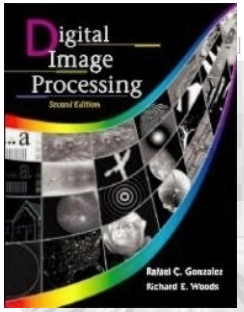


Aula 10.2

Segmentação de Imagens

Técnicas mais avançadas para detecção de bordas

Os métodos de detecção de bordas já discutidos baseiam-se simplesmente na filtragem de uma imagem com uma ou mais máscaras, sem levar em consideração informações referentes às características da borda ou ruído

Técnicas mais elaboradas para detecção de bordas

Merecem destaque as técnicas de:

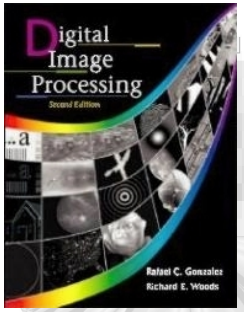
- Marr e Hildreth (1980)
- Canny (1986)

Marr-Hildreth

Técnicas de Marr e Hildreth

Marr e Hildreth consideram:

1) que as mudanças de intensidade não são independentes da escala da imagem e, portanto, sua detecção requer o uso de operadores de diferentes tamanhos
2) que uma mudança súbita de intensidade dá origem a um pico ou um vale na primeira derivada ou, ainda, a um cruzamento por zero da segunda derivada

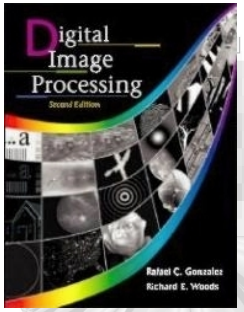


Marr-Hildreth

Assim, um operador usado para a detecção de bordas deve ter duas características principais:

1) Deve ser um operador diferencial, capaz de calcular uma aproximação digital da primeira ou segunda derivada em cada ponto na imagem

2) Deve ser capaz de agir em qualquer escala desejada, de modo que os grandes operadores possam ser usados para detectar bordas borradas, e os pequenos operadores, para detectar detalhes finos com foco nítido



Marr-Hildreth

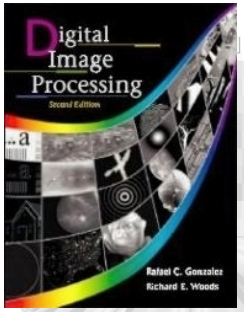
Marr e Hildreth

Algoritmo

consiste na convolução do filtro LoG com uma imagem de entrada, $f(x,y)$

$g(x, y) = [\nabla^2 G(x, y)] * f(x, y)$

e, então, encontrar o cruzamento por zero de $g(x,y)$ para determinar a localização das bordas em $f(x,y)$



Marr-Hildreth

ou ainda:

O algoritmo pode ser resumido da seguinte forma:

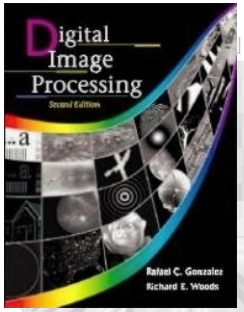
1. Filtrar a imagem de entrada com um filtro $n \times n$ gaussiano passa-baixa

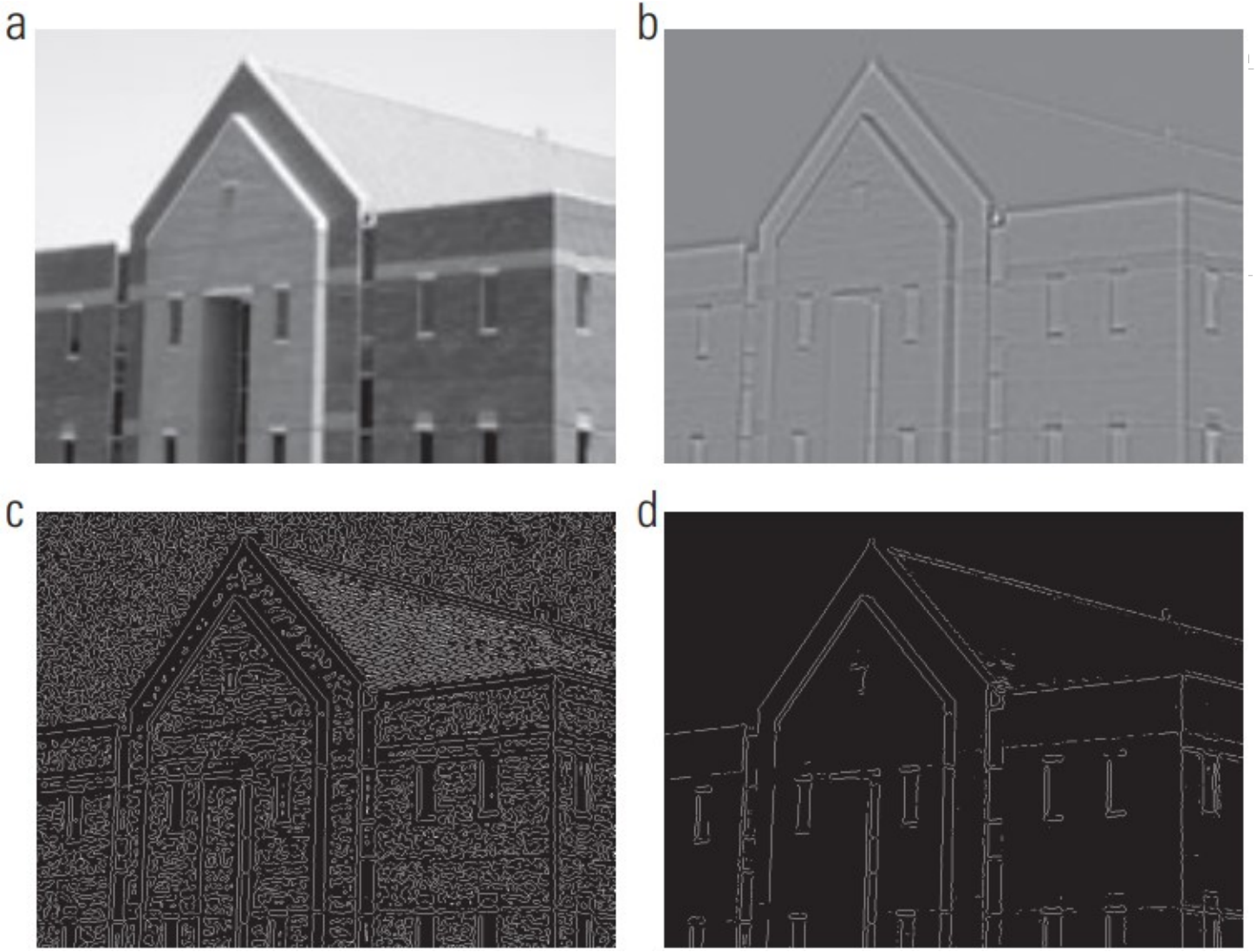
$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

2. Calcular o laplaciano da imagem resultante da Etapa 1, utilizando, por exemplo, a máscara 3×3 na (Os passos 1 e 2 utilizam a Equação $g(x,y) = \nabla 2[G(x,y)^* f(x,y)]$)

1	1	1
1	−8	1
1	1	1

3. Encontrar o cruzamento por zero da imagem obtida na Etapa 2



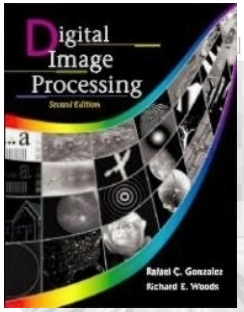


(a) Imagem original

(b) Resultado das etapas 1 e 2 de Marr-Hildreth com $\sigma = 4$ e $n = 25$

(c) Cruzamentos por zero de (b) usando o valor 0 como limiar
(repare nas bordas parecidas com contornos fechados)

(d) Cruzamentos por zero encontrados utilizando um limiar igual a 4% do maior valor de pixel encontrado em (b) → **Observe as bordas finas**



Canny

Detector de Canny

Embora seja mais complexo, o desempenho do detector de bordas de Canny [Canny (1986)] discutido nesta seção é superior, em geral, aos detectores de borda discutidos até agora

Canny

A abordagem de Canny

baseia-se em três objetivos básicos:

1. *Baixa taxa de erro* - Todas as bordas deverão ser encontradas e não deve haver respostas espúrias

As bordas detectadas devem ser o mais próximas possível das bordas verdadeiras

Canny

2. *Os pontos de borda devem estar bem localizados* - As bordas detectadas devem ser o mais próximas possível das bordas verdadeiras

A distância entre um ponto marcado como uma borda pelo detector e o centro da borda verdadeira deve ser mínima

Canny

3. *Resposta de um único ponto de borda* - O detector deve retornar apenas um ponto para cada ponto de borda verdadeiro

O número de máximos locais em torno da borda verdadeira deve ser mínimo

Isso significa que o detector não deve identificar múltiplos pixels de borda em que apenas um único ponto de borda existe

Canny

O algoritmo de detecção de bordas de Canny é composto pelas seguintes etapas básicas:

1. Suavizar a imagem de entrada com um filtro gaussiano

2. Calcular a magnitude do gradiente e os ângulos das imagens

3. Aplicar a supressão não máxima na imagem da magnitude do gradiente

4. Usar a dupla limiarização e a análise de conectividade para detectar e conectar as bordas

Canny

Detector de Canny

O detector de bordas de Canny usa um algoritmo de multiplos estágios para obter bordas de boa qualidade

- finas

- sem quebras

Canny

passos do detector:

1 – Redução de ruído

Aplica um filtro gaussiano

2 – Cálculo dos gradientes

Sobel

Canny

3 – Afinamento das bordas

Canny usa a Supressão não máxima, que considera a intensidade da borda

4 – Threshold duplo

Usa dois limiares: menor e maior

pixels acima do maior são mantidos

pixels abaixo do menor são descartados

pixels entre menor e maior são analisados no passo 5

Canny

5 – Histeresys (transforma pixels fracos em fortes)

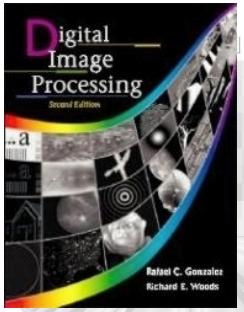
Se o pixel p tem tom maior que o limiar superior, é mantido

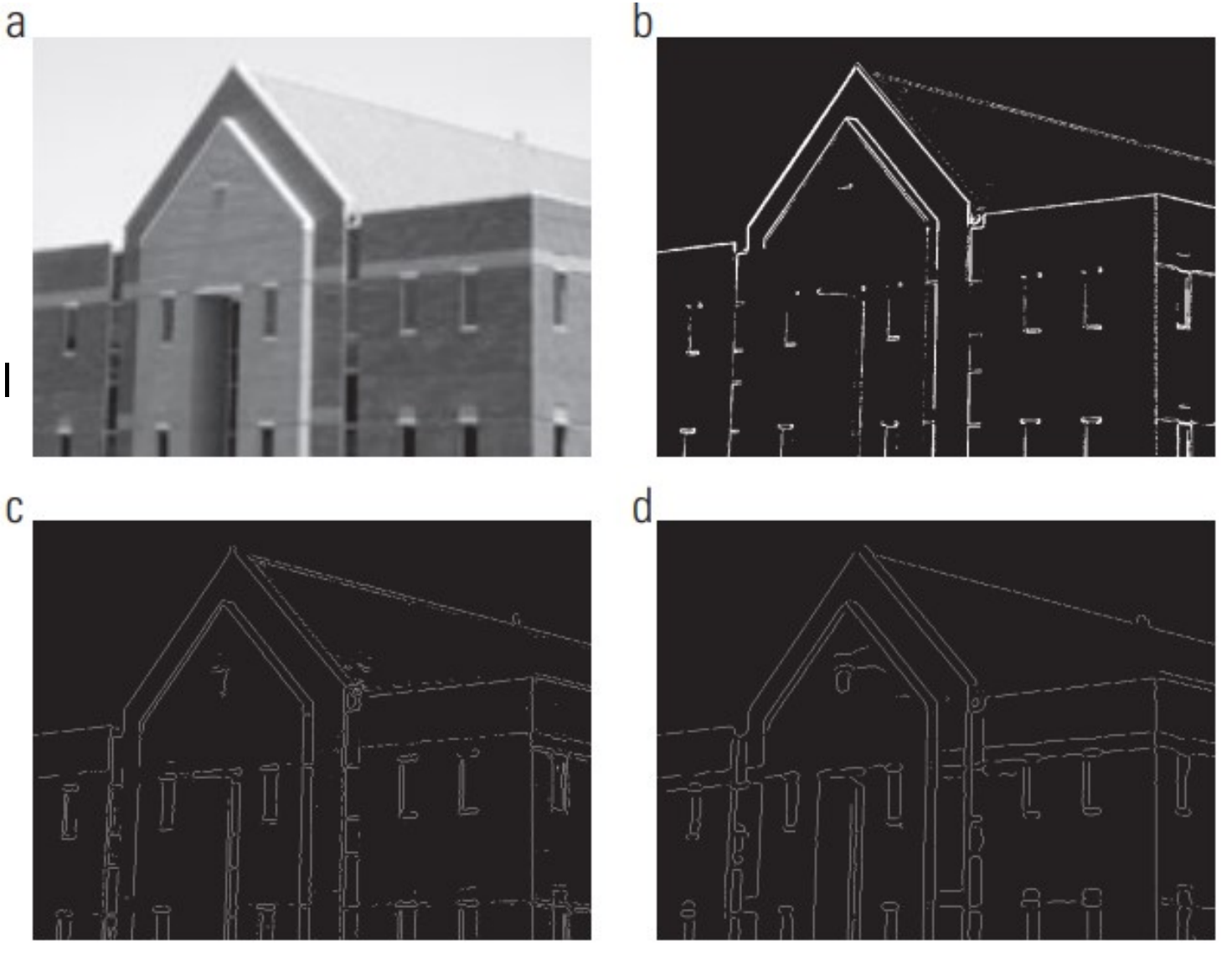
Se o pixel p tem tom menor que o limiar inferior, é apagado

Os pixels p com intensidade entre limiar inferior e limiar superior são analisados com relação à sua vizinhança p-1 e p+1

Se ele está entre vizinhos acima do limiar superior, ele é mantido (Isto evita a quebra exagerada das linhas)

p-1	p	p+1





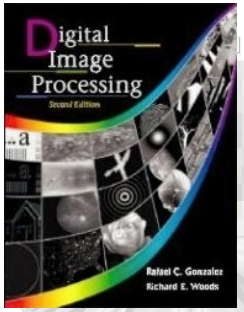
(a) Imagem original

(b) Gradiente limiarizado da imagem suavizada

(c) Imagem obtida utilizando Marr-Hildreth

(d) Imagem obtida utilizando o algoritmo de Canny

Observe a melhora significativa da imagem de Canny em comparação às outras duas

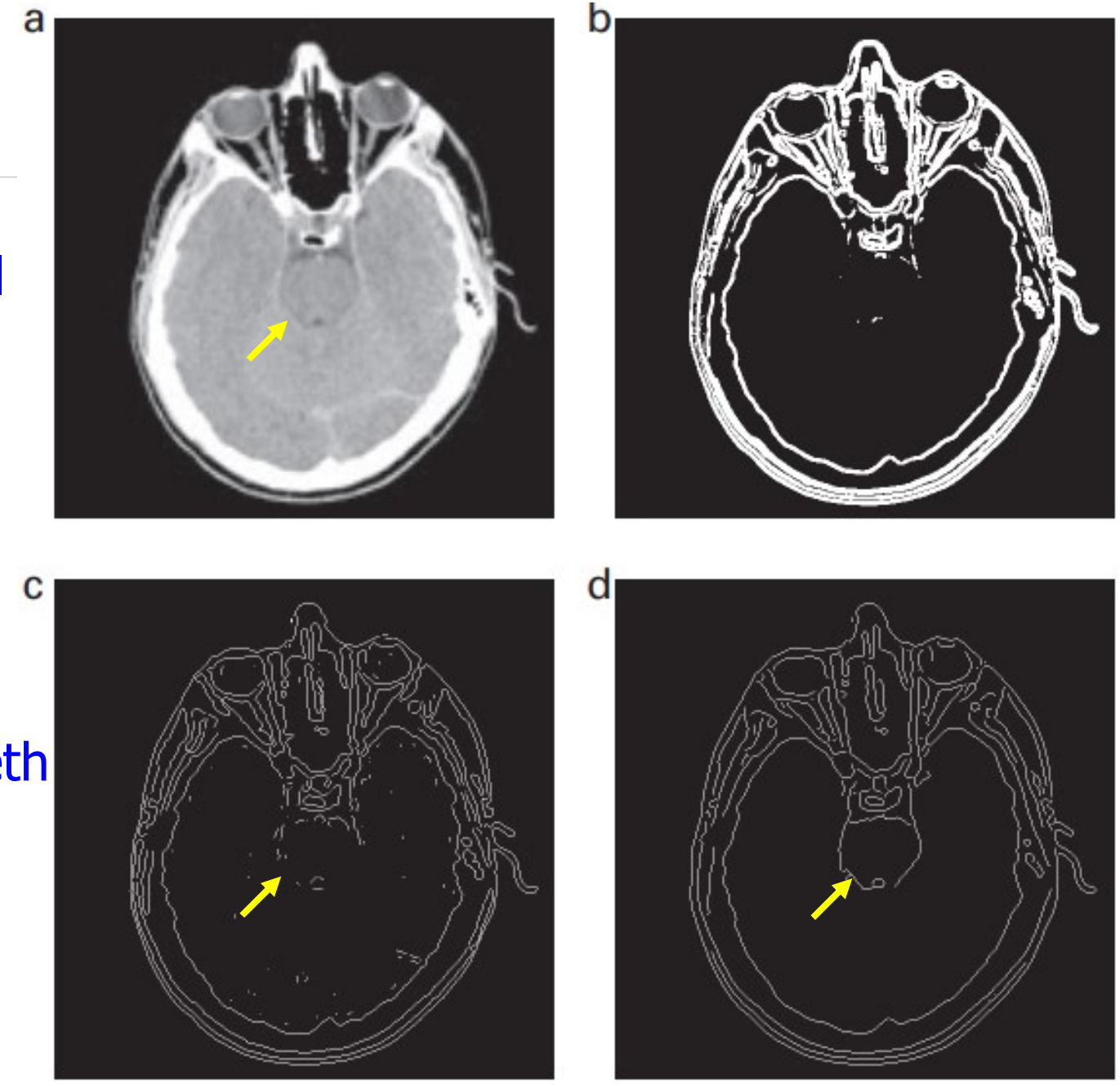


(a)Imagem original

(b) Gradiente limiarizado da imagem suavizada

(c) Imagem obtida usando Marr-Hildreth

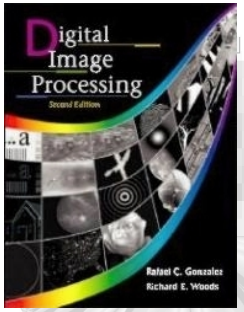
(d) Imagem obtida usando Canny



Canny

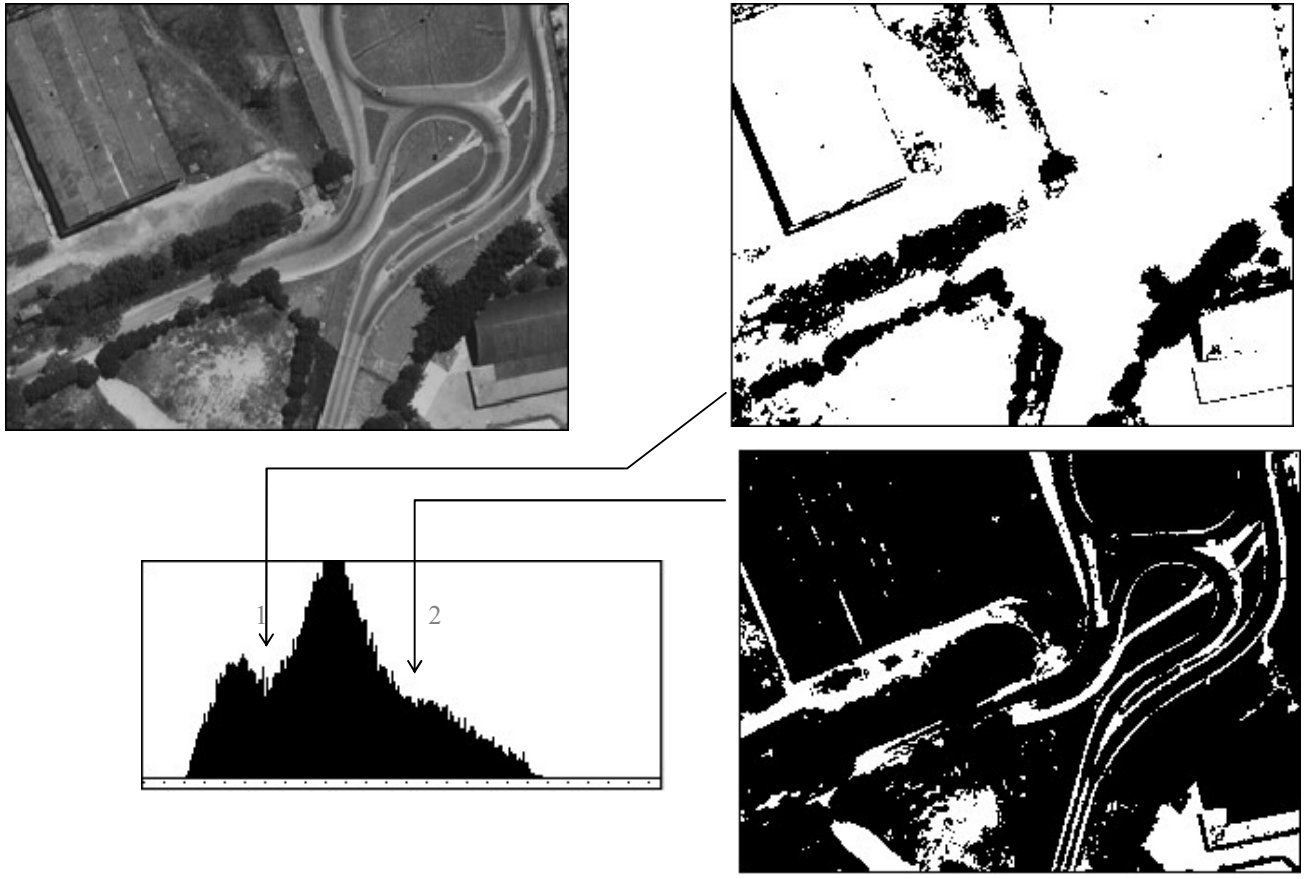
Sobel, afinado e limiarizado (muitas bordas quebradas e muitas bordas pequenas)

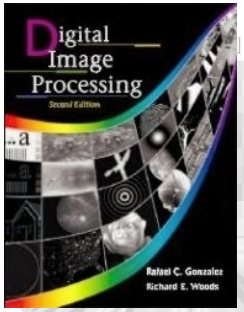
Canny – Suavizado, Sobel e afinado e limiarizado duplo e Histeresys (bordas sem quebras e sem as bordas pequenas)



Segmentação de Imagens

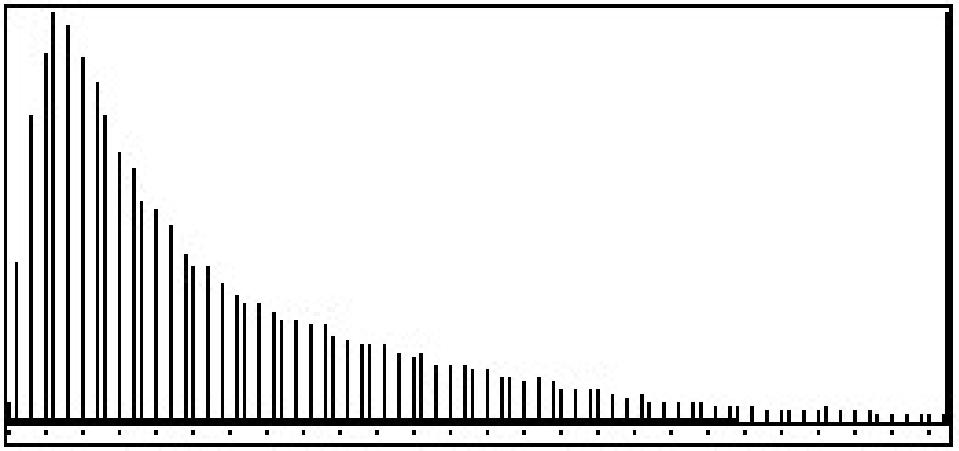
Em vista dos problemas com os algoritmos baseados em descontinuidade que particionam a imagem baseado nas mudanças dos níveis de cinza

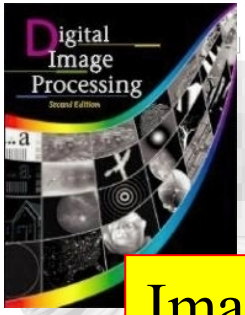




Segmentação de Imagens

A Limiarização de Bordas pode ser um solução, pois as imagens de bordas possuem apenas duas classes:
- branco é borda
- preto é fundo





Segmentação de Imagens – linhas retas

Imagem Original

(1)

Suavização (*Smoothing*) - reduzir variações exageradas, que produzem bordas falsas na imagem (é preciso preservar as bordas verdadeiras);

(2)

Detecção de Bordas (*Edge Detection*) - aplicação de um detector de bordas;

(3)

Limiarização (*Thresholding*) - elimina bordas de baixa magnitude (simplificando o processamento posterior);

(4)

Afinamento de bordas (*Thinning*) - bordas com uma espessura de mais de um pixel precisam ser afinadas, para uma melhor definição de sua verdadeira localização;

(5)

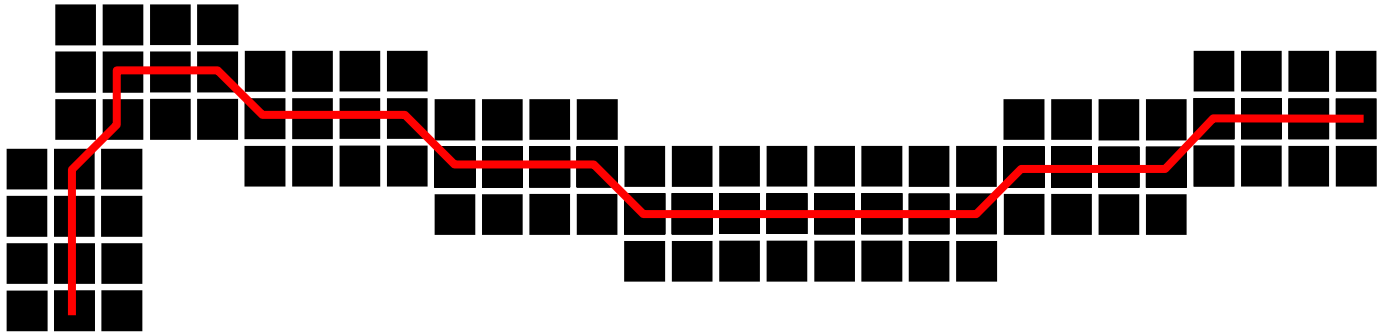
Ligação (*Linking*) – conexão dos pixels que compõem uma linha única;

Segmentação de Imagens

afinamento

- Afinamento de bordas *(Thinning)* – bordas com uma espessura de mais de um pixel precisam ser afinadas, para uma melhor definição de sua verdadeira localização;
- O método mais conhecido é o de **Zhang-Suen**, que opera com imagens binárias, baseados no eixo central, a ser visto na aula 11

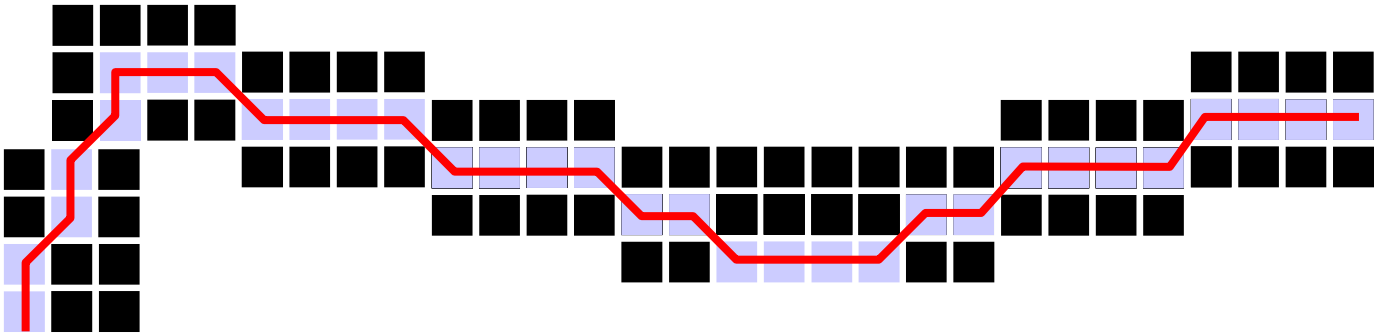
Borda Binária

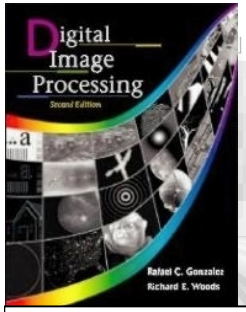


Segmentação de Imagens

Afinamento por supressão Não máxima

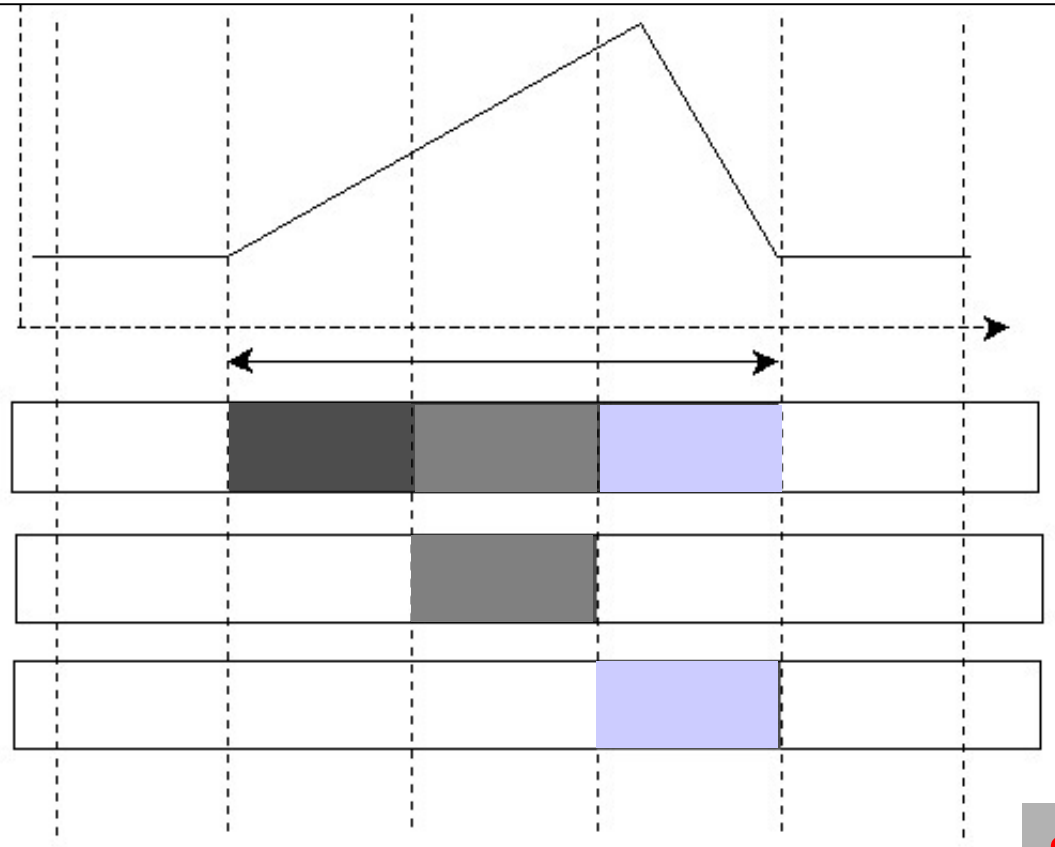
- Mantém o pixel de borda com a tonalidade mais alta **(pixel mais forte)**, mesmo que ele não esteja no centro da borda





Afinamento por supressão não máxima

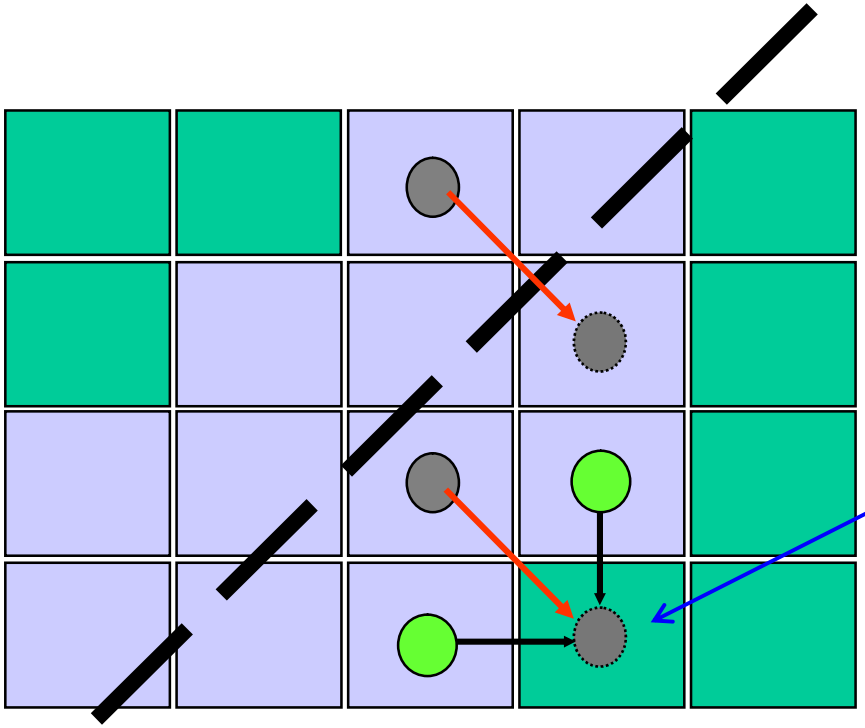
- Procede-se um processamento sobre todos os pixels da imagem, de forma a eliminar os pixels vizinhos pertencentes a uma borda no sentido perpendicular a borda (segundo a direção do ponto)



Supressão Não Máxima

Segmentação de Imagens

Afinamento



Se o ponto preto tem magnitude menor que o ponto cinza, então o ponto preto será apagado

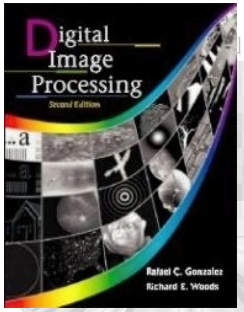
ponto cinza = resultante entre os pontos indicados por verdes (interpolação)

Segmentação de Imagens

Ligação de bordas e detecção de fronteiras

Processamento que segue a detecção de bordas, com o objetivo de tornar os contornos mais regulares, ligando pontos isolados e pequenos segmentos por causa de problemas decorrentes de ruído

Em um próximo nível, podem ser formados objetos mais complexos, como linhas retas, círculos e elipses



Segmentação de Imagens

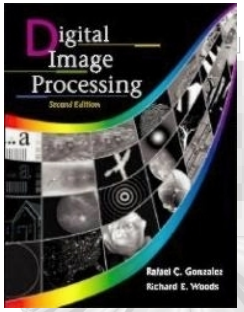
Ligação de bordas e detecção de fronteiras

Processamento Local

Neste processo, são investigadas pequenas vizinhanças dos pixels, e aqueles que apresentam informações similares são conectados (recebem um rótulo) no mesmo objeto (por exemplo, uma linha)

Algumas informações importantes usadas no processo são:

- magnitude da borda
- ângulo da borda



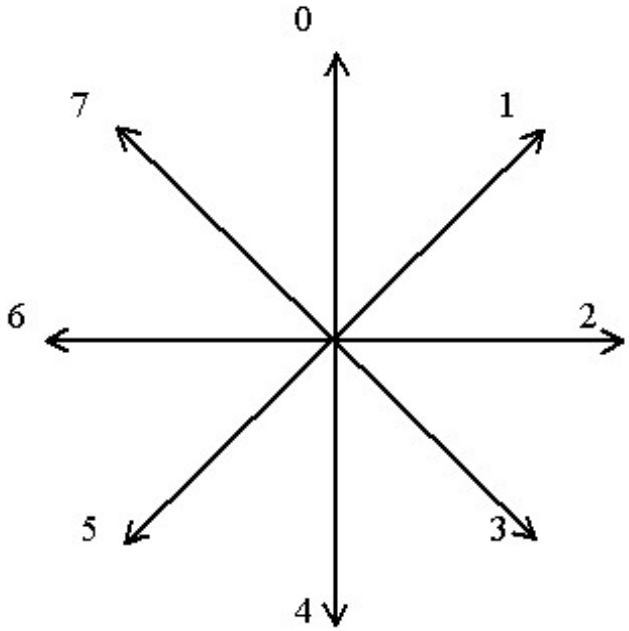
Segmentação de Imagens

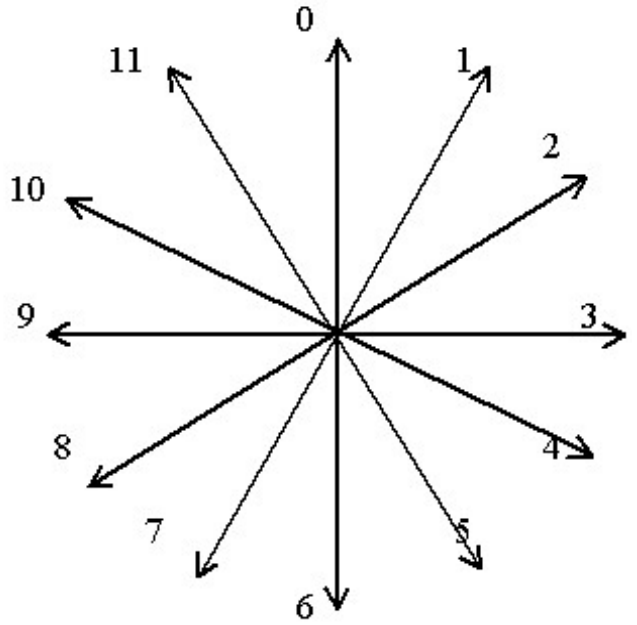
CONEXÃO (LINKING)

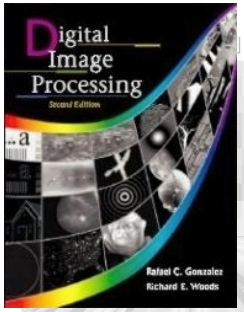
- Método de Conexão por varredura e rotulação - consiste na varredura e rotulação dos pixels;
- Pixels de mesma direção e vizinhos recebem um mesmo rótulo;
- Pode-se verificar a magnitude também;
- Utiliza uma estrutura de dados para guardar as informações das retas
(ponto inicial, ponto final, direção, número de pixels);
- Percorre a imagem da esquerda para a direita e de cima para baixo;
- É verificada a vizinhança e caso seja encontrado algum vizinho já rotulado, o pixel atual recebe então o mesmo;
- Uma simplificação é controlar a busca na vizinhança, conforme sugerem

Segmentação de Imagens

CONEXÃO (LINKING)



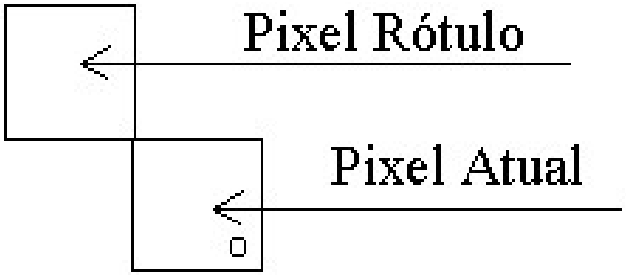


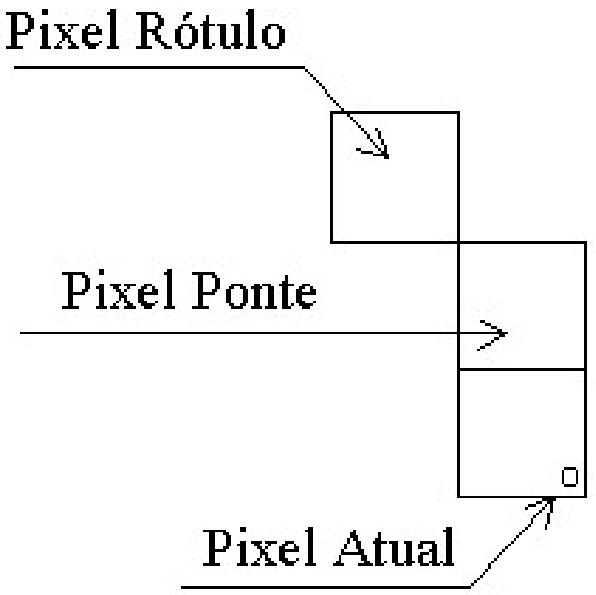


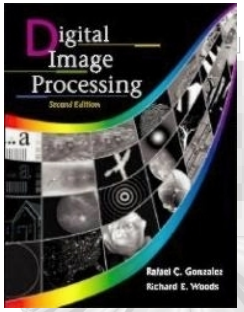
Segmentação de Imagens

CONEXÃO (LINKING)

- Classificação dos pixels envolvidos no processo de conexão



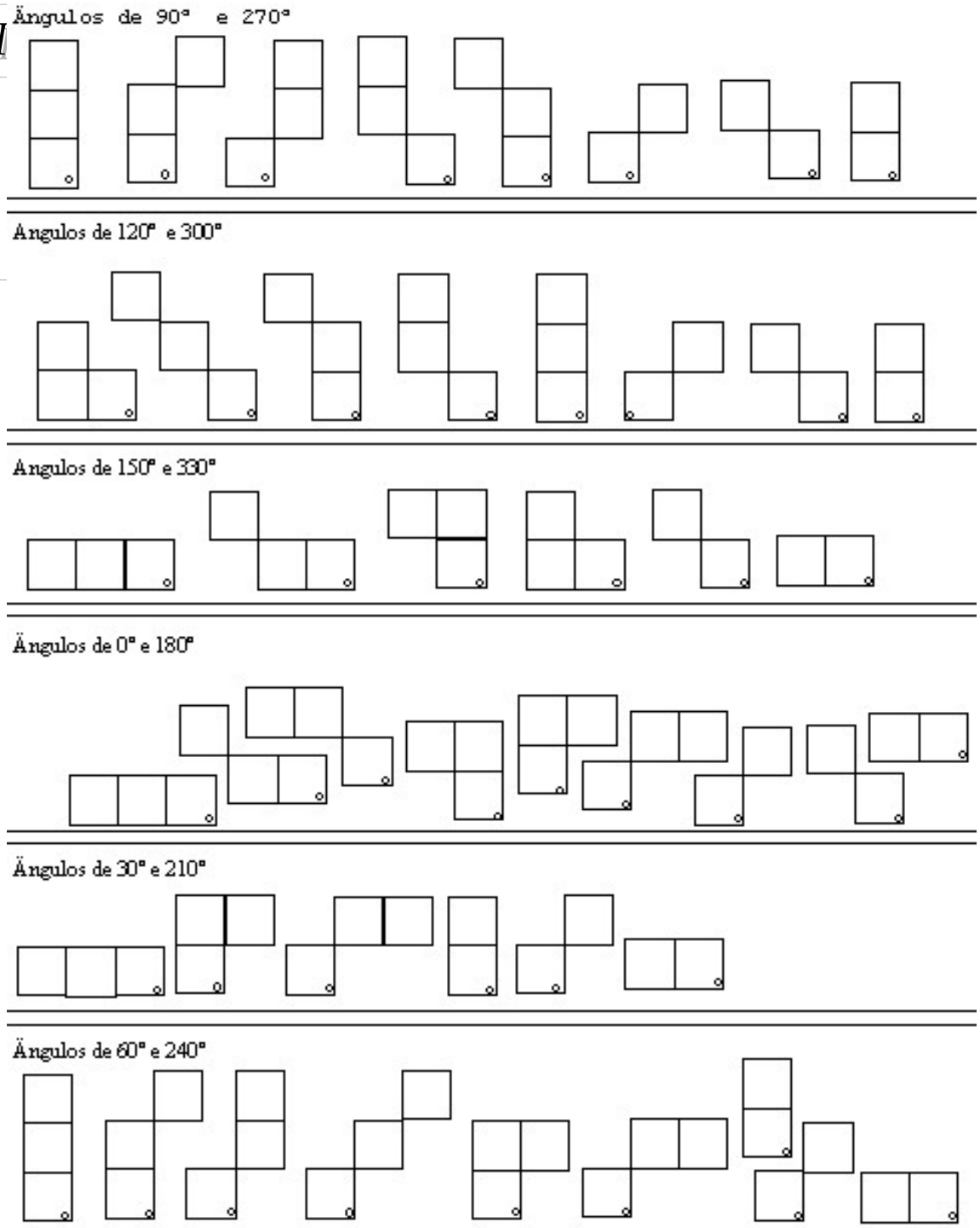


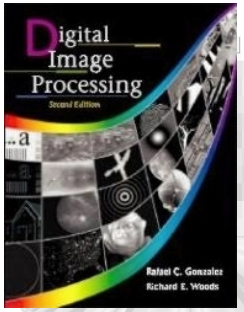


segmentação de imagens

CONEXÃO (LINKING)

- Vizinhança examinada no processo de varredura e rotulação





CONEXÃO (LINKING)

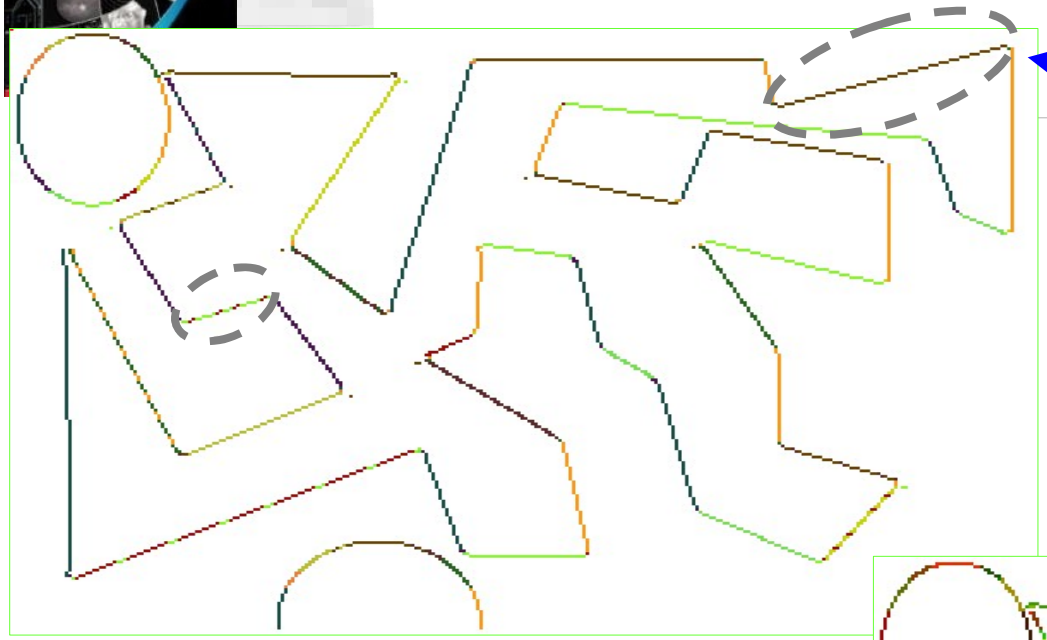
- Esta técnica permite a utilização de uma estrutura para armazenar os parâmetros dos segmentos de retas construídos

```
    struct linha
{
  int xi;    a[i].xi  ( coord x do ponto inicial da reta i  )
  int yi;    a[i].yi  ( coord y do ponto inicial da reta i  )
  int xf;    a[i].xf  ( coord x do ponto final da reta i    )
  int yf;    a[i].yf  ( coord y do ponto final da reta i    )
  int dir;   a[i].dir ( direção da reta i                   )
  int n;     a[i].n   ( número de pixels na reta i          )
};

        struct linha a[max_rot];  ( número máximo de linhas em a  )
```

- O último elemento anexado à reta deve ser o ponto inicial ou final da reta.

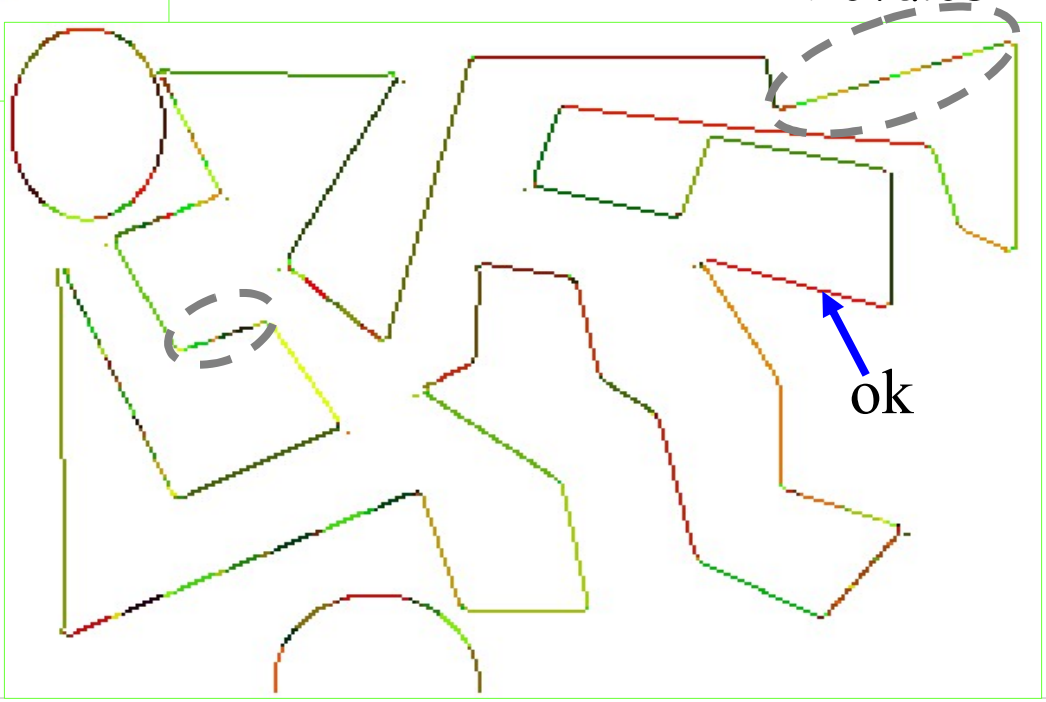
Segmentação de Imagens



ângulos iguais para estes pixels

por causa da implementação do rotulador, receberam rótulos diferentes

Rótulos

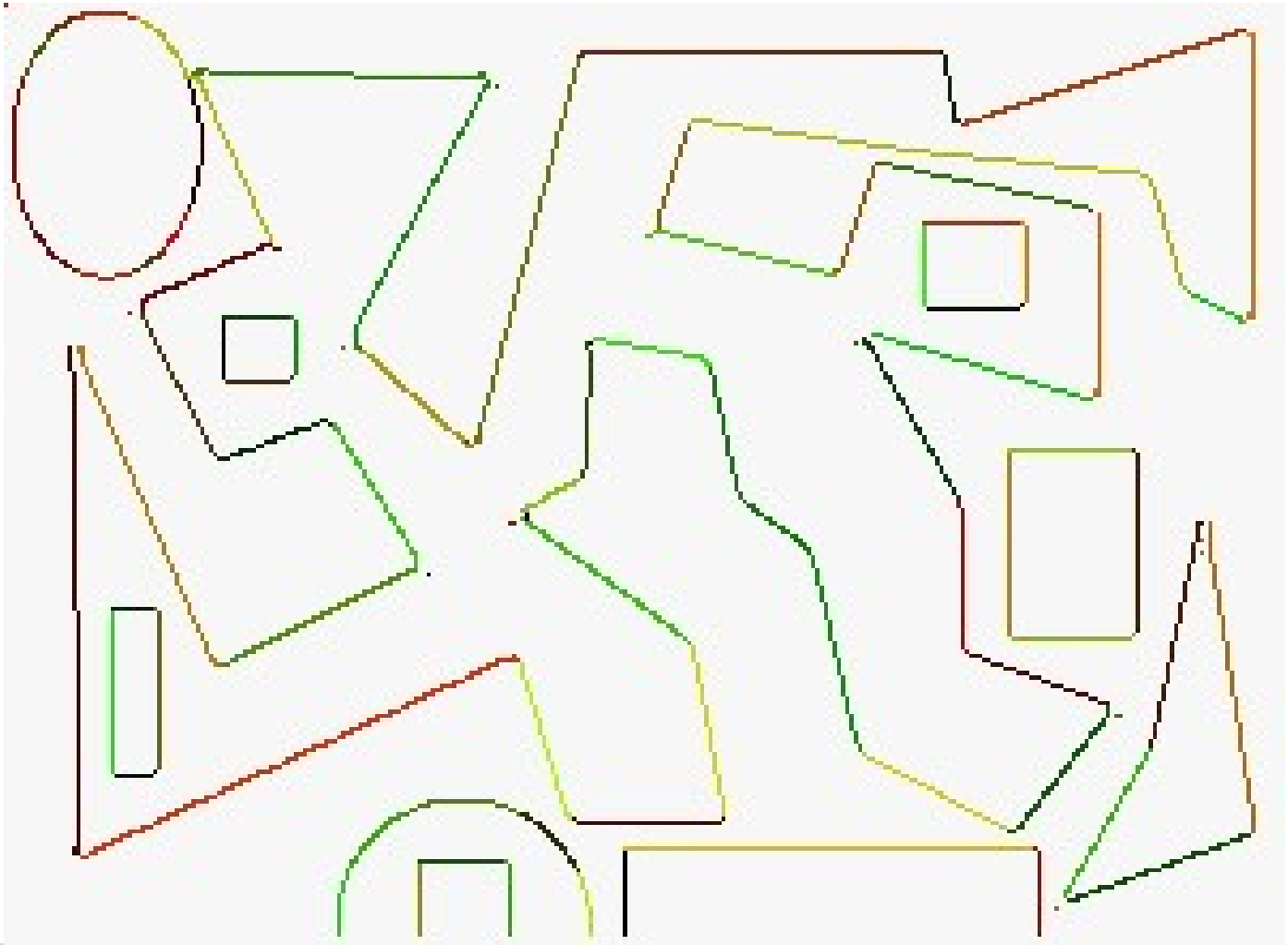


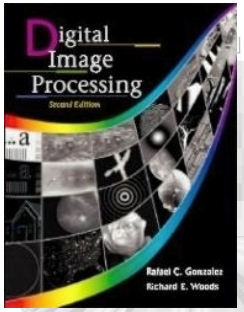
Ângulos

Imagem original

Imagem dos rótulos após a conexão de segmentos colineares

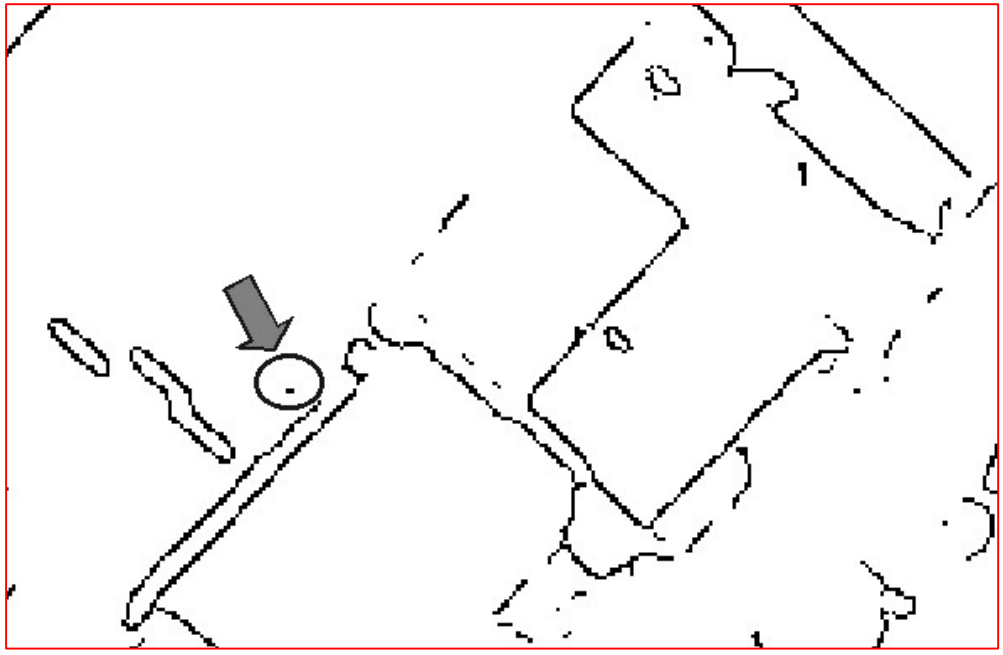
Troca os rótulos de segmentos colineares, para torná-los um só



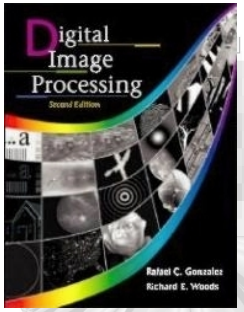


CONEXÃO (LINKING)

- Eliminação de linhas insignificantes



Linhas obtidas pelo processo de varredura e rotulação



CONEXÃO (LINKING)

- Uma outra proposta para a etapa de conexão
  - Inundação (Seed-Fill)

• Apresenta bons resultados na rotulação;

• Implementação extremamente simples (usando recursividade - vantagens e desvantagens)

• Identificação dos pontos inicial e final da reta ainda não implementada com sucesso para todos os casos devido o processo de inundação não apresentar uma direção bem definida de encaminhamento;

• Uma outra alternativa é identificar o início e o fim da reta na etapa de ajustamento

Demonstração usando o b27 →

Ligação de bordas e detecção de fronteiras

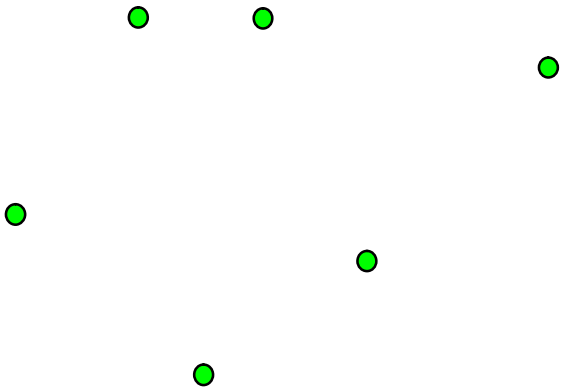
Processamento Global

Neste processo, os objetos linhas, círculos, etc. são obtidos observando as relações globais entre os pixels

O método mais conhecido é a **Transformada de Hough**

Transformada de Hough para círculos

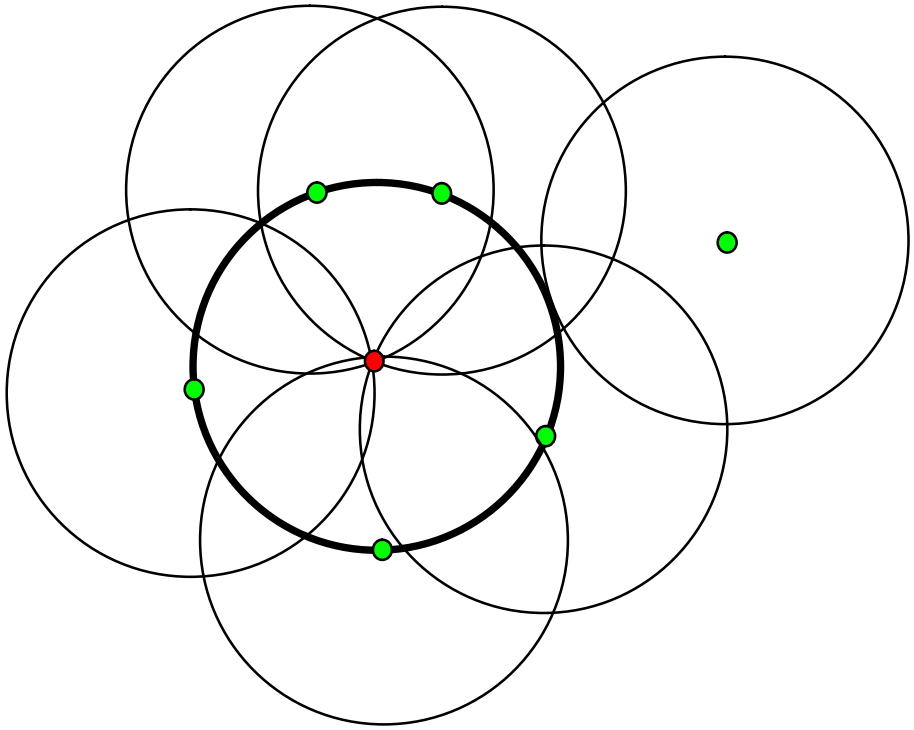
Dado um conjunto de pontos



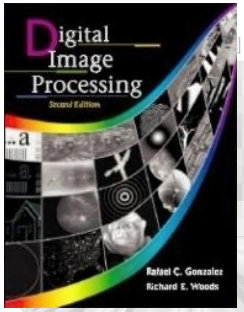
Como encontrar a circunferência que passa por eles?

Transformada de Hough para círculos

Para cada ponto, trace uma circunferência centrada nele, com o raio desejado (experimente vários)

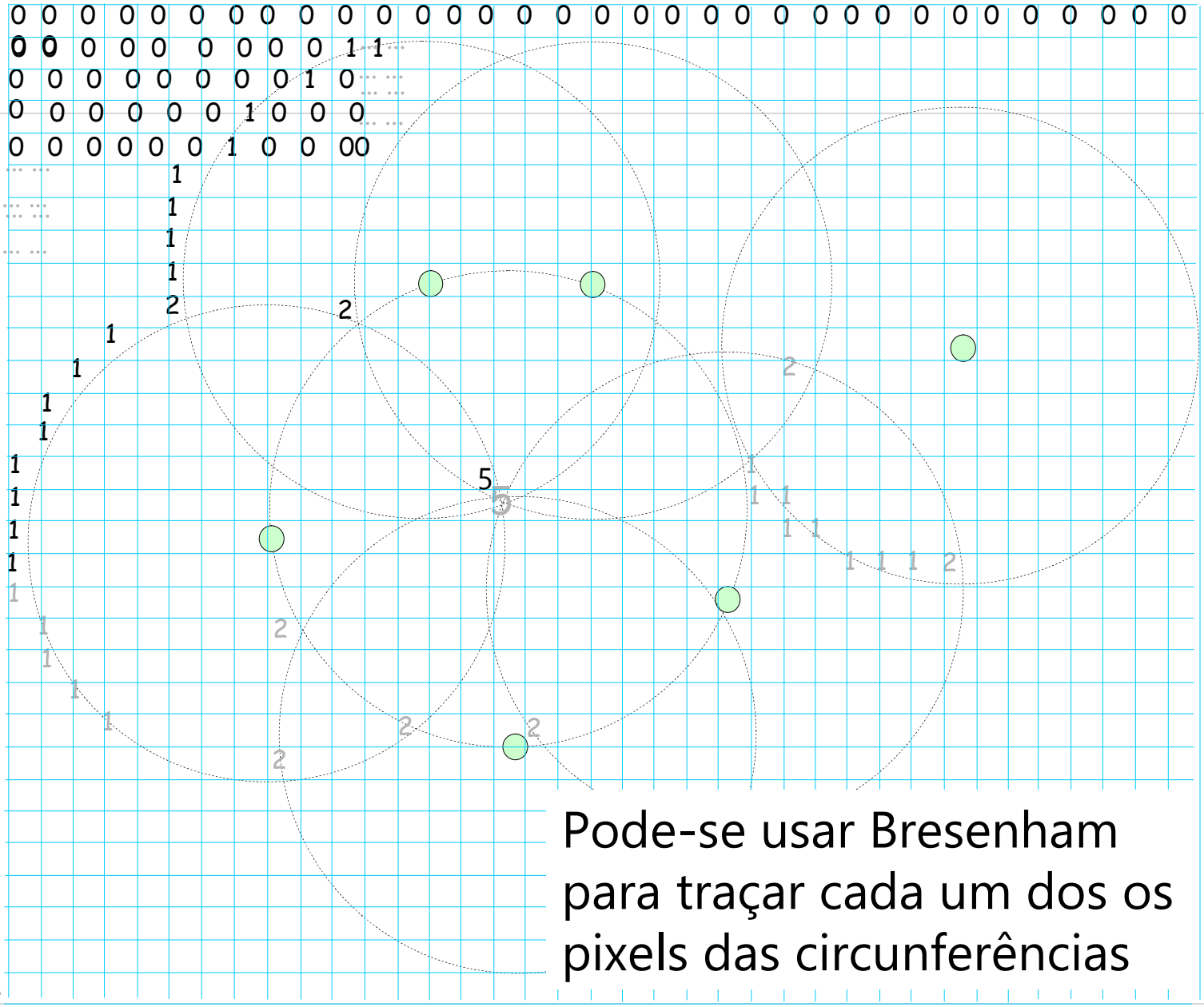


o ponto de maior inteserção destas circunferências determina o centro da circunferência que passa pelos pontos iniciais

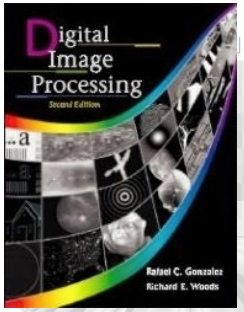


Transformada de Hough para círculos

Necessidade de uma estrutura capaz de identificar os locais de interseção

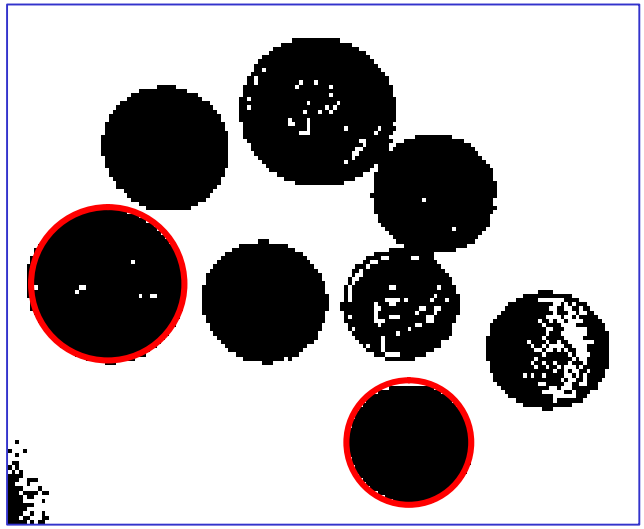


Pode-se usar Bresenham para traçar cada um dos os pixels das circunferências

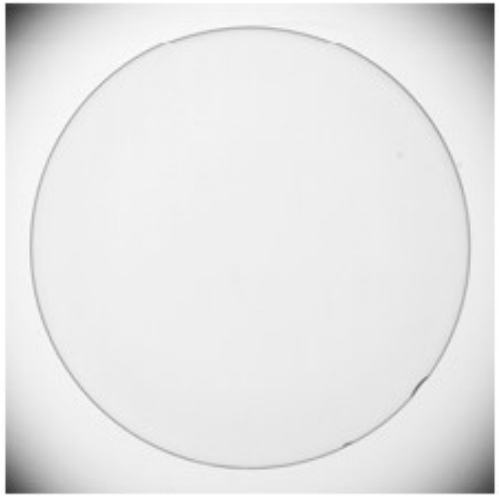


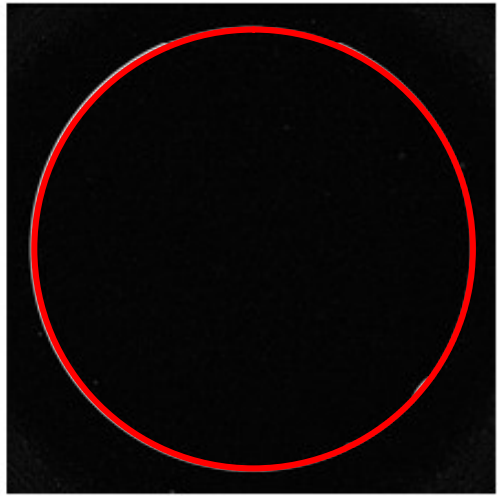
Transformada de Hough - Círculos

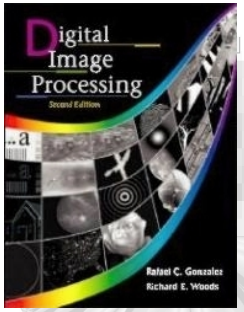
Algumas aplicações



Na implementação, é preciso testar alguns raios diferentes, até encontrar um acumulador com grande valor







Transformada de Hough - Círculos

Algumas aplicações