



# Aula 11.1

## Representação e Descrição de Imagens

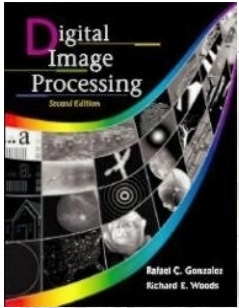


## Representation & Description

Depois que uma imagem é segmentada em regiões ou objetos, o resultado precisa ser representado e descrito em uma forma adequada para ser processada por um computador

Por exemplo, a etapa de classificação de objetos, tradicionalmente, é feita usando um classificador como o Classificador Bayesiano ou uma Rede Neural, que operam com valores numéricos

Daí a necessidade de extrair medidas numéricas dos objetos que precisam ser classificados, objetos estes contidos nas imagens

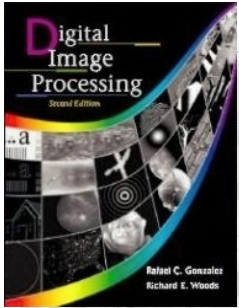


## Representation & Description

Como obter medidas dos objetos nas imagens,  
para que eles possam ser identificadas?



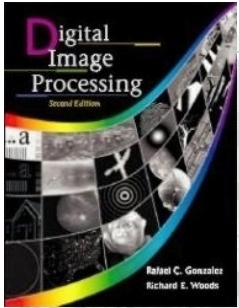
Altura do objeto  
Largura do objeto  
Comprimento do perímetro  
Área do objeto  
Textura  
Cor  
etc.



## **Seguidor de fronteira (contorno)**

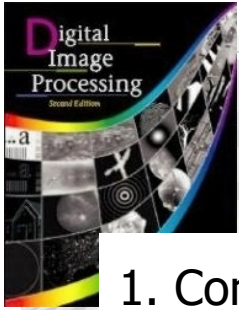
Vários dos algoritmos discutidos neste capítulo exigem que os pontos na fronteira de uma região estejam ordenados em sentido horário (ou anti-horário)

**CEGUINHO**



Assume-se que:

- 1) Trabalhar com imagens binárias em que os pontos do objeto e do fundo estão marcados com 1 e 0, respectivamente
- 2) As imagens foram preenchidas com uma fronteira de 0s para eliminar a possibilidade de um objeto se fundir com a borda da imagem



1. Considere que o ponto de partida,  $b_0$ , seja o ponto *mais alto e mais à esquerda* na imagem que esteja rotulada com valor 1

Denote por  $c_0$  o vizinho a oeste de  $b_0$  (Fig. b)

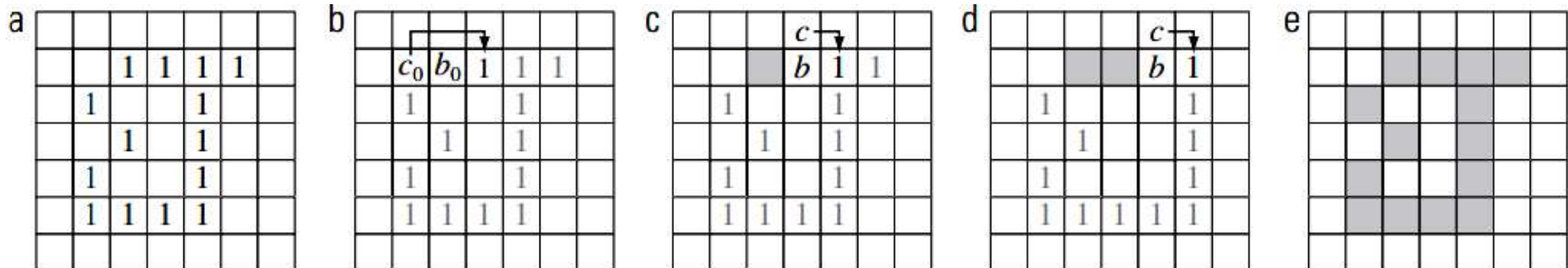
Claramente,  $c_0$  sempre é um ponto do fundo

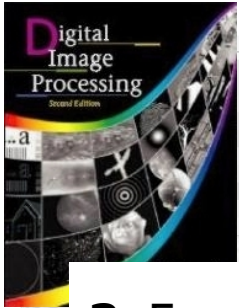
Examine os vizinhos-8 de  $b_0$ , a partir de  $c_0$ , seguindo no sentido horário

Seja  $b_1$  o *primeiro* vizinho encontrado cujo valor é 1 e seja  $c_1$  o ponto (de fundo) imediatamente anterior a  $b_1$  na sequência

Conserve a localização de  $b_0$  e  $b_1$  para utilizá-la na Etapa 5.

2. Considere que  $b = b_1$  e  $c = c_1$  (Fig. c)





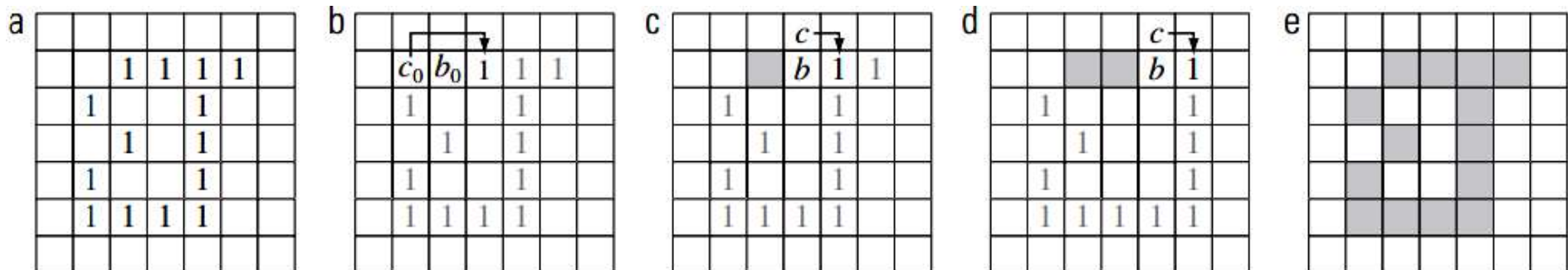
**3.** Faça com que os vizinhos-8 de  $b$ , a partir de  $c$  e seguindo no sentido horário, sejam indicados por  $n_1, n_2, \dots, n_8$

Encontre o primeiro  $n_k$  rotulado com 1

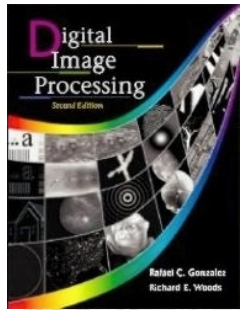
**4.** Considere que  $b = n_k$  e  $c = n_k - 1$

**5.** Repita as etapas 3 e 4 até que  $b = b_0$  e o próximo ponto de fronteira encontrado seja  $b_1$

A sequência de pontos  $b$  encontrados quando o algoritmo para constitui o conjunto de pontos de fronteira ordenados



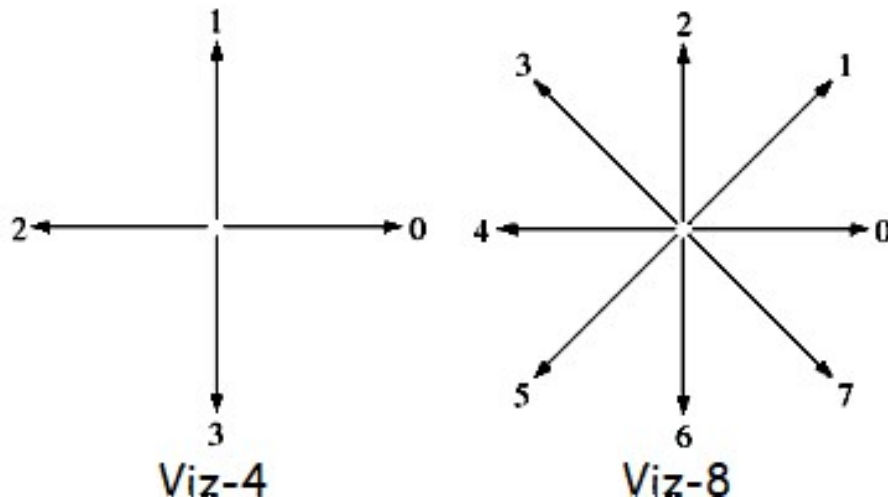




## Representation & Description

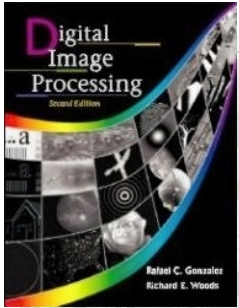
### Chain codes (códigos cadeia)

Códigos de cadeia - São usados para representar uma borda, através de uma sequência conectada de pequenos segmentos de retas. As direções dos segmentos de reta podem ser as apresentadas na figura abaixo



Um código de fronteira formado como uma sequência desses números direcionais é chamado de **código da cadeia de Freeman**





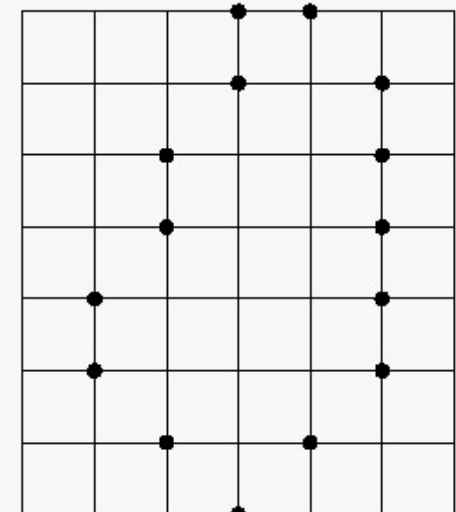
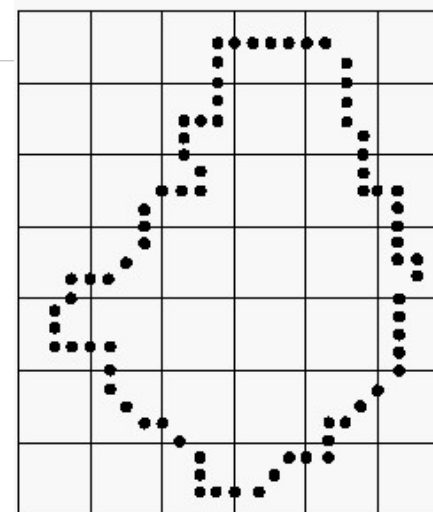
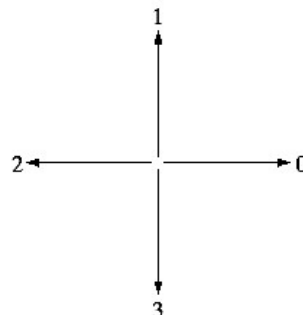
## Representation & Description

### Chain codes (códigos cadeia)

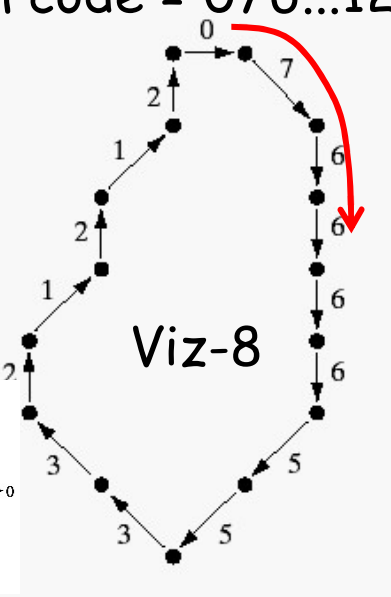
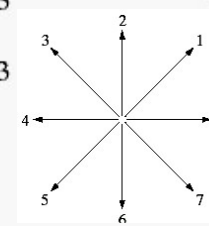
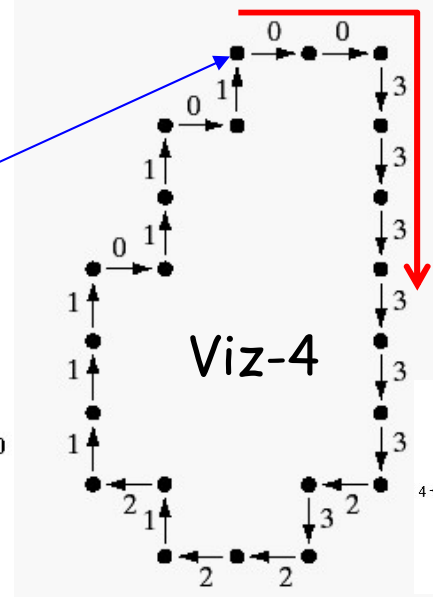
Para obter a cadeia, faz-se uma reamostragem usando uma grade bem larga

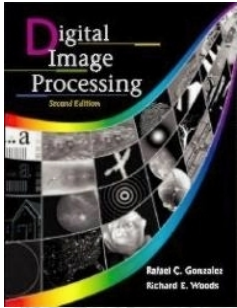
Conforme a fronteira vai sendo percorrida, um ponto na fronteira é atribuído a cada nó da grade maior, em função da proximidade da fronteira original

**Chain code = 0033...01**



Chain code = 076...12





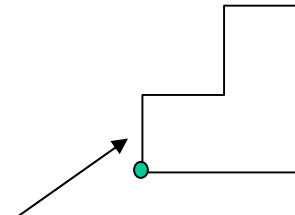
## Representation & Description

### Chain codes (códigos cadeia)

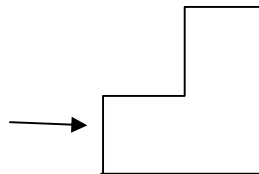
O código depende do ponto de partida, entretanto, o código pode ser normalizado, girando seus valores, até se obter o menor valor possível

#### Exemplo:

chain code 1 = 10103322

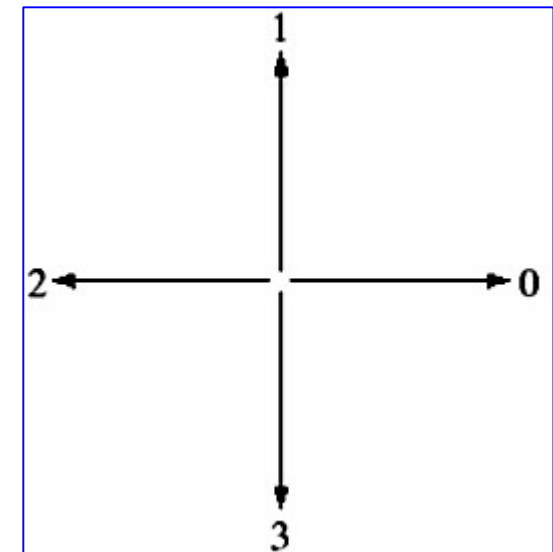
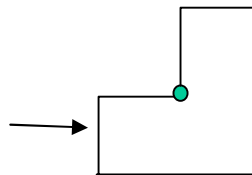


chain code 2 = 32210103



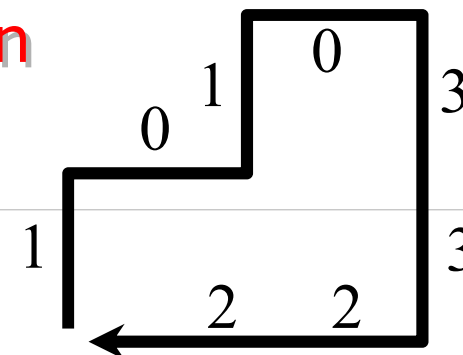
Após girar até  
Mínimo

= 01033221





## Representation & Description

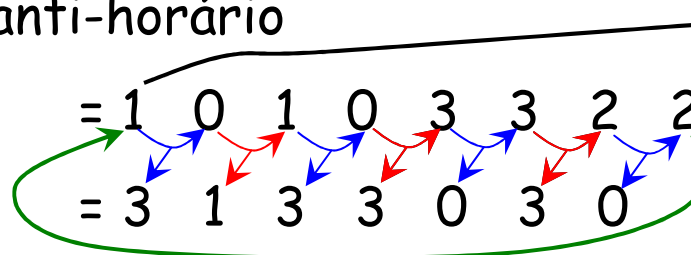


### Chain codes (códigos cadeia)

O código também pode ser normalizado, para ficar invariante a rotação, tomando a primeira diferença entre os valores na cadeia, adotando o sentido anti-horário

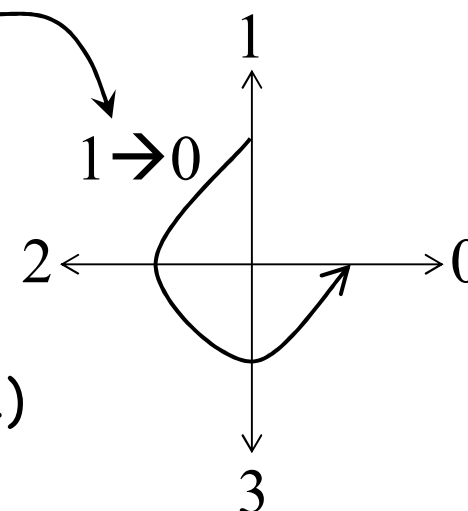
Ex: chain code

Primeira diferença

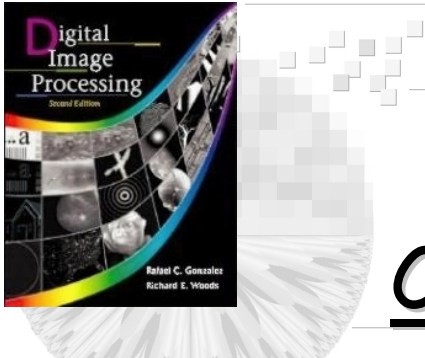


Por fim, a sequência recebe no início

A diferença (3) entre o último (2) e o primeiro (1) (2→1)  
elemento da sequência, ficando: **33133030**

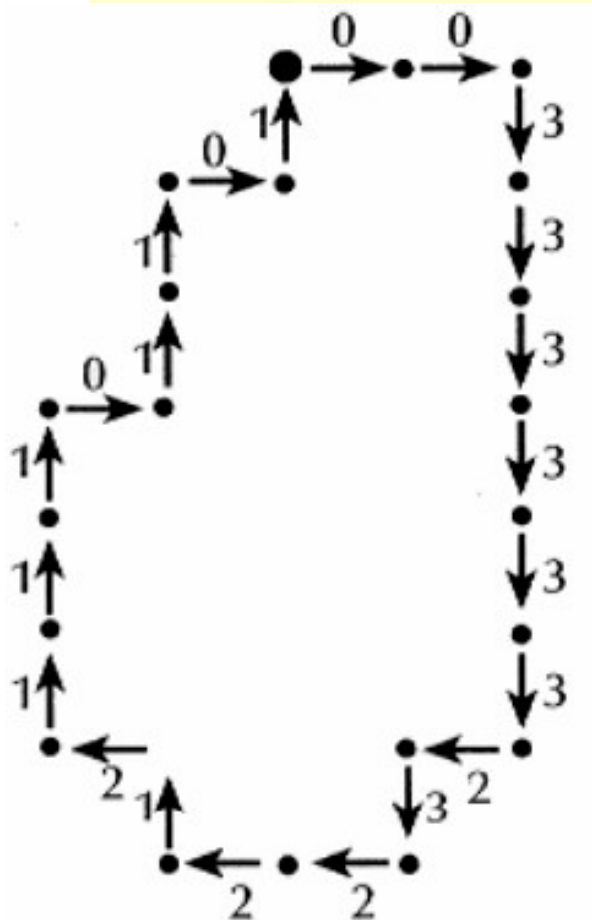


**De 1 até chegar no 0 são 3 passos**  
**De 2 até chegar no 0 são 2 passos**  
**etc.**



# Representation & Description

## Exemplo:



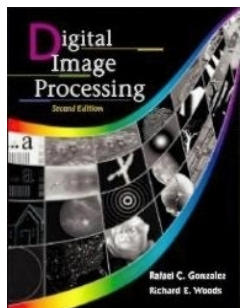
Código da Cadeia:

0033333323221211101101

$$1 \Rightarrow 0 = -\frac{\pi}{2} = 3$$

## Código com a Primeira Diferença (Derivativo)

3 0 3 0 0 0 0 3 1 3 0 3 1 3 0 0 3 1 0 3 1



## Representation & Description

### Exemplo

a) Imagem ruidosa

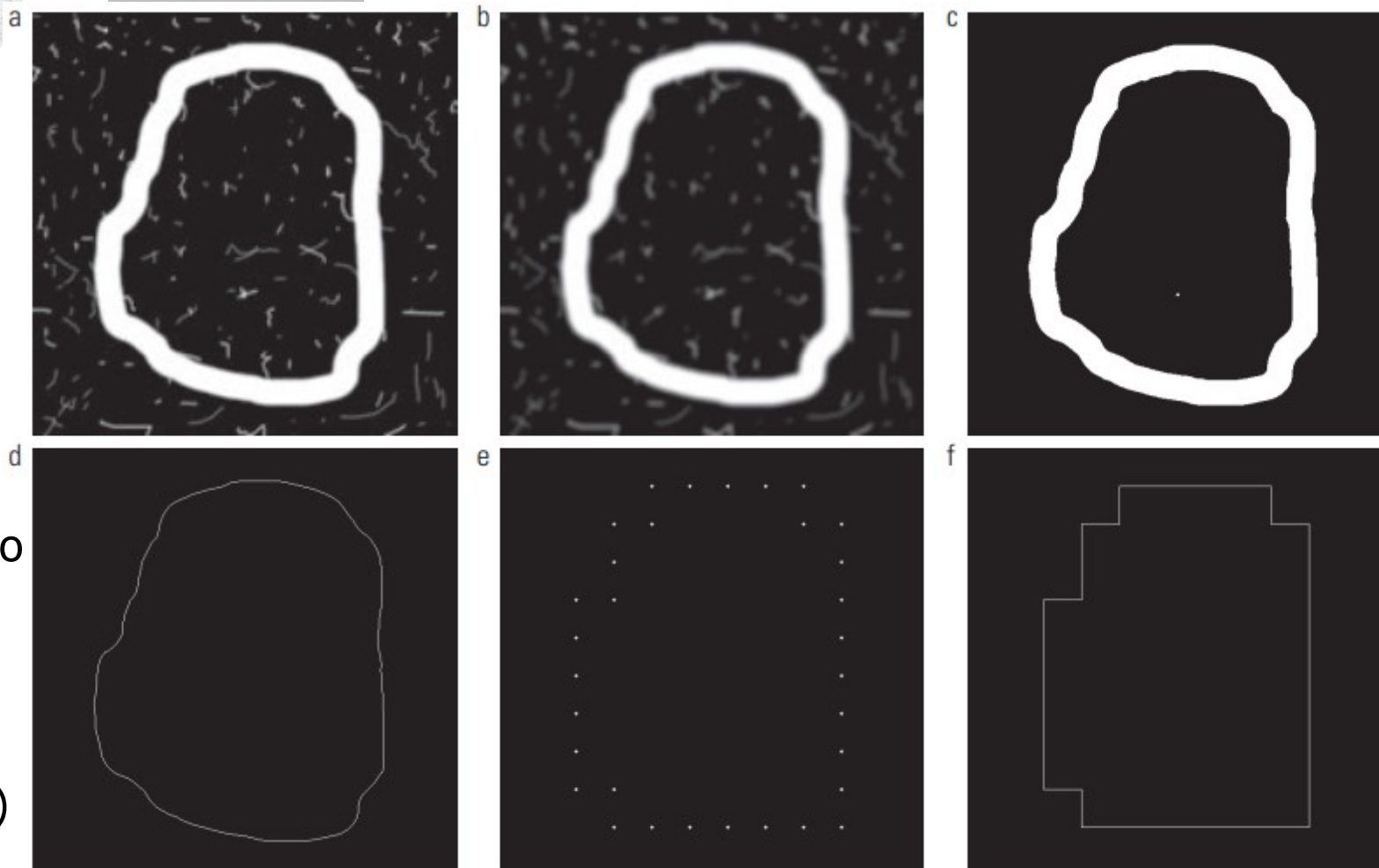
b) Imagem suavizada com média  $9 \times 9$

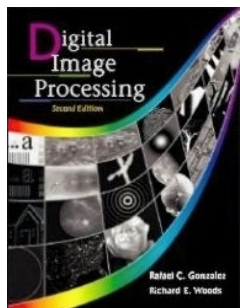
c) Limiarização de (b) com Otsu

d) Borda externa de (c)

e) Fronteira subamostrada

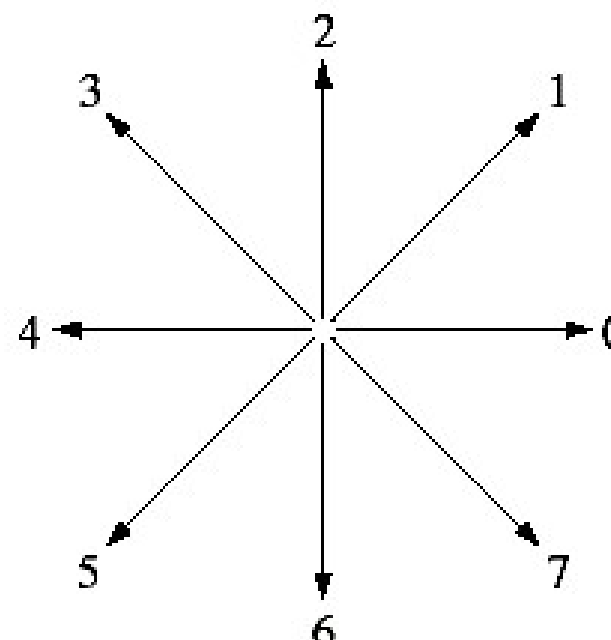
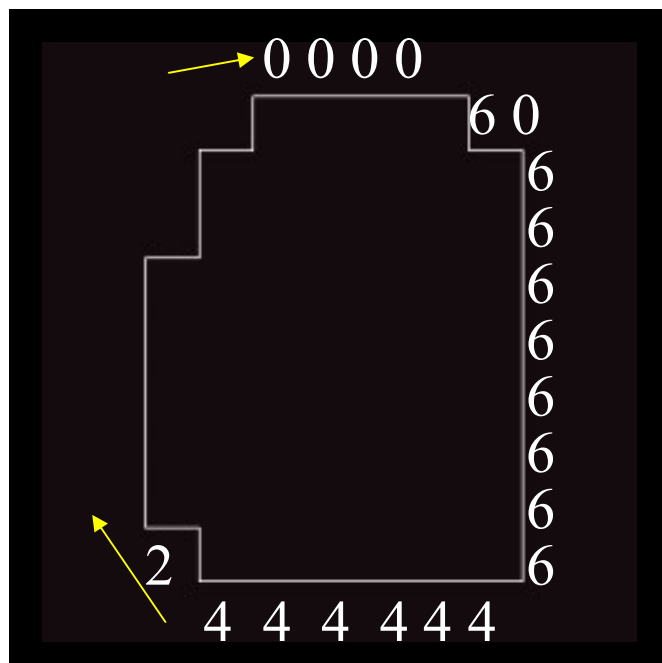
f) Pontos conectados a partir de (e)





# Representation & Description

## Exemplo



O código da cadeia de Freeman de oito direções da fronteira simplificada é

0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2

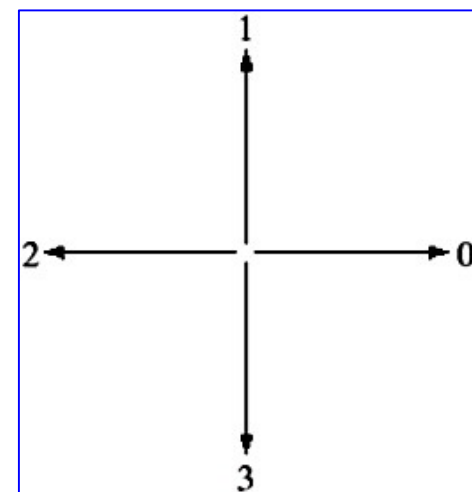
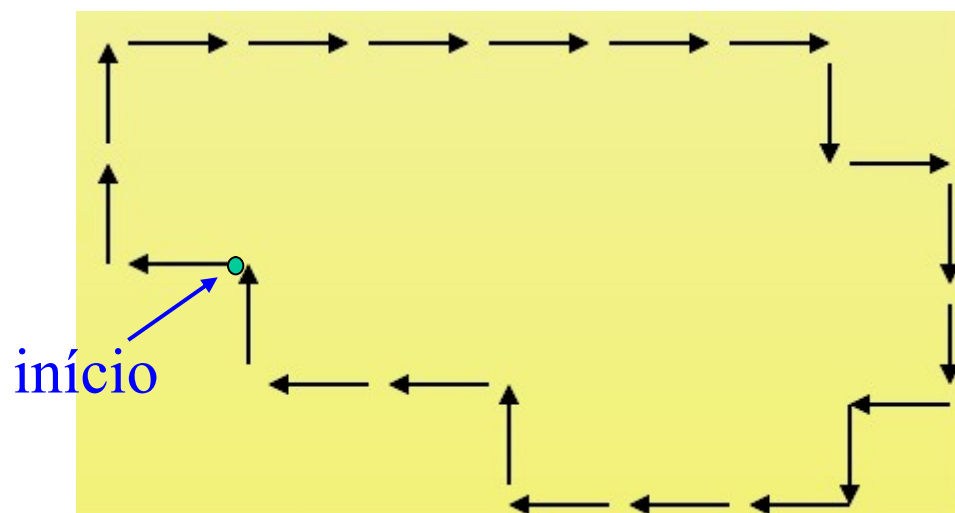




## Representation & Description

### Exercício para entregar

Obtenha o código da cadeia para a forma abaixo, gire até obter a menor diferença e calcule a primeira diferença



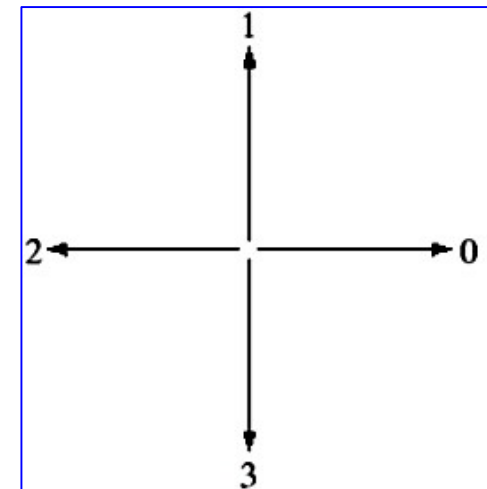
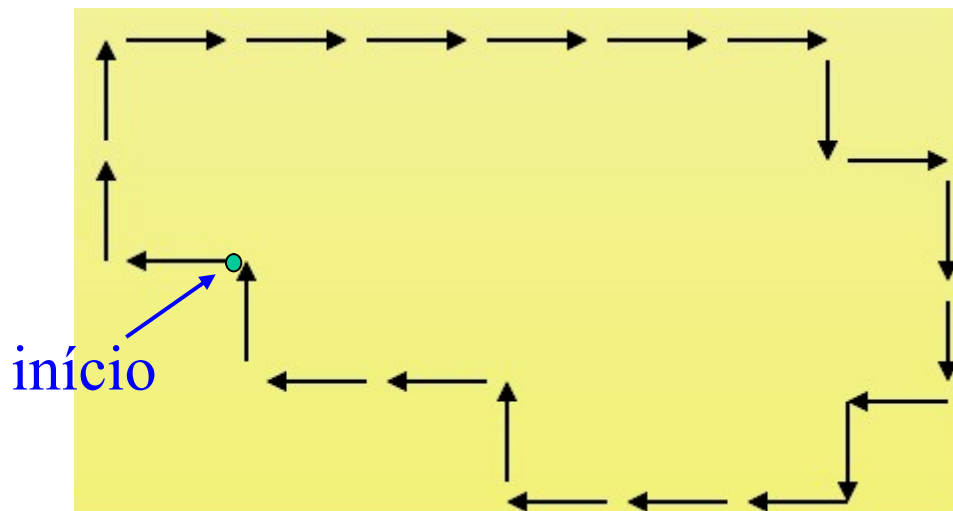




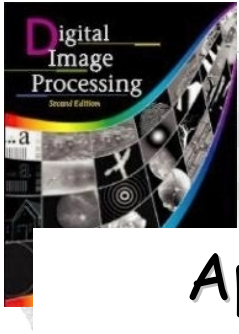
# Representation & Description

## Exercício para entregar

Obtenha o código da cadeia para a forma abaixo usando a primeira diferença e inserindo o a diferença do ultimo para o primeiro no início



**Solução:** 3 0 0 0 0 0 3 1 3 0 3 1 3 0 0 3 1 0 3 1 3 0

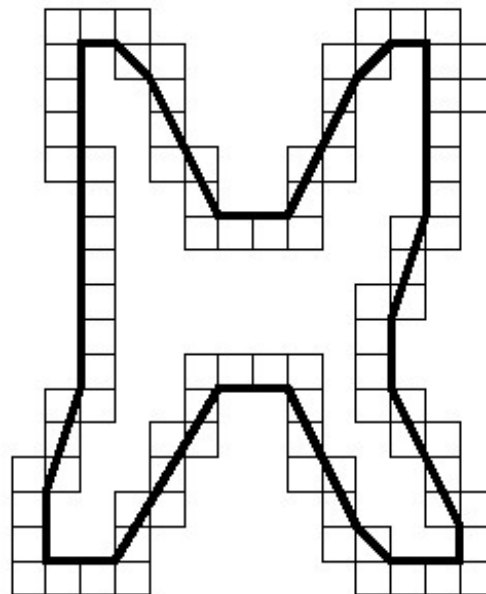
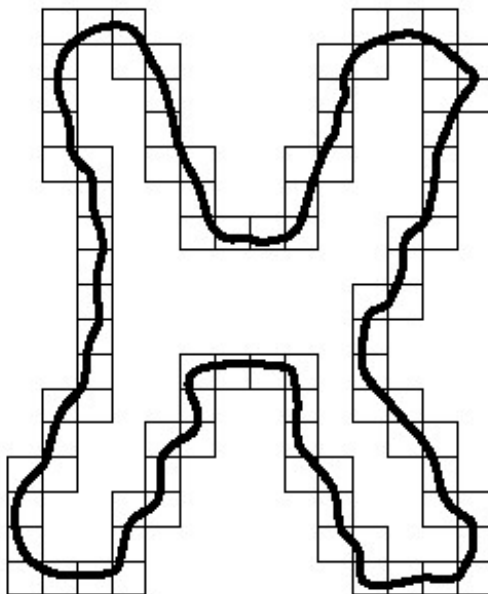


## Representation & Description

### Aproximação poligonal (MPP - *minimum-perimeter polygon* )

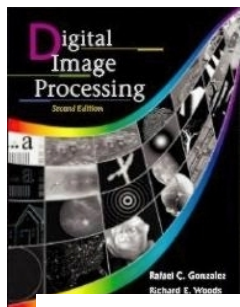
Se baseia na suposição de que uma borda pode ser aproximada por um polígono

Encontrar o polígono não é uma tarefa trivial, mas existem alguns métodos que apresentam resultados satisfatórios, com pouco esforço computacional, como é o caso do método que procura encontrar o perímetro mínimo do polígono



o objeto é delimitado por células e então é encontrado o perímetro mínimo

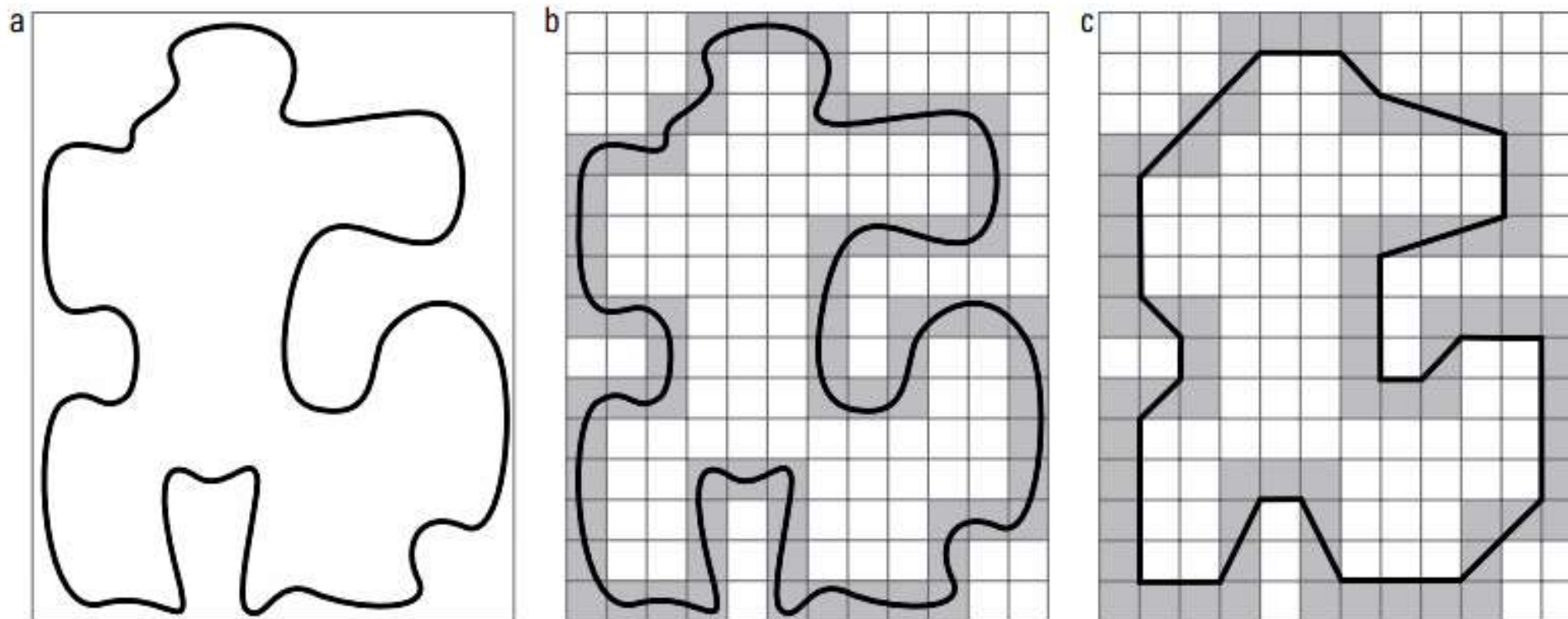
Supor que a borda é composta por um fio de borracha que então é esticado

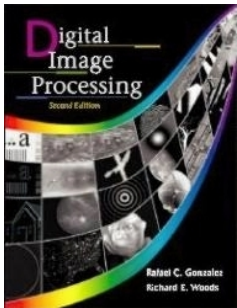


## Representation & Description

### Aproximação poligonal (MPP - *minimum-perimeter polygon* )

o objeto é delimitado por células e então é encontrado o perímetro mínimo. Supor que a borda é composta por um fio de borracha que então é esticado

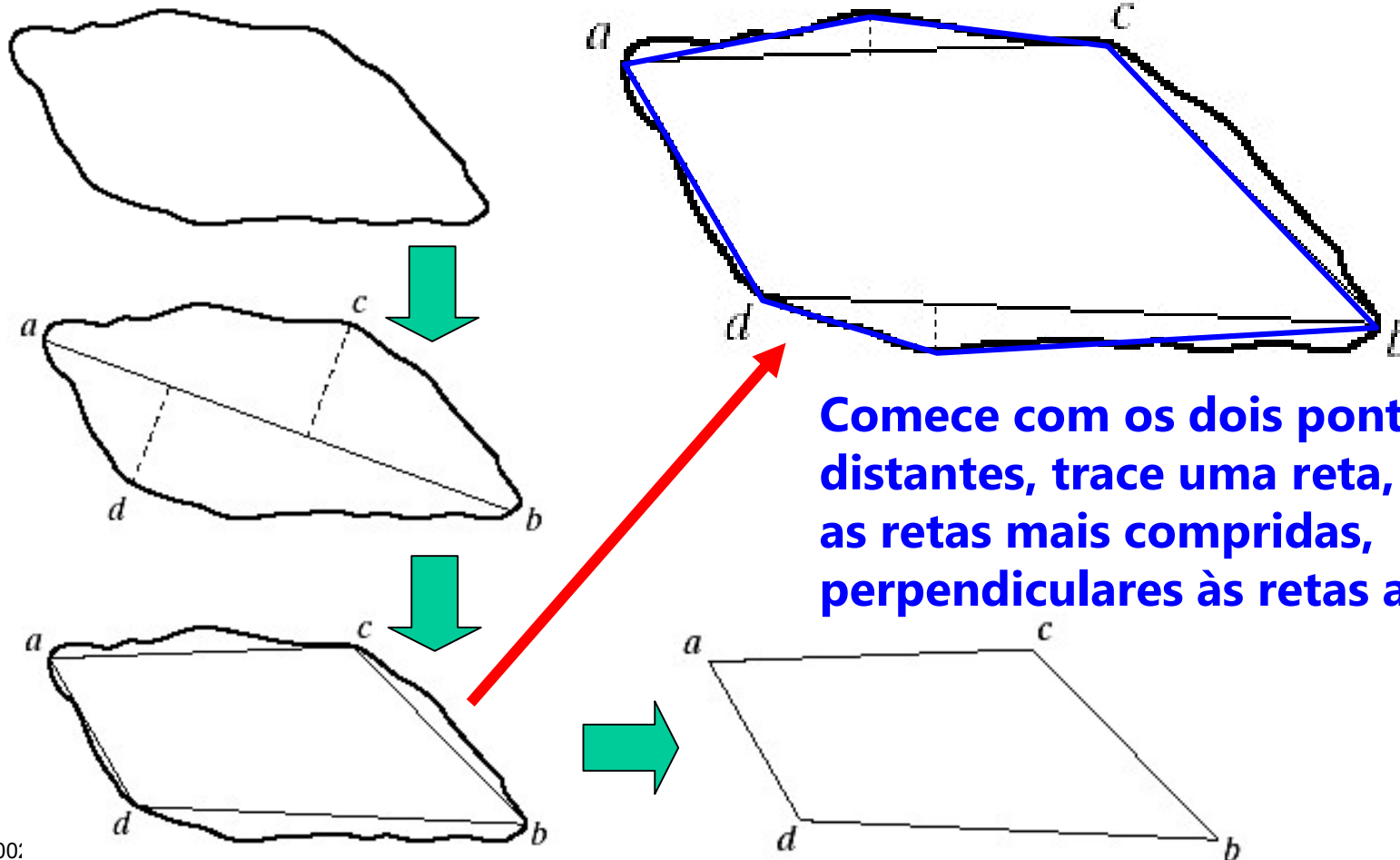




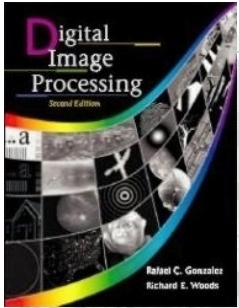
## Representation & Description

### Aproximação poligonal

Uma outra proposta consiste em encontrar retas que se ajustam ao contorno, usando um método de aproximação sucessiva



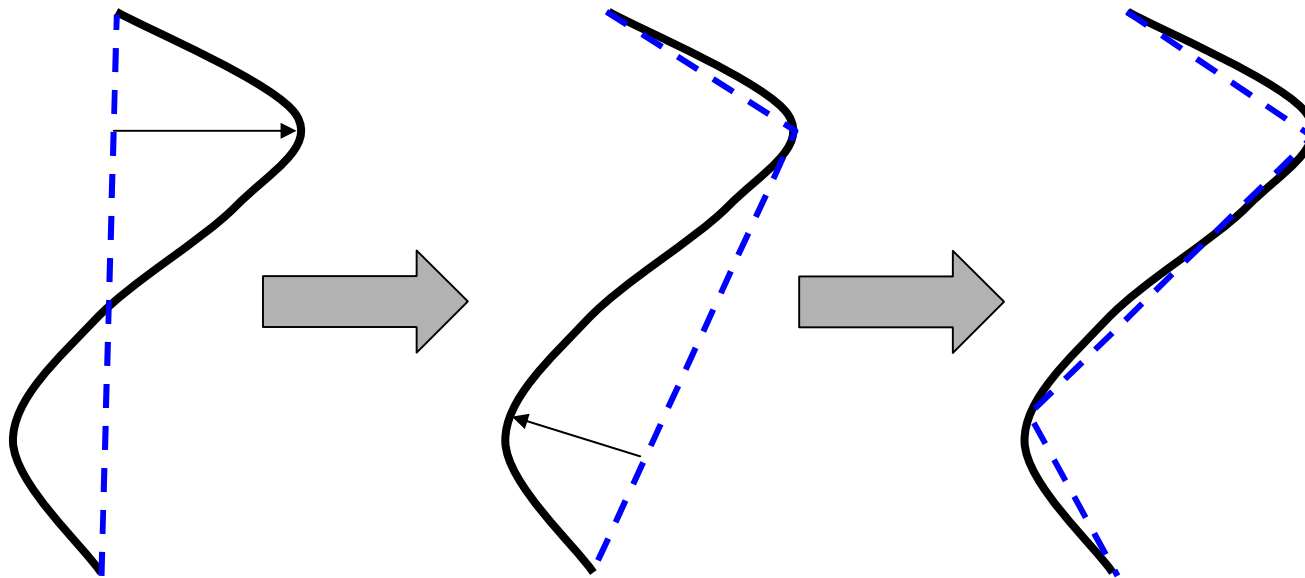
**Comece com os dois pontos mais distantes, trace uma reta, vá obtendo as retas mais compridas, perpendiculares às retas atuais**

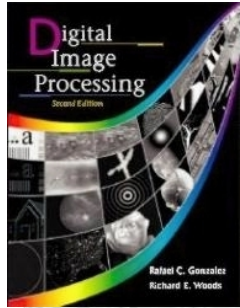


## Representation & Description

### Aproximação poligonal

Uma outra proposta consiste em encontrar retas que se ajustam ao contorno, usando um método de aproximação sucessiva





## Representation & Description

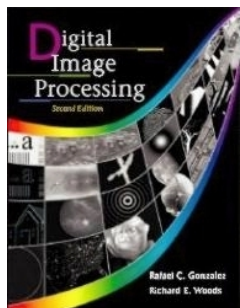
# Assinaturas

Consistem em uma representação unidimensional da fronteira

- pode ser obtida de várias maneiras

Assinaturas são:

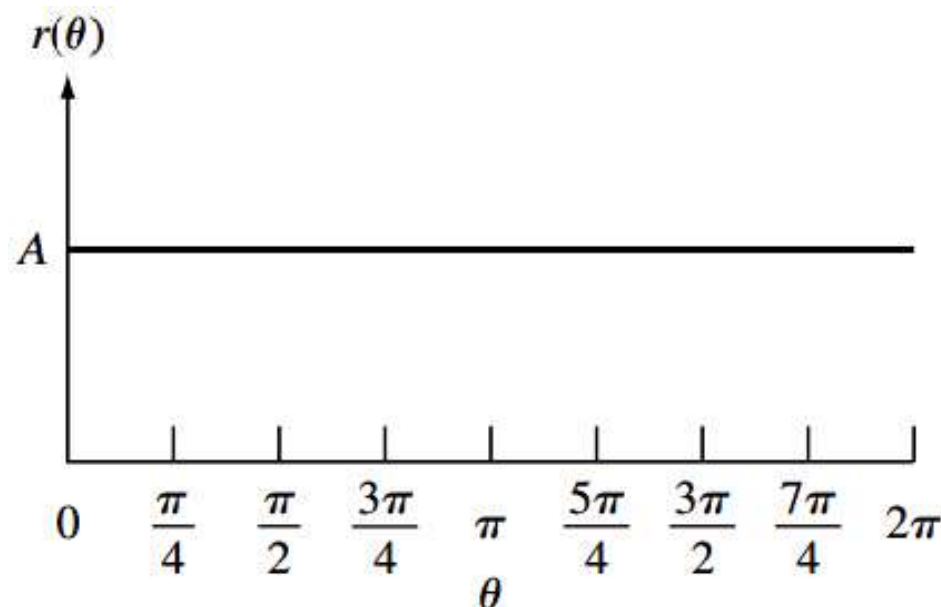
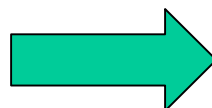
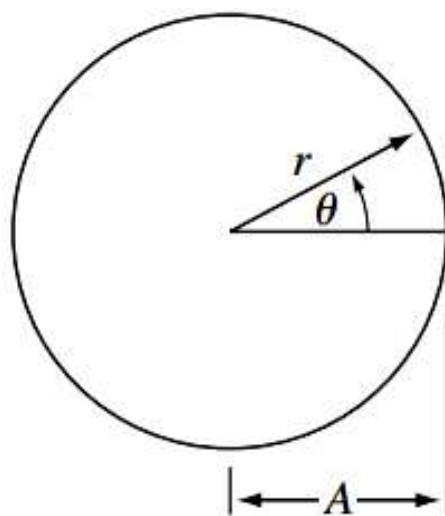
- invariantes a translação
- dependentes da rotação e escala



## Representation & Description

### Assinaturas

Pode ser obtida pelo gráfico da distância da fronteira ao centroide em função do ângulo

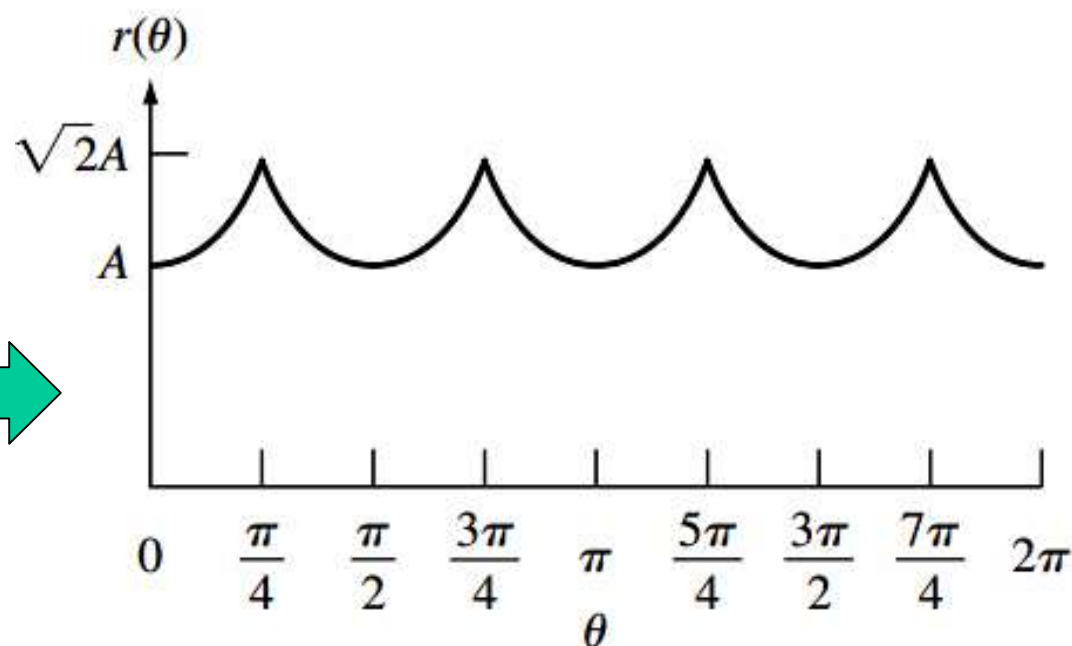
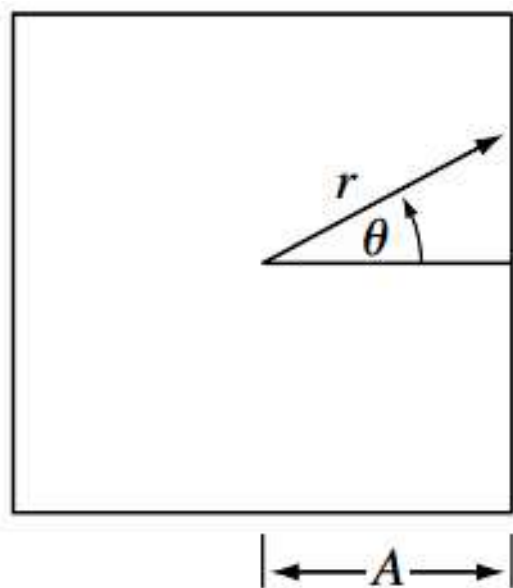






## Representation & Description

### Assinaturas



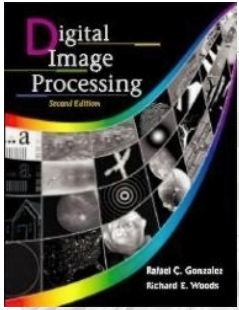


## Representation & Description

### Assinaturas

Maneiras de normalizar em relação a rotação:

- Iniciar sempre no ponto mais afastado do centróide (quando existir um único ponto que satisfaça esta condição)
- Selecionar o ponto no eixo mais alongado
- Obter a assinatura e depois tratá-la, do mesmo modo usado com o código da cadeia

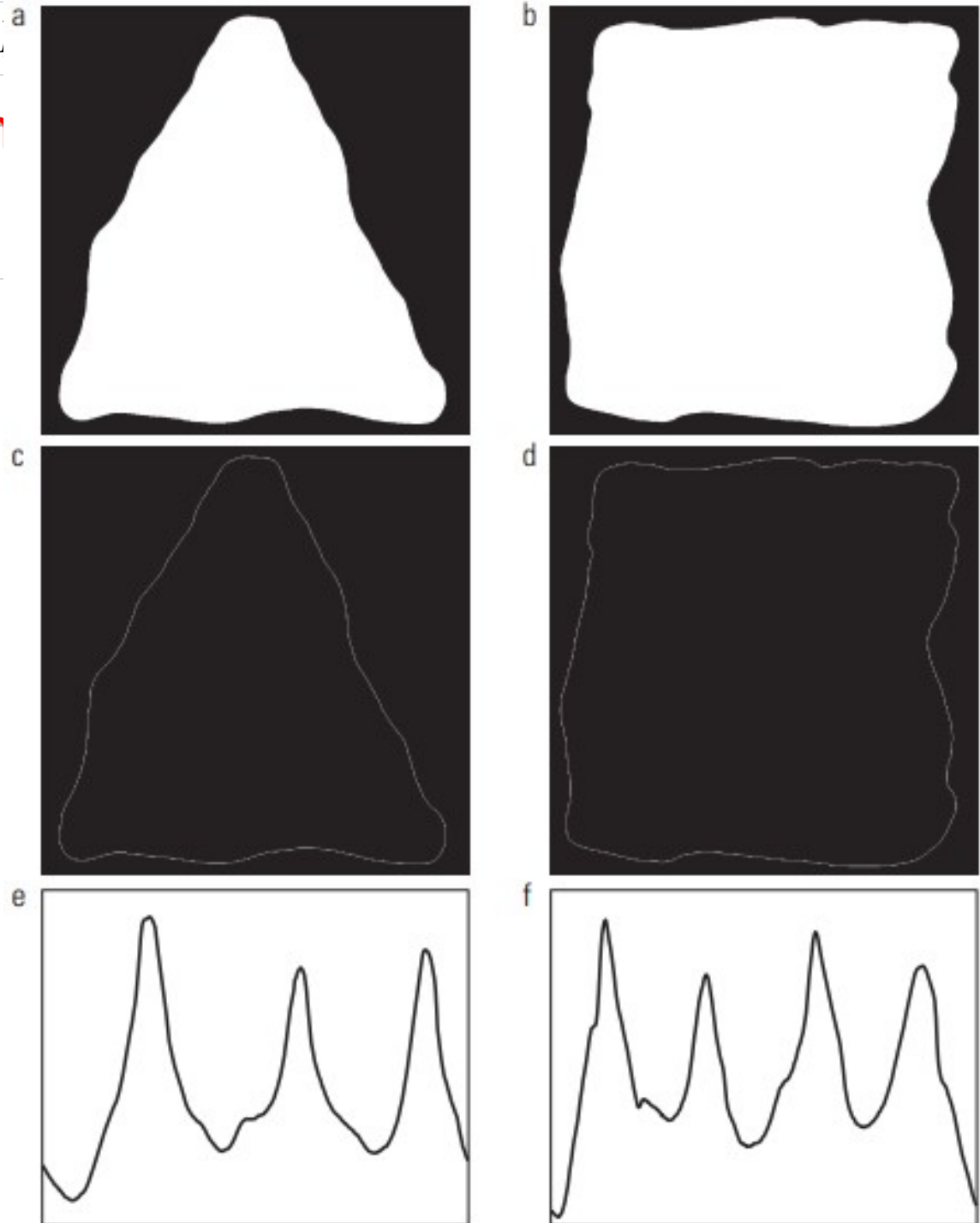


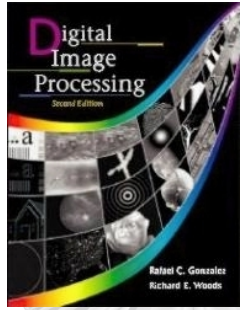
# Representation & Description

a) e b) Duas regiões binárias

c) e d) suas fronteiras  
externas

e) e f) suas assinaturas  $r(\theta)$   
correspondentes (eixos  
horizontais em ângulos de  
 $0^\circ$  a  $360^\circ$ )





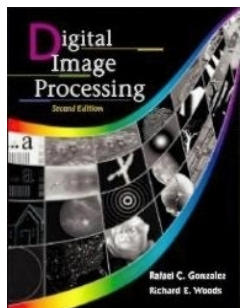
## Representation & Description

### Segmentos de fronteira

A decomposição de uma fronteira em segmentos reduz a complexidade da fronteira, simplificando o processo de descrição

É mais atrativa quando a fronteira possui uma ou mais concavidades significativas que carregam a informação do formato

O uso do fecho convexo da região delimitada pela fronteira é útil para decompor a fronteira



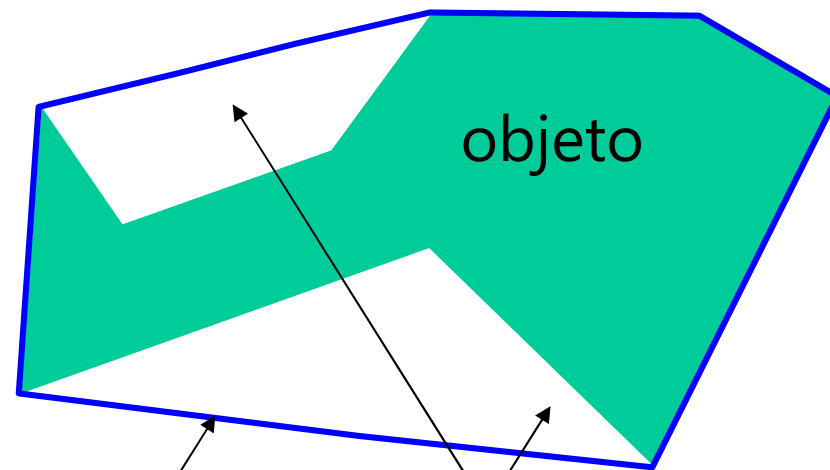
## Representation & Description

### Segmentos de fronteira - **Fecho convexo**

O fecho convexo  $H$  de um conjunto arbitrário  $S$  é o menor conjunto convexo que contém  $S$

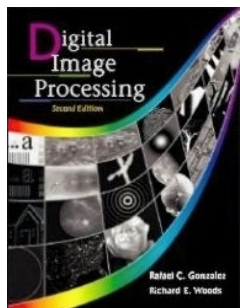


fecho convexo



fecho convexo      deficiência convexa

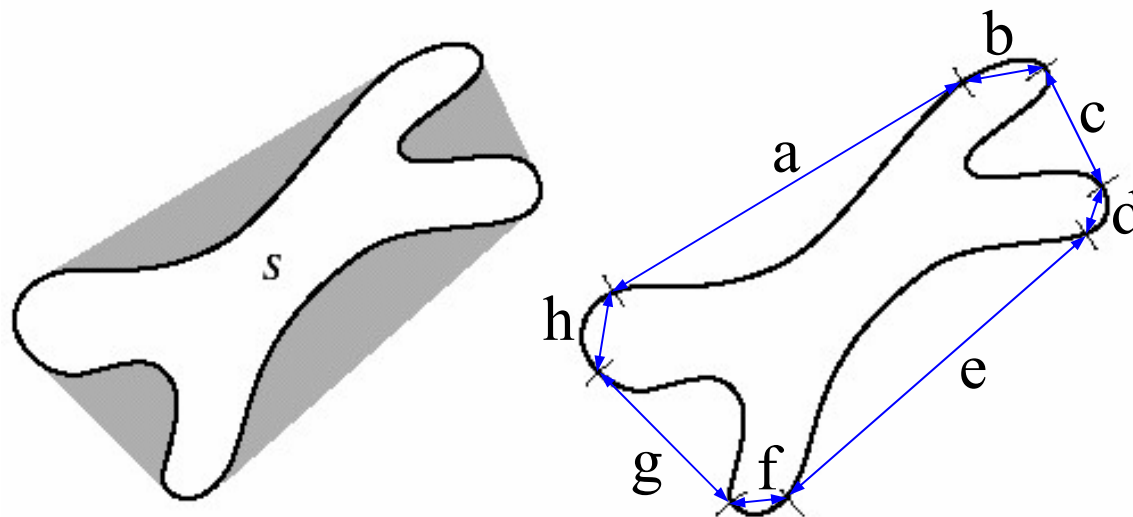
**Um conjunto convexo  
é um objeto que não  
possui concavidades**



## Representation & Description

### Fecho convexo

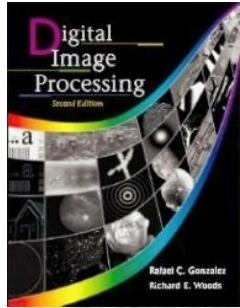
A fronteira pode ser particionada anotando os pontos que ela entra e sai do fecho



### Descrição

As razões entre os comprimentos obtidos é invariante a escala, rotação e translação, também se pode obter um grafo a partir dos pontos e proceder a análise sobre ele

$$\frac{a}{b}, \frac{a}{c}, \frac{a}{d}, \dots, \frac{b}{c}, \frac{b}{d}, \dots, \frac{c}{d}, \frac{c}{e}, \dots,$$



## Representation & Description

# Fecho convexo

Por exemplo,

A **descrição** de uma região também pode estar baseada em:

- sua área e na área de sua deficiência convexa
- no número de componentes na deficiência convexa
- na posição relativa desses componentes
- etc.

A aula 12 apresenta um algoritmo morfológico para encontrar o fecho convexo





## Representation & Description

# Esqueletos

Uma maneira de representar a estrutura de uma forma qualquer é reduzi-la a um gráfico, o que pode ser feito obtendo o esqueleto da forma

O esqueleto pode ser obtido através de algum processo de afinamento (*thinning ou skeletonizing*)

Estes métodos são intensamente usados em problemas de inspeção automática

Os esqueletos podem ser obtidos por uma transformação de eixo médio (*MAT – medial axis transform*)



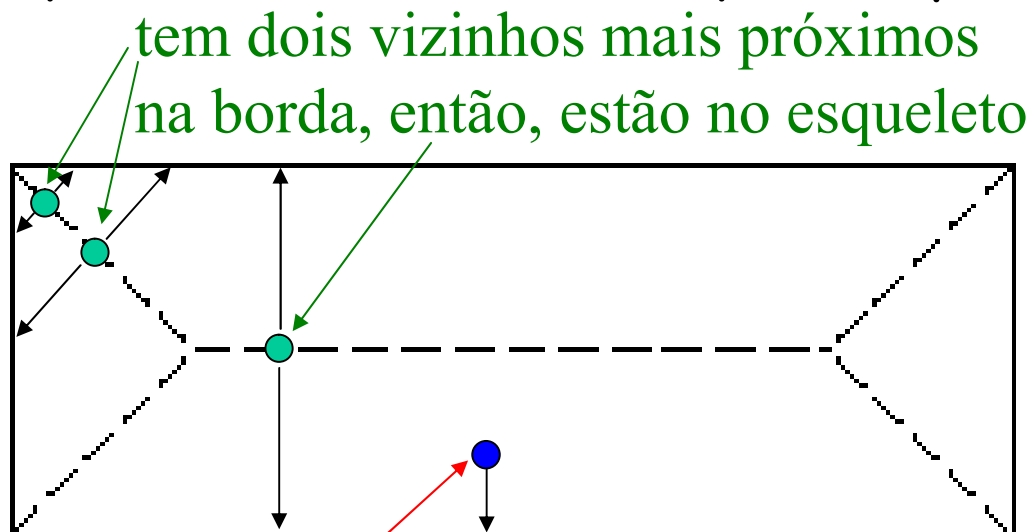
## Representation & Description

### Esqueletos

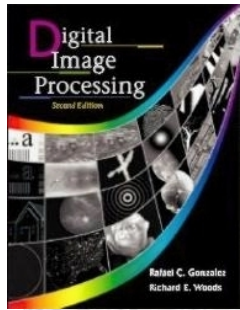
*MAT - medial axis transform*

*Obtenção do Esqueleto de uma região R com borda B*

- Para cada ponto  $p$  in  $R$ , encontre seu vizinho mais próximo em  $B$ . Se  $p$  tem mais que um vizinho, diz-se que ele pertence ao esqueleto de  $R$



só tem um vizinho mais próximo na borda, logo não está no esqueleto

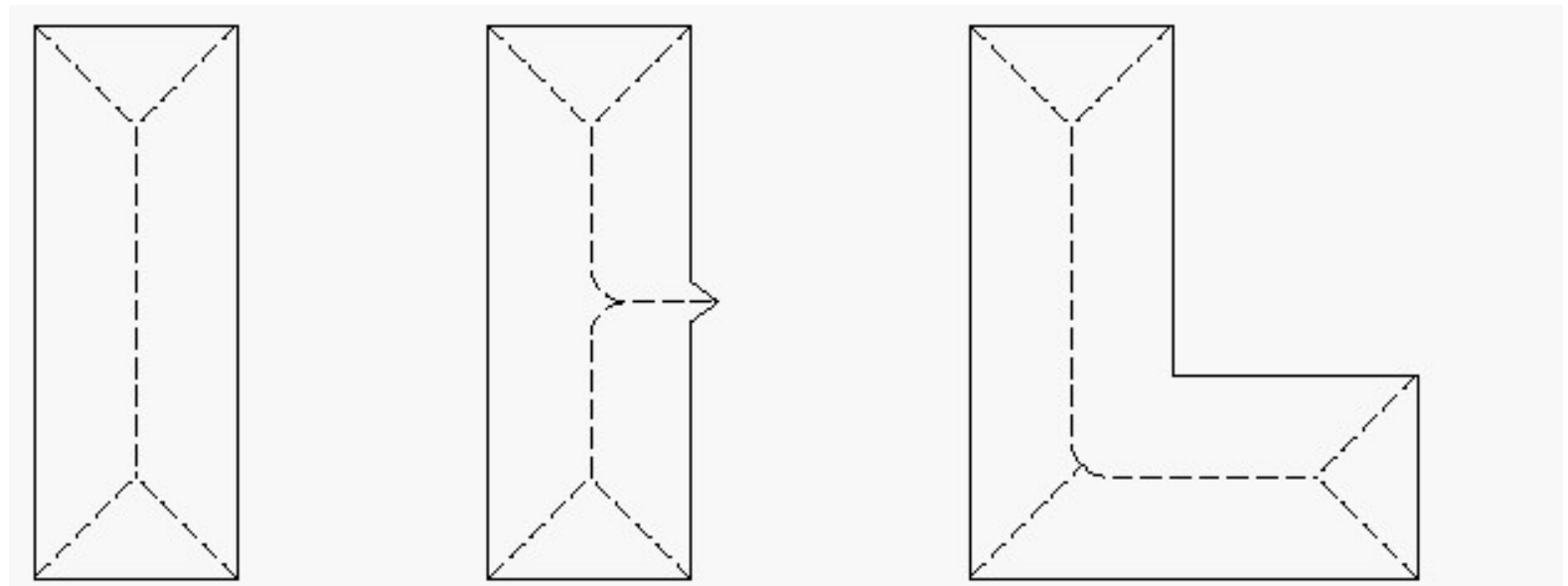


## Representation & Description

### Esqueletos

*MAT - medial axis transform*

**O problema desta abordagem é a necessidade de se calcular as distâncias entre todos os pontos da imagem**





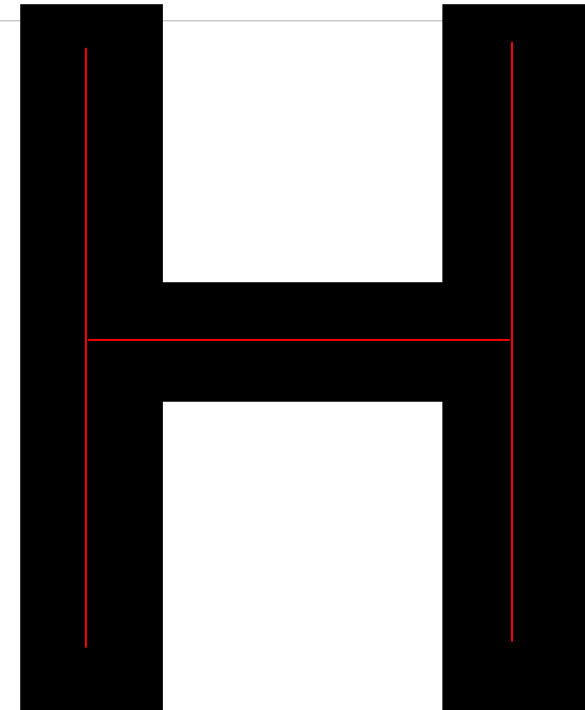
## Representation & Description

### Esqueletos

Afinamento de imagens binárias  
(Zhang e Suen, 1984)

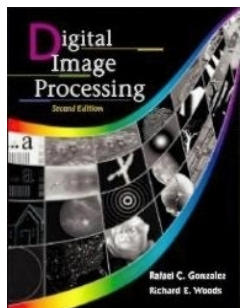
Assume-se que:

- objeto = 1
- fundo = 0



**vai eliminar os pontos externos, até ficar apenas o esqueleto**

Zhang, T.Y., and Suen, C.Y. [1984]. "A Fast Parallel Algorithm for Thinning Digital Patterns." *Comm. ACM*, vol. 27, no. 3, pp. 236–239.



## Representation & Description

### Esqueletos - Afinamento de imagens binárias

O método é aplicado a todos os pontos de borda, através de dois passos sucessivos:

#### Passo 1

a)  $2 \leq N(p_1) \leq 6$

b)  $S(p_1) = 1$

c)  $p_2 \cdot p_4 \cdot p_6 = 0$

d)  $p_4 \cdot p_6 \cdot p_8 = 0$

O ponto é marcado para ser eliminado se satisfaz as 4 condições do passo 1

Produto

pode ser:  $P_2$  ou  $P_4$  ou  $P_6 = 0$   
e  $P_4$  ou  $P_6$  ou  $P_8 = 0$

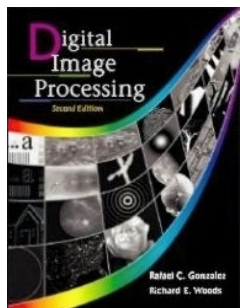
$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$

Onde:  $N(p_1)$  é o número de vizinhos não nulos de  $p_1$

$$N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$$

e  $S(p_1)$  é o número de transições 0-1 na sequência

$$p_2, p_3, \dots, p_8, p_9, p_2$$



## Representation & Description

### Esqueletos - Afinamento de imagens binárias

**Exemplo:**

0	0	1
1	$p_1$	0
1	0	1

$$a) N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$$

$$N(p_1) = 0 + 1 + 0 + 1 + 0 + 1 + 1 + 0 = 4$$

$$\mathbf{N(p_1) = 4}$$

**b) FALHOU**

b)  $S(p_1)$  = número de transições 0-1 na sequência  $p_2, p_3, \dots, p_8, p_9, p_2$

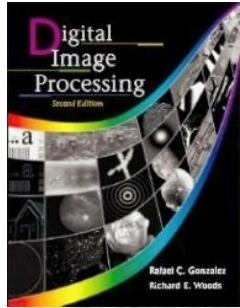
$$p_2 \rightarrow p_3 = p_4 \rightarrow p_5 = p_6 \rightarrow p_7 = 0 \rightarrow 1 \rightarrow \mathbf{S(p1) = 3}$$

$$c) p_2 \cdot p_4 \cdot p_6 = 0 \quad \mathbf{ok}$$

$$d) p_4 \cdot p_6 \cdot p_8 = 0 \quad \mathbf{ok}$$

Logo,  $p_1$  **NÃO** deve ser marcado para ser excluído, após todos os pontos da imagem serem processados

$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$



## Representation & Description

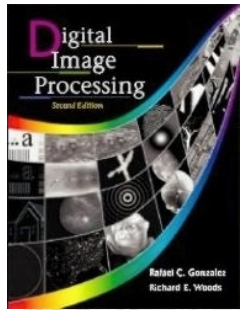
### Esqueletos - Afinamento de imagens binárias

Após todos os pontos de borda terem sido processados no passo 1, os pixels marcados para apagar, devem ser apagados

#### Pixel de Borda

Se um pixel = 1 e tem 8 vizinhos = 1, então, ele não é pixel de borda





## Representation & Description

### Esqueletos - Afinamento de imagens binárias

$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$

$$a) N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$$

$$b) S(p_1) = \text{número de transições 0-1 na sequência } p_2, p_3, \dots, p_8, p_9, p_2$$

No passo 2, as condições (a) e (b) continuam as mesmas, mas os passos (c) e (d) ficam

$$c') p_2 \cdot p_4 \cdot p_8 = 0$$

$$d') p_2 \cdot p_6 \cdot p_8 = 0$$

Quando todas as condições são satisfeitas, o ponto é marcado para ser apagado, o que deve ser feito apenas depois que todos os pontos forem processados



## Representation & Description

### Esqueletos - Afinamento de imagens binárias

A condição (a) é violada quando o ponto  $p_1$  do contorno possui apenas um ou sete vizinhos com valor 1

O fato de possuir apenas um vizinho implica que  $p_1$  é uma extremidade de um segmento do esqueleto, não devendo ser apagado

Se  $p_1$  possui sete vizinhos, ele não deve ser eliminado, para não gerar um buraco no local

A condição (b) trata de pixels em segmentos com um pixel de largura e, vai prevenir uma quebra nestes segmentos do esqueleto

As condições (c) e (d) são satisfeitas pelo conjunto mínimo de valores ( $p_4 = 0$  ou  $p_6 = 0$ ) ou ( $p_2 = 0$  e  $p_8 = 0$ )



## Representation & Description

### Esqueletos - Afinamento de imagens binárias

#### Algoritmo

#### ***Repita***

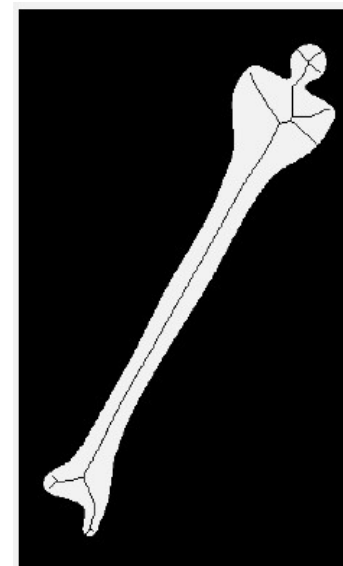
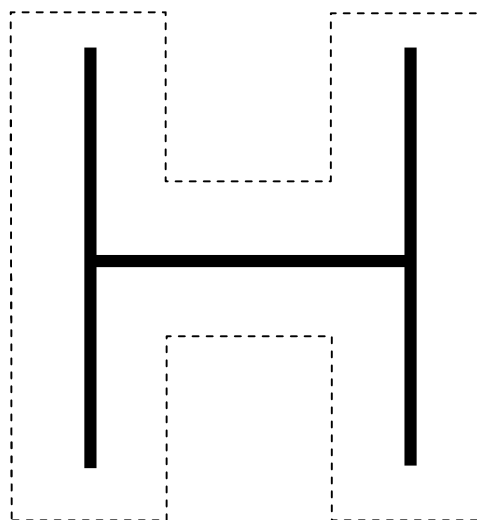
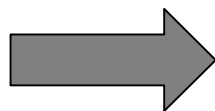
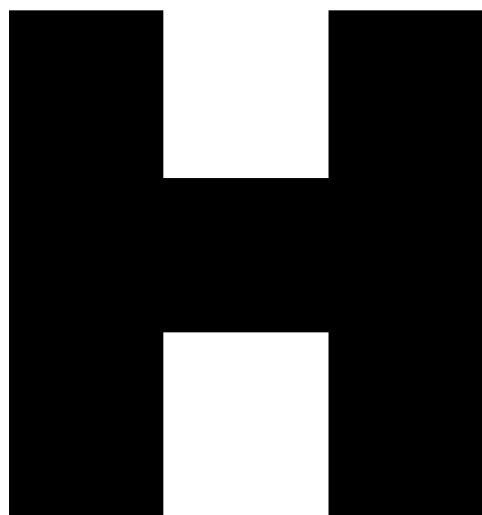
Aplique o passo 1 nos pontos de borda, marque os pontos a apagar

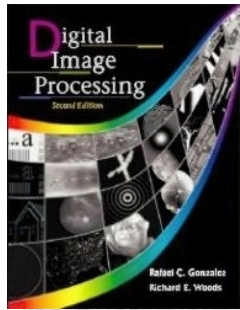
Elimine os pontos que atenderam todas as condições (a), (b), (c) e (d)

Aplique o passo 2 nos pontos de borda, marque os pontos a apagar

Elimine os pontos que atenderam todas as condições (a), (b), (c') e (d')

#### ***Enquanto houver pixels eliminados***





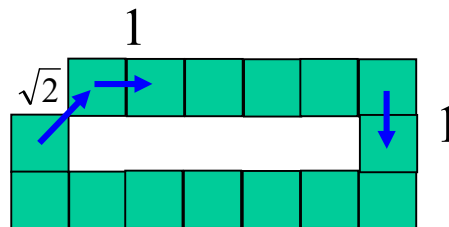
## Representation & Description

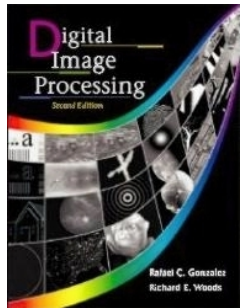
### Descritores de fronteiras (bordas)

A fronteira pode ser usada para discriminar um objeto

Alguns descritores simples são:

- **Comprimento do contorno** - obtido por meio de uma contagem dos pixels ao longo do contorno, oferece uma aproximação do comprimento. No caso de uma fronteira representada pelo código da cadeia, os cálculos ficam facilitados, observando que pixels bordas horizontais e verticais tem tamanho 1 e bordas diagonais tem tamanho  $\sqrt{2}$





## Representation & Description

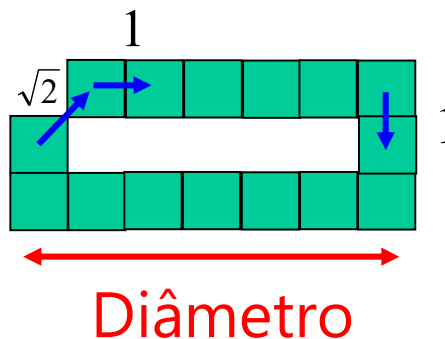
### Descritores de fronteiras (bordas)

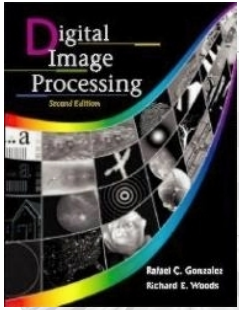
A fronteira pode ser usada para discriminar um objeto

Alguns descritores simples são:

- **Diâmetro da fronteira** - obtido com a maior distância entre dois pontos do contorno

$$\text{Diâmetro}(B) = \max[D(p_i, p_j)]$$

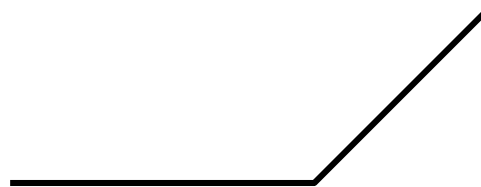


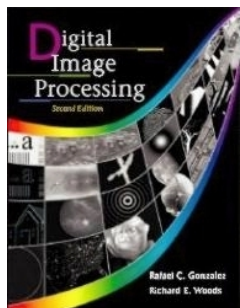


## Representation & Description

### Descritores de fronteiras (bordas)

- **Curvatura** - é definida pela taxa de mudança da inclinação, podendo ser aproximada pela primeira diferença





## Representation & Description

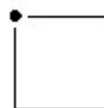
### Números de formas ou formatos

Conforme mencionado anteriormente, a primeira diferença de uma fronteira representada pelo código da cadeia, depende do ponto inicial. O número de forma de tal fronteira é definido como a primeira diferença de menor magnitude

Diferença = Recebe no início a diferença entre o último e o primeiro elemento da sequência

Shape Nº = Vá rotacionando para a esquerda e incluindo o valor que sai pela Esquerda, no lado direito, até obter um valor mínimo (número de forma)

Order 4



Chain code: 0 3 2 1

Difference: 3 3 3 3

Shape no.: 3 3 3 3

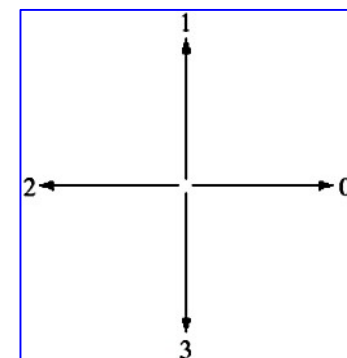
Order 6



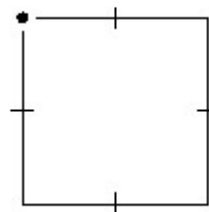
Chain code: 0 0 3 2 2 1

Difference: 3 0 3 3 0 3

Shape no.: 0 3 3 0 3 3



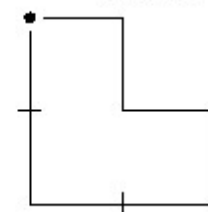
Order 8



Chain code: 0 0 3 3 2 2 1 1

Difference: 3 0 3 0 3 0 3 0

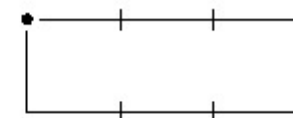
Shape no.: 0 3 0 3 0 3 0 3



Chain code: 0 3 0 3 2 2 1 1

Difference: 3 3 1 3 3 0 3 0

Shape no.: 0 3 0 3 3 1 3 3

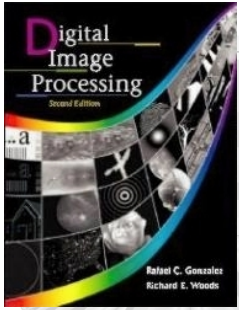


Chain code: 0 0 0 3 2 2 2 1

Difference: 3 0 0 3 3 0 0 3

Shape no.: 0 0 3 3 0 0 3 3

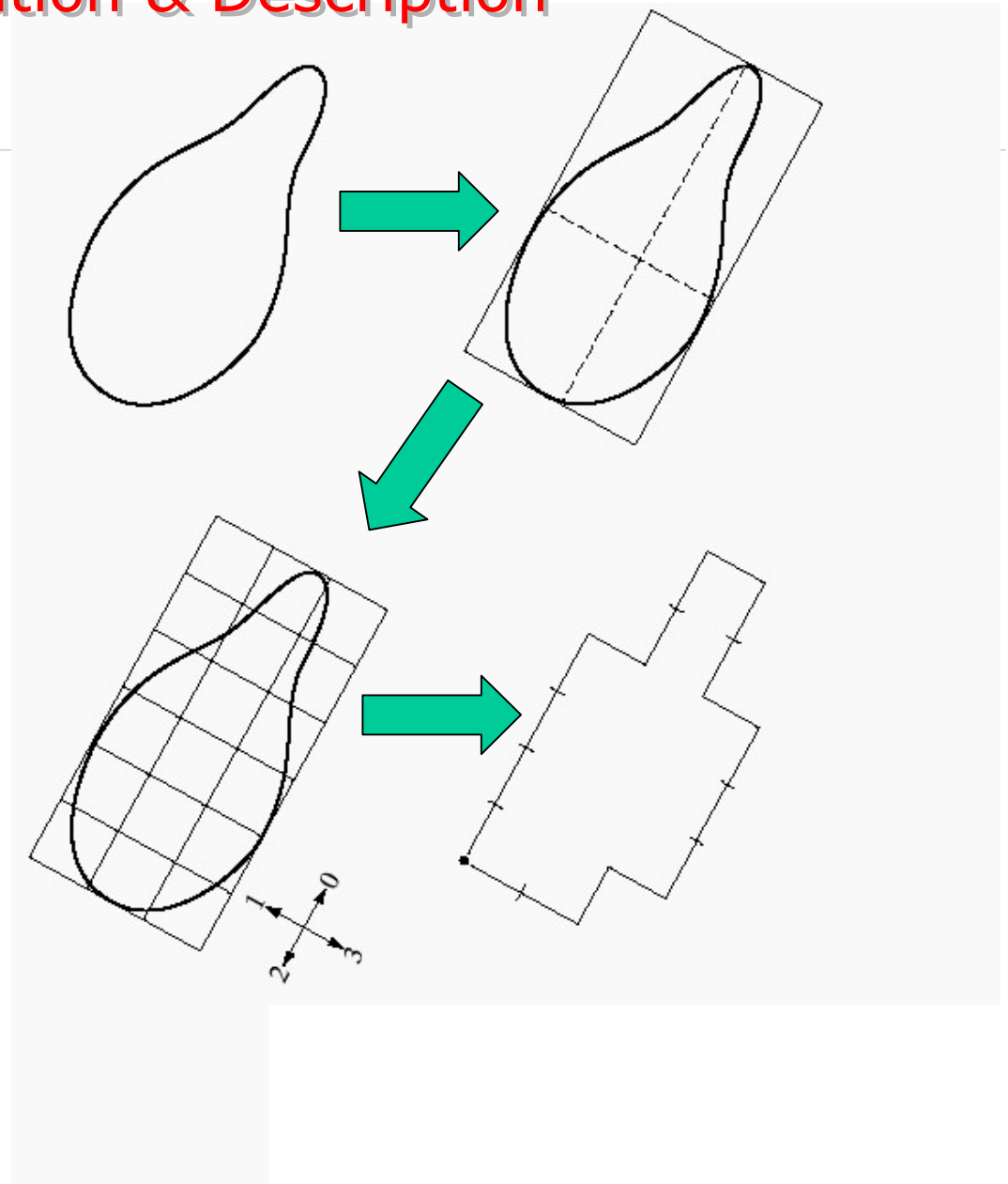




## Representation & Description

### Números de formas

Passos na geração do número de forma



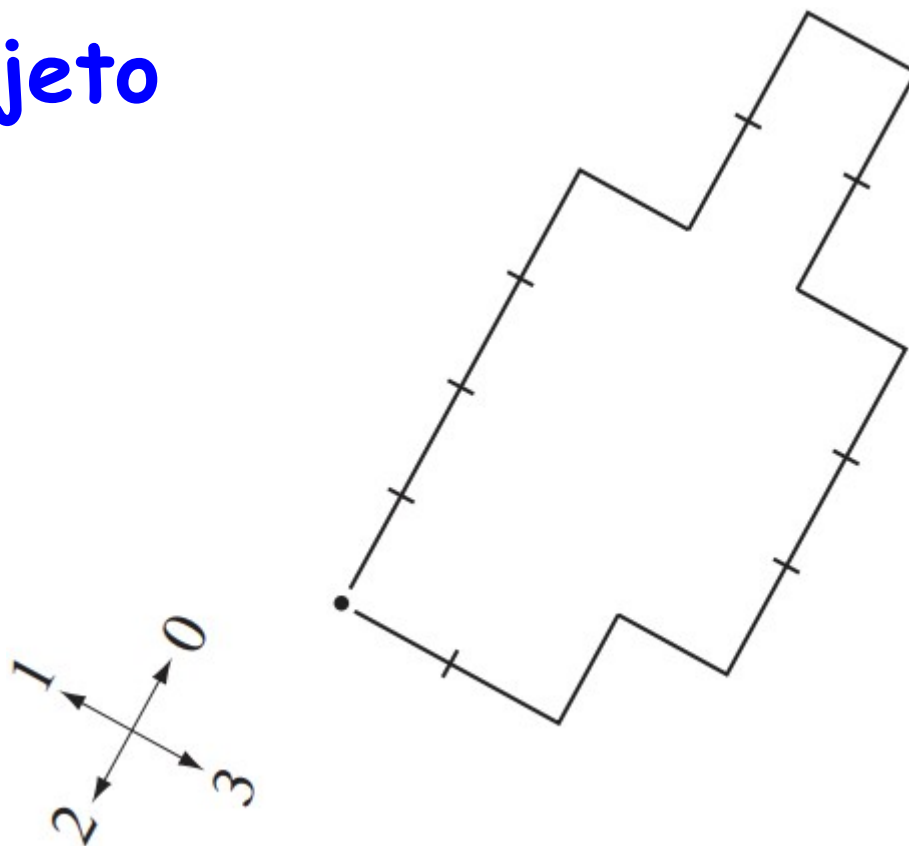




## Representation & Description

# Números de formas

Exercício - encontre o Número de Forma para o objeto



Código da cadeia:  
Diferença:  
Número do formato:

000030032232221211  
300031033013003130  
000310330130031303