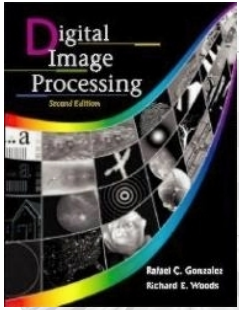


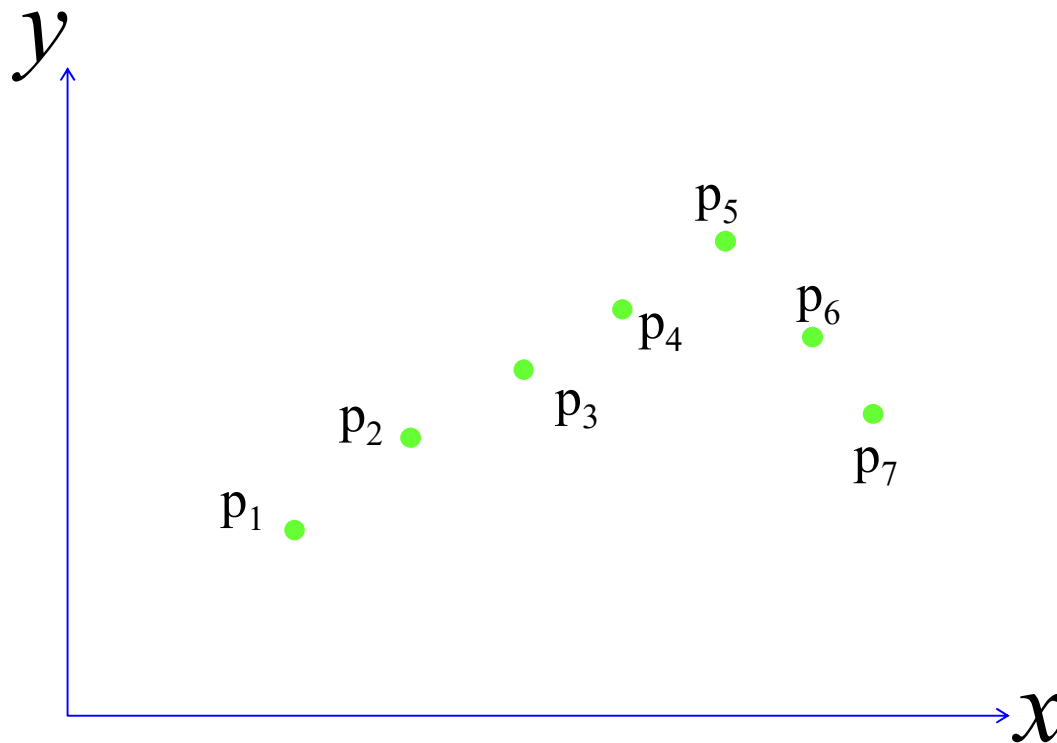
Aula 10.3

Segmentação de Imagens

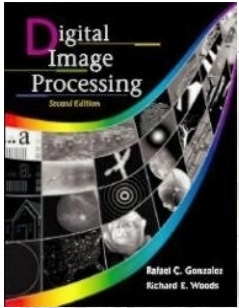


Transformada de Hough - Linhas Retas

Dado um conjunto de pontos

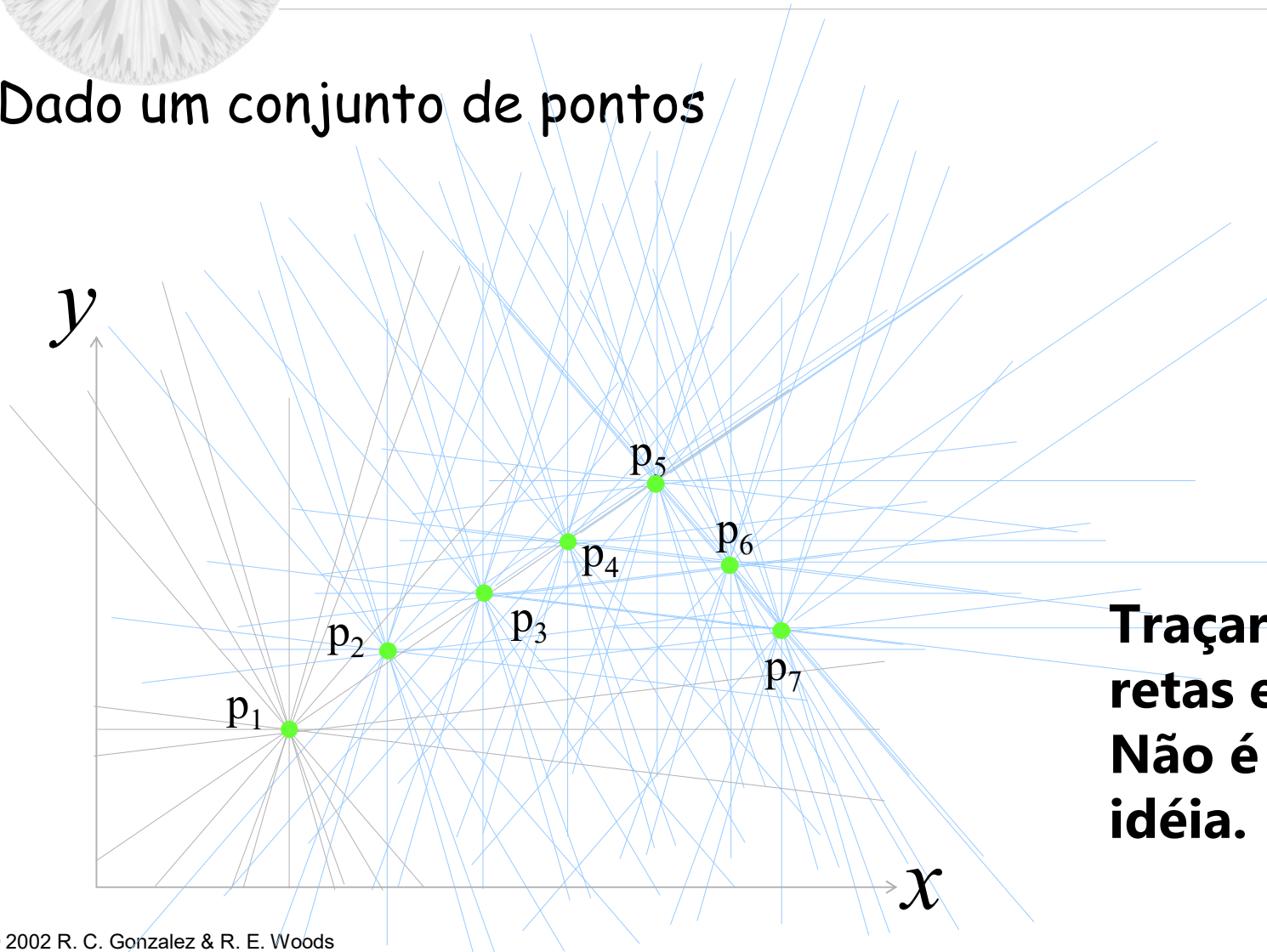


como determinar
a reta que passa
por eles?



Transformada de Hough - Linhas Retas

Dado um conjunto de pontos



**Traçar várias
retas entre eles?
Não é uma boa
idéia.**



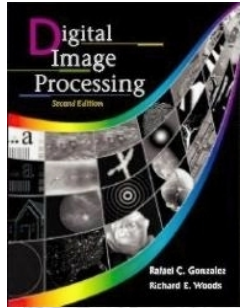
A proposta de Hough

Dada a equação da reta com os parâmetros x e y

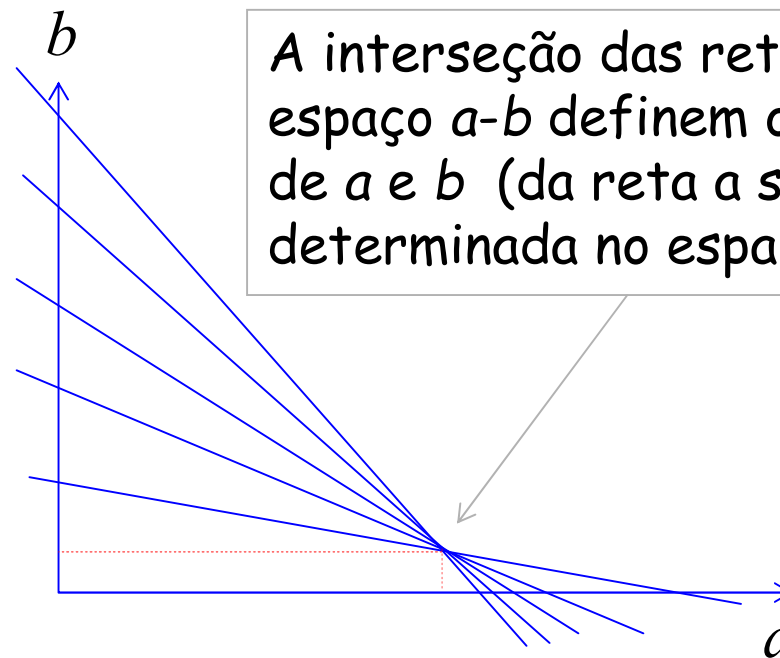
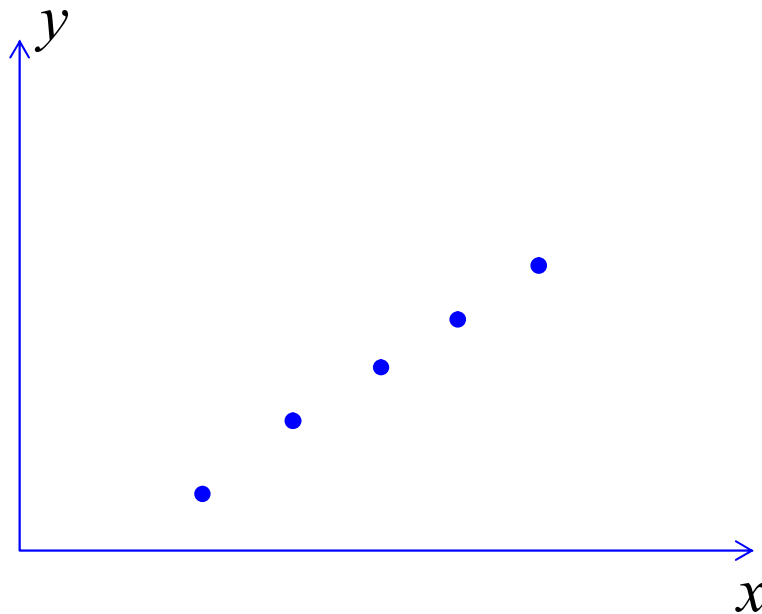
$$y = ax + b$$

Cada ponto p_i , de coordenadas (x_i, y_i) , gera uma equação (parâmetros a e b .)

$$b = y_i - ax_i$$

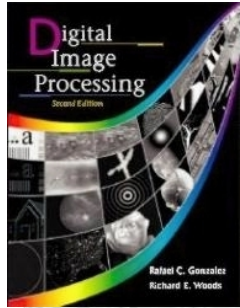


Transformada de Hough



A interseção das retas no novo espaço a - b definem os valores de a e b (da reta a ser determinada no espaço x - y).

Necessidade de uma estrutura capaz de identificar os locais de interseção - ACUMULADOR



Transformada de Hough

- Algoritmo → Transformada de Hough - (espaço a, b)

1 - Faça acumuladores $A(a, b) = 0$;

(Quantize o espaço do parâmetros (a, b) adequadamente;)

2 - Diferencie a imagem utilizando Operador de Sobel

(a) $g_x = \text{gradiente-x}$ (b) $g_y = \text{gradiente-y}$

3 - Se magnitude da borda é maior que threshold, calcule o gradiente

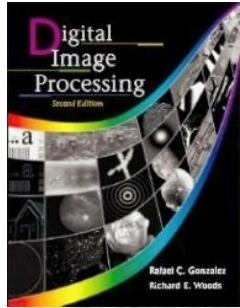
$a = (g_y / g_x)$;

4 - Calcule $b = -ax_i + y_i$;

5 - Incremente o acumulador, $A(a, b) = A(a, b) + 1$;

6 - Repita (3) até (5) para todos os pontos da imagem;

7 - Os picos no acumulador $A(a, b)$ fornecem os gradientes da linha e as interseções;



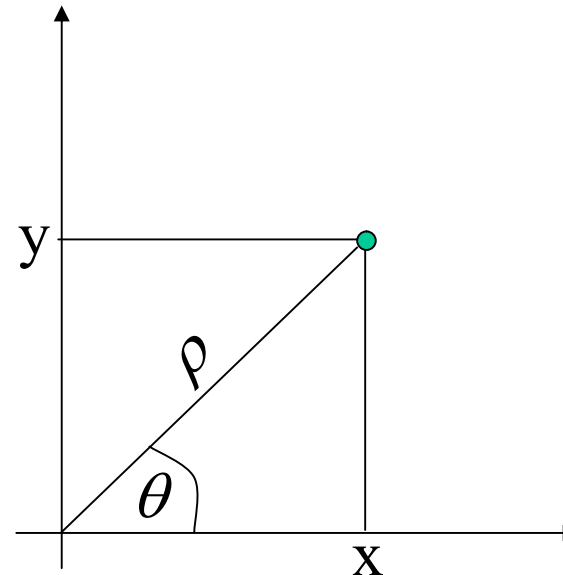
Transformada de Hough

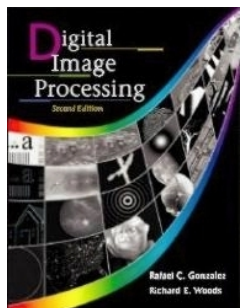
Problemas com retas verticais

($a \rightarrow +\infty$ ou $a \rightarrow -\infty$)

Uma nova parametrização (forma polar)

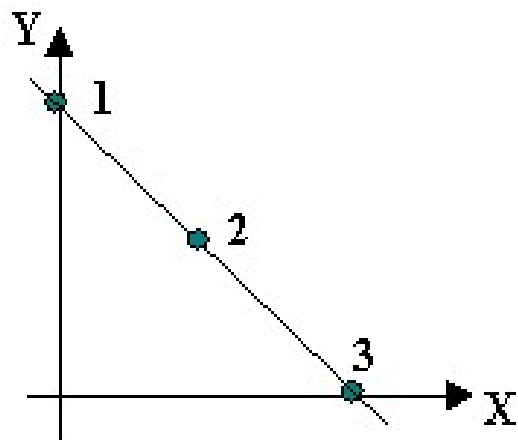
$$\rho = x \cos\theta + y \sin\theta$$



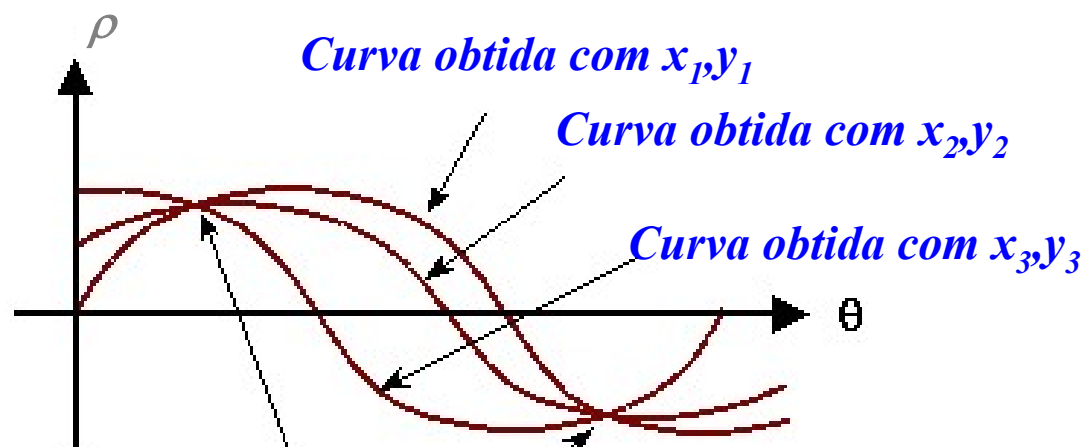


Transformada de Hough

Os valores de x e y são conhecidos (coordenadas dos pixels), assim, para cada valor de θ , é calculado o valor de ρ e um novo gráfico é obtido, desta vez tendo como parâmetros os elementos ρ e θ .



Pontos no espaço $x-y$



Os pontos de interseção determinam os valores de ρ e θ , para a linha que passa pelos pontos no espaço $x-y$



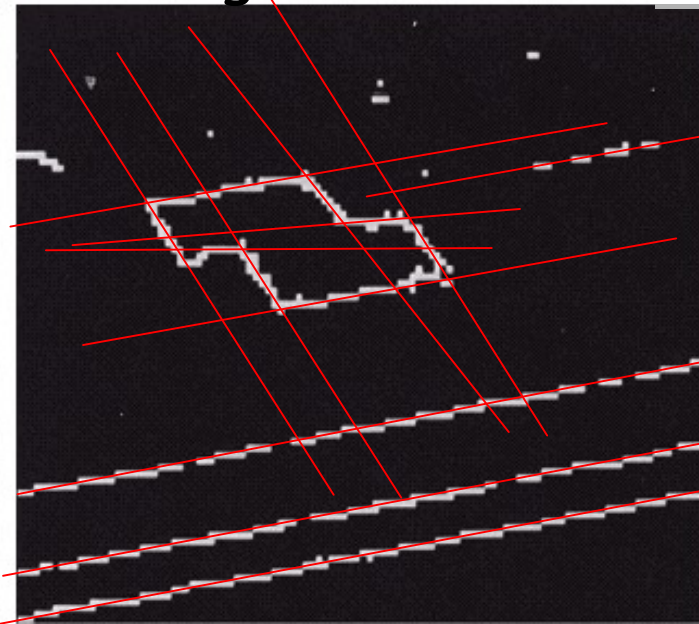
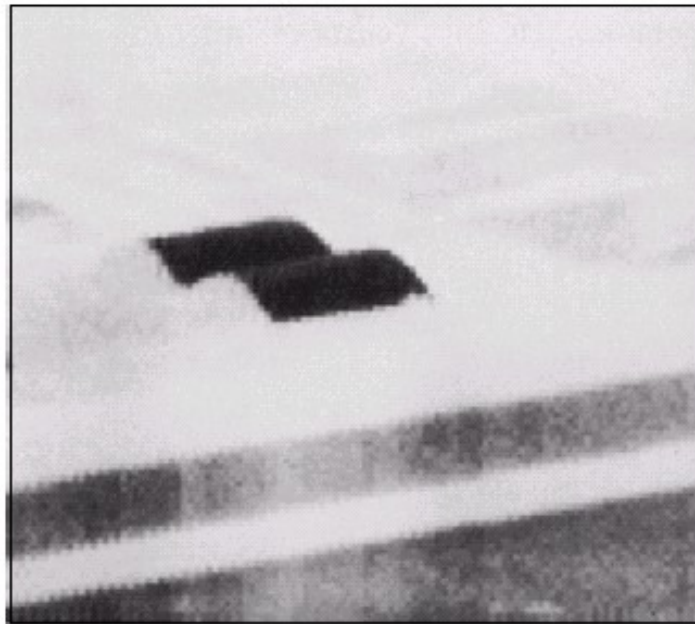
Transformada de Hough

Algoritmo → Transformada de Hough - (espaço θ - ρ)

- 1 - Quantize o espaço do parâmetros (θ - ρ) adequadamente;
- 2 - Assuma que cada célula no espaço dos parâmetros é um acumulador; inicialize todas as células com zero;
- 3 - Para cada ponto (x,y) no espaço imagem, incremente em 1 cada um dos acumuladores(ρ,θ), que satisfazem a expressão $\rho = x \cos \theta + y \sin \theta$ (fazendo θ variar e calculando ρ);
- 4 - Os máximos nos acumuladores correspondem aos valores de ρ e θ procurados;

A necessidade de um acumulador de grandes dimensões é um problema a ser resolvido na implementação da Transformada de Hough.

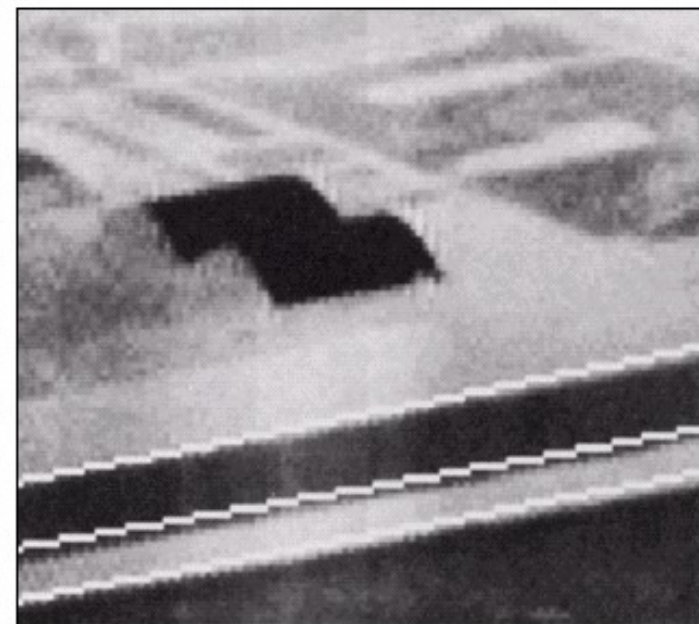
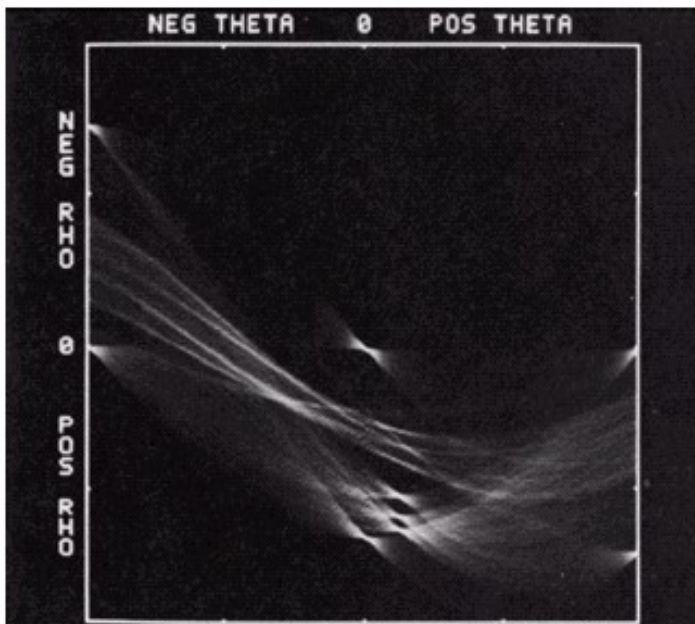
Transformada de Hough;



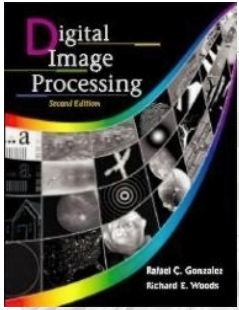
a) Imagem original

b) Bordas

c) Transf. de Hough

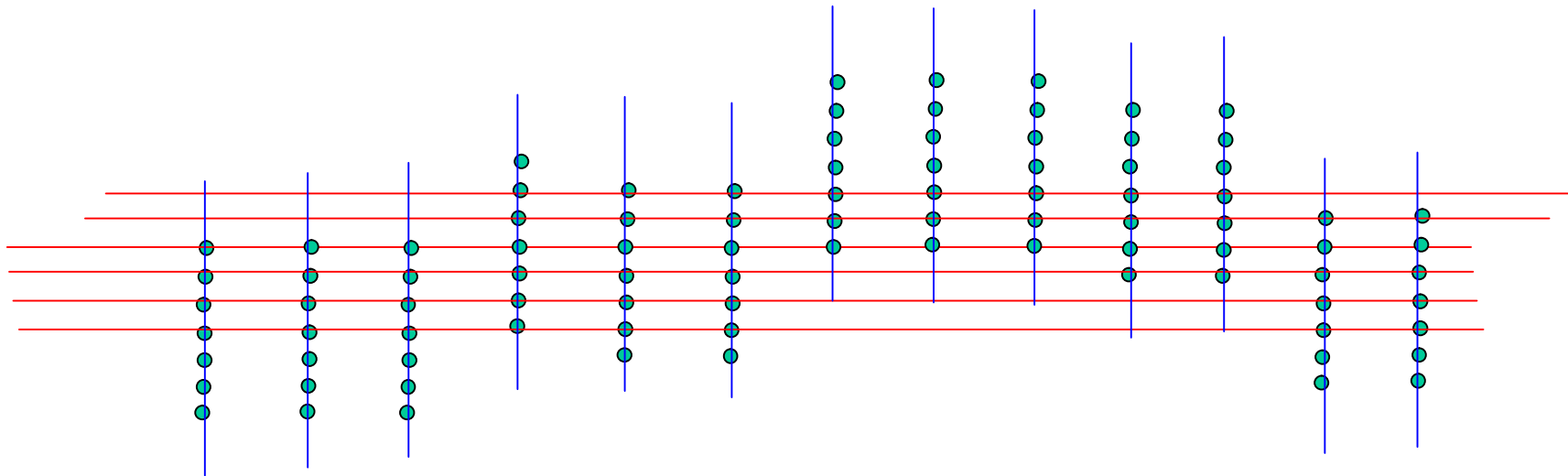


d) Pontos Detectados



Desvantagem da Transformada de Hough

Vários pontos alinhados sempre indicam uma linha reta, mesmo não pertencendo ao mesmo segmento de reta



Neste caso, Hough vai preferir detectar as linhas vermelhas do que as linhas azuis.

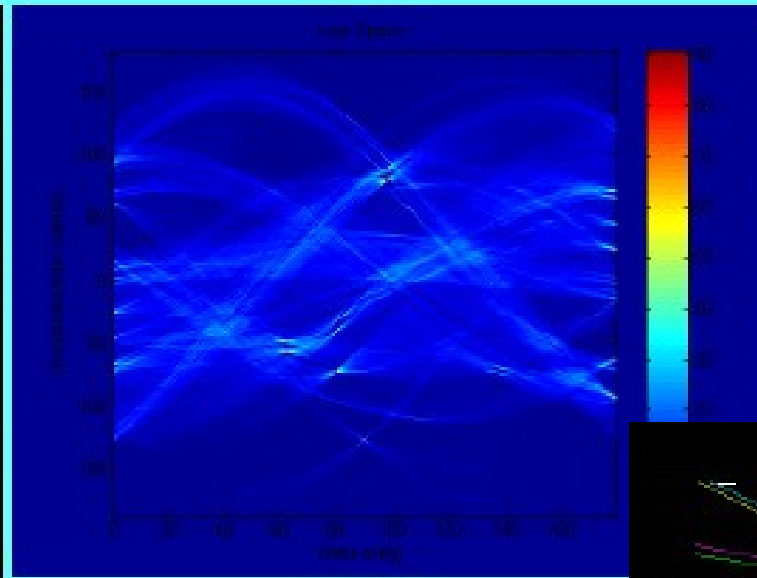
O método local, de varredura e rotulação não tem este problema



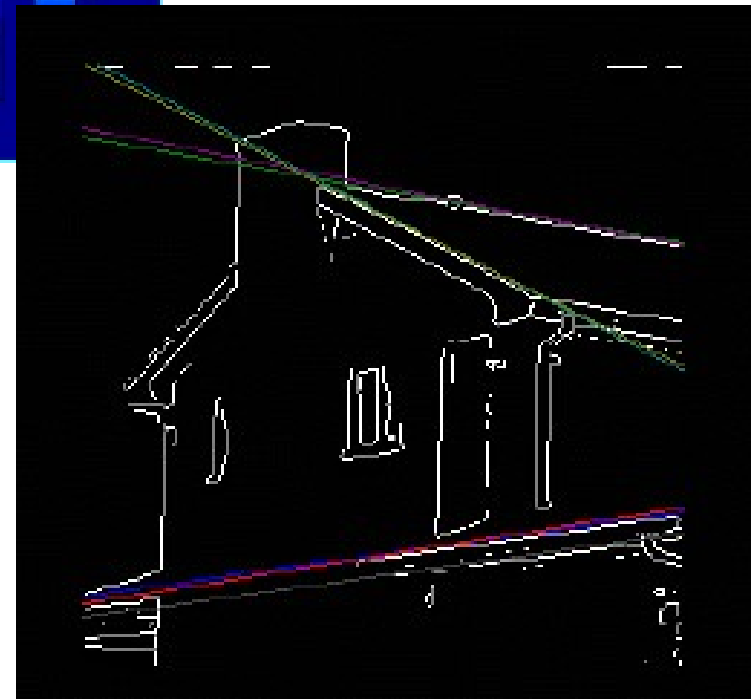
Transformada de Hough



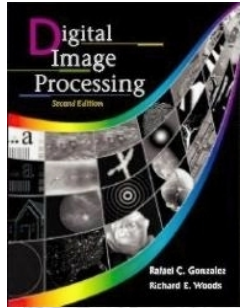
Imagem original



Linhas Detectadas



Aplicações

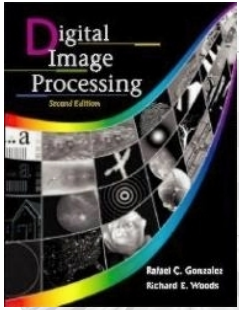


Segmentação de Imagens

Os algoritmos baseados em similaridade

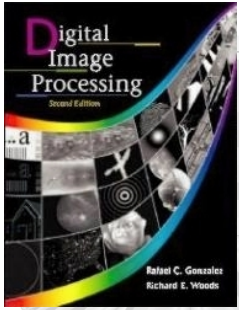
-baseiam-se em:

- limiarização
- crescimento de regiões
- divisão/fusão de regiões



Limiarização

A limiarização pode ser vista como um problema teórico de decisão estatística, cujo objetivo é minimizar o erro médio, incorrido na atribuição de pixels para dois ou mais grupos (também chamados de classes)



Limiarização

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < \text{limiar} \\ f(x, y) & \text{se } f(x, y) \geq \text{limiar} \end{cases}$$

Binarização

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < \text{limiar} \\ 255 & \text{se } f(x, y) \geq \text{limiar} \end{cases}$$



Limiarização

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < \text{limiar} \\ f(x, y) & \text{se } f(x, y) \geq \text{limiar} \end{cases}$$





Segmentação de Imagens

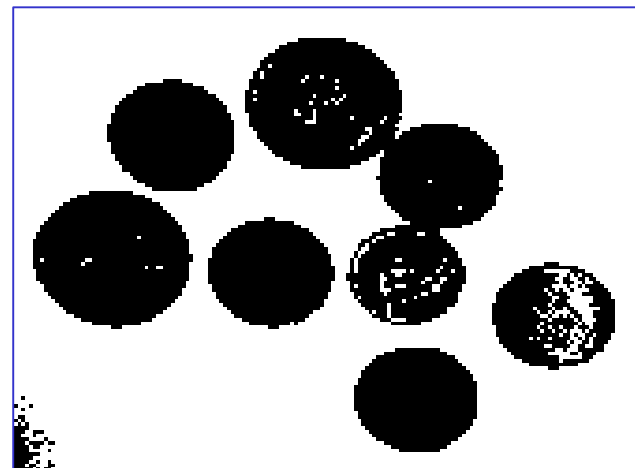
Binarização

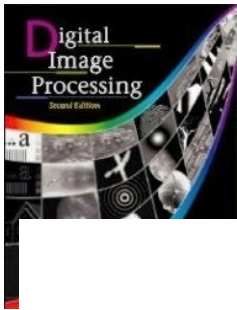
É uma das abordagens mais simples para a segmentação

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < \text{limiar} \\ 1 & \text{se } f(x, y) \geq \text{limiar} \end{cases}$$

Nesta imagem:

pixels 1 são o fundo da imagem (claros)
enquanto que pixels com valor 0 (escuros)
constituem os objetos





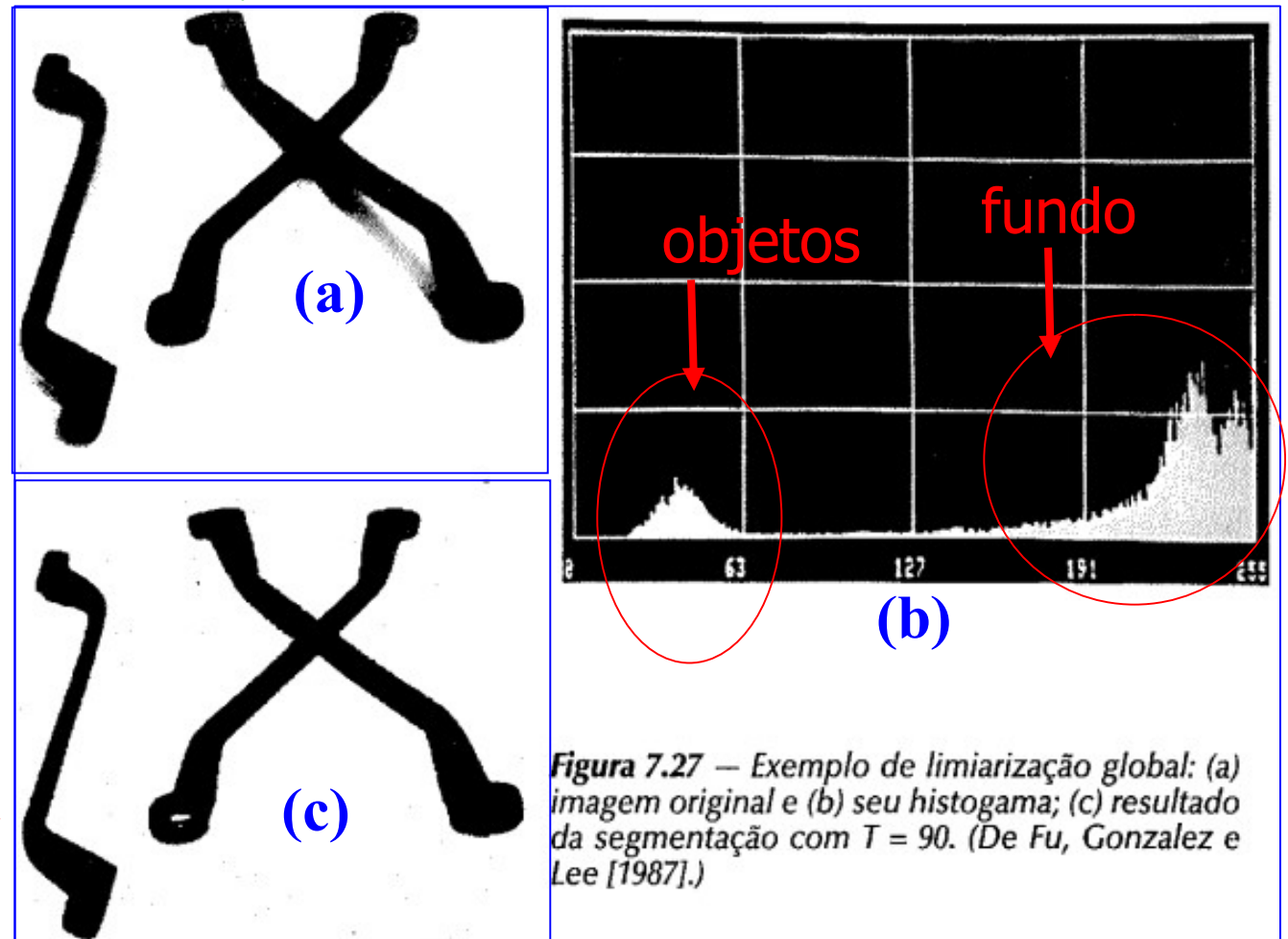
Segmentação de Imagens

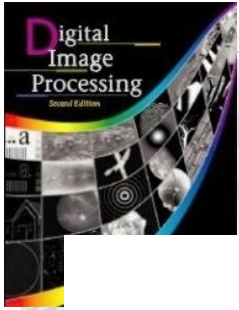
Limiarização Binarização - Como achar o limiar?

As técnicas mais comuns utilizam a informação contida em um histograma para orientar o processamento a ser realizado

Os vales dos histogramas indicam os valores que devem ser usados como limiares T_i

binária →





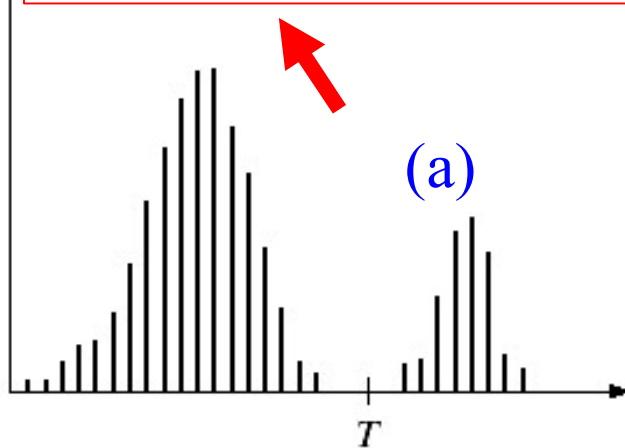
Segmentação de Imagens

Limiarização - Como encontrar o limiar??

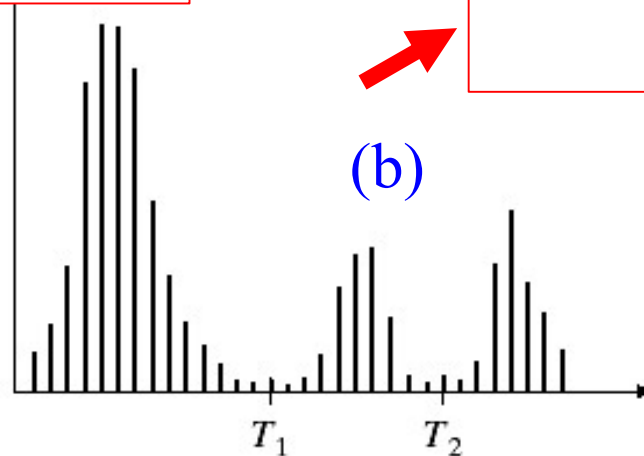
Existem casos em que um único limiar divide (segmenta) as partes da imagem

Em outros casos, pode ser necessário mais de um limiar T_i

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$



$$g(x, y) = \begin{cases} a & \text{se } f(x, y) > T_2 \\ b & \text{se } T_1 < f(x, y) \leq T_2 \\ c & \text{se } f(x, y) \leq T_1 \end{cases}$$

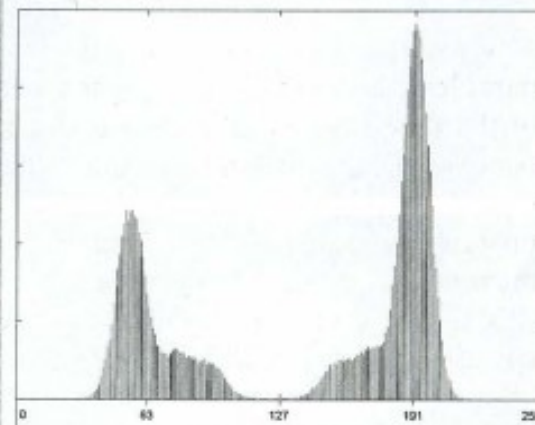


Histogramas que podem ser particionados usando
a) um limiar T e b) usando dois limiares



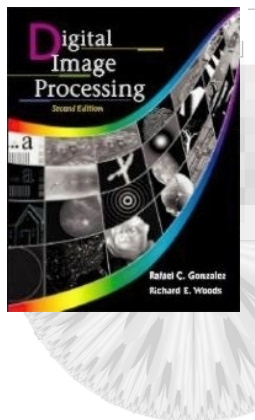
Limiarização

Algumas imagens tem a sua limiarização óbvia →



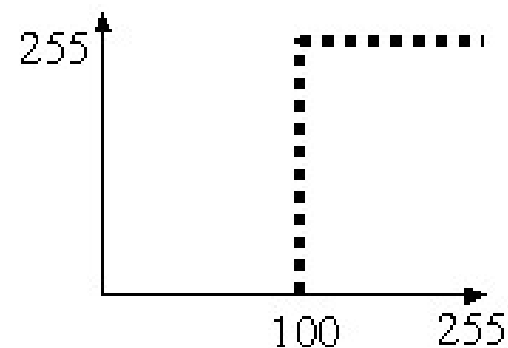
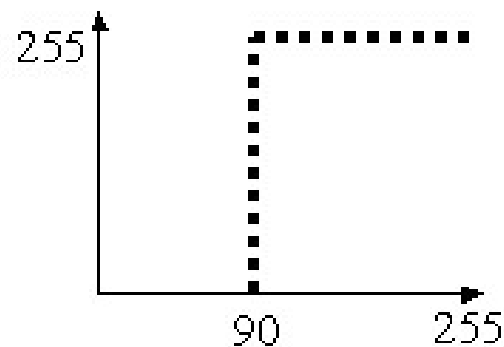
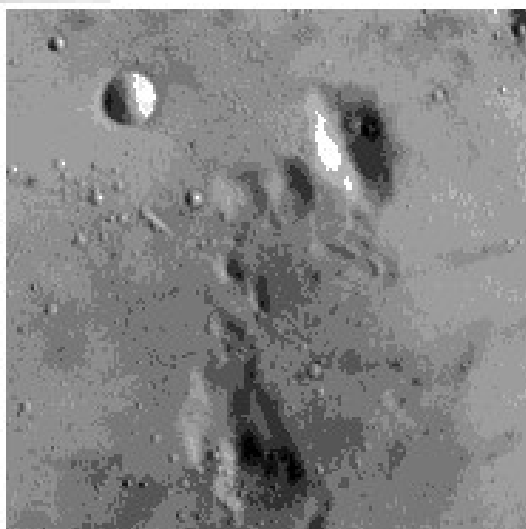
Para outras imagens, isto não ocorre →



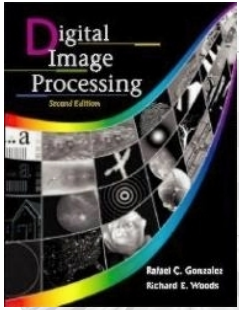


Limiarização - histograma

Segmentação de Imagens



Exemplos de imagens sem um limiar óbvio



Limiarização - Algoritmo Iterativo

1. Selecionar uma estimativa inicial para o limiar global, T

2. Segmentar a imagem usando T

Isso dará origem a dois grupos de pixels:

G_1 , composto por todos os pixels com valores de intensidade $> T$

G_2 , composto de pixels com valores $\leq T$

3. Calcular os valores de intensidade média de m_1 e m_2 para os pixels em G_1 e G_2 , respectivamente

4. Calcular um novo valor de limiar: $T = \frac{1}{2}(m_1 + m_2)$

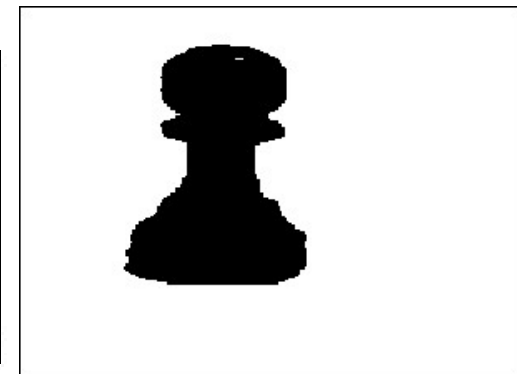
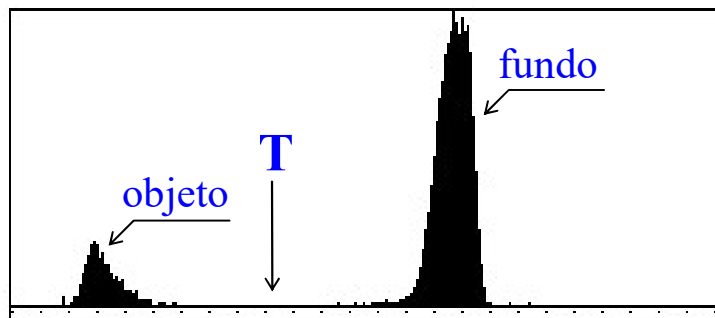
5. Repita as etapas 2 a 4 até que a diferença entre os valores de T em iterações sucessivas seja menor que o parâmetro predefinido ΔT



Limiarização

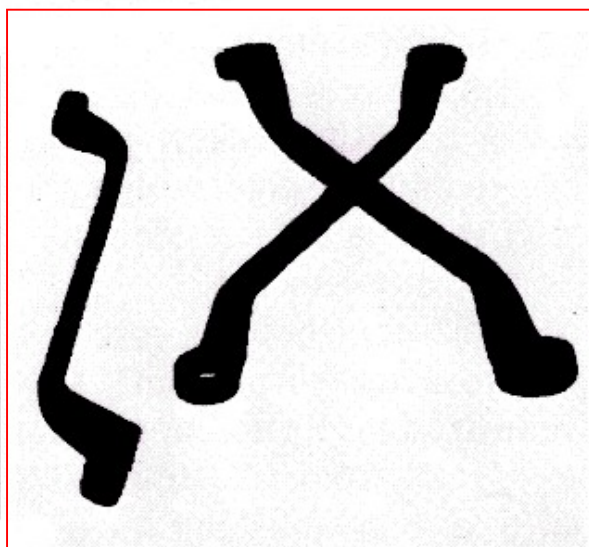
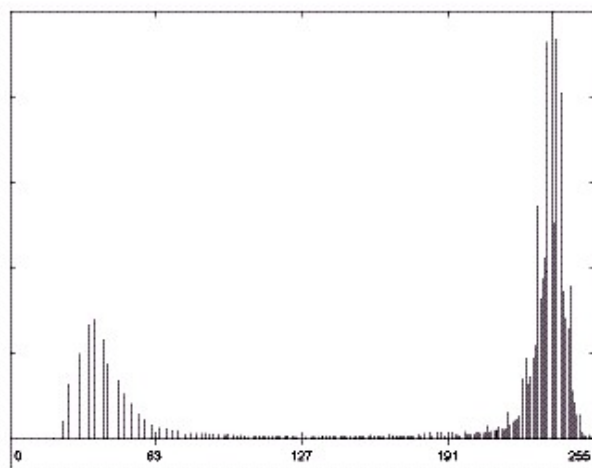
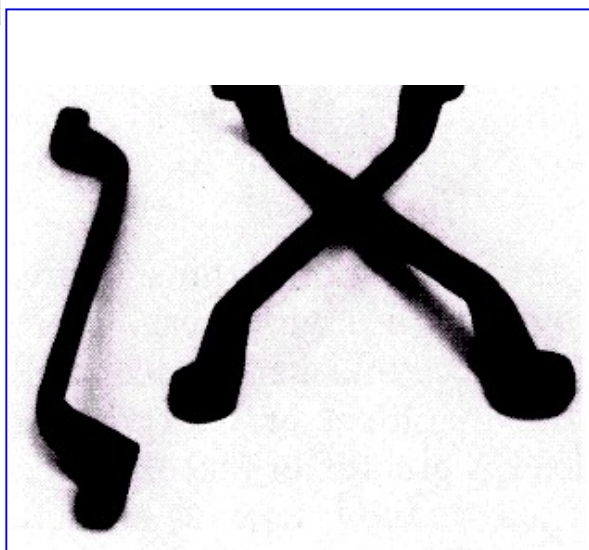
P-TILE

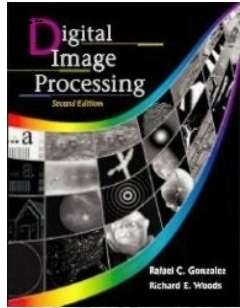
- A técnica mais simples obriga conhecer a área de ocupação do objeto na imagem.
- Sabendo-se que o objeto ocupa $p\%$ da imagem, então o limiar T é definido pelo valor que resulta em $p\%$ da imagem acima deste valor (supondo o objeto com altos e o fundo com valores baixos).
- Esta técnica é adequada para situações restritas





Exemplos





Método de OTSU

Este método é baseado na análise discriminante, e o valor do limiar é obtido supondo que os pixels da imagem podem ser classificados em duas classes (C_0 e C_1) que são o objeto e o fundo

Tomando σ_B^2 e σ_T^2 as variâncias entre as classes e total respectivamente.



Método de OTSU

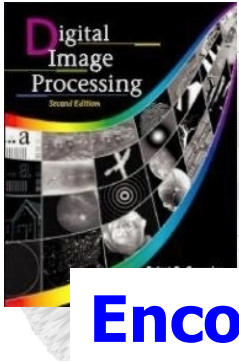
Supondo n_i o número de pixels com a tonalidade i
O total de pixels na imagem é

$$n = \sum_{i=0}^{l-1} n_i$$

a probabilidade do pixel com intensidade i ocorrer é

$$p_i = \frac{n_i}{n}$$

considera-se que o nível de cinza mais escuro é 0
e o nível mais claro é $l-1$ 0 → 255



Encontrar t , tal que: $t^* = \underset{t \in G}{\text{Arg Min}} \eta,$

sendo $\eta = \frac{\sigma_B^2}{\sigma_T^2}$

t é o candidato a limiar
 l é 256 níveis de cinza

$$\sigma_T^2 = \sum_{i=0}^{l-1} (i - \mu_T)^2 p_i, \quad \mu_T = \sum_{i=0}^{l-1} i p_i,$$

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2, \quad \omega_0 = \sum_{i=0}^t p_i, \quad \omega_1 = 1 - \omega_0,$$

$$\mu_1 = \frac{\mu_T - \mu_t}{1 - \omega_0}, \quad \mu_0 = \frac{\mu_t}{\omega_0}, \quad \mu_t = \sum_{i=0}^t i p_i.$$



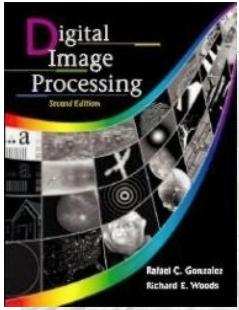
OTSU

Limitações

Exibe um desempenho bom se o histograma possui distribuição bimodal e possui um vale profundo entre dois picos

Se a área do objeto é pequena em comparação com a área de fundo, o histograma não exibirá mais bimodalidade

Se a imagem é muito corrompida por ruído aditivo, o vale nítido do histograma de nível de cinza é degradado (resulta no erro de segmentação)

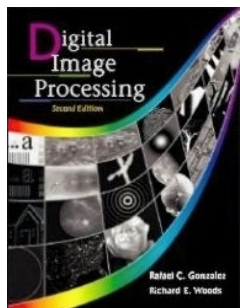


Método de Pun

Este método é baseado na teoria da informação, que se baseia na premissa de que a geração de informação pode ser modelada como um processo probabilístico

Por esta teoria define-se a entropia (quantidade de código necessária para representar um símbolo) de cada pixel da imagem por:

$$\text{Entropia}(x) = x \cdot \log(x)$$



Método de Pun

Pelo método, para cada possível valor de limiar são definidas duas entropias à posteriori (do objeto e do fundo da imagem), dadas por:

$$H_b = - \sum_{i=0}^t p_i \log_e p_i$$

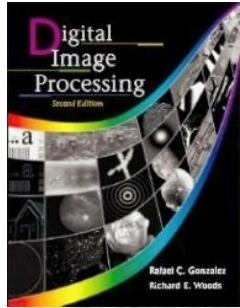
$$H_w = - \sum_{i=t+1}^{l-1} p_i \log_e p_i$$

e o valor do limiar ótimo é dado por

$$T = \text{Arg máximo } \{ H_b(t) + H_w(t) \}$$

Na implementação é preciso observar as situações em que $p_i = 0$, sendo que geralmente observa-se nos algoritmos o uso da seguinte função

$$f \log(x) = \begin{cases} \log(x) & \text{se } x > 0 \\ 0 & \text{caso contrario} \end{cases}$$



Método de Kapur, Sahoo e Wong

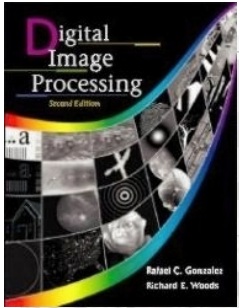
Como se trata de um outro método baseado na teoria da informação, este método também faz uso da entropia, porém, definindo H_b e H_w

$$H_b = - \sum_{i=0}^t \frac{p_i}{p_t} \log_e \left(\frac{p_i}{p_t} \right)$$

$$H_w = - \sum_{i=t+1}^{l-1} \frac{p_i}{1 - p_t} \log_e \left(\frac{p_i}{1 - p_t} \right)$$

e o valor do limiar ótimo é dado por

$$T = \text{Arg máximo } \{ H_b(t) + H_w(t) \}$$



Método de Johannsen e Bille

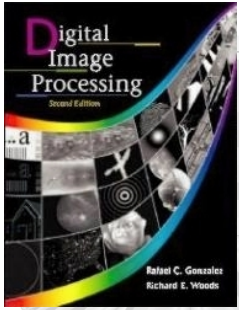
Um outro método baseado na teoria da informação, neste caso, o valor do limiar é dado por :

$$T = \text{Arg máximo } \{ S1(t) + S2(t) \}$$

com:

$$S_1 = \log_e \left(\sum_{i=0}^t p_i \right) - \frac{1}{\sum_{i=0}^t p_i} \left[p_t \log_e p_t + \left(\sum_{i=0}^{t-1} p_i \right) \log_e \left(\sum_{i=0}^{t-1} p_i \right) \right]$$

$$S_2 = \log_e \left(\sum_{i=t}^{l-1} p_i \right) - \frac{1}{\sum_{i=t}^{l-1} p_i} \left[p_t \log_e p_t + \left(\sum_{i=t+1}^{l-1} p_i \right) \log_e \left(\sum_{i=t+1}^{l-1} p_i \right) \right]$$



O impacto do ruído na limiarização

O ruído afeta muito o histograma das imagens e, conseqüentemente, as técnicas de limiarização baseadas no histograma

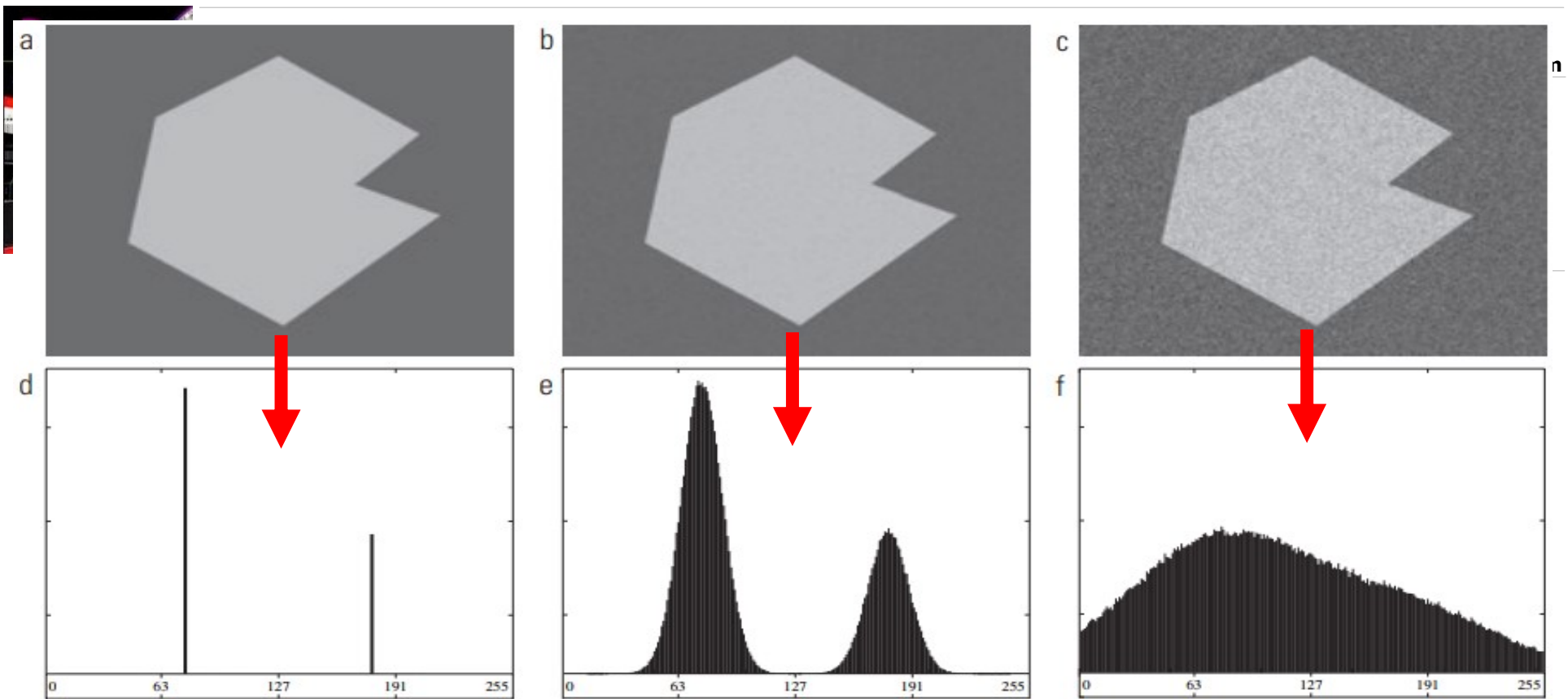


Imagem sintética
sem ruído

O histograma
consiste em dois
picos

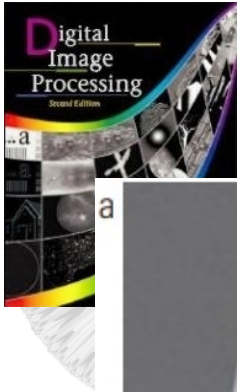
Fácil de limiarizar

Imagem original com
ruído gaussiano de
média zero e desvio
padrão de 10 níveis de
intensidade

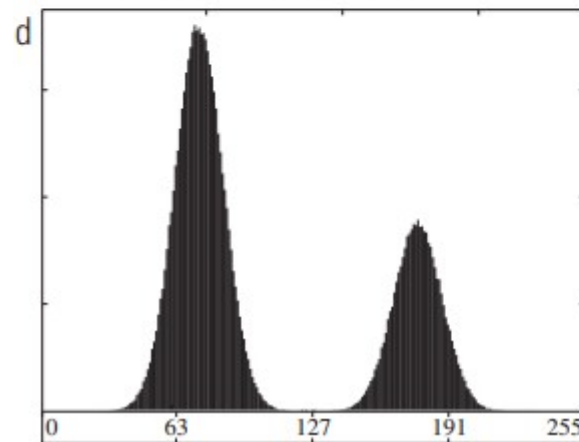
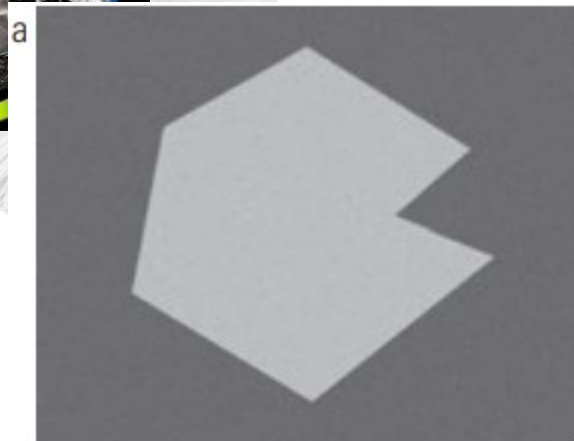
O histograma ainda
mostra uma boa
separação

Imagem original com
um ruído gaussiano de
média zero e desvio
padrão de 50 níveis de
intensidade

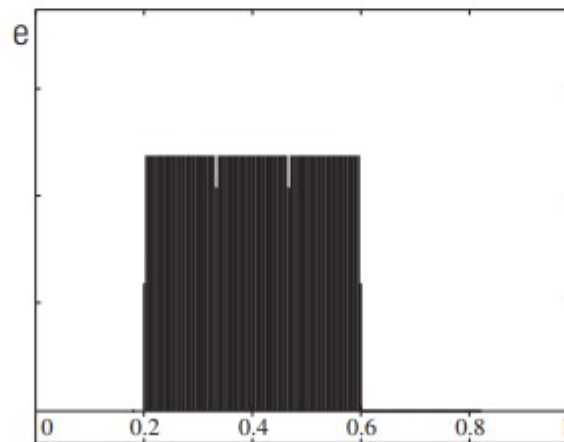
Como mostra o
histograma não há como
achar um limiar



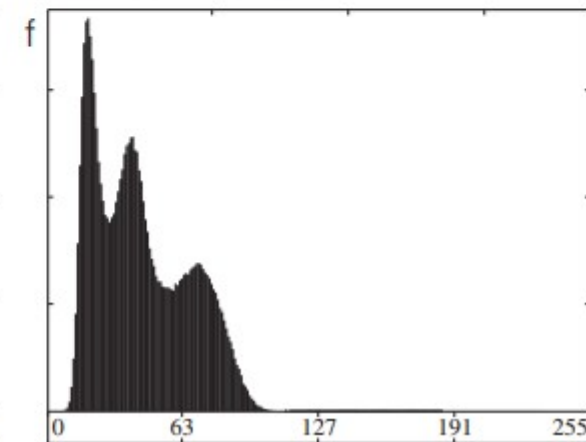
O impacto da iluminação e a refletância



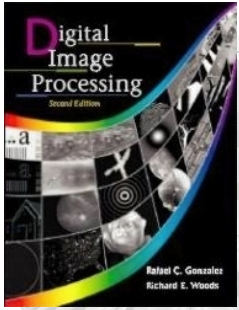
(a) Imagem ruidosa e seu Histograma



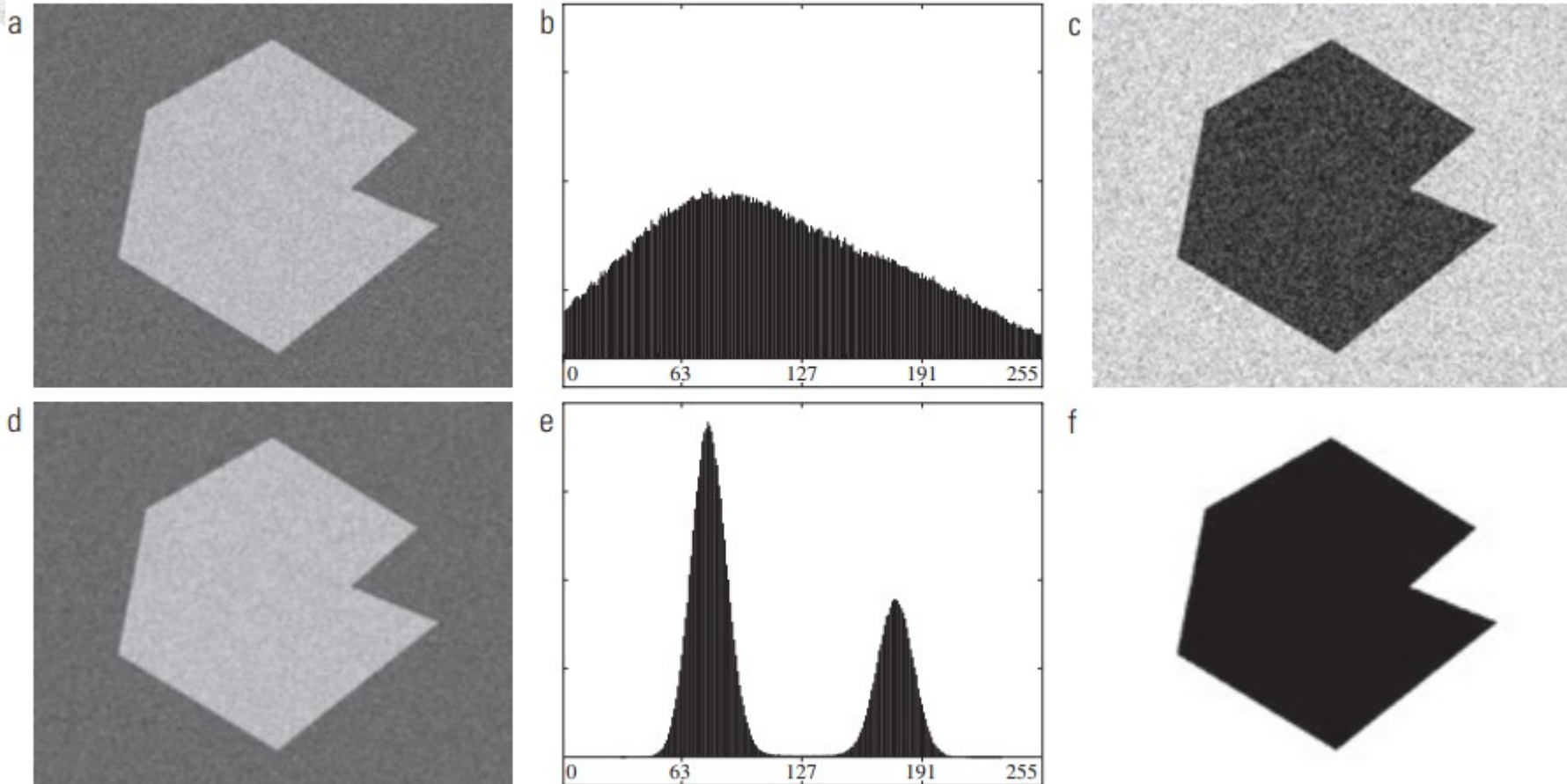
(b) Rampa de intensidade no intervalo $[0,2, 0,6]$ e seu Histograma



Produto de (a) e (b) e seu Histograma

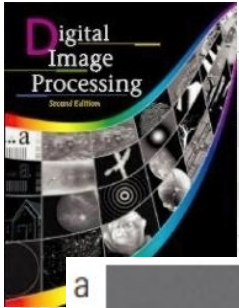


Usando a suavização da imagem para melhorar a limiarização

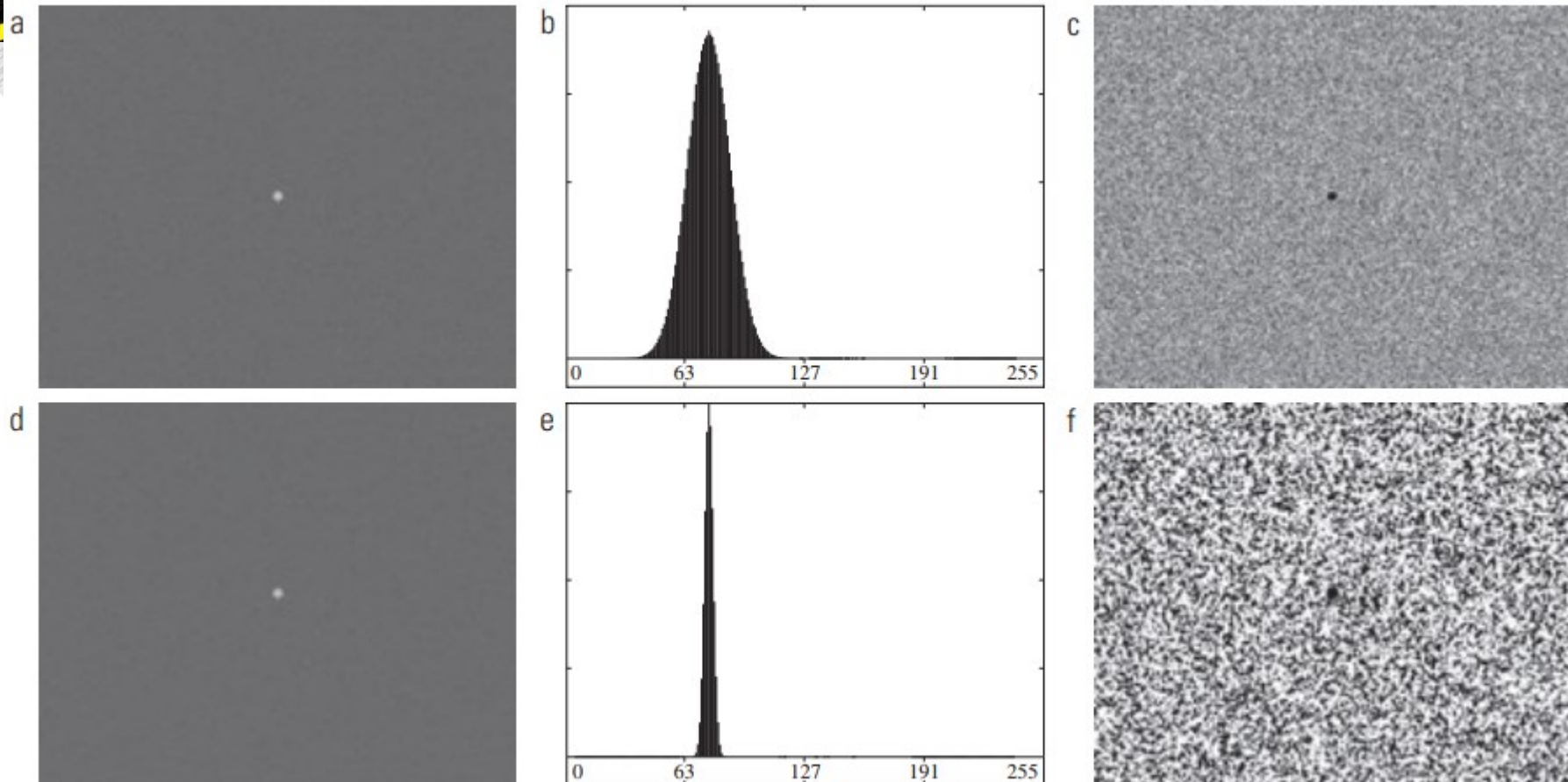


(a) Imagem ruidosa (b) seu histograma (c) Resultado com Otsu

(d) (a) suavizada com filtro da média 5×5 (e) seu histograma (f) Otsu

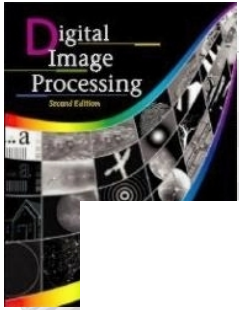


Impacto do tamanho do objeto em relação ao fundo



(a) Imagem ruidosa (b) Seu histograma (c) Resultado obtido com Otsu
(d) (a) suavizada com filtro da média 5×5 (e) seu histograma (f) Otsu

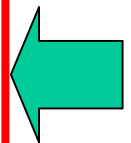
A limiarização falhou em ambos os casos



Limiarização Local

Os resultados empíricos mostram que o desempenho das técnicas de **limiares globais** (incluindo o método de Otsu) são limitados pelo tamanho pequeno do objeto, a pequena diferença média entre pixels de primeiro e segundo plano, grandes variações dos pixels que pertencem ao objeto e aqueles que pertencem ao fundo, a grande quantidade de ruído, etc.

A limiarização local aplica os mesmos conceitos, porém, sempre opera em uma vizinhança pequena





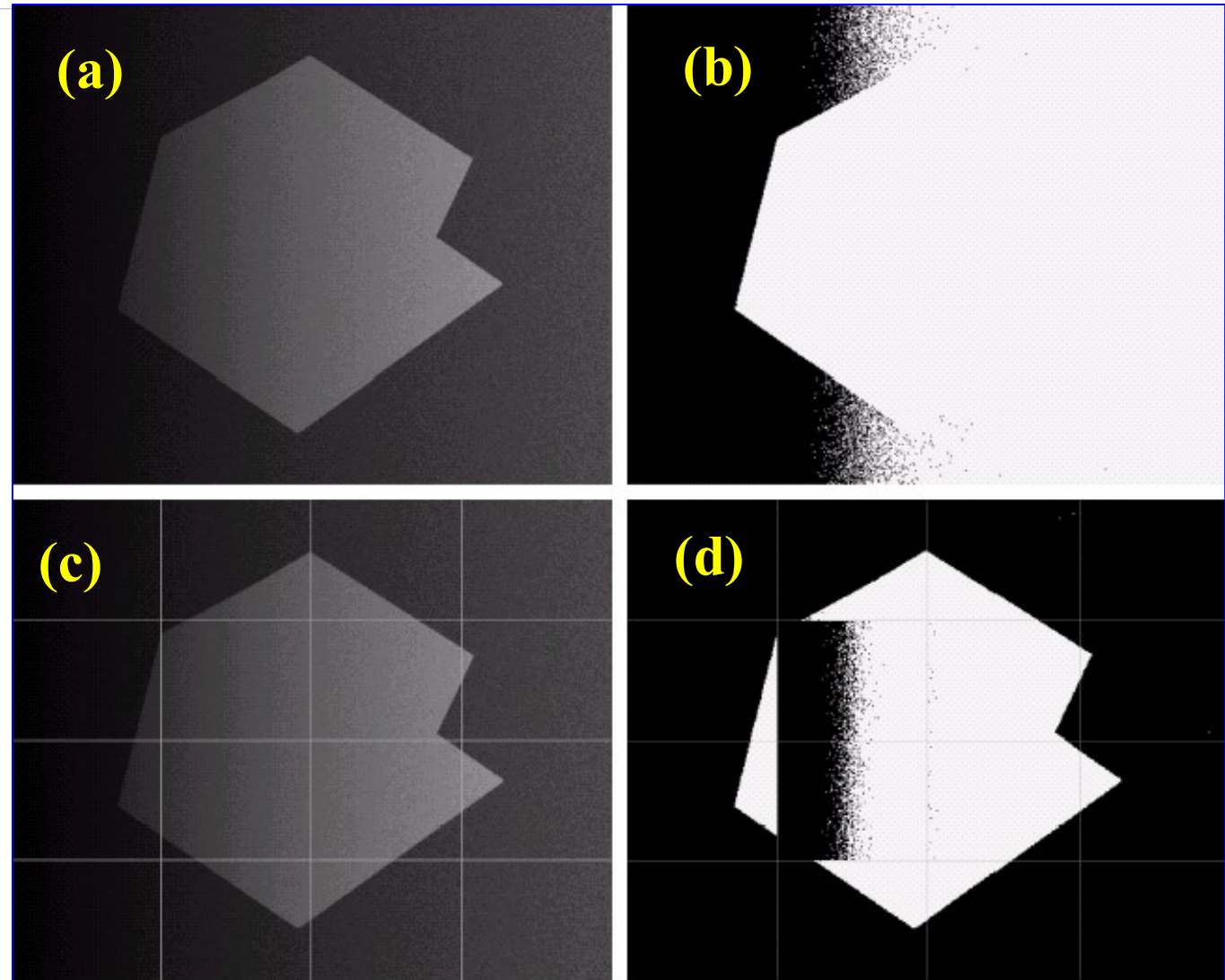
Limiarização variável por meio de particionamento da imagem

a) Imagem
Original

b) Usando Limiar
Global

c) Imagem dividida
em regiões

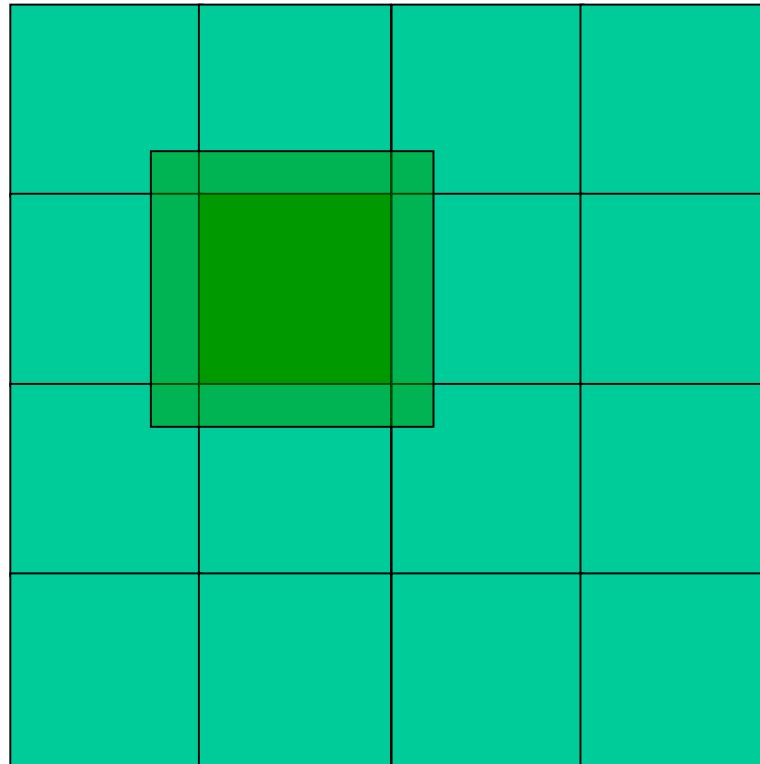
d) Usando Limiares
Locais





Limiarização Local

Uma boa opção é usar áreas
sobrepostas





Limiarização variável baseada nas propriedades locais da imagem

Uma abordagem mais geral do que o método de subdivisão da imagem é calcular um limiar para cada ponto, (x, y) , com base em uma ou mais propriedades calculadas em sua vizinhança

O hardware atual permite o processamento rápido da vizinhança, especialmente para as funções comuns, como as operações lógicas e aritméticas



Limiarização Local

Usando o **desvio padrão** e a **média dos pixels na vizinhança** de cada ponto de uma imagem

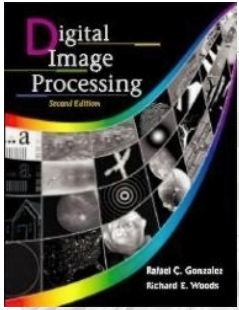
Estes dois parâmetros são bastante úteis para a determinação de limiares locais, pois são descritores de contraste local e intensidade média

Supondo

σ_{xy} – o desvio padrão e

m_{xy} – o valor médio do conjunto de pixels contidos em S_{xy}

com S_{xy} uma vizinhança centrada em (x, y)



Limiarização Local

As seguintes são formas comuns de limiares variáveis locais:

1)

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

em que a e b são constantes não negativas

ou

2)

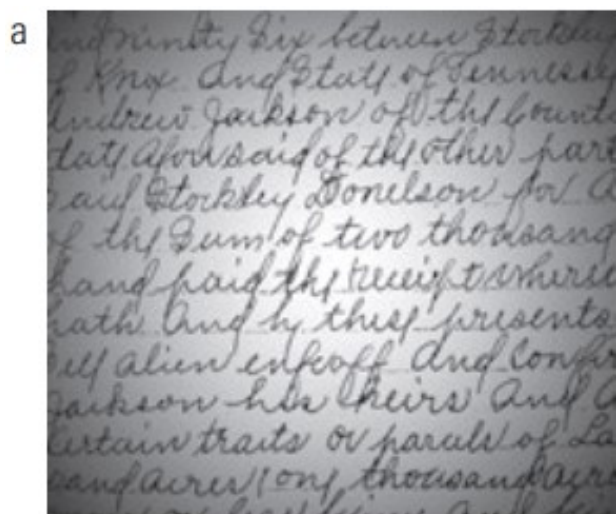
$$T_{xy} = a\sigma_{xy} + bm_G$$

em que m_G é a média global da imagem

$$\rightarrow g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T_{xy} \\ 0 & \text{se } f(x, y) \leq T_{xy} \end{cases}$$



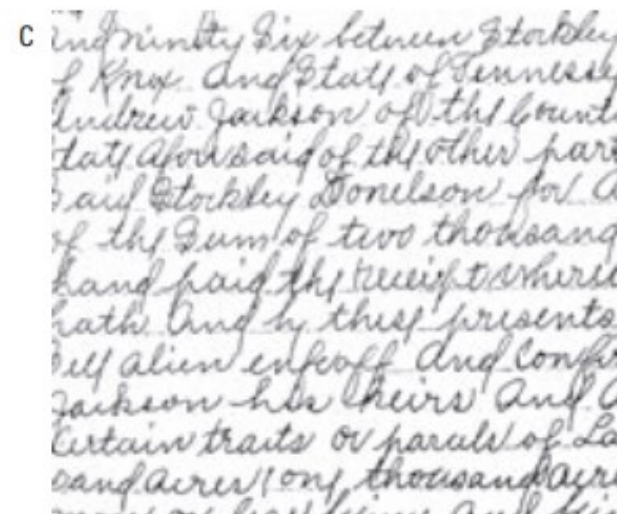
Limiarização Local



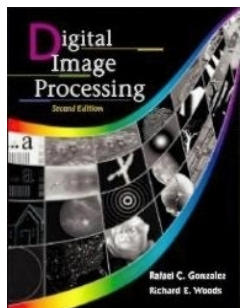
a) Imagem original



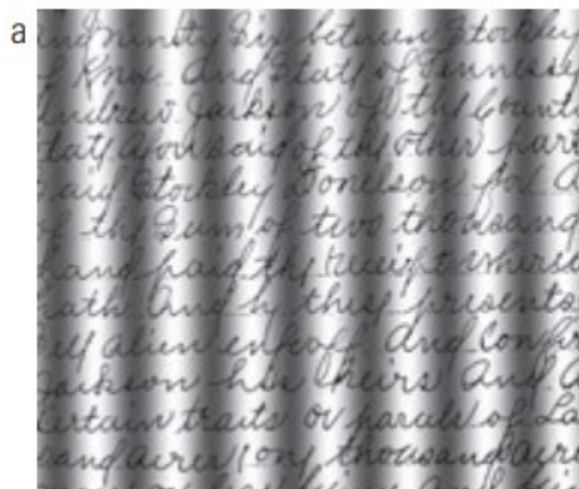
b) Resultado
com Otsu Global



c) Baseado na
Vizinhança



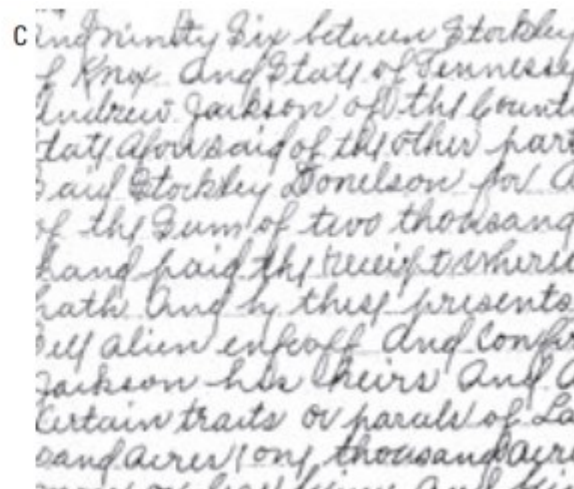
Limiarização Local



a) Imagem original



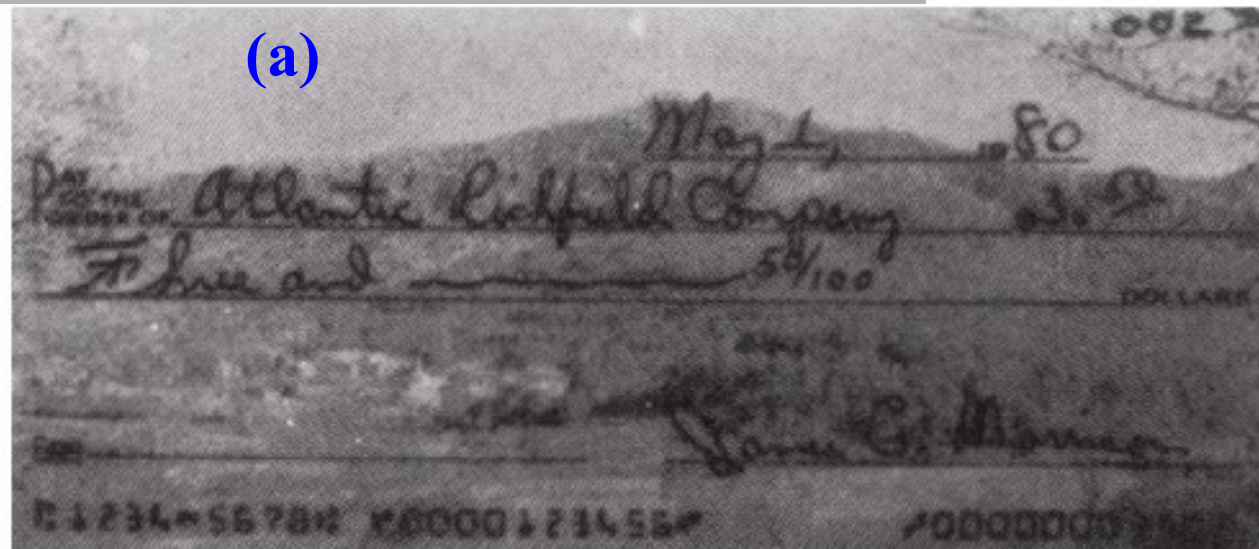
b) Resultado
com Otsu Global



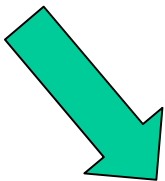
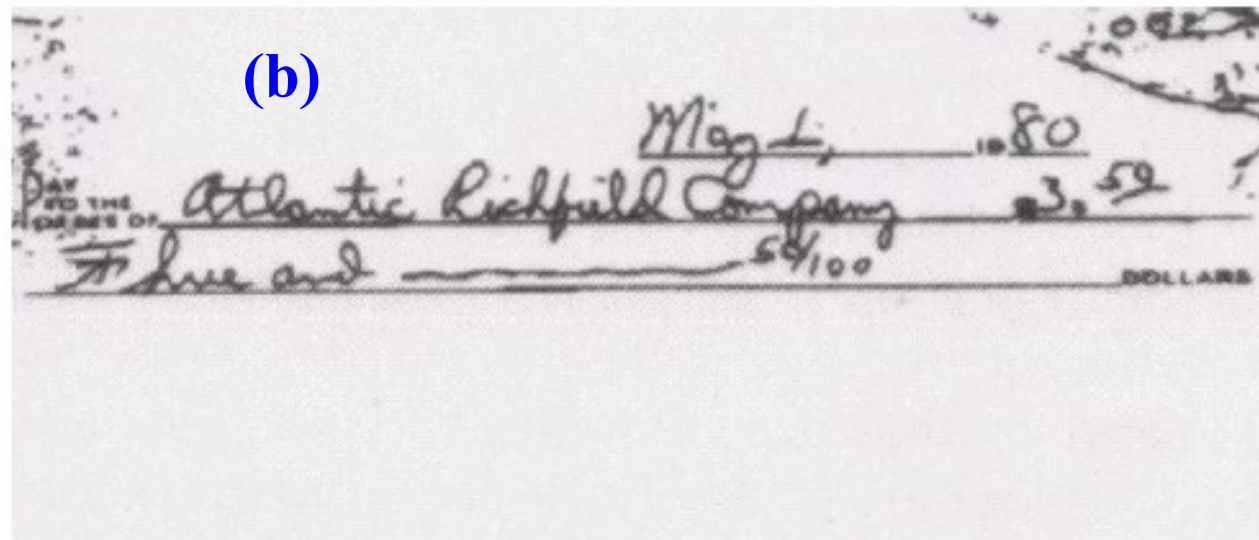
c) Baseado na
Vizinhança



a) Imagem Original



b) Limiarização Local

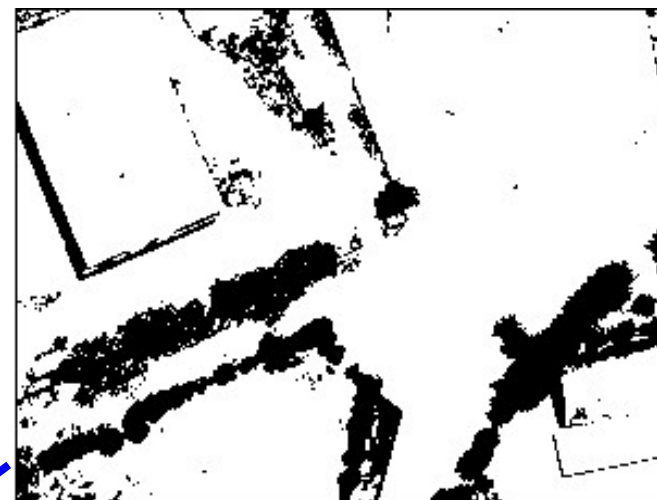


Se usasse uma limiarização global, não detectaria a escrita, por causa das áreas escuras e claras da imagem de fundo

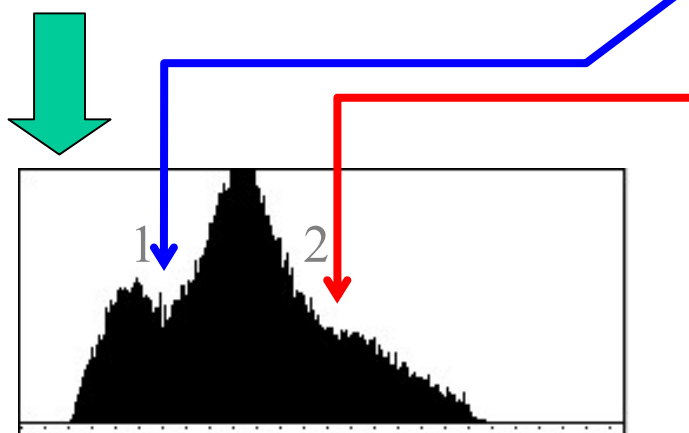


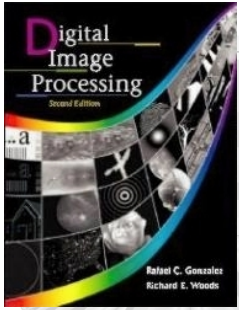
Segmentação de Imagens

Limiarização



Dificuldades
Qual o limiar
adequado ?





Divisão e fusão de regiões (split and merge)

Nesta técnica:

1) Divide-se **(split)** sucessivamente a imagem em regiões menores, até que cada região obtida tenha apenas uma tonalidade, cor ou textura
(pode-se admitir pequenas diferenças)

2) Regiões similares e adjacentes são unidas
(merge)

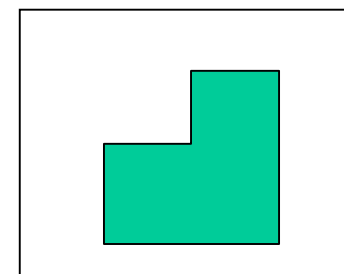
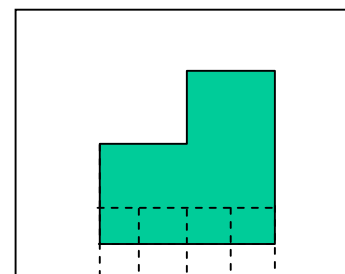
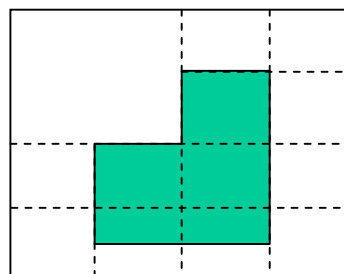
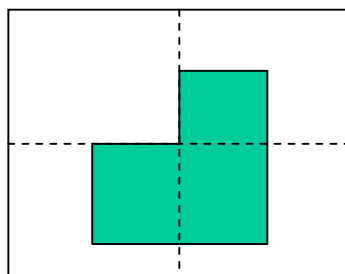
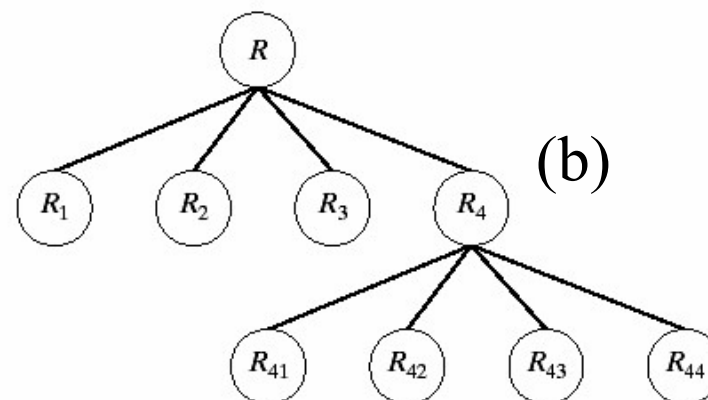
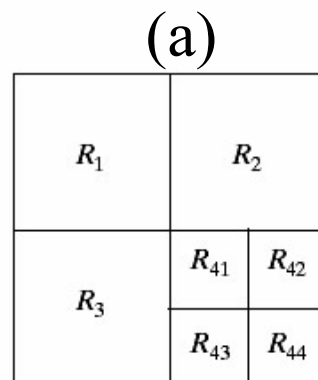


Segmentação de Imagens

Divisão e fusão de regiões (split and merge)

a) Imagem
Particionada

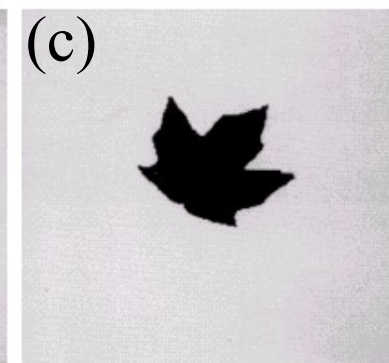
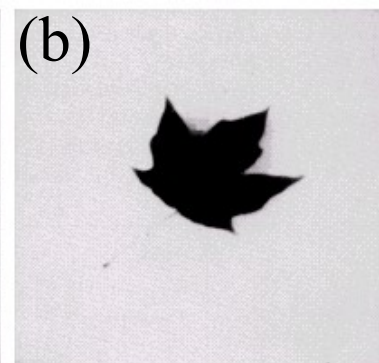
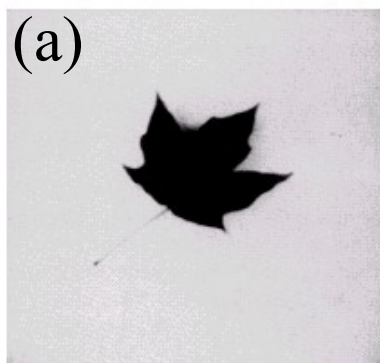
b) Quadtree
correspondente



a) Imagem Original

b) Resultado do split-merge

c) Limiarizada (limiar
obtido com o histograma)





Segmentação de Imagens

Crescimento de regiões

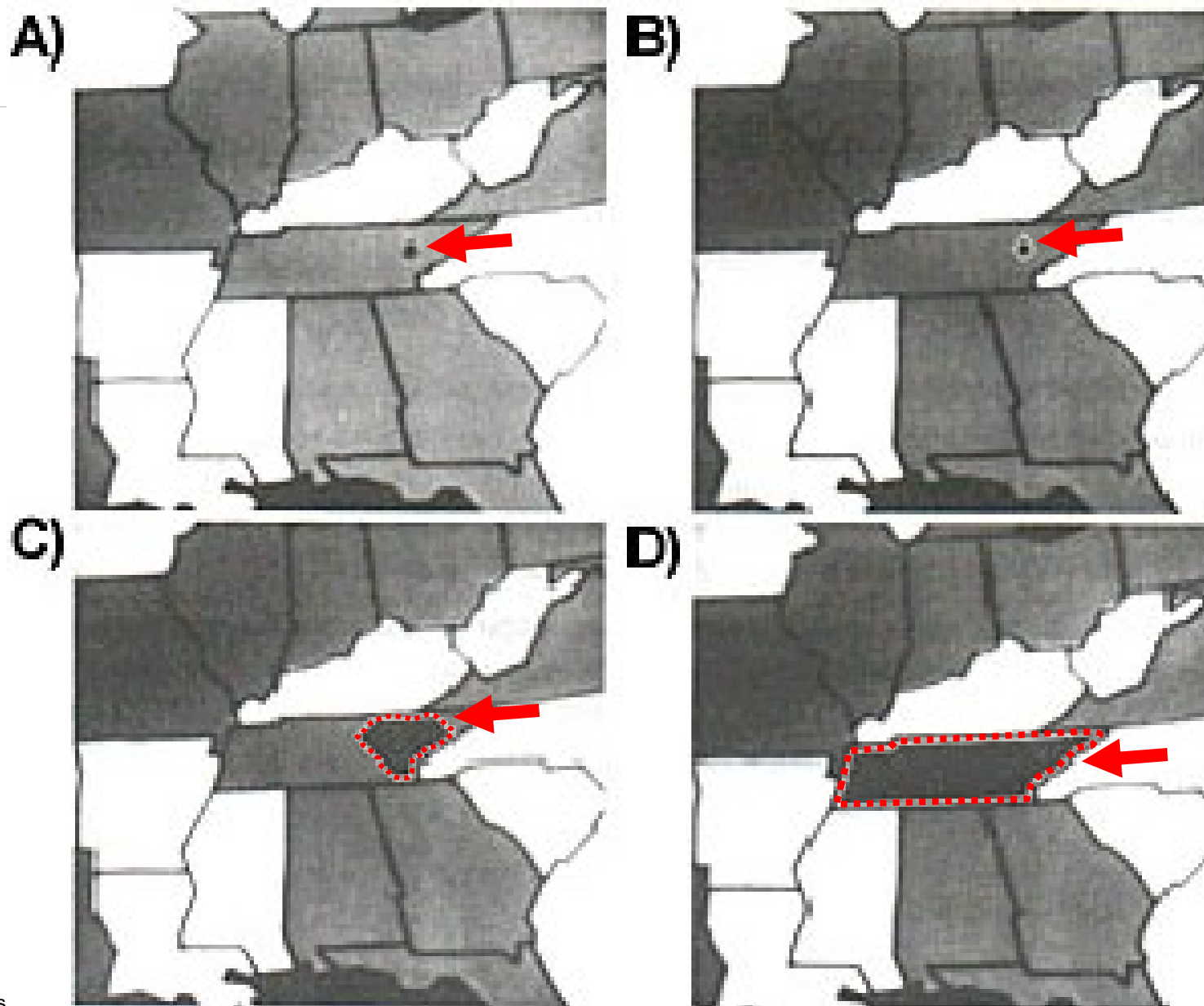
Nesta técnica, parte-se de um ponto do objeto e, pontos adjacentes similares a ele são unidos a ele

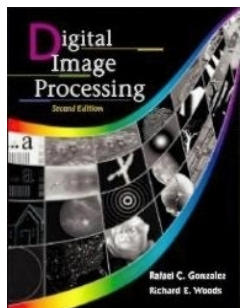
Este processo se repete, até não ser possível unir mais pontos



Segmentação de Imagens

Crescimento
de regiões





Segmentação de Imagens

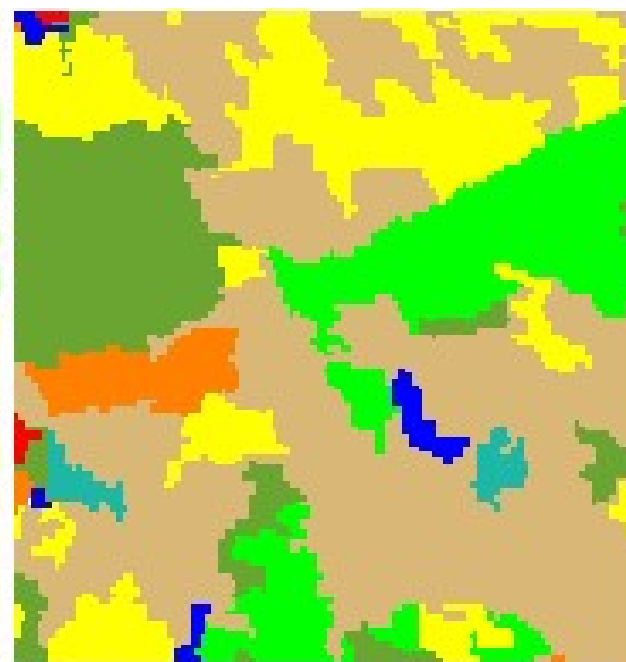
Crescimento de regiões



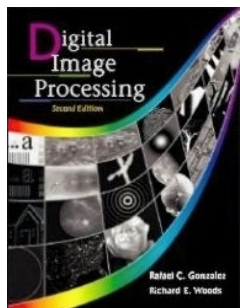
Imagem do Satélite



Imagem Segmentada



Mapa final



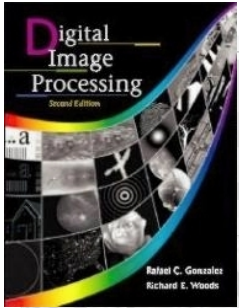
Códigos para o Crescimento de regiões

**Menos
chamadas
recursivas**

```
void floodFill (x, y, cor, novaCor)
if (pixel (x, y) == cor)
{
    pixel (x, y) ← novaCor
    floodFill (x + 1, y, cor , novaCor)
    floodFill (x, y + 1, cor , novaCor)
    floodFill (x - 1, y, cor , novaCor)
    floodFill (x, y - 1, cor , novaCor)
}
```

**Mais
simples**

```
void floodFill (x, y, cor, novaCor)
if (pixel (x, y) == cor)
{
    pixel (x, y) ← novaCor
    if (pixel (x+1, y) == cor) floodFill (x + 1, y, cor , novaCor)
    if (pixel (x, y+1) == cor) floodFill (x, y + 1, cor , novaCor)
    if (pixel (x -1, y) == cor) floodFill (x - 1, y, cor , novaCor)
    if (pixel (x, y-1) == cor) floodFill (x, y - 1, cor , novaCor)
}
```



Utilização do movimento na segmentação

Diferenças cumulativas

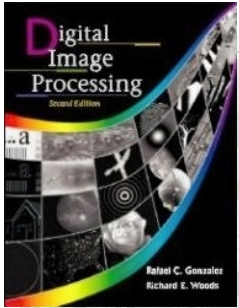
Consiste na comparação de uma sequência de quadros, com um referênciã, para então detectar um objeto em movimento

$$d_{ij}(x, y) = \begin{cases} 1 & \text{se } |f(x, y, t_i) - f(x, y, t_j)| > \theta \\ 0 & \text{caso contrário} \end{cases}$$

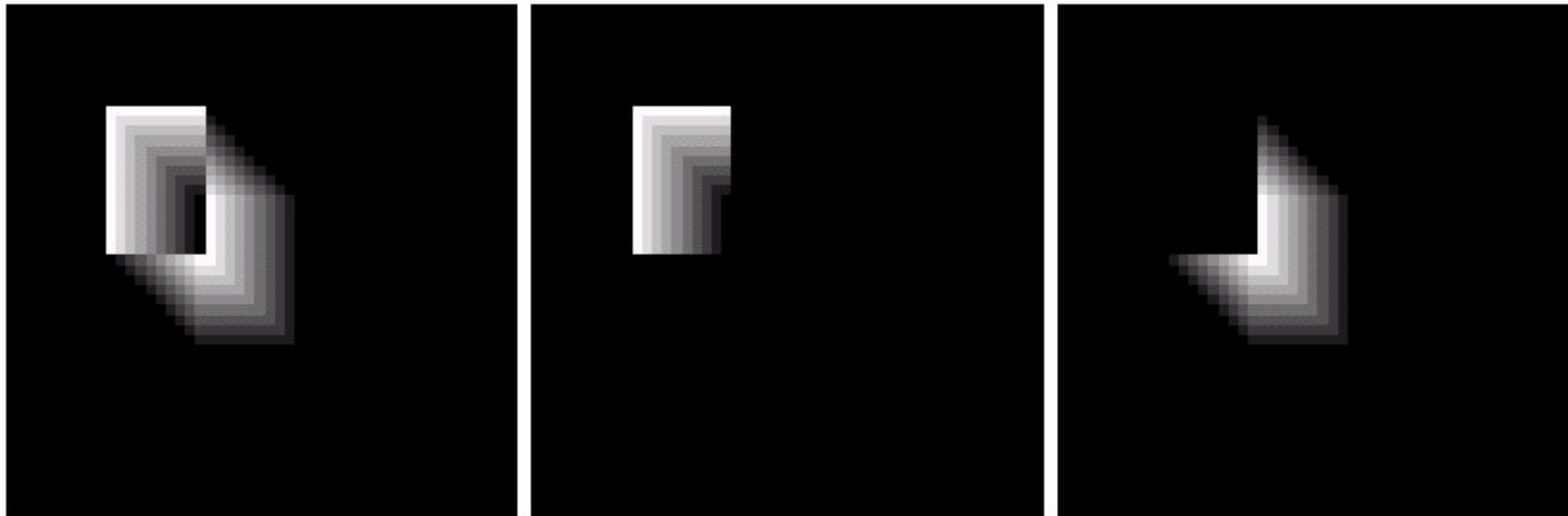
Imagem de Referência

A chave do sucesso desta abordagem é ter uma imagem de referência, com a qual se possam fazer as comparações

Quando não se tem uma imagem estática para referência, torna-se necessário construir uma, a partir de um conjunto de imagens com objetos em movimento



Utilização de movimento na segmentação

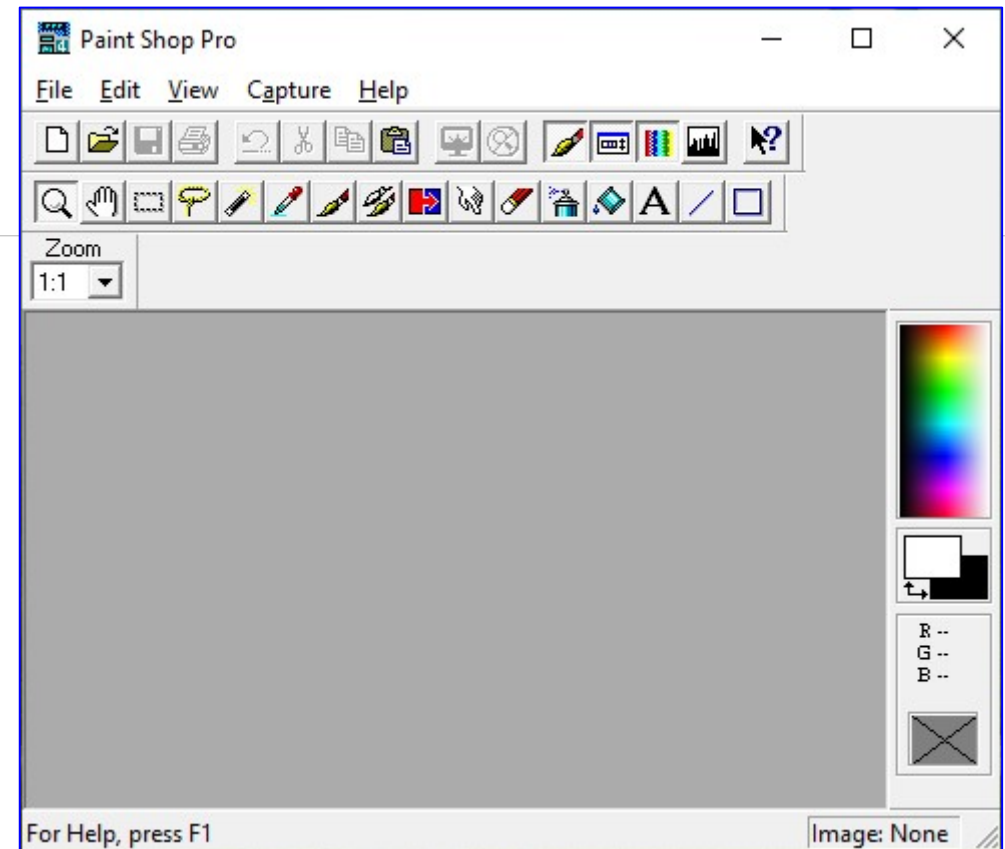


Objeto retangular detectado pelo seu movimentos em alguns frames

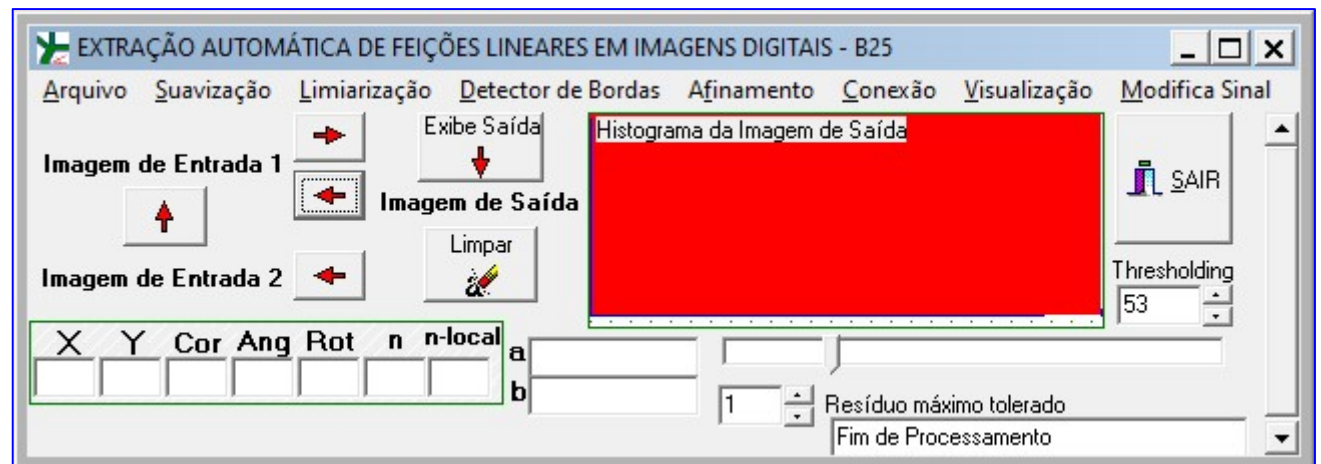


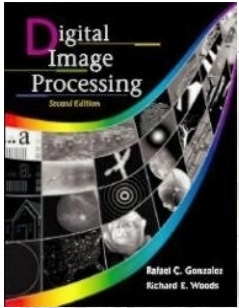
Ambientes de Processamento de Imagens

Comerciais



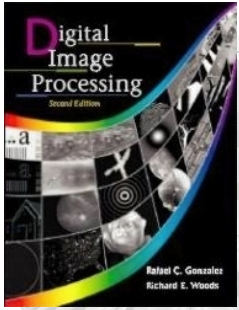
Acadêmicos
(TCCs, Mestrados
e Doutorados)





Ambientes de Processamento de Imagens

- Os comerciais tem muitas técnicas implementadas e facilitam o trabalho (basta instalar e usar)
- Os desenvolvidos pelos usuários podem contemplar técnicas que não existem nos sistemas comerciais
- O desenvolvimento tem sido facilitado com o uso de bibliotecas, como o OpenCV
- A implementação sem o uso das bibliotecas permite aos implementador ter um maior domínio das técnicas



Prática

Implementar o método de OTSU, para usá-lo na limiarização e binarização, já implementados