

Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Procesarea Imagineilor Digitale

Curs - Morfologie Matematică

Universitatea "Transilvania" din Brașov

Cuprins

1 Concepte de bază

2 Operații morfologice pentru imagini binare

- Erodare
- Dilatarea
- Opening
- Closing
- Aplicații

3 Operații morfologice pe imagini în tonuri de gri

4 Operații morfologice pe imagini color

Morfologia matematică - Introducere

- oferă instrumente de preprocesare a imaginii.
- permite extragerea de componente sau schelete.
- are la bază teoria mulțimilor și operațiilor logice.
- consideră mulțimi din $\mathbb{Z}^2 =$ mulțimi de coordonate.

Notări din teoria mulțimilor

- $C_A = \{w \mid w \notin A\}$ - complementara mulțimii A
- $A - B = \{w \mid w \in A \text{ și } w \notin B\} = A \cap C_B$
- $\hat{B} = \{w \mid w = -b, b \in B\}$ - simetricul lui B față de origine
- $(A)_z = \{w \mid w = a + z, a \in A\}$ - translația lui A cu $z = (z_1, z_2)$

Observație: mulțimile A și B sunt mulțimi de coordonate.

Operații logice cu imagini binare

p	q	NOT p	p AND q	p OR q	p XOR q
0	0				
0	1				
1	0				
1	1				

Tabelă: În imagini binare considerăm 1 = pixel alb, 0 = pixel negru.

Operații logice cu imagini binare

p	q	NOT p	p AND q	p OR q	p XOR q
0	0	1			
0	1	1			
1	0	0			
1	1	0			

Tabelă: În imagini binare considerăm 1 = pixel alb, 0 = pixel negru.

Operații logice cu imagini binare

p	q	NOT p	p AND q	p OR q	p XOR q
0	0	1	0		
0	1	1	0		
1	0	0	0		
1	1	0	1		

Tabelă: În imagini binare considerăm 1 = pixel alb, 0 = pixel negru.

Operații logice cu imagini binare

p	q	NOT p	p AND q	p OR q	p XOR q
0	0	1	0	0	
0	1	1	0	1	
1	0	0	0	1	
1	1	0	1	1	

Tabelă: În imagini binare considerăm 1 = pixel alb, 0 = pixel negru.

Operații logice cu imagini binare

p	q	NOT p	p AND q	p OR q	p XOR q
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Tabela: În imagini binare considerăm 1 = pixel alb, 0 = pixel negru.

Operații logice cu imagini binare

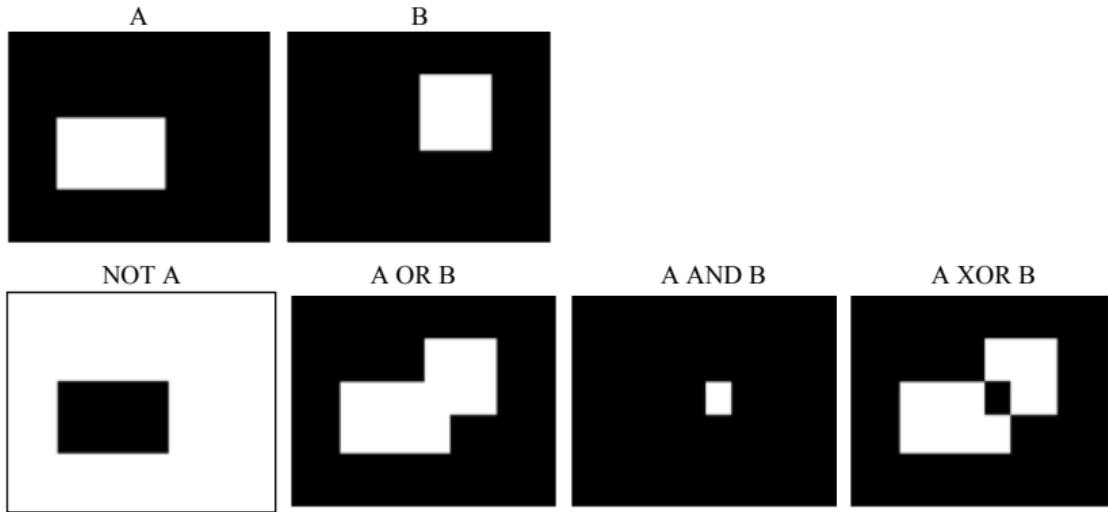


Figura: Imaginele binare conținând mulțimile A și B împreună cu imaginile rezultate prin aplicarea operațiilor logice NOT, AND, OR și XOR asupra acestor imagini.

Erodare

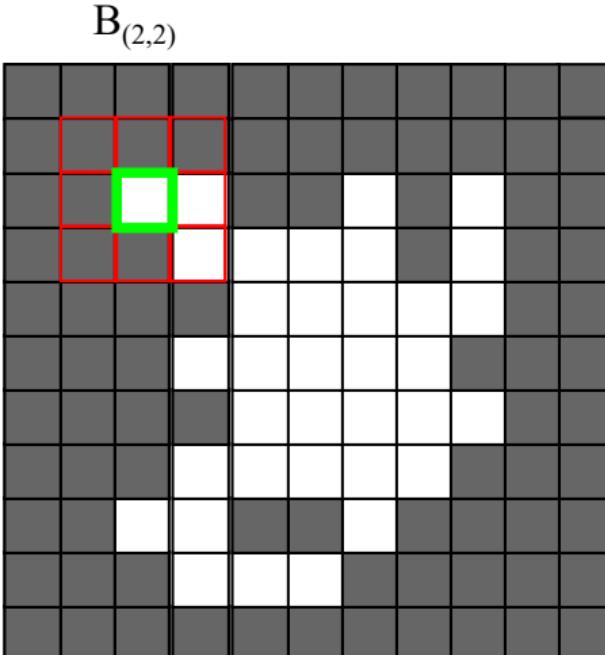
- **Definiție matematică:** Fie A și B în \mathbb{Z}^2 . Erodarea lui A cu B este:

$$A \ominus B = \{z = (x, y) \in \mathbb{Z}^2 | (B)_z \subseteq A\}$$

- **Element structurant:** B - de obicei $\hat{B} = B$, se consideră centrat în $(0, 0)$

Erodare

$$B = \begin{matrix} & -1 & 0 & 1 \\ -1 & \boxed{} & \boxed{} & \boxed{} \\ 0 & \boxed{} & \boxed{} & \boxed{} \\ 1 & \boxed{} & \boxed{} & \boxed{} \end{matrix}$$

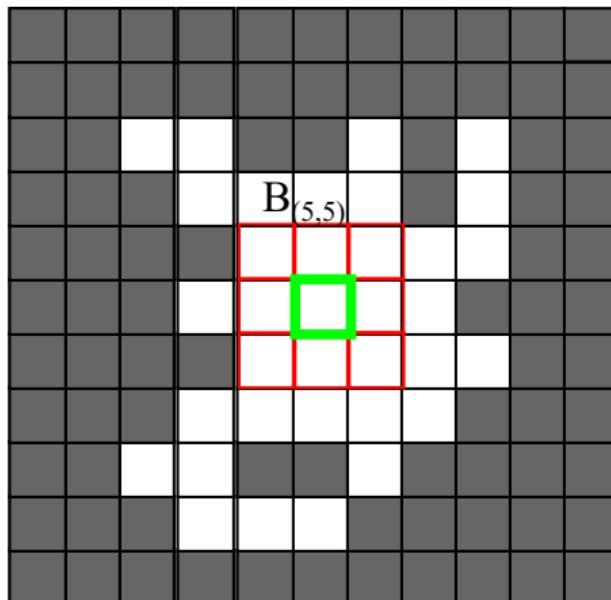


$A =$ zona marcată cu alb

pixelul marcat cu verde
se elimină din
multimea A erodată

Erodare

$$B = \begin{matrix} & -1 & 0 & 1 \\ -1 & & & \\ 0 & & & \\ 1 & & & \end{matrix}$$



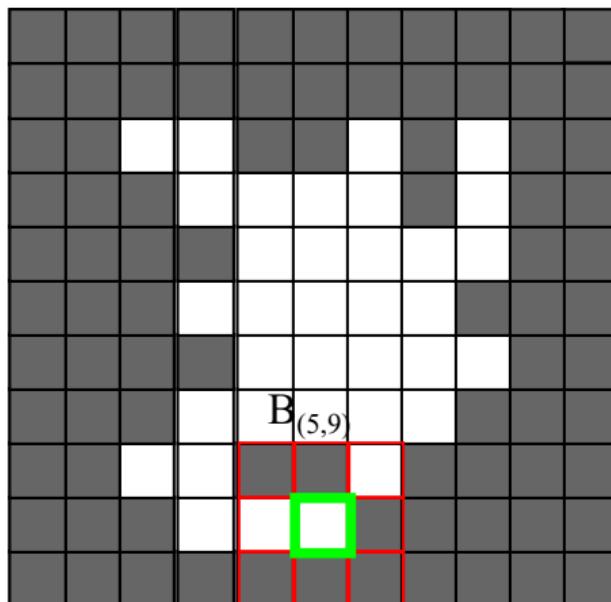
A = zona marcată cu alb

pixelul marcat cu verde
 NU se elimină din
 multimea A erodată

Erodare

B

$$\begin{matrix} & -1 & 0 & 1 \\ -1 & \boxed{} & \boxed{} & \boxed{} \\ 0 & \boxed{} & \boxed{} & \boxed{} \\ 1 & \boxed{} & \boxed{} & \boxed{} \end{matrix}$$



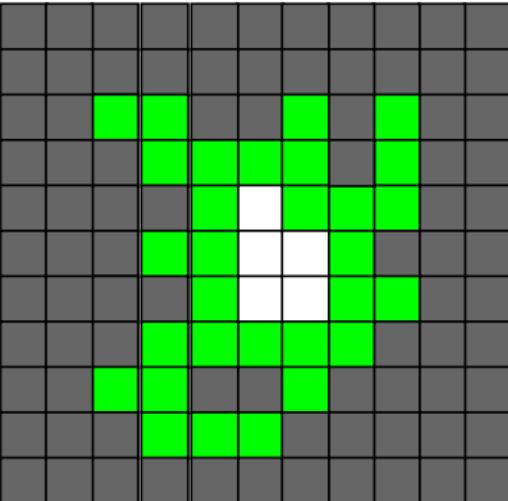
$A =$ zona marcată cu alb

pixelul marcat cu verde
se elimină din
multimea A erodată

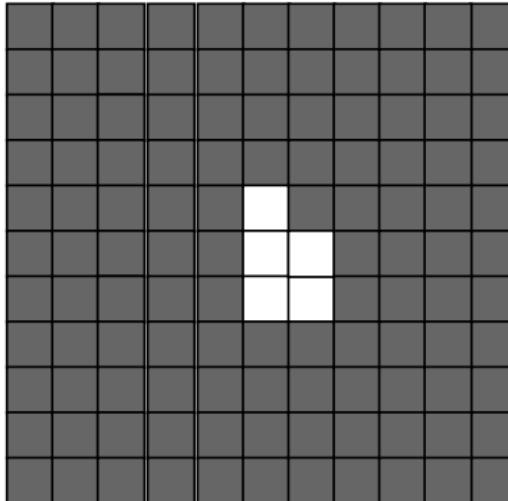
Erodare

B	-1	0	1
-1			
0			
1			

Pixelii erodați - marcați cu verde.

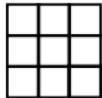
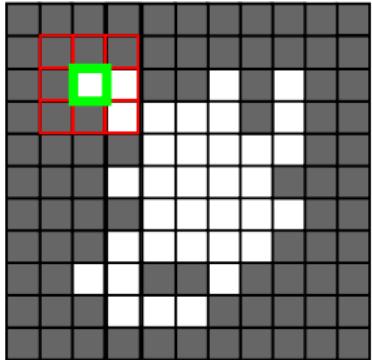


Rezultatul erodării



Erodare

B

 $B_{(2,2)}$ 

Algoritm 1: Erodare

Intrare: Imaginea sursă f binară, masca B

Iesire: Imaginea rezultat g binară

pentru fiecare (x, y) coordonată **executa**

daca $\exists(s, t) \in B_{xy}$ cu $f(s, t) = 0$ **atunci**
 | $g(x, y) = 0$

sfarsit_daca

altfel

| $g(x, y) = 255$

sfarsit_daca

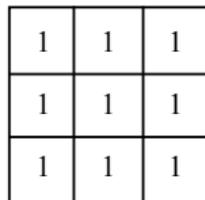
sfarsit_for

Erodare - Exemplu

1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1



c)



a)



d)



e)

Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Erodare

Dilatarea

Opening

Closing

Aplicații

Erodare - Proprietăți

...
•

Erodare - Proprietăți

- ...
- Reduce obiectele din imagine.

Erodare - Proprietăți

- ...
- Reduce obiectele din imagine.
- Elimină detalii nerelevante mai mici decât elementul structurant (zgomot).

Erodare - Proprietăți

- ...
- Reduce obiectele din imagine.
- Elimină detalii nerelevante mai mici decât elementul structurant (zgomot).
- Mărește fragmentarea contururilor.

Dilatarea

- **Definiție matematică:** Fie A și B în \mathbb{Z}^2 dilatarea lui A cu B este:

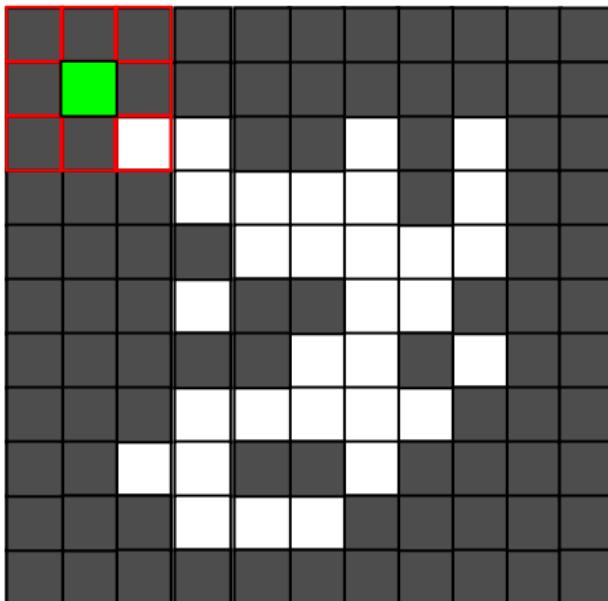
$$A \oplus B = \{z = (x, y) \in \mathbb{Z}^2 | (\hat{B})_z \cap A \neq \emptyset\}$$

- **Element structurant:** B - de obicei $\hat{B} = B$, se consideră centrat în $(0, 0)$

Dilatarea

$$B = \begin{matrix} & -1 & 0 & 1 \\ -1 & & & \\ 0 & & & \\ 1 & & & \end{matrix}$$

$$B_{(1,1)}$$



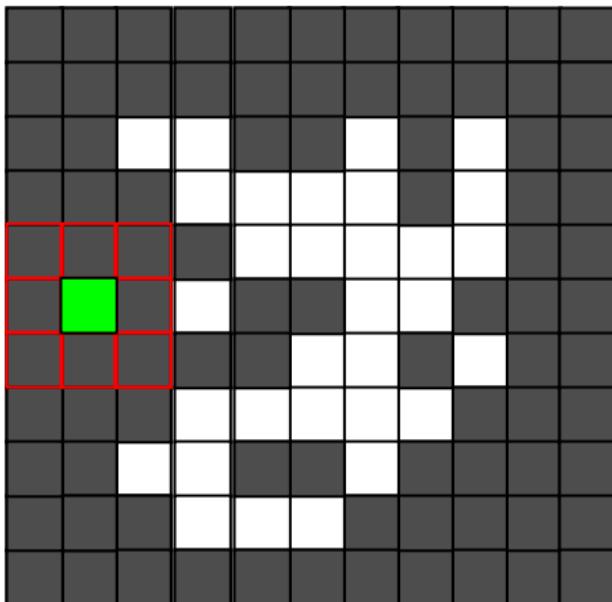
A = zona marcată cu alb

pixelul marcat cu verde
va face parte din dilatarea
multimii A

Dilatarea

$$B = \begin{matrix} & -1 & 0 & 1 \\ -1 & & & \\ 0 & & & \\ 1 & & & \end{matrix}$$

$$B_{(1,5)}$$

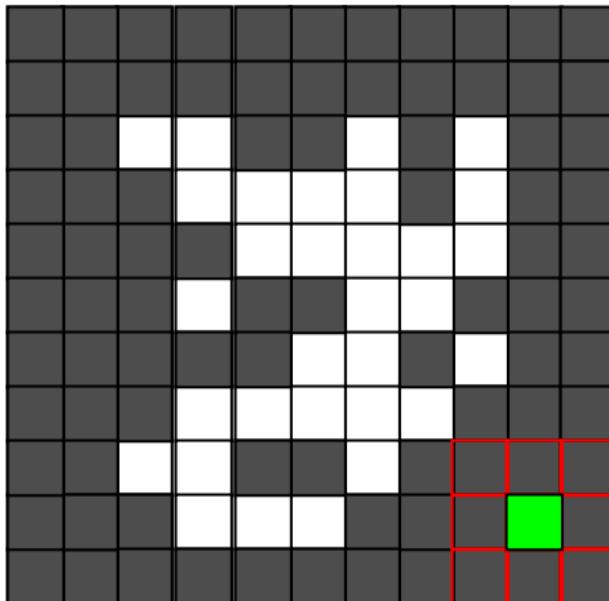


A = zona marcată cu alb

pixelul marcat cu verde
NU va face parte
din dilatarea mulțimii A

Dilatarea

$$B = \begin{matrix} & -1 & 0 & 1 \\ -1 & & & \\ 0 & & & \\ 1 & & & \end{matrix}$$



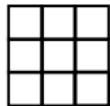
A = zona marcată cu alb

pixelul marcat cu verde
NU va face parte
din dilatarea mulțimii A

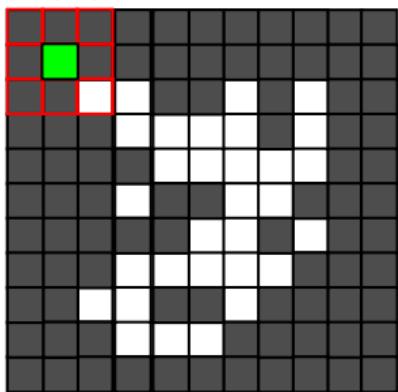
B_(9,9)

Dilatarea

B



$B_{(1,1)}$



Algoritm 2: Dilatare

Intrare: Imaginea sursă f binară, masca B

Iesire: Imaginea rezultat g binară

pentru fiecare (x, y) coordonată **executa**

daca $\exists(s, t) \in B_{xy}$ cu $f(s, t) = 255$ **atunci**
| $g(x, y) = 255$

sfarsit_daca

altfel

| $g(x, y) = 0$

sfarsit_daca

sfarsit_for

Observație: echivalență cu operația OR asupra pixelilor din mască.

Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Erodare

Dilatarea

Opening

Closing

Aplicații

Dilatarea - Exemplu

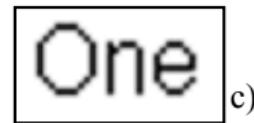
Dilatarea - Diferite elemente structurante

1	1	1
1	1	1
1	1	1

a)

0	1	0
1	1	1
0	1	0

b)



c)



d)



e)

Figura: a) Element structurant pătrat; b) Element structurant circular; c) Imagine originală; d) Imagine dilatătă cu a); e) Imagine dilatătă cu b).

Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Erodare

Dilatarea

Opening

Closing

Aplicații

Dilatarea - Proprietăți

...
•

Dilatarea - Proprietăți

- ...
- Expandează obiectele din imagine.

Dilatarea - Proprietăți

- ...
- Expandează obiectele din imagine.
- Umple mici goluri.

Dilatarea - Proprietăți

- ...
- Expandează obiectele din imagine.
- Umple mici goluri.
- Reduce fragmentarea contururilor.

Dilatarea - Exemplu

1	1	1
1	1	1
1	1	1



Dilatare /Erodare -Observație

- În toate exemplele date s-a considerat că se erodează / dilată multimi de pixeli albi. DAR la fel de bine se poate consideră că multimile sunt de pixeli negri.
- Putem dilata/ eroda obiectele SAU fundalul - depinde de scop.

Dualitatea Dilatare - Erodare

Între dilatare și erodare există o relație de dualitate în raport cu complementara și relecția, adică:

$$C_{(A \ominus B)} = C_A \oplus \hat{B}$$

și

$$C_{(A \oplus B)} = C_A \ominus \hat{B}$$

Adică erodarea lui A cu B este complementul dilatării lui C_A cu \hat{B} și invers.

Dualitatea Dilatare - Erodare

Între dilatare și erodare există o relație de dualitate în raport cu complementara și relecția, adică:

$$C_{(A \ominus B)} = C_A \oplus \hat{B}$$

și

$$C_{(A \oplus B)} = C_A \ominus \hat{B}$$

Adică erodarea lui A cu B este complementul dilatării lui C_A cu \hat{B} și invers.

Observație: Atunci când elementul structurant B este simetric față de origine se poate de exemplu obține erodarea obiectului prin dilatarea fundalului.

Problemă

Doresc:

- Eliminare goluri, defragmentare contururi.

sau

- Eliminare zgomot, detalii nerelevante.

NU doresc:

Modificarea dimensiunii obiectelor!

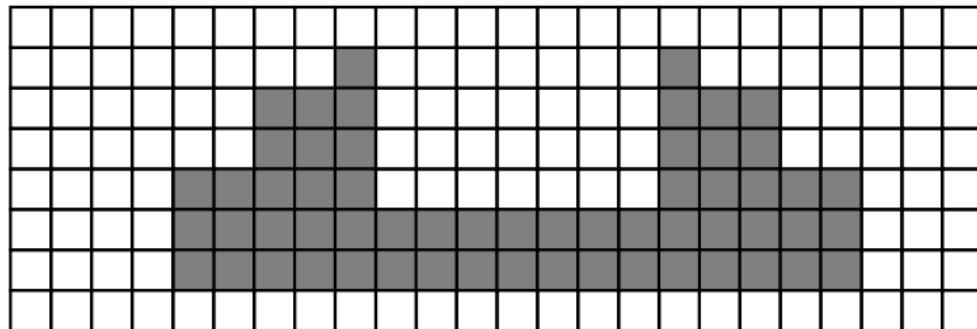
Ce fac?

Opening

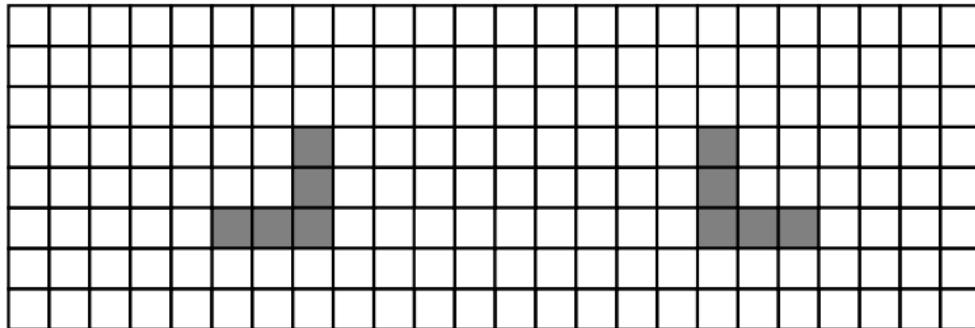
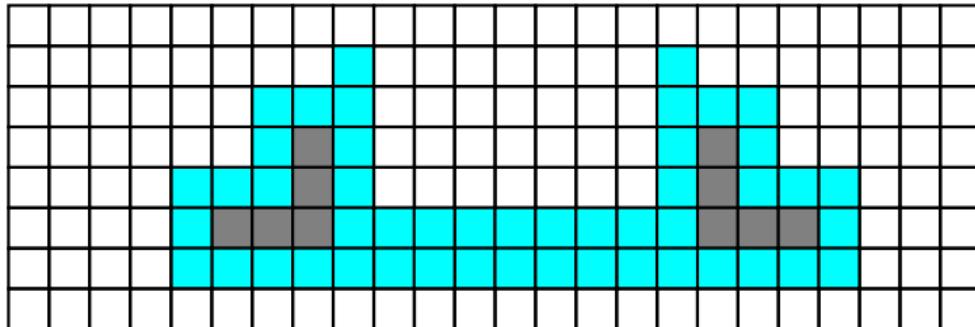
Definiție:

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

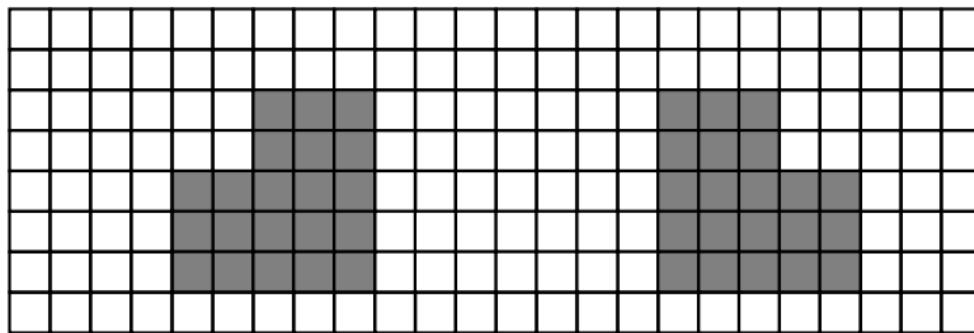
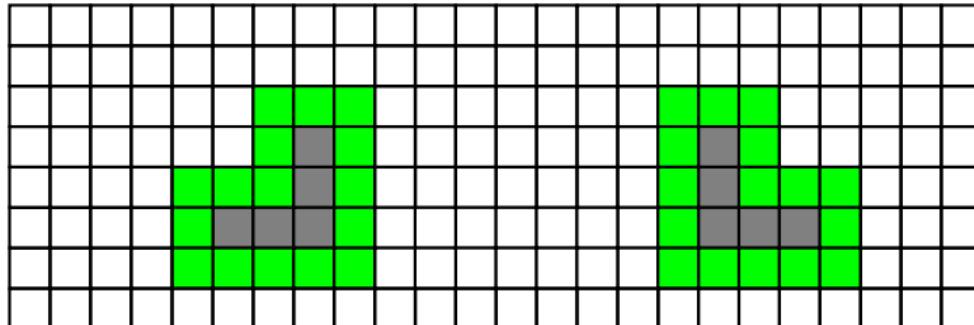
Opening - Exemplu



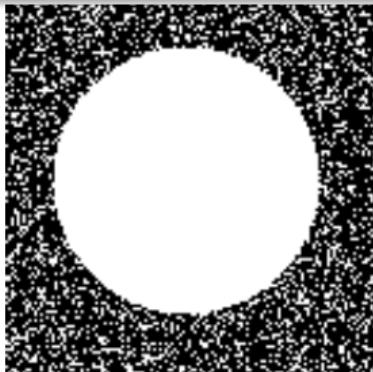
Opening - Exemplu



Opening - Exemplu



Opening



Proprietăți:

- Netezește contururile obiectelor.
- Separă obiectele legate prin "punți" subțiri.
- Elimină mici protuberanțe.



Opening

(1) $A \circ B \subseteq A$

(2) Dacă $C \subset D$, atunci $C \circ B \subset D \circ B$

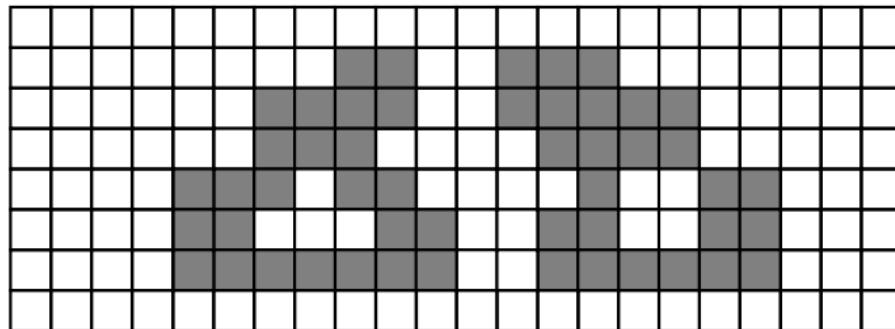
(3) $(A \circ B) \circ B = A \circ B$

Closing

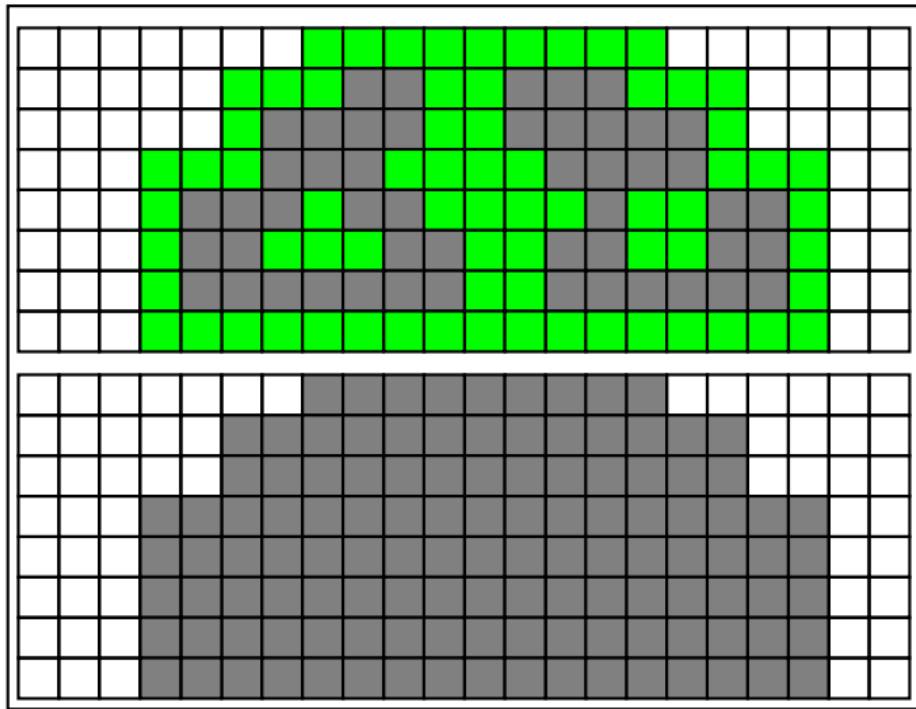
Definiție:

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

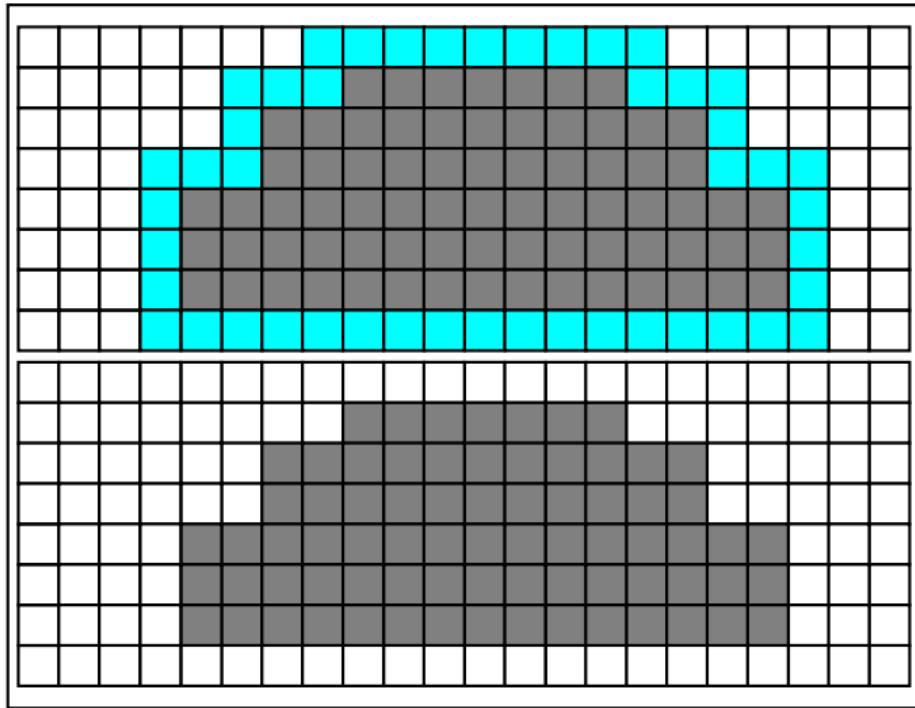
Closing - Exemplu



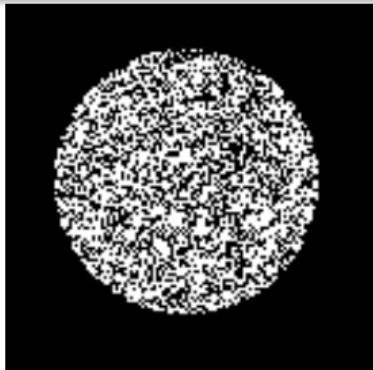
Closing - Exemplu



Closing - Exemplu

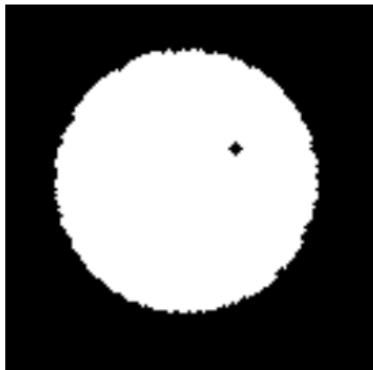


Closing



Proprietăți:

- Netezește contururile obiectelor.
- Umple goluri mici în obiecte și pe contururi.
- Unește obiecte aflate la distanță mică.



Closing

$$(1) \ A \subseteq A \bullet B$$

$$(2) \ \text{Dacă } C \subset D, \text{ atunci } C \bullet B \subset D \bullet B$$

$$(3) \ (A \bullet B) \bullet B = A \bullet B$$

Aplicații

- Îmbunătățirea imaginii binare prin:

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;
 - evidențierea anumitor trăsături (culori) - mai ales în cazul grayscale sau color

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;
 - evidențierea anumitor trăsături (culori) - mai ales în cazul grayscale sau color
 - efecte speciale (grayscale / color)

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;
 - evidențierea anumitor trăsături (culori) - mai ales în cazul grayscale sau color
 - efecte speciale (grayscale / color)
- Extragerea contururilor obiectelor.

Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;
 - evidențierea anumitor trăsături (culori) - mai ales în cazul grayscale sau color
 - efecte speciale (grayscale / color)
- Extragerea contururilor obiectelor.
- Extragerea de componente conexe.

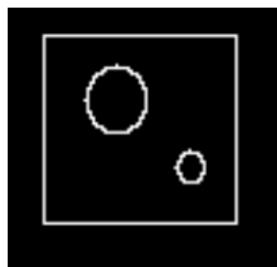
Aplicații

- Îmbunătățirea imaginii binare prin:
 - eliminarea de zgomot;
 - umplerea golurilor;
 - defragmentarea contururilor;
 - evidențierea anumitor trăsături (culori) - mai ales în cazul grayscale sau color
 - efecte speciale (grayscale / color)
- Extragerea contururilor obiectelor.
- Extragerea de componente conexe.
- etc.

Extragerea contururilor



Variante



Extragerea componentelor conexe

- Se poate realiza morfologic prin dilatare.
- Se consideră:
 - mulțimea componentelor conexe A ;
 - $Y \subset A$ componentă conexă;
 - $p \in Y$;
- Componenta Y se obține iterativ la convergența sirului Y_k :

$$Y_k = (Y_{k-1} \oplus B) \cap A, k = 1, 2, 3, \dots$$

$Y_0 = p$, B element structurant potrivit.

Extragerea componentelor conexe - Practic - Alg1

Notății: C - coadă de pixeli, p - pixel etichetat.

Algoritm: Inițial coada $C = \{p\}$

cât timp $C \neq \emptyset$

$q \Leftarrow C$

pentru q_i vecin al lui q , $i = \overline{1,8}$

dacă $q_i \in A$ și q_i nemarcat atunci

marchează q_i cu eticheta

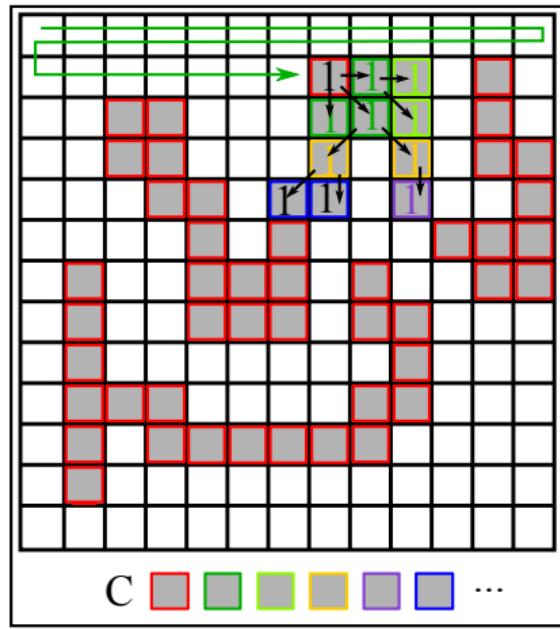
corespunzătoare lui q

$q_i \Rightarrow C$

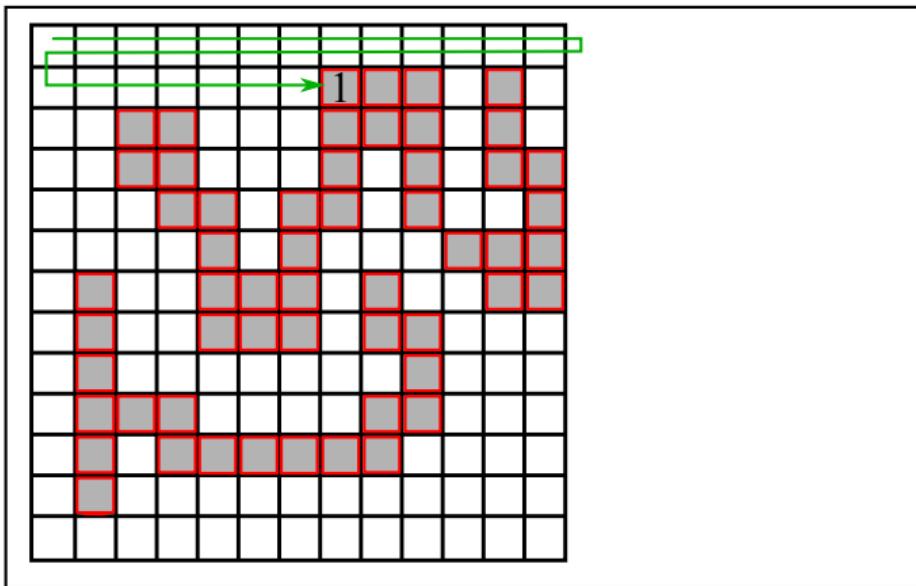
sfârșit dacă

sfârșit pentru

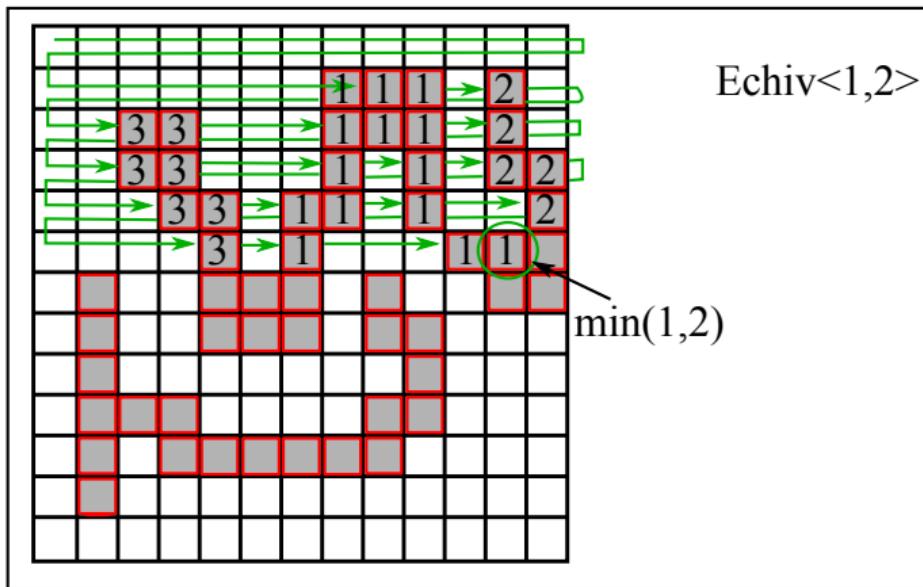
sfârșit cât timp



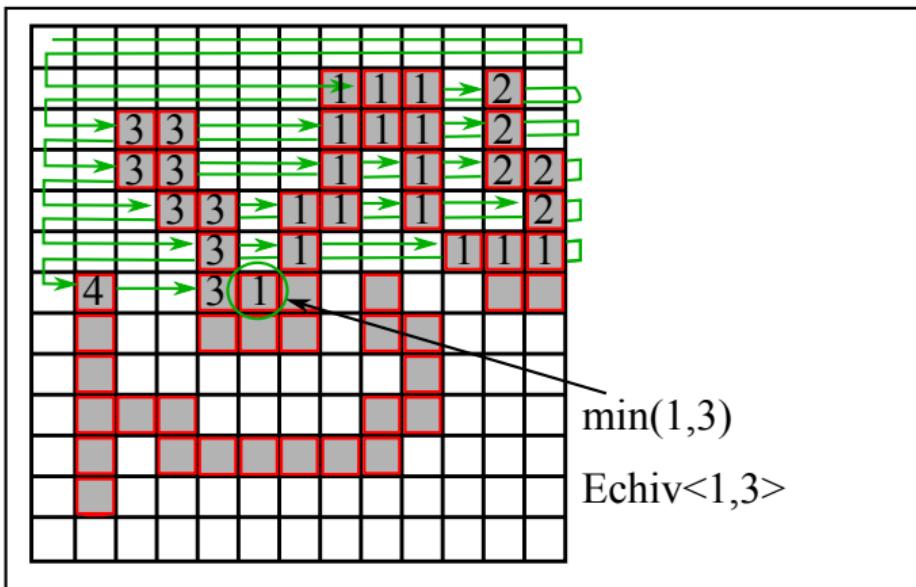
Extragerea componentelor conexe - Practic - Alg2



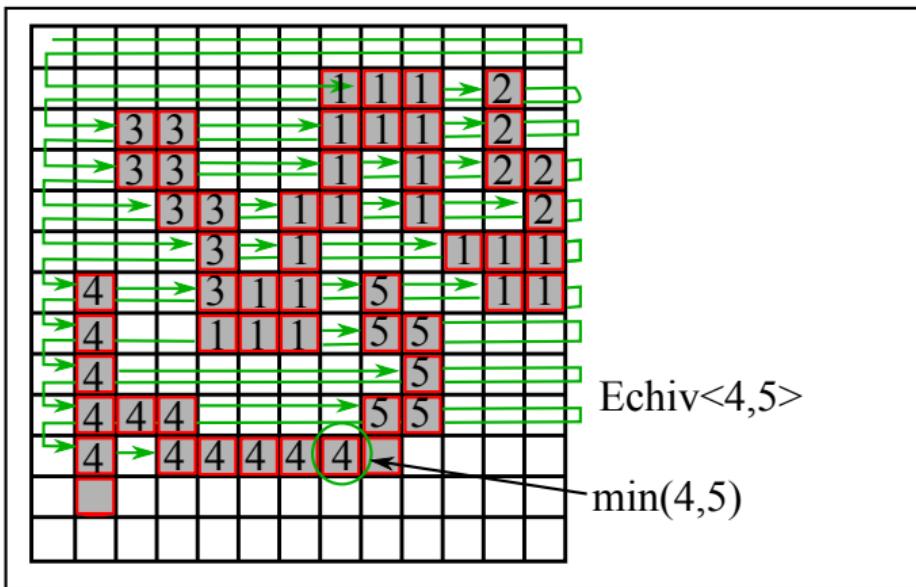
Extragerea componentelor conexe - Practic - Alg2



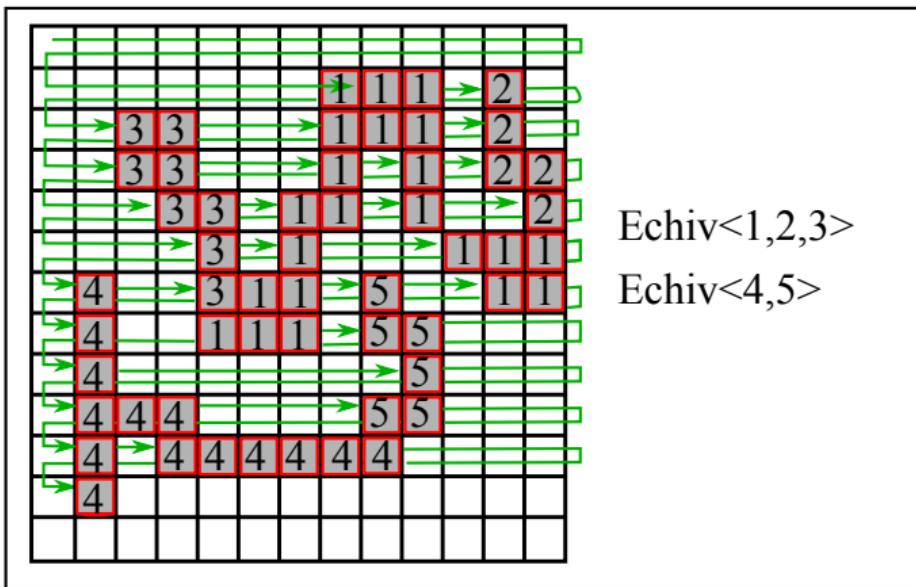
Extragerea componentelor conexe - Practic - Alg2



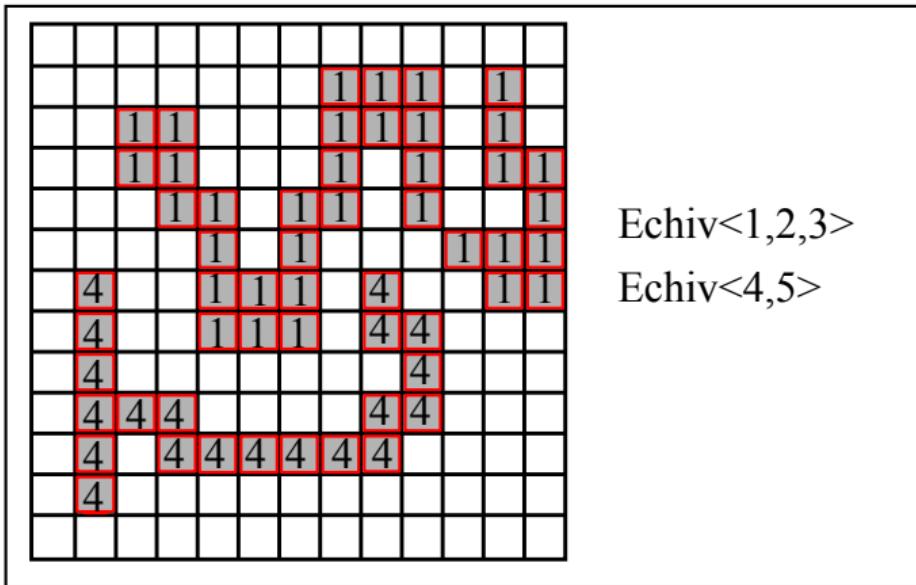
Extragerea componentelor conexe - Practic - Alg2



Extragerea componentelor conexe - Practic - Alg2



Extragerea componentelor conexe - Practic - Alg2



Păstrarea listelor de echivalență cu mulțimi disjuncte

Disjoint Set - structură de date $S = \{S_1, S_2, \dots, S_n\}$

- S_1, S_2, \dots, S_3 = mulțimi dinamice disjuncte
- fiecare S_i este identificat printr-un reprezentant

Operații:

- MAKE_SET(x) - crează o mulțime nouă în S care conține doar elementul x
- UNION(x, y) - reunește două mulțimi disjuncte S_x și S_y într-o nouă mulțime
- FIND_SET(x) - determină reprezentantul mulțimii care îl conține pe x

(Documentație: "Introduction to algorithms", T. Cormen at al.)

Păstrarea listelor de echivalență cu mulțimi disjuncte

Implementare

- Liste înănțuite - nu este foarte eficient (vezi Cormen)
- Pădure - mulțime de arbori

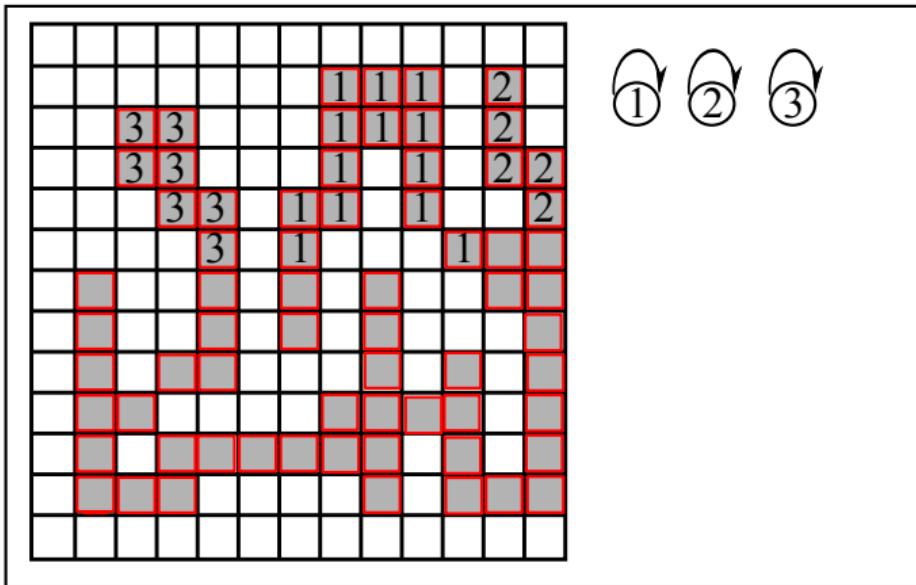
Păstrarea listelor de echivalență cu mulțimi disjuncte

Implementare Fiecare arbore este alcătuit din noduri - fiecare nod x :

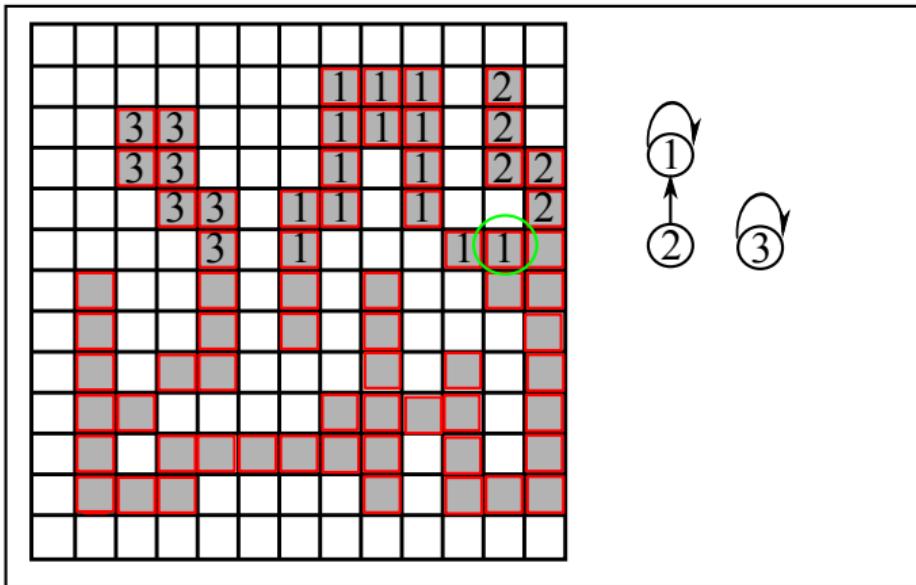
- - are un câmp eticheta
- - are un câmp de legătură către părinte.

Rădăcina - legată de ea însăși = reprezentantul mulțimii.

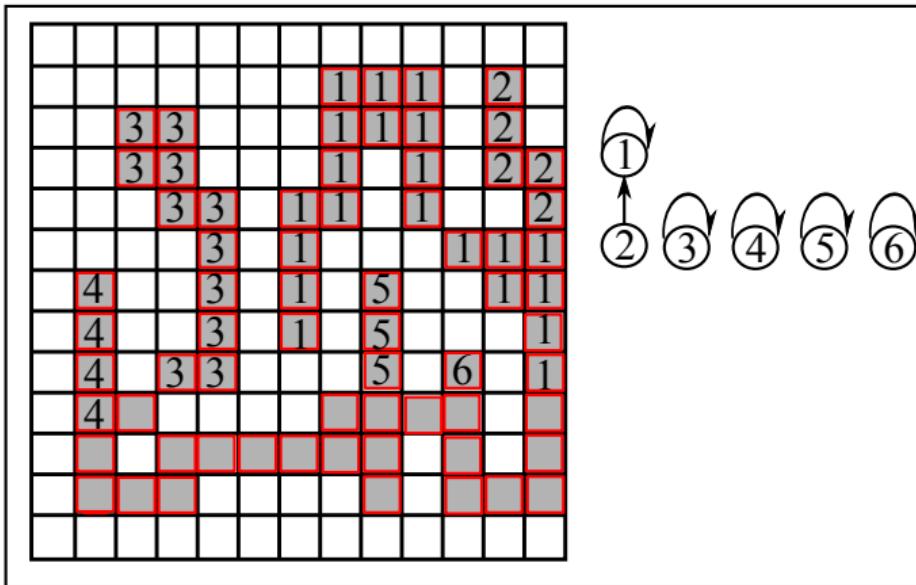
Extragerea componentelor conexe - Disjoint Set - Exemplu



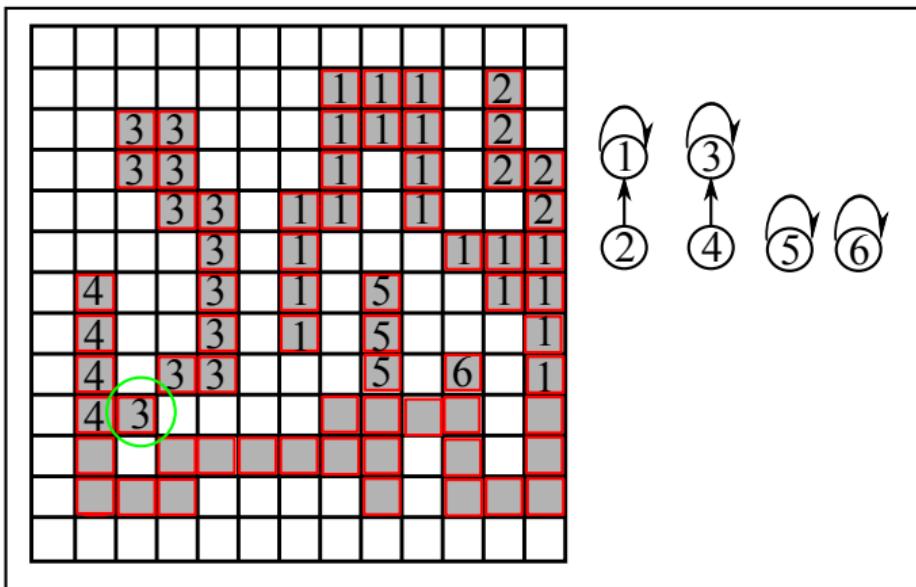
Extragerea componentelor conexe - Disjoint Set - Exemplu



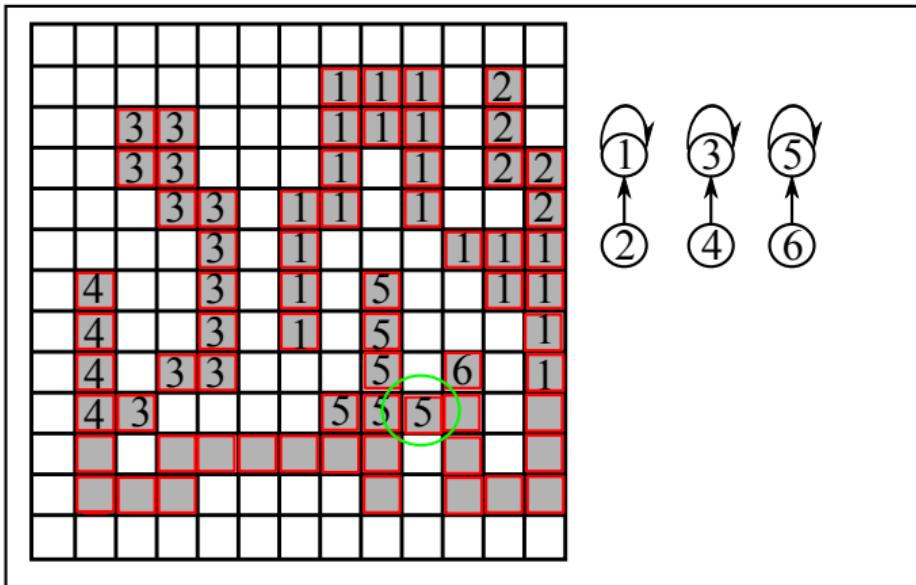
Extragerea componentelor conexe - Disjoint Set - Exemplu



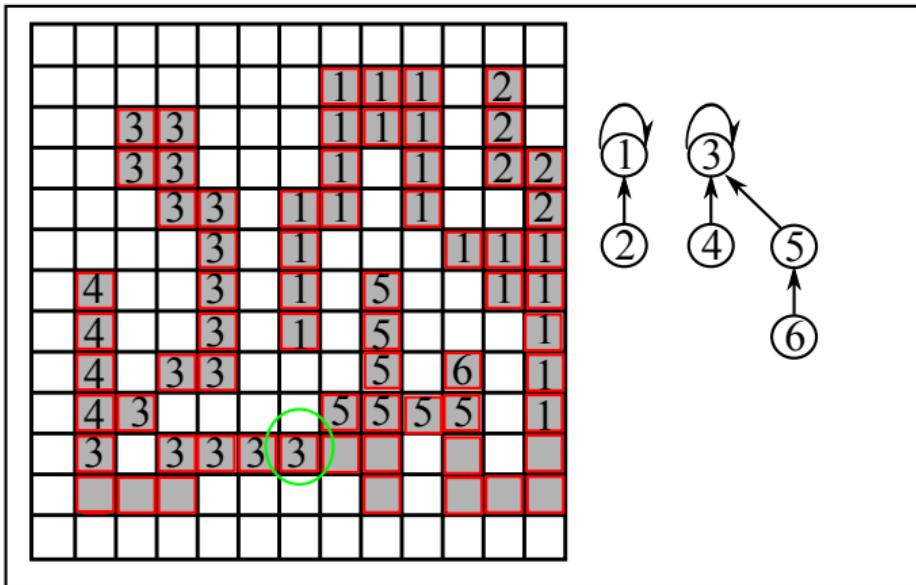
Extragerea componentelor conexe - Disjoint Set - Exemplu



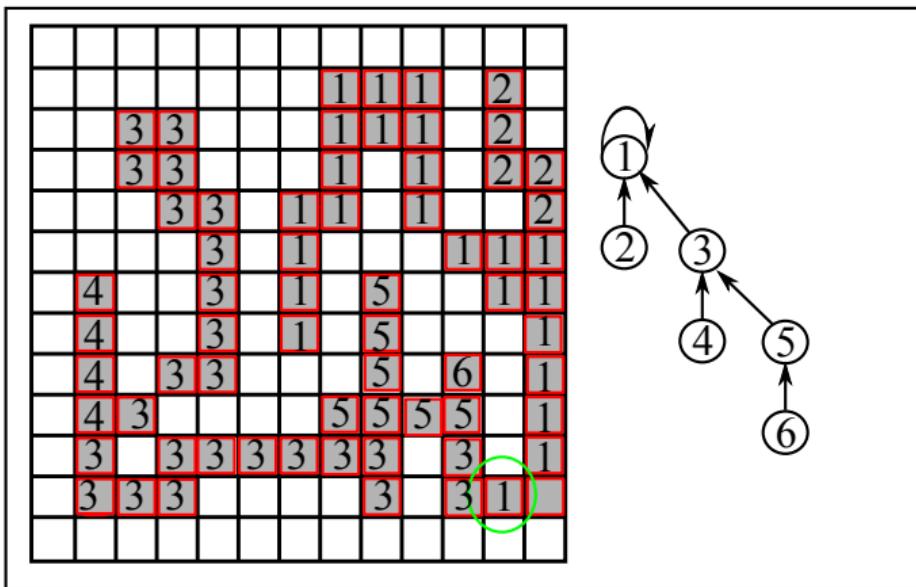
Extragerea componentelor conexe - Disjoint Set - Exemplu



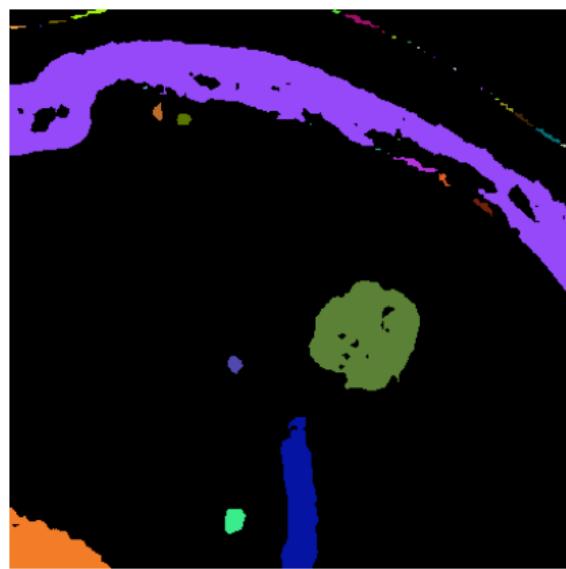
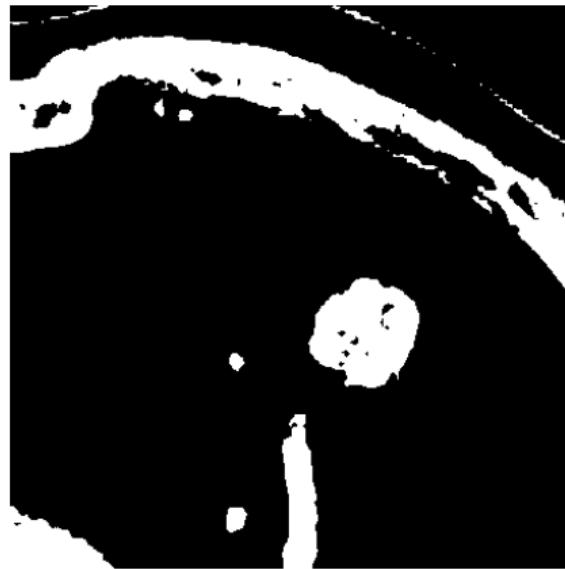
Extragerea componentelor conexe - Disjoint Set - Exemplu



Extragerea componentelor conexe - Disjoint Set - Exemplu



Extragerea componentelor conexe - Exemplu



Operații morfologice pentru imagini în tonuri de gri

0	0	0
0	0	0
0	0	0

Dilatarea imaginii $f(x, y)$ cu masca $b(s, t)$, $s = \overline{-k, k}$, $t = \overline{-k, k}$:

$$(f \oplus b)(x, y) = \max\{f(x - s, y - t) + b(s, t) |$$

$$s = \overline{-k, k}, t = \overline{-k, k}\}$$

0	1	0
1	2	1
0	1	0

Dilatare - Exemplu



Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Dilatare - Exemplu



Dilatare - Proprietăți

- (1) Dacă toate valorile din masca b sunt pozitive, imaginea rezultat este mai deschisă decât imaginea originală.
- (2) Detalii închise sunt reduse sau eliminate.

Operații morfologice pentru imagini în tonuri de gri

0	0	0
0	0	0
0	0	0

Erodarea imaginii $f(x, y)$ cu masca $b(s, t)$,
 $s = \overline{-k, k}$, $t = \overline{-k, k}$:

$$(f \ominus b)(x, y) = \min\{f(x - s, y - t) - b(s, t) |$$

$$s = \overline{-k, k}, t = \overline{-k, k}\}$$

0	1	0
1	2	1
0	1	0

Erodare - Exemplu



Concepțe de bază

Operații morfologice pentru imagini binare

Operații morfologice pe imagini în tonuri de gri

Operații morfologice pe imagini color

Erodare - Exemplu



Erodare - Proprietăți

- (1) Dacă toate valorile din masca b sunt pozitive, imaginea rezultat este mai închisă decât imaginea originală.
- (2) Detalii deschise sunt reduse sau eliminate.

Opening și Closing

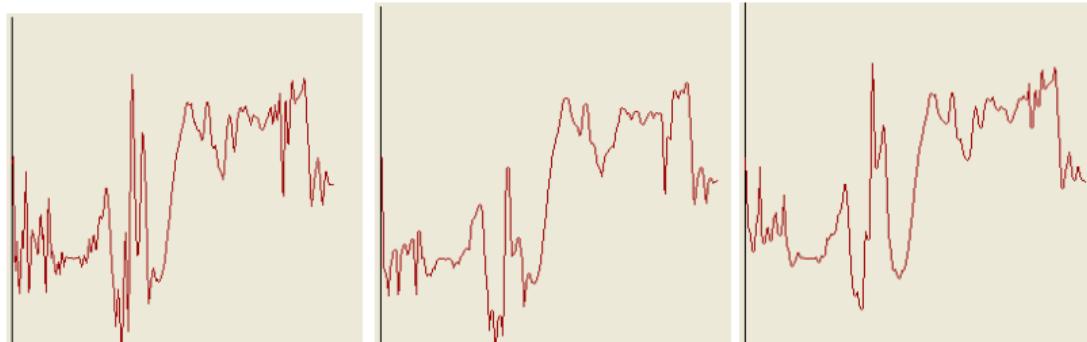
Opening:

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

Closing:

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

Opening și Closing - Exemple



Opening

Closing

Opening și Closing - Aplicații

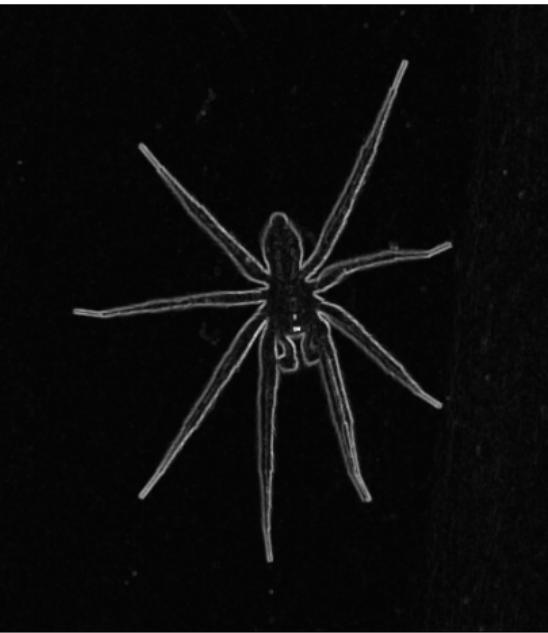
- **Opening:** - elimină mici detaliu deschise, păstrând nivelul mediu de gri și detaliu deschise mai mari neschimbate.
- **Closing:** - elimină mici detaliu de culoare închisă.

Netezire morfologică



Procedeu: $g = (f \circ b) \bullet b$

Gradient morfologic



Procedeu: $g = (f \oplus b) - (f \ominus b)$

Operații morfologice pentru imagini color

Observații:

- Operațiile morfologice au la bază operații de minim / maxim

Operații morfologice pentru imagini color

Observații:

- Operațiile morfologice au la bază operații de minim / maxim
- Culoarea este un vector cu 3 componente

Operații morfologice pentru imagini color

Observații:

- Operațiile morfologice au la bază operații de minim / maxim
- Culoarea este un vector cu 3 componente

⇒ este necesară definirea unei ordonări în spațiul de culoare

Operații morfologice pentru imagini color

- **Metode marginale (*M-ordering*):** se consideră separat fiecare componentă. Tind să producă culori false (inexistente în imaginea originală).

Operații morfologice pentru imagini color

- **Metode marginale (*M-ordering*)**: se consideră separat fiecare componentă. Tind să producă culori false (inexistente în imaginea originală).
- **Metode de ordonare condițională (*C-ordering*)**: vectorii sunt ordonați pe baza uneia / unora dintre componente și a unor condiții de selectare în cadrul setului de date. Exemple:

Operații morfologice pentru imagini color

- **Metode marginale (*M-ordering*)**: se consideră separat fiecare componentă. Tind să producă culori false (inexistente în imaginea originală).
- **Metode de ordonare condițională (*C-ordering*)**: vectorii sunt ordonați pe baza uneia / unora dintre componente și a unor condiții de selectare în cadrul setului de date. Exemple:
 - **Ordonarea lexicografică**. Are dezavantajul că atribuie o prioritate crescută anumitor componente ale vectorului și ar trebui utilizată doar atunci, când acest lucru este dezirabil.

Operații morfologice pentru imagini color

- **Metode marginale (*M-ordering*)**: se consideră separat fiecare componentă. Tind să producă culori false (inexistente în imaginea originală).
- **Metode de ordonare condițională (*C-ordering*)**: vectorii sunt ordonați pe baza uneia / unora dintre componente și a unor condiții de selectare în cadrul setului de date. Exemple:
 - **Ordonarea lexicografică**. Are dezavantajul că atribuie o prioritate crescută anumitor componente ale vectorului și ar trebui utilizată doar atunci, când acest lucru este dezirabil.
 - **Ordonarea în spațiul HSV**, în care se ignoră componenta de culoare *Hue*.

Operații morfologice pentru imagini color

- Metode de ordonare parțială (*P-ordering*), care partiționează setul de vectori în clase de echivalentă relativ la o anumită ordonare.

Operații morfologice pentru imagini color

- **Metode de ordonare parțială (*P-ordering*)**, care partiționează setul de vectori în clase de echivalență relativ la o anumită ordonare.
- **Metode de ordonare redusă (*R-ordering*)**. Presupun o metodă de asociere pentru fiecare vector a unei valori scalare. De fapt se încearcă determinarea unei funcții $f : \mathbb{R}^n \leftarrow \mathbb{R}$.

În cazul spațiului RGB, $n = 3$, putem asocia unui vector culoare distanța față de un element de referință, de exemplu originea sistemului de coordonate. Dacă funcția f să nu fie injectivă, este vorba de o *pre-ordonare*

Operații morfologice cu ordonarea culorilor relativ la origine

