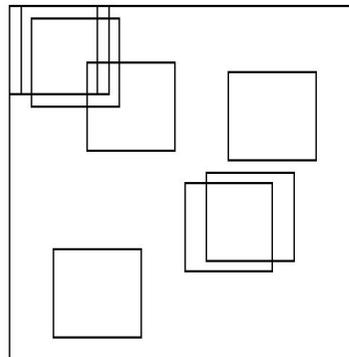
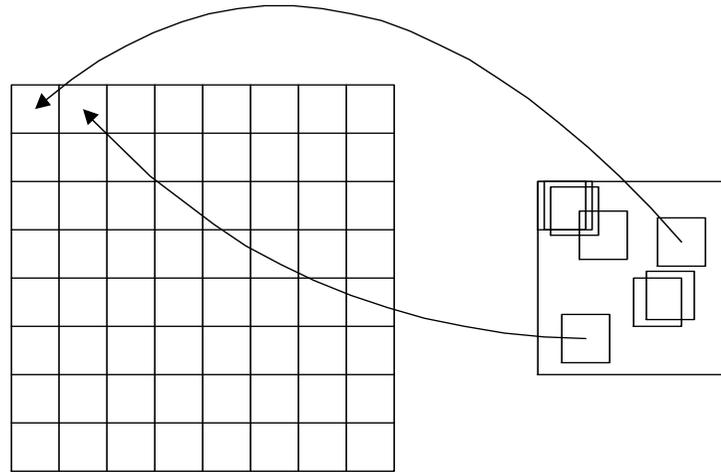


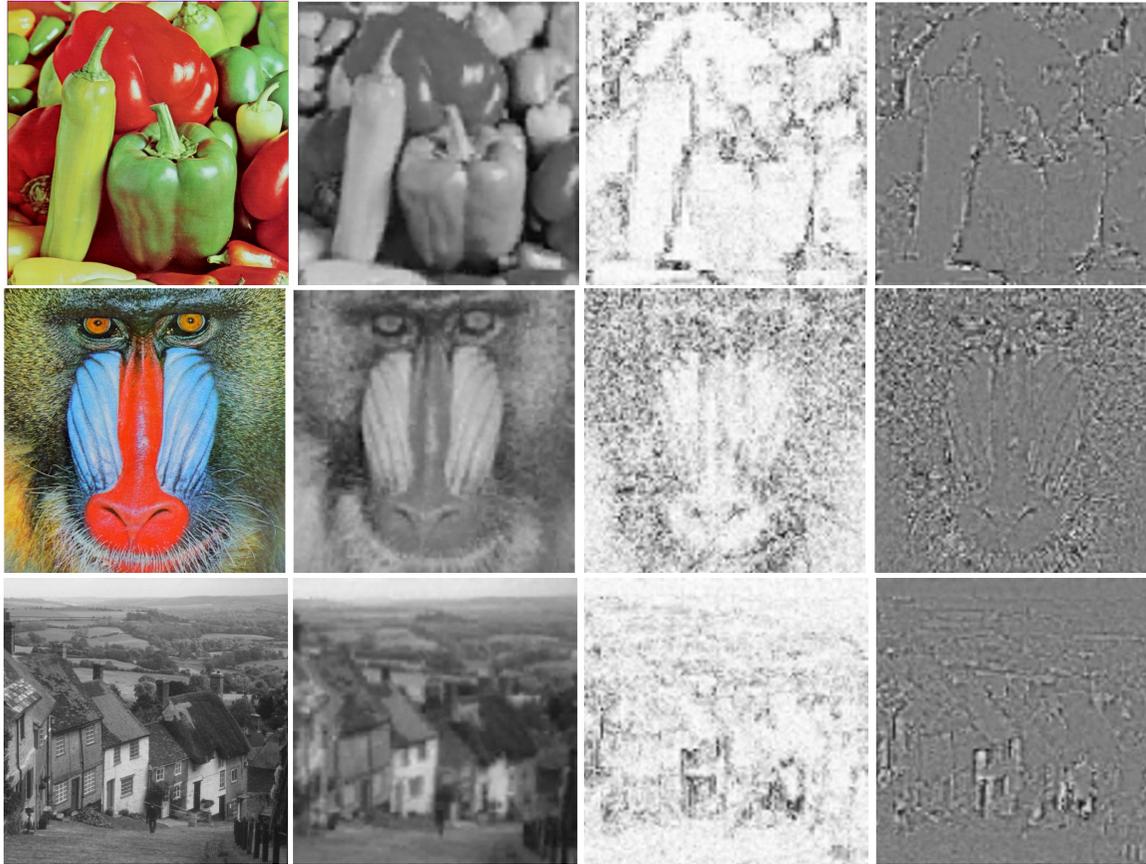
Imagem reduzida por um fator igual a  $\frac{1}{2}$ , de onde sairão os blocos domínio reduzidos.

# Quarta etapa: Busca do melhor par bloco domínio reduzido- molde



**Formação dos pares ótimos entre os blocos domínio e molde.**

## Quinta etapa:



**Resultados obtidos da compressão Fractal usando a técnica de Sub Busca Local com  $\Delta = 8$  e blocos molde de  $4 \times 4$  pixels.**



**Resultados obtidos da compressão Fractal usando a técnica de Sub Busca Local com  $\Delta = 8$  e blocos molde de  $4 \times 4$  pixels.**

**Resultados obtidos com o método de Sub Busca Local com  $\Delta = 8$  e blocos molde de 4x4 *pixels* para as imagens da figura 8.1 e 8.20.**

<b>Nome da imagem</b>	<b><math>e_{rms}</math></b>	<b><math>SNR_{rms}</math></b>	<b>PSNR (dB)</b>
<i>Lena</i>	9, 9057	10, 3072	28, 431
<i>Peppers</i>	10, 6933	8, 5312	27, 5485
<i>Mandrill</i>	15, 4048	7, 6832	24, 3777
<i>Goldhill</i>	9, 2310	10, 0295	28, 8259
<i>Woman1</i>	13, 5039	8, 5180	25, 546
<i>Milk</i>	12, 0226	6, 7880	26, 5309

## Transformada de Fourier-Mellin

- O domínio Fourier-Mellin é uma combinação do domínio de Fourier com um mapeamento log-polar.
- Na tentativa de solucionar o problema da sensibilidade as distorções geométricas aplicadas a imagem, alguns métodos buscaram domínios diversos do espacial, que seja domínios invariantes a translações, rotações, e escalamento *RST* da imagem.

## O domínio Fourier-Mellin

- A seguir serão apresentadas algumas propriedades do domínio de Fourier, que
- serão interessantes para a construção do domínio invariante  $RST$  de Fourier-Mellin.

## O domínio Fourier-Mellin

# Translação

- Um deslocamento espacial equivale a um deslocamento linear na fase.

$$f(x_1 + a, x_2 + b) \Leftrightarrow F(k_1, k_2)e^{-j(ak_1 + bk_2)}$$

# O domínio Fourier-Mellin

## Escalamento

- Um escalamento no espaço equivale a um escalamento inverso na frequência.

$$f(\rho x_1, \rho x_2) \Leftrightarrow \frac{1}{\rho} F\left(\frac{k_1}{\rho}, \frac{k_2}{\rho}\right)$$

# O domínio Fourier-Mellin

## Rotação

- Uma rotação espacial de um ângulo da imagem equivale a uma rotação da representação frequencial pelo mesmo ângulo.

$$f(x_1 \cos \theta - x_2 \sin \theta, x_1 \sin \theta + x_2 \cos \theta) \Leftrightarrow F(k_1 \cos \theta - k_2 \sin \theta, k_1 \sin \theta + k_2 \cos \theta)$$

## O domínio Fourier-Mellin (cont)

- Observando as propriedades apresentadas, fica claro que a magnitude, no domínio da frequência, não é afetada por um deslocamento espacial.
- Somente a fase sofre um deslocamento. Conclui-se, então, que a **magnitude é**
- **invariante a translação** circular.

# O domínio Fourier-Mellin

## Invariância ao Escalamento e a Rotação

- Utilizando um mapeamento log-polar, a invariância a translação apresentada pode ser convertida em invariância a rotação e ao escalamento. O mapeamento possui a seguinte forma:

- $(x, y) \in \mathcal{R}^2$

$$\begin{cases} x = e^\mu \cos \theta \\ y = e^\mu \sin \theta \end{cases}$$

- Onde

- $\mu \in \mathcal{R}$
  - $0 \leq \theta < 2\pi$

Para cada ponto  $(x, y)$  existe um único ponto  $(\mu, \theta)$  que correspondente. E neste novo sistema o escalamento e a rotação foram transformados em uma simples translação. Assim:

$$(\rho x, \rho y) \Leftrightarrow (\mu + \log \rho, \theta)$$

$$(x \cos(\theta + \delta) - y \sin(\theta + \delta), x \sin(\theta + \delta) + y \cos(\theta + \delta)) \Leftrightarrow (\mu, \theta + \delta)$$

Como já notado anteriormente, a magnitude da transformada de Fourier é invariante a translação.

- Desta forma, basta que seja novamente aplicada a transformada de Fourier a este mapeamento para que seja encontrado um domínio invariante a rotação, escalamento, e translação (invariante *RST*).
- Este é o domínio de Fourier-Mellin.
- Sendo assim, a magnitude da transformada de Fourier-Mellin é invariante *RST*.

A transformada de **Fourier-Mellin** é definida da seguinte forma:

$$F_M(k_1, k_2) = \int_{-\infty}^{\infty} \int_0^{2\pi} f(e^\mu \cos \theta, e^\mu \sin \theta) e^{i(k_1 \mu + k_2 \theta)} d\mu d\theta$$