

Morfologia Matemática em Imagens

Prof. Luiz Otavio Murta Jr.

Departamento de Computação e Matemática
(FFCLRP/USP)

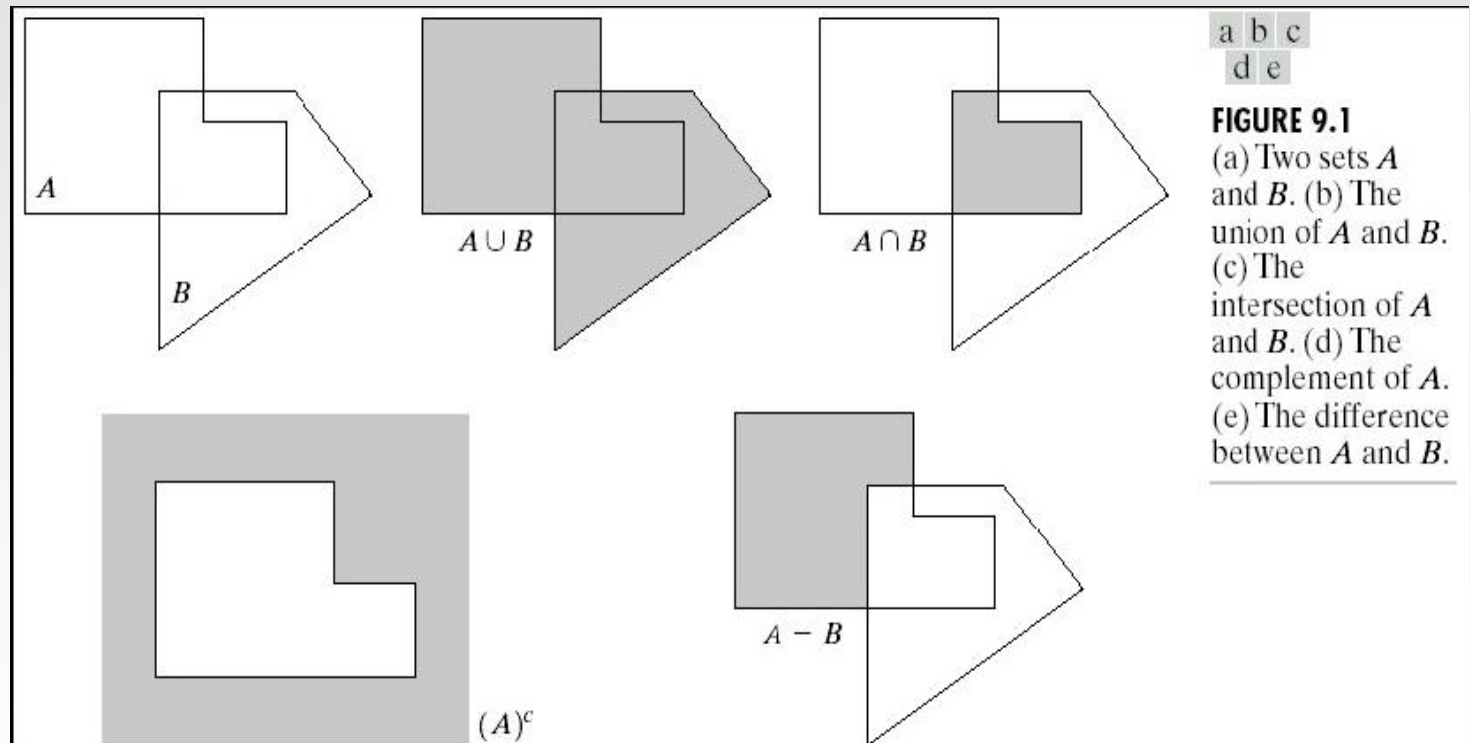
Principais Tópicos

- Introdução
- Morfologia Binária em Imagens
- Morfologia de Tons de Cinza
- Segmentação Morfológica
- Bibliografia

Operadores Morfológicos Binários

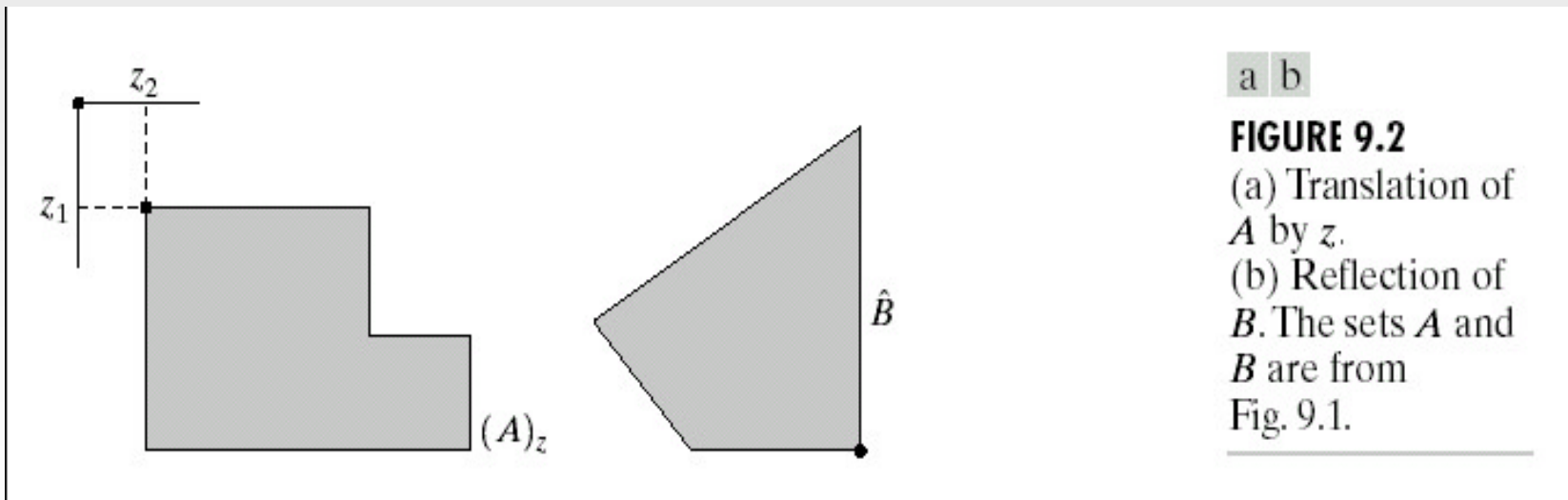
Algoritmos: Operadores morfológicos

Teoria Básica de Conjuntos



Algoritmos: Operadores morfológicos

Reflexão e $\hat{B} = \{w \mid w \in -b, \text{ para } b \in B\}$
Translação $(A)_z = \{c \mid c \in a + z, \text{ para } a \in A\}$



Algoritmos: Operadores morfológicos

Operações Lógicas

| p | q | p “e” q (p.q) | p “ou” q (p+q) | “não” p |
|----------|----------|----------------------|-----------------------|----------------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

Algoritmos: Operadores morfológicos

Operações Lógicas:

- Não
- E
- Ou
- X ou
- X e

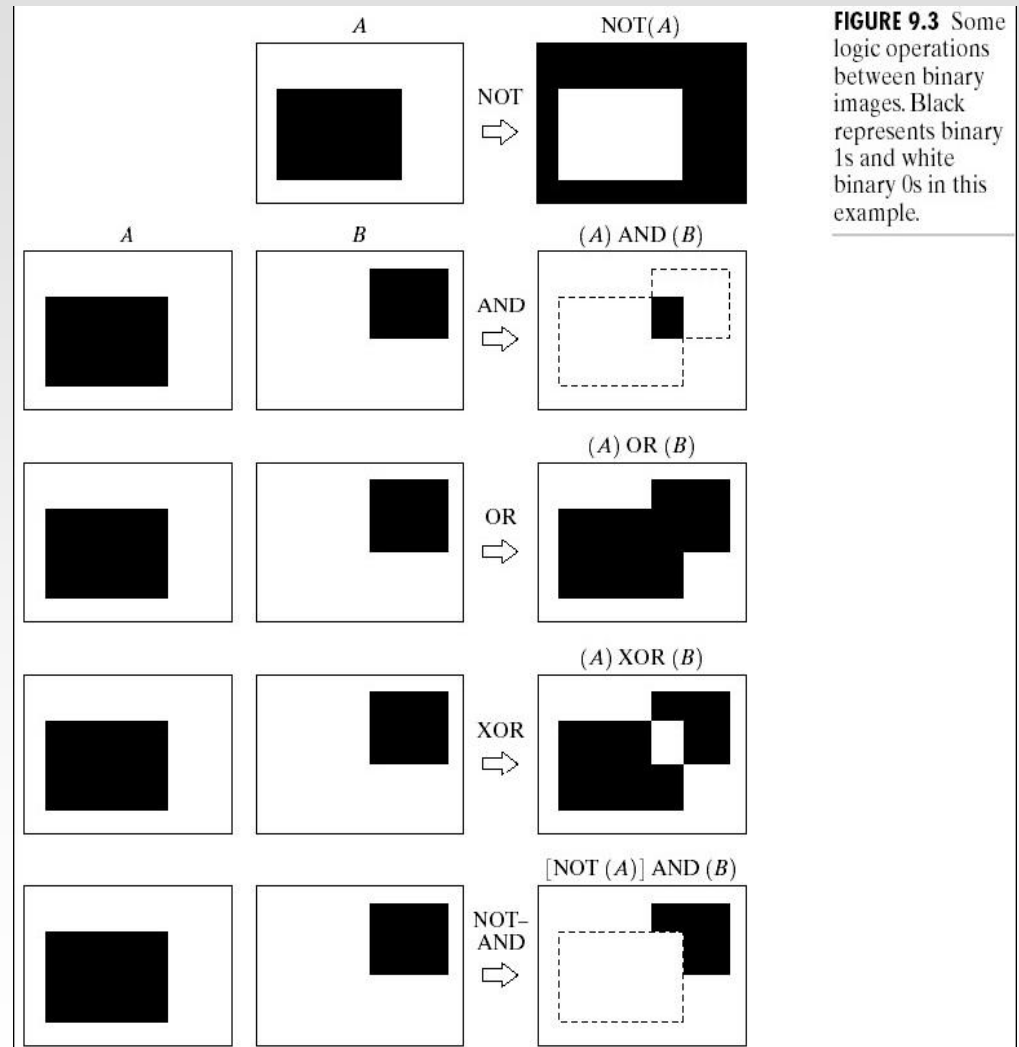


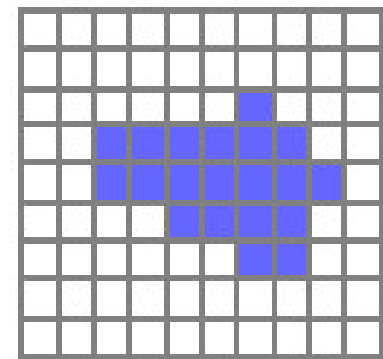
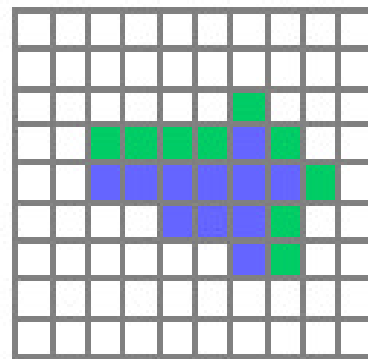
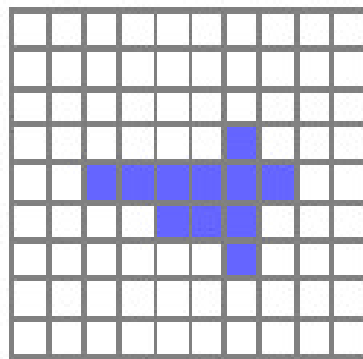
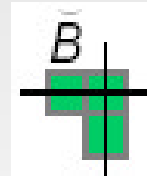
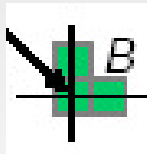
FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Algoritmos: Operadores morfológicos

Dilatação

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

origem



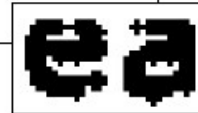
Algoritmos: Operadores morfológicos

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.



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.



| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

a c
b

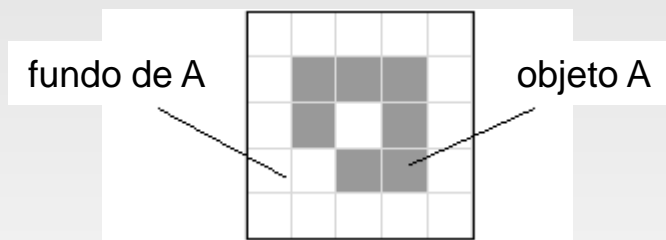
FIGURE 9.5

(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Algoritmos: Operadores morfológicos

- **Enquadramento**

- A noção de objeto implica conectividade de pixels



- O objeto A tem conectividade 4
- O fundo tem conectividade $12-4(=8)$

- **Definições**

- As operações fundamentais sobre objetos são:

- » Translação: $A + x = \{ a + x \mid a \in A \}$

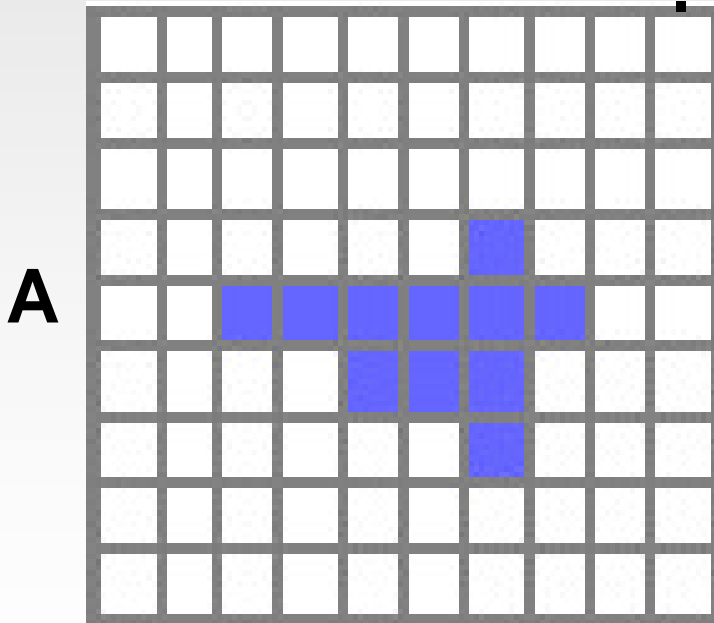
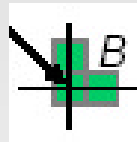
- » Adição/subtração: $A \oplus B = \cup_{b \in B} (A+b)$; $A \ominus B = \cap_{b \in B} (A+b)$

- » Complemento (fundo)

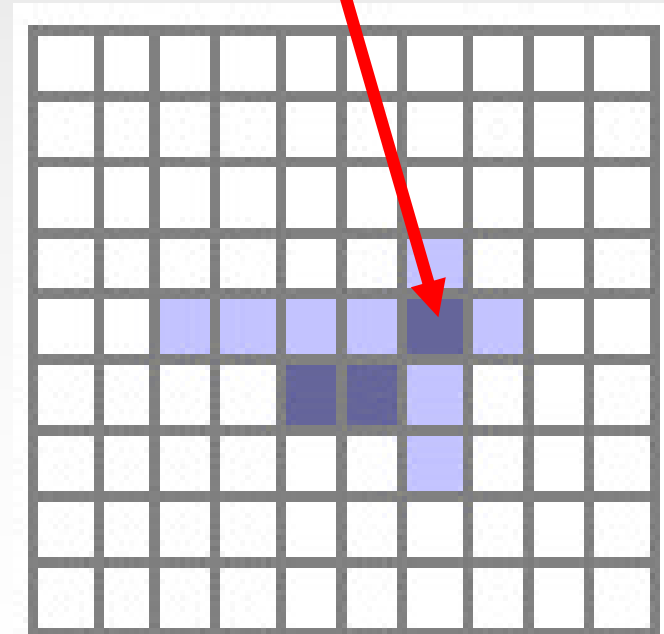
- » Simetria: $-A = \{ -a \mid a \in A \}$

Algoritmos: Operadores morfológicos

Erosão $A \ominus B = \{z \mid (B)_z \subseteq A\}$
origem



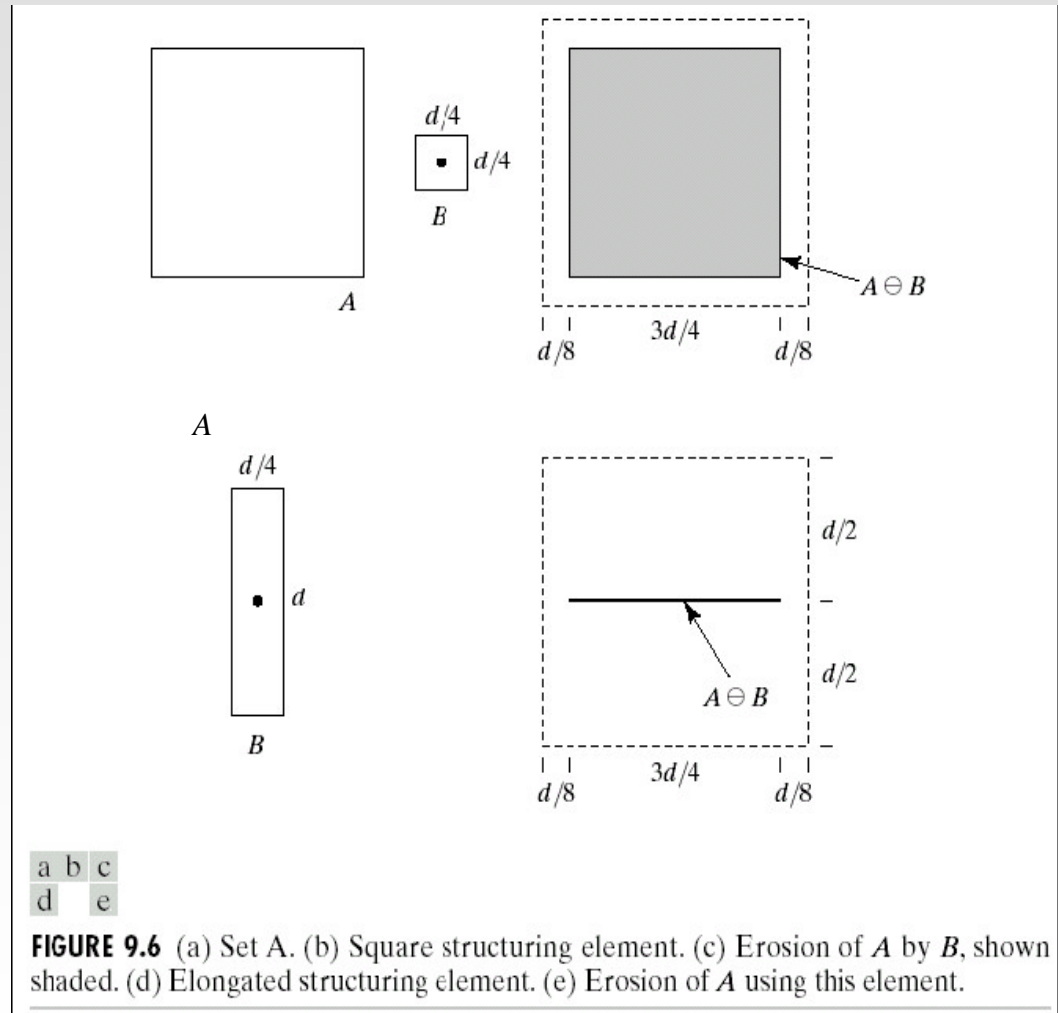
Resultado da Erosão: apenas os quadrados destacados



Algoritmos: Operadores morfológicos

Erosão: $A \ominus B = \{z \mid (B)_z \subseteq A\}$

- Tamanho dos objetos são reduzidos
- Objetos menores do que o el. est. são eliminados
- Número de componentes pode aumentar

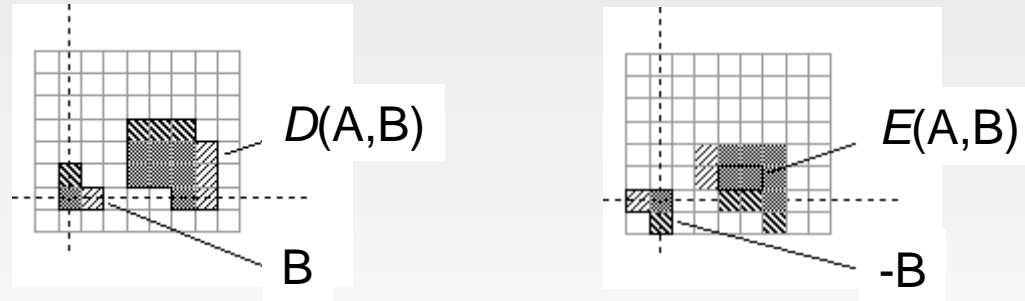


Algoritmos: Operadores morfológicos

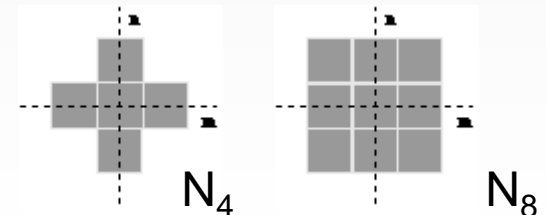
- **Dilatação e Erosão**

- A dilatação $D(A,B)$ corresponde à adição $A \oplus B$
- A erosão $E(A,B)$ corresponde à subtração $A \ominus (-B)$

- Exemplos



- Tipicamente A é uma imagem e B é um “estruturante” (equivale ao núcleo de convolução dos filtros lineares)
- Exemplos de estruturantes comuns



Algoritmos: Operadores morfológicos

- **Dilatação e Erosão**

- Teorema da Decomposição

- » Para um estruturante B finito, simétrico, sem buracos e contendo o seu centro ($[0,0] \in B$), verifica-se que

- $$D(A,B) = A \cup (\partial A \oplus B)$$
 [nota: ∂A é o contorno de A]

- » Resulta que apenas basta processar os pixels do contorno de A, não sendo necessário processar os pixels do seu interior

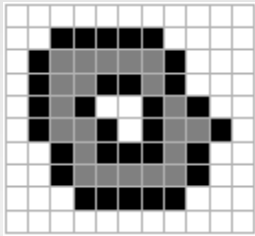
- » Algoritmo “rápido” de dilatação: ***para cada pixel do objeto, todos os seus pixels vizinhos (viz_C) de fundo passam a pertencer ao objeto***

- » Algoritmo “rápido” de erosão: ***cada pixel do objeto que possua um pixel vizinho (viz_C) de fundo passa a fundo***

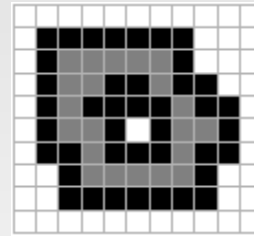
Algoritmos: Operadores morfológicos

- **Dilatação e Erosão**

- Exemplos “rápidos”



Dilatação com
estrututante N_4



Dilatação com
estrututante N_8

- Importante: $D(E(A,B),B) \neq A \neq E(D(A,B),B)$

- **Abertura e Fechamento**

- Abertura: $O(A,B) = D(E(A,B),B)$

- » Tende a suavizar o contorno pelo interior

- Fechamento: $C(A,B) = E(D(A,-B),-B)$

- » Tende a suavizar o contorno pelo exterior

Algoritmos: Operadores morfológicos

Erosão $A \ominus B = \{z \mid (B)_z \subseteq A\}$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$A \ominus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Algoritmos: Operadores morfológicos

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

- Começando com a definição de erosão:

$$\begin{aligned}(A \ominus B)^c &= \{z \mid (B)_z \subseteq A\}^c = \{z \mid (B)_z \cap A^c = \emptyset\}^c \\ &= \{z \mid (B)_z \cap A^c \neq \emptyset\}\end{aligned}$$

- Sabendo a definição de dilatação

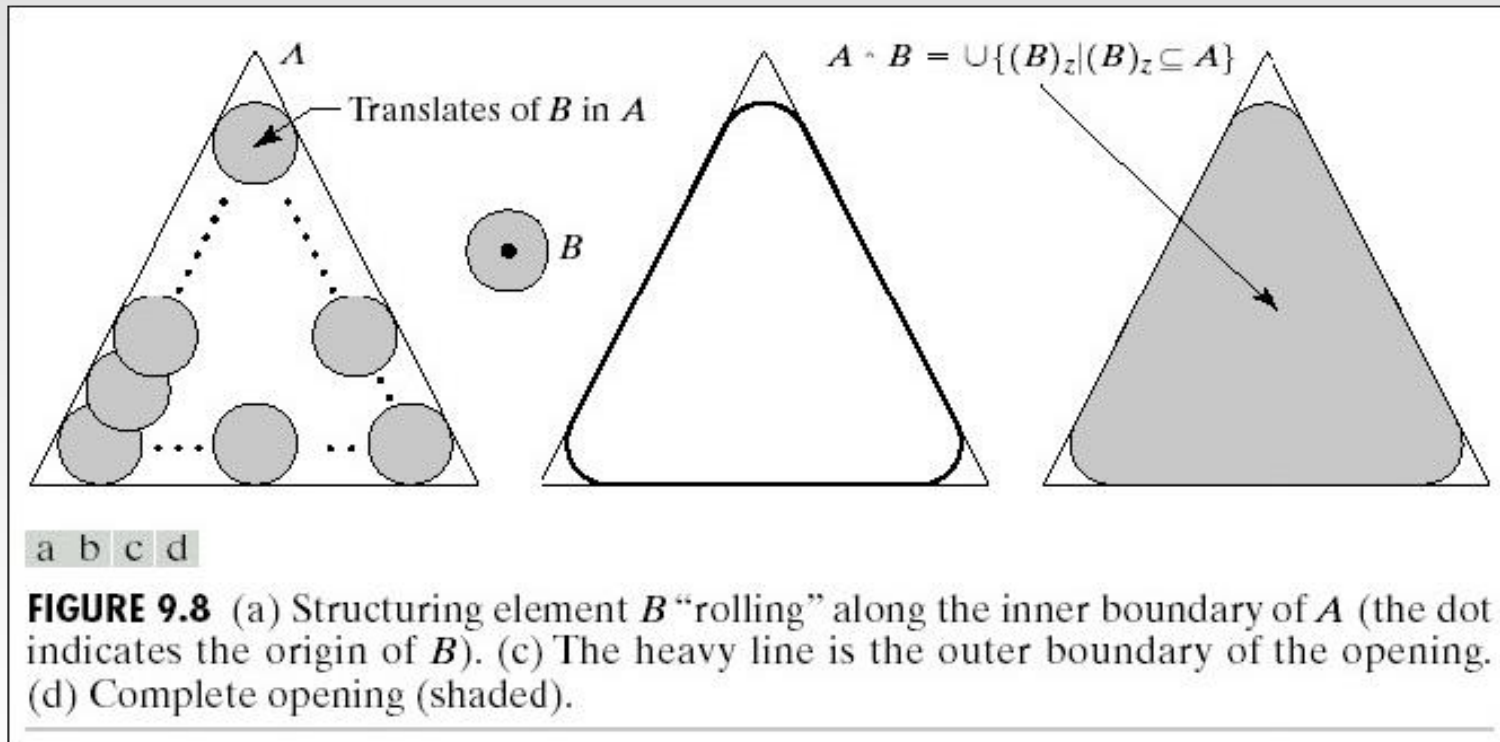
$$\begin{aligned}A \oplus B &= \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \\ (A^c \oplus \hat{B}) &= \{z \mid (B)_z \cap A^c \neq \emptyset\}\end{aligned}$$

- Podemos chegar a:

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

Algoritmos: Operadores morfológicos

Abertura

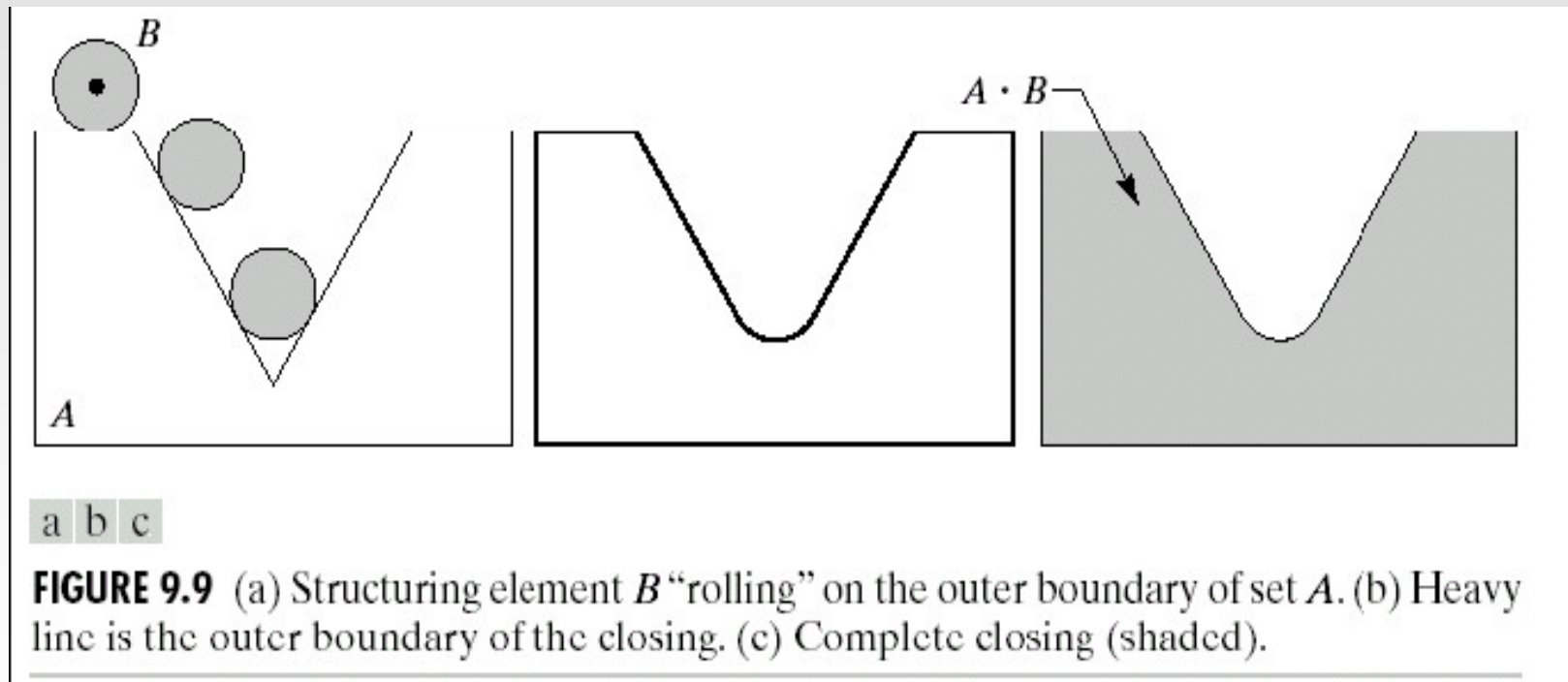


$$A \circ B = (A \ominus B) \oplus B$$

$$\underline{A \circ B = \cup \{(B)_z \mid (B)_z \subseteq A\}}$$

Algoritmos: Operadores morfológicos

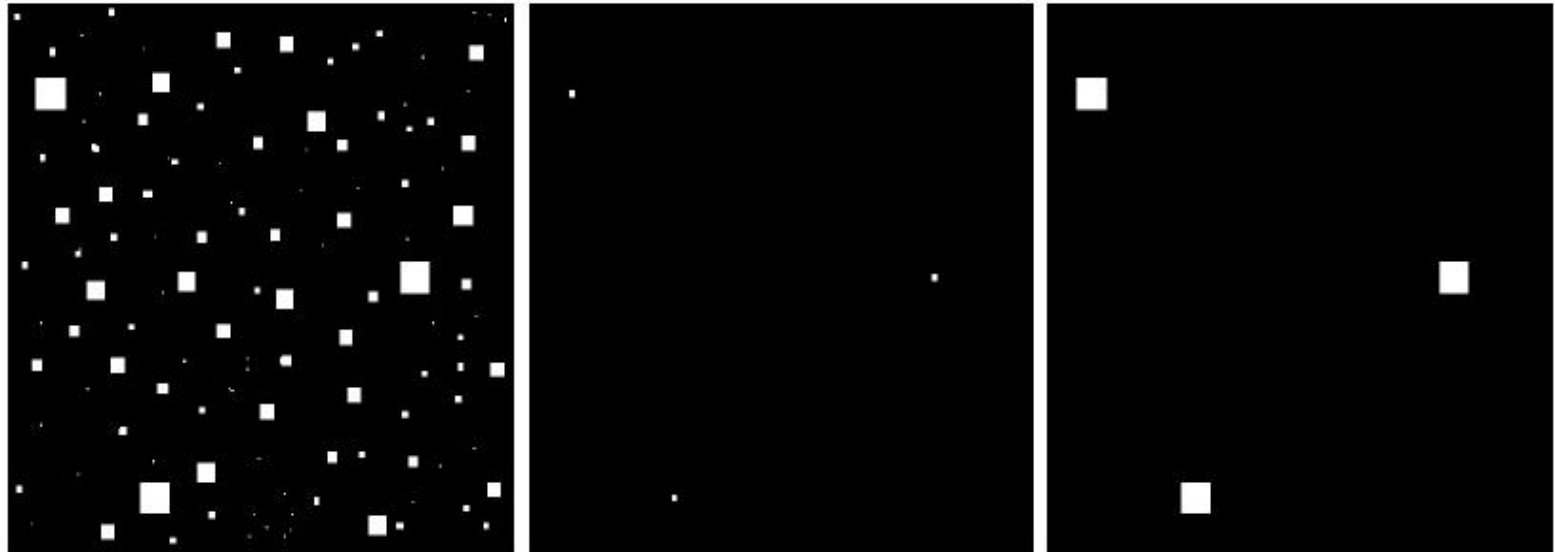
Fechamento



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

Algoritmos: Operadores morfológicos

Abertura: eliminação de detalhes irrelevantes



a b c

FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Erosão seguida de dilatação usando elemento estruturante
 $B = 13 \times 13$ pixels de nível de cinza 1

Algoritmos: Operadores morfológicos

Propriedades e características

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

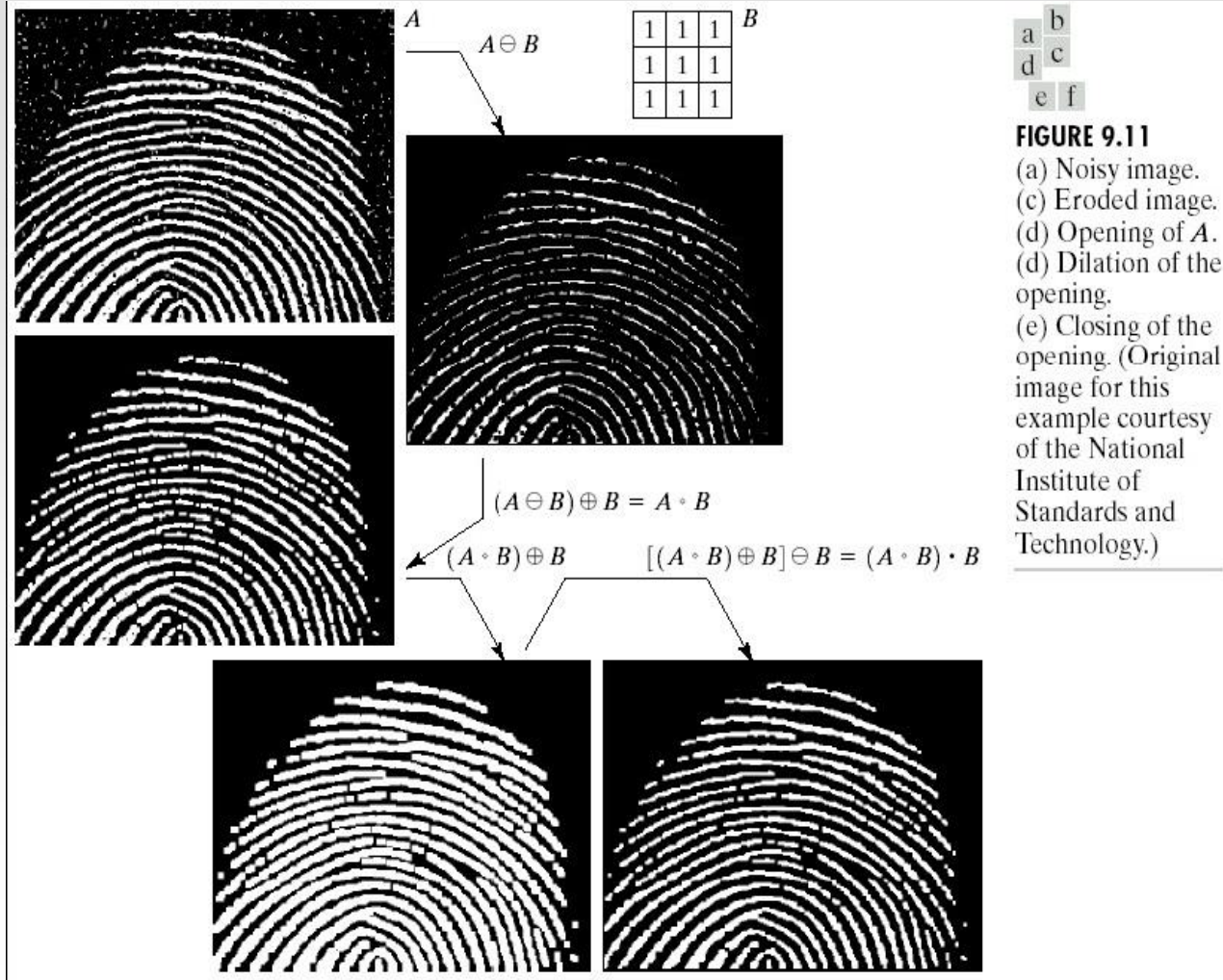
I. Abertura

- (i) $A \circ B$ é um subconjunto (subimagem) de A
- (ii) Se C é um subconjunto de D , então $C \circ B$ é um subconjunto de $D \circ B$
- (iii) $(A \circ B) \circ B = A \circ B$
- Partes pequenas do objeto são eliminadas.
- Objetos menores do que o elemento estruturante são eliminados.
- Separa objetos conectados por ligações finas.

2. Fechamento

- (i) A é um subconjunto (subimagem) de $A \cdot B$
- (ii) Se C é um subconjunto de D , então $C \cdot B$ é um subconjunto de $D \cdot B$
- (iii) $(A \cdot B) \cdot B = A \cdot B$
- Preenche pequenos espaços entre objetos.
- Preenche pequenos buracos.
- Preenche reentrâncias estreitas.

Morfologia



Algoritmos: Operadores morfológicos

- ***HitAndMiss***

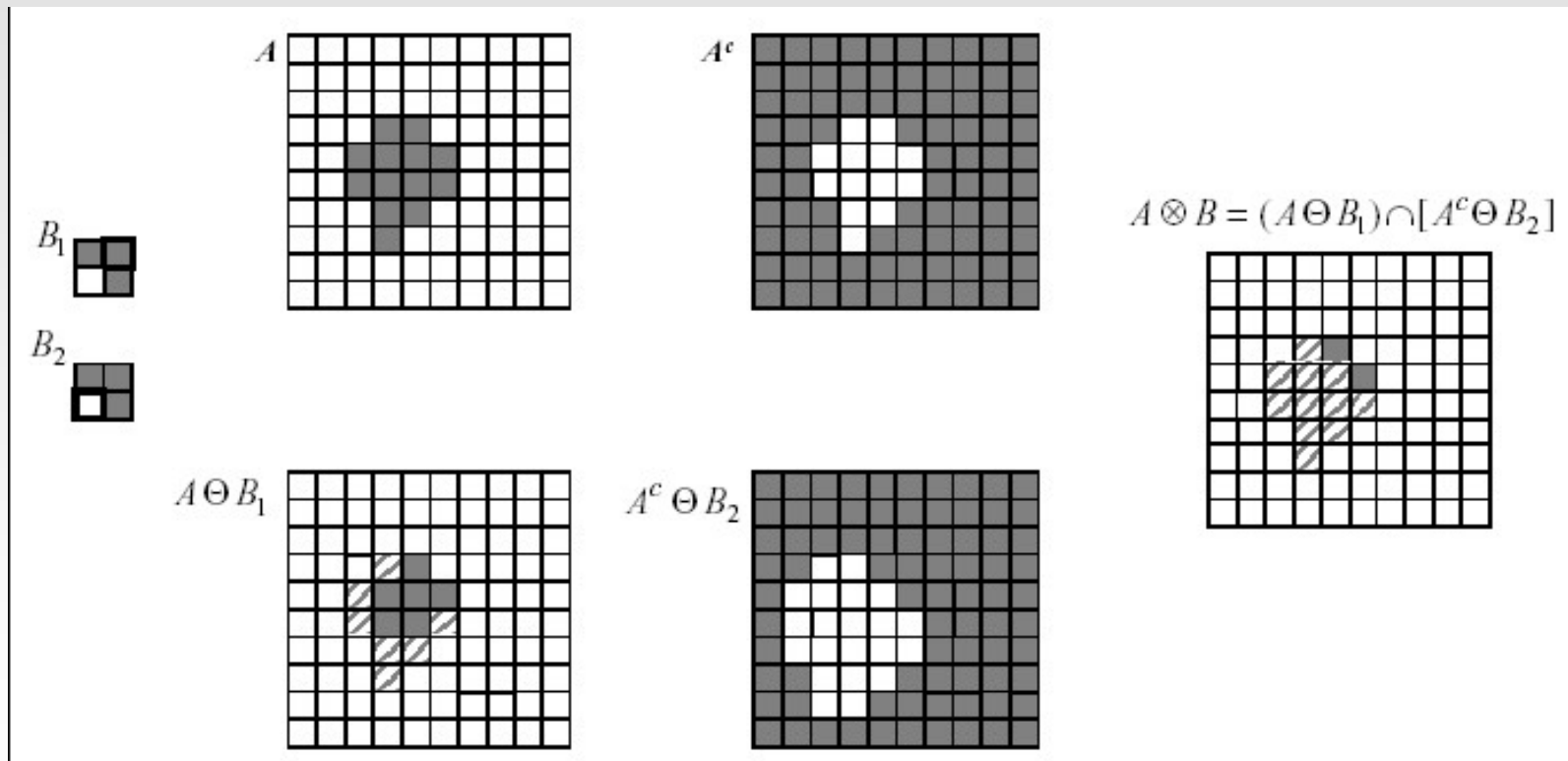
- Operador de alto nível

$$HitMiss(A,B) = E(A,B_1) \cap E^C(A^C,B_2)$$

- B_1 e B_2 são estruturantes finitos e disjuntos entre si
- Este operador é o equivalente morfológico do *template matching* (técnica usada para comparar padrões com base na sua correlação)
 - » B_1 funciona como *template* para o objeto
 - » B_2 funciona como *template* para o fundo

Algoritmos: Operadores morfológicos

HitAndMiss

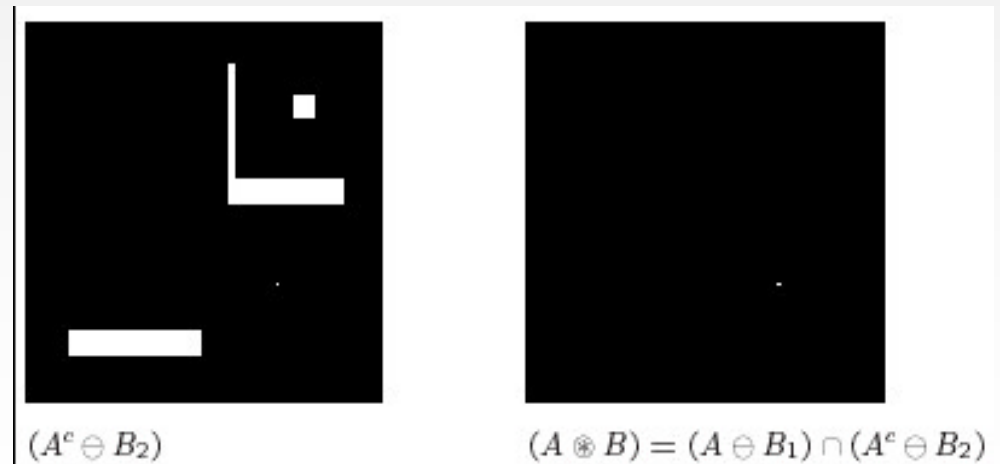
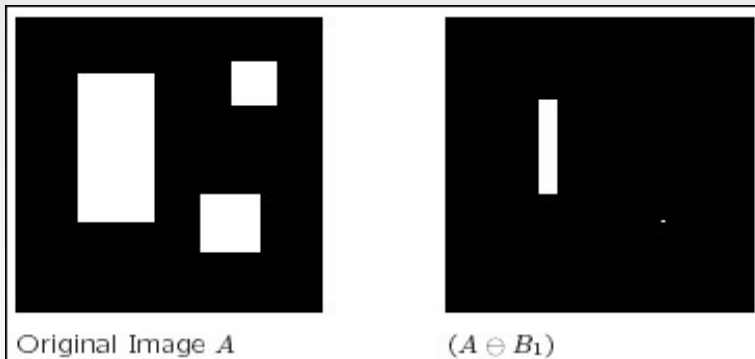
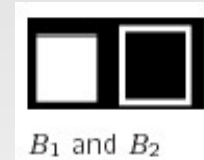


Algoritmos: Operadores morfológicos

HitAndMiss $(A \odot B) = (A \ominus B_1) \cap (A^c \ominus B_2)$

A = imagem

B1, B2 são elementos
estruturantes



Algoritmos: Operadores morfológicos

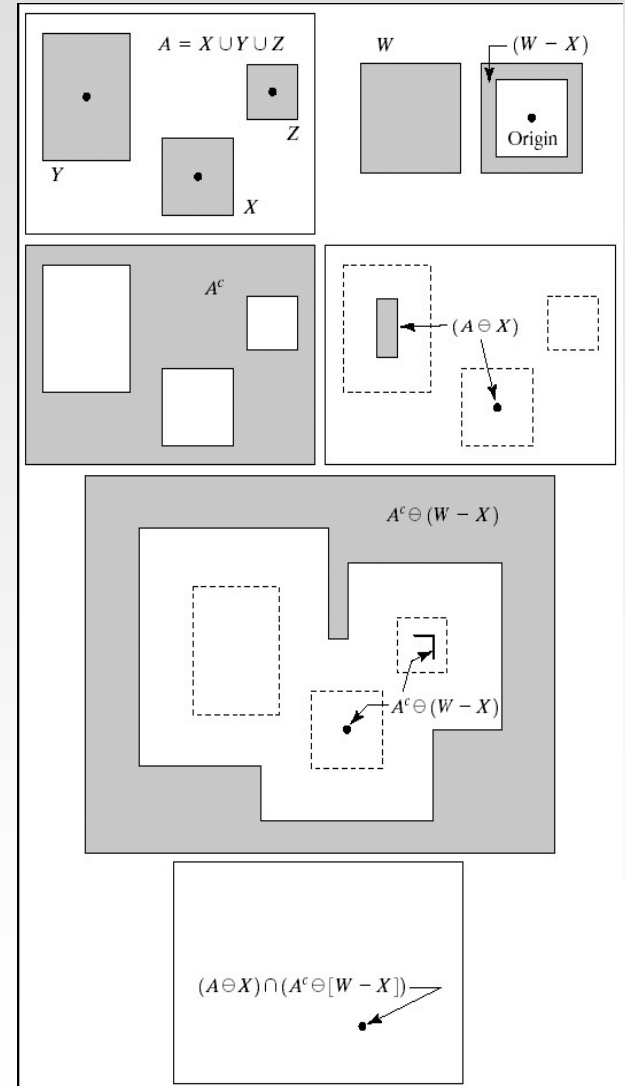
HitAndMiss $(A \circledast B) = (A \ominus B_1) \cap (A^c \ominus B_2)$

Utilizado para buscar objetos na imagem

$$(A \circledast B) = (A \ominus X) \cap [A^c \ominus (W - X)]$$

Dependente do elemento estruturante

FIGURE 9.12
 (a) Set A . (b) A window, W , and the local background of X with respect to W , $(W - X)$.
 (c) Complement of A . (d) Erosion of A by X .
 (e) Erosion of A^c by $(W - X)$.
 (f) Intersection of (d) and (e), showing the location of the origin of X , as desired.

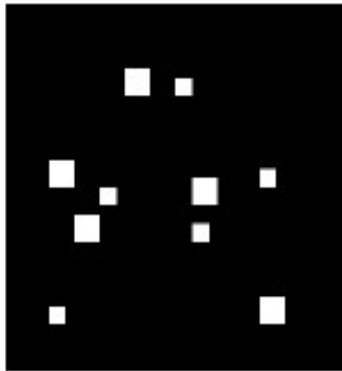


Algoritmos: Operadores morfológicos

HitAndMiss $(A \circledast B) = (A \ominus B_1) \cap (A^c \ominus B_2)$



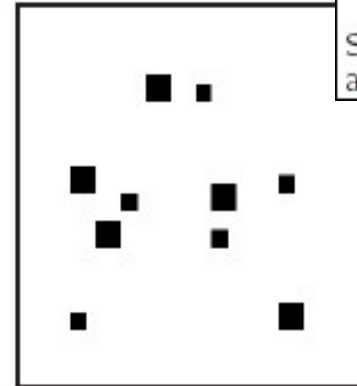
Structuring elements B_1 and B_2 . (Not to scale)



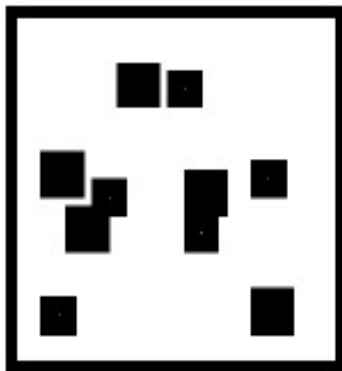
Original Image A



$C_1 = (A \ominus B_1)$



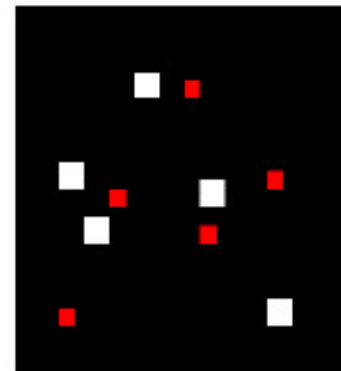
A^c



$C_2 = (A^c \ominus B_2)$



$(A \circledast B) = C_1 \cap C_2$



Detected objects shown in red.

Algoritmos: Operadores morfológicos

Extração de Contorno

A

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

E^c

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

$A \cap E^c$

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

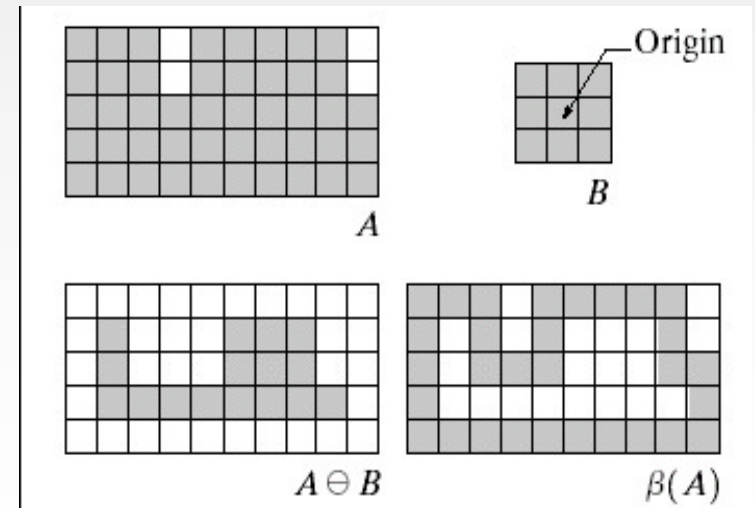
$A \ominus B$

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

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

$$\beta(A) = A \cap (A \ominus B)^c$$



Algoritmos: Operadores morfológicos

Extração de componentes conectados

Desenvolva um algoritmo para encontrar componentes conectados.

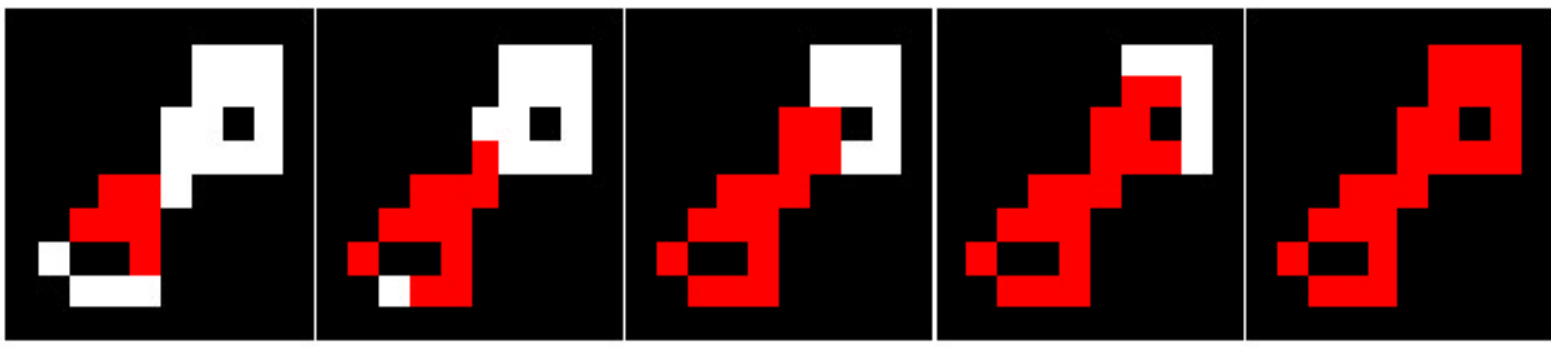
1. Seja B um elemento estruturante do tipo $N4$, Nd ou $N8$, dependendo da conectividade desejada.
2. Selecione um pixel p dentro do componente conectado a ser encontrado.
3. Inicialize X_0 como uma matriz de pixels '0' exceto $X_0[p]=1$.
4. Faça a iteração:

$$X_k = (X_{k-1} \oplus B) \cap A \text{ para } k = 1, 2, 3, \dots \text{ até } X_k = X_{k-1}$$

Algoritmos: Operadores morfológicos

Extração de componentes conectados

- Procura pelo elemento conectado começando pelo pixel vermelho



Algoritmos: Operadores morfológicos

Preenchendo Buracos

Desenvolva um algoritmo para preencher buracos (regiões fechadas).

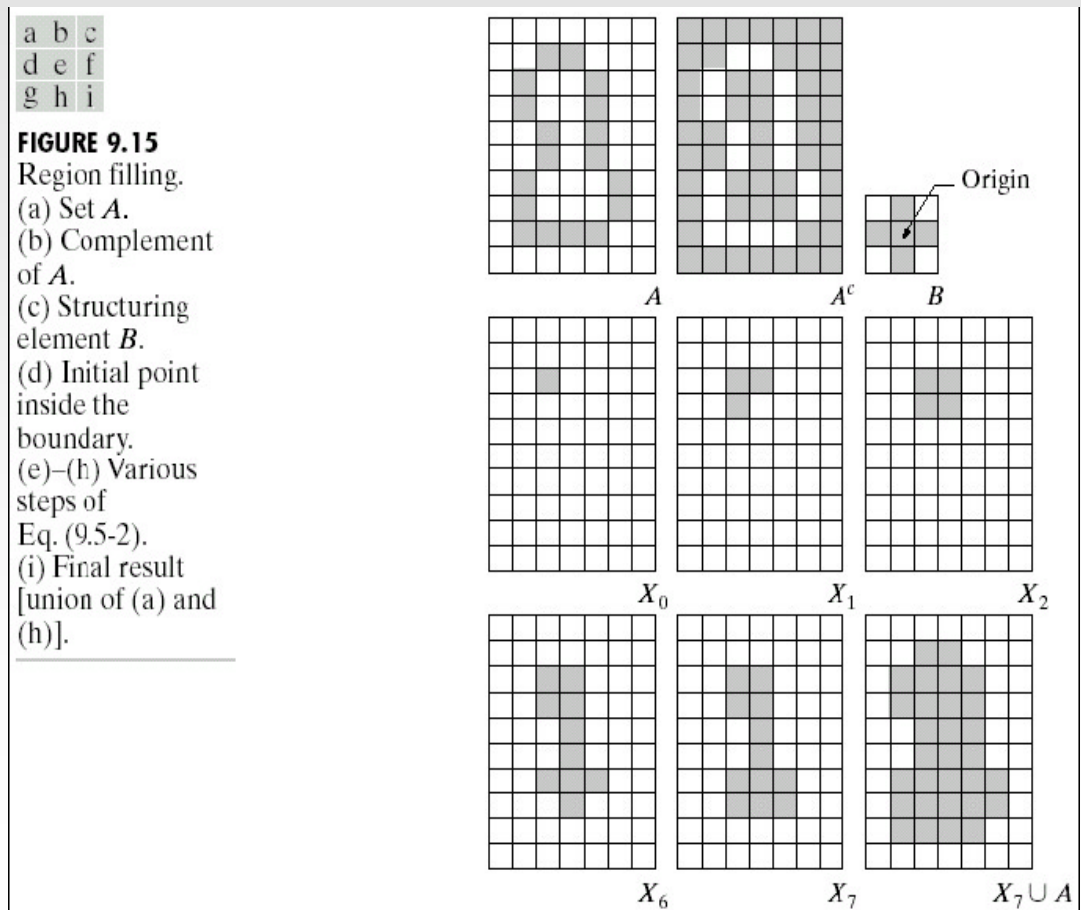
1. Seja B um elemento estruturante do tipo $N4$, Nd ou $N8$, dependendo da conectividade desejada.
2. Selecione um pixel p dentro do buraco a ser preenchido.
3. Inicialize X_0 como uma matriz de pixels '0' exceto $X_0[p] = 1$.
4. Faça a iteração:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad \text{para } k = 1, 2, 3, \dots \quad \text{até } X_k = X_{k-1}$$

Algoritmos: Operadores morfológicos

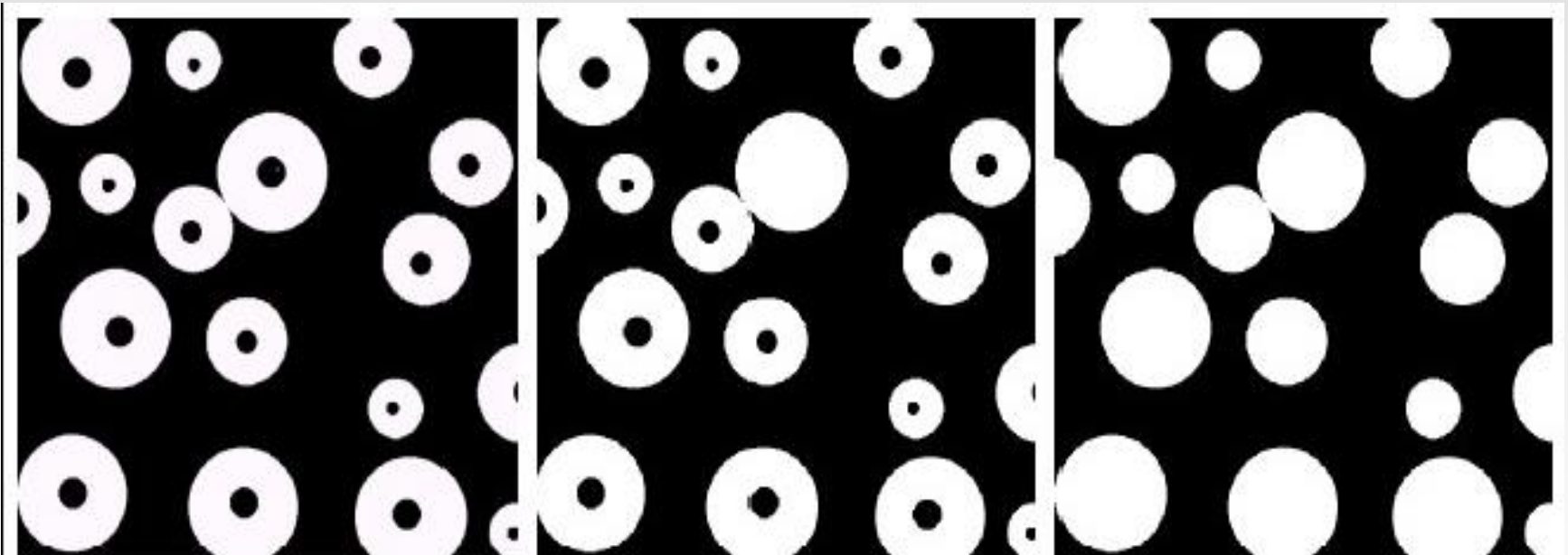
Preenchendo Buracos

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots \text{ até } X_k = X_{k-1}$$



Algoritmos: Operadores morfológicos

Preenchendo Buracos



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the region-filling algorithm). (b) Result of filling that region (c) Result of filling all regions.

Algoritmos: Operadores morfológicos

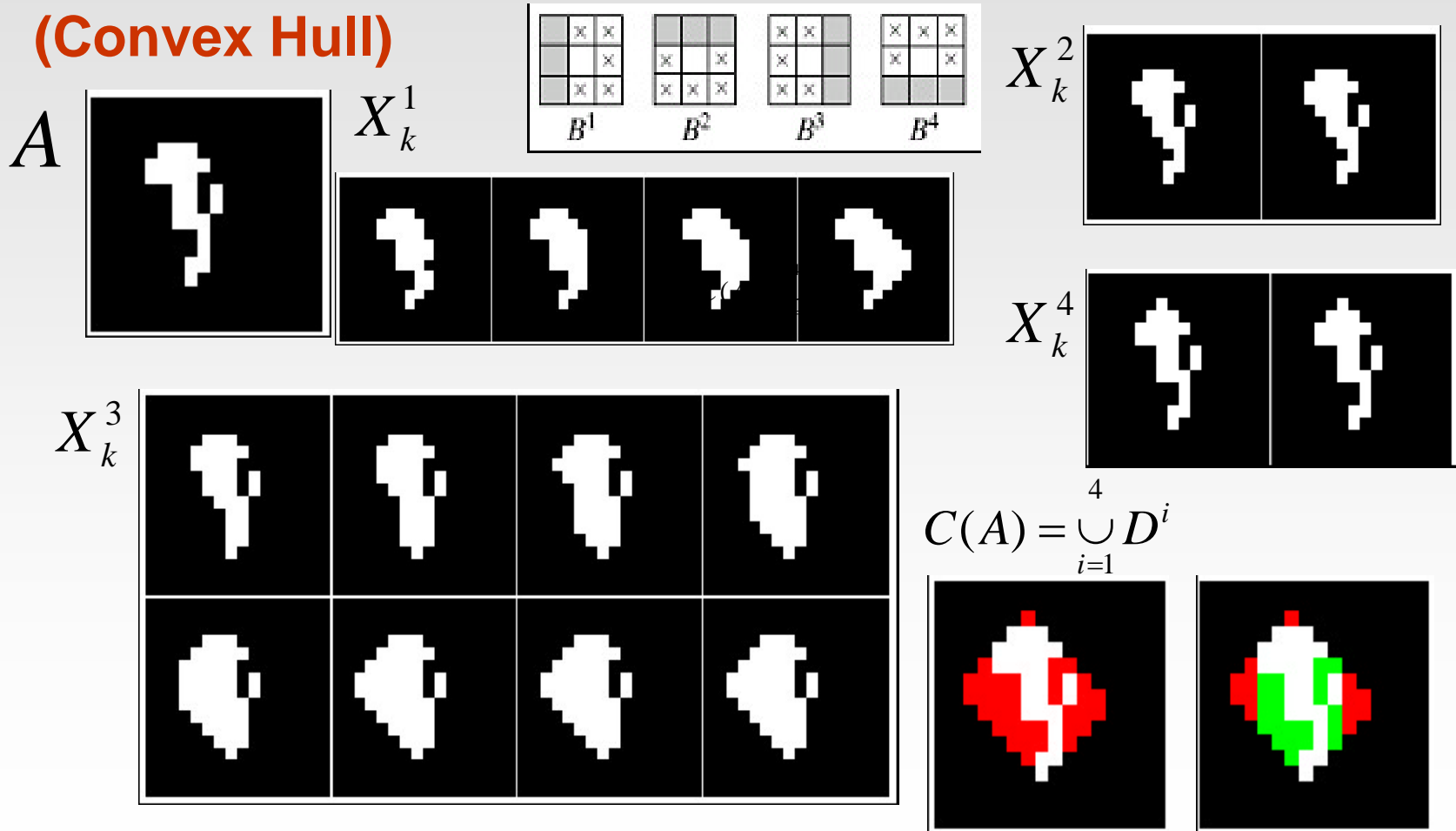
Feicho Convexo (convex hull)

- Um conjunto A é dito convexo se dois elementos quaisquer podem se unir por um caminho reto que não passa fora do conjunto.
- O '*convex hull*' H de um conjunto S é o menor conjunto convexo tal que $S \subseteq H$.
- O conjunto diferença $H - S = H \cap S^c$ é a *deficiência convexa* de S .

Algoritmos: Operadores morfológicos

Fecho convexo $X_k^i = (X_k^i \circledast B^i) \cup A, \quad i=1,2,3,4 \text{ e } k=1,2,3\dots$

(Convex Hull)



Algoritmos: Operadores morfológicos

Afinamento (Thining)

Afinamento remove pixels de um conjunto até ficar apenas um estreito conjunto.

É usado para revelar estruturas em aplicações de reconhecimento de padrões.

Afinamento usa uma seqüência de elementos estruturantes. $B = \{B_1, B_2, \dots, B_n\}$

Operação básica

1. Faça $X_n = A$ e $Y = X_0 = [0]_{N \times M}$

2. Repita enquanto $X_n \neq Y$

$$Y = X_n$$

$$X_0 = X_n$$

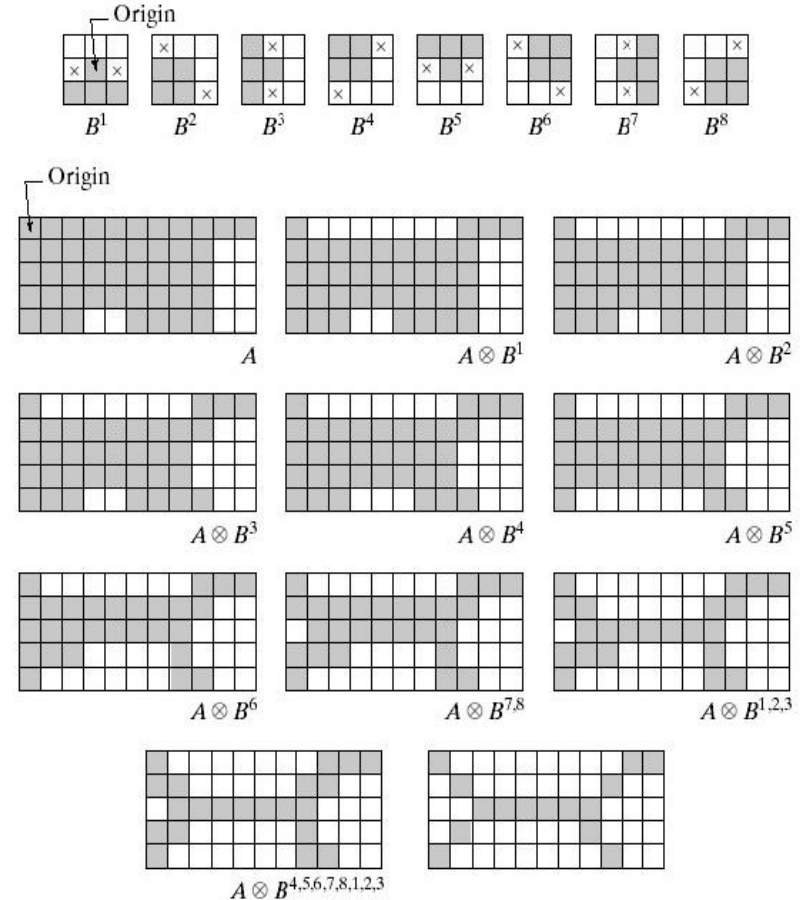
$$X_i = X_{i-1} \cap (X_{i-1} \otimes B_i)^c \quad \text{para } i = 1, 2, 3, \dots, n$$

Algoritmos: Operadores morfológicos

Afinamento (Thining)

$$A \otimes B = A - (A \circledast B)$$

$$= A \cap (A \circledast B)^c$$



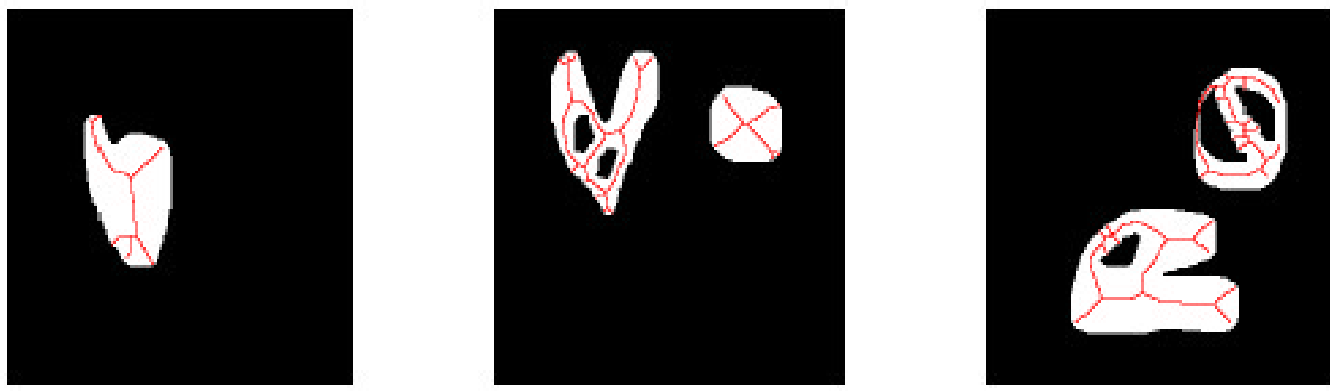
| |
|-------|
| a |
| b c d |
| e f g |
| h i j |
| k l |

FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A . (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m -connectivity.

Algoritmos: Operadores morfológicos

Afinamento (Thining)

Exemplo: Emagreça as bolhas



A estrutura básica é capturada pelos objetos emagrecidos (vermelho). As pequenas terminações poderiam ser removidas por um processamento adicional para melhorar a definição.

Esperçamento (Thickening)

- Esperçamento é o dual morfológico de Afinamento e é definido por

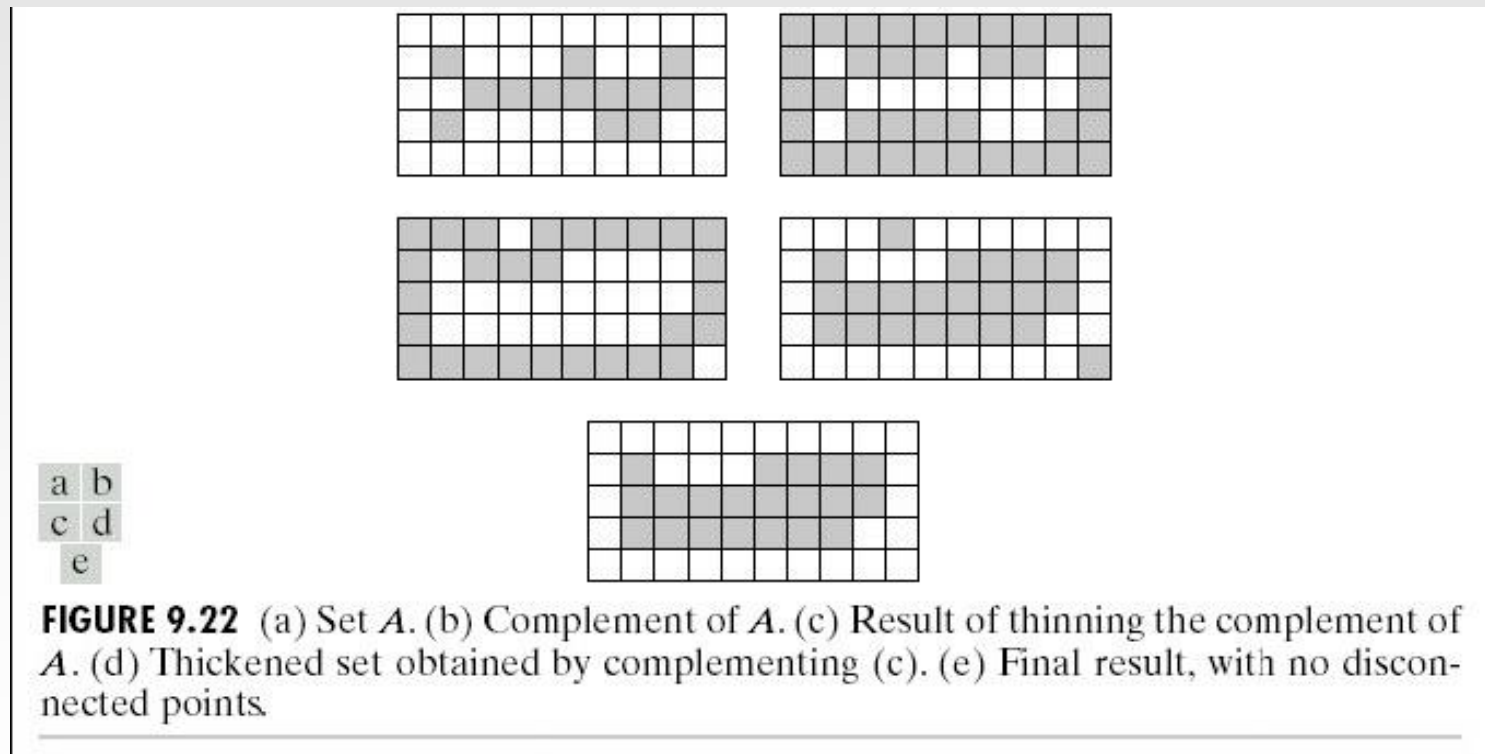
$$A \odot B = A \cup (A \circledast B)$$

- O algoritmo pode ser programado diretamente da definição. Entretanto, descobriu-se que é mais efetivo fazer o thickening através do thinning de A^c .
1. Faça $C = A^c$
 2. Faça D ser afinado
 3. $T = D^c$ é o objeto esperçado.

Algoritmos: Operadores morfológicos

Esperçamento

$$A \odot B = A \cup (A \circledast B)$$



- **Esqueleto**

- Quando realizável, é uma polilinha que:
 - » Tem largura de 1 pixel
 - » Passa pelo “meio” do objeto
 - » Preserva a topologia do objeto
- Pode ser obtido por um processo iterativo baseado no B_8 , não se alterando o pixel central para “fundo” se

- » 1) o pixel estiver isolado



- » 2) a remoção do pixel alterar a conectividade



- » 3) a remoção do pixel encurtar a linha



Algoritmos Operadores morfológicos

Esqueletos

Pontos centrais de bolas máximas (geradas a partir do elemento estruturante B) contidas no objeto da imagem de entrada.

$$S(A) = \bigcup_{k=0}^K S_k(A)$$

$$S_k(A) = (A - kB) - (A - kB) \circ B$$

$$K = \max\{k \mid (aA - kB) \neq \Phi\}$$

$$A = \bigcup_{k=0}^K (S_k(A) \oplus kB)$$

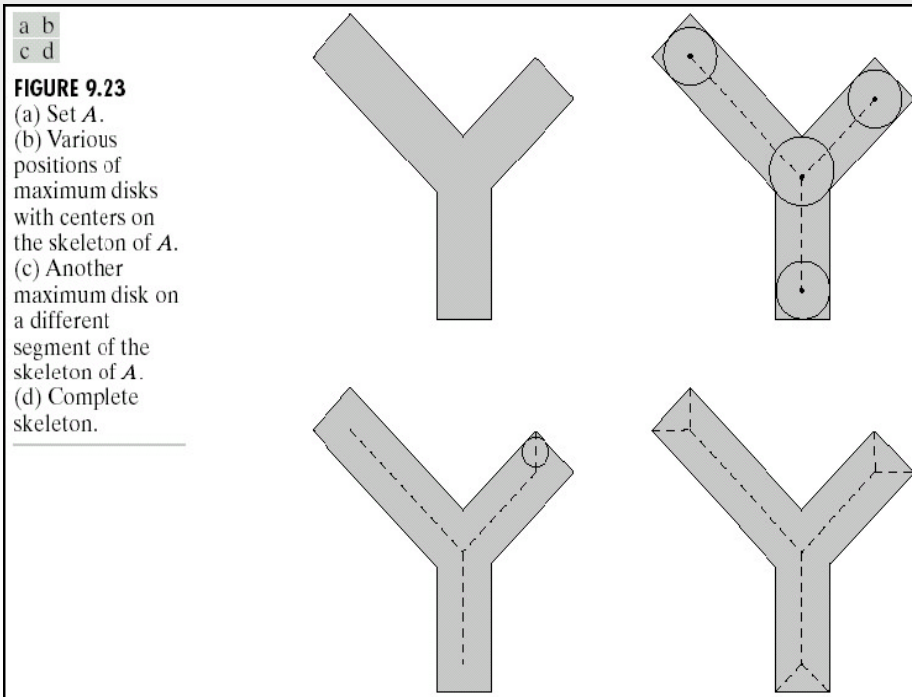


FIGURE 9.23
 (a) Set A .
 (b) Various positions of maximum disks with centers on the skeleton of A .
 (c) Another maximum disk on a different segment of the skeleton of A .
 (d) Complete skeleton.

Algoritmos: Operadores morfológicos

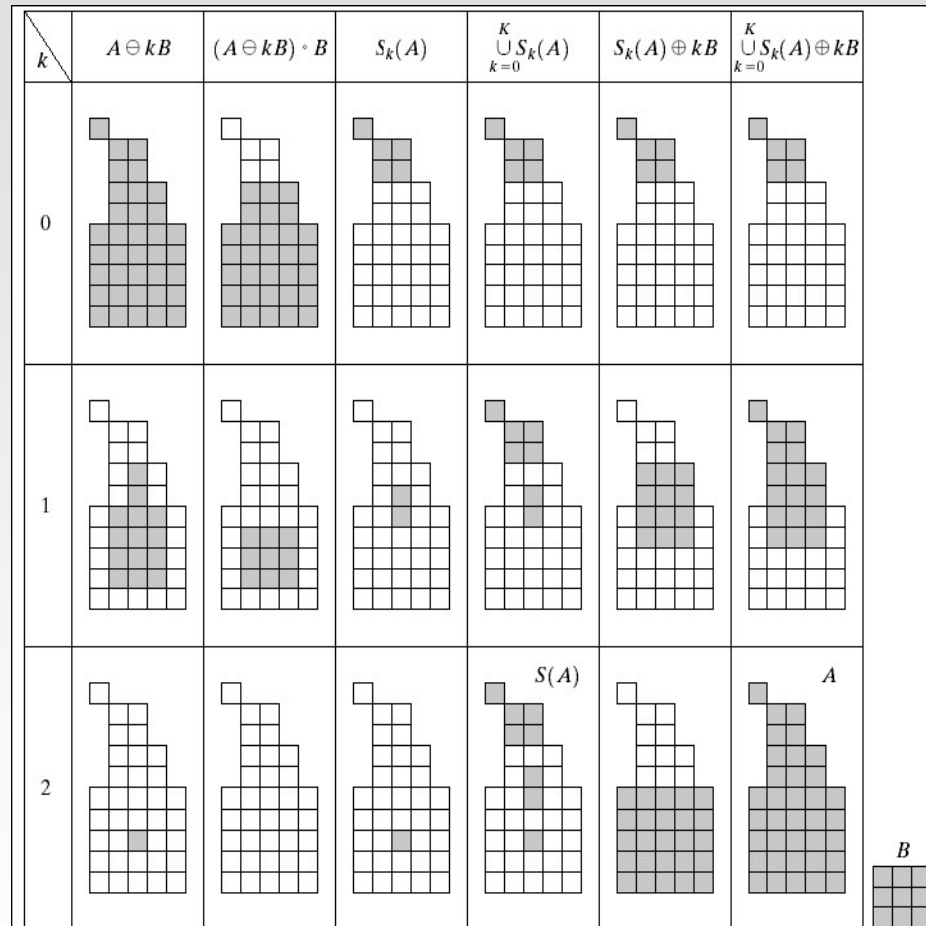
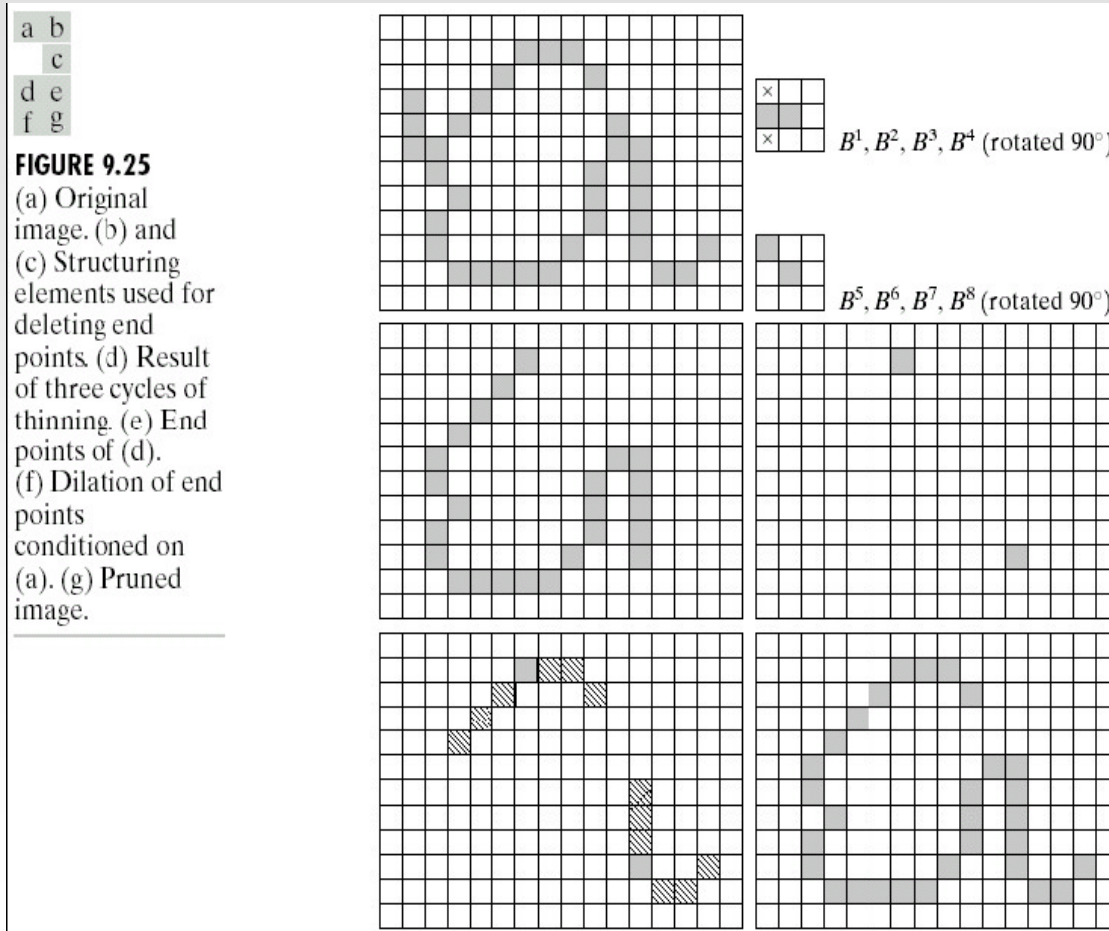


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

Algoritmos: Operadores morfológicos

Poda H = elemento estruturante 3x3 de 1's



$$X_1 = A \otimes \{B\}$$

$$X_2 = \bigcup_{k=1}^K (X_1 \otimes B^k)$$

$$X_3 = (X_2 \oplus H) \cap A$$

$$X_4 = X_1 \cup X_3$$

Algoritmos: Baseados em morfologia

- Exemplos

- Estruturantes

$$B = N_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} - & - & - \\ - & 1 & - \\ - & - & - \end{bmatrix}$$

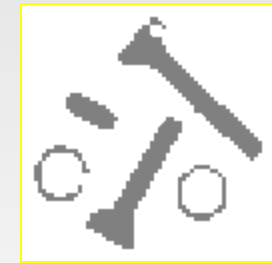
$$B_2 = \begin{bmatrix} - & 1 & - \\ 1 & - & 1 \\ - & 1 & - \end{bmatrix}$$



Imagem A



Dilatação com 2B



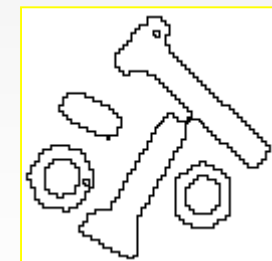
Erosão com 2B



Abertura com 2B
(separa objetos)



Fechamento com 2B
(preenche buracos)



HitAndMiss com B_1 e B_2
 $[\Leftrightarrow \partial A = A - E(A, N_8)]$

TABLE 9.2
Summary of
morphological
operations and
their properties.

| Operation | Equation | Comments |
|-------------|--|---|
| Translation | $(A)_z = \{w w = a + z, \text{ for } a \in A\}$ | (The Roman numerals refer to the structuring elements shown in Fig. 9.26). Translates the origin of A to point z . |
| Reflection | $\hat{B} = \{w w = -b, \text{ for } b \in B\}$ | Reflects all elements of B about the origin of this set. |
| Complement | $A^c = \{w w \notin A\}$ | Set of points not in A . |
| Difference | $A - B = \{w w \in A, w \notin B\}$ $= A \cap B^c$ | Set of points that belong to A but not to B . |
| Dilation | $A \oplus B = \{z (\hat{B})_z \cap A \neq \emptyset\}$ | “Expands” the boundary of A . (I) |
| Erosion | $A \ominus B = \{z (B)_z \subseteq A\}$ | “Contracts” the boundary of A . (I) |
| Opening | $A \circ B = (A \ominus B) \oplus B$ | Smooths contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I) |
| Closing | $A \bullet B = (A \oplus B) \ominus B$ | Smooths contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I) |

| | | |
|-----------------------|---|---|
| Hit-or-miss transform | $A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus \hat{B}_2)$ | The set of points (coordinates) at which, simultaneously, B_1 found a match (“hit”) in A and B_2 found a match in A^c . |
| Boundary extraction | $\beta(A) = A - (A \ominus B)$ | Set of points on the boundary of set A . (I) |
| Region filling | $X_k = (X_{k-1} \oplus B) \cap A; X_0 = p \text{ and } k = 1, 2, 3, \dots$ | Fills a region in A , given a point p in the region. (II) |
| Connected components | $X_k = (X_{k-1} \oplus B) \cap A; X_0 = p \text{ and } k = 1, 2, 3, \dots$ | Finds a connected component Y in A , given a point p in Y . (I) |
| Convex hull | $X_k^i = (X_{k-1}^i \otimes B^i) \cup A; i = 1, 2, 3, 4;$ $k = 1, 2, 3, \dots; X_0^i = A; \text{ and}$ $D^i = X_{\text{conv}}^i.$ | Finds the convex hull $C(A)$ of set A , where “conv” indicates convergence in the sense that $X_k^i = X_{k-1}^i$. (III) |

| Operation | Equation | Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26). |
|------------|---|--|
| Thinning | $A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\dots ((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$ | <p>Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)</p> |
| Thickening | $A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $((\dots (A \odot B^1) \odot B^2 \dots) \odot B^n)$ | <p>Thickens set A. (See preceding comments on sequences of structuring elements.) Uses IV with 0's and 1's reversed.</p> |

TABLE 9.2
Summary of morphological results and their properties.
(continued)

Skeletons

$$S(A) = \bigcup_{k=0}^K S_k(A)$$
$$S_k(A) = \bigcup_{k=0}^K \{ (A \ominus kB) - [(A \ominus kB) \circ B] \}$$

Reconstruction of A :

$$A = \bigcup_{k=0}^K (S_k(A) \oplus kB)$$

Finds the skeleton $S(A)$ of set A . The last equation indicates that A can be reconstructed from its skeleton subsets $S_k(A)$. In all three equations, K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the k th iteration of successive erosion of A by B . (I)

Pruning

$$X_1 = A \otimes \{B\}$$
$$X_2 = \bigcup_{k=1}^8 (X_1 \otimes B^k)$$
$$X_3 = (X_2 \oplus H) \cap A$$
$$X_4 = X_1 \cup X_3$$

X_4 is the result of pruning set A . The number of times that the first equation is applied to obtain X_1 must be specified. Structuring elements V are used for the first two equations. In the third equation H denotes structuring element I .