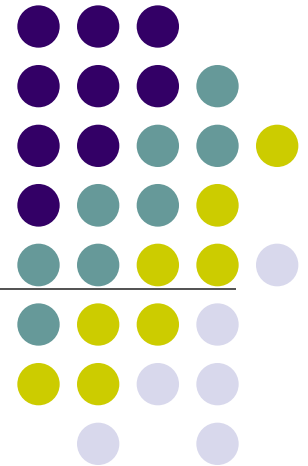


# Processamento de Imagens – Parte III

29.Junho.2016

Prof. Celso Kurashima

ESZI009 – Processamento de Vídeo



# Segmentação de Imagem

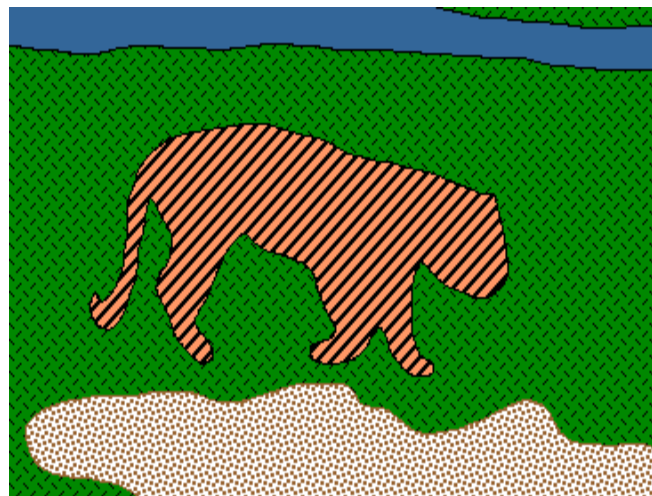
---



From [Sandlot Science](#)

# De imagens para objetos

---



## O que define um Objeto?

- Problema Subjetivo, mas tem sido bem estudado
- Leis Getalt procuram formalizar isto
  - proximidade, similaridade, continuação, fechamento, fato comum
  - [notes](#) by Steve Joordens, U. Toronto

# Segmentação de Imagem

---

Podemos considerar diferentes métodos

Manual:

- *Intelligent Scissors* (baseado em contorno, manual)

Hoje—métodos automáticos:

- Clusterização K-means (*color-based*)
- Recortes Normalizados (*region-based*)



# Histogramas de Imagens

---



Quantos pixels “laranja” há nesta imagem?

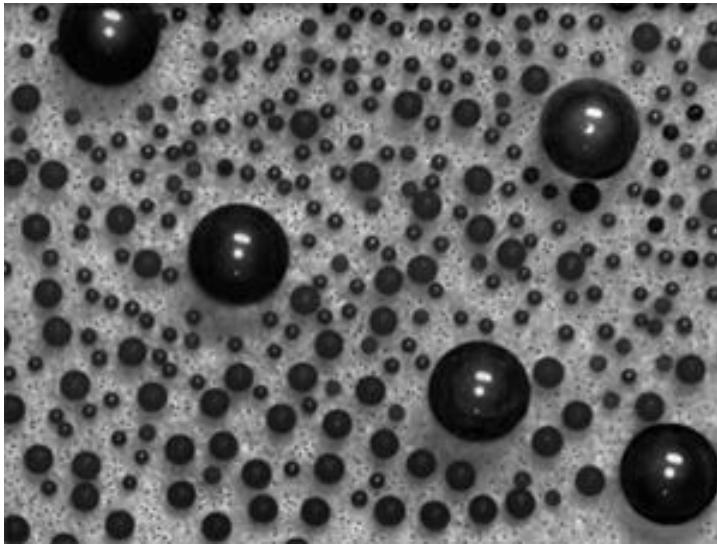
- Este tipo de pergunta é respondida olhando-se o *histograma*
- Um histograma conta o numero de ocorrencia de cada cor
  - Dado uma imagem  $F[x, y] \rightarrow RGB$
  - O histograma é  $H_F[c] = |\{(x, y) \mid F[x, y] = c\}|$ 
    - » i.e., para cada cor de valor  $c$  (eixo-x), plote # de pixels com aquela cor (eixo-y)
  - Qual é a dimensão de um histograma de imagem  $N \times N$  RGB?



# O que os histogramas se parecem?

---

## Demonstração Photoshop



## Quando Modos existem?

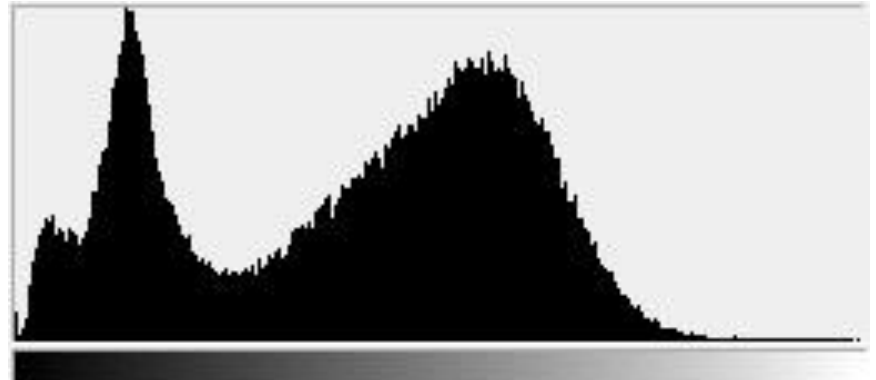
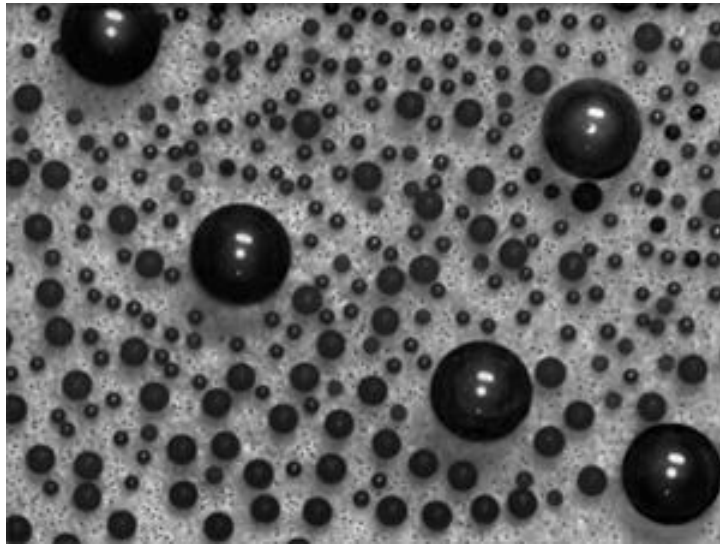
- Fácil de ver, difícil de computar

# Segmentação baseada em Histograma

---

## Objetivo

- Quebrar a imagem em K regiões (segmentos)
- Resolver isto pela redução do número de cores para K e mapear cada pixel para a cor mais proxima
  - photoshop demo



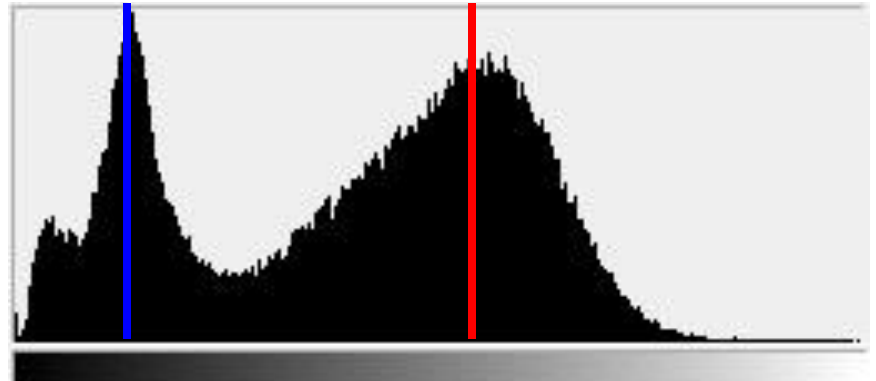
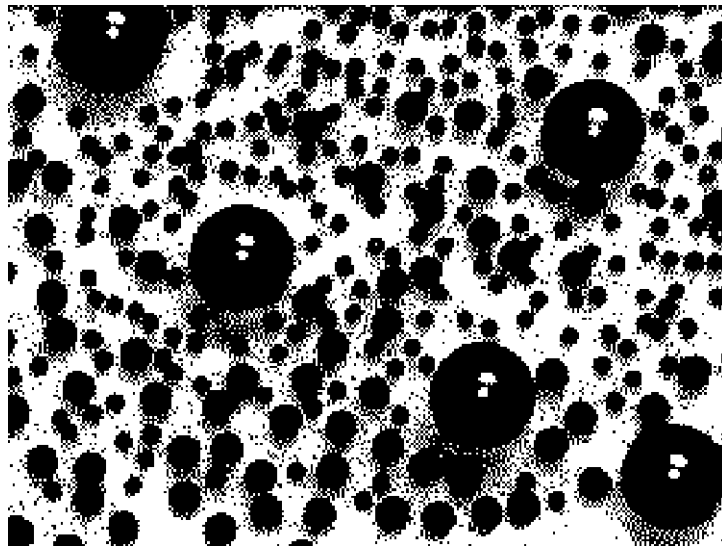


# Segmentação baseada em Histograma

---

## Objetivo

- Quebrar a imagem em K regiões (segmentos)
- Resolver isto pela redução do número de cores para K e mapear cada pixel para a cor mais proxima
  - photoshop demo



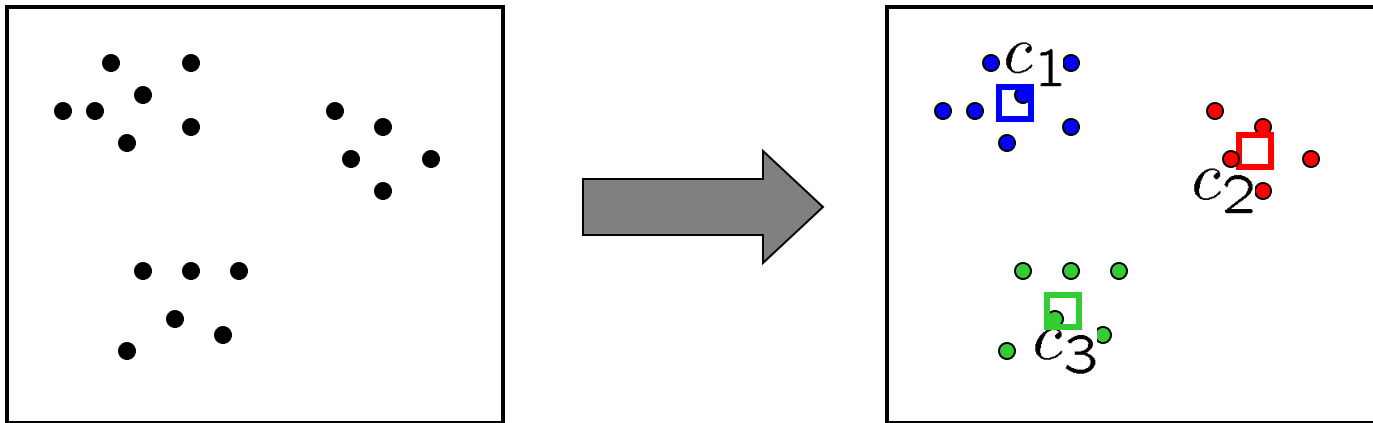
Este é o resultado quando usamos duas cores



# Clusterização

## Como escolher cores representativas?

- Isto é um problema de clusterização!



## Objetivo

- Cada ponto deve estar o mais perto possível de um centro de cluster (agrupamento)
  - Minimize a soma da distancia ao quadrado de cada ponto para o centro mais proximo

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} ||p - c_i||^2$$

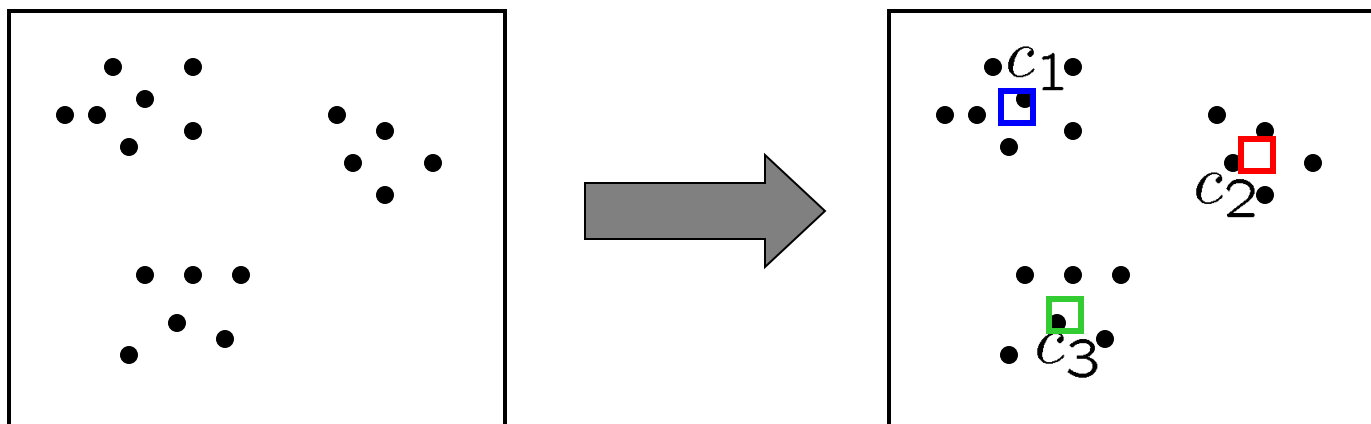


# Quebrar em subproblemas

---

Suponha que os centros dos clusters  $c_i$  são conhecidos

- P: como determinar quais pontos associar com cada  $c_i$ ?
- R: para cada ponto  $p$ , escolha o  $c_i$  mais próximo



Suponha que os pontos de cada cluster são conhecidos

- P: como determinar os centros dos clusters?
- R: escolha  $c_i$  como sendo a média de todos os pontos naquele cluster

# Clusterização K-means

---

## Algoritmo K-means clustering

1. Inicialize aleatoriamente o centro do cluster,  $c_1, \dots, c_K$
2. Dado o centro do cluster, determine pontos em cada cluster
  - Para cada ponto  $p$ , encontre o mais proximo  $c_i$ . Coloque  $p$  no cluster  $i$
3. Dado pontos em cada cluster, resolva para  $c_i$ 
  - Faça  $c_i$  ser a média dos pontos no cluster  $i$
4. Se  $c_i$  teve mudança, repita Passo 2

Java demo: [http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/AppletKM.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html)

## Propriedades

- Sempre converge para *alguma* solução
- Pode ser um “minimo local”
  - Nem sempre encontra o minimo global de uma função objetiva:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$



# Clareando o resultado

---

## Problema:

- Segmentação baseada em Histograma pode produzir regiões sujas
  - segmentos não são necessariamente conectados
  - Pode conter buracos

Como isso pode ser resolvido?

# Operador Dilation: $G = H \oplus F$

Assume-se:  
imagem binária

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	0	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

1	1	1
1	1	1
1	1	1

$H[u, v]$

Dilatação:  $H$  “sobrepõe”  $F$  ao redor de  $[x, y]$ ?

- $G[x, y] = 1$  se  $H[u, v]$  e  $F[x+u-1, y+v-1]$  são ambos 1 **em algum lugar**  
0 caso contrário

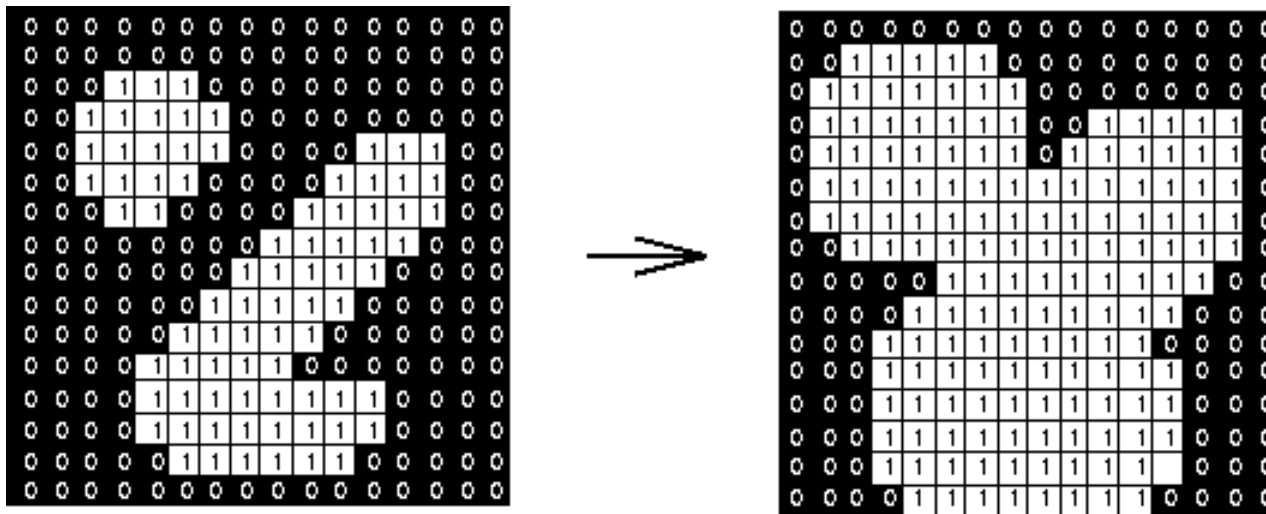
- Notação  $G = H \oplus F$

# Operador Dilatação

---

## Demo

- <http://www.cs.bris.ac.uk/~majid/mengine/morph.html>



# Operador Erosion: $G = H \ominus F$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	0	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

1	1	1
1	1	1
1	1	1

$H[u, v]$

Erosão: está H “contido em” F ao redor de [x,y]?

- $G[x,y] = 1$  se  $F[x+u-1,y+v-1]$  é 1 **em todo lugar** que  $H[u,v]$  é 1  
0 caso contrário

- Notação  $G = H \ominus F$

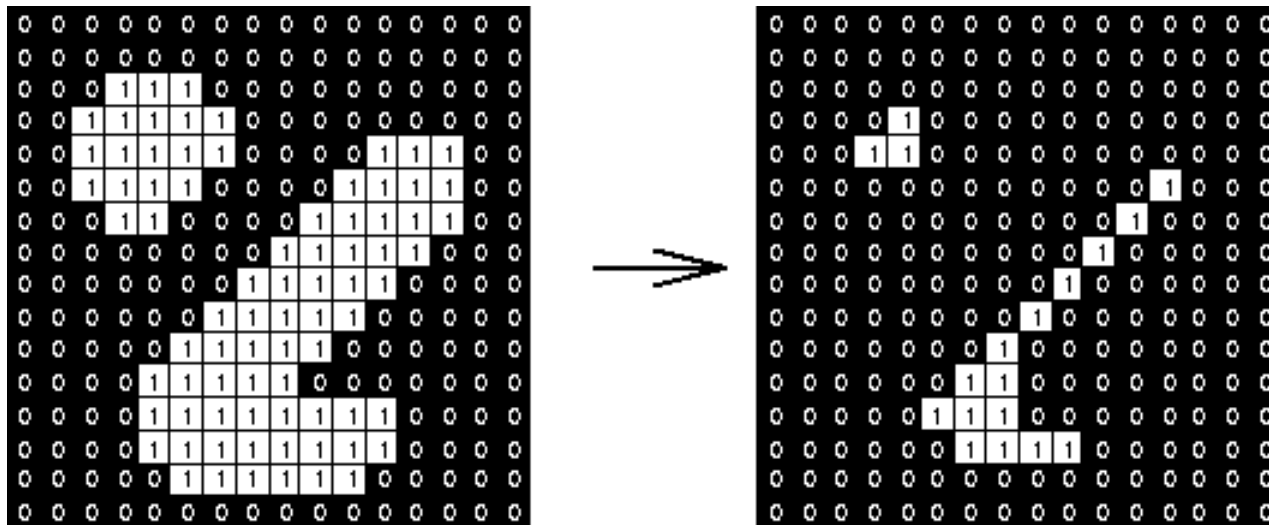


# Operador Erosão

---

## Demo

- <http://www.cs.bris.ac.uk/~majid/mengine/morph.html>

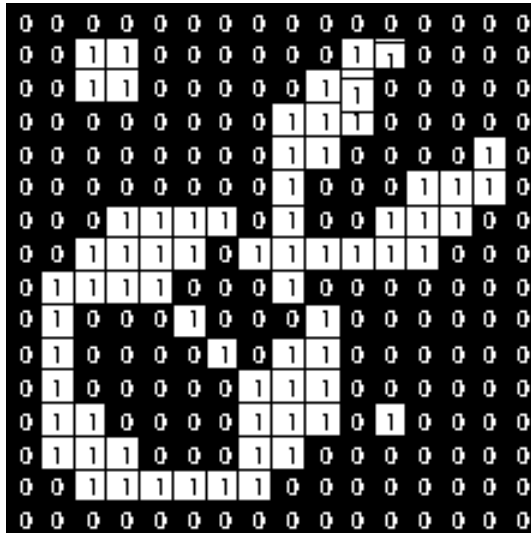


# Dilatações e erosões aninhados

---

O que faz esta operação?

$$G = H \ominus (H \oplus F)$$

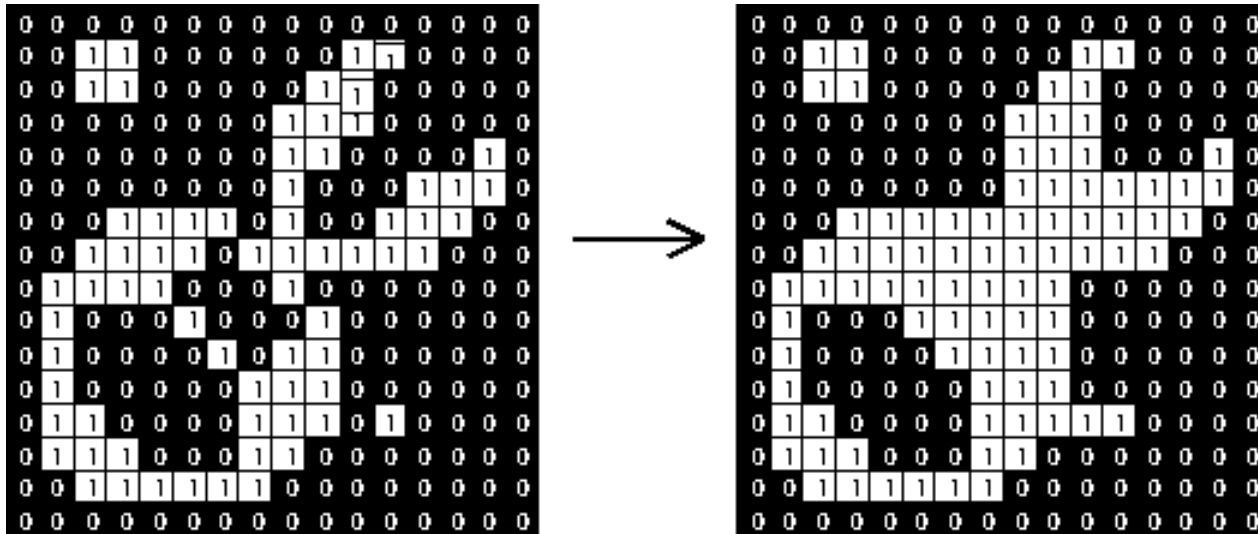


- esta é chamada de operação **fechamento**

# Dilatações e erosões aninhados

O que faz esta operação?

$$G = H \ominus (H \oplus F)$$



- esta é chamada de operação **fechamento**

É a mesma coisa que esta seguinte?

$$G = H \oplus (H \ominus F)$$

# Dilatações e erosões aninhados

---

O que faz esta operação?

$$G = H \oplus (H \ominus F)$$

- esta é chamada de operação **abertura**
- <http://www.dai.ed.ac.uk/HIPR2/open.htm>

Pode-se clarear imagens binárias pela aplicação combinada de dilatações e erosões

Operações de dilatações, erosões, abertura, e fechamento são conhecidas como **operações morfológicas**

- <http://www.dai.ed.ac.uk/HIPR2/morops.htm>

# Referencias

---

Richard Szeliski. “**Computer Vision: Algorithms and Applications**”. Springer, 2010. Draft: May 17, 2010.

University of Washington, Disciplina “**Computer Vision**”, 2009 e 2008:

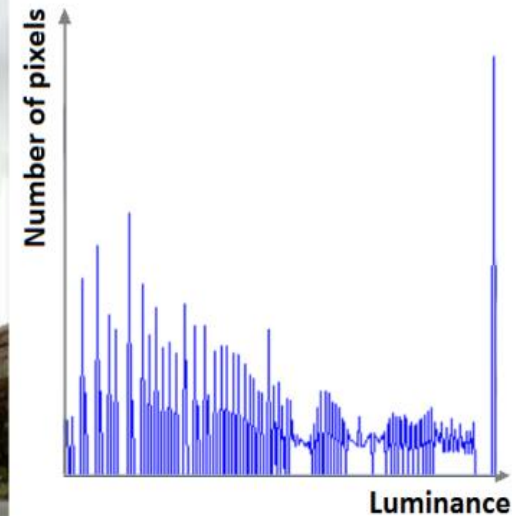
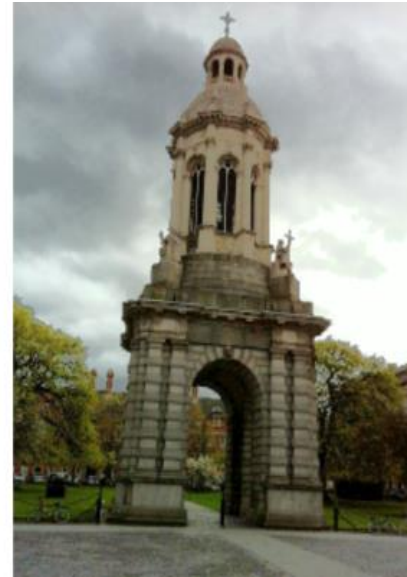
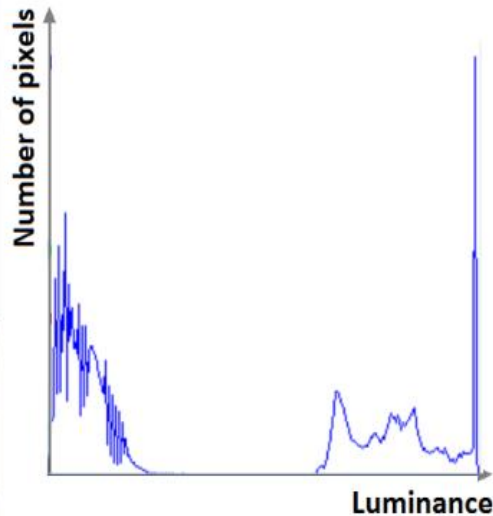
- <http://www.cs.washington.edu/education/courses/576/09sp/>
- <http://www.cs.washington.edu/education/courses/cse576/08sp/>

Leitura recomendada:

- Morphology, *in* HIPR - The Hypermedia Image Processing Reference
  - <http://www.dai.ed.ac.uk/HIPR2/morops.htm>
  - Dilation, erosion, opening, closing



# Equalisation



If an image has insufficient contrast  
Human can distinguish 700-900 greyscales  
Evenly distribute the greyscales...

- Result has missing greyscales

Normally equalise only the greyscales / luminance

# Edges

---

An approach to segmentation.

The analysis of the discontinuities in an image.



No correct answer?

An alternative to region based processing.

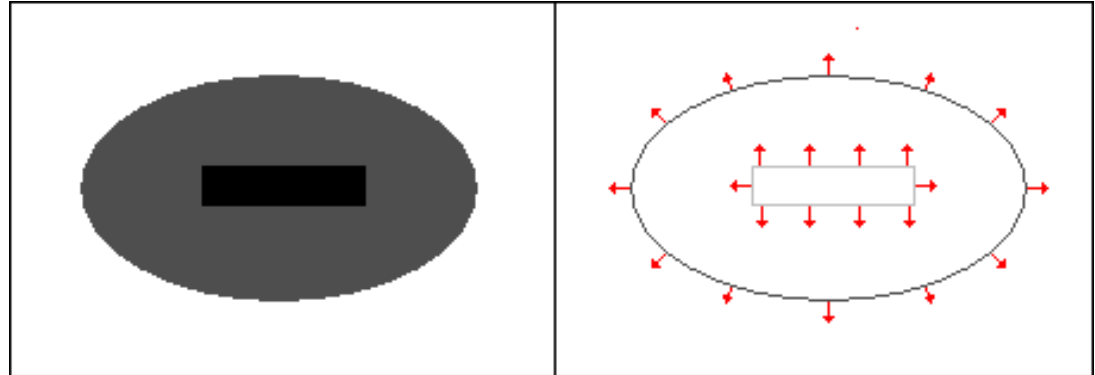


# Edge Detection – What is an edge?

Where brightness changes abruptly

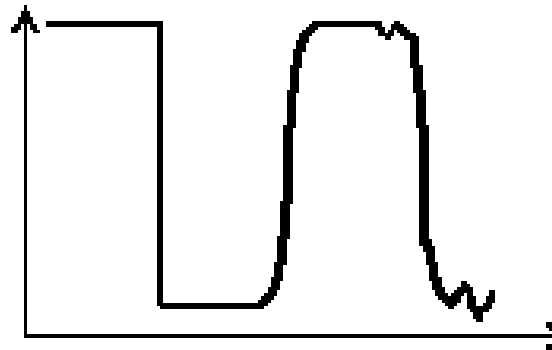
Edges have

- Magnitude (Gradient)
- Direction (Orientation)

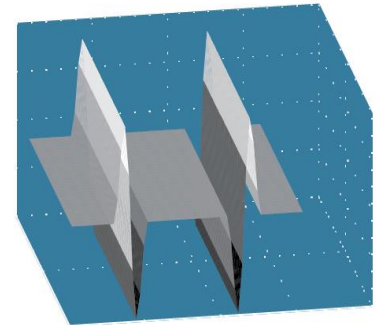
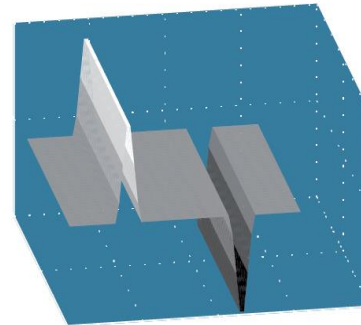
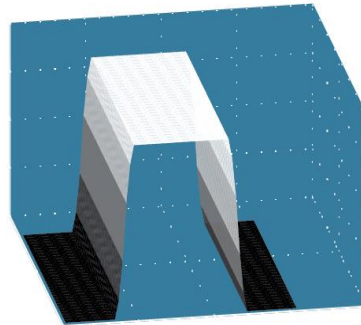


Edge Profiles

- Step
- Real
- Noisy



# Edge Detection – 1<sup>st</sup> derivative definitions

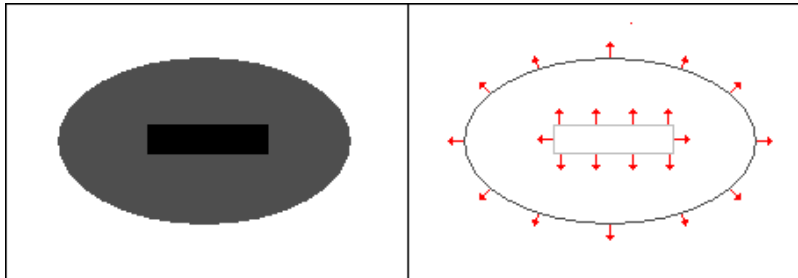


## Calculus...

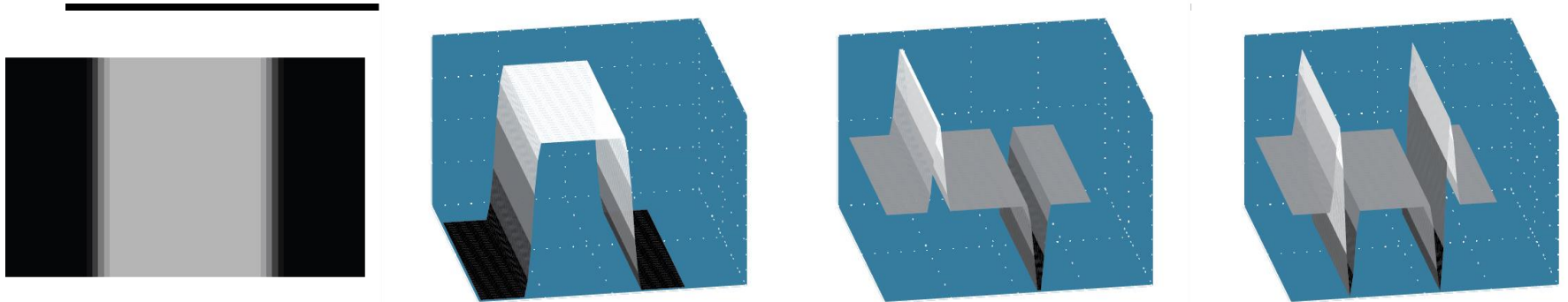
- Rate of change in two directions
- Vector variable:
  - Gradient Magnitude
  - Orientation (0° is East)

$$\nabla f(i,j) = \sqrt{\left(\frac{\delta f(i,j)}{\delta i}\right)^2 + \left(\frac{\delta f(i,j)}{\delta j}\right)^2}$$

$$\phi(i,j) = \arctan\left(\frac{\delta f(i,j)}{\delta j}, \frac{\delta f(i,j)}{\delta i}\right)$$



# Edge detection – 2<sup>nd</sup> derivative – Laplace operator



## Laplace operator

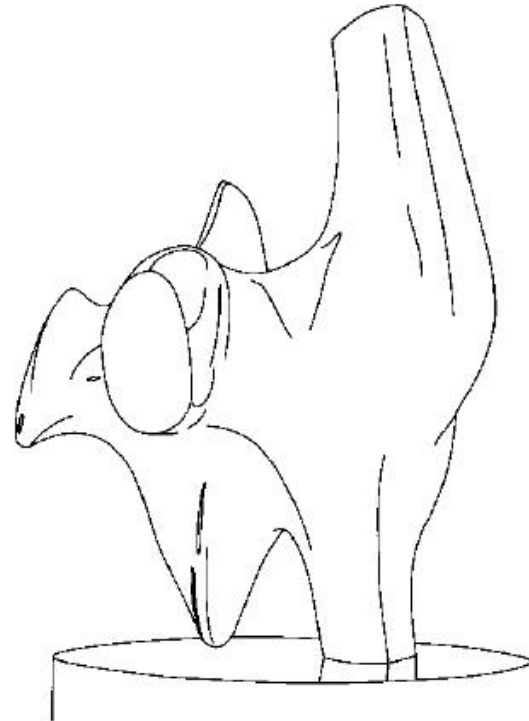
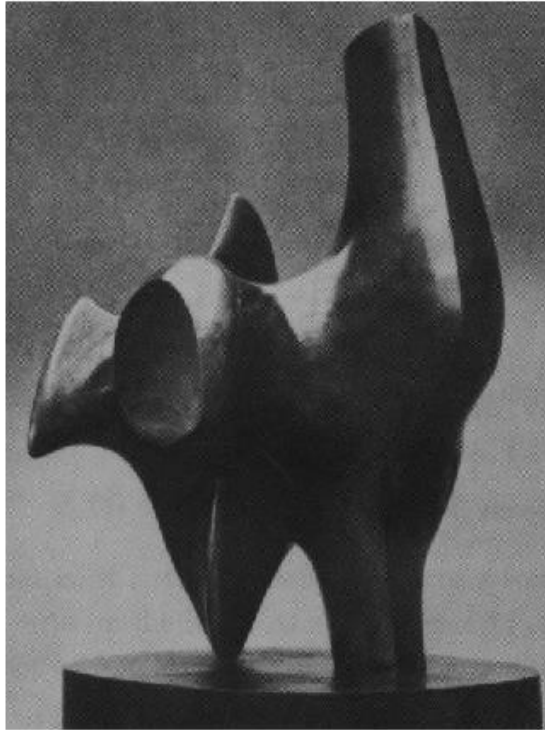
- Orientation independent
- Determines gradient magnitude
- Linear ramps
- Noise
- Zero-crossings

$$h(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h(i, j) = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

## EDGE – Borda ou Aresta

# Detecção de Bordas

---

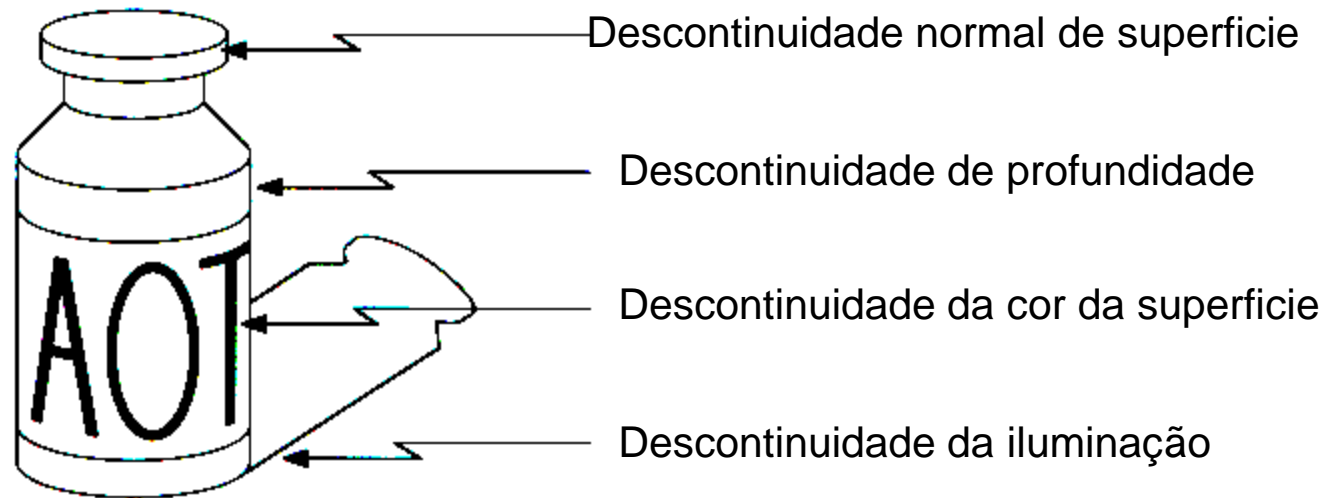


Converter uma imagem 2D num conjunto de curvas

- Extração de “features” salientes da cena
- Mais compacto do que pixels

# Origem de Bordas

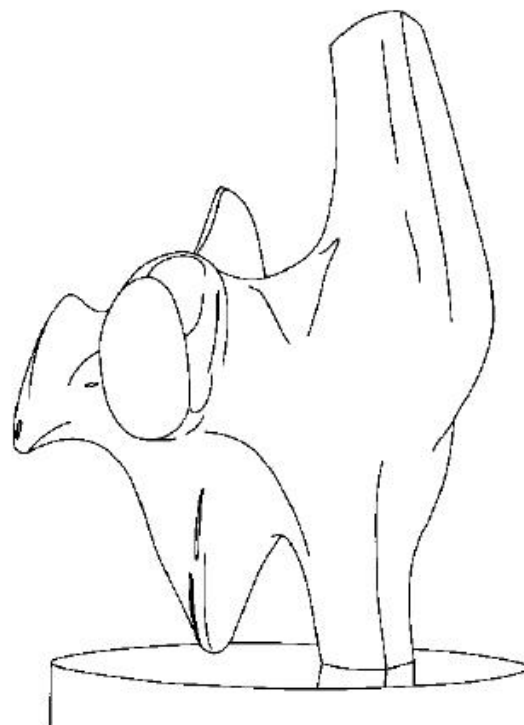
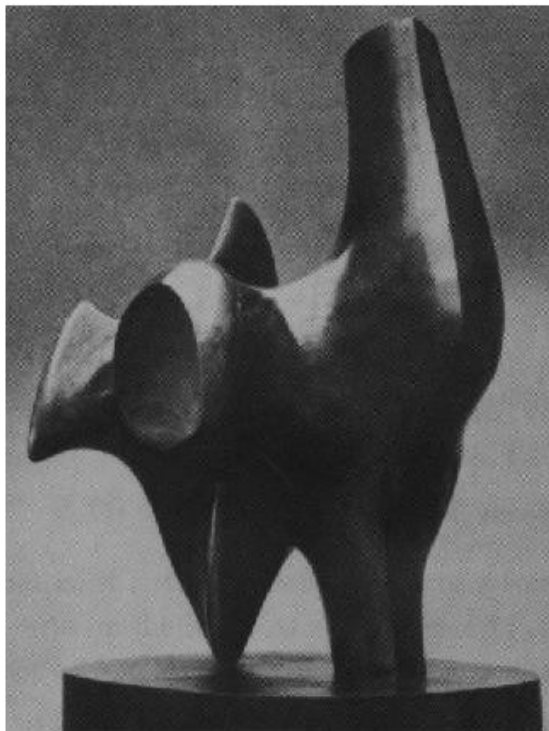
---



Bordas são causadas por uma variedade de fatores

# Detecção de Borda

---

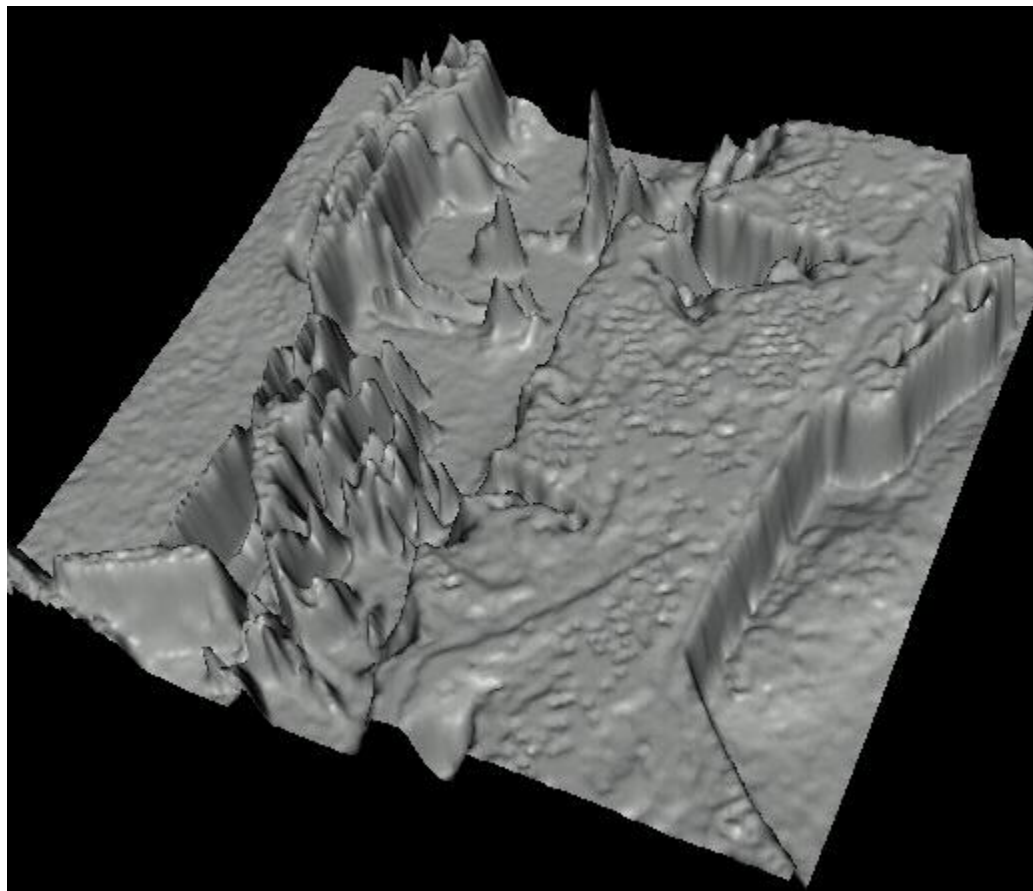


Como podemos dizer que um pixel está na borda?



# Imagens como funções...

---



Arestas se parecem penhascos íngremes

# Detectando bordas

---

O que é uma borda?

- descontinuidade de intensidade (= mudança rápida)

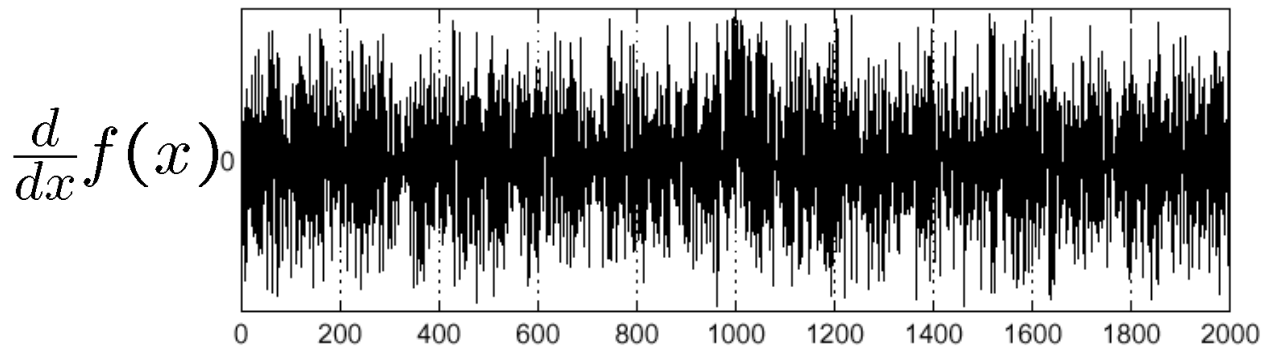
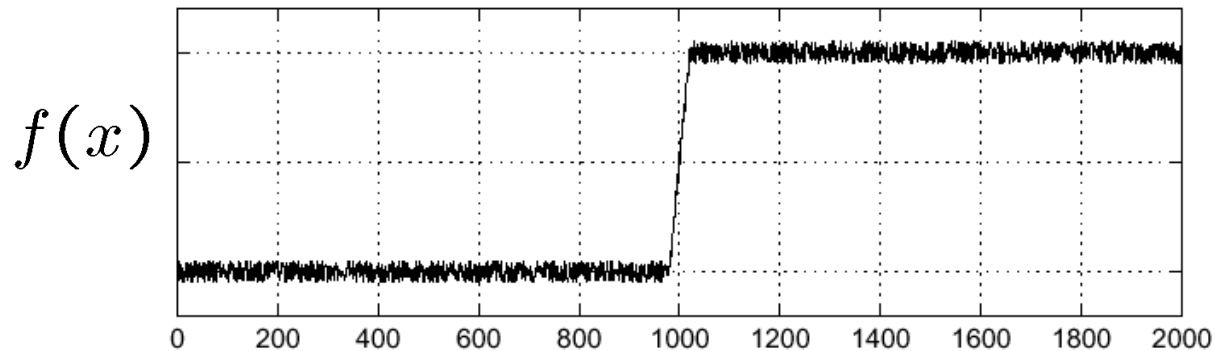
Como podemos encontrar grandes mudanças de intensidade?

- operador gradient parece ser a solução correta

# Efeitos do ruído

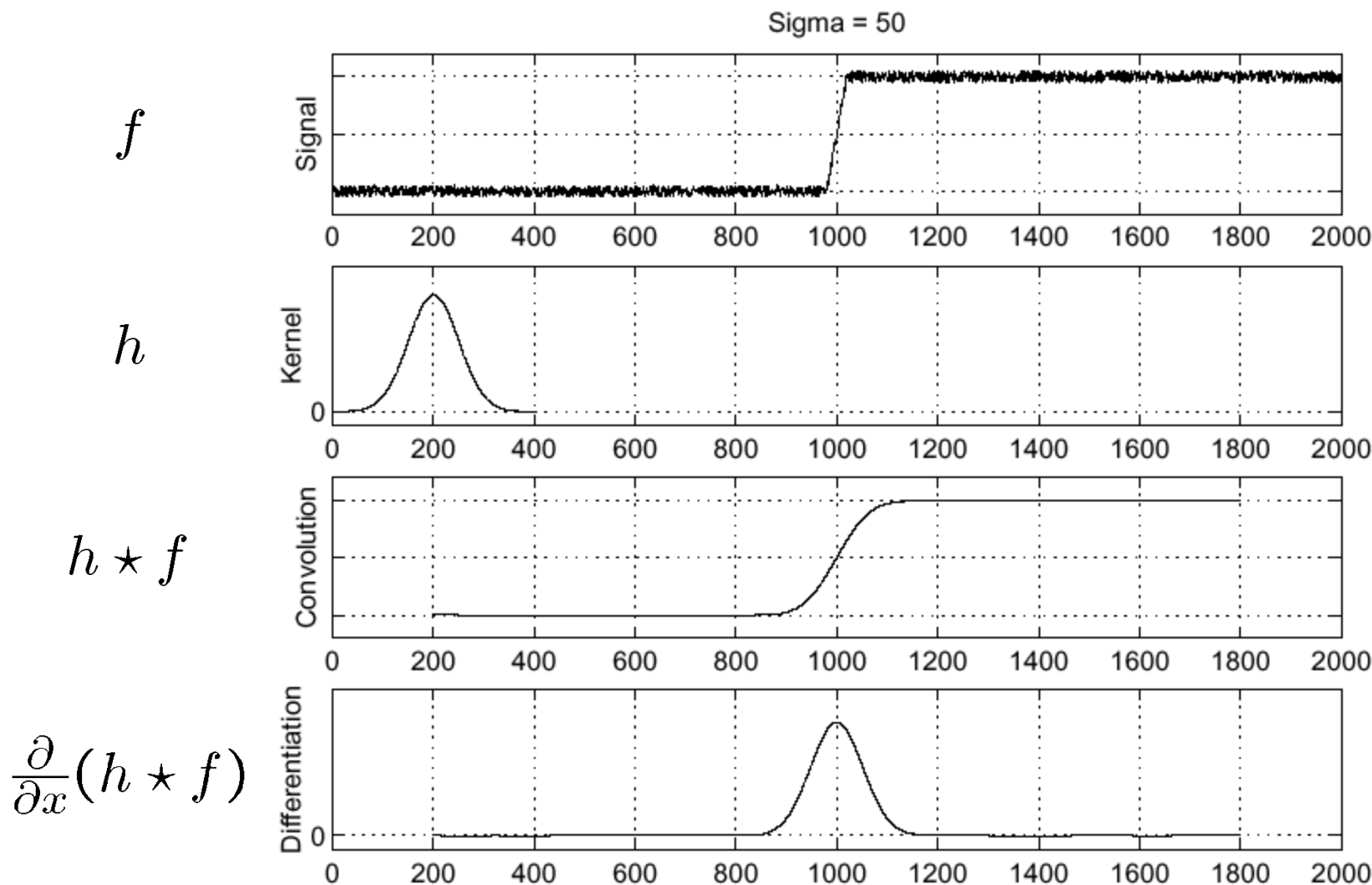
Considere uma unica linha ou coluna da imagem

- Plotando intensidade como uma função da posição tem-se um sinal



Onde está a borda?

# Solução: primeiro suavize



Onde está a borda? Procure por picos em  $\frac{\partial}{\partial x}(h \star f)$

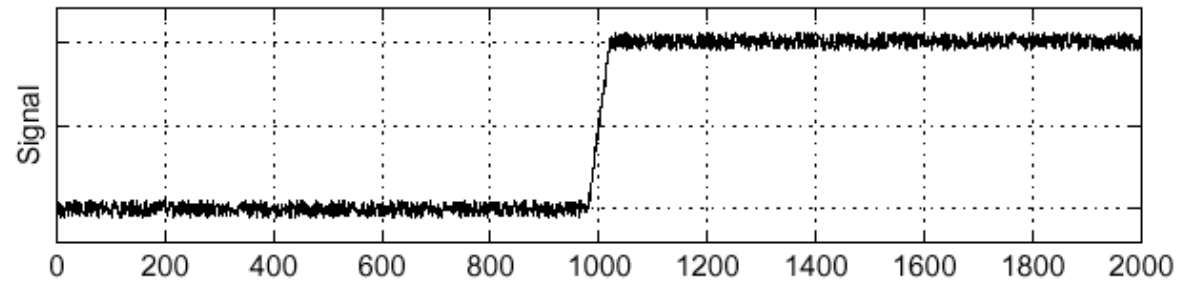
# Propriedade Associativa da convolução

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

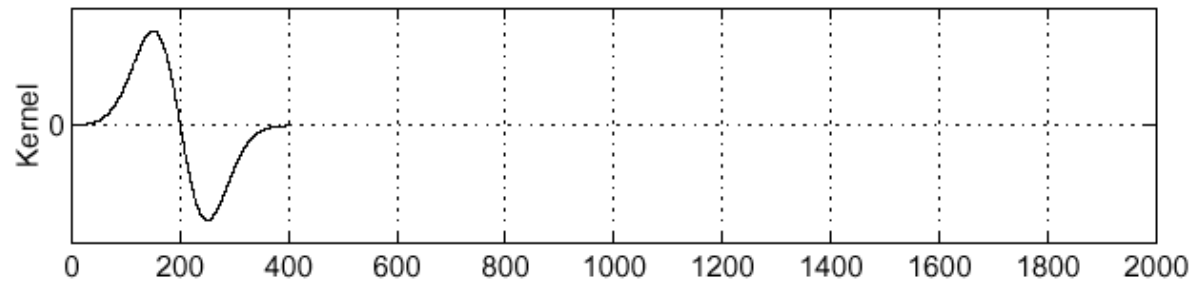
Isto nos economiza uma operação:

Sigma = 50

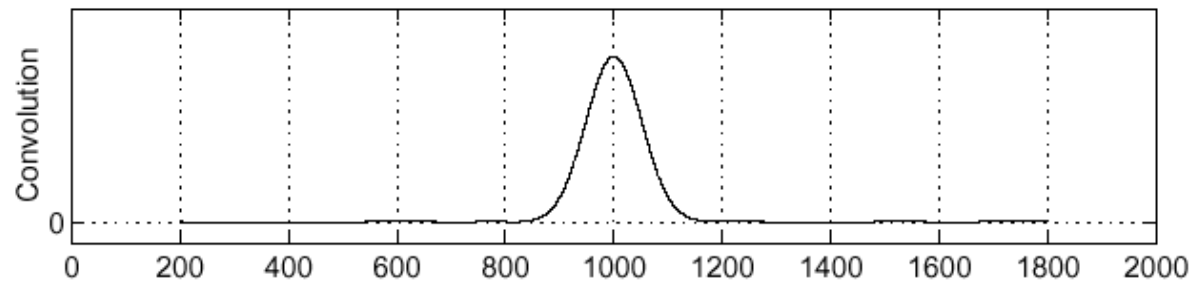
$f$



$\frac{\partial}{\partial x}h$



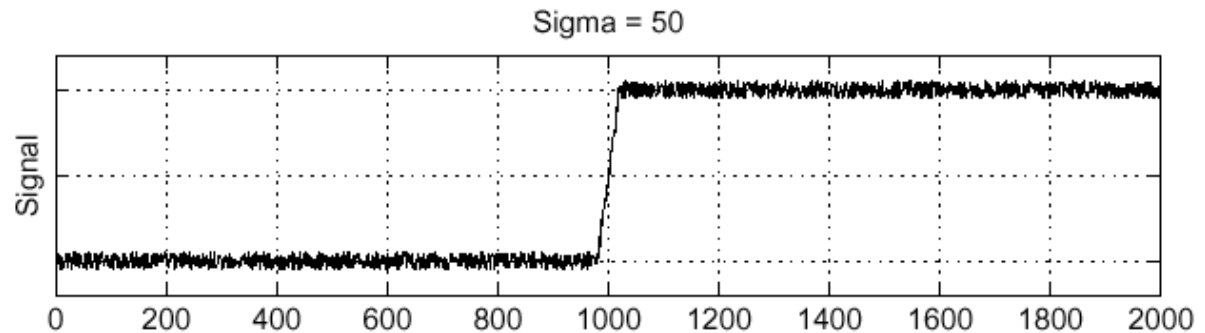
$(\frac{\partial}{\partial x}h) \star f$



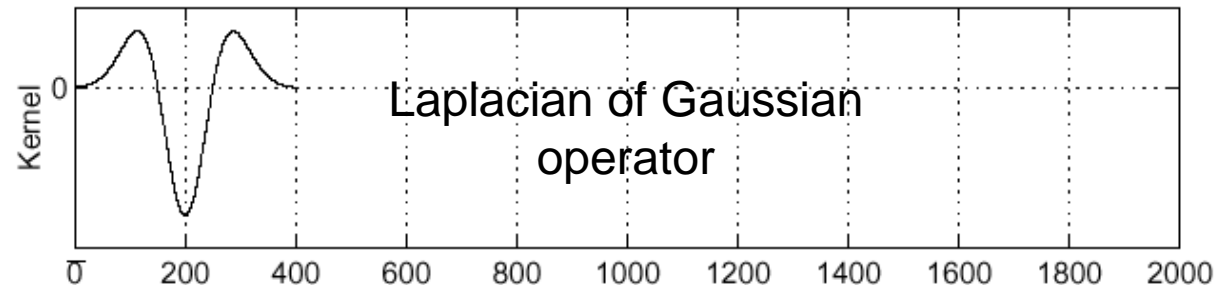
# Laplaciano da Gaussiana

Considere  $\frac{\partial^2}{\partial x^2}(h \star f)$

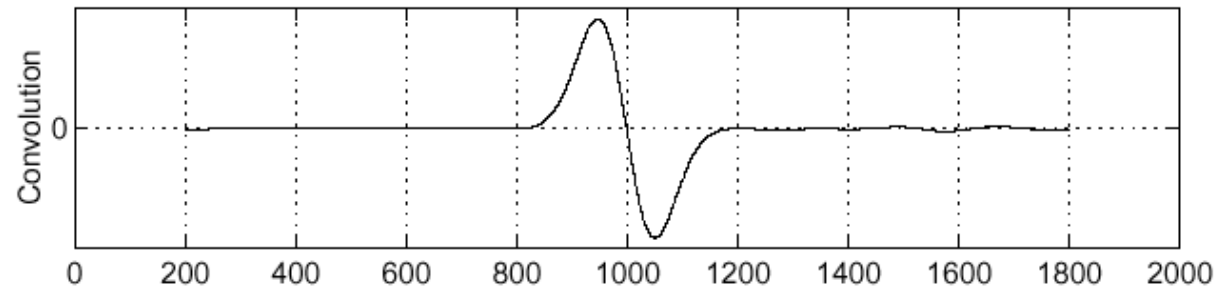
$f$



$\frac{\partial^2}{\partial x^2}h$



$(\frac{\partial^2}{\partial x^2}h) \star f$

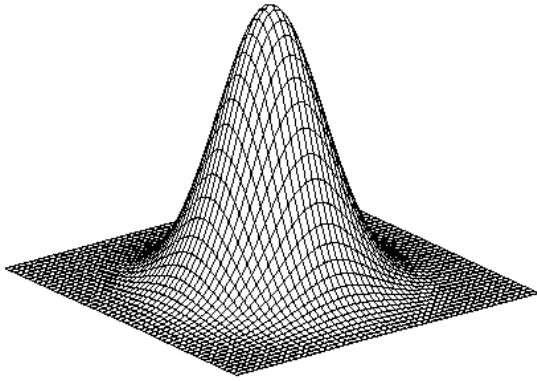


Onde está a borda?

Cruzamento-por-zero no grafico inferior

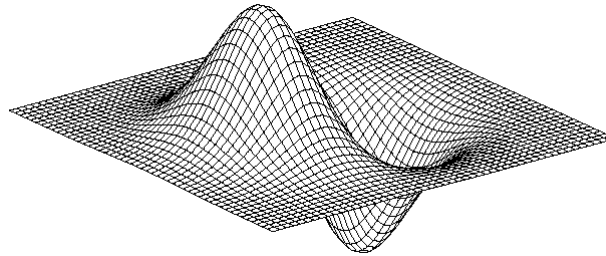
# Filtros 2D p/ detecção de borda

---



Gaussiana

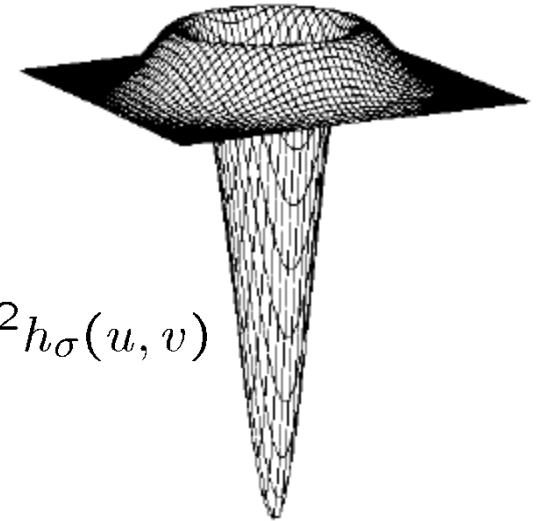
$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivativa da Gaussiana

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

Laplaciano da Gaussiana



$$\nabla^2 h_{\sigma}(u, v)$$

$\nabla^2$  é o operador **Laplaciano** :

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

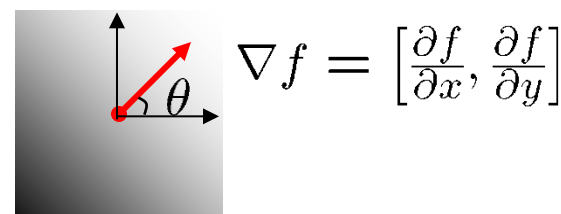
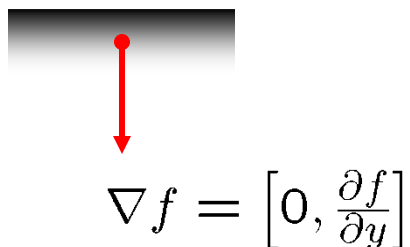
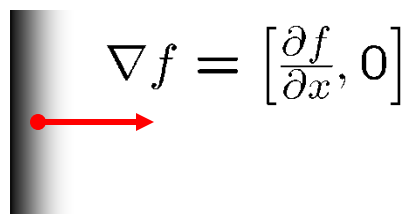


# Gradiente da Imagem

---

Gradiente de uma imagem:

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$



O gradiente aponta na direção do aumento mais rápido na intensidade

A direção do gradiente é dada por:

$$\theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

- Como isso tem a ver com a direção da borda?

A *força da borda* é dada pela magnitude do gradiente

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

# O Gradiente Discreto

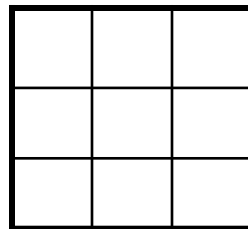
---

Como podemos diferenciar uma imagem *digital*  $F[x,y]$ ?

- Opção 1: reconstruir uma imagem continua, e depois fazer o gradiente
- Opção 2: fazer a derivada discreta (“diferença finita”)

$$\frac{\partial f}{\partial x}[x, y] \approx F[x + 1, y] - F[x, y]$$

Como voce implementaria isto com a correlação cruzada?



$H$

# O operador Sobel

---

Aproximação comum da derivada da Gaussiana

$$\frac{1}{8} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

$s_x$

$$\frac{1}{8} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

$s_y$

- A definição padrão do operador Sobel operator omite o termo  $1/8$ 
  - Isso não faz diferença para a detecção de borda
  - O termo  $1/8$  é necessário para se obter valor correto do gradiente, porém.

# Detector de borda “Canny”

---



imagem original (Lena)

# Detector de borda “Canny”

---



norma do gradiente

# Detector de borda “Canny”

---



*thresholding* - limiarização

# Detector de borda “Canny”

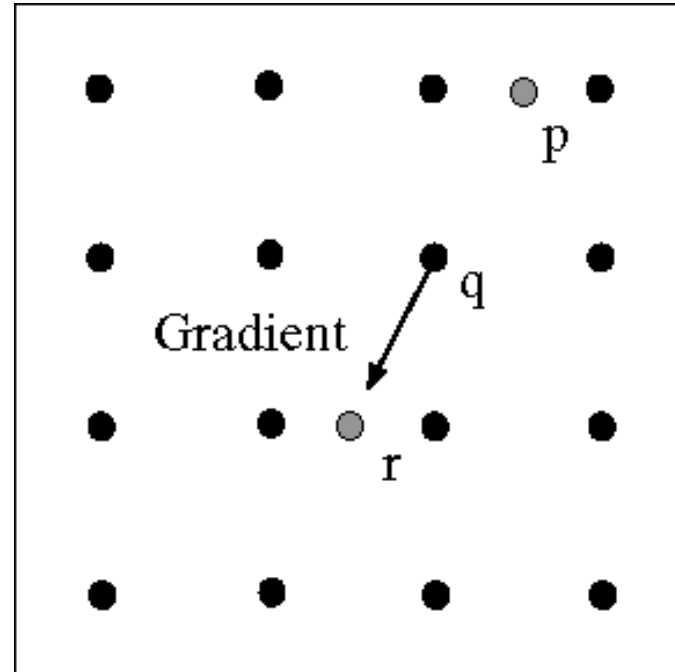
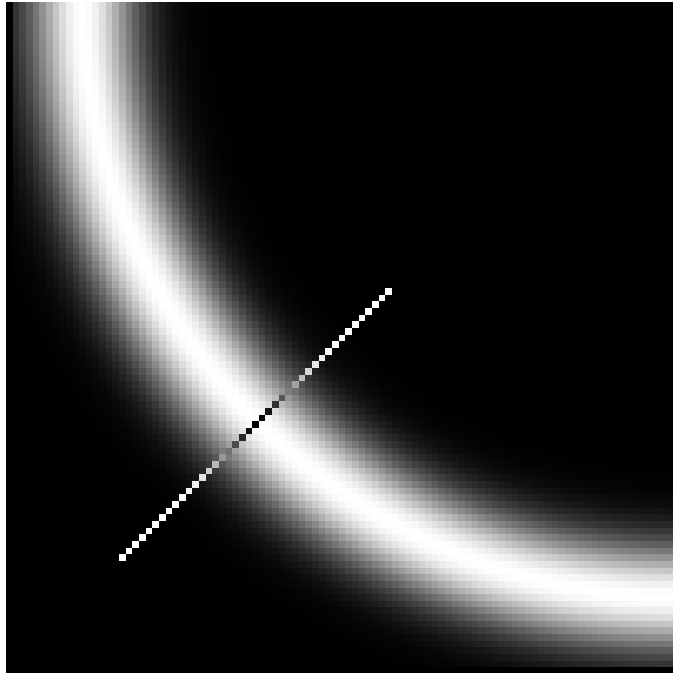
---



thinning - afinamento  
(supressão não-maxima)

# Supressão Não-maxima

---



Verifique se o pixel é um maximo local ao longo da direção do gradiente

- requer verificar os pixels interpolado  $p$  e  $r$



# Efeito do $\sigma$ (Gaussian kernel spread/size)



original



Canny com  $\sigma = 1$



Canny com  $\sigma = 2$

A escolha do  $\sigma$  depende do comportamento desejado

- grande  $\sigma$  detecta bordas de grandes escala
- pequeno  $\sigma$  detecta característica detalhadas

# Detecção de Borda por subtração

---



original

# Detecção de Borda por subtração

---



suavizado (Gaussiana 5x5)

# Detecção de Borda por subtração

---



Porque isto funciona?

suavizada – original  
(escalonada por 4, offset +128)

# Referencias

---

Richard Szeliski. “**Computer Vision: Algorithms and Applications**”. Springer, 2010. Draft: May 17, 2010.

University of Washington, Disciplina “**Computer Vision**”, 2009 e 2008:

- <http://www.cs.washington.edu/education/courses/576/09sp/>
- <http://www.cs.washington.edu/education/courses/cse576/08sp/>

## Leituras recomendadas

- Artigo de [Cipolla & Gee sobre detecção de borda](#)
- Szeliski – Seção 4.2 – Edges

