

Visão Computacional

Aula 10

Detecção de Bordas
Filtro de Canny

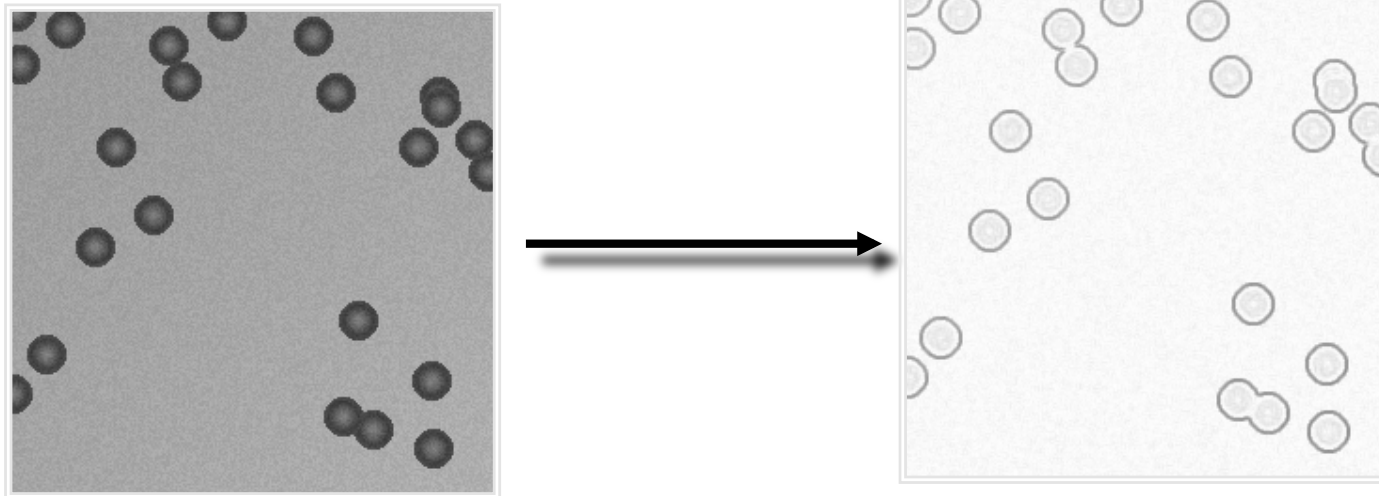
Detecção de Bordas em Imagens

Detecção de bordas em imagens é uma das operações mais comuns em análise de imagens.

As bordas formam as linhas de contorno dos objetos.

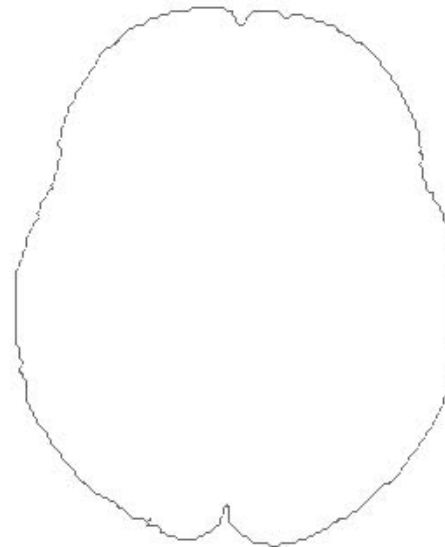
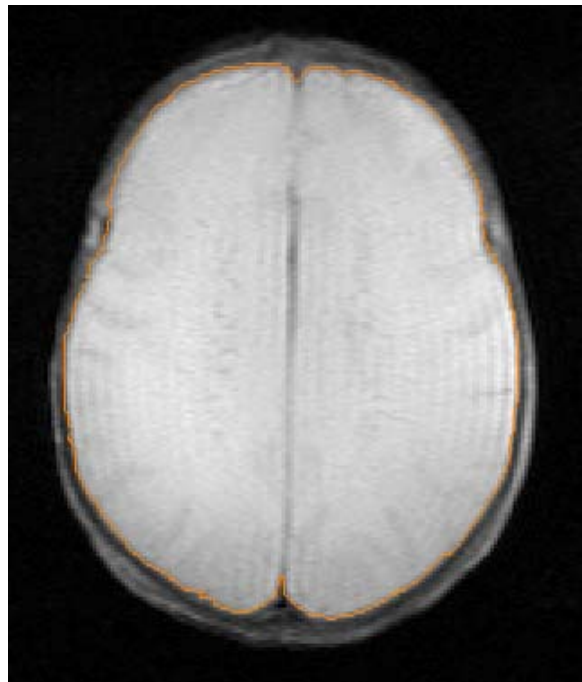
- área
- perímetro
- determinar formas

Exemplo:



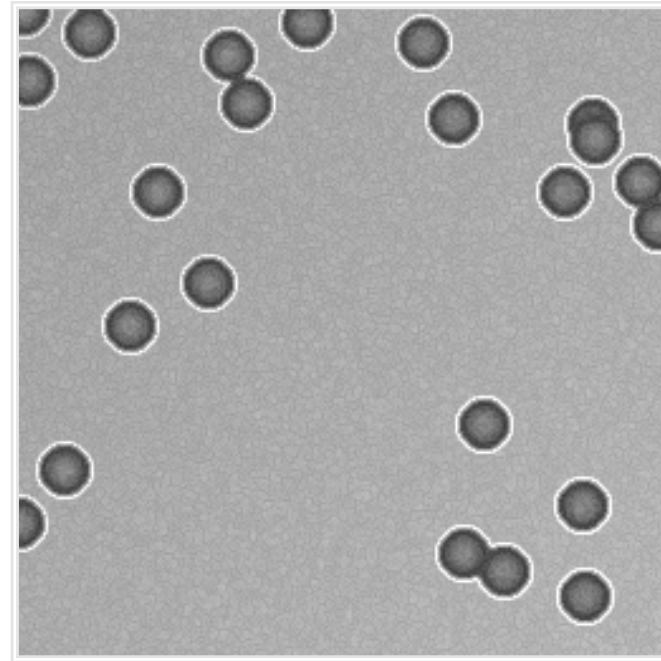
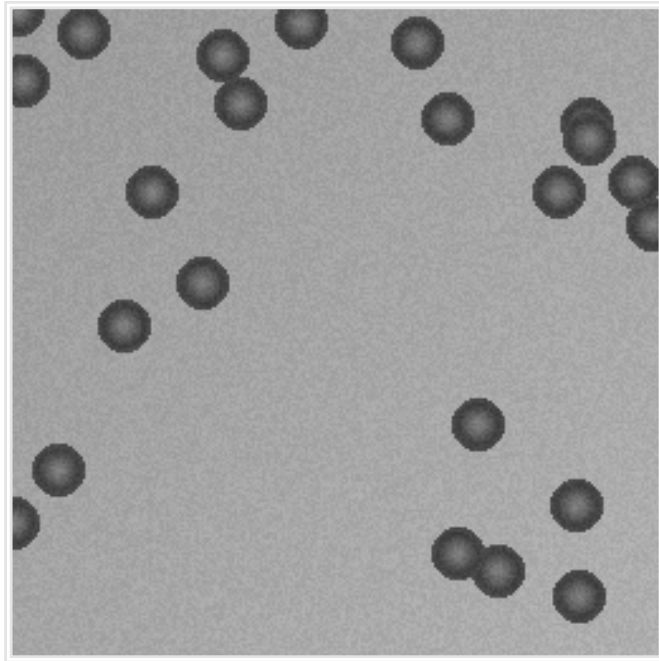
Definições

- *detecção de bordas (edge detection)* → processo de localização dos pixels situados nas linhas de contorno dos objetos.



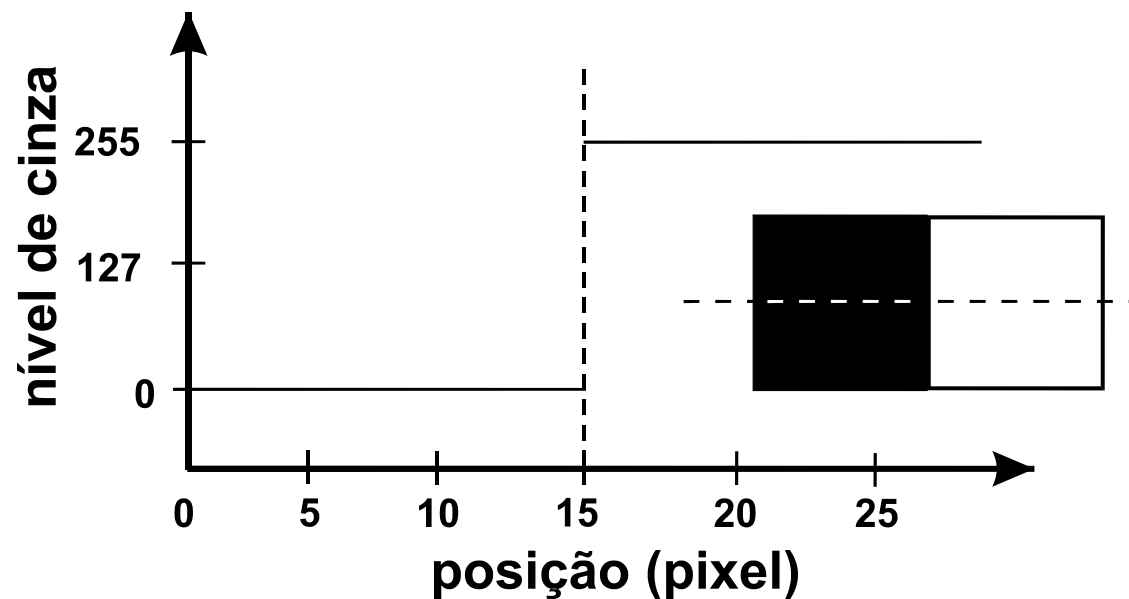
Definições:

- *realce de bordas (edge enhancement)* → processo de aumento do contraste das bordas tornando-as mais visíveis.



Conceito de Borda

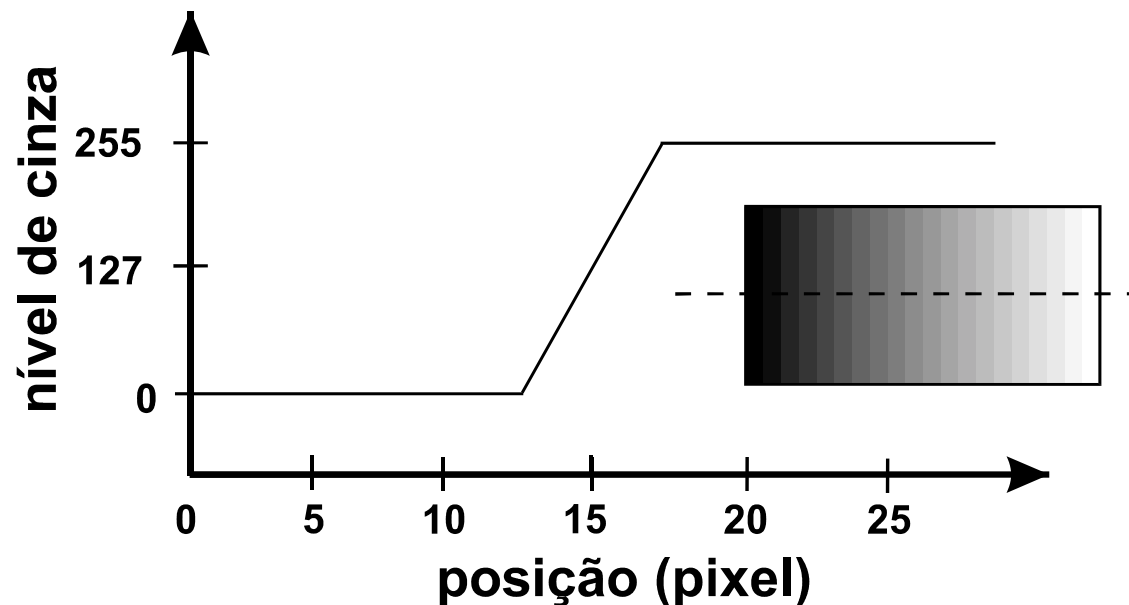
- *Borda ideal*: aquela em que a mudança no nível de cinza é a maior possível.



Conceito de Borda

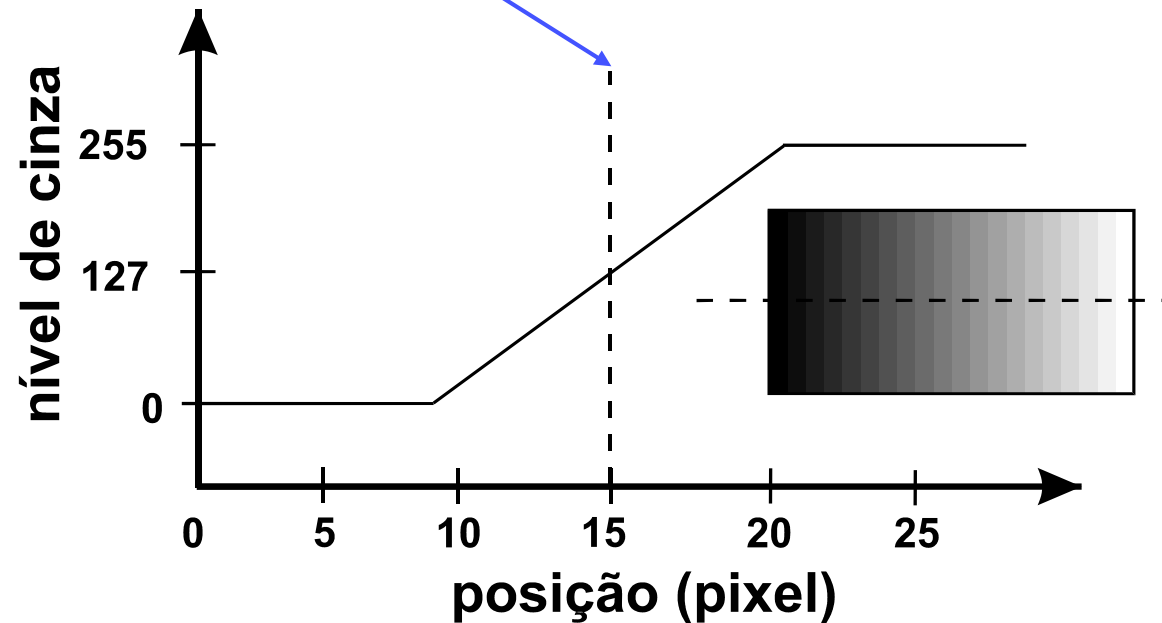
Borda ideal raramente ocorre em processamento de imagens.

A borda dos objetos ocorre ao longo de alguns pixels com diferentes níveis de cinza pixels.



Posição de uma Borda

A posição da borda é definida no centro da rampa.



Exemplo de Bordas:

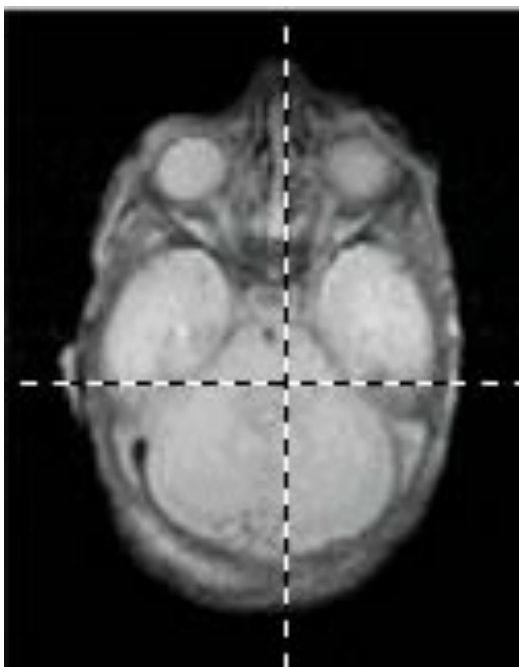


Imagem de RMN



Perfil horizontal



Perfil vertical



Problemas na Detecção de Bordas

Em uma cena é difícil que dois pixels que pertençam a um mesmo objeto tenham exatamente o mesmo nível de cinza na imagem digitalizada.

A presença de ruído, ou alterações aleatórias nos níveis de cinza, só pode ser caracterizada estatisticamente.

Imagens com variações aleatórias nos níveis de cinza e rampas suaves jamais terão bordas ideais para serem detectadas.

Etapas da Detecção de Bordas

Em geral 3 passos devem ser executados na detecção de borda:

- **Filtragem:** Sendo o gradiente calculado a partir de dois pontos, o resultado é sensível ao ruído. Assim utiliza-se algum processo de filtragem para melhorar a resposta do detector de borda.
Compromisso: \uparrow filtragem \rightarrow \downarrow definição das bordas
- **Realce :** A detecção das bordas depende das variações de intensidade entre pixels vizinhos. O cálculo do gradiente realça os pixels onde as mudanças de intensidade são significativas.
- **Detecção :** Seleção dos pixels com mudanças intensas de intensidade. Vários pontos na imagem possuem gradientes diferentes de zero e não são bordas. Em geral, usa-se algum processo de limiarização após o gradiente para selecionar os pixels que formam as bordas.

Considerações:

Bordas → são mudanças significativas no nível de cinza entre dois pixels próximos ou vizinhos.

Derivada → é uma técnica para determinar as variações presentes em uma função.

Realce de bordas → a derivada é uma técnica de realce das bordas em imagens

Gradiente

Operador derivativo → fornece uma estimativa da direção da borda. Deriva-se a imagem em relação a x e y e calcula-se a soma vetorial das derivadas.

Cálculo do gradiente:

$$\nabla A(x, y) = \left(\frac{\partial A}{\partial x}, \frac{\partial A}{\partial y} \right)$$

onde a função $A(x, y)$ representa a função imagem.

Gradiente

Numericamente, o cálculo do gradiente (∇) em uma imagem pode ser aproximado pelas seguintes relações discretas:

$$\nabla_x(x, y) = \frac{\Delta A(x, y)}{\Delta x} = \frac{A(x, y) - A(x - 1, y)}{\Delta x}$$

$$\nabla_y(x, y) = \frac{\Delta A(x, y)}{\Delta y} = \frac{A(x, y) - A(x, y - 1)}{\Delta y}$$

Máscara para o Gradiente

As aproximações mais simples para o gradiente são:

$$\begin{cases} G_x \approx f(x, y) - f(x + 1, y) \\ G_y \approx f(x, y) - f(x, y + 1) \end{cases}$$

As quais podem ser aproximadas pelas máscaras:

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

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

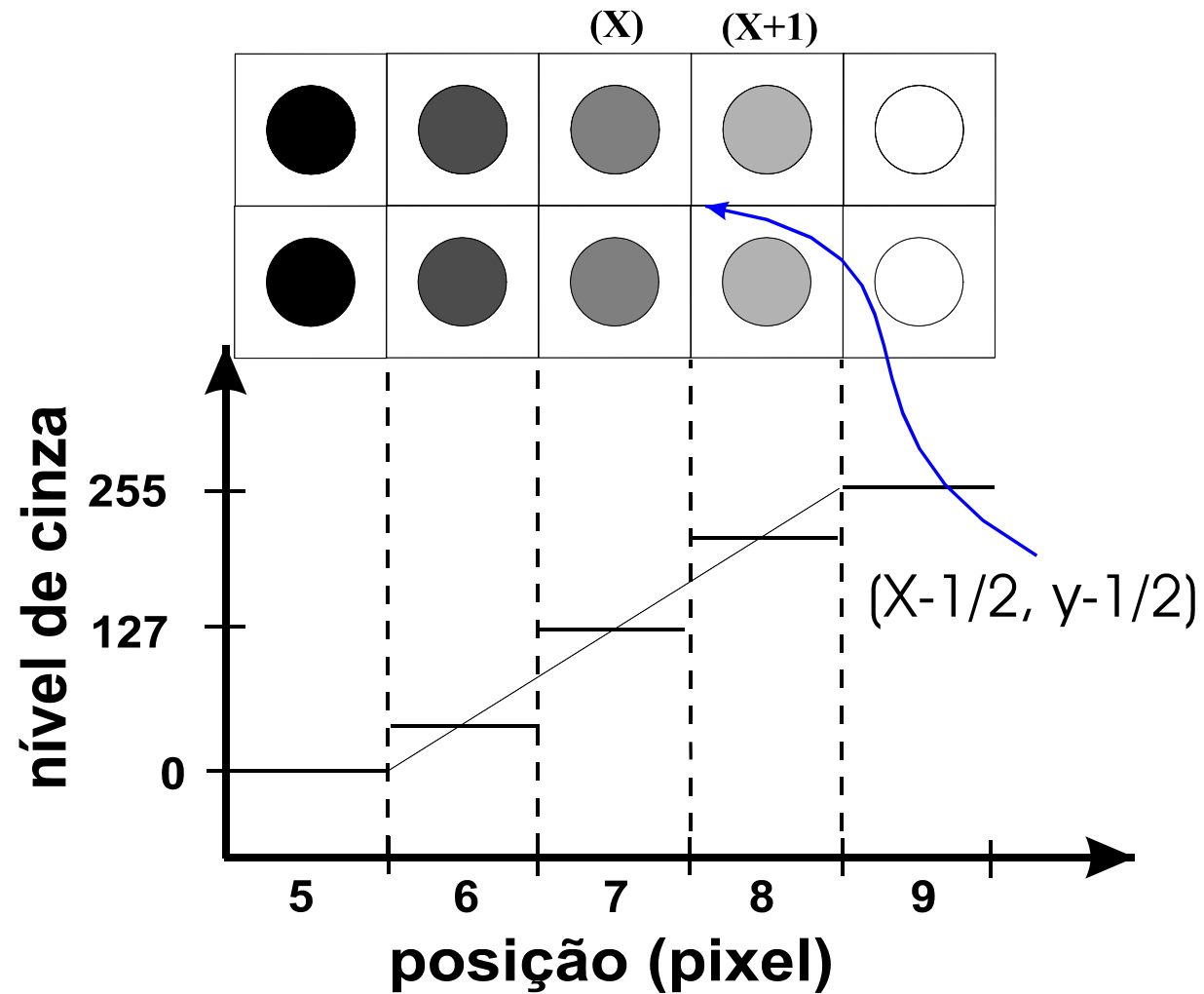
Nessas aproximações G_x corresponde ao ponto $(x+1/2, y)$ e G_y a $(x, y+1/2)$. Assim, poderia se utilizar uma máscara 2x2:

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

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

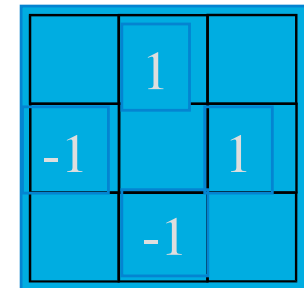
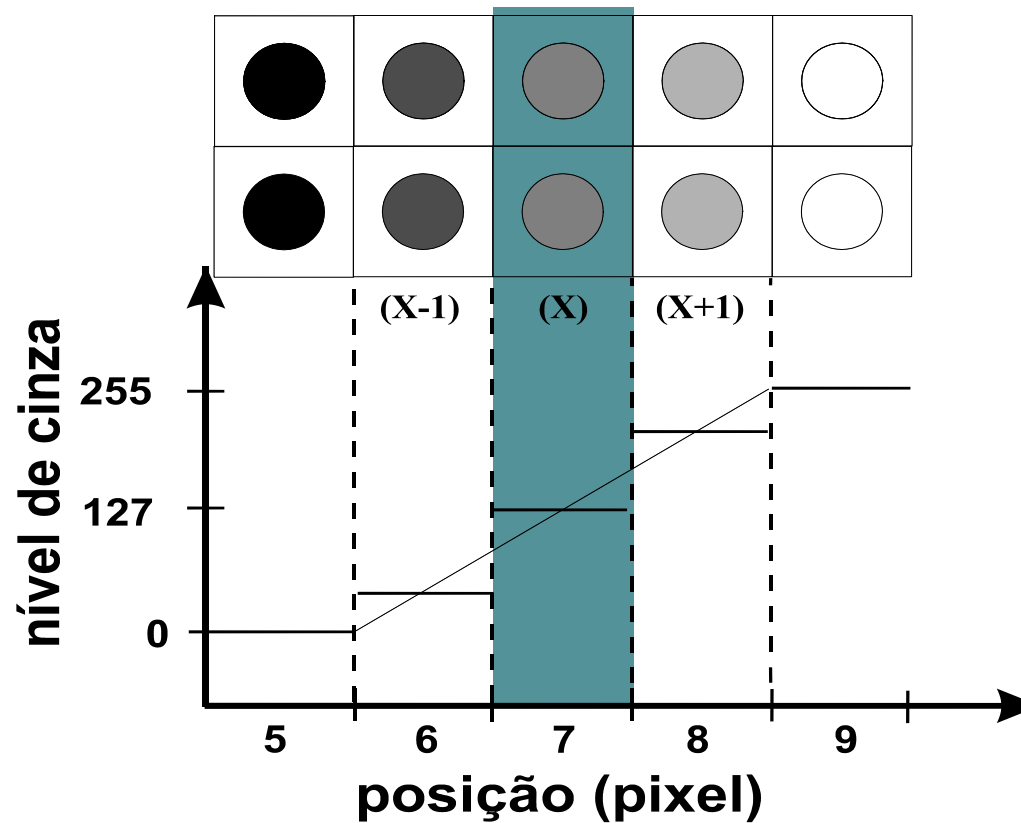
Nesse caso o ponto para o qual a derivada é calculado é $(x+1/2, y+1/2)$

Cálculo do Gradiente



Melhor aproximação para localização da borda

$$\left. \begin{aligned} \nabla_x(x, y) &= A(x+1, y) - A(x-1, y) \\ \nabla_y(x, y) &= A(x, y+1) - A(x, y-1) \end{aligned} \right\} \begin{array}{l} \text{Operação simétrica} \\ \text{em relação ao pixel} \\ (x, y) \end{array}$$



Cálculo do Gradiente

A magnitude do gradiente é o comprimento da hipotenusa:

$$G_{mag} = \sqrt{\left(\frac{\partial A}{\partial x}\right)^2 + \left(\frac{\partial A}{\partial y}\right)^2}$$

e a direção aproximada da borda:

$$G_{direção} = \tan^{-1} \left(\frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right)$$

Cálculo do Gradiente

Observações:

Note que a magnitude do gradiente não depende da direção da borda (operação denominada de isotrópica)

Pode-se aproximar o cálculo da magnitude:

$$G_{mag} \approx |G_x| + |G_y|$$

$$G_{mag} \approx \max \left(|G_x|, |G_y| \right)$$

Cálculo do Gradiente utilizando “Templates”

A idéia do uso de “templates” é substituir a aplicação direta dos conceitos de derivada por uma máscara com coeficientes que sejam sensíveis/simulem uma borda.

Existe uma variedade de templates aplicados na detecção de bordas, entre eles:

- Roberts
- Sobel
- Kirsch
- Prewitt
- Frein-Chen
- Laplaciano

Operador de Roberts

O operador Roberts é uma aproximação simples da magnitude do gradiente.

$$G[f(x, y)] = |G_x| + |G_y|$$

$$G[f(x, y)] = |f(x, y) - f(x + 1, y + 1)| + |f(x, y + 1) - f(x + 1, y)|$$

$$\mathbf{G}_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{G}_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

O operador Roberts é uma operação aproximada do gradiente no ponto $(x+1/2, y+1/2)$ e não no ponto (x, y) .

Detector de Sobel

Operador Sobel na direção x:

$$\nabla_x = A(x, y) - A(x - 1, y)$$

Template
3x3

-1	1	
-1	1	

	-1	1
	-1	1

	-1	1
	-1	1

-1	1	
-1	1	

-1	0	1
-2	0	2
-1	0	1

 = S_x

Detector de Sobel

Operador Sobel na direção y:

$$\nabla_y = A(x, y) - A(x, y - 1)$$

Template
3x3

	-1	-1
	1	1

	-1	-1
	1	1

-1	-1	
1	1	

-1	-1	
1	1	

-1	-2	-1
0	0	0
1	2	1

= Sy

Detector de Borda de Sobel

A magnitude é dada por:

$$G_{mag} = \sqrt{Sx^2 + Sy^2}$$

Essa operação envolve a determinação de uma raiz quadrada e duas potências de dois. Operações computacionais lentas.

Em geral resume-se essa operação no cálculo da soma:

$$|Sx| + |Sy|$$



Detector de Kirsch

As máscaras de Kirsch modelam a mudança de nível de cinza observadas nas bordas.

Existe uma máscara para cada direção.

Templates de Kirsch

$K_0 =$

-3	-3	5
-3	0	5
-3	-3	5

$K_1 =$

-3	5	5
-3	0	5
-3	-3	-3

$K_2 =$

5	5	5
-3	0	-3
-3	-3	-3

$K_3 =$

5	5	-3
5	0	-3
-3	-3	-3

$K_4 =$

5	-3	-3
5	0	-3
5	-3	-3

$K_5 =$

-3	-3	-3
5	0	-3
5	5	-3

$K_6 =$

-3	-3	-3
-3	0	-3
5	5	5

$K_7 =$

-3	-3	-3
-3	0	5
-3	5	5

Templates de Kirsch

Por exemplo, se a maior resposta for para a máscara K_0 implica em uma borda vertical (gradiente horizontal) no pixel correspondente ao centro da máscara.

Para encontrar as bordas dos objetos a imagem é convoluída com as 8 máscaras.

Seleciona-se a maior resposta entre as 8 máscaras.

Outros Templates

Prewitt:

$$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

(vertical)

$$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(horizontal)

Frei-Chen:

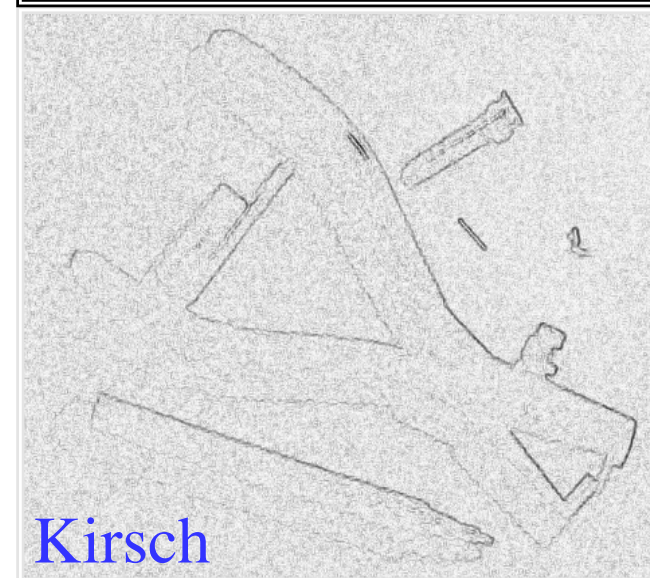
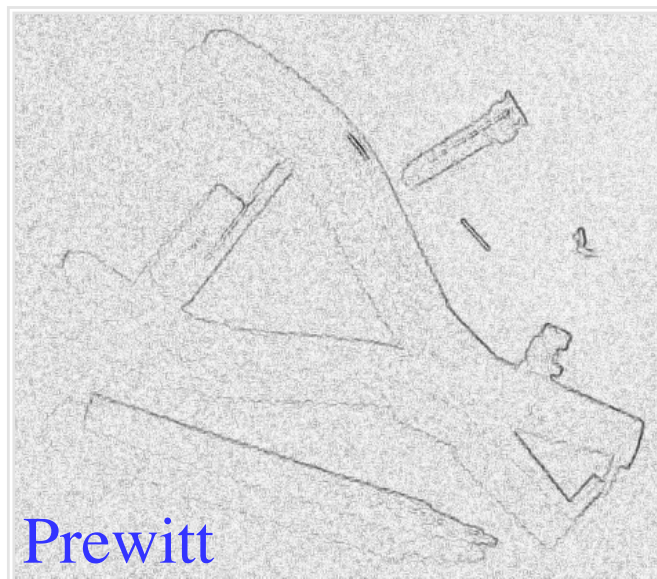
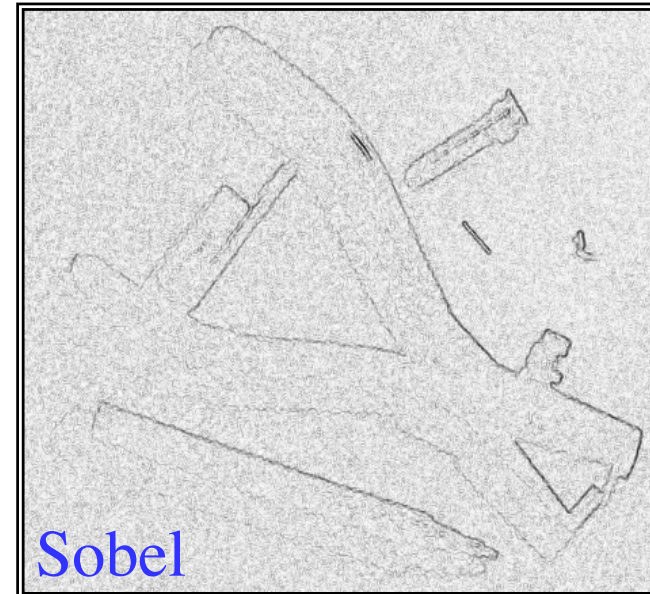
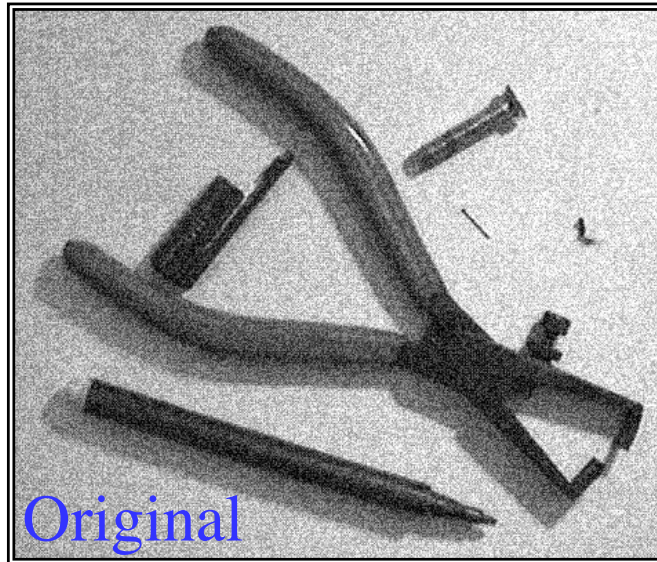
$$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} 1 & 0 & 1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & 1 \end{bmatrix}$$

(vertical)

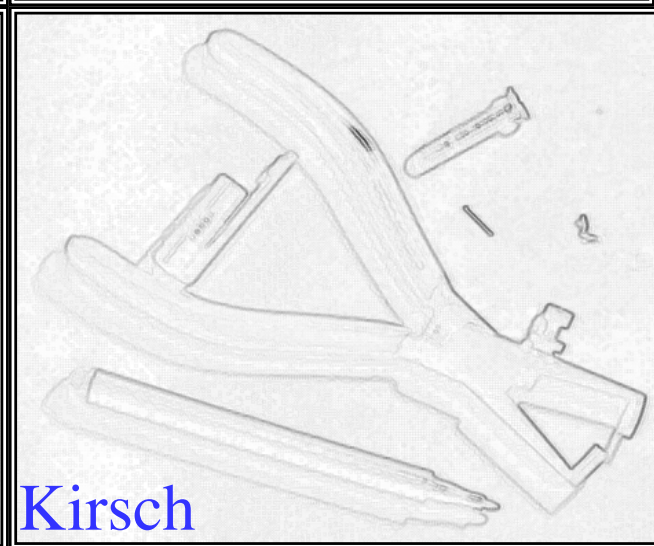
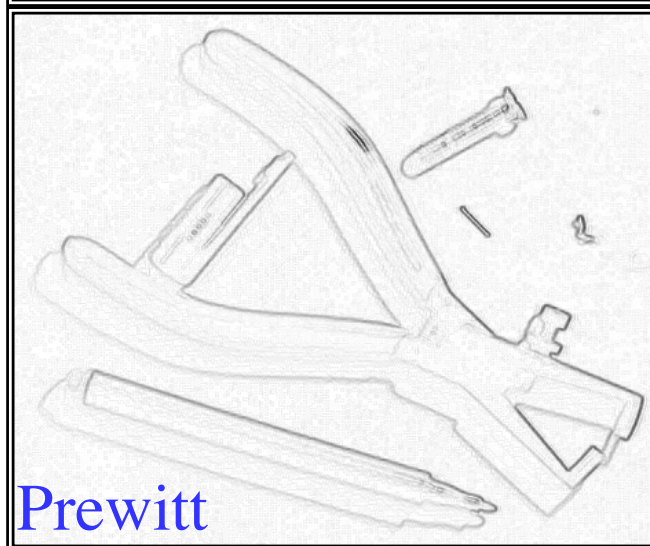
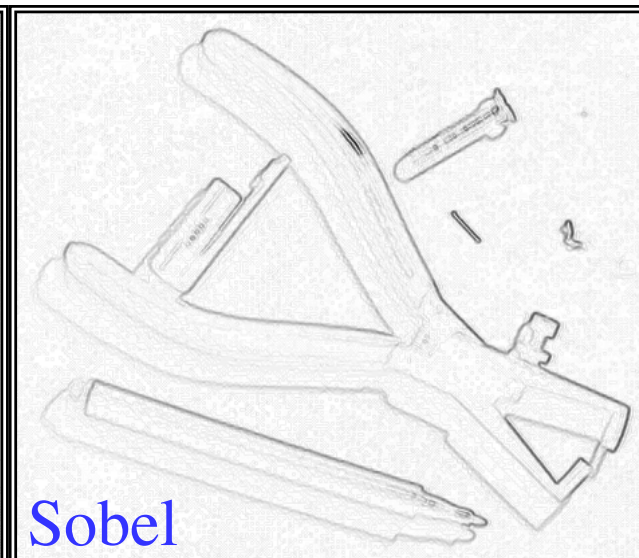
$$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} 1 & -\sqrt{2} & 1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

(horizontal)

Resultado Comparativo (imagem Ruidosa)



Resultado Comparativo (imagem sem ruído)





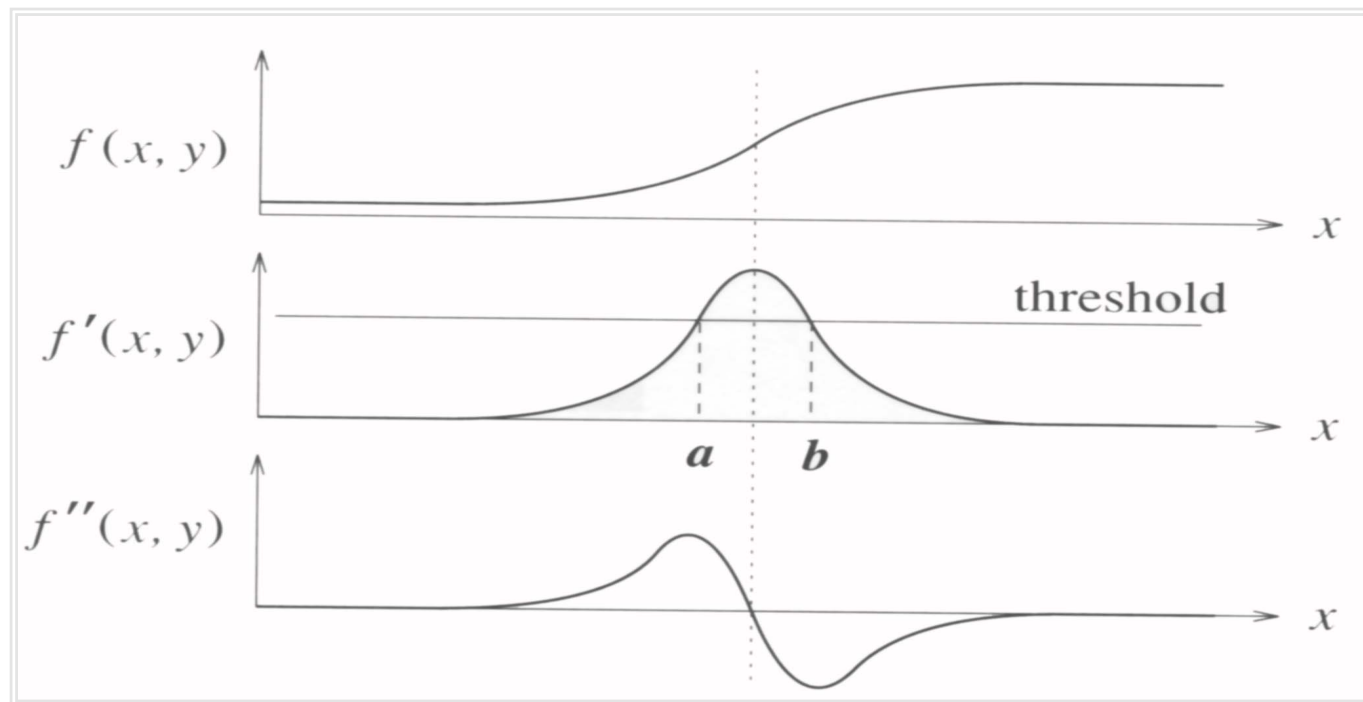
Operadores Derivativos de 2ª Ordem

Os operadores baseados na 1ª derivada determinam a presença da borda quando o resultado é superior a um certo limiar.

Em geral, essa operação resulta em bordas com muitos pixels ou espessas.

Operadores de 2ª Ordem

A melhor aproximação é localizar os pontos de máximos no gradiente:



ou seja, um pico na derivada 1a corresponde a um cruzamento por zero na derivada 2a. As bordas podem então ser identificadas por um detetor de cruzamento por zero.

Operador Laplaciano (Template)

O operador Laplaciano é definido por:

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

e pode ser aproximado pelas seguintes máscaras:

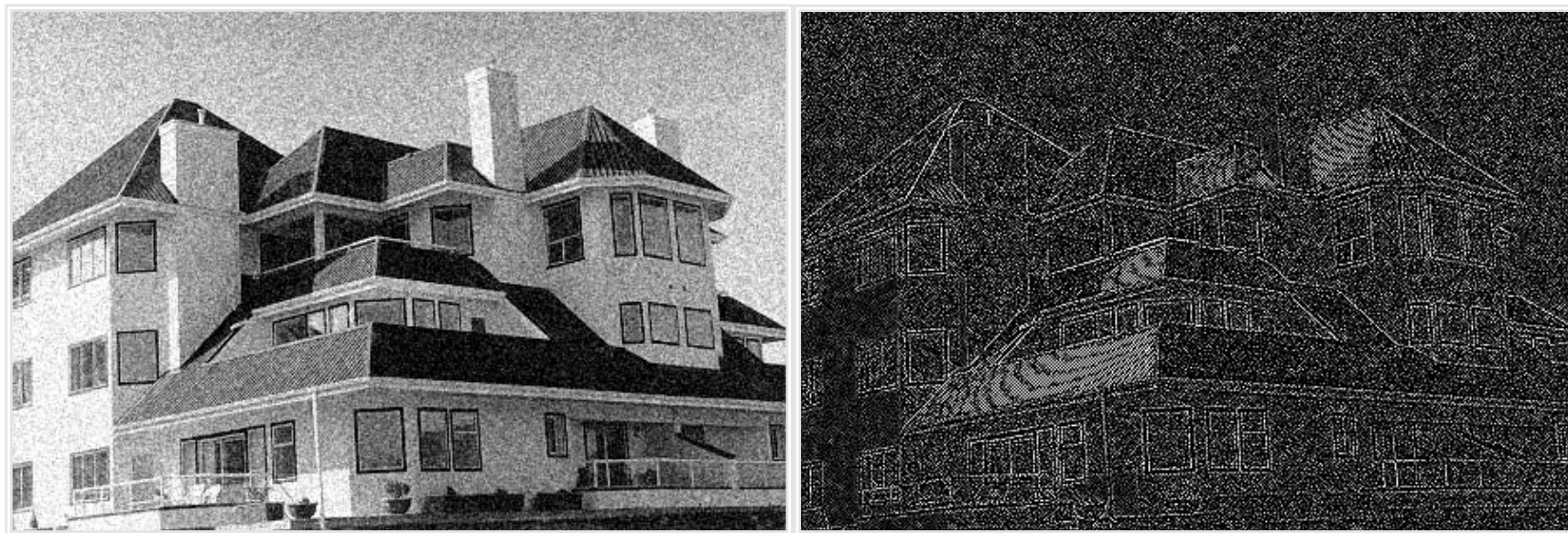
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Exemplo Laplaciano (3x3)



Exemplo Laplaciano (3x3 + Ruído)



Embora o Laplaciano seja insensível a direção da borda e, portanto capaz de realçar as bordas em qualquer direção, seu uso é restrito devido a sua grande sensibilidade a ruído (Marques Filho, 1999).

Detector de Bordas de Marr-Hildreth

Marr observou cinco pontos principais sobre o sistema visual:

1. As características de interesse em uma imagem dependem de diferentes fatores, tais como: tons de cinza, bordas, suavidade, texturas, etc.
 - Não existe um operador único capaz de levar em conta todos esses fatores simultaneamente.
 - É necessário combinar os resultados de diferentes operadores.
2. Uma cena não é constituída apenas de padrões de difração ou outro efeito de onda de luz e, alguma forma de média local (*smoothing*) deve estar presente.



Detector de Bordas de Marr-Hildreth

3. O filtro de suavização que vai de encontro com as observações dos sistema de visão biológicos (suavização e localização no domínio espacial e banda limitada no domínio da frequência) é o Gaussiano.
4. Quando ocorre uma mudança de intensidade (borda) a derivada primeira possui altos valores. Isso representa um cruzamento por zero da segunda derivada.
5. O operador diferencial que independe da direção é o Laplaciano.



Detector de Bordas de Marr-Hildreth

Baseado nestas observações propôs-se o seguinte:

1. Convoluir a imagem I com uma função Gaussiana de 2D;
2. Aplicar na imagem convoluída o Laplaciano;
3. Os pixels da borda são aqueles para os quais há um cruzamento por zero.

Implementação do Algoritmo

A função Gaussiana de 2D é dada por:

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

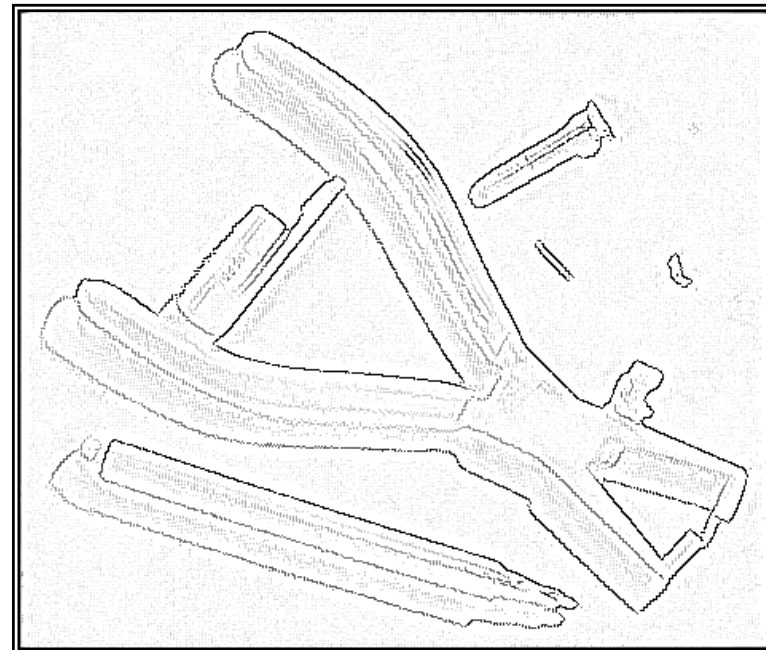
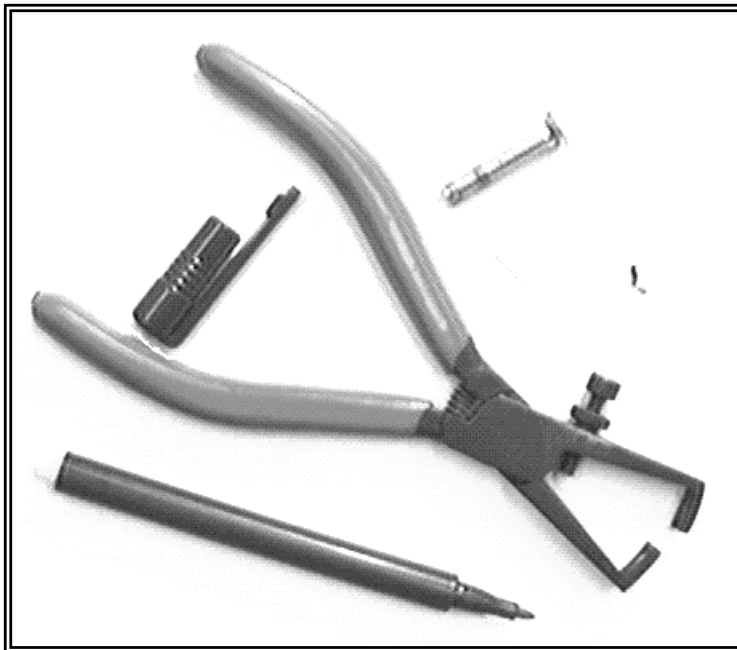
e o operador Laplaciano:

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

Pode-se, ainda, calcular o Laplaciano do Gaussiano (LdG) e, a partir de amostras, implementar uma máscara de convolução. O Laplaciano do Gaussiano é igual a:

$$\nabla^2 G_{\sigma} = \left(\frac{r^2 - 2\sigma^2}{\sigma^4} \right) e^{\frac{-r^2}{2\sigma^2}} \quad \text{onde} \quad r = \sqrt{x^2 + y^2}$$

Exemplo do Detector de Borda



Detector de borda Canny

- “Taxonomicamente” este operador deveria estar junto aos operadores de primeira derivada, mas devido a sua “boniteza” foi deixado para o final
 - Marr Hildreth (1980)
 - Canny (1986)
 1. Baixa taxa de erro : encontrar todas as bordas
 2. Boa localização: bordas detectas o mais próximo das bordas reais
 3. Resposta única: um único ponto de borda retornado (nro de máximos locais deve ser mínimo)

Detector de borda Canny

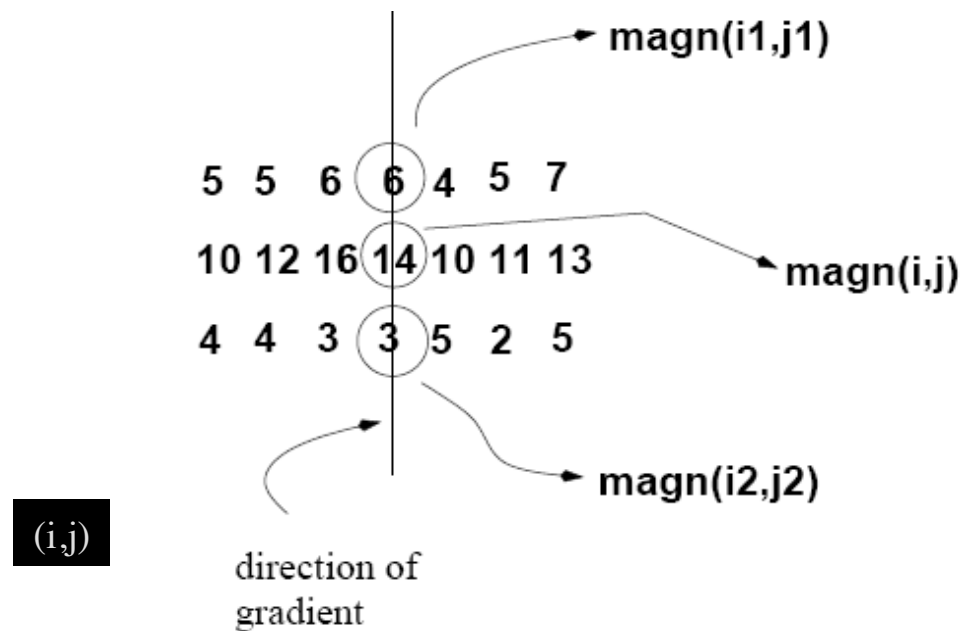
- Atacando os critérios taxa de erro e localização
 - Canny mostrou que uma boa aproximação para um detector ótimo de bordas de degrau é a primeira derivada de uma Gaussiana.
 - A análise é baseada em bordas de degrau (step-edges) corrompidas por ruído branco Gaussiano
 - (ruído branco : ruído com espectro de frequência contínuo e uniforme sobre uma banda específica)
 - (ruído branco Gaussiano: ruído branco em que a distribuição dos valores de amplitude é Gaussiano)
- Critério única resposta
 - Aproximação numérica

Canny - Etapas

1. Suavizar uma imagem f por uma Gaussiana
 - $F_s(x,y) = f(x,y) * G(x,y)$
2. Para cada pixel, computar a magnitude $M(x,y)$ e a direção do gradiente $\alpha(x,y)$
 - Usar qq filtro de primeira derivada visto antes
 - $M(x,y)$ fatalmente conterá cristas largas em torno dos máximos locais. Para afinar estas cristas utilize algoritmo de supressão dos não máximos, afinando $M(x,y)$
3. Limiarizar $M(x,y)$ e analisar conectividade.

Canny

Supressão dos não máximos



Algorithm

For each pixel (x, y) do:

if $magn(i, j) < magn(i_1, j_1)$ or $magn(i, j) < magn(i_2, j_2)$
then $I_N(i, j) = 0$
else $I_N(i, j) = magn(i, j)$

Canny

Limiarização por Histerese

- Usar dois limiares:
 - Um limiar baixot_l
 - Um limiar altot_h (normalmente, $t_h = 2t_l$)

Algorithm

1. Produce two thresholded images $I_1(i, j)$ and $I_2(i, j)$.

(note: since $I_2(i, j)$ was formed with a high threshold, it will contain fewer false edges but there might be gaps in the contours)

2. Link the edges in $I_2(i, j)$ into contours

2.1 Look in $I_1(i, j)$ when a gap is found.

2.2 By examining the 8 neighbors in $I_1(i, j)$, gather edge points from $I_1(i, j)$ until the gap has been bridged to an edge in $I_2(i, j)$.

Exemplo - qual o efeito de σ ?

(left:Sobel, middle: thresh=35, right: thersh=50)

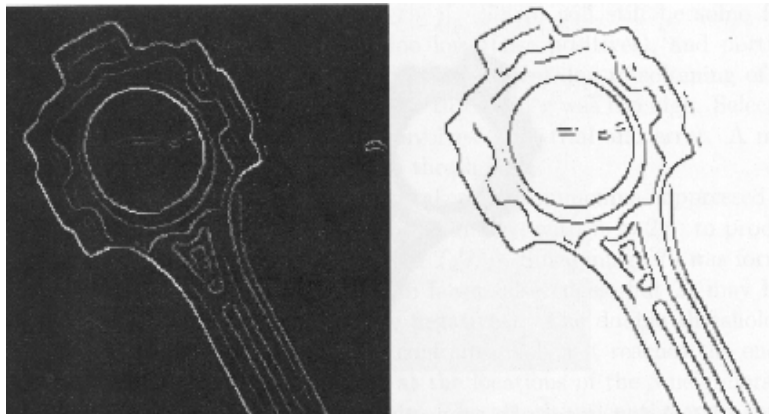


(Canny - left: $\sigma=1$, middle: $\sigma=2$, right: $\sigma=3$)

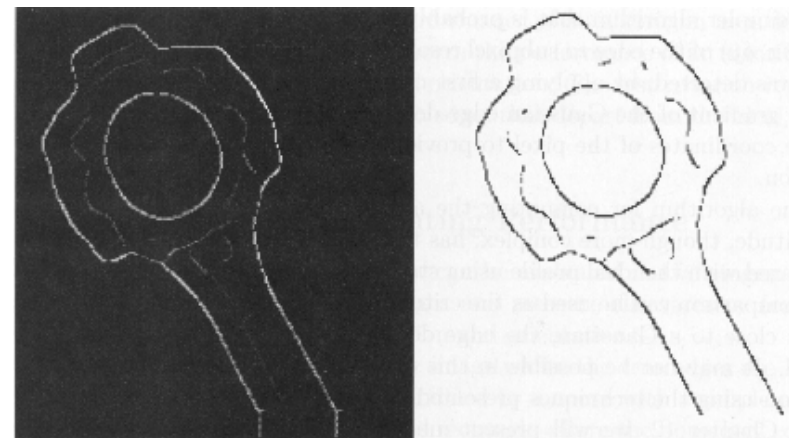


Exemplo - qual o efeito de σ ?

(Canny - 7x7 Gaussian, more details)



(Canny - 31x31 Gaussian, less details)





Próxima aula...

- Análise de Texturas