

Análise de Imagens - 2020

Tema 3

Outras Operações em Imagens



Lembrando as operações em uma imagem

podem ser realizadas:

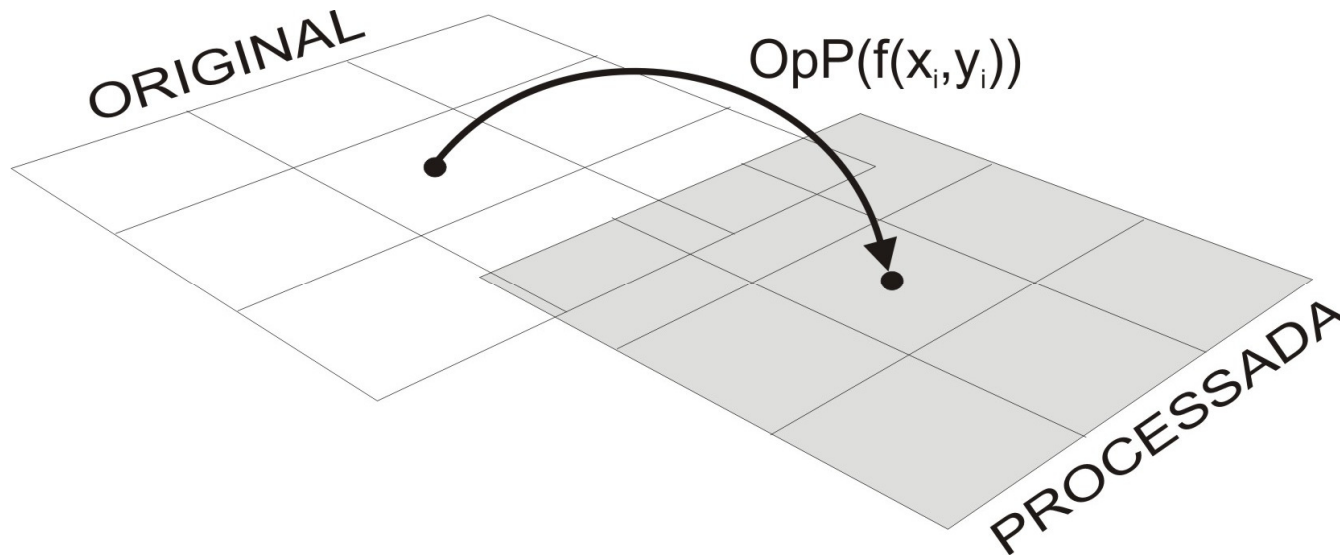
- pontualmente nos *pixels*;
- em partes da imagem
 - fixas ou
 - dependendo de algum contexto; e,
- em toda a imagem .

Vamos considerar outras operações em uma imagem monocromática ou uma banda de imagem multibanda

- 1. Operações Pontuais**
- 2. Operações Locais**
- 3. Operações Globais**
- 4. Transformações Geométricas**

1. Operações Pontuais

O *pixel*, na posição (x_i, y_i) , da imagem resultante depende apenas do *pixel* na imagem original.



Esquema de operações pontuais em imagens.

Operações pontuais em imagens

1. Operações pontuais baseadas na curva de tom

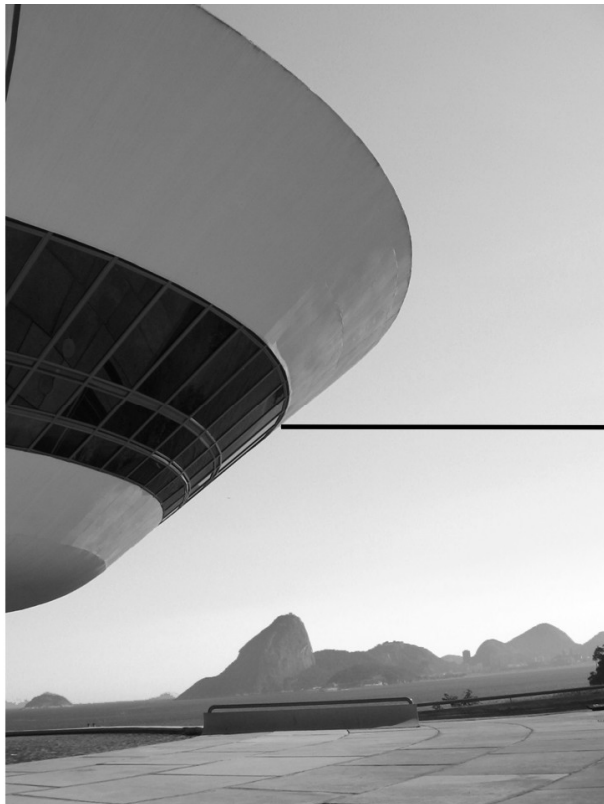
2. Operações baseadas em histograma

3. Técnicas Baseadas no Histograma de Imagens Coloridas

O histograma de uma imagem:

- Que foi a que mais trabalhamos ate agora segundo essa classificação é de qual tipo?

Um *pixel* é caracterizado pelo valor de **intensidade** de cor e pela sua **localização** na imagem.



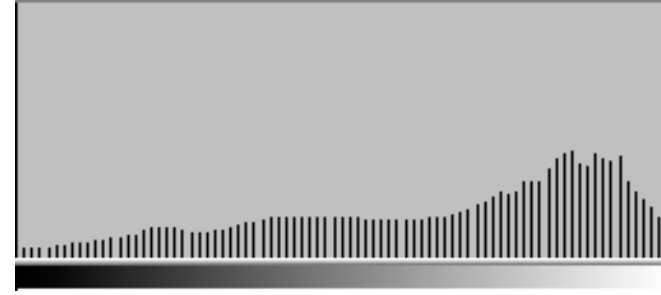
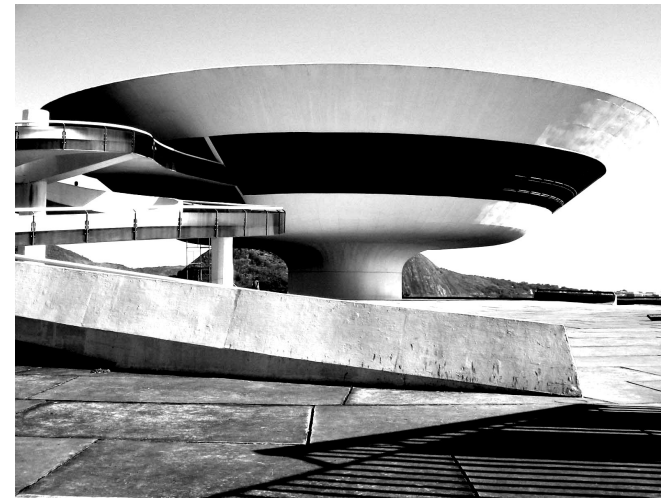
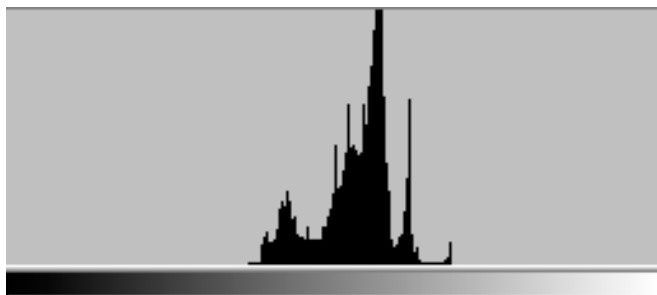
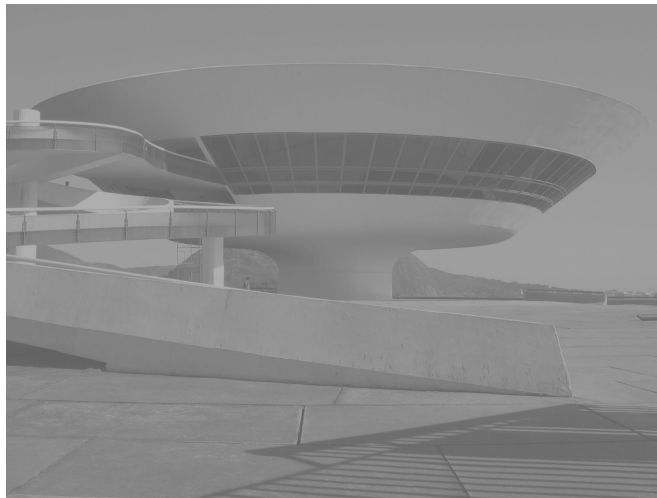
47	52	64	132	153
51	58	121	149	142
49	99	143	144	164
94	135	161	170	199
138	165	180	212	213

O histograma conta quantos pixels tem em cada intensidade de uma região da imagem ou toda ela.

Quando se altera um tom,
Se PERDE A
REFERENCIA DE ONDE
ESTA O TOM e se altera
todos os tons iguais na
imagem !

Logo é uma operação ?

GLOBAL .

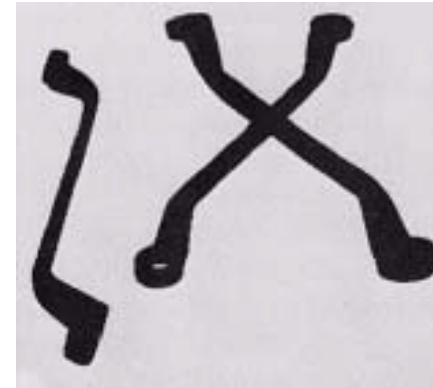
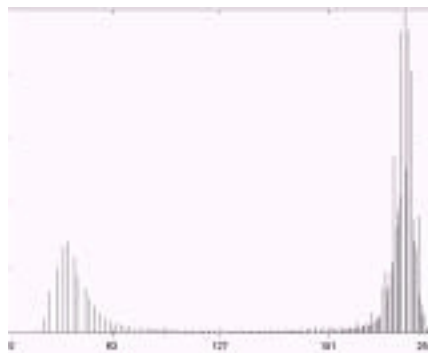
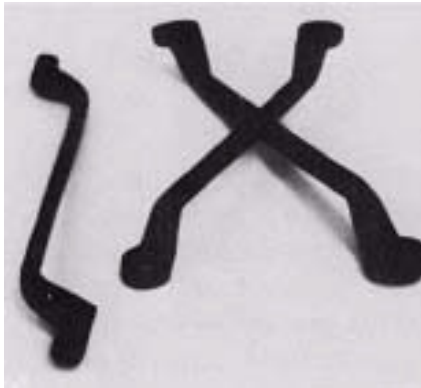


Imagens em tons de cinza e seus respectivos histogramas: com pouco (direita) e bom **contraste** (esquerda) depois de uma TRANSFORMAÇÃO LINEAR DO HISTOGRAMA PARA IR DO 0 AO 255.

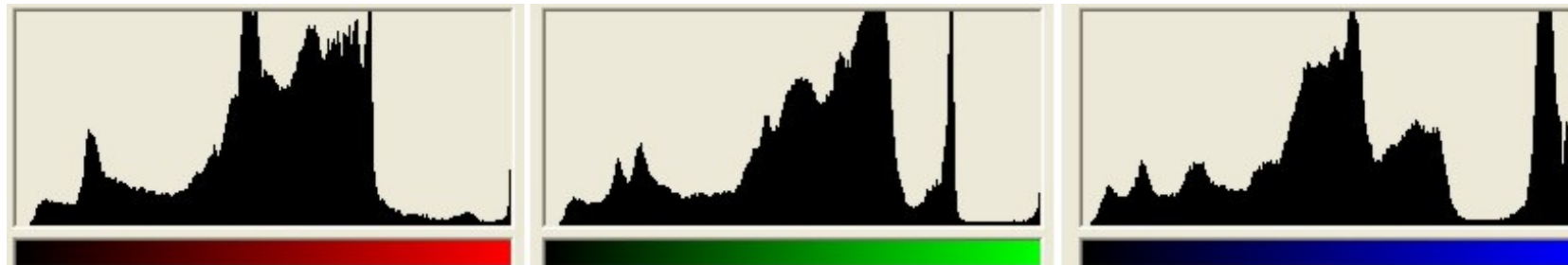
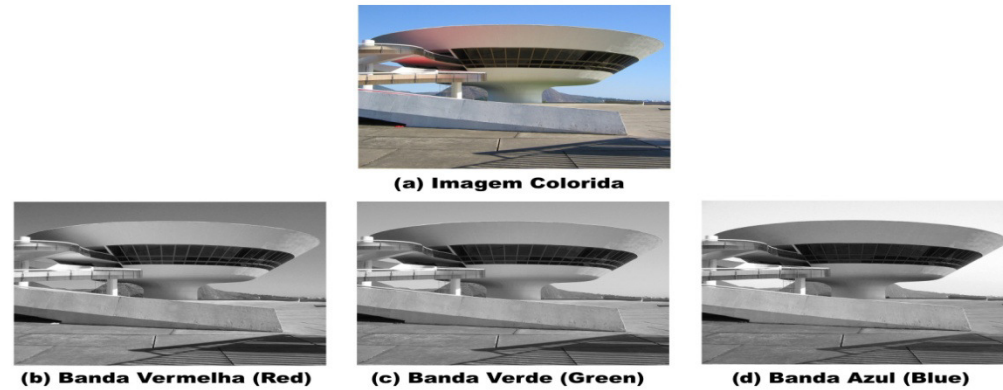
Limiarização Global

Método de Otsu
おつ

T = 90



Histogramas das **bandas** de uma imagem colorida



Histogramas da Imagem colorida separado por canal R,G B.

Podem ser unidos

- Em um gráfico 3D fazendo um histograma 3d de cores e depois quantificando as cores mais populosas.

Leia mais em:

[3D Color Inspector/Color Histogram](#), by Kai Uwe Barthel. (Free [Java applet](#).)
[Stanford Student Project on Image Based Retrieval](#) - more in depth look at
equations/application

[MATLAB/Octave code for plotting Color Histograms and Color Clouds](#) - The
source code can be ported to other languages

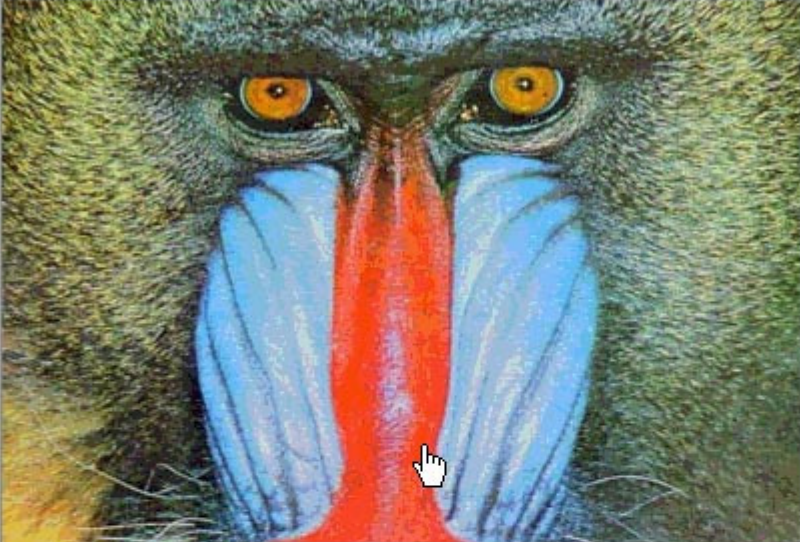
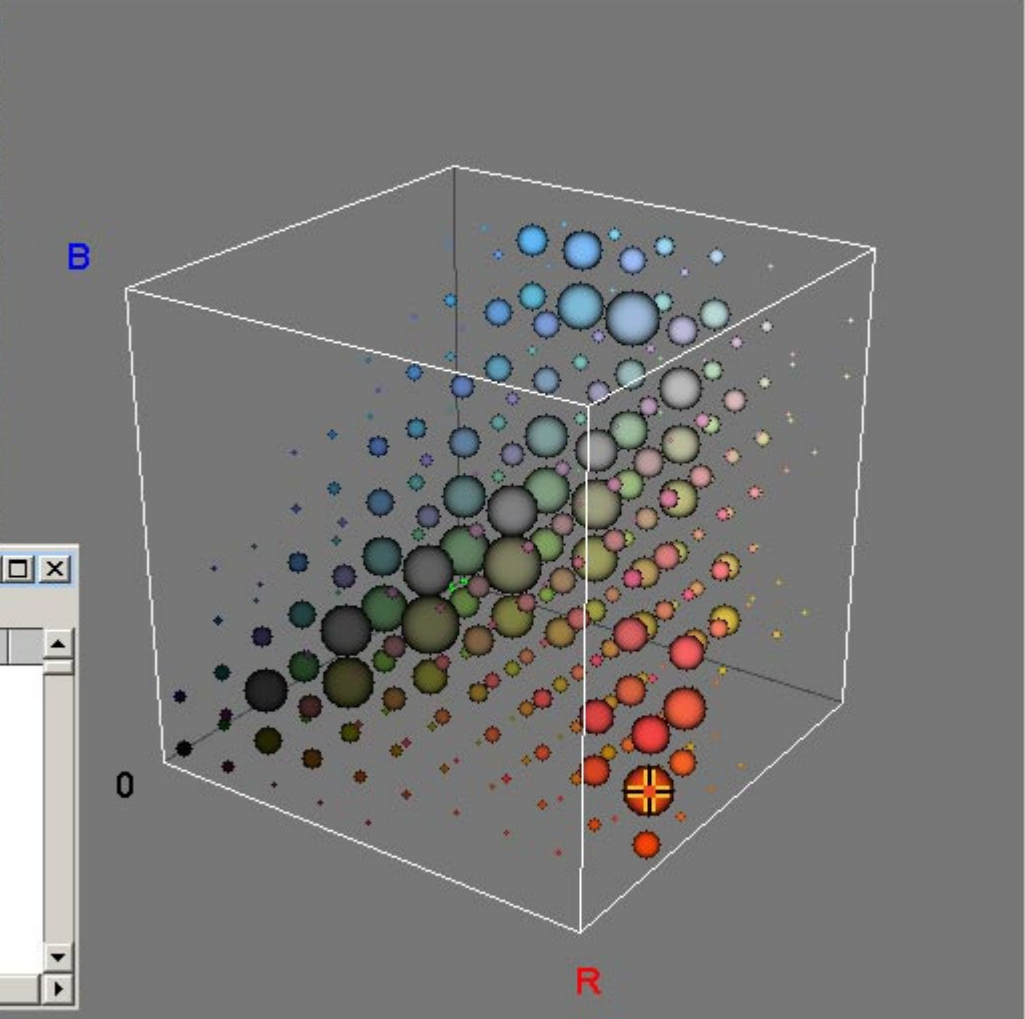
<https://imagej.nih.gov/ij/plugins/color-inspector.html>

Color Inspector 3D (v2.0) /images/baboon400.jpg

File Options Help

Color Space: RGB Display: Histogram Number of Color Cells LUT

160000 Pixels, 271 Colors Position x: 210 y: 222 Color: RGB(248, 68, 38) Frequency: 3868 (2.4%)

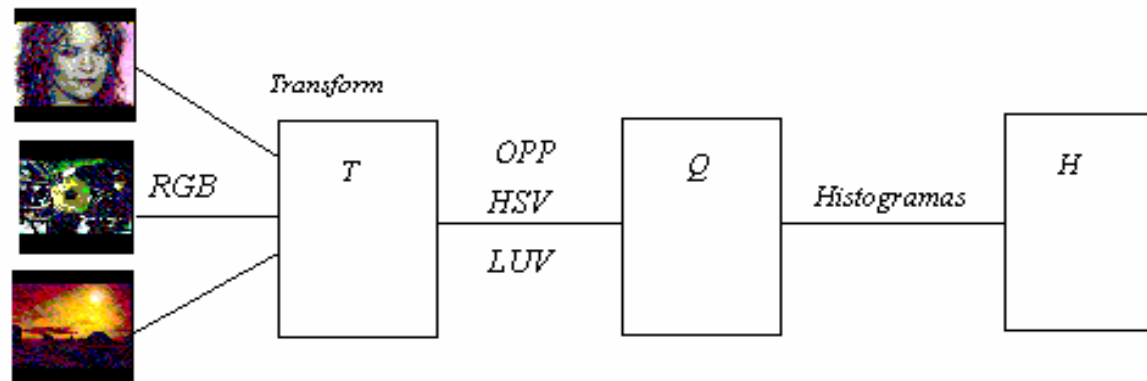
LUT

	Red	Green	Blue	Frequency	%
1	38	128	158	3	0.002
2	38	128	188	3	0.002
3	158	98	8	25	0.016
4	158	98	38	132	0.083
5	158	98	68	160	0.100
6	158	98	98	278	0.174
7	158	98	128	54	0.034
8	8	38	8	56	0.035
9	8	38	38	163	0.102

Brightness (+0) Contrast (x1.0) Saturation (x1.0) Color Rotation (0°) Perspective Scale

Pode ser usado para caracterizar uma imagem em *queries*

- Ou sistemas de busca baseado em conteúdo na banco de dados ou na WEB



Esquema de query usado em “COLOR IMAGE RETRIEVAL SYSTEM: A COMPARISON OF APPROACHES, by A. Conci , E. M. M. M. de Castro”

Histograma em cores pode ser quantizado para 1D

- De modo a representar um certo numero de cores em um espaço pré definido
- Por exemplo : 256, 216 ou 162.
- Assim o *color histograms* representa um ponto em um *m-dimensional space* relacionado a uma imagem.
- Se essa relação for biunívoca em um *dataset*
- Pode ser usada como um FEATURE ou CARACTERISTICA para identificar cada imagem

Como “quantizar” para 1D

- First, each pixel of the image is originally represented as a point in the RGB-color space:
- A color point is specified by the values of the three channels that contain the information of the color, $I[x,y]$, that is
- $I[x,y] = (I_R[x,y], I_G[x,y], I_B[x,y])$

Como fazer quantização 1D (cont):

- Second, the RGB image is transformed to the NUMBER of most important colors .
- Third the quantization is performed in the space \mathcal{C} assuming uniform number n_1, n_2, n_3 of bins per axis:
- Fourth: each image point is analyzed to compute the 1D normalized histogram with m cells:

$$h[i] = \frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} f(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ color} = i \\ 0 & \text{if } (x, y) \text{ color} \neq i \end{cases} \quad i=0,1,2,\dots,m$$

$$f(x, y) = Q_c^m T_c(v_c)$$

Where: X, Y are the number of pixels in each direction,

$$m = n_1 \times n_2 \times n_3 = 256, 216 \text{ or } 162.$$

Como fazer quantizado para 1D

- For huge images with big quantization :
- Color histograms are points in an m-dimensional space related "one-to-one" to the images.

Netscape - COLOR IMAGE RETRIEVAL SYSTEM: A COMPARISON OF APPROACHES

**COLOR IMAGE RETRIEVAL SYSTEM:
A COMPARISON OF APPROACHES**

choose color space and metric

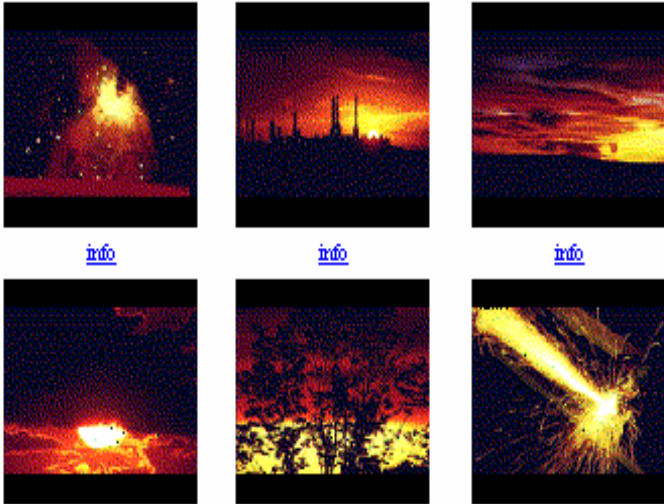
click an image to find other with similar color

RANDOM

HSV

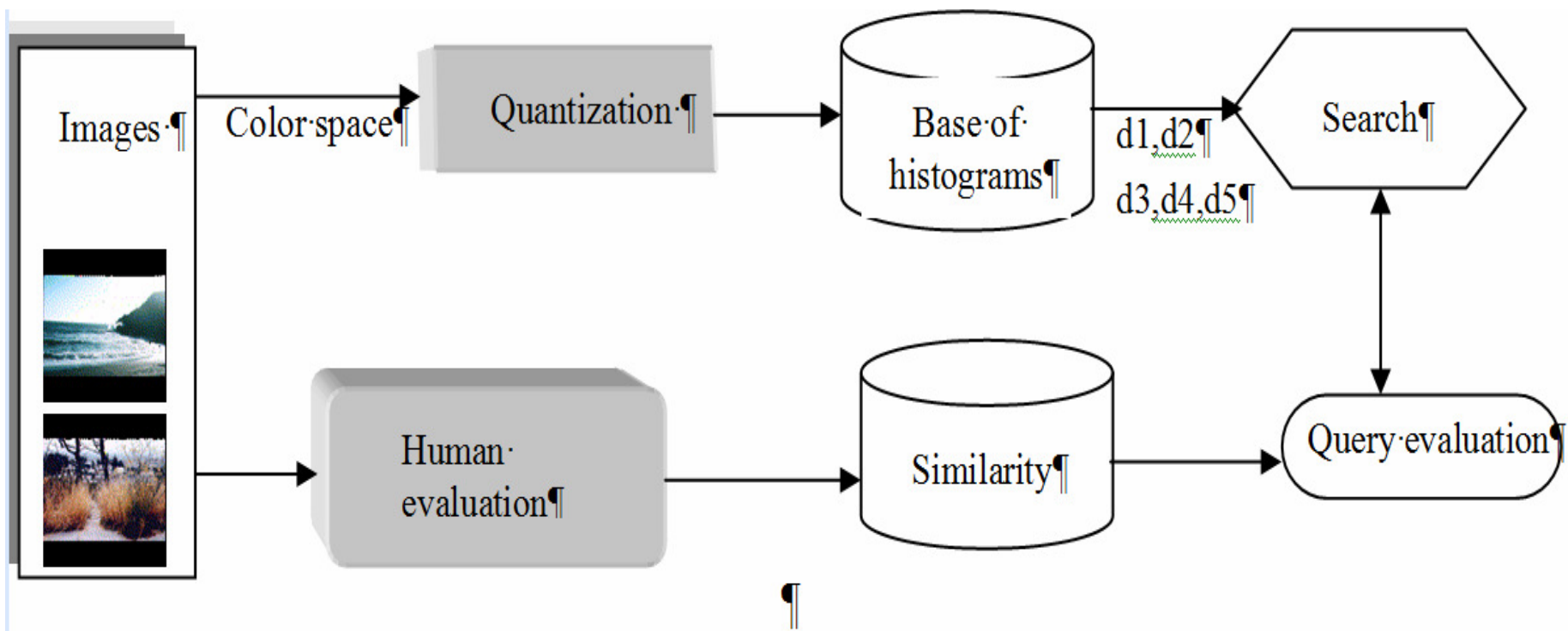
Intersecao

HELP



info info info

info info info



Query simulator diagram

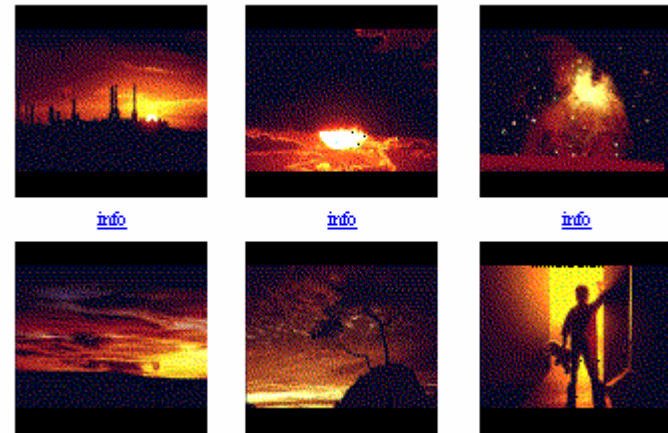
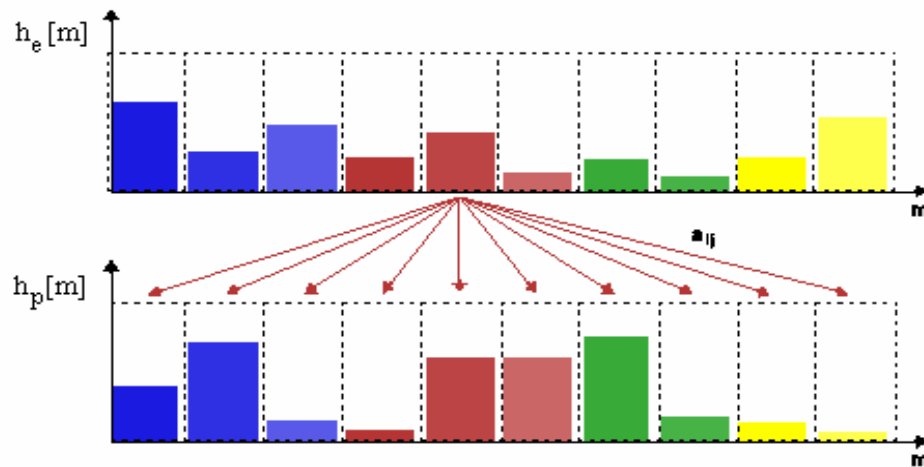
First 5 query results using d5 on the HSV(216)

All distances can be used as the Euclidian (d2)

The best distance measure form or (d5) use the expression:

$$(d_{e,p})^2 = (h_e - h_p)^T A (h_e - h_p)$$

where A is a matrix of similarity weights, $A=[a_{ij}]$, $0 \leq a_{ij} \leq 1$ and $a_{ii} = 1$



HSV216 (24x3x3) bins

Outras Operações Pontuais

Operações Aritméticas

Operações Lógicas

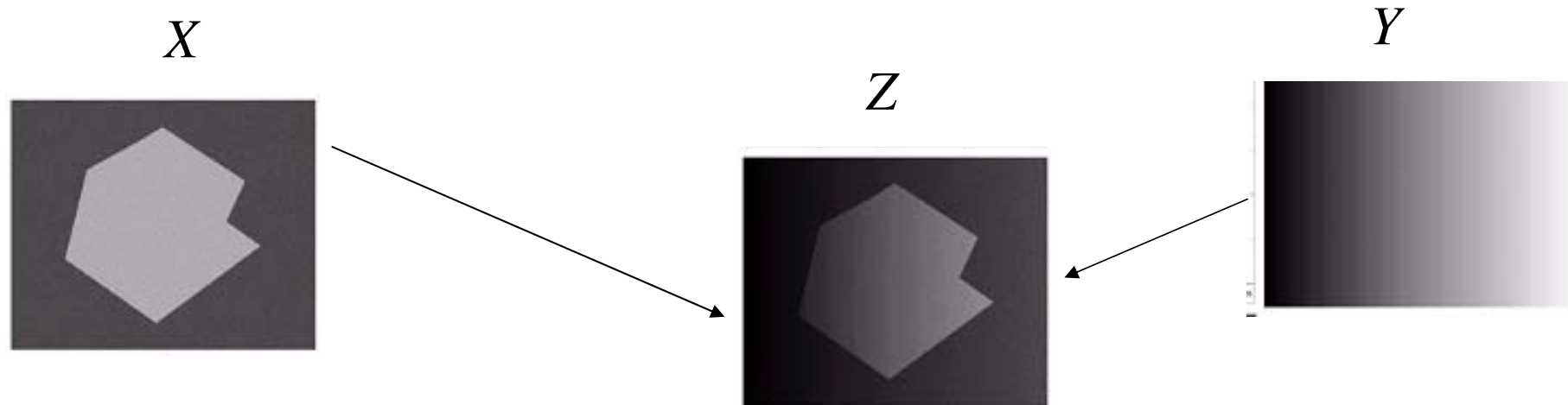
Operações em imagens

Global x local

Uma operação pontual **global** *m* em uma imagem digital *r* é uma função *f(r)* aplicada a **todo** *pixel* da imagem:

$$r: s = f(r)$$

O processamento pode levar em consideração dados globais de mais de uma imagem, como por exemplo, 2 combinadas resultando uma terceira

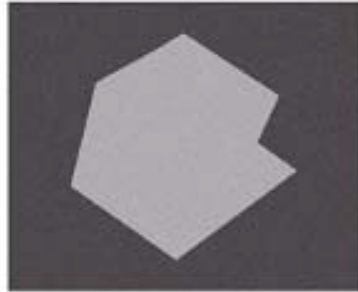


As operações *locais pixel-a-pixel* de duas imagens podem ser descritas pela expressão:

$$Z = (X \text{ Op}P Y)$$

OpP é um *operador qualquer* inclusive aritmético ou lógico.

Ex. iluminação inadequada

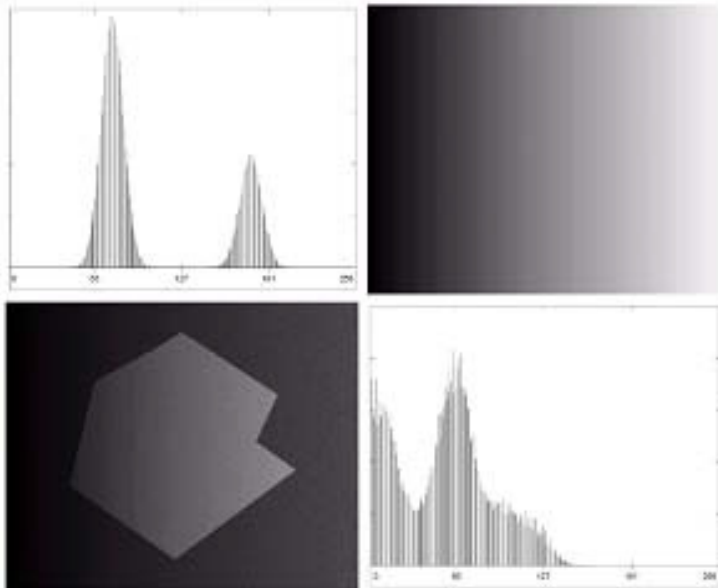


$$f(n_1, n_2) = i(n_1, n_2)r(n_1, n_2)$$

reflectância

$$f(n_1, n_2) = i(n_1, n_2)r(n_1, n_2)$$

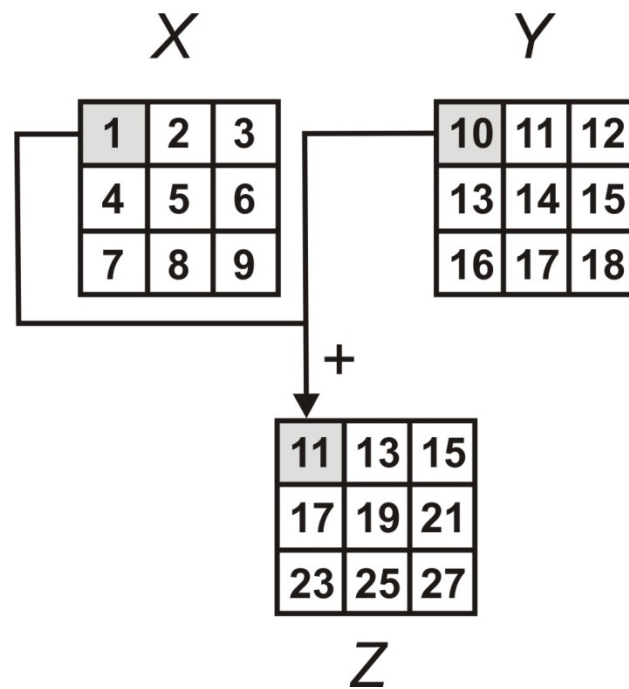
iluminação



$$f(n_1, n_2) = i(n_1, n_2)r(n_1, n_2)$$

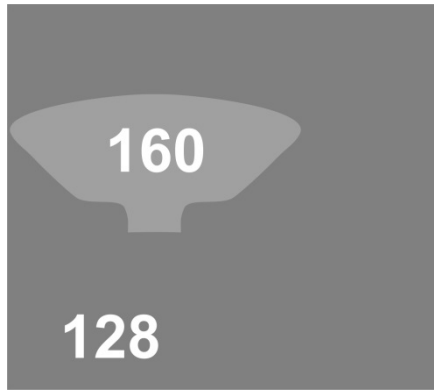
a limiarização global só obtém sucesso na separação de um objeto do fundo em ambientes altamente controlados.

Operações Aritméticas

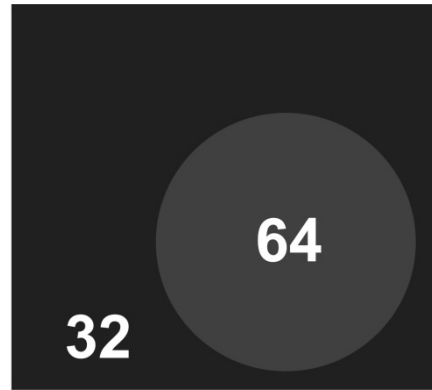


Exemplo de operação aritmética de soma

Imagens X e Y utilizadas como exemplos.

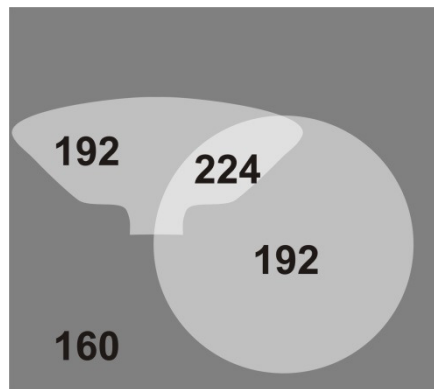


(a) imagem X



(b) imagem Y

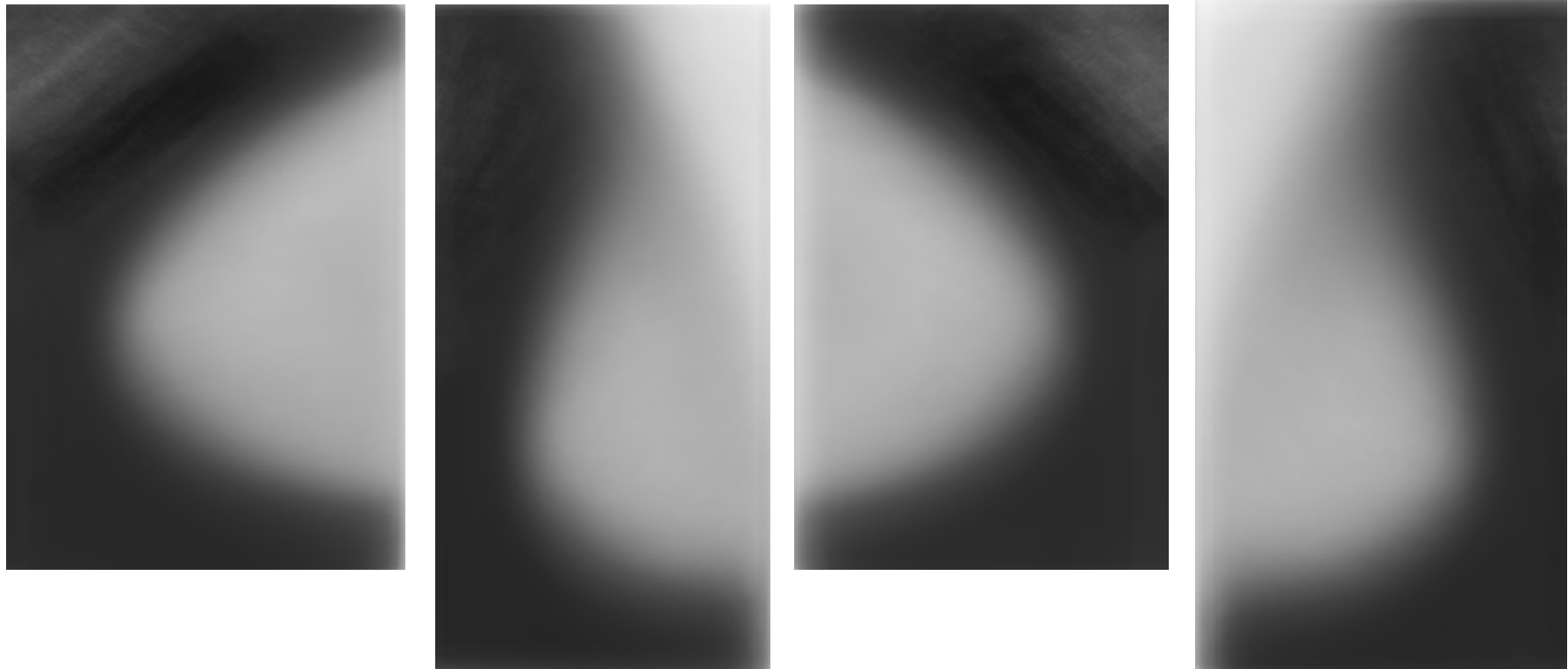
Exemplos de operações aritméticas com as imagens X e Y



(a) $Z = X + Y$



(b) $W = X - Y$



Resultado padrão médio de imagens mamó gráficas.

Etapas posteriores deverão definir os **contornos** destas imagens, mas essas ajudam por exemplo para projetar filtros para extrair o texto e ruído de fundo das imagens

Exemplo de aplicação de soma de imagens.

Busca de um **padrão médio** de imagens mamográficas.

Os passos para a geração das imagens foram os seguintes:

1 - Foi utilizado o primeiro volume de mamas normais do DDSM, quantizadas em 8bits. Foram utilizadas: **102 imagens** do tipo LCC, **110 imagens** LMLO, **101 imagens** RCC e **110 imagens** RMLO (L=left, R=Right, CC=Cranio Caudal, ML=médio lateral) .

2 - Calculado o **tamanho médio** das imagens

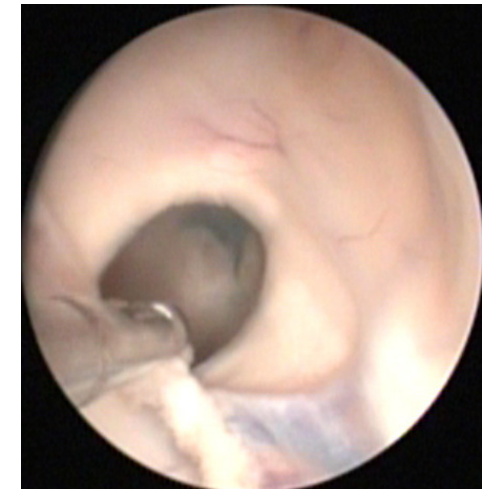
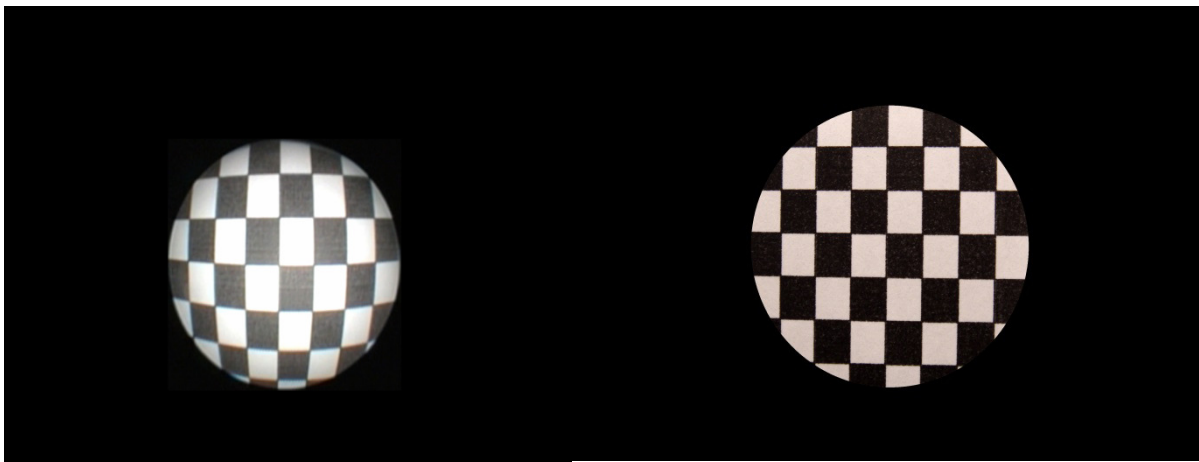
Como cada imagem tem um tamanho diferente, foi feito a média e escalou-se todas as imagens para esse tamanho médio.

3 - Calculada a **média das imagens** pela média aritmética dos valores de pixel das imagens.

Esses processos são **etapas** em problemas complexos, por exemplo:

Correção das imagens de um endoscópio, que causa sabidamente distorção *radial-fish eye*:

- 1- Binarização de um padrão de comparação *acquired*;
- 2- Subtração das imagens *standard* x *s. acquired*;
- 3- Identificar as distorções;
- 4- Corrigi-las aplicando técnicas de morphing (transformações geométricas)



Exemplo de subtração

- traffic analysis system for the detection and **tracking of vehicles**
- website demo gallery:
<http://www.axis.com/solutions/video/gallery.html>.
- we used the images from one static top-placed camera at Petrovka Street, Kiev (Ukraine).



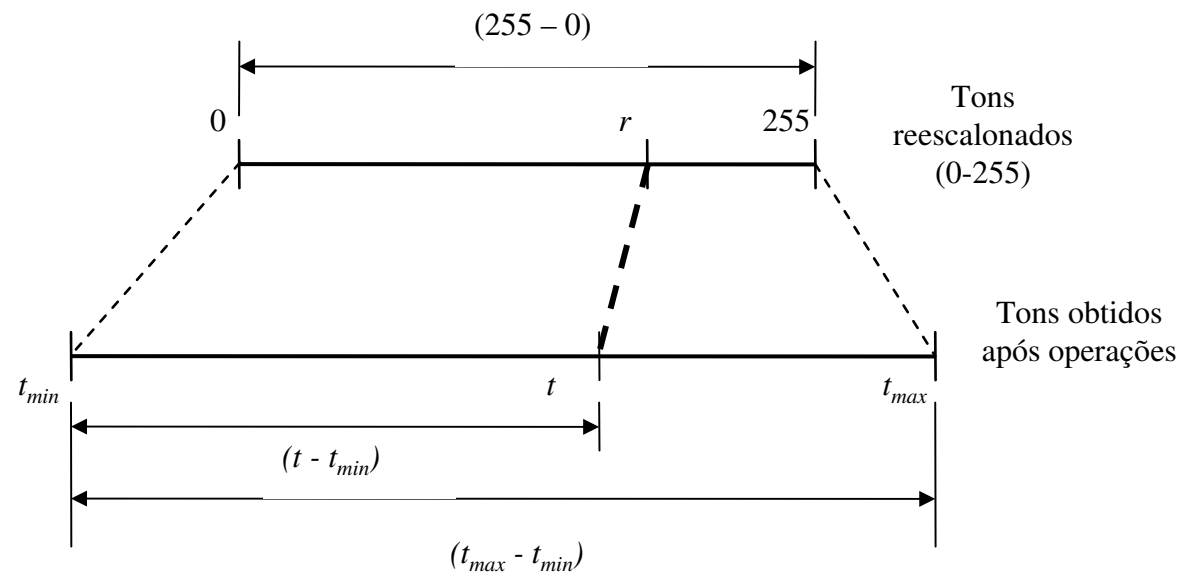
Traffic tracking visual results on a 15'' test video sequence (only 3 sample frames are shown).

Exemplo de subtração: tracking

(Left) Sample frame with rail lines drawn (in white), and (right) corresponding binary image obtained by **background subtraction** containing the detected targets (rail lines also help in a more robust detection of these targets).



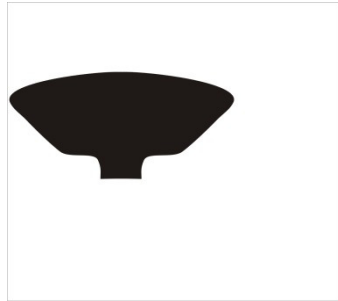
Detalhe importante nas Operações aritmeticas



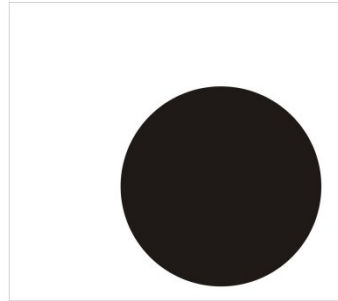
Limites Inferior e Superiores nas Operações

Re escolanamento em casos de *underflow* e de *overflow*.

Operações Lógicas



(b) Y



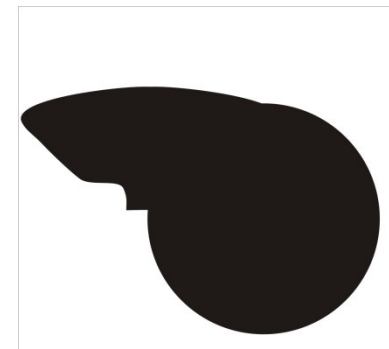
(a) X

Equivalentes as operações de **União, Interseção e Subtração (União-Interseção e X-Y)** de conjuntos

X e Y utilizados como modelo.

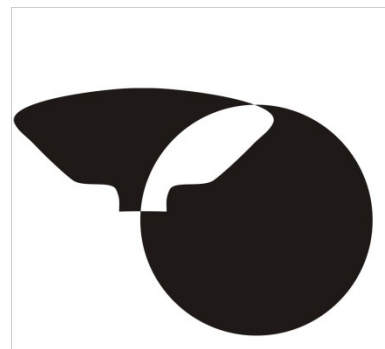


(a) (X) E (Y)

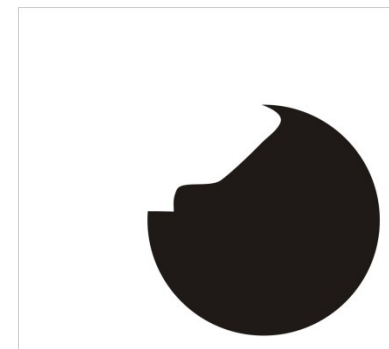


(b) (X) OU (Y)

Exemplos de operações lógicas com objetos X e Y.



(c) (X) XOU (Y)



(b) (NÃO X) OU (Y)

Trabalho prático.

É sempre muito útil procurar por a teoria em pratica:

- Alguma dessas operações podem ser úteis no estudo deste ano?
- Poderiam ser usadas para eliminar, fundos, ou detectar lugares prováveis de localização de elementos, por combinação de imagens com operações lógicas (na procura por formas específicas)?
- Podem ser uma técnica útil para diversas aplicações.

Nas imagens dos veterinários

- Por exemplo podemos usar a imagem media de todos os dados do banco de imagens dos veterinários, para tentar retirar áreas de possíveis anotações nas imagens e obter o lugar mais provável de estarem informações úteis.
- Onde mais?

Bem, além do histograma já estudamos as transformações da imagem ligadas as características (features) de Textura

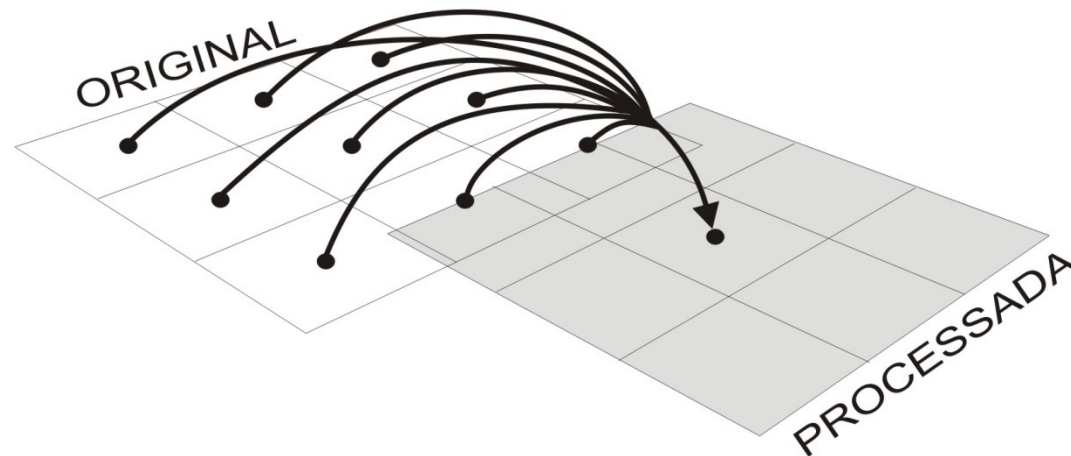
Que tipo de Operações nas imagens foram elas (as de Textura) ?

Essas operações usavam uma area local para definir o novo valor de Pontos/pixeis

Operações Locais

Um *pixel* da imagem resultante depende de uma **vizinhança** do mesmo *pixel* na imagem original

$$\text{OpL}(f(x_i, y_i), f(x_{i-1}, y_{i-1}), f(x_{i+1}, y_{i+1}), f(x_{i-1}, y_{i+1}), f(x_{i+1}, y_{i-1}), \dots)$$



Exemplo de uma operação local em uma área em torno do *pixel* (x_i, y_i) .

Filtro de média 3x3 :

$$g(x_i, y_i) = \frac{1}{9} \left(\begin{array}{l} f(x_{i-1}, y_{i-1}) + f(x_i, y_{i-1}) + f(x_{i+1}, y_{i-1}) + \\ f(x_{i-1}, y_i) + f(x_i, y_i) + f(x_{i+1}, y_i) + \\ f(x_{i-1}, y_{i+1}) + f(x_i, y_{i+1}) + f(x_{i+1}, y_{i+1}) \end{array} \right)$$



(c)



(d)

Imagem com Aliasing (c) e o efeito da aplicação de filtro de média (d).

Filtragem no Domínio Espacial

Os métodos de filtragem que trabalham no domínio espacial operam **diretamente sobre os *pixels***, normalmente utilizando operações de convolução com máscaras.

O uso de máscaras nas imagens no domínio espacial é usualmente chamado de filtragem espacial e as máscaras são chamadas de **filtros espaciais**.

As operações de convolução com máscaras são muito usadas e resultam inclusive em formas de se usar **Inteligência artificial** de maneira mais “bruta” e direta nas imagens .

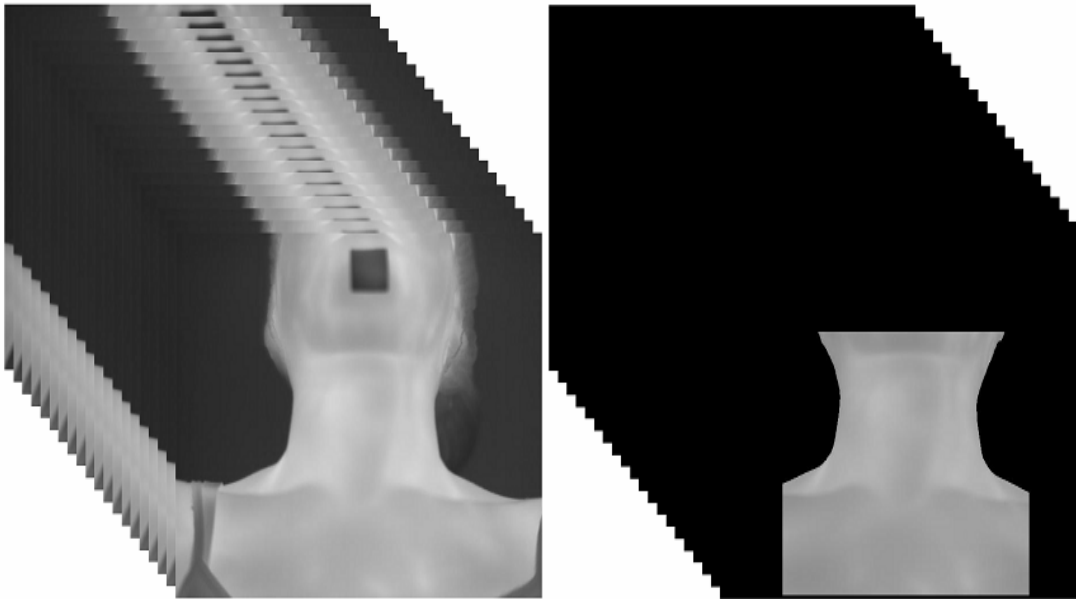
Exemplo de aplicação

CNNs for image

- related medical tasks:
 - GoogLeNet
 - AlexNet
 - VGG
 - ResNet

Experiments: thyroid Dataset

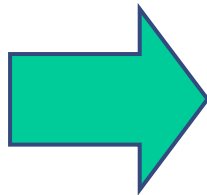
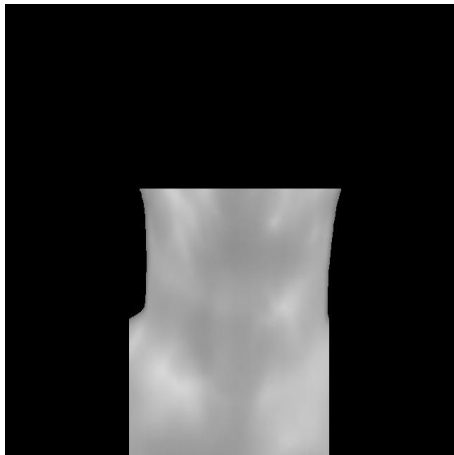
- ROI normalized grayscale images of the thyroid area.



Experiments: Segmentation

- Calculation of the parameter of asymmetry PA for each pixel:
 - $> 0.3^{\circ}\text{C}$ - the point with the higher value is considered, and the other has its intensity changed to 0.
 - $< 0.3^{\circ}\text{C}$ - intensity value of both points changed to 0.
- The outcome is an image that only contains regions with considerable temperature asymmetry.
- Morphological erosion operator for eliminating noise:
 - Structural element: 10-pixels square.

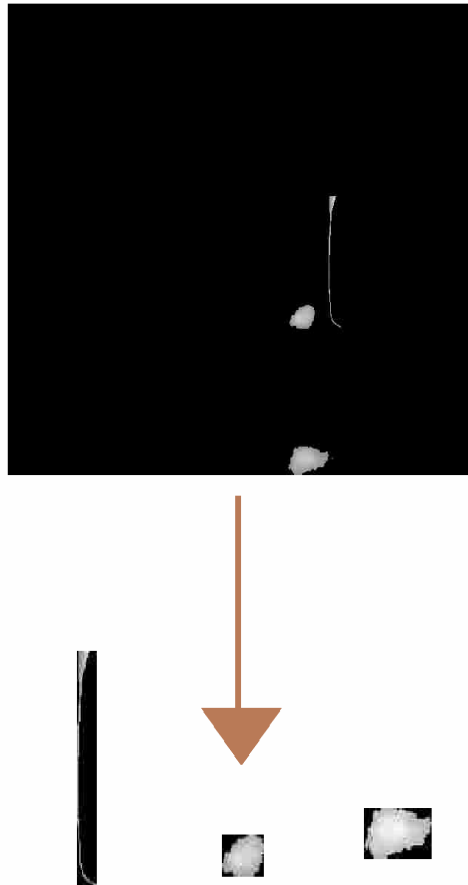
Experiments: Segmentation



Regioes candidatas a thyroid nodules

Experiments: Image processing

- New images are created from regions found in the segmentation.
- These images will be then analyzed by different CNNs.



Traditional Intelligent Approach



Full Artificial Intelligence Approach



Machine learning technique

- Convolutional neural network – CNN;
- Deep learning;
- Each time we have a considerable number of correct solution of any particular aspect to be analysed.

Deep learning

- is a special form of machine learning which is included in the concept of artificial intelligence.
- CNN is a neural network multilayered and multineural.
- For CNN there is no need to feature extraction.
- In CNN features are formed by network learning.

CNNs

- Can decide for themselves what information to learn instead of using the information presented to themselves.
- They can produce more successful results than classical NN.
- Disadvantage: CNN algorithms require **hardware and data capacity**.
- CNNs perform enormous computational processing (require parallel hardware like GPUs).

CNN

Learns the distinctive features itself according to a large number of inputs.

- The system should be adequately trained.
- The feature learning phase consists of layers.

CNN cont.

- Lower-level features provide the basis for producing more meaningful features.
- The features at the lower levels are less distinctive, while the properties in the upper layers, which are composed of combining the lower layers, have more distinctiveness.

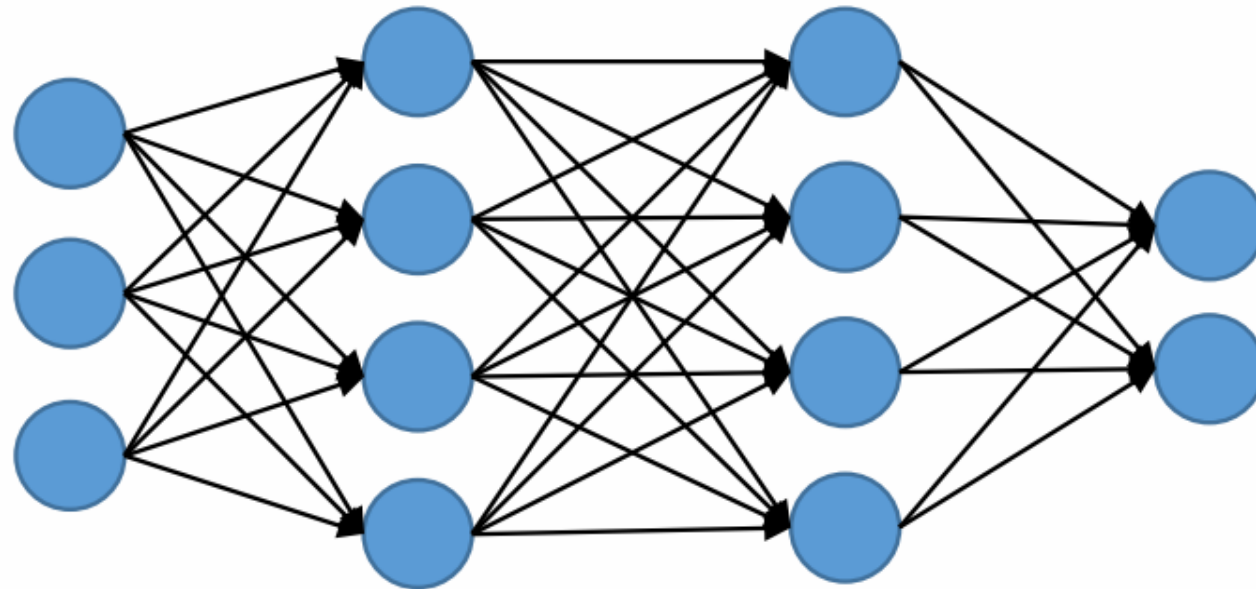
Artificial Neural Network - ANN

input layer

hidden layer 1

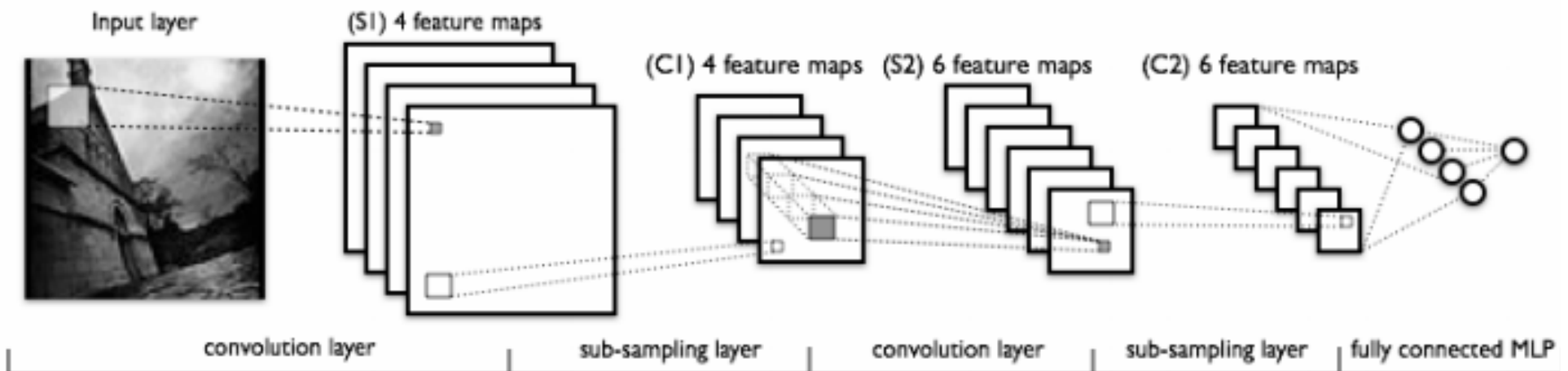
hidden layer 2

output layer

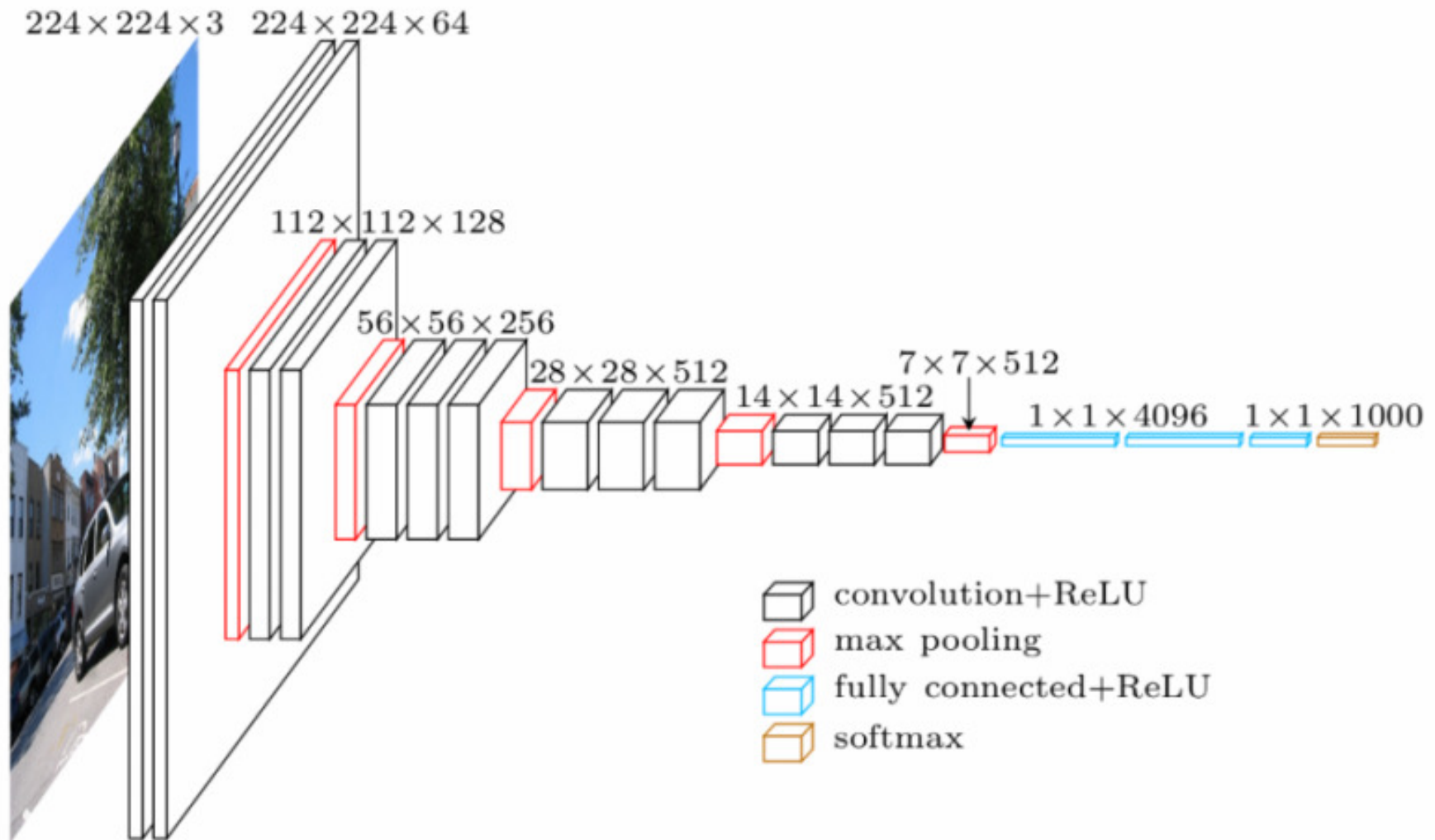


Nets on CNN consideration

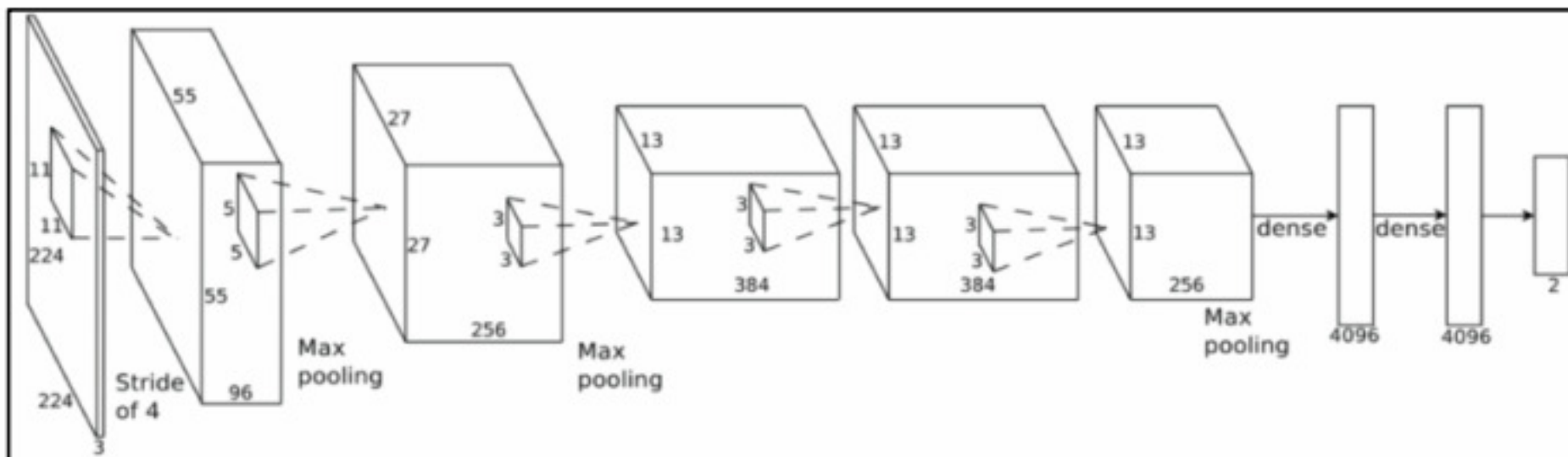
- LeNet architecture



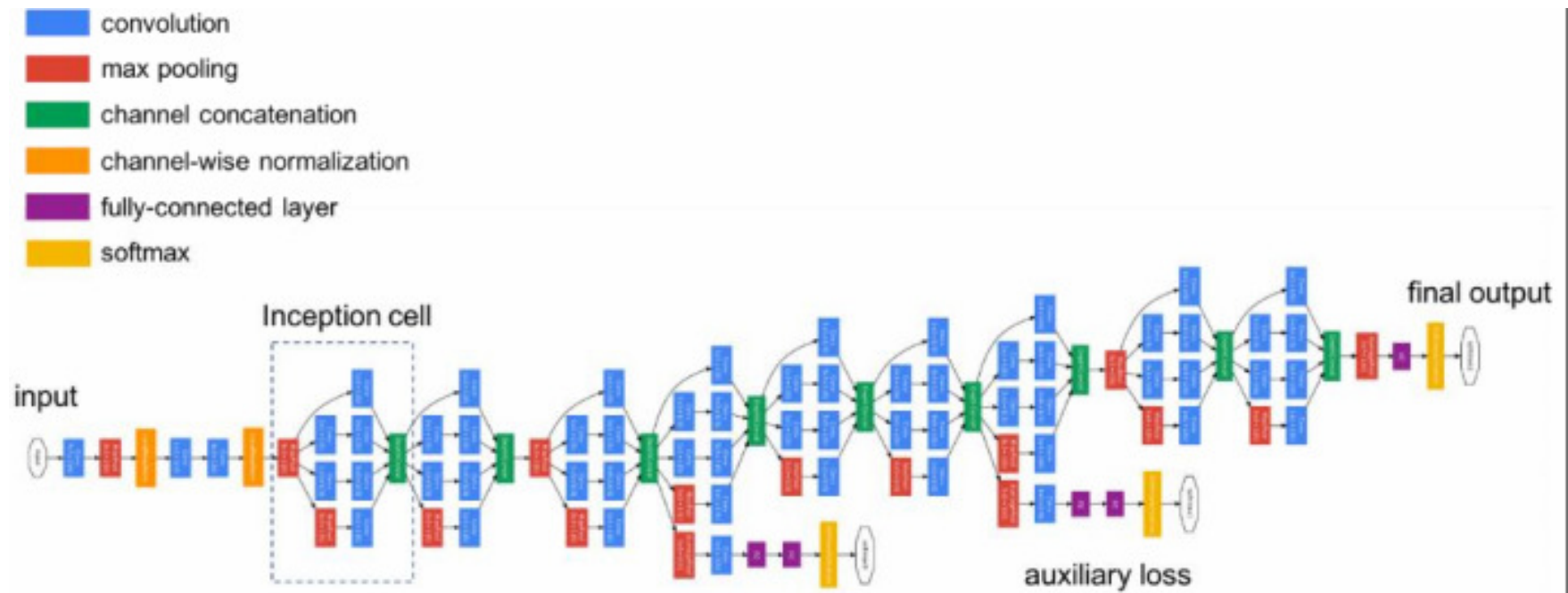
VGG(-16) architecture



AlexNet architecture



GoogLeNet architecture



ResNet architecture



Experiments: Classification


- Thyroid regions extracted:
 - 43% nodule related
 - 57% healthy regions
- Learning rate: 0.01
- 50 epochs
- 60% - training data.
- 40% - test data.

Results

Model	Accuracy	Validation accuracy
GoogLeNet	0.86	0.81
AlexNet	0.77	0.65
Resnet	0.94	0.90
VGG	0.74	0.61

Continuando só com a Resnet

Table 1. Results of the training process of each model in percentage.



Learning rate	Accuracy		Loss	
	In-sample	Validation	In-sample	Validation
0.1	96	95	14	8
0.01	94	90	16	29
0.001	92	87	18	25

Table 2. Confusion matrix of the selected model.

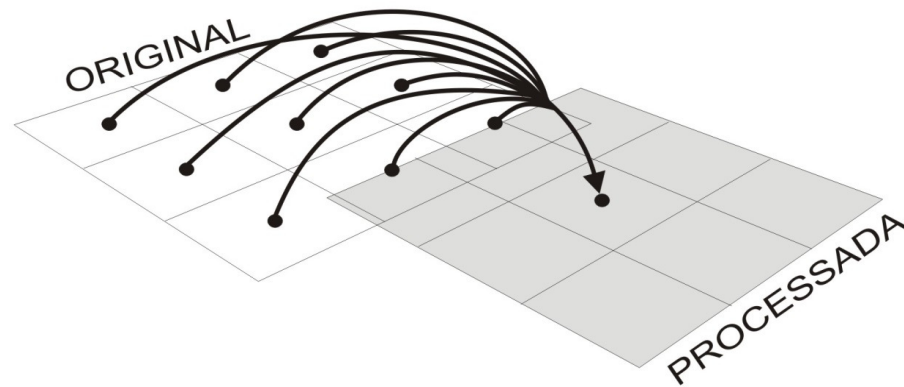
		Predicted	
		Nodular	Not nodular
Actual	Nodular	32%	8%
	Not nodular	0	60%

Mas é preciso ter muitas imagens classificadas para treinar a rede

Assim voltando para as operações de convolução com máscaras na maneira tradicional nas imagens

Processo de convolução com máscara no domínio do espaço

$OpL(f(x_i, y_i), f(x_{i-1}, y_{i-1}), f(x_{i+1}, y_{i+1}), f(x_{i-1}, y_{i+1}), f(x_{i+1}, y_{i-1}), \dots)$



$$g(x_i, y_i) = T[f(x_i, y_i)]$$

Onde:

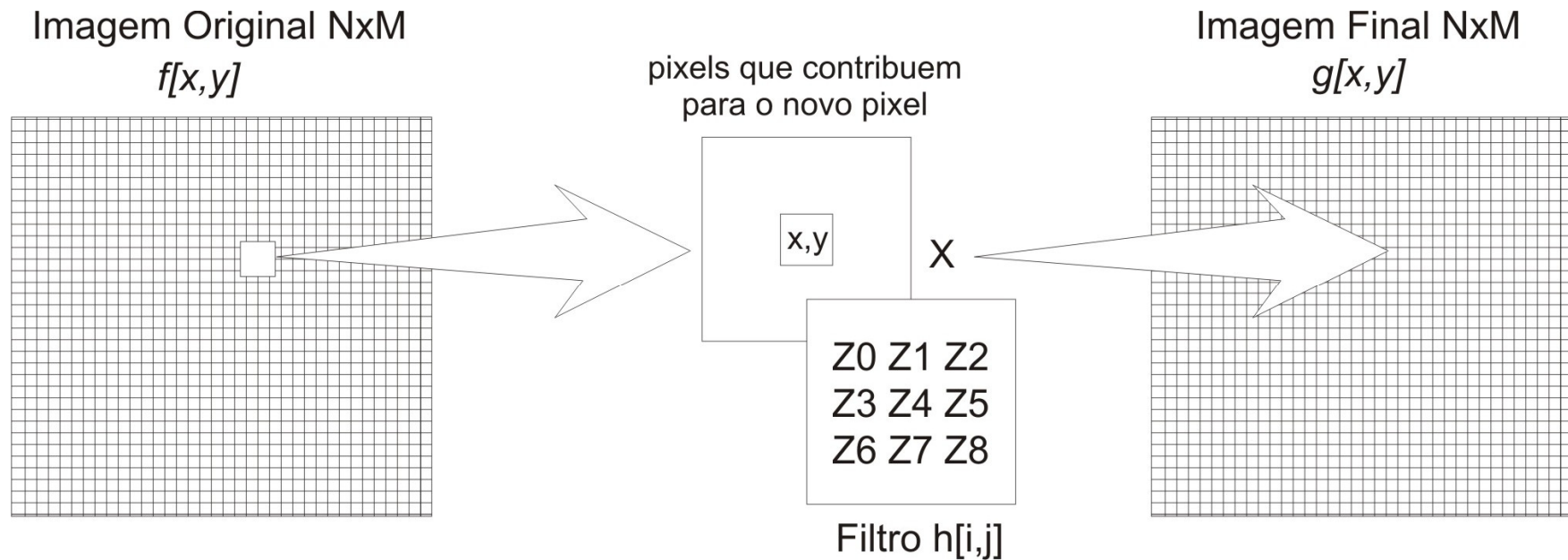
$f(x_i, y_i)$ é a imagem de entrada a ser filtrada,

$g(x_i, y_i)$ é a imagem na saída, processada, e,

T é um operador sobre f , definido em alguma vizinhança do *pixel* de posição (x_i, y_i) .

Convolução no domínio espacial :

$$g[x, y] = \sum_{i=1}^n \sum_{j=1}^m f[x-i, y-j] \cdot h[i, j]$$



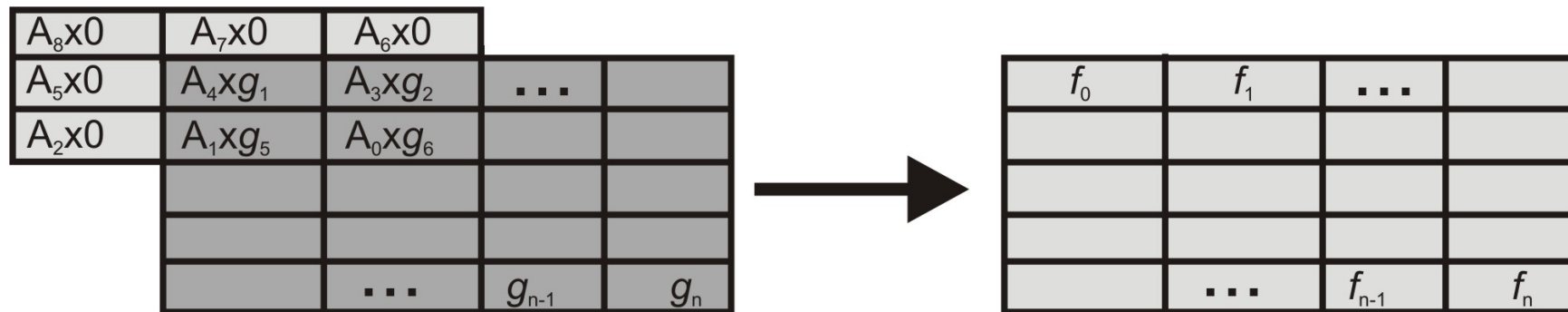
Esquema do processo de convolução por filtragem espacial usando uma máscara $n \times m$.

$$g(x,y) = T[f(x,y)] =$$

$$g(x,y) = Z_1 f(x-1, y-1) + Z_2 f(x, y-1) + Z_3 f(x+1, y-1) +$$

$$Z_4 f(x-1, y) + Z_5 f(x, y) + Z_6 f(x+1, y) + Z_7 f(x-1, y+1)$$

$$+ Z_8 f(x, y+1) + Z_9 f(x+1, y+1)$$



Processo de convolução com máscara no domínio do espaço

Filtros Lineares e Não Lineares

Filtros lineares: usam máscaras que realizam somas ponderadas das intensidades de *pixels* ao longo da imagem.

Se pesos diferentes forem usados em partes diferentes da imagem, o filtro linear é considerado espacialmente variante.

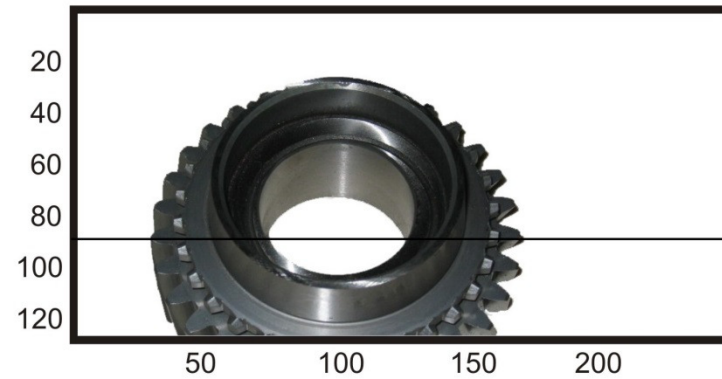
Filtros não-lineares: qualquer filtro que não é uma soma ponderada de *pixels*.

Filtros que buscam Patamares e Descontinuidades nas Intensidades da Imagem

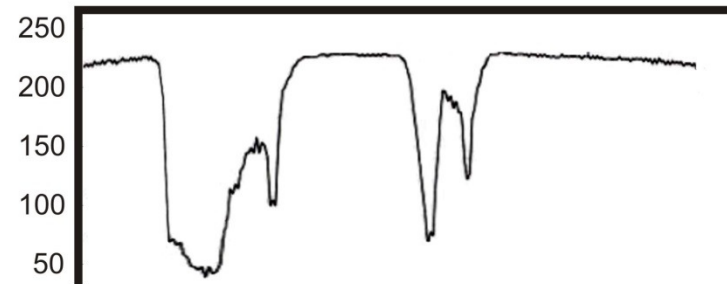
Podem ser pontos, linhas ou bordas de objetos da imagem.

O objetivo principal da **acentuação de contornos** é enfatizar detalhes dos limites de objetos de modo a permitir sua melhor identificação e análise posterior.

Porque a intensidade significa borda de objetos ou fundo



(a)



(b)

Exemplo de variação de intensidade ao longo da linha 80 da imagem Engrenagem.

Exemplos onde a intensidade significa borda de objetos ou regiões

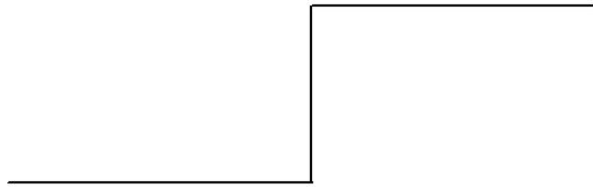
- **Contorno**: identificados por mudanças locais de intensidade significativas na imagem, ocorrendo tipicamente na separação de duas regiões diferentes.

- **Regiões ou objetos**: identificados por patamares mais ou menos constantes de tons e cores.

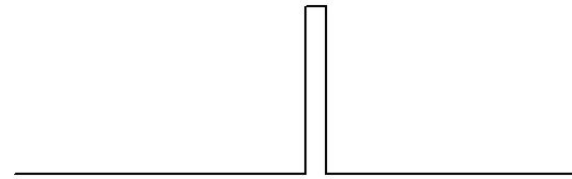
- **Fronteira**: ocorre onde a função de intensidade da imagem,

$f(x_i, y_i)$, varia bruscamente, sendo limites de regiões cujos valores de tonais apresentam grandes diferenças.

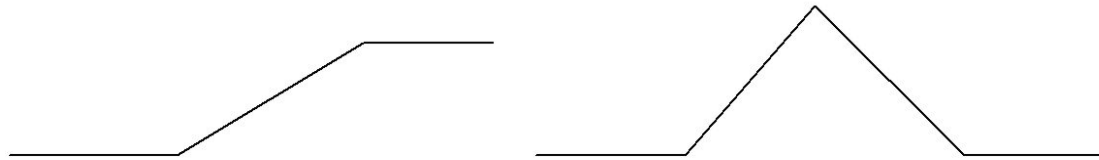
Descontinuidades:



Descontinuidade do tipo degrau.



. Descontinuidade do tipo pico.



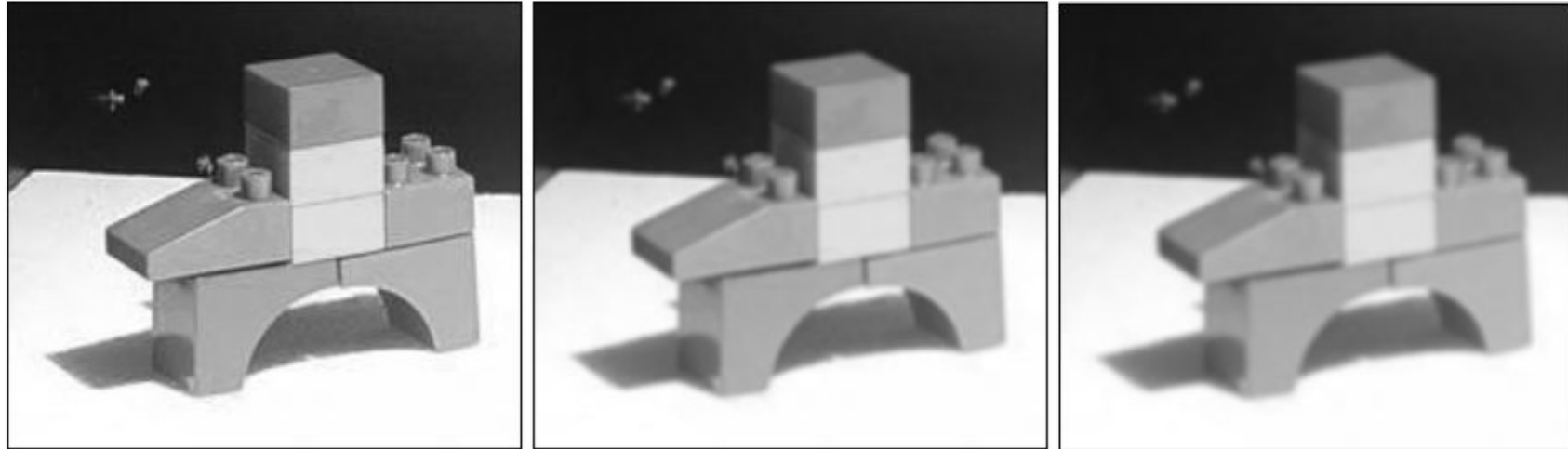
Modelos de descontinuidade em rampa.

Filtros Passa Baixa

Filtros de Suavização

Objetivos:

- **Suavizar** a imagem pela **redução das variações nos de níveis de cinza** que dão à aparência de “serrilhado” nos patamares de intensidade.
- Atenuar as **altas frequências**, que correspondem às transições abruptas.
- Minimizar ruídos.



(a)

(b)

(c)

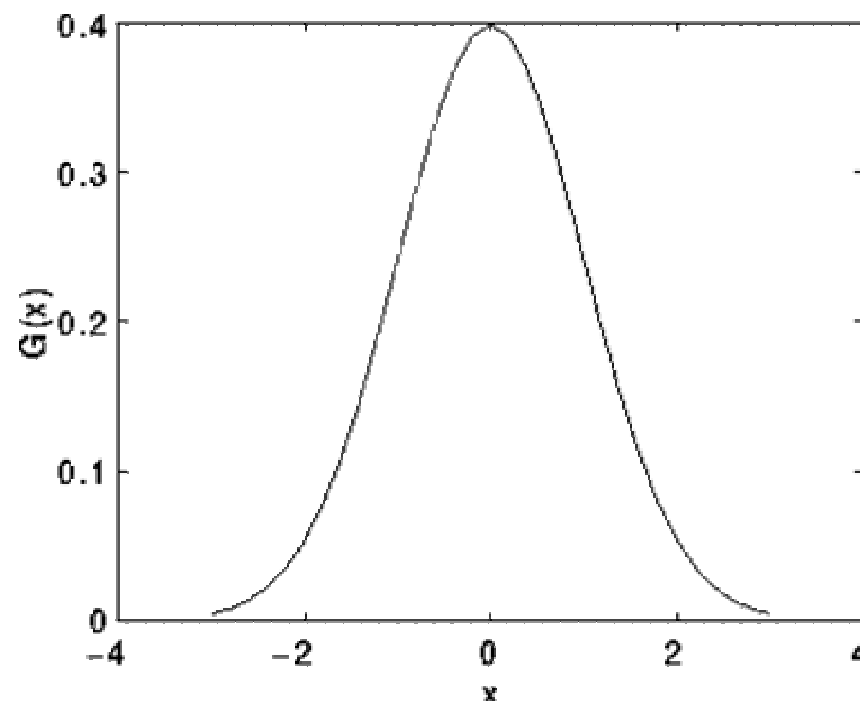
Imagem Blocos original (a) e o resultado da aplicação do filtro de média com máscara 3x3 (b) e 5x5 (c).

O que ocorre com o aumento da área usada na média?

Filtro Gaussiano

O filtro de suavização Gaussiano é baseado em uma aproximação digital da função gaussiana. O **Filtro Gaussiano em 1-D** é descrito por:

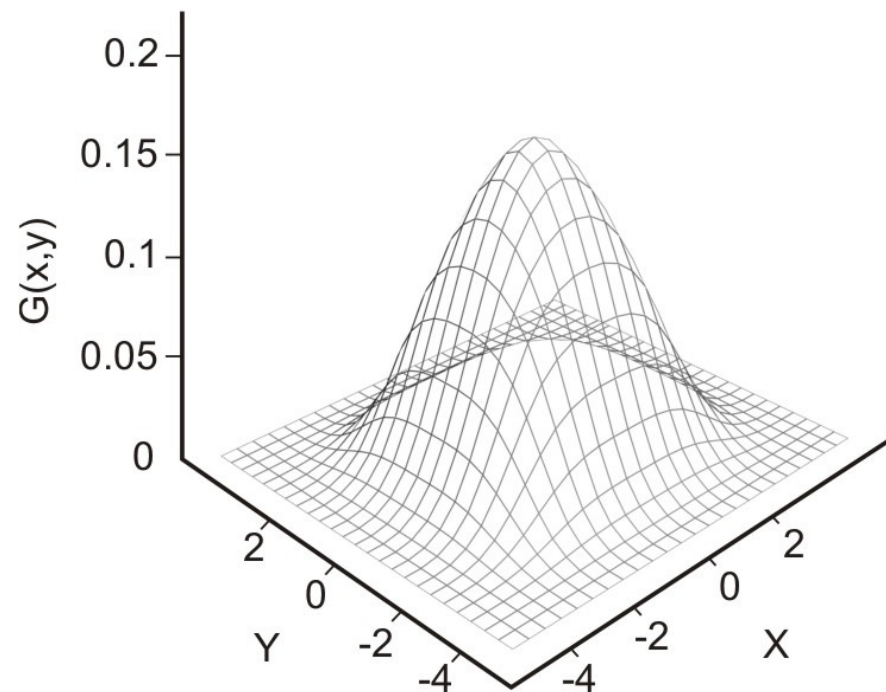
$$G(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



Forma 1D da função Gaussiana com média igual a zero e desvio padrão igual a um.

Em 2-D sua forma é dada por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$$



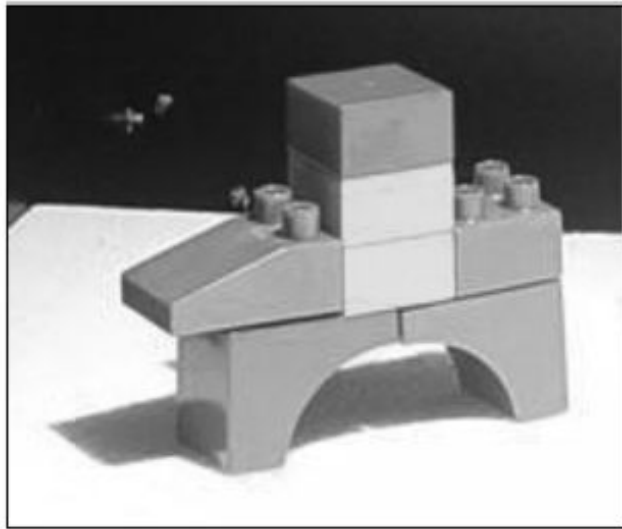
Representação da função Gaussiana em 2D com média em (0,0) e desvio padrão $\sigma = 1$.

Uma **forma aproximada** da Gaussiana para $\sigma = 1,0$ é apresentada no kernel 5x5 :

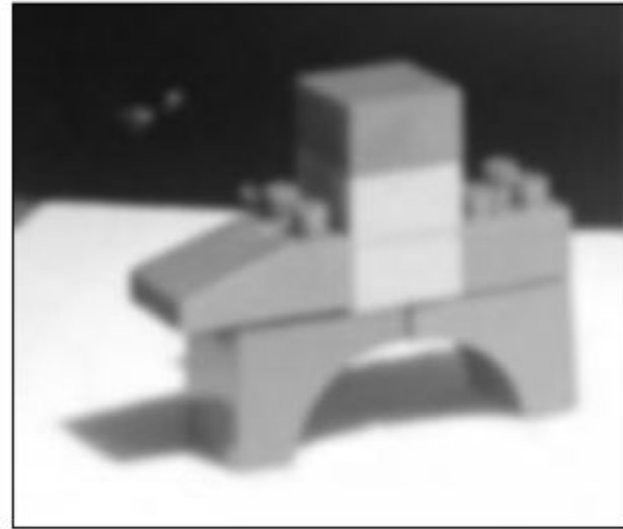
$$Z = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

A forma discreta **aproximada** 3x3 da função Gaussiana:

$$Z = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



(a)



(b)

Resultado da aplicação do filtro Gaussiano (b) à imagem Blocos (a) (Silva, 2004).

Filtros combinados

- Os filtro de média (e demais) podem ser generalizados usando outras formas de ponderação dos valores além do usual:

$$g[x, y] = \sum_{i=1}^n \sum_{j=1}^m f[x-i, y-j] \cdot h[i, j]$$

- Considerando uma função

$u(f)$

e sua inversa:

$u^{-1}(f)$

sendo $h[i, j]$ os pesos do kernel:

$$g[x, y] = u^{-1} \sum_{i=1}^n \sum_{j=1}^m u(f[x-i, y-j]) h[i, j]$$

Filtros de média não lineares

• Exemplos são:

• Média aritmética $u(f) = f$:

$$g[x, y] = \sum_{i=1}^n \sum_{j=1}^m f[x-i, y-j] \cdot h[i, j]$$

• Média harmônica : $u(f) = 1/f$:

• Média geométrica: $u(f) = \log f$

cuja inversa é a $\exp f$

$$g[x, y] = u^{-1} \sum_{i=1}^n \sum_{j=1}^m u(f[x_i, y_j]) h[i, j]$$

Filtros Separáveis

- Em 2D ou 3D significa que o mesmo kernel 1D (filtro) pode ser passado 2 ou 3 vezes em direções ortogonais.
- Reduzindo os custos computacionais(nos caso dos 5x5: 25 multiplicações + 24 somas ficam 10 multiplicações e 8 somas.

filtro Gaussiano

$$\frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4} [1 \ 2 \ 1] = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Média

$$\frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \frac{1}{3} [1 \ 1 \ 1] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtro de Mediana

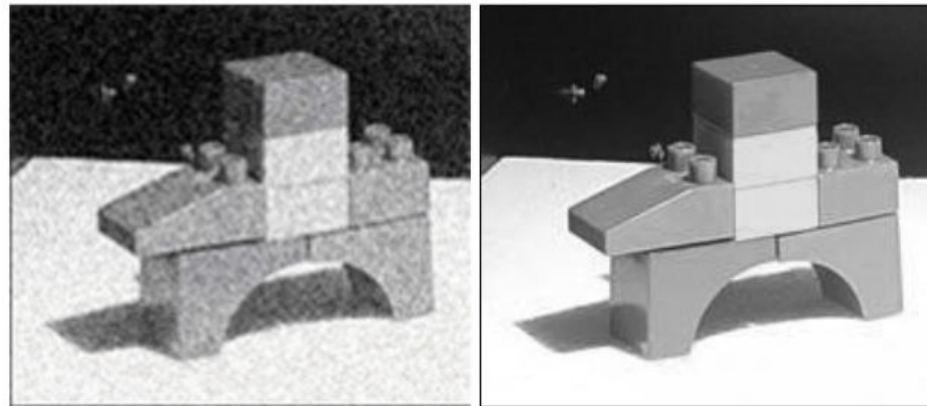
Não linear e separável

Ordena a intensidade dos *pixels* dentro da área da máscara em ordem **crescente ou decrescente**, alocando ao *pixel* da imagem correspondente à posição central da máscara, o valor da intensidade do *pixel* que corresponde à posição intermediária do respectivo intervalo ordenado.

Observe que nas definições dos filtros seu **tamanho** é muito importante e deve ser relacionado com o **tamanho do ruído** que se deseja eliminar

Aplicações:

- ruído é impulsivo (do tipo não contínuo, consistindo em pulsos irregulares de grandes amplitudes),
- ruído do tipo *sal e pimenta* (representando descontinuidades abruptas e isoladas na imagem).



(a)

(b)

Resultado da aplicação do filtro de mediana (b) na imagem Blocos degradada com ruído impulsivo (a).

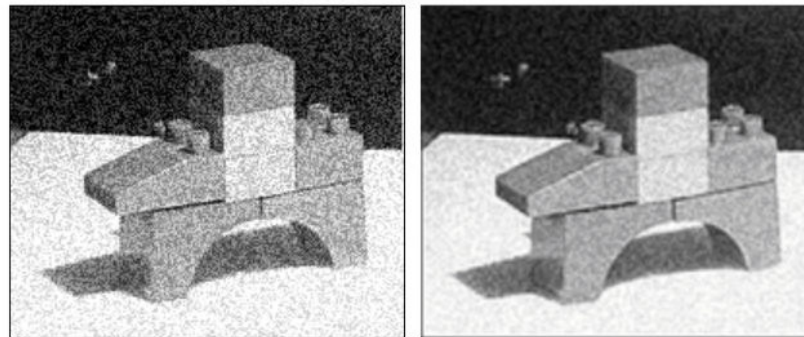
Filtros de Ordem e de Moda

Filtro de ordem: as intensidades dos pontos da vizinhança do *pixel* $f(x,y)$, dentro de uma janela da imagem, são ordenadas e é tomado o **valor máximo** ou o valor de uma ordem qualquer desta ordenação, como novo valor para $g(x,y)$.

Filtro de moda: as intensidades dos pontos da vizinhança do *pixel* (x,y) , dentro de uma janela da imagem, são ordenadas e é tomado como novo valor para $f(x,y)$, o valor **$g(x,y)$ mais freqüente** da vizinhança.

Filtro de Média dos k Vizinhos Selecionados

- É um híbrido do método de filtragem pela média.
- Utiliza a diferença absoluta entre o valor de cada elemento da máscara de filtragem e o ponto central para selecionar os k elementos que participarão do cálculo da média.



(a)

(b)

Resultado da aplicação do filtro de média dos k vizinhos (b) sobre a imagem Blocos degradada (a) (Silva, 2004)

Filtros Passa-Alta ou de Acentuação

- Atenuam ou eliminam as baixas frequências, realçando as altas frequências.
- Usados para realçar os detalhes na imagem (*intensificação ou “sharpening”*).
- Destacam características como pontos, bordas, linhas, curvas e manchas.
- Tornam mais nítidas as transições entre regiões diferentes (como os contornos), **realçando o contraste**.

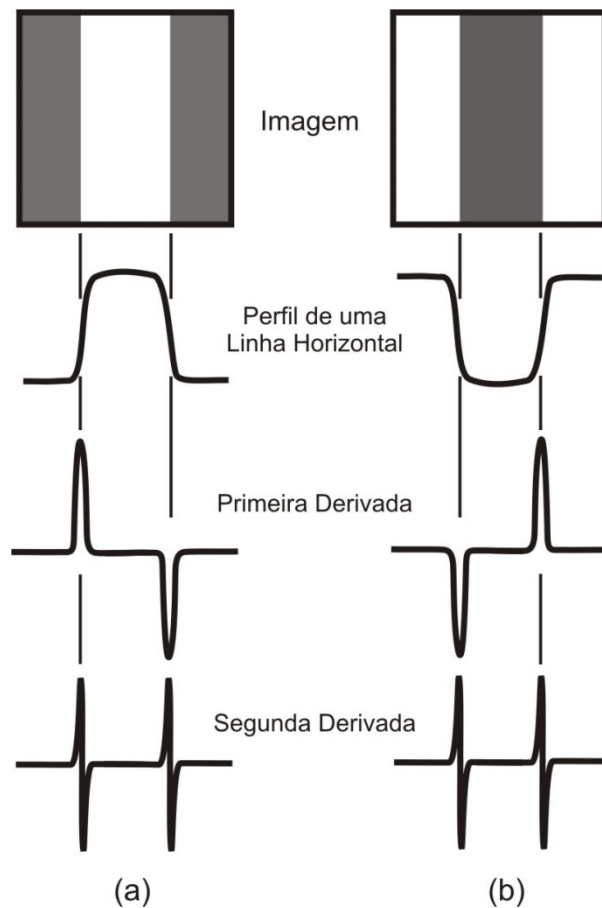
Um filtro **passa-alta ideal**, no domínio de frequência (u, v) , é descrito pela seguinte relação:

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

onde D_0 é a frequência de corte e $D(u, v)$ é a distância do ponto (u, v) à origem do plano de frequência, isto é:

$$D(u, v) = \sqrt{(u^2 + v^2)}$$

Filtros Gradiente ou Derivada de Imagens



Derivadas de uma Imagem.

O gradiente de uma função de duas variáveis $f(x,y)$, nas coordenadas (x,y) é definido por:

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

a magnitude deste vetor é:

$$|\nabla f| = \left[G_x^2 + G_y^2 \right]^{1/2}$$

A direção do vetor gradiente também é uma característica importante para muitas aplicações, ela é dada por:

$$\alpha(x, y) = \text{arc tang} \frac{G_y}{G_x}$$

O gradiente pode ser aproximado por:

$$\nabla[f(x, y)] \cong \left\{ [f(x, y) - f(x+1, y)]^2 + [f(x, y) - f(x, y+1)]^2 \right\}^{1/2}$$

que pode ser aproximada usando a diferença na direção y e na direção x combinadas como:

$$\nabla f \cong \left[(Z_5 - Z_8)^2 + (Z_5 - Z_6)^2 \right]^{1/2}$$

$$Z = \begin{bmatrix} Z_1 & Z_2 & Z_3 \\ Z_4 & Z_5 & Z_6 \\ Z_7 & Z_8 & Z_9 \end{bmatrix}$$

Pode-se também considerar:

$$\nabla f \cong |Z_5 - Z_8| + |Z_5 - Z_6|$$

Filtro de Sobel

Aplicação de duas máscaras:

$$Z_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

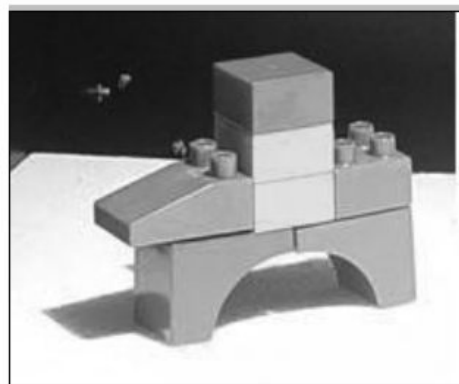
$$Z_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Forma de aplicação como separáveis, $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * [+1 \ 0 \ -1] * A$

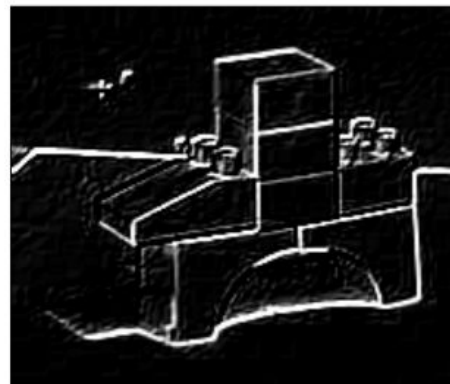
Considerando Z_1 a Z_9 os tons de cinza em torno do ponto onde o contorno será avaliado este filtro é dado por:

$$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$

$$G_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)$$



(a)



(b)

Imagem original (a) e o resultado da aplicação dos filtros de Sobel combinados como soma de imagens (b).

Filtro de Roberts

O filtro de Roberts pode ser representado pelas seguintes máscaras de convolução:

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$



(a)



(b)

Imagem Blocos original (a) e o resultado da aplicação do filtro de Roberts (b).

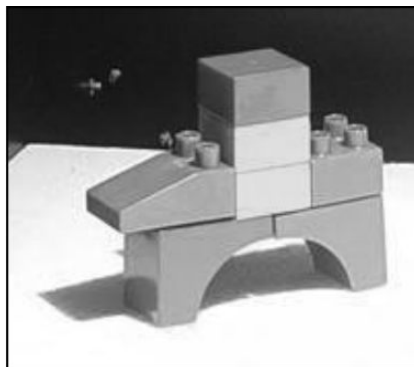
O filtro de Prewitt

O filtro de Prewitt tem o mesmo conceito do de Sobel (sem o peso para o *pixel* mais central) e de Roberts (sua máscara abrange uma área de 3 x 3):

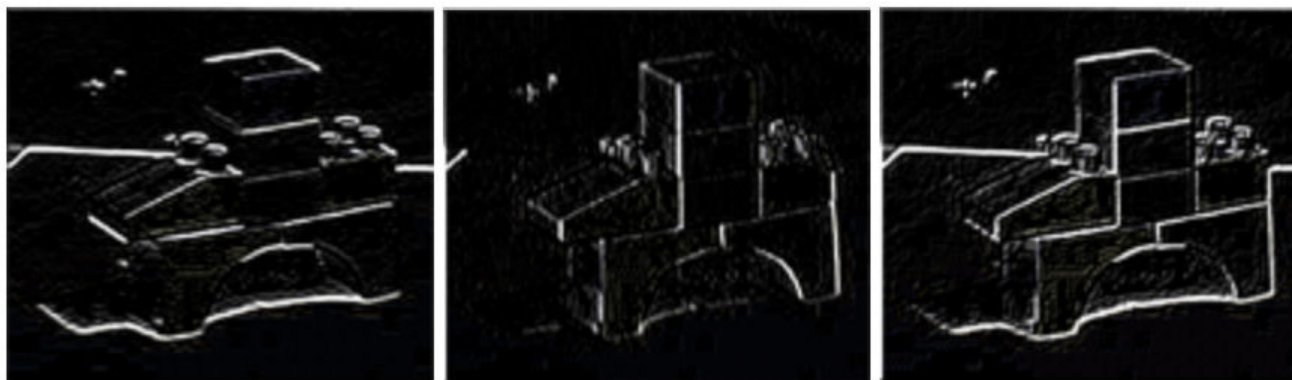
$$\nabla f = \left| (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3) \right| \dots + \left| (Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7) \right|$$

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$



(a)



(b)

(c)

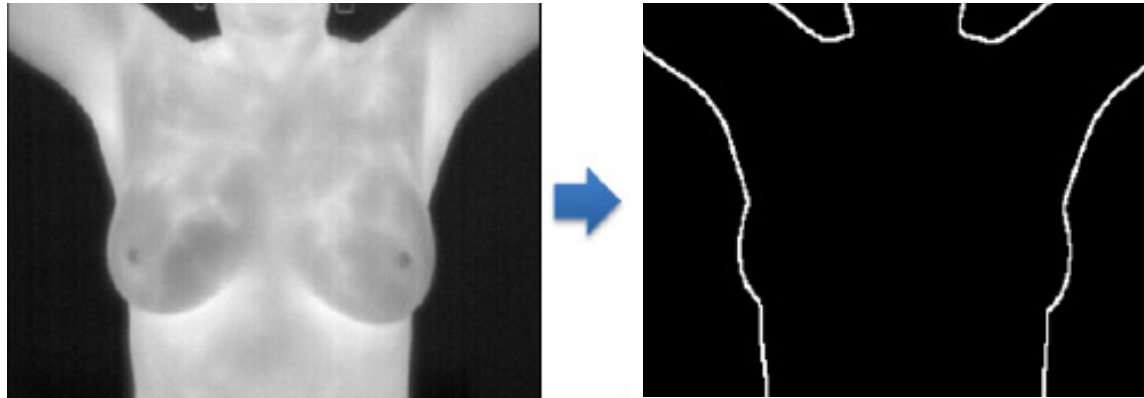
(d)

Aplicação do operador gradiente na imagem Blocos original (a), com detecção de contorno no sentido horizontal (b), no sentido vertical (c) e o resultado da soma dos sentidos vertical e horizontal (d) .

Filtro Laplaciano

O Laplaciano de uma função bidimensional é **a derivada de segunda ordem** definida como:

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



Derivação:

Considere uma janela 3x3 de uma imagem f no ponto (x, y) como se segue:

$f(x - 1, y + 1)$	$f(x, y + 1)$	$f(x + 1, y + 1)$
$f(x - 1, y)$	$f(x, y)$	$f(x + 1, y)$
$f(x - 1, y - 1)$	$f(x, y - 1)$	$f(x + 1, y - 1)$

Derivação:

Logo:

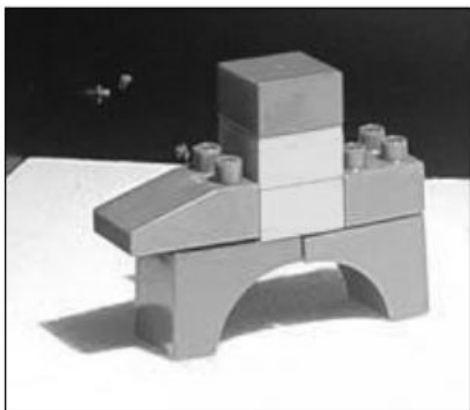
$$\begin{aligned}\frac{\partial^2 f}{\partial^2 x} &= [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial^2 y} &= [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)] \\ &= f(x, y+1) + f(x, y-1) - 2f(x, y)\end{aligned}$$

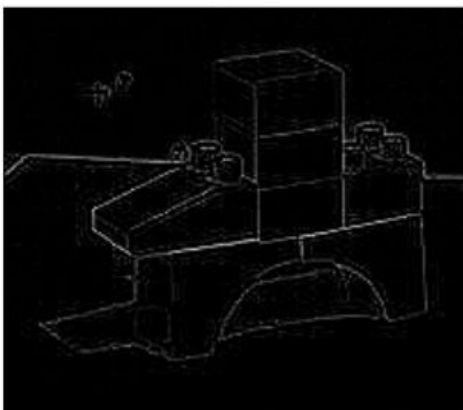
Derivação:

$$\begin{aligned}\nabla^2 f(x, y) &= [f(x + 1, y) + f(x - 1, y) - 2f(x, y)] + \\ &\quad [f(x, y + 1) + f(x, y - 1) - 2f(x, y)] \\ &= f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)\end{aligned}$$

$$\begin{bmatrix} 0, & 1, & 0 \\ 1, & -4, & 1 \\ 0, & 1, & 0 \end{bmatrix}$$



(a)



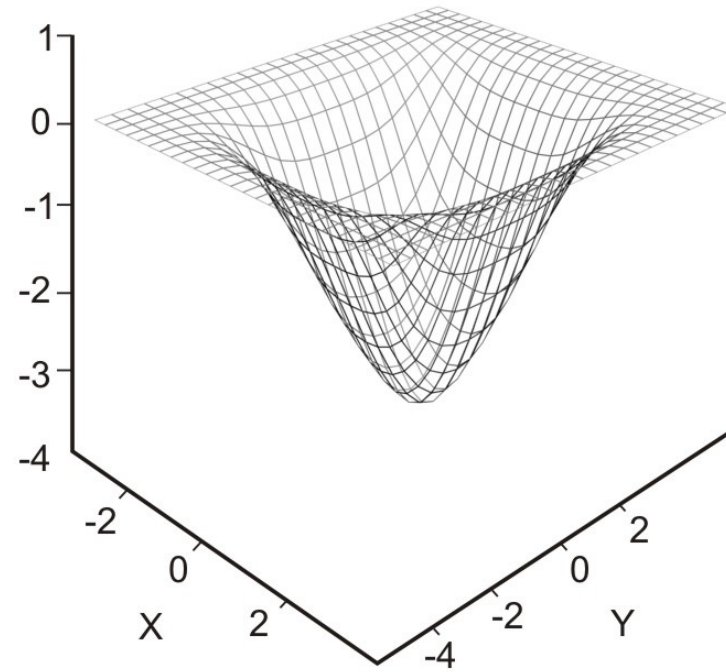
(b)

Imagem Blocos original (a) e o resultado da aplicação do filtro Laplaciano (b).

Filtro LoG ou Laplaciano do Gaussiano

O filtro LoG (*Laplacian of Gaussian*) ou Filtro Laplaciano do Gaussiano é representado por:

$$LoG(x, y) = -\frac{1}{\pi \sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



Forma em 2-D do filtro Laplaciano do Gaussiano (LoG).

Para uma Gaussiana com $\sigma = 1.4$ esta função pode ser aproximada na forma digital.

$$Z = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -24 & -40 & -24 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}$$

Outros Filtros Direcionais de Ampliação de Contraste

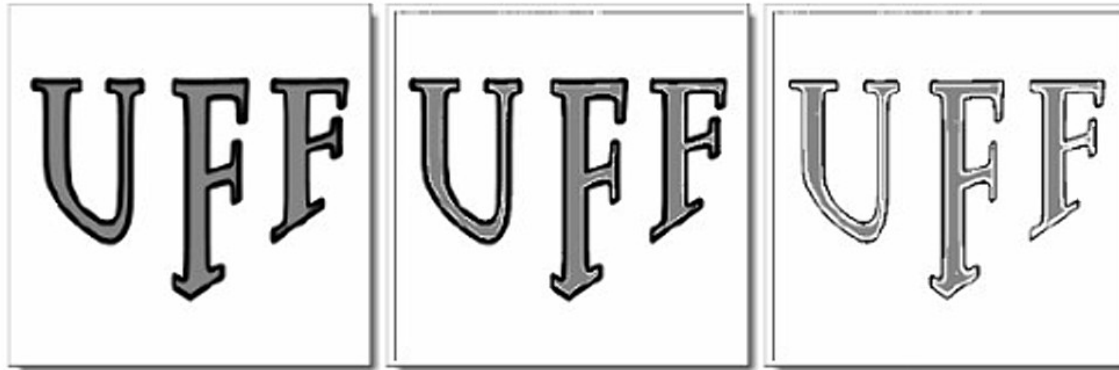
Exemplo de filtros direcionais.

Norte	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -2 & -1 \end{bmatrix}$	Nordeste	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$
Oeste	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	Noroeste	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$
Sudeste	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	Sudeste	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$
Sul	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	Leste	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$

Comparando os Diversos Filtros de Realce

Resumo das máscaras para realce.

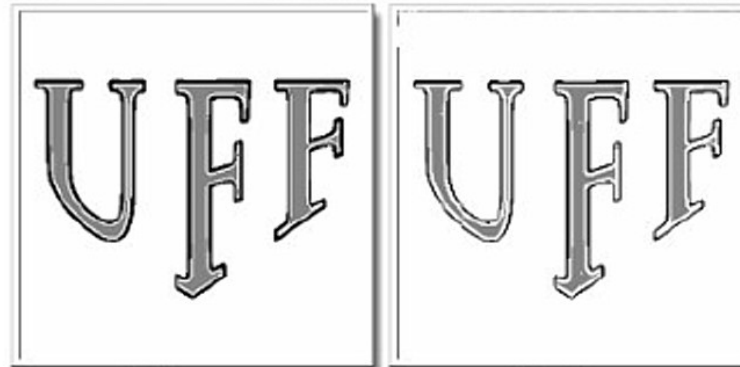
Operador	Vertical	Horizontal
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Sobel	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Prewit	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Laplaciano	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	



a) Imagem Original

b) Lapaciano

c) Prewit

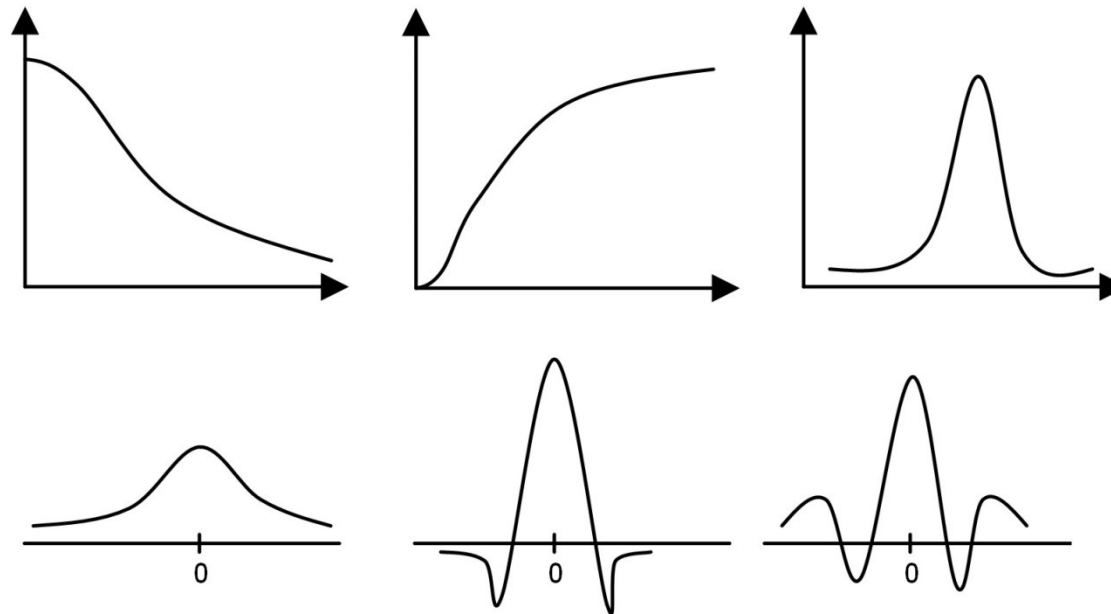


d) Robert

e) Sobel

Exemplo de aplicação de diferentes operadores de detecção de borda.

Filtros passa-banda ou elimina faixa em 1D



Filtros passa-baixa, passa-alta e passa-faixa em frequência (acima) e no espaço (abaixo).

Trabalho

- Cada um dos alunos deve tentar usar nas imagens exemplos deste ano os filtros passa alta no domínio espacial disponível na ferramenta que esta usando para ver qual daria mais contraste e verificaria melhor as bordas das imagens de raio x que estamos usando no curso
- Com o objetivo de deixar as imagens mais realçadas antes de serem binarizadas (por ex.: Sobel, Roberts, Laplaciano do Gaussiano, Bilinear, Canny, Mar-Hildret, etc.)
- Alias, vocês tem acesso a Biblioteca Virtual Pearson?

Filtro alto reforço

Uma imagem filtrada por um *passa alta* também pode ser considerada com resultado da retirada na **imagem original** do seu resultado filtrado por um filtro **passa baixa**:

$$\text{Passa-Alta} = \text{Original} - \text{Passa-baixa}$$

Pode-se então pensar em ampliar algum aspecto da imagem antes de retirar dela os elementos de passa alta:

$$\text{Passa-Alta com Reforço} = a \text{ Original} - \text{Passa-Baixa}$$

Onde a é um fator de amplificação.

Filtro alto reforço (cont)

No filtro “alto reforço” ou técnica de ênfase em alta frequência tem-se:

$$\begin{aligned}\text{Alto-Reforço} &= (a) \text{ Original} - \text{Passa-baixa} \\ &= a \text{ Original} - (\text{Original} - \text{Passa-Alta})\end{aligned}$$

$$\text{Alto-Reforço} = (a - 1) \text{ Original} - (-\text{Passa-Baixa})$$

$$\text{Alto-Reforço} = (a-1) \text{ Original} + \text{Passa-Alta}$$

Onde $a \geq 1$ é um fator de amplificação.

Unsharp masking

Na *unsharp masking* usa-se 2 constantes: *a* e *b*, *a* amplifica a imagem original e *b* a resultante do passa baixa. Sendo *a > b* frequência tem-se:

$$\begin{aligned} \textit{unsharp masking} &= a \text{ Original} - b \text{ PassaBaixa} \\ &= a (\text{PassaBaixa} + \text{PassaAlta}) - b \text{ PassaBaixa} \\ &= (a - b) \text{ PassaBaixa} + a (\text{PassaAlta}) \\ &= c \text{ PassaBaixa} + a \text{ Passa-Alta} \end{aligned}$$

Onde *a* ≥ 1 é um fator de amplificação e *c* > 0 são obtidas por experimentação .

FILTRO DE CANNY

Criado por John F.Canny em 1986

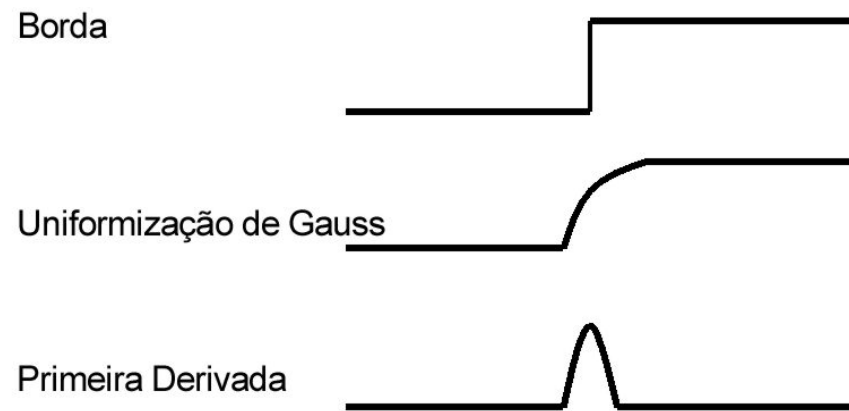
usa um algoritmo multi-passos para detectar a possibilidade de bordas

um filtro de convolução que usa a primeira derivada

suaviza o ruído e localiza bordas, combinando um operador diferencial com um filtro Gaussiano.

Canny

Filtro de convolução de primeira derivada, que suaviza o ruído e localiza bordas, combinando um operador diferencial com um filtro Gaussiano.



Processo de detecção de bordas por Canny

Funcionamento da detecção de Borda

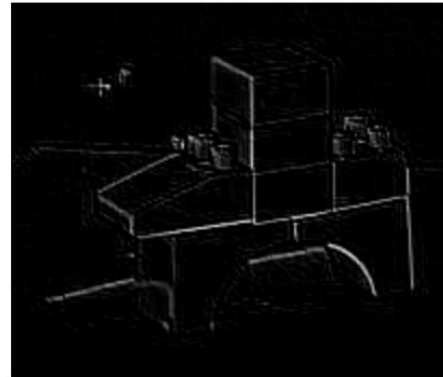
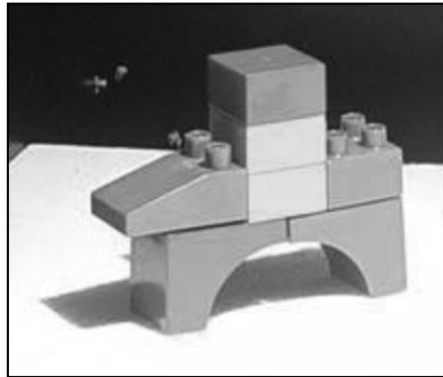
Pontos de máximo indicam a borda da imagem

Pode-se realizar este processo usando uma função Gaussiana em duas dimensões, ou uma função 1D na direção de x e depois na direção y .

As formas das máscaras Gaussianas dependem da escolha do desvio padrão, σ , usado, ou seja, o valor de sigma na equação da função gaussiana.

Convolvendo a imagem com $G'(x)$ obtém-se uma imagem I que mostrará as bordas, mesmo na presença de ruído.

$$G'(x) = \frac{-x}{\sqrt{2\pi\sigma^3}} \mathcal{E}^{\frac{-x^2}{2\sigma^2}}$$



Resultado da aplicação do filtro Canny (b) a imagem original (a).

Funcionamento da detecção de Borda

Função Gaussiana de uma dimensão:

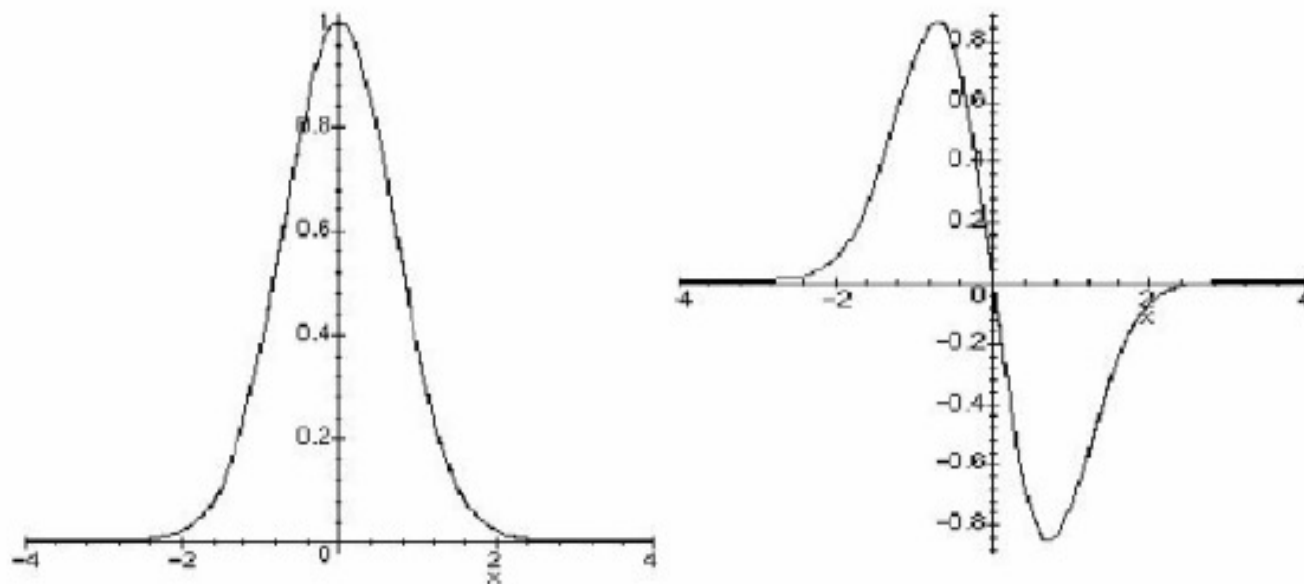
$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

e sua primeira derivada é:

$$G'(x) = \frac{-x}{\sqrt{2\pi\sigma^3}} e^{-\frac{x^2}{2\sigma^2}}$$

Funcionamento da detecção de Borda

A forma destas duas expressões, isto é a Gaussiana e sua derivada em 1D, no gráfico foi usado um desvio padrão que tornasse o valor máximo 1.0 , isto é usando $\sigma^2 = 1/(2\pi) \approx 0,1592$



Funcionamento da detecção de Borda

O Filtro de Canny para detecção de bordas é usar $G'(x)$

a operação de convolução da imagem com $G'(x)$ obtêm-se uma imagem I que mostrará as bordas, mesmo na presença de ruído.

A convolução é relativamente simples de ser computada, mas é custosa computacionalmente

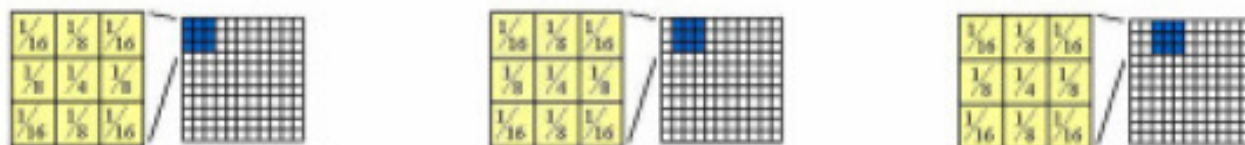
$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}}$$

Funcionamento da detecção de Borda

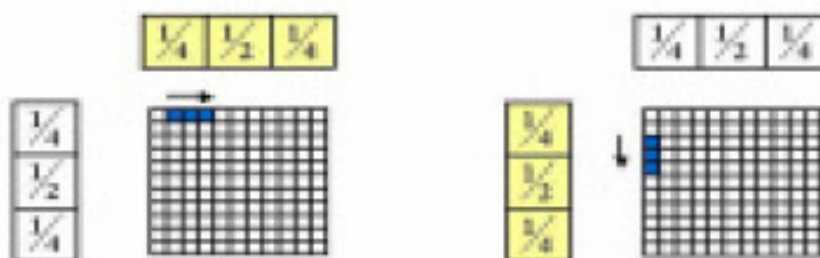
Gaussianas 2D dependem da escolha dos desvios padrão nas duas direções, sendo geralmente usando em imagens ambos iguais σ , ou seja, o mesmo valor de sigma em na equação $\sigma = \sigma_x = \sigma_y$ da função Gaussiana.

Funcionamento da detecção de Borda

- Etapas da passagem do filtro Gaussiano 2D com $\sigma^2 = 2/\pi \approx 0,6366$, ou seja $\sigma = 0,8$



- Etapas da passagem do filtro Gaussiano 1D com $\sigma^2 = 2/\pi \approx 0,6366$, ou seja $\sigma = 0,8$



Funcionamento da detecção de Borda

Todo valor situado acima de determinado limite superior e rejeitando todo valor situado abaixo de um limite inferior.

Pontos situados entre os dois limites serão aceitos como de borda se eles estiverem relacionados fortemente com os pixels.

Custo computacional do detector de bordas de Canny é relativamente alto.

Os resultados são geralmente pós-processados para maior clareza.

O algoritmo é bem eficiente no processamento de imagens com ruídos ou com bordas difusas.

Funcionamento da detecção de Borda



Algoritmo de Canny

O alvo do algoritmo de Canny é obter um método de detecção ótima da borda. Uma detecção ótima da borda significa que:

- o algoritmo deve marcar tantas bordas quanto possível;
- as bordas marcadas devem estar tão perto quanto possível da borda real;
- cada borda na imagem deve ser marcada uma vez, e o ruído da imagem não deve criar bordas falsas.

Usa cálculo variacional para encontrar a função que satisfaz melhor a um funcional dado.

Detector ótimo é descrito pela soma de quatro termos exponenciais, mas aproximado pela primeira derivada de uma Gaussiana.

Etapas do Algoritmo

Redução de ruído - Começa-se convoluindo a imagem uma máscara Gaussiana. A saída do primeiro passo é uma versão ligeiramente mais desfocada da original. Isto conseqüentemente reduz o efeito de pixels ruidosos na imagem.

Calculo dos gradientes da intensidade da imagem - uma borda em uma imagem pode estar em vários sentidos. Assim 4 máscaras são usadas para detectar bordas horizontais, verticais e diagonais (45 e 135 graus).

Etapas do Algoritmo

Os resultados da convolução da imagem original com cada uma destas máscaras são armazenados. Para cada pixel, marca-se então o maior resultado do gradiente nesse pixel, e o sentido da máscara que produziu essa borda. Da imagem original, cria-se dois mapas de gradientes da intensidade em cada ponto na imagem, e o do sentido do gradiente no ponto.

Detecção das bordas da imagem - os gradientes de intensidade mais elevada são mais prováveis de serem bordas. Mas não há um valor absoluto pré-definido de quanto uma dada intensidade passa a ser uma borda. A ideia de Canny é usar 2 limiares para isto (thresholds): um superior e outro inferior.

Etapas do Algoritmo

- Thresholding com histeresse requer dois limiares iniciais um superior outro inferior
- Supõe que bordas são linha continuas, linhas de pouca intensidade são investigadas, evita pixels que não constituem linhas
- Assim aplica-se o primeiro limiar elevado.
- Marca as bordas que podem ser genuinas.
- A partir dessas, usa informação direcional para encontrar as bordas da imagem.
- Ao seguir uma linha, usa o limiar inferior para permitir seguir fracas possibilidades de bordas a partir do ponto inicial

Etapas do Algoritmo

- Ao final do processo tem-se uma imagem binária onde cada pixel é marcado como:
 - pixel da borda
 - pixel da não-borda

Parâmetros ajustáveis

- Tamanho do filtro Gaussiano
 - filtros menores causam menos blur e detectam linhas pequenas e mais nitidas
 - filtros maiores causam mais blur e é mais util para detectar bordas maiores mais suaves
- Limiares
 - limiares altos causam perda de informação
 - limiares baixos causam identificação falsa a ser considerada importante

Filtros **HOMOMÓRFICOS**

- se os pesos $h[i,j]$ forem constantes o filtro é chamado **HOMOMORFICO** [6]:

$$g[x, y] = \sum_{i=1}^n \sum_{j=1}^m f[x-i, y-j] \cdot h[i, j]$$



Filtragem Homomórfica

- When an image with large dynamic range is recorded on a medium with a small dynamic range, its contrast is significantly reduced, particularly in its darker and brighter areas.

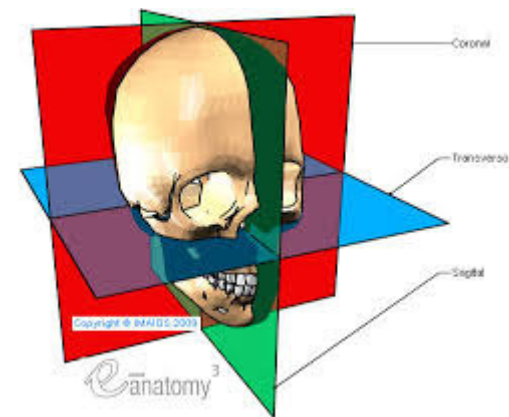
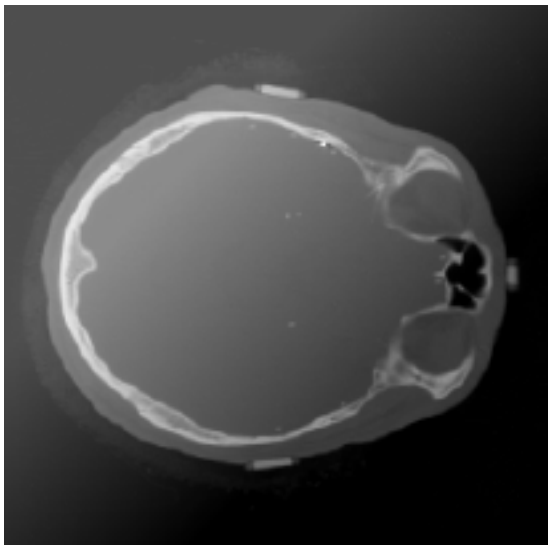
$$f(n_1, n_2) = i(n_1, n_2)r(n_1, n_2)$$

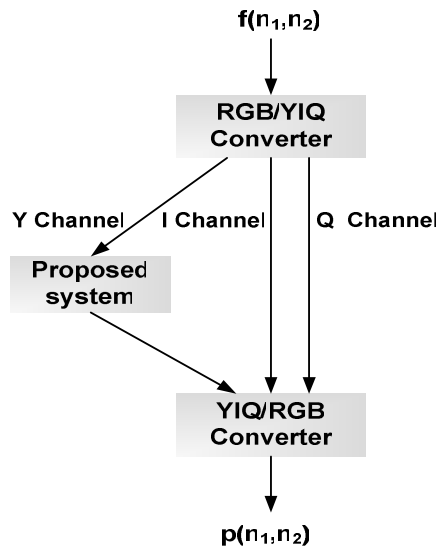
$$\log(f(n_1, n_2)) = \log(i(n_1, n_2)) + \log(r(n_1, n_2))$$

Note:

- if the tonal level quantization is from zero, then $\log(1+f(n_1, n_2))$ is the transformation used to separate the illumination and reflectance components.

Pode corrigir a iluminação inadequada antes de uma limiarização

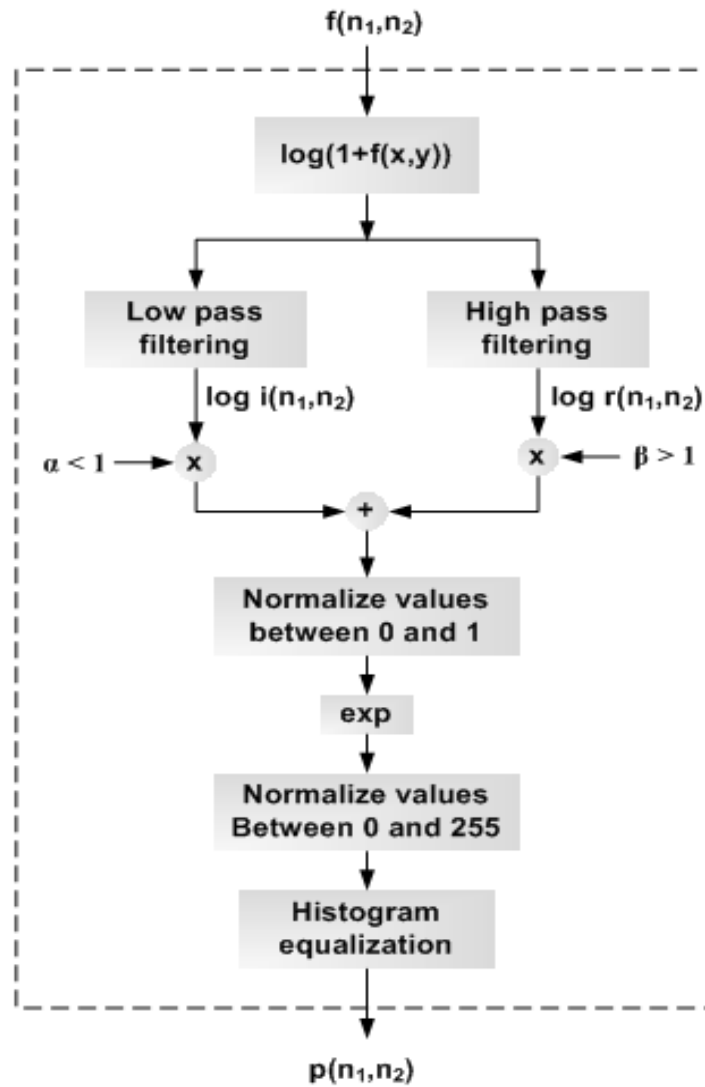




-1	-1	-1
-1	9	-1
-1	-1	-1

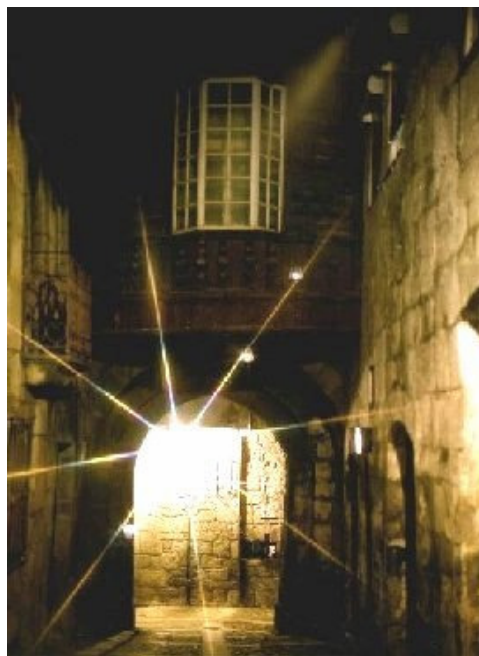
$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1



exemplos

- Original Equalizada homomorfo

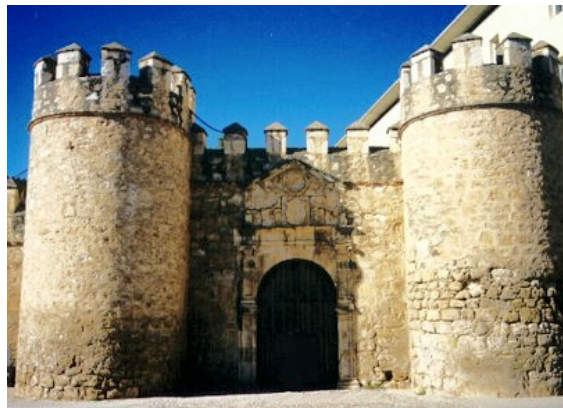


exemplo

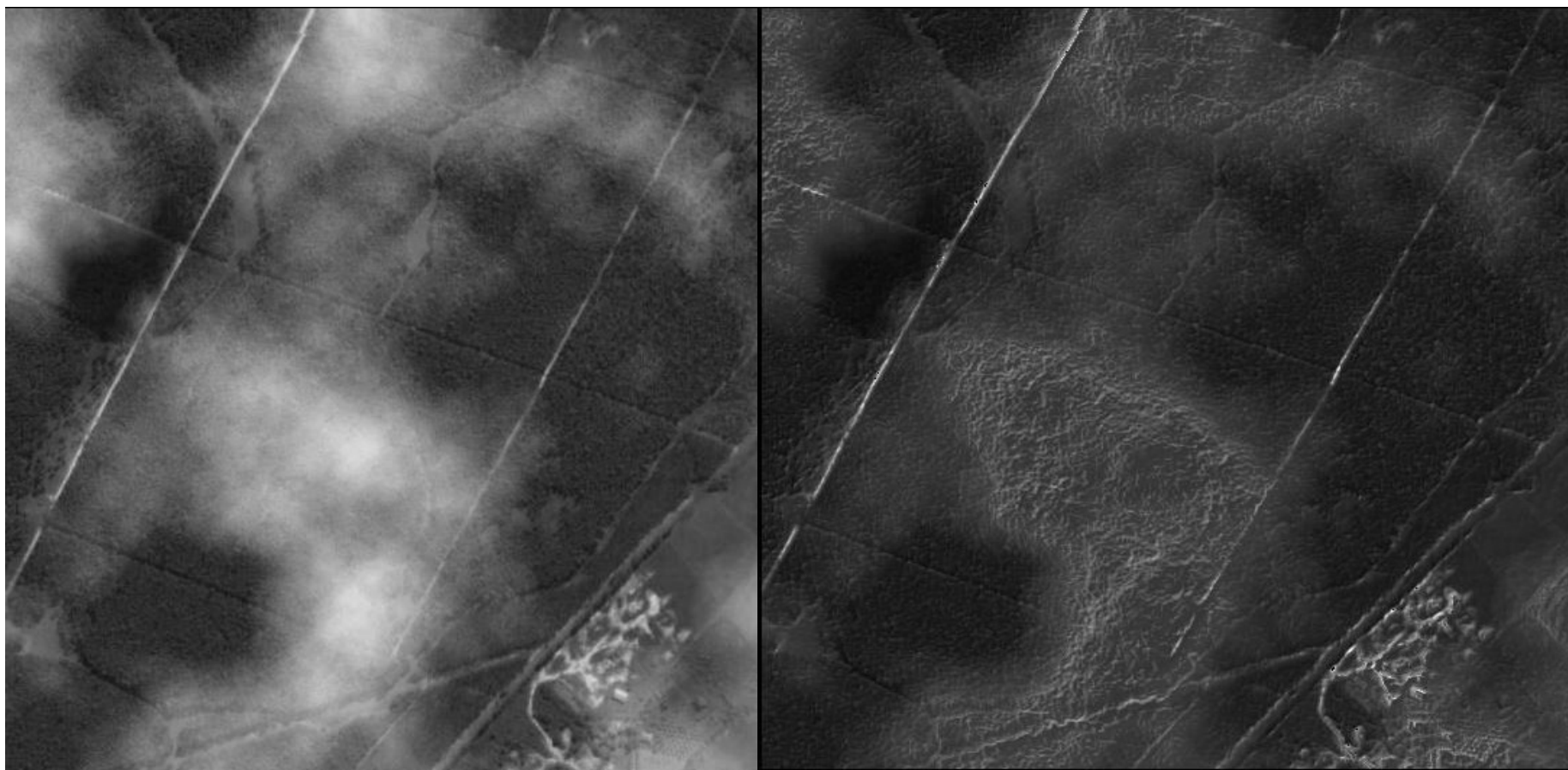
Original

X

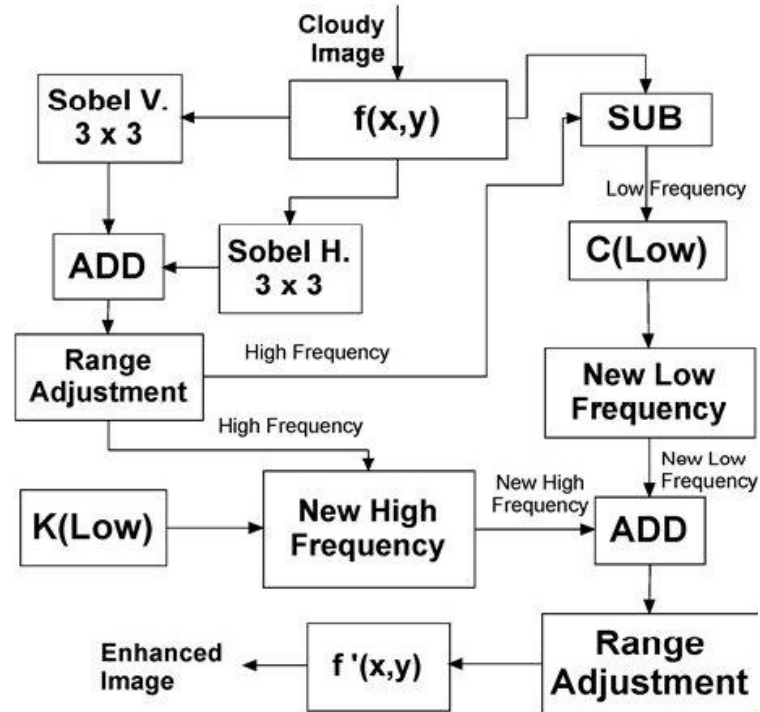
processada



Exemplo: **Diminuir efeito de nuvens**

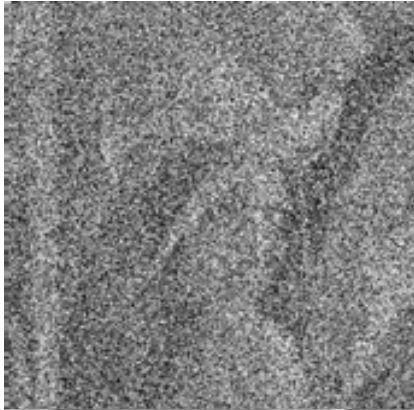


Esquema usado



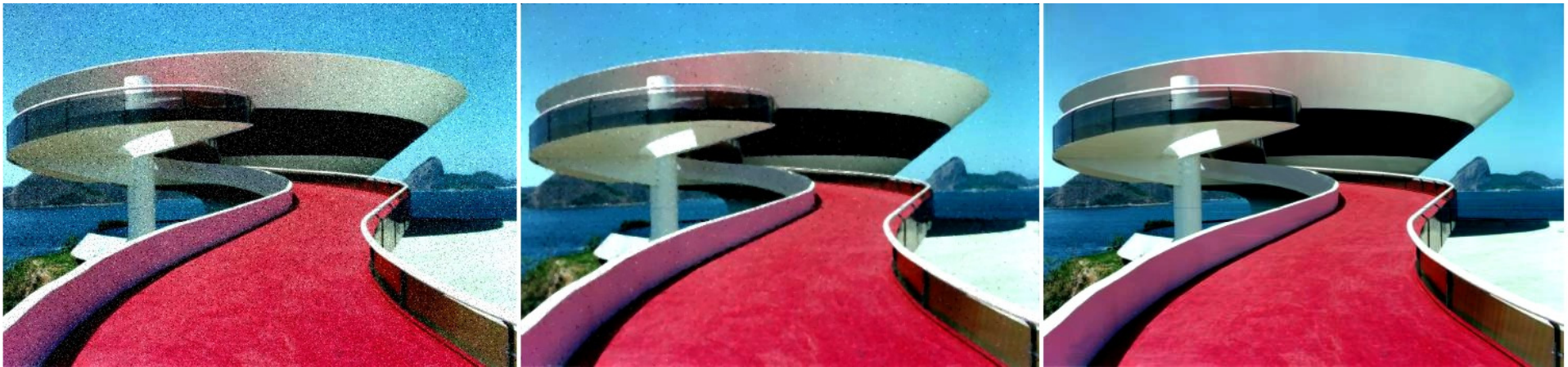
- J.G.F.M. GAZOLLA , E.G. CLUA , A. CONCI *An Adaptive System to Diminish the Influence of Clouds in Satellite Images for Texture Segmentation*
- <http://www.matmidia.mat.puc-rio.br/sibgrapi2009/media/posters/59792.pdf>

Imagens **Lena** e **Diamantina** com 70% de ruído salt & pepper



- Filipe Xavier, R. C. Leal Toledo, A. Conci, *Image Noise Reduction Based on Cellular Automata Filter*, Simpósio Brasileiro de Computação Gráfica, Processamento de Imagens e Visão Computacional, 08-11 de outubro de 2006

Em todas as operações feitas até aqui o *pixels* era caracterizado pela posição (x_0, y_0) na *imagem origem*,
Mas seu tom era alterado



(a)

(b)

(c)

Redução de ruídos na imagem. (a) Imagem com ruído. (b) Redução do ruído usando filtro de *media* (*perde um pouco da nitidez = blur*). (c) Imagem original.

Operações Globais

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

Transformações geométricas e as diversas Transformadas (mudam a forma de representação do espaço de representação 2D para outras formas como as: Transformadas Fourier, Transformadas Cosenos (usada para codificação); Transformadas Gabor, Transformadas Wavelet, Transformadas Hough, Funções iterativas (IFS) ou fractal, Etc..

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, etc.

Transformações Geométricas

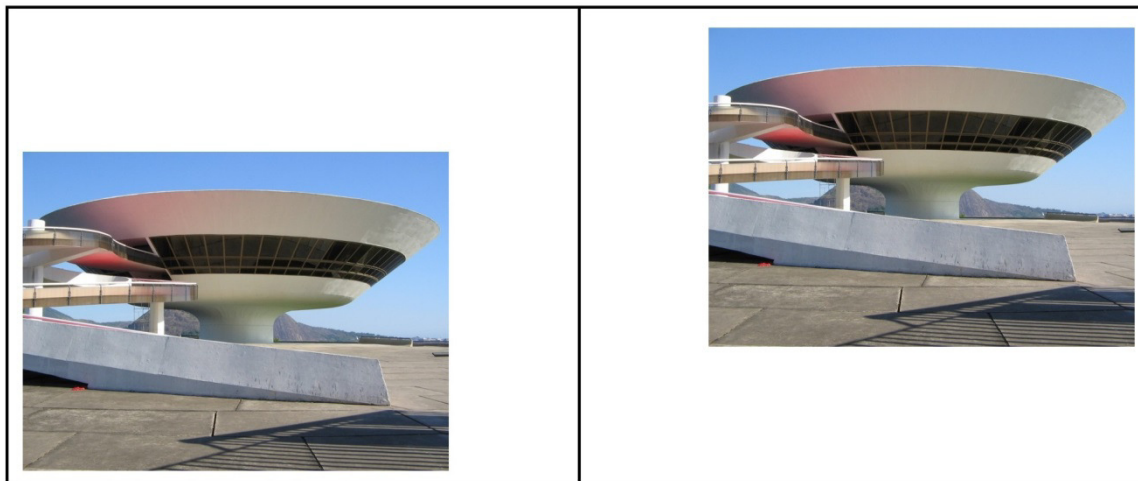
o *pixels* muda a posição (x_0, y_0) na *imagem origem*,
Mas é considerado o mesmo pois seu tom fica inalterado

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

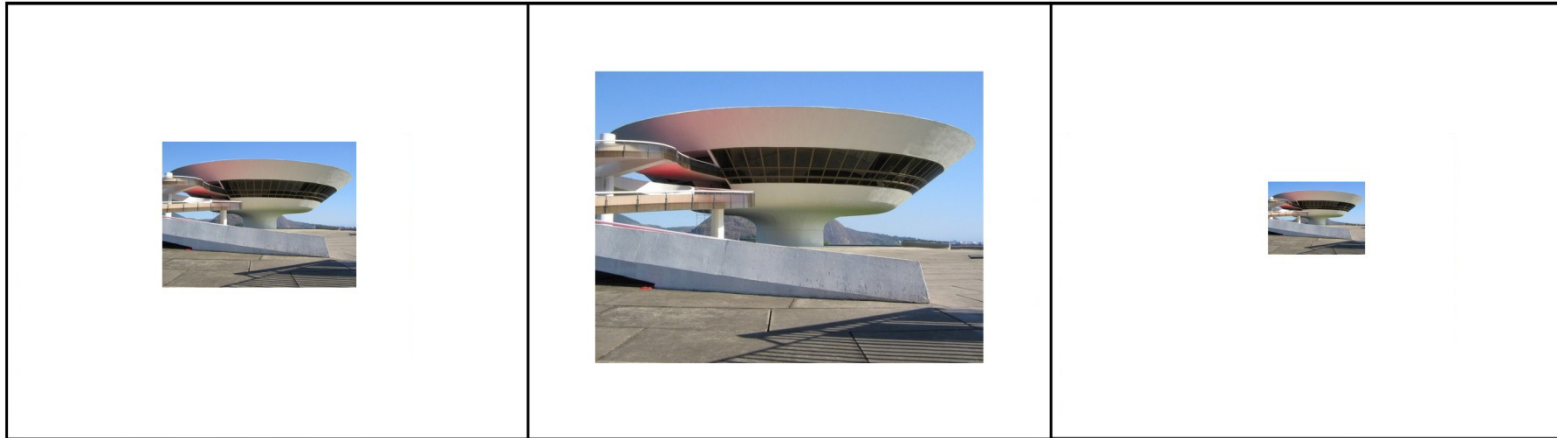
$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

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° no sentido horário

Rotação em duas passadas:

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

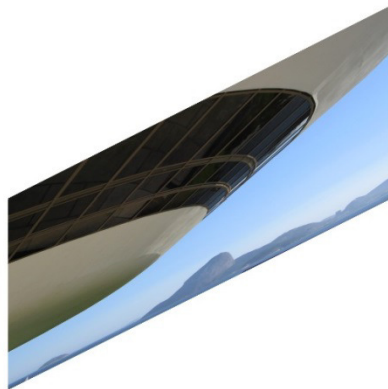
$$y_d = x_o \operatorname{sen} \theta + y_o \cos \theta$$

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

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



(a)



(b)



(c)

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

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.

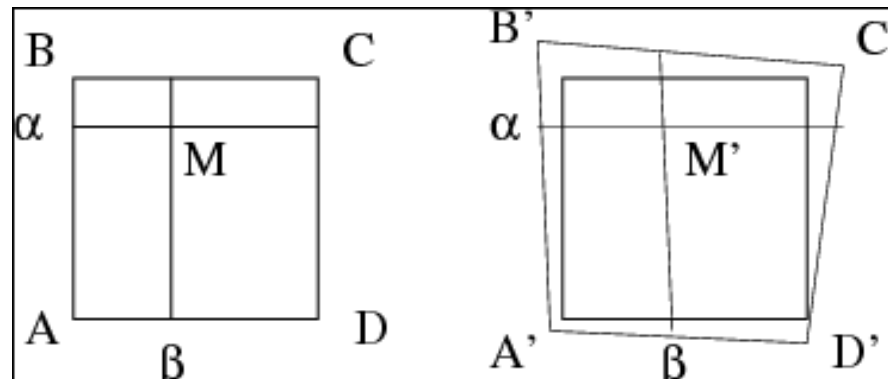
Manipulações básicas de Re-amostragem

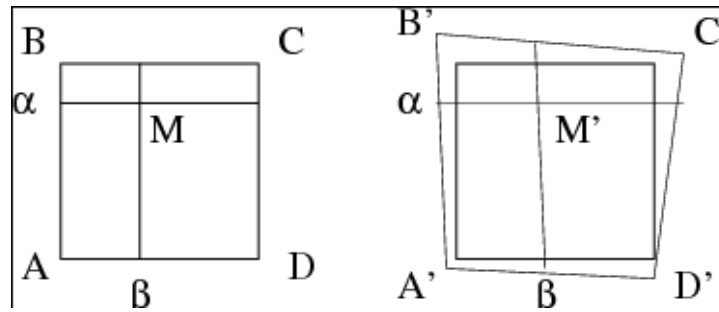
- São geralmente feitas por *interpolação bilinear*.
- Essa transformação é encontrada nos softwares de imagem existentes (*transformação de imagens elementar*).

- Se $A, B, C,$ e D são vértices de uma imagem, ou parte dela, então
- um ponto M da imagem pode ser expresso por

- $M = a (b A + (1 - b) D) + (1 - a) (b B + (1 - b) C) ,$

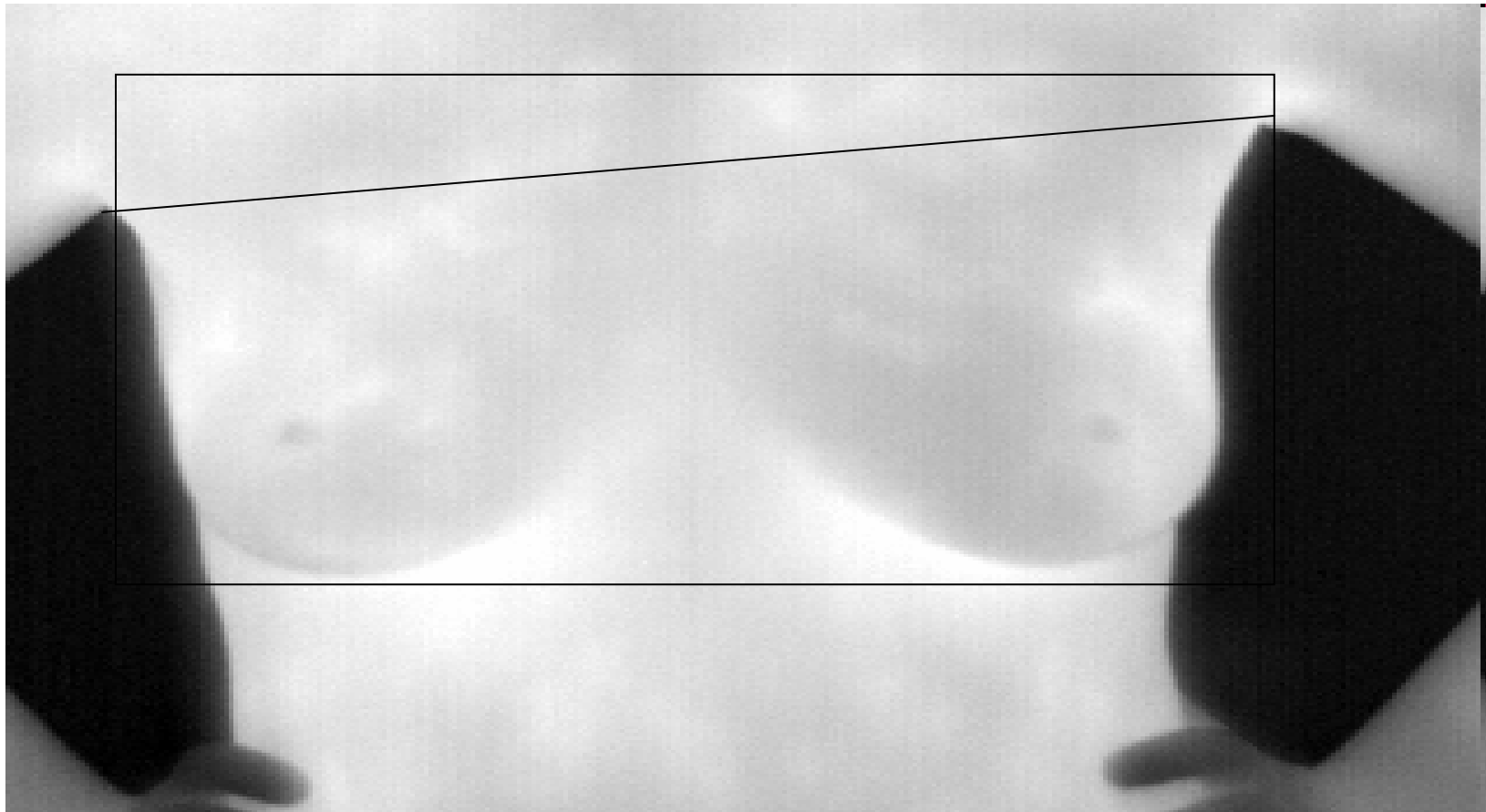
- onde $0 \leq a, b \leq 1$ são coordenadas de M (ou *distância normalizada*) relativas ao vértice A .





- A distorção pode ser inserida movendo-se os vértices por uma quantidade aleatória pequena em ambas as direções.
- As novas coordenadas de M são dadas pela fórmula prévia, mantendo-se (a e b) constantes mas utilizando os novos vértices: $A, B, C,$ e D
- Note que se esta transformação for *inversível*, conseqüentemente ela não eliminará dados.

A etapa de extração da ROI ficará obviamente ruim pois a paciente esta “torta” . Por transformação linear pode-se fazer as duas axilas e a extremidade inferior dos seios ficarem alinhados na mesma linha reta.

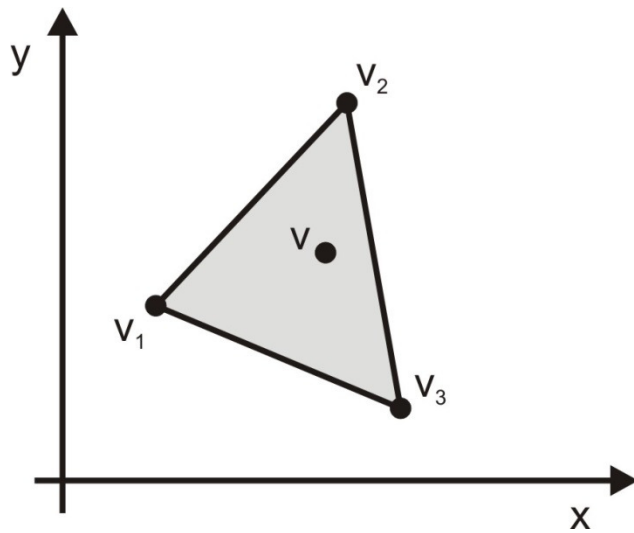


Deformações e *Morphing*

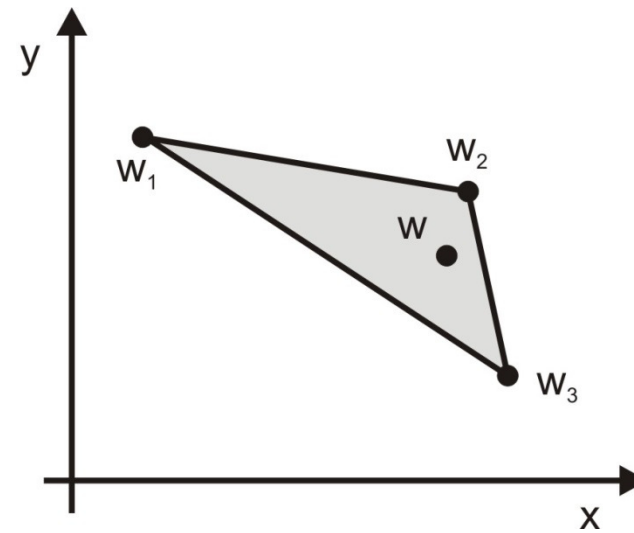
Deformação: considera que a imagem está envolvida por segmentos de reta com seus respectivos vértices e altera **a forma** dos objetos a partir de mudanças de posição dos segmentos envolventes.

Morphing: considera o processo de **deformação** aliado à decomposição de suas **cores**.

Deformações

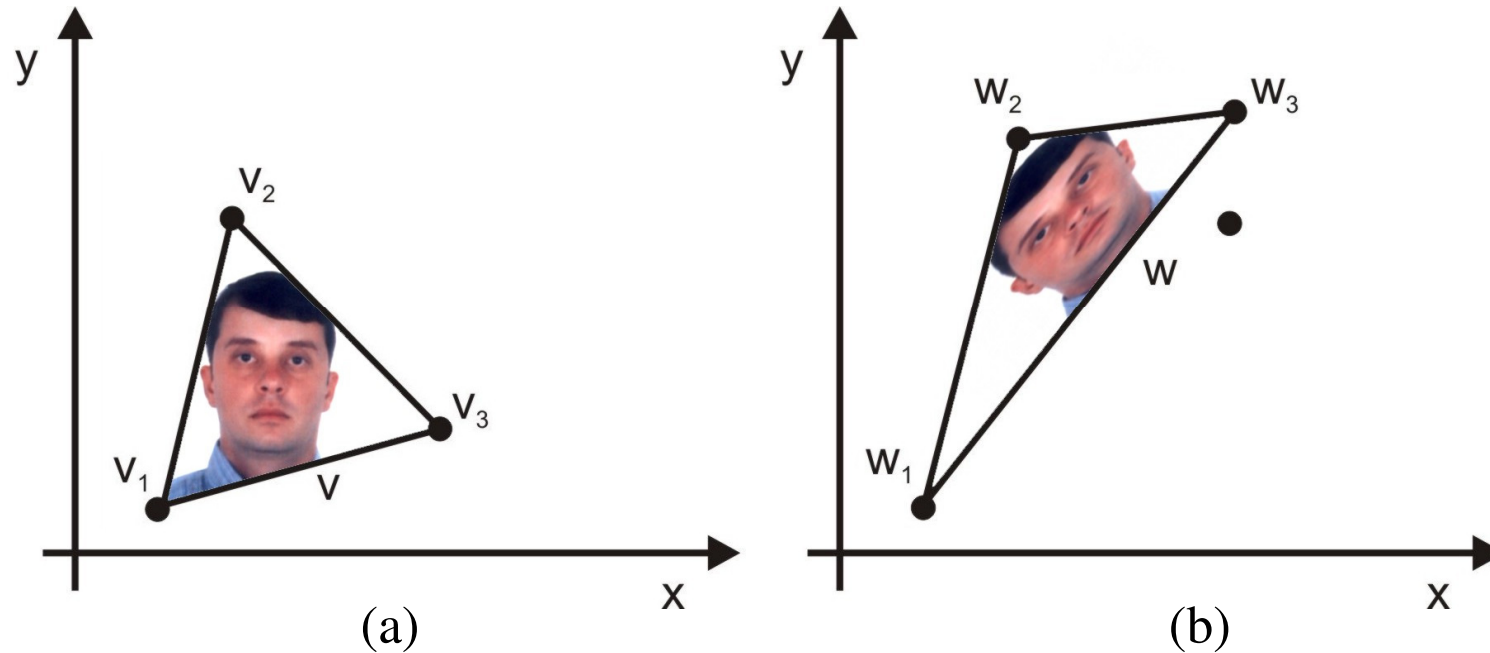


(a)



(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: w_1, w_2, w_3 e v_1, v_2, v_3

6 valores conhecidos: w_1, w_2, w_3 e v_1, v_2, v_3

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

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

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

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

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

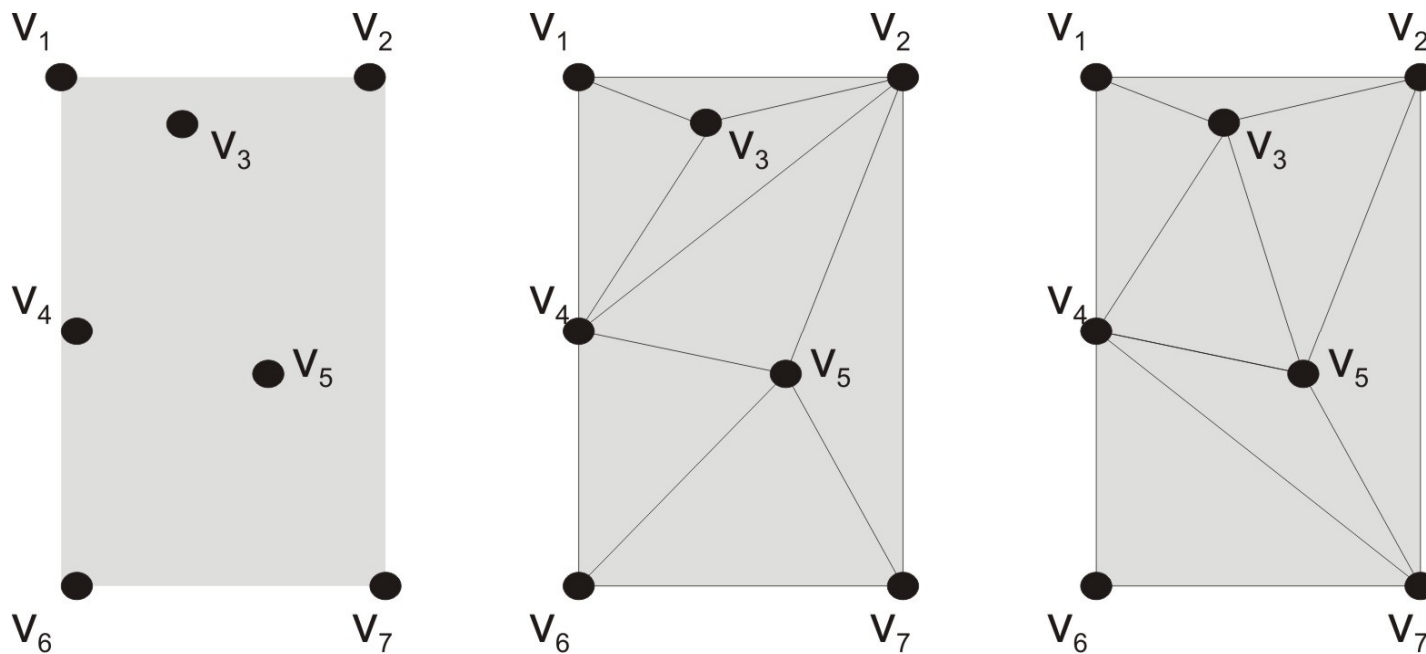
6 incognitas

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

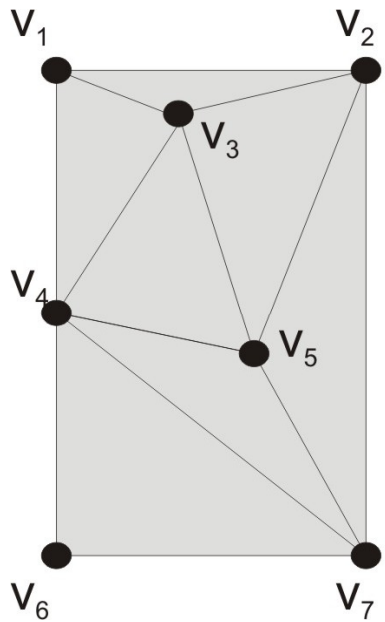
$$w_{3y} = m_{21} v_{3x} + m_{22} v_{3y} + b_2$$

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

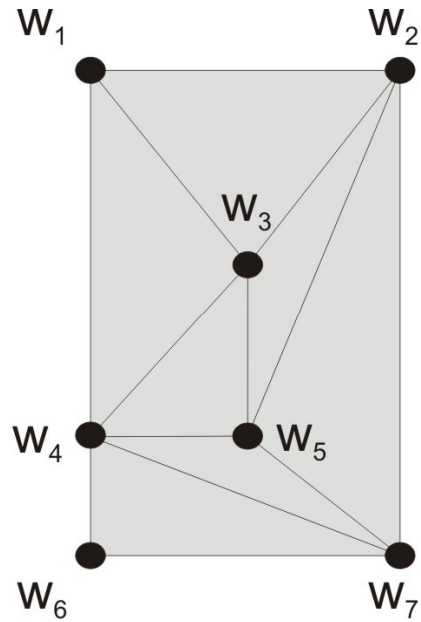
Pode-se repartir uma imagem em várias regiões triangulares e deformar cada região de uma maneira diferente:



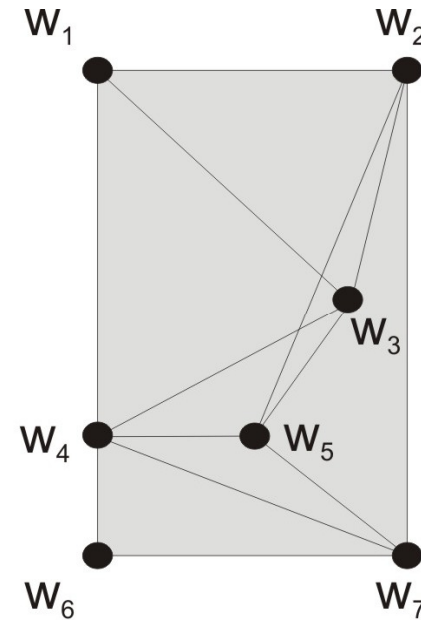
Fazendo Triangulações



(a)



(b)



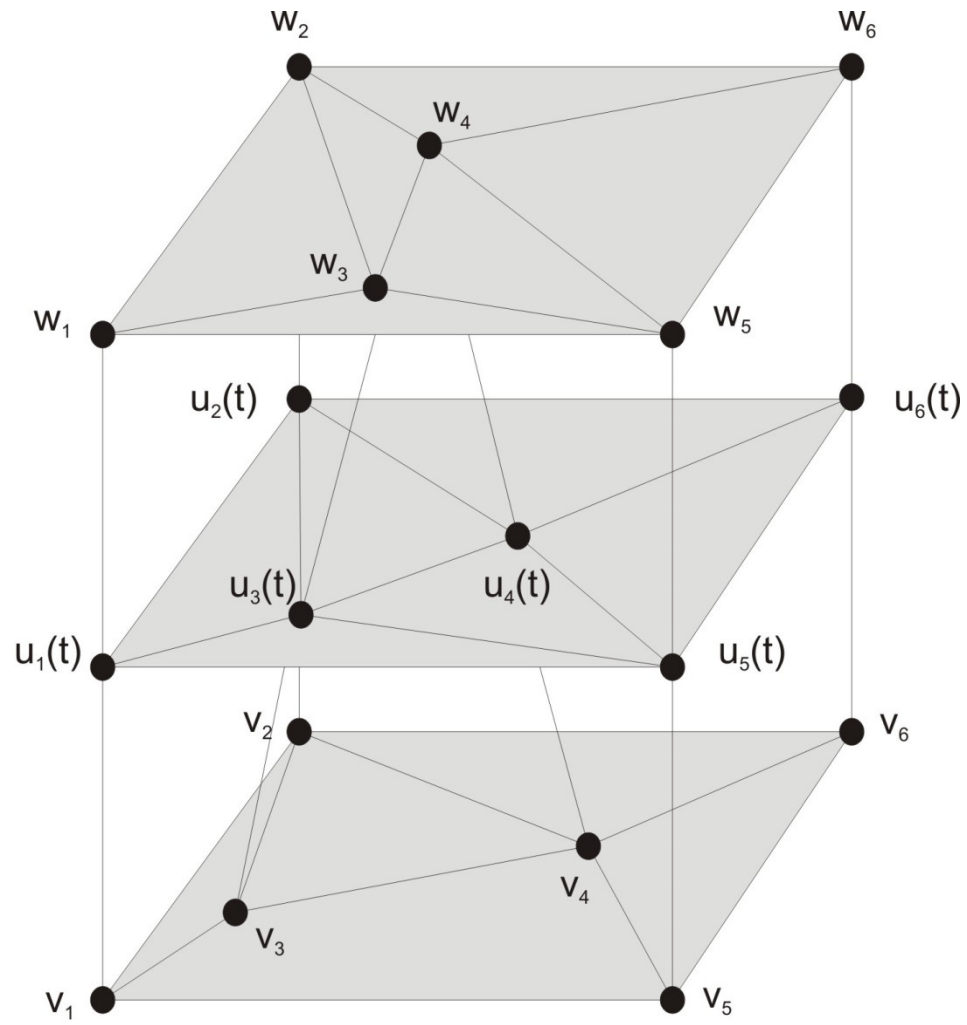
(c)

Deformações por movimento de pontos dos vértices permitida (b) e não permitida (c)

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.

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



Triangulação dependente do tempo de uma região

Morphing

O termo *morph* tem como origem a palavra grega *morhos* que significa **forma**. A ciência que estuda as formas chamada de *Morfologia*. *Morphing* é uma redução da palavra *metamorfose*.

O *morphing* envolve simultaneamente dois tipos de transformação:

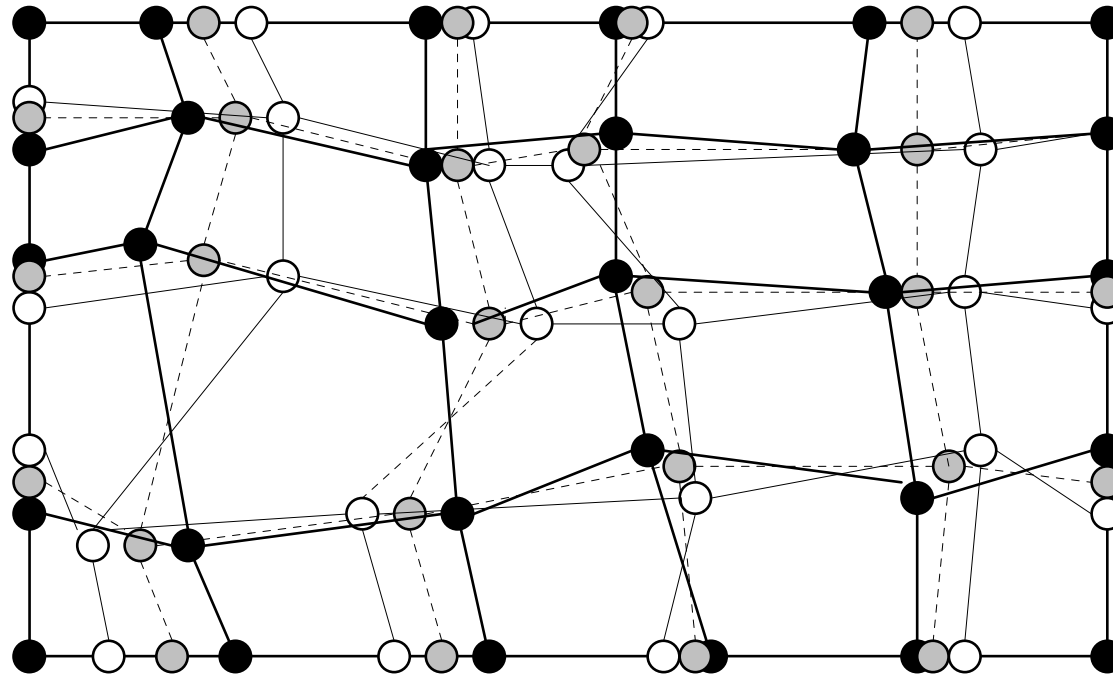
(a) de **deformação** (*warping*);

(b) de **tons e cores** (*cross-dissolve* ou decomposição cruzada).

$$\text{ново_red} = (r_origem + r_destino) / 2$$

$$\text{ново_green} = (g_origem + g_destino) / 2$$

$$\text{ново_blue} = (b_origem + b_destino) / 2$$

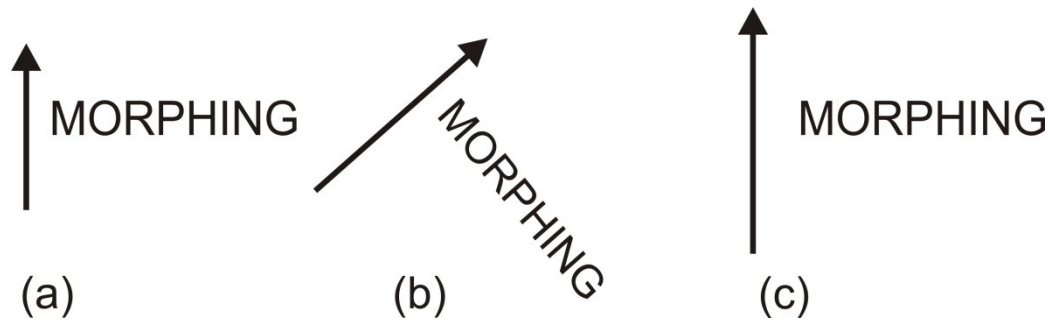


—● Origem

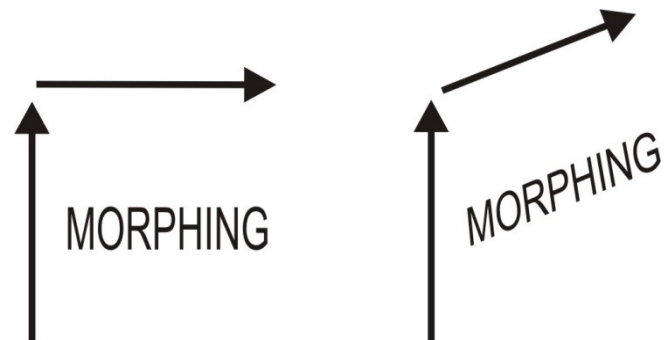
—○ Destino

---● Intermediária

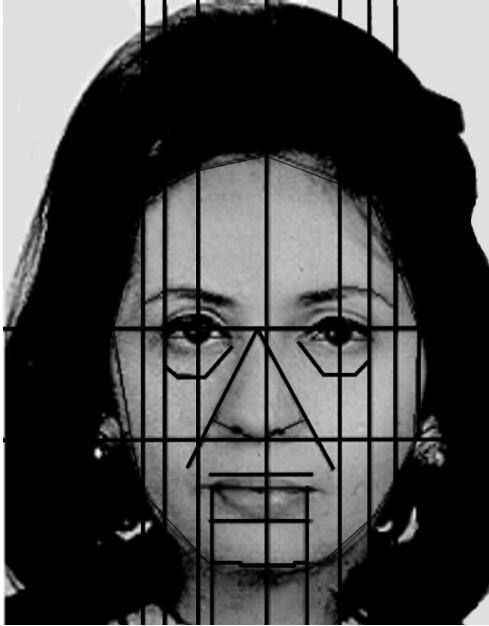
Transformação de *pixels* origem em destino.



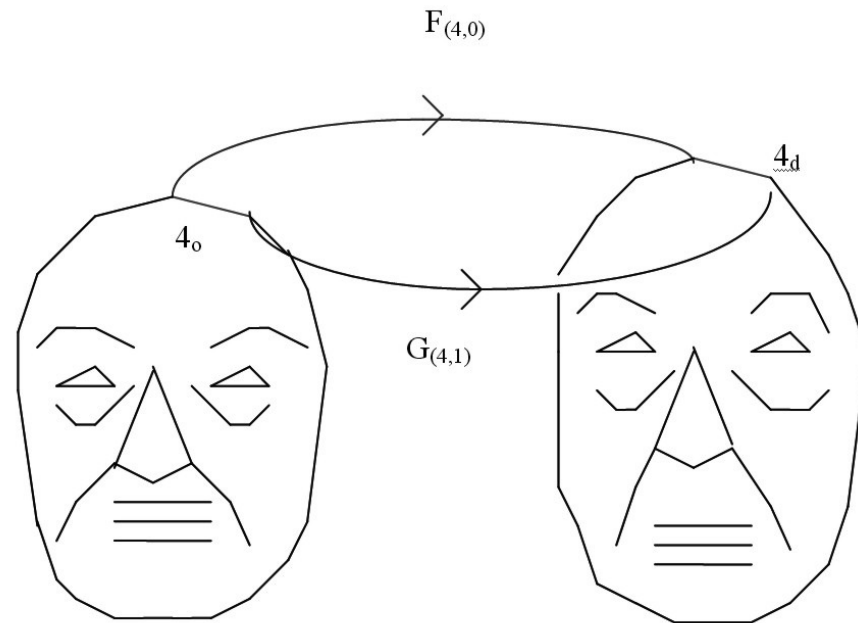
Efeitos decorrentes de alterações em uma **única linha de controle**.



Efeito de **duas linhas de controle**.



Segmentos de Controle e linhas auxiliares.



Exemplo de interpolação de segmentos.

Para entender mais experimente :

- Entrar no site do HIPR2:
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/pntops.htm>
- Ler sobre as operações pontuais, e exemplos que são comentados lá.
- Você acha que algo pode ser aplicada ao trabalho.
- Ainda veja trabalhos deste curso de anos anteriores como as perguntas 1 a 3 do estudo dirigido do cap. 9 do site do nosso curso:
<http://www.ic.uff.br/~aconci/estdirigidocap9.pdf>

Bibliografia

1. Burgiss Jr., S. G., Goodridge, S. G. (2001) 'Multiframe Averaging and Homomorphic Filtering for Clarification of Dark and Shadowed Video Scenes', SPIE vol. 4232 - 480.
2. Etemadnia, H. and Alsharif, M. R. (2003) 'Automatic Image Shadow Identification using LPF in Homomorphic Processing System'. Proceedings of the VIIth Digital Image Computing: Techniques and Applications, Sydney, Dec.
3. Lim, J. S. (1990) 'Two-Dimensional Signal and Image Processing', Prentice Hall Signal Processing Series, New Jersey, 880 p.
4. Melo, R. H. C., Vieira, E. A., Touma, V. L. and Conci, A. (2005) 'Sistema de realce de detalhes ocultos em imagens com grande diferença de iluminação fazendo uso de filtragem não-linear', in: J.M. Balthazar et al. (Eds) Proceedings of 4th Dincon (Bauru, 6-10. June), SP Bauru, Brazil, 2005.
5. [Milan Sonka](#), [Vaclav Hlavac](#), [Roger Boyle](#) , **Image Processing, Analysis and Machine Vision**, Thomson, 2008. 3rd Edition
<http://user.engineering.uiowa.edu/~sonka/ps-files/cfai.pdf>
6. [Linda G. Shapiro](#) , [George C. Stockman](#), **Computer Vision**, Prentice Hall, 2001.
7. <ftp://91.193.236.10/pub/docs/linux-support/computer%20science/computer%20vision/Computer%20Vision%20-%20Linda%20Shapiro.pdf>

Bibliografia Complementar

Mapeamentos e Compressão de Imagens:
..../~aconci/curso/cap3.pdf

<http://www.ic.uff.br/~aconci/Fractais.html>

Livros em pdf ?

R. C. Gonzalez and R. E. Woods - **Digital Image Processing**, Addison Wesley Pub. Co. 1993 - ISBN 0-201-60078-1. UFF Bib CTC.: 006.42 G 643 - 1993