



# Morfología Matemática

Guillermo Cámara-Chávez

# Morfologia Matemática



- Foi desenvolvida inicialmente por Georges Matheron e Jean Serra na década de 60
- Baseada na Teoria dos Conjuntos
- Originalmente desenvolvida para análise de imagens microscópicas

# Introdução



- Uma vez segmentada, operações morfológicas podem ser utilizadas para **remover imperfeições** na imagem segmentada e **prover informações** a respeito da **forma e estrutura da imagem**.

# Premissas



- Para realizar as operações morfológicas a imagem deve estar **previamente segmentada**, sendo usualmente 0 (zero) para background e 1 (um) para os objetos
- Em alguns exemplos, representaremos objetos segmentados com pixels pretos e em outros casos como pixels brancos

# O que é Processamento morfológico de imagens ?



- É uma área que descreve **várias técnicas de processamento de imagem** que lidam com as **características de forma** em uma imagem
- Operações morfológica são tipicamente aplicadas na **remoção de imperfeições** introduzidas durante o **processo de segmentação**

# Exemplo



Imagem após segmentação



Imagem após segmentação e  
processamento morfológico

# Definições Básicas



- Sejam  $A$  e  $B$  conjuntos de  $Z^2$ , com componentes:

$$a = (a_1, a_2) \quad b = (b_1, b_2)$$

$$\text{Sendo } z = (z_1, z_2)$$

**Translação** de  $A$  por  $x$ :

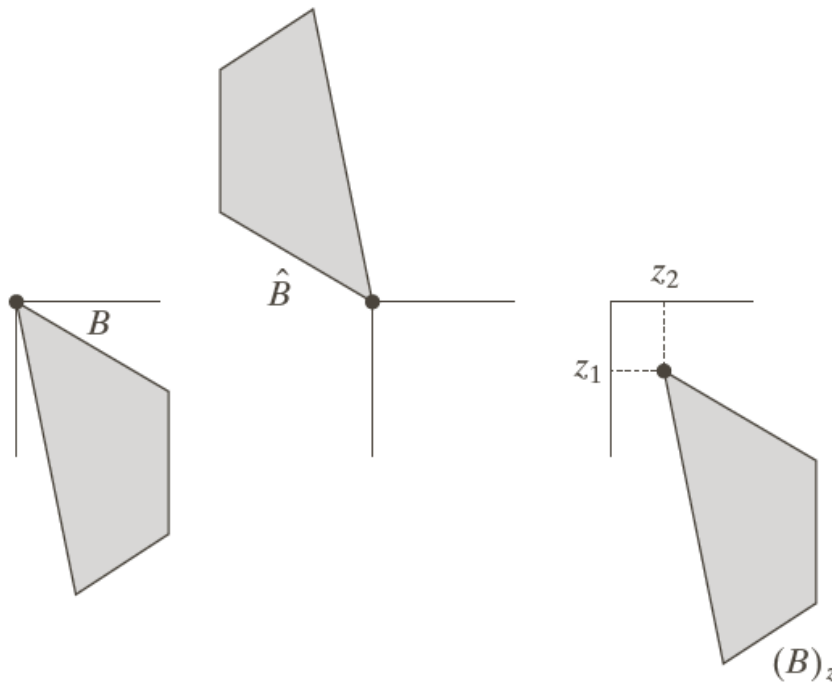
$$(A)_z = \{c \mid c = a + x, \text{ para } a \in A\}$$

# Definições Básicas



- Reflexão de  $B$

$$\hat{B} = \{x \mid x = -b, \text{ para } b \in B\}$$



a b c

**FIGURE 9.1**

(a) A set, (b) its reflection, and (c) its translation by  $z$ .

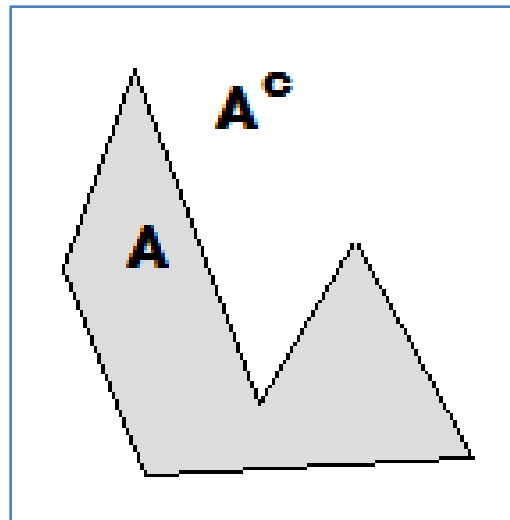


# Definições Básicas



- **Complemento** do conjunto  $A$ :

$$A^c = \{x \mid x \notin A\}$$



# Definições Básicas



- **Interseção** de dois conjuntos  $A$  e  $B$  é o conjunto de pixels pertencentes a ambos  $A$  e  $B$ :  
$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$
- **União** de dois conjuntos  $A$  e  $B$  é o conjunto de pixels que pertencem ou  $A$  ou  $B$  ou ambos:

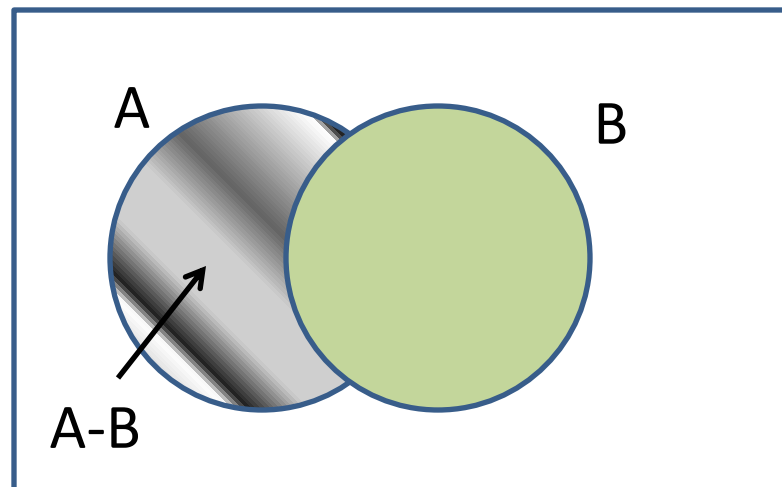
$$A \cup B = \{x \mid (x \in A) \vee (x \in B)\}$$

# Definições Básicas

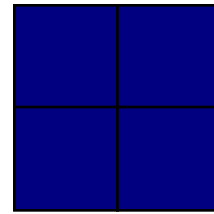
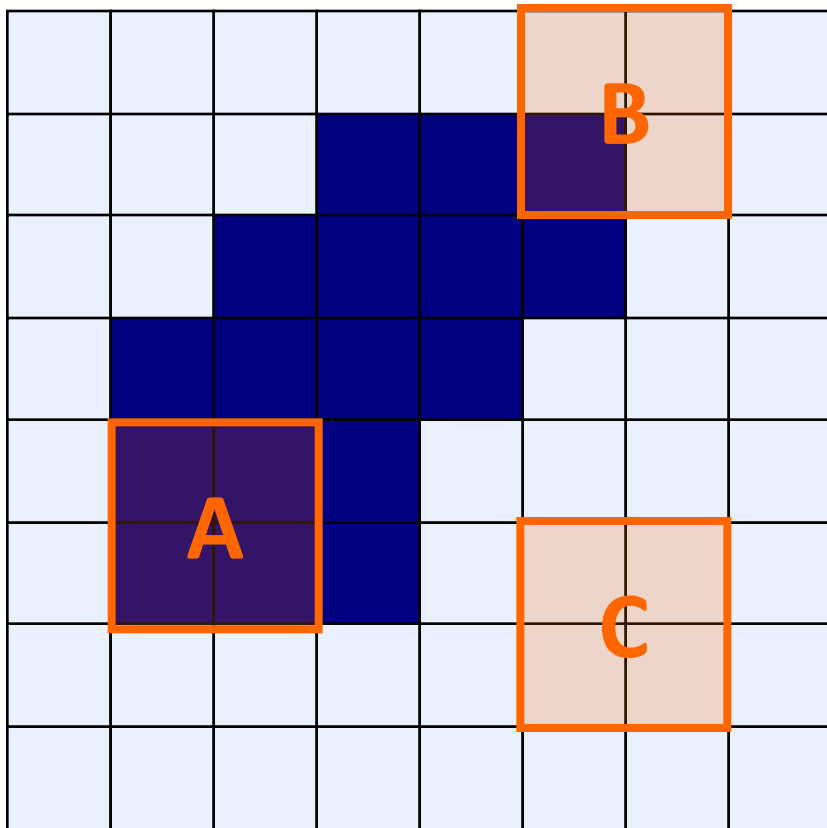


- **Diferença** de dois conjuntos  $A - B$ :

$$A - B = \{x \mid (x \in A) \wedge (x \notin B)\}$$



# Elementos estruturantes, *Hit* e *Fit*



Elemento estruturante

***Fit***: Todos os pixels 1 no elemento estruturante cobrem uma área na imagem também com valores 1 (área segmentada)

***Hit***: Qualquer pixel 1 do elemento estruturante cobre um elemento 1 da imagem

Todas as operações morfológicas estão baseadas nesse dois simples conceitos.

# Elementos Estruturantes



- Elementos estruturantes podem ser de qualquer tamanho e de qualquer forma
- Entretanto, por simplicidade, serão usados elementos estruturantes retangulares com suas origens no pixel central

1	1	1
1	<b>1</b>	1
1	1	1

0	1	0
1	<b>1</b>	1
0	1	0

0	0	1	0	0
0	1	1	1	0
1	1	<b>1</b>	1	1
0	1	1	1	0
0	0	1	0	0

# Fit & Hit

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	<b>B</b>	1	1	1	0	<b>C</b>	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	<b>A</b>	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

Elemento  
estruturante 1

0	1	0
1	1	1
0	1	0

Elemento  
estruturante 2

# Operações Básicas



- Fundamentalmente técnicas de processamento morfológico de imagens são como técnicas de filtragem espacial
- O elemento estruturante é movimentado por cada pixel da imagem original para gerar um pixel na nova imagem processada

# Operações Básicas



- O valor do novo pixel dependerá da operação que está sendo realizada
- Há duas operações morfológicas básicas: erosão e dilatação
- Outras operações elaboradas a partir delas



# Dilatação



- $A$  e  $B$  conjuntos de  $Z^2$  (imagens binárias)
- A **Dilatação** de  $A$  por  $B$  é definida como

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\}$$

- Reflexão de  $B$  em torno de sua origem
- Translação dessa reflexão por  $x$

# Dilatação



- **A dilatação por B:** conjunto de todos os deslocamentos de  $x$  tais que  $B$  refletido e  $A$  se sobreponham em pelo menos um elemento não nulo

# Dilatação



	1	2	3	4	5
1					
2		●	●		
3		●	●		
4		●	●		
5		●	●	●	
6			●	●	
7					

*A*

	-1	0	1
-1	●		●
0		●	
1	●		●

*B*

# Dilatação



	1	2	3	4	5
1					
2					
3	●	●			
4	●	●			
5	●	●			
6	●	●	●		
7		●	●		

$A_{(1,-1)}$

	1	2	3	4	5
1	●	●			
2	●	●			
3	●	●			
4	●	●	●		
5		●	●		
6					
7					

$A_{(-1,-1)}$

# Dilatação



	1	2	3	4	5
1			●	●	
2		■	●	●	
3		■	●	●	
4		■	●	●	●
5		■	■	●	●
6			■	■	
7					

$A_{(-1,1)}$

	1	2	3	4	5
1					
2		■	■		
3		■	●	●	
4		■	●	●	
5		■	●	●	
6			●	●	●
7				●	●

$A_{(1,1)}$

# Dilatação



- O elemento estruturante  $s$  é posicionado com sua origem em  $(x, y)$  e o novo valor de pixel é determinado através da regra:

$$g(x, y) = \begin{cases} 1 & \text{se } s \text{ hits } f \\ 0 & \text{caso contrário} \end{cases}$$

# Dilatação

Imagem Original

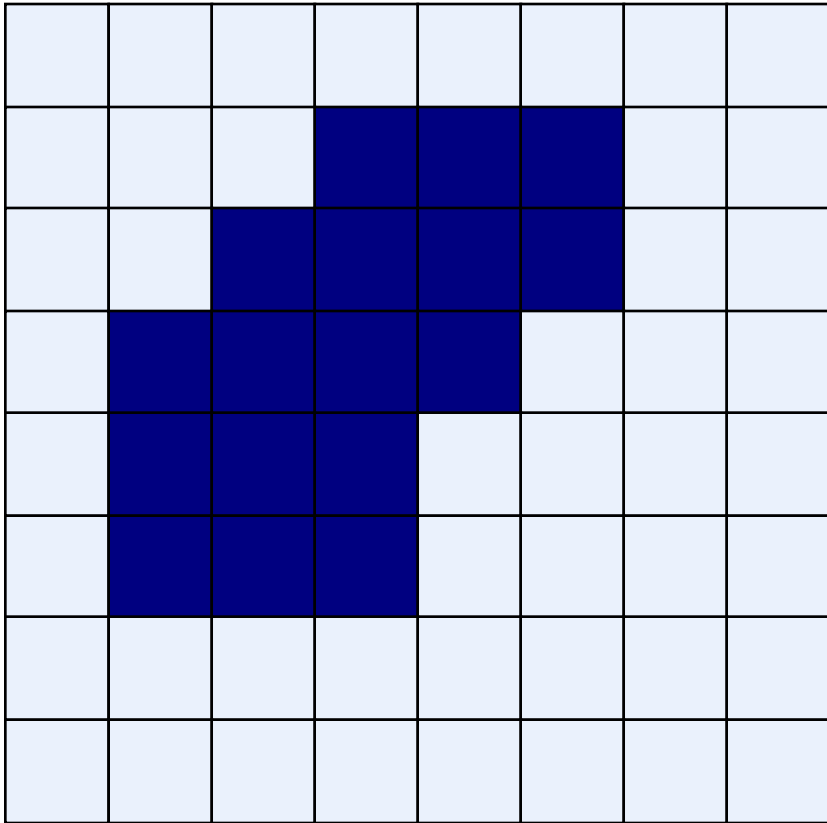
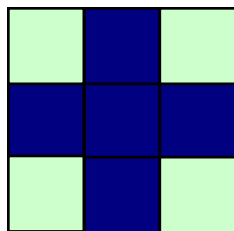
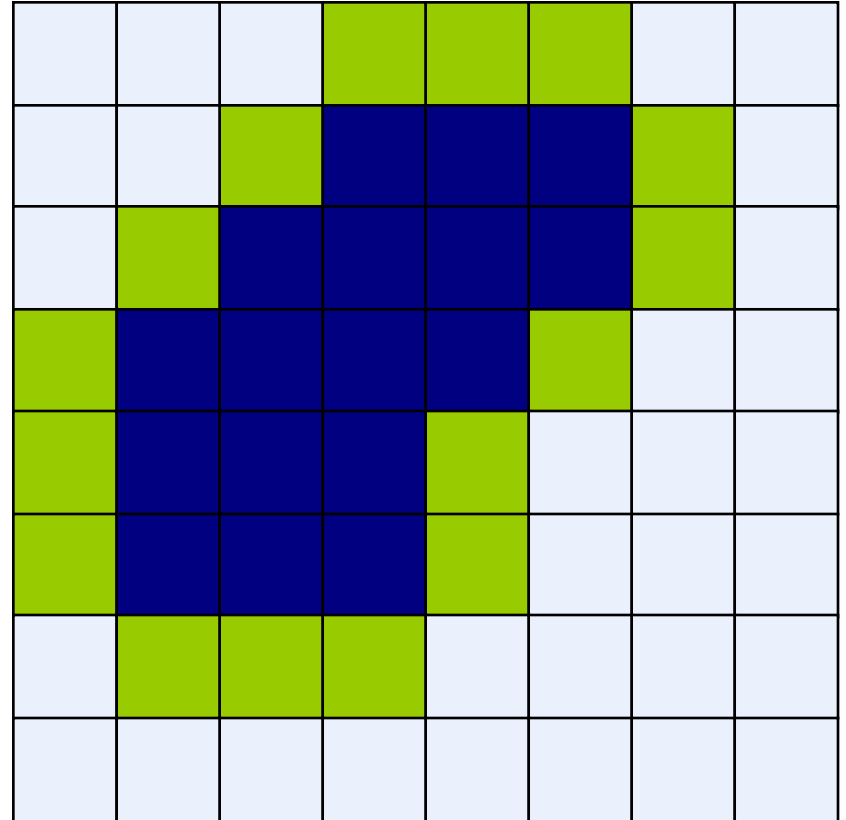


Imagem processada com pixels dilatados



Elemento Estruturante

# Dilatação



Imagem Original



Dilatação por um  
elemento  
estruturante  
quadrado 3\*3



Dilatação por um  
elemento estruturante  
quadrado 5\*5

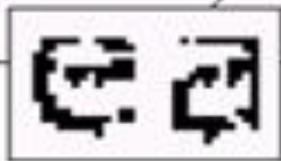
**Observação:** Nos exemplos a seguir o valor 1 representa um pixel preto!



# Dilatação – Exemplo Aplicação 2

Imagem Original

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Após dilatação

**Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.**



Elementos estruturante

0	1	0
1	1	1
0	1	0

# Onde usar métodos de Dilatação?



- Dilatação pode reparar quebras



- Dilatação pode reparar intrusões



Cuidado: Dilatação aumenta objetos

# Exemplo de aplicação de Dilatação



```
A = imread('text.png');  
se = strel('diamond', 1);  
A2 = imdilate(A, se);  
imshow(A2);
```

# Erosão



- $A$  e  $B$  conjuntos de  $Z^2$  (imagens binárias)
- A **Erosão** de  $A$  por  $B$  é definida como

$$A \ominus B = \{x \mid (B)_x \subseteq A\}$$

O conjunto resultante da erosão de  $A$  por  $B$  é o conjunto de todos os pontos  $x$  tais que  $B$ , quando transladado por  $x$ , fique contido em  $A$

# Erosão



	1	2	3	4	5	6
1		●				
2	●	●	●	●	●	●
3			●	●	●	●
4			●	●		●
5		●	●	●	●	●
6			●	●		

*A*

	-1	0	1
-1		●	
0	●	●	●
1		●	

*B*

# Erosão



	1	2	3	4	5	6
1		●				
2	●	●	●	●	●	●
3			●	●	●	●
4			●	●		●
5		●	●	●	●	●
6			●	●		

	1	2	3	4	5	6
1		●				
2	●	●	●	●	●	●
3			●	●	●	●
4			●	●		●
5		●	●	●	●	●
6			●	●		

	1	2	3	4	5	6
1		●				
2	●	●	●	●	●	●
3			●	●	●	●
4			●	●		●
5		●	●	●	●	●
6			●	●		

# Erosão



	1	2	3	4	5	6
1						
2						
3				●		
4						
5						
6						

	1	2	3	4	5	6
1						
2						
3						
4						
5			●			
6						

	1	2	3	4	5	6
1						
2						
3						
4						
5				●		
6						

# Erosão



	1	2	3	4	5	6
1						
2						
3				●		
4						
5			●	●		
6						

$A \ominus B$



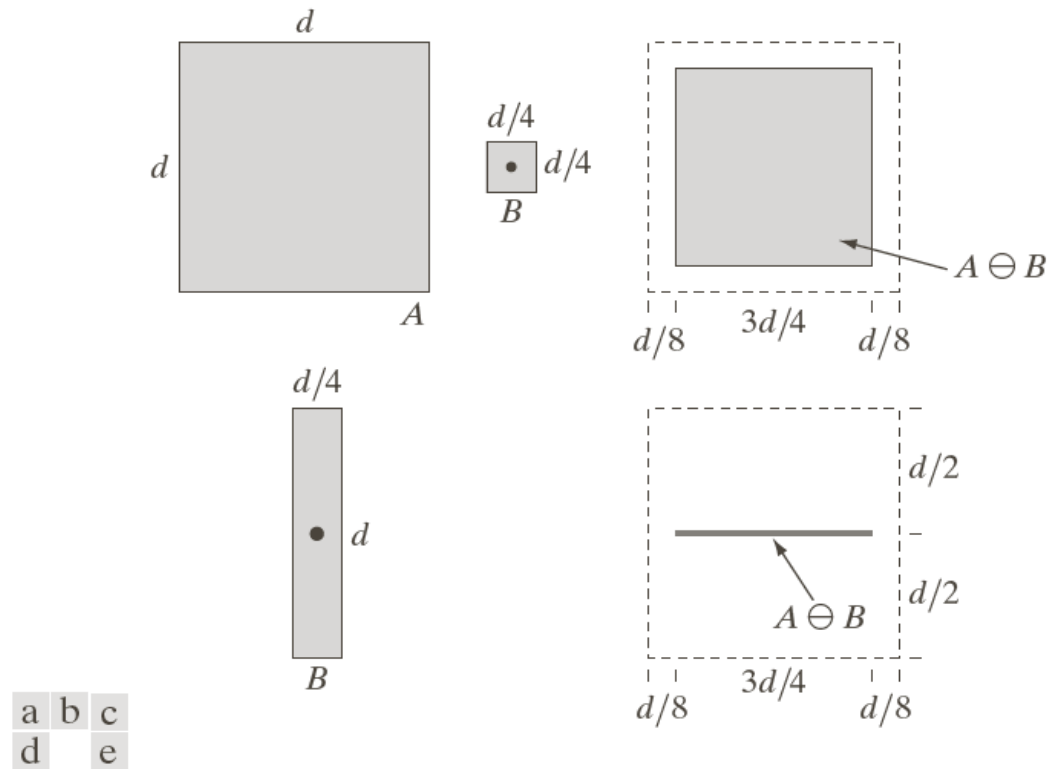
# Erosão



- O elemento estruturante  $s$  é posicionado com sua origem em  $(x, y)$  e o valor do pixel na nova imagem é determinado usando-se a seguinte regra:

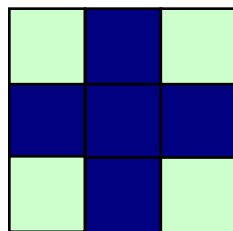
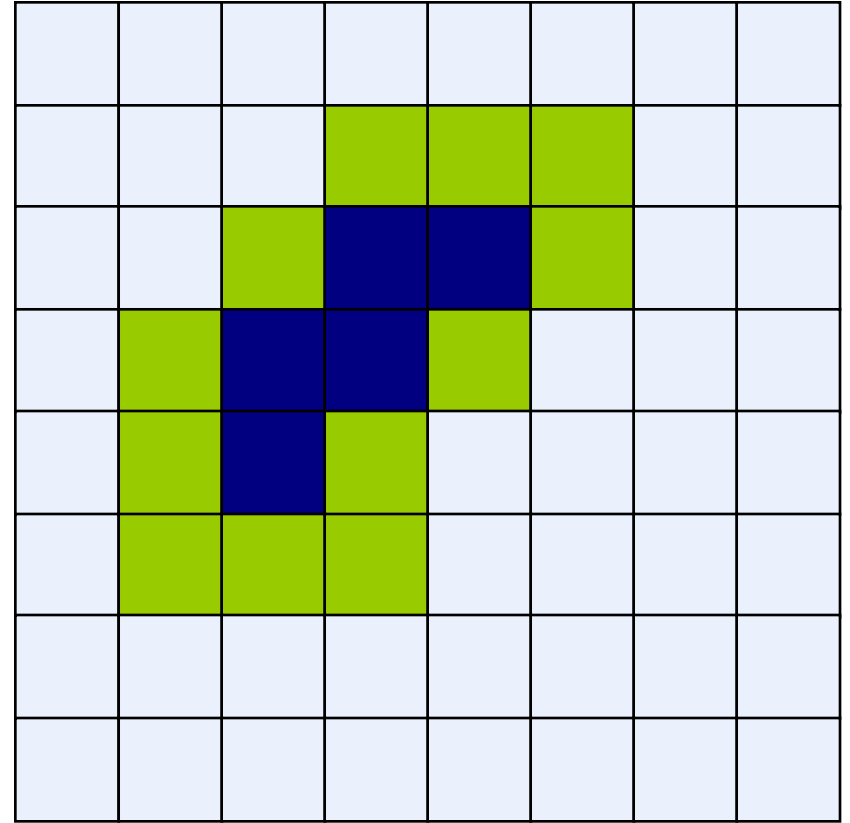
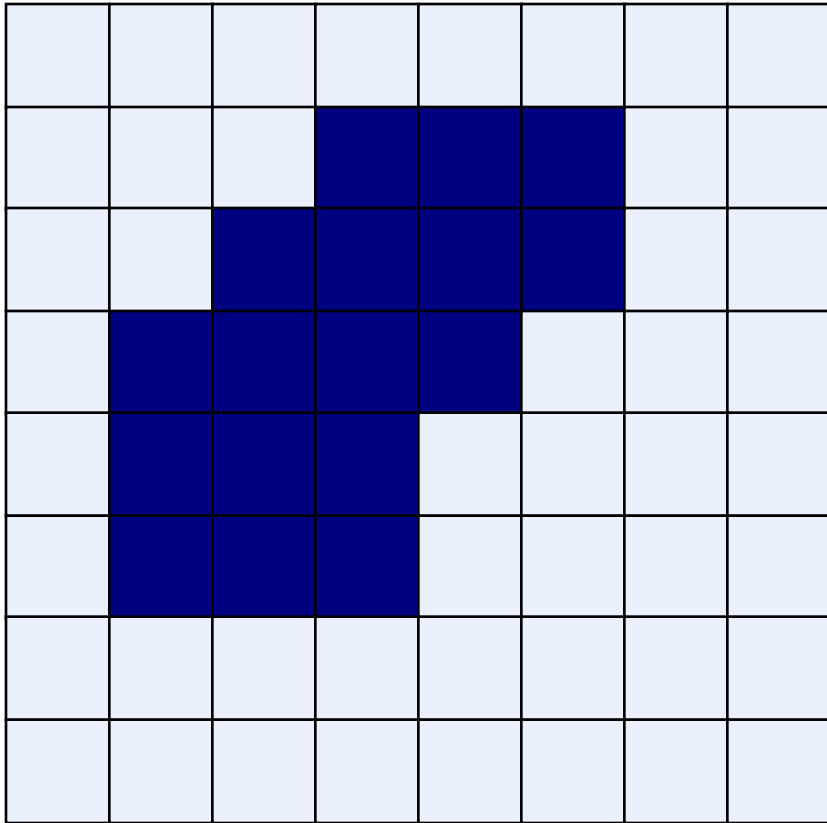
$$g(x, y) = \begin{cases} 1 & \text{se } s \text{ fits } f \\ 0 & \text{caso contrário} \end{cases}$$

# Erosão



**FIGURE 9.4** (a) Set  $A$ . (b) Square structuring element,  $B$ . (c) Erosion of  $A$  by  $B$ , shown shaded. (d) Elongated structuring element. (e) Erosion of  $A$  by  $B$  using this element. The dotted border in (c) and (e) is the boundary of set  $A$ , shown only for reference.

# Exemplo de Erosão



Elemento Estruturante

# Exemplo 2 de Erosão



Imagem Original



Erosão por um elemento estruturante 3\*3



Erosão por um elemento estruturante 5\*5

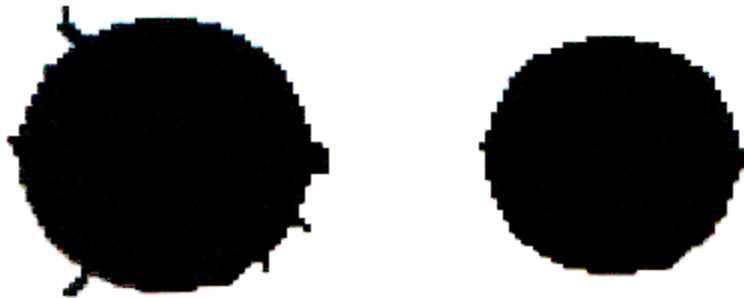
**Observação:** Nos exemplo a seguir o valor 1 representa um pixel preto!

# Onde usar métodos de Erosão

Erosão pode dividir objetos



Erosão pode remover extrusões

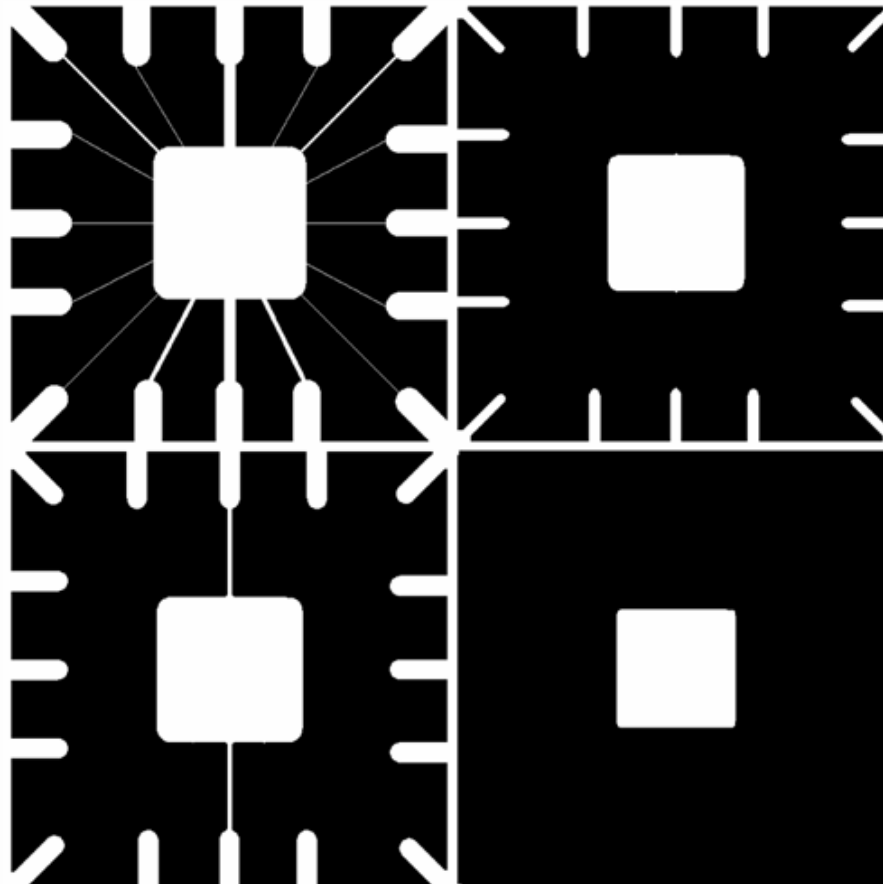


**Cuidado:** Erosão também reduz objetos

# Exemplo de aplicação de Erosão



Imagem Original



Após erosão com um disco de raio 10

Após erosão com um disco de raio 5

Após erosão com um disco de raio 20

# Exemplo de aplicação de Erosão



```
A = imread('wirebond.png');
```

```
se = strel('disk', 10);
```

```
A2 = imerode(A, se);
```

```
imshow(A2);
```

# Considerações



- A **Dilatação** expande uma imagem e a **Erosão** reduz
- A **Erosão** não é o inverso da **Dilatação**
- **Erosão** e **Dilatação** são operações duais:

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

- O complementos de uma **Erosão** é o mesmo que uma **Dilatação** do complemento da imagem pelo elemento estruturante refletido



# Combinando Operações



- Operações morfológicas mais interessantes podem ser obtidas combinando operações de erosão e dilatação
- As operações compostas mais utilizadas são as seguintes:
  - Abertura
  - Fecho

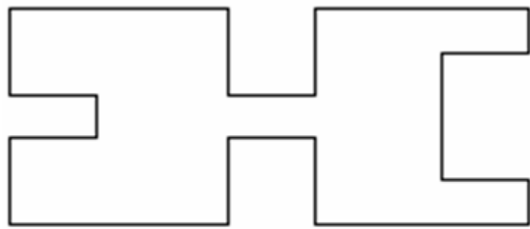
# Abertura



- A Abertura é uma operação que geralmente **suaviza o contorno** de uma imagem, **quebra istmos e elimina protuberâncias finas**
- A Abertura de uma imagem  $f$  por um elemento estruturante  $s$ , definida por  $f \circ s$  é simplesmente uma erosão seguida por uma dilatação:

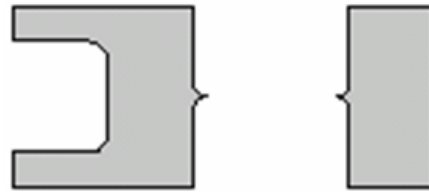
$$f \circ s = (f \ominus s) \oplus s$$

# Abertura



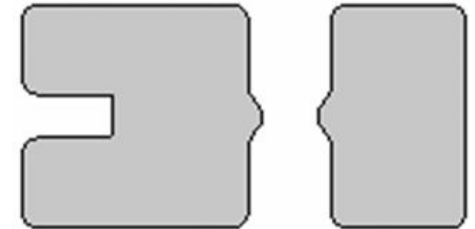
$A$

Forma original



$A \ominus B$

Após erosão



$A \circ B = (A \ominus B) \oplus B$

Após dilatação  
(abertura)

Neste exemplo um elemento estruturante em forma de disco foi utilizado

# Exemplo de Abertura

Imagem  
Original



Imagem  
após  
abertura



# Exemplo de Abertura

Imagem Original

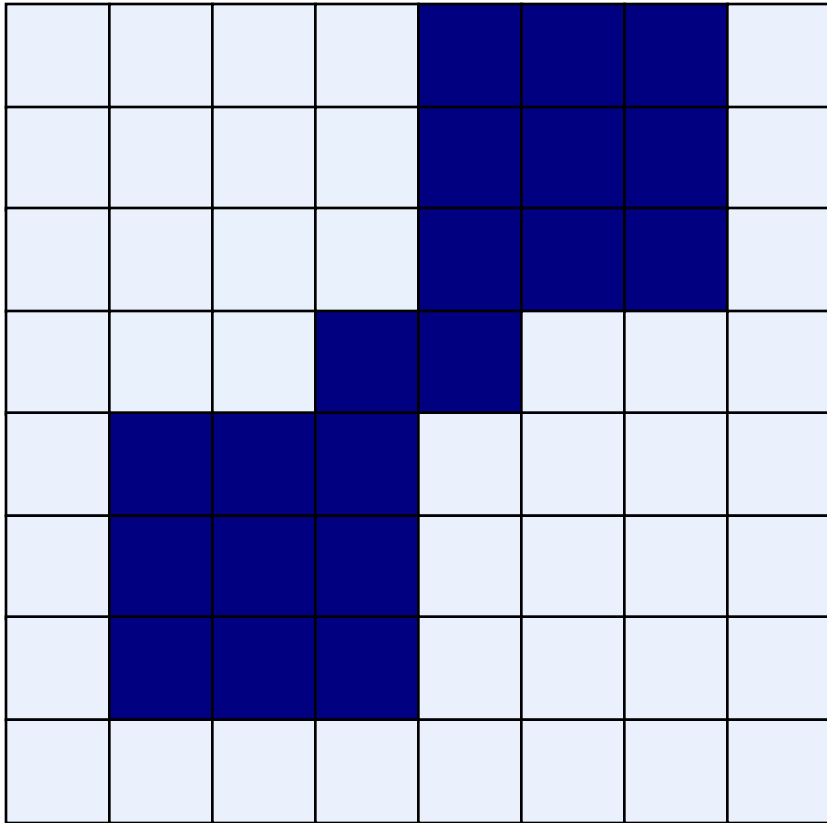
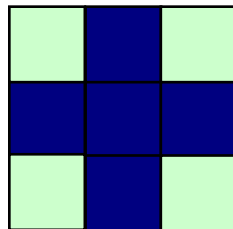
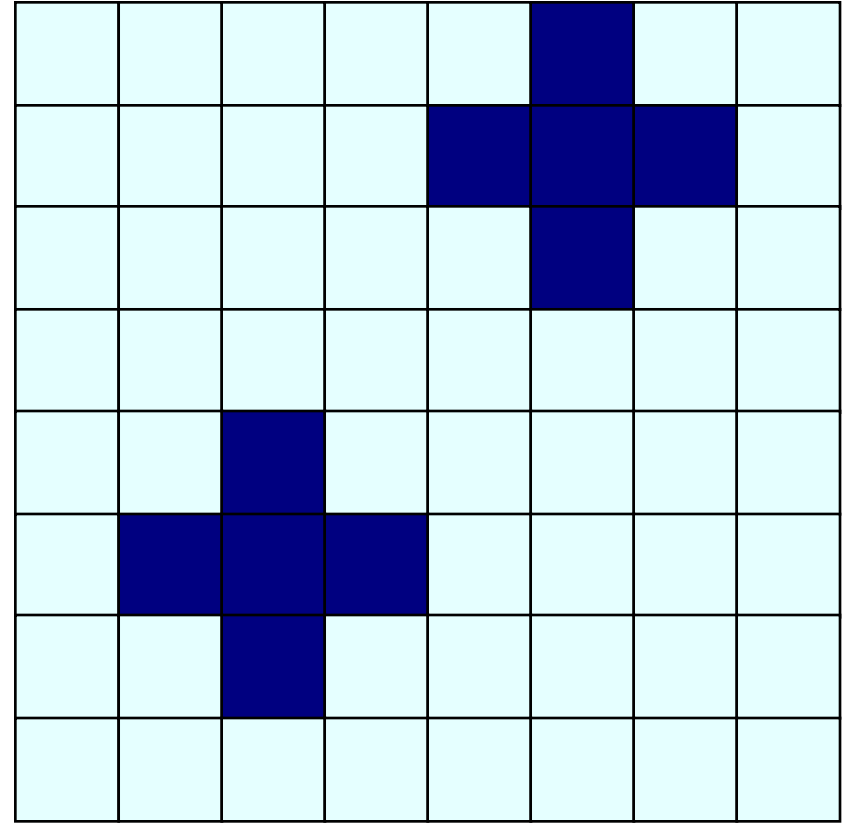


Imagem Processada



Elementos estruturantes

# Considerações



- A abertura tende a abrir pequenos vazios ou espaços entre objetos próximos
- Também é usada para remover ruídos da imagem
- Pontos pretos aleatórios e isolados podem ser removidos e a forma dos objetos é recuperada pela dilatação sem restaurar o ruído.

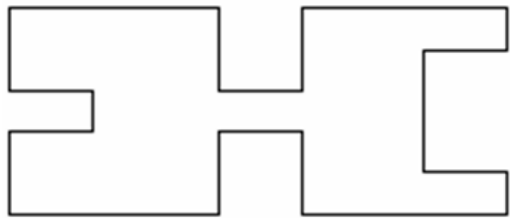
# Fecho



- O fecho também tende a suavizar os contornos, mas geralmente **funde partes**. Elimina pequenos buracos e preenche fendas em um contorno
- O fecho de uma imagem  $f$  pelo elemento estruturante  $s$  definido como  $f \bullet s$  é simplesmente uma operação de dilatação seguida de uma operação de erosão

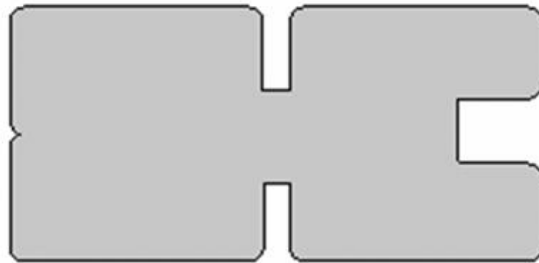
$$f \bullet s = (f \oplus s) \ominus s$$

# Fecho



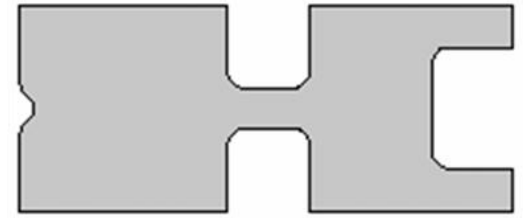
$A$

Forma Original



$A \oplus B$

Após dilatação



$A \cdot B = (A \oplus B) \ominus B$

Após Erosão  
(Fecho)

Neste exemplo um elemento estruturante em forma de disco foi utilizado



# Exemplo de Fecho

Imagem Original



Imagem após  
operação de  
fecho



# Exemplo de Fecho

Imagem Original

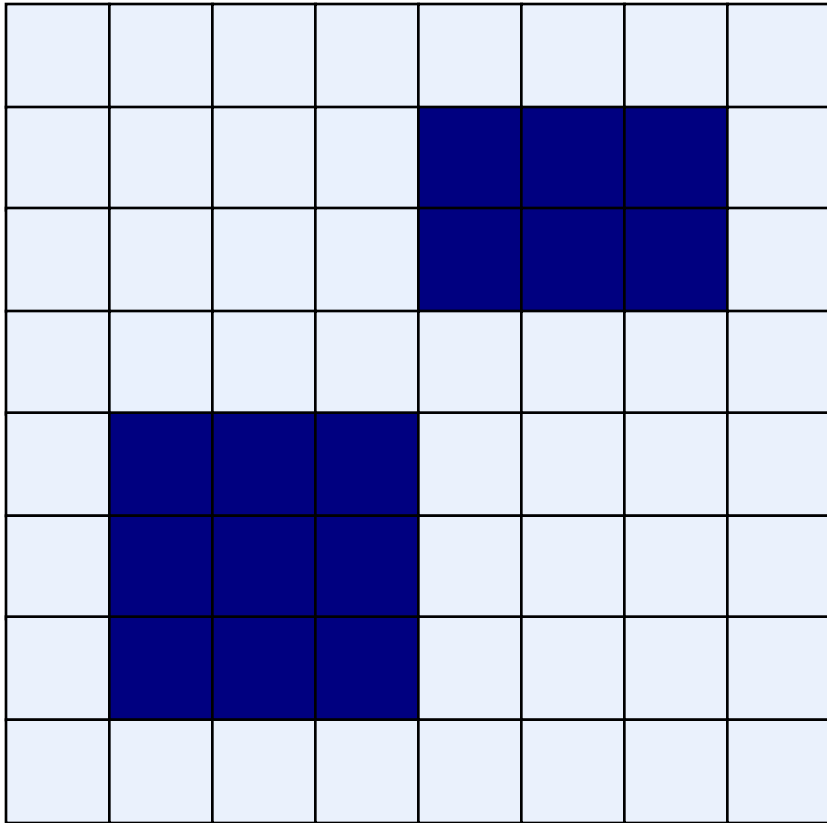
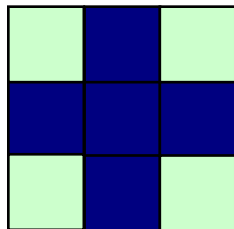
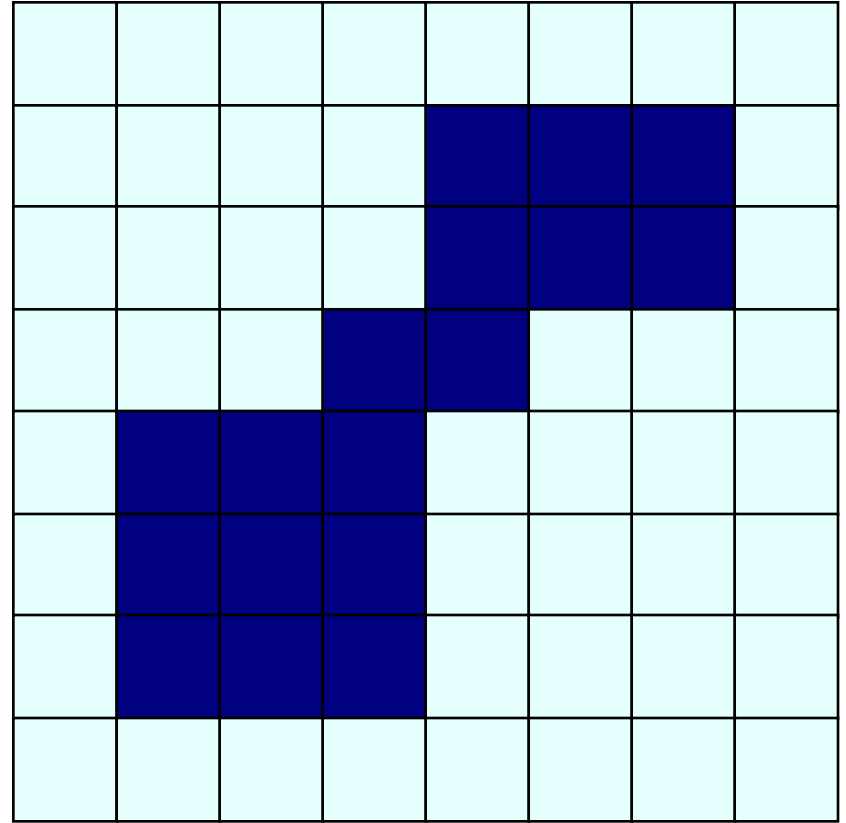


Imagem Processada



Elemento Estruturante

# Considerações



- O fechamento irá preencher ou fechar os vazios
- Pode remover muitos dos pixels brancos de ruído
- A abertura e o fechamento são operações duais relativamente à complementação e reflexão de conjuntos

$$(A \bullet B)^c = (A^c \circ \hat{B})$$

# Processamento Morfológico-Exemplo



# Algoritmos Morfológicos



- Fazendo uso das técnicas vistas até agora, podemos começar a considerar alguns algoritmos de processamento morfológico mais interessantes
- Serão vistos algoritmos para:
  - Extração de Borda
  - Preenchimento de Região

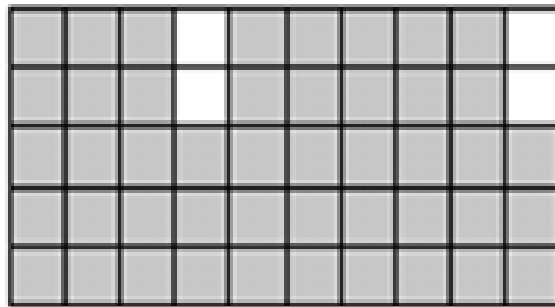
# Extracção de bordas



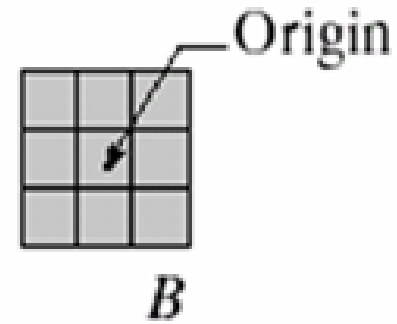
- Extração de borda de um objeto é uma operação extremamente útil
- Uma borda pode ser definida como

$$\beta(A) = A - (A \ominus B)$$

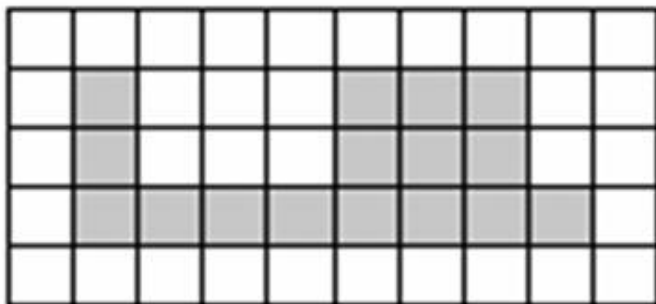
# Exemplo



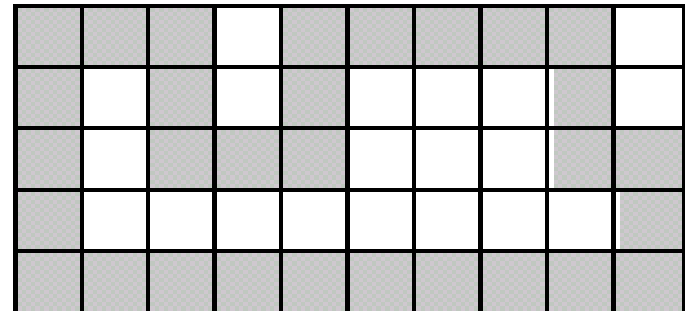
*A*



*B*



$A \ominus B$



$B(A)$

# Exemplo de extracção de bordas



- Uma imagem simples segmentada e o resultado da realização da extração de borda usando um elemento estruturante  $3 \times 3$



Imagem Original

Bordas Extraídas

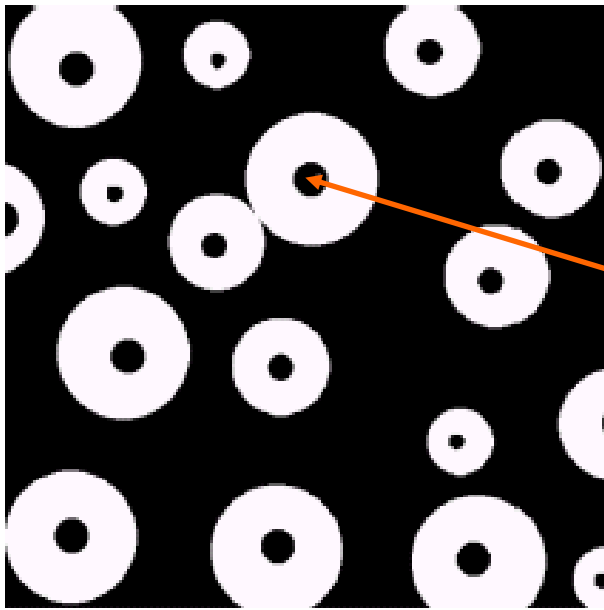


# Preenchimento de Região



- A partir de um pixel dentro da uma região definida por uma borda, a técnica de “preenchimento de região” procura preencher com pixels ligados até o limite da borda

# Preenchimento de Região



A partir de uma ponto  
podemos preencher o  
círculo?

# Preenchimento de Região



- A equação chave para essa operação é
$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$
- Onde  $X_0$  é simplesmente um ponto dentro da fronteira,  $B$  é o elemento estruturante e  $A^c$  é o complemento de  $A$
- Essa equação é aplicada repetidamente até que  $X_k$  seja igual a  $X_{k-1}$
- Por fim, o resultado é unido com a fronteira original

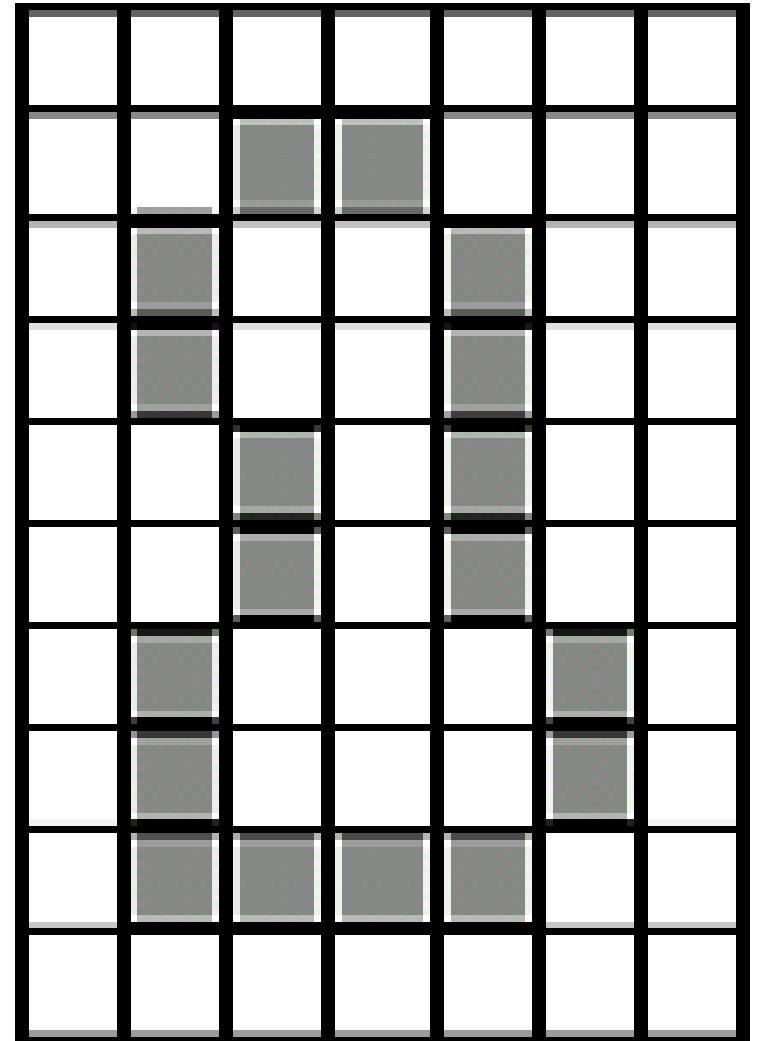
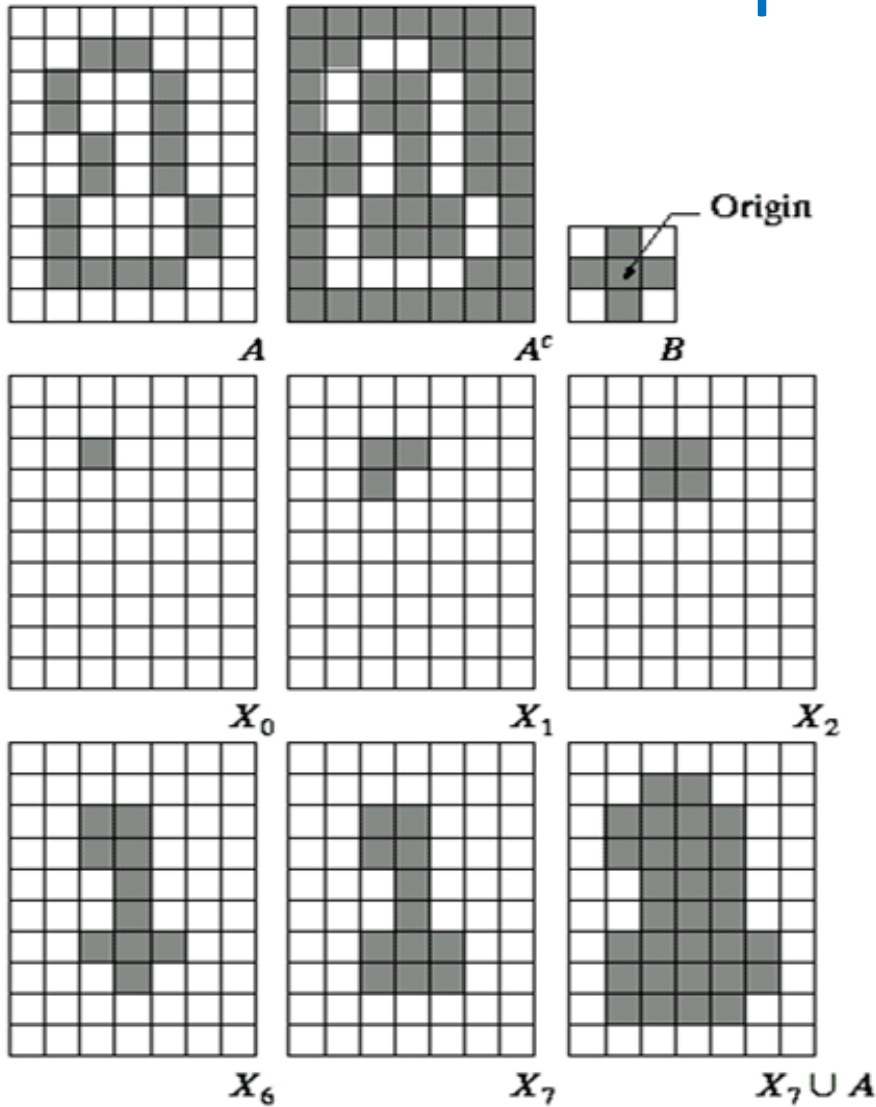
# Preenchimento de Região



Set Operation	MATLAB Expression for Binary Images	Name
$A \cap B$	$A \& B$	AND
$A \cup B$	$A   B$	OR
$A^c$	$\sim A$	NOT
$A - B$	$A \& \sim B$	DIFFERENCE

**TABLE 9.1**  
Using logical expressions in MATLAB to perform set operations on binary images.

# Preenchimento de Região: Passo a passo



# Exemplo de Preenchimento de Região

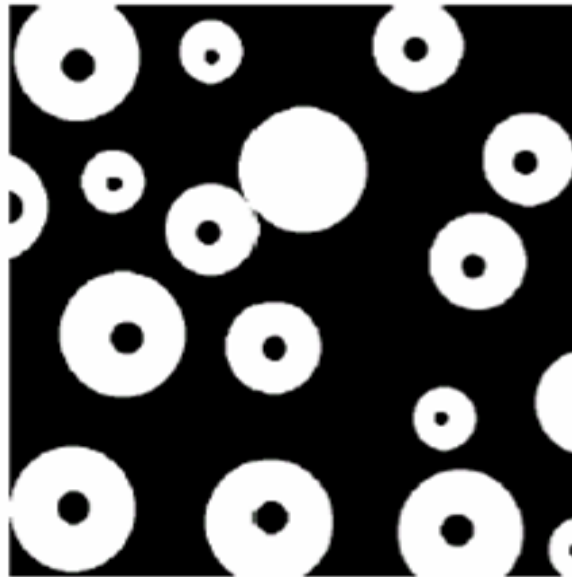


Imagem Original

Uma região  
preenchida

Todas as regiões  
preenchidas

# Detecção de Bordas



- Bordas internas

$$A - (A \ominus B)$$

- Bordas externas

$$(A \oplus B) - A$$

# Resumo



- O **propósito** do processamento morfológico é primordialmente **remover imperfeições adicionadas** durante a **segmentação**
- As operações básicas são **erosão** e **dilação**
- Através das operações básicas pode-se realizar operações de **abertura** e **fecho**
- Operações morfológicas mais avançadas podem ser realizadas através da combinação das operações acima