Operaţii morfologice

1 Introducere

Morfologia matematică reprezintă un instrument pentru extragerea de componente ale imaginii precum contururi sau schelete, utile în reprezentarea și descrierea de obiecte respectiv regiuni. De asemenea operațiile morfologice pot fi utilizate în tehnici de preprocesare a imaginii.
La baza morfologiei matematice se află teoria mulțimilor și operațiile logice. În cazul imaginilor, mulțimile se referă la mulțimi de pixeli cu o anumită proprietate, de exemplu pixelii albi dintr-o imagine binară.
Operațiile morfologice reprezintă tranziția de la operații ale căror intrare și ieșire sunt imagini la operații care la intrare au o imagine iar la ieșire un set de date, deci fac tranziția către procesare de nivel mediu reprezentată în principal prin segmentare.

2 Competențe conferite

La sfârșitul acestui capitol, studenții vor fi capabili să înțeleagă:

- Operații logice pe imagini binare.
- Ce este dilatarea / erodarea unui obiect binar.
- Ce este operația de opening / closing
- Cum se determină componentele conexe într-o imagine binară.
- Ce sunt operațiile morfologice pe imagini în tonuri de gri.
- Ce este netezirea morfologică.
- Ce este gradientul morfologic.
- Cum se realizează operații morfologice pe imagini color.

3 Durata medie de studiu individual

Parcurgerea de către studenți a acestei unități de învățare se face în 2 ore.

4 Operații morfologice pentru imagini binare

În prima parte a acestui curs vom discuta despre operații morfologice pe imagini binare. În acestea obiectele vor fi considerate mulțimile conexe de pixeli albi, iar fundalul va fi reprezentat de pixelii negri. Mulțimile de pixeli albi vor fi considerate ca submulțimi din $\mathbb{Z}^2$, adică mulțimi de perechi de coordonate (x, y).

Notații din teoria mulțimilor (considerate din $\mathbb{Z}^2$) :

Alături de notațiile standard din teoria mulțimilor precum operațiile de reuniune, intersecție și apartenență bine cunoscute, considerăm următoarele notații:

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

Operații logice cu imagini binare
Imaginile binare sunt imagini în care există doar pixeli albi și negri. Se poate considera valoarea 1 pentru un pixel alb și 0 pentru un pixel negru (sau invers - în funcție de convenție). Astfel pot fi definite operații logice între imagini similar cu operațiile logice uzuale. În tabela 1 sunt prezentate operațiile logice utilizate iar în figura 1 sunt ilustrate rezultatele operațiilor logice asupra a două imagini binare A și B.

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 1: Operațiile logice NOT, AND, OR și XOR

4.1 Dilatarea

Definiție: Considerând două mulțimi A și B în $\mathbb{Z}^2$ *dilatarea* mulțimii A cu B este:

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

Mulțimea $A \oplus B$ obținută prin dilatare reprezintă de fapt mulțimea tuturor punctelor z pentru care translația cu z simetricului lui B față de origine are cel puțin un punct comun cu A.

B se numește *element structurant* și este o mulțime de coordonate centrate în origine. De obicei B este simetrică față de origine și din acest motiv vom considera în continuare$\hat{B} =$

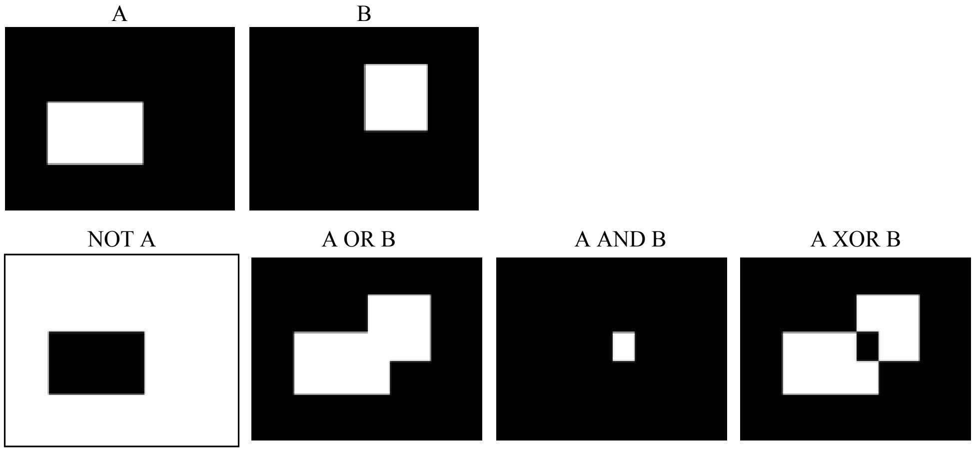
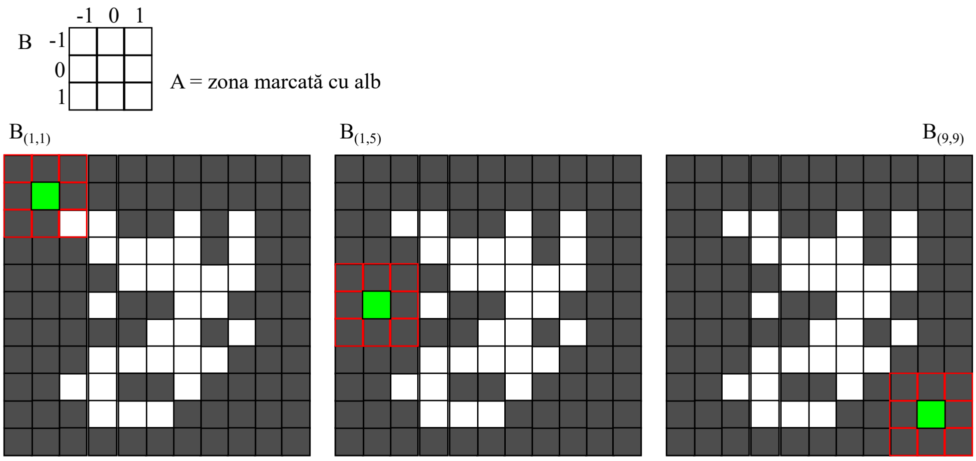


Figura 1: Imaginile binare A și B împreună cu imaginile rezultate prin aplicarea operațiilor logice NOT, AND, OR și XOR asupra acestor imagini.

B.

Exemplu: de poziționare a unui element structurant B de dimensiune 3×3 asupra unei imagini binare pentru dilatare. În imaginea din stânga B este translatat pe poziția $(1, 1)$. În vecinătatea definită de $B(1, 1)$ se află un pixel alb, deci în imaginea rezultat, pixelul colorat cu verde, care aparține fundalului, va deveni alb. În imaginea centrală B se translatează pe poziția $(1, 5)$, iar vecinătatea definită de B este complet încadrată în zona de fundal, deci pixelul central (verde) va rămâne și în imaginea dilatată tot negru. Același lucru se întâmplă și în cazul figurii din dreapta.



Considerând un element structurant B de digresiune 3×3 centrat în origine, precum cel din exemplu, dilatarea unei imagini $f(x, y)$ cu acest element produce imaginea binară

$g(x, y)$:

$$g(x,y) = \begin{cases} 255, & \text{dacă } f(x,y) = 255 \\ 255, & \text{dacă } \exists (x_1, y_1) \in B_{xy} \text{ a. î } f(x_1,y_1) = 255 \\ 0 & \text{altfel} \end{cases}$$

$$B_{xy} = \{(x-1,y-1),(x,y-1),(x+1,y-1),(x-1,y),(x,y),(x+1,y),(x-1,y+1),$$
$$(x,y+1),(x+1,y+1)\}$$

unde prin mulțimea B_{xy} am reprezentat mulțimea coordonatelor vecinătății pixelului (x, y) din imaginea sursă f.

Observații:

- Din formula de mai sus, se observă faptul că pixelii care în imaginea inițială erau albi, rămân și în imaginea rezultat tot albi. Deci, la aplicarea operației, ne interesează doar pixelii negri.

- În exemple s-a considerat că, mulțimea care se dilată este mulțimea pixelilor albi, dar evident că, se poate considera și altă mulțime de pixeli, de exemplu mulțimea pixelilor negri, sau orice altă mulțime de pixeli (reprezentați prin perechi de coordonate)

Algoritmic: pentru fiecare pixel negru (x, y), dacă în vecinătatea sa, determinată de elementul structurant centrat pe poziția (x, y) există cel puțin un pixel alb ⇒ în imaginea rezultat pixelul (x, y) va fi alb.

Algoritm: Dilatarea (mulțimii de pixeli albi)

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
Un element structurant nu are neapărat o formă pătrată. Alte exemple de elemente structurante sunt prezentate în figura 2. Pentru simplificarea implementării elementele structurante pot fi considerate reprezentate prin măști dreptunghiulare în care pozițiile aparținând elementelor structurante sunt marcate prin 1.

Dilatarea poate fi utilizată de exemplu pentru umplerea de mici goluri, cum ar fi cele care apar la scanarea unui text sau prin zgomot prezent înainte de binarizare.

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

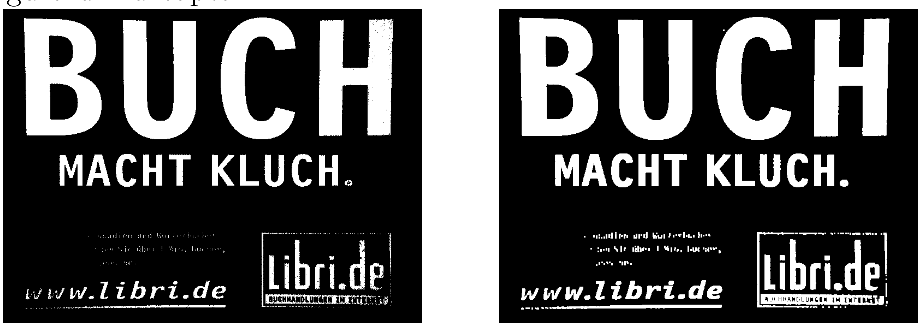
a)

0	1	0
1	1	1
0	1	0

b)

Figura 2: a) Element structurant dreptunghiular, simetric față de origine. b) Element structurant aproximativ circular.

Exemplu: Dilatarea unei imagini reprezentând un text (stânga) cu un element structurant pătrat de dimensiune 3×3. Rezultatul este prezentat în figura din dreapta.



Proprietăți ale dilatării. Dilatarea obiectelor (albe) dintr-o imagine binară are următoarele efecte:

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

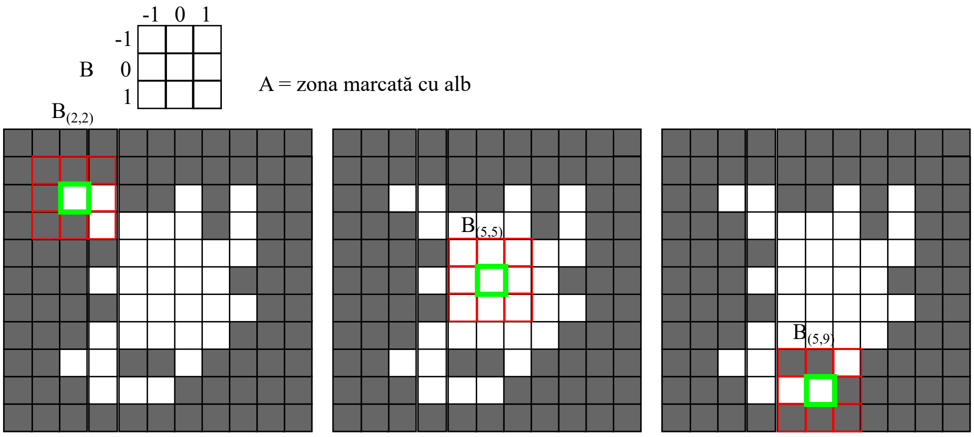
4.2 Erodarea

Definiție Considerând două mulțimi A și B în $\mathbb{Z}^2$ *erodarea* mulțimii A cu B este:

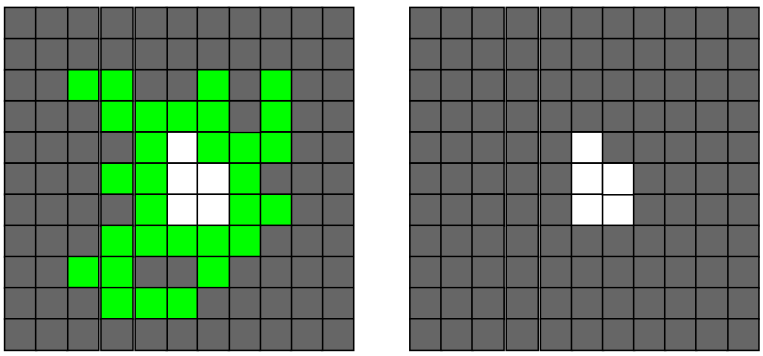
$$A \ominus B = \{z | (B)_z \subseteq A\}$$

Mulțimea $A \ominus B$ obținută prin erodare reprezintă de fapt mulțimea tuturor punctelor z pentru care translația lui B cu z este inclusă în A. B este un element structurant similar ca pentru dilatare.

Exemplu: de poziționare a unui element structurant B de dimensiune 3×3 asupra unei imagini binare pentru erodare. În imaginea din stânga B este translatat pe poziția $(2, 2)$, centrat pe un pixel alb. În vecinătatea definită de $B(2, 2)$ se află pixeli negri, deci în imaginea rezultat, pixelul colorat cu verde, care aparține obiectului, va deveni negru. În imaginea centrală B se translatează pe poziția $(5, 5)$, iar vecinătatea definită de B este complet încadrată în obiect, deci pixelul central (verde) va rămâne și în imaginea erodată tot alb. În imaginea din dreapta elementul structurant este translatat pe poziția $(5, 9)$, centrat pe un pixel alb, dar în vecinătatea definită de $B(5, 9)$ se găsesc și pixeli negri, deci pixelul central va fi negru în imaginea rezultat.



Rezultatul erodării figurii este ilustrată mai jos. În stânga este obiectul original în care sunt marcați cu verde pixelii ce vor fi erodați, iar în figura din dreapta este prezentat rezultatul erodării.

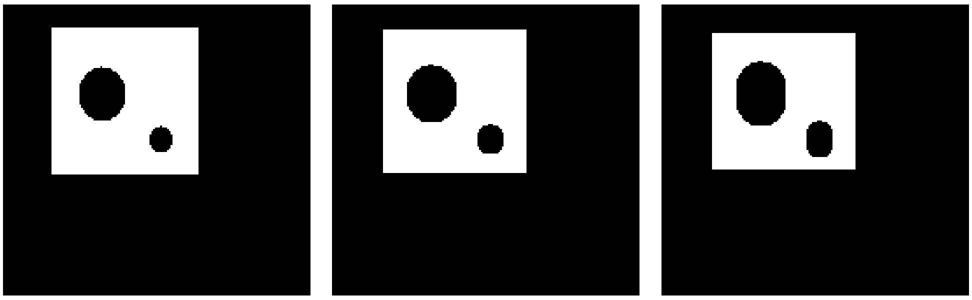


Operația de erodare poate fi descrisă prin algoritmul următor, în care f reprezintă imaginea sursă binară, g imaginea binară rezultat, iar B elementul structurant.

Algoritm: Erodarea (mulțimii de pixeli albi)

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

Exemplu: Un exemplu de erodare a unei imagini binare cu elementele structurante din fig. 2 a) și fig. 2 b) este prezentată în figura de mai jos, în care imaginea (a) este imaginea originală, imaginea (b) este rezultatul erodării cu elementul structurant pătrat, iar imaginea (c) este rezultatul erodării cu elementul dreptunghiular.



a) b) c)

Proprietăți ale erodării. Erodarea obiectelor (albe) dintr-o imagine binară are următoarele efecte:

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

Observații:

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

Dualitatea Erodare - Dilatare: Între dilatare și erodare există o relație de dualitate în raport cu complementara și reflecția:

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

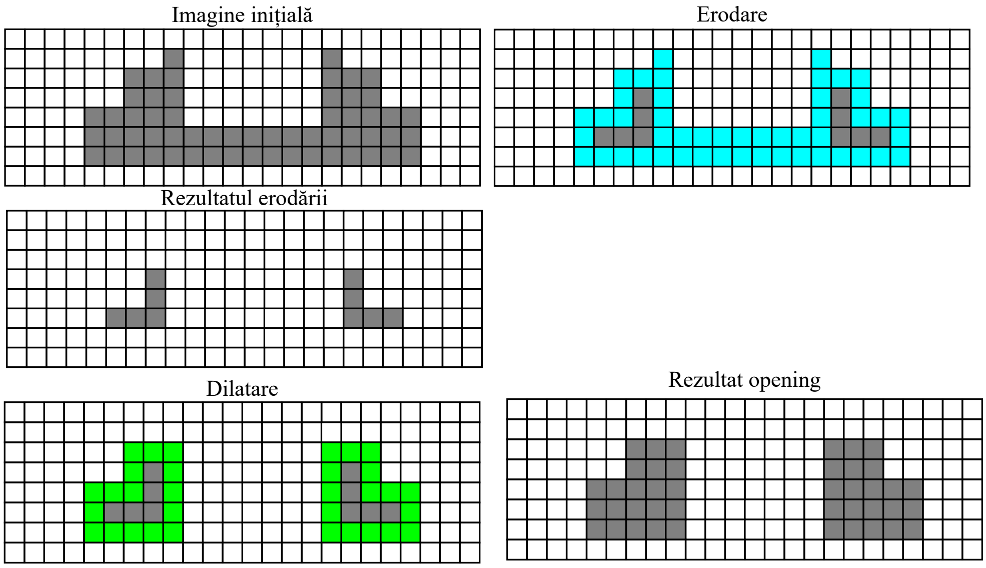
și
$$C_{(A \oplus B)} = C_A \ominus \hat{B}$$
Adică erodarea lui A cu B este complementul dilatării lui C_A cu $\hat{B}$ și invers. Acest lucru are drept consecință faptul că, atunci când elementul structurant B este simetric față de origine se poate de exemplu obține erodarea obiectului prin dilatarea fundalului. În mod similar se întâmplă și în cazul dilatării unui obiect.

4.3 Opening și closing

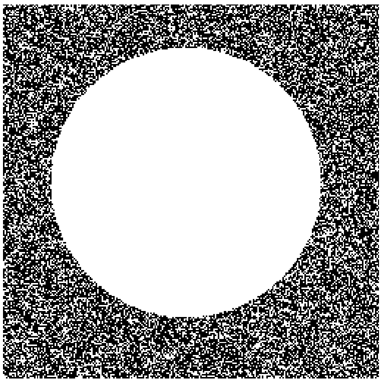
Cele două operații morfologice discutate mai sus pot fi combinate pentru a obține operații mai complexe.

Operația de Opening: Considerând o mulțime A și un element structurant B, operația definită prin:
$$A \circ B = (A \ominus B) \oplus B$$
se numește *opening* al mulțimii A prin elementul structurant B. În următorul exemplu este ilustrat rezultatul unei operații de opening asupra unei imagini binare conținând un disc alb pe un fundal negru, în care fundalul este caracterizat prin zgomot.

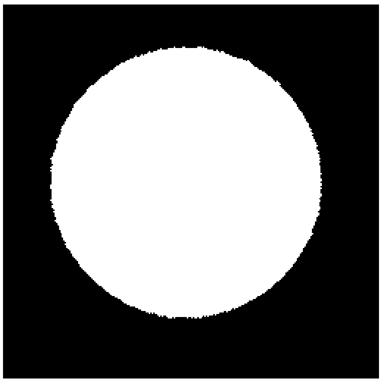
Exemplu: Aplicarea operației de opening asupra unei imagini binare. Se consideră pixelii gri ca reprezentând obiectul și pixelii albi ca fundal.



Exemplu: Aplicarea operației de opening asupra unei imagini binare (a) cu un element structurant pătrat de dimensiune 3×3 produce rezultatul din (b).



a)



b)

Observații: Operația de opening

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

Proprietăți ale operației de opening

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

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

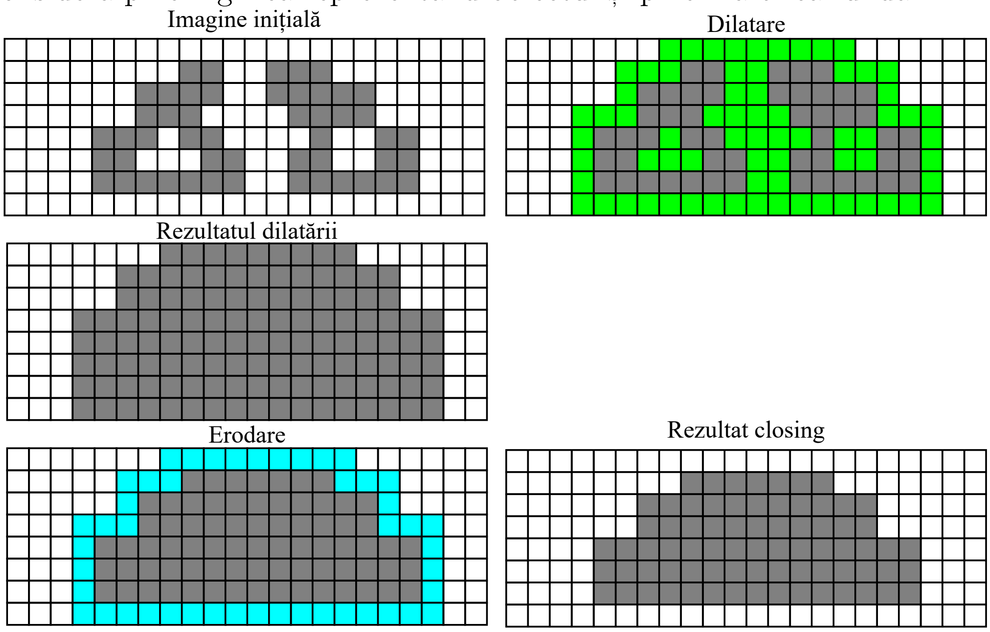
(3) $(A \circ B) \circ B = A \circ B$

Operația de Closing: Considerând o mulțime A și un element structurant B, operația definită prin:

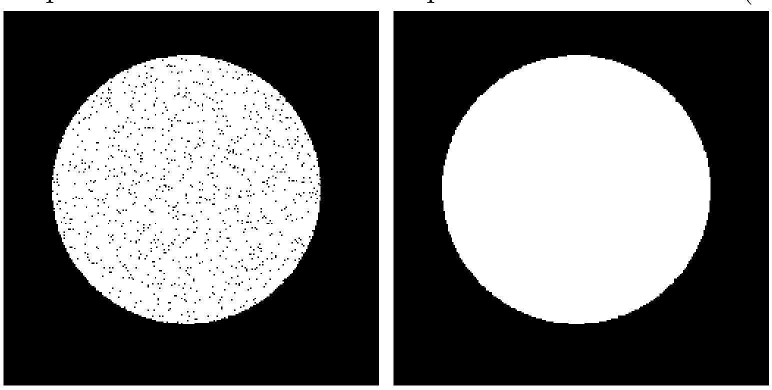
$$A \bullet B = (A \oplus B) \ominus B$$

se numește *closing* al mulțimii A prin elementul structurant B. În exemplul următor este ilustrat rezultatul unei operații de opening asupra unei imagini binare conținând un disc alb pe un fundal negru, în care discul este caracterizat prin zgomot.

Exemplu: Aplicarea operației de closing asupra unei imagini binare. Se consideră pixelii gri ca reprezentând obiectul și pixelii albi ca fundal.



Exemplu: Aplicarea operației de closing asupra imaginii (a) cu un element structurant pătrat de dimensiune 3×3 produce rezultatul din (b).



a) b)

Observații: Operația de closing

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

Proprietăți ale operației de closing

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

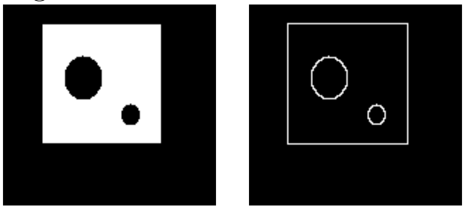
(2) Dacă $C \subset D$, atunci $C \bullet B \subset D \bullet B$

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

4.4 Aplicații ale operațiilor morfologice

Principalele aplicații ale operațiilor morfologice asupra imaginilor binare constau în extragerea de componente conexe, și determinarea de contururi ale obiectelor sau de schelete utile pentru reprezentarea și descrierea obiectelor. Pot fi de asemenea utilizate pentru îmbunătățirea imaginii binare, prin eliminarea de zgomot, umplerea golurilor sau defragmentarea contururilor. În continuare vor fi prezentate câteva dintre cele mai utilizate operații morfologice.

Exemplu: de detectare de contur folosind operația de XOR între imaginea binară originală și imaginea dilatată.



4.4.1 Extragerea componentelor conexe

Extragerea componentelor conexe dintr-o imagine binară poate fi realizată cu ajutorul operației de dilatare, pornind de la câte un punct al fiecărei componente, în modul următor. Considerând o imagine binară, mulțimea componentelor conexe notată prin A și un punct p aparținând componentei conexe Y, mulțimea de puncte din A aparținând componentei Y se obține iterativ prin:

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

unde $Y_0 = p$ și B element structurant potrivit. Algoritmul converge când $Y = Y_k = Y_{k-1}$.

Din punct de vedere practic, componenta conexă Y poate fi obținută pornind de la punctul p și în modul următor. Se consideră o coadă C, inițial conținând doar punctul p și o etichetă (de exemplu un număr întreg) corespunzătoare componentei conexe. p este marcat cu eticheta respectivă.

`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`

Exemplu: de detectare de componente conexe într-o imagine binară . În stânga este reprezentată imaginea originală, iar în dreapta componentele conexe reprezentate prin culori diferite.

a) b)

Un algoritm mai eficient din punct de vedere al complexității este acela care folosește seturi disjuncte [?] și este prezentat pe larg în [?].

4.4.2 Extragerea contururilor

Cu ajutorul operațiilor morfologice poate fi realizată extragerea contururilor obiectelor dintr-o imagine binară, de exemplu prin scăderea imagnii erodate din imaginea originală. Operația corespunzătoare este dată prin:

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

unde A reprezintă mulțimea obiectelor din imaginea binară, iar B un element structurant pătrat de dimensiune 3×3.

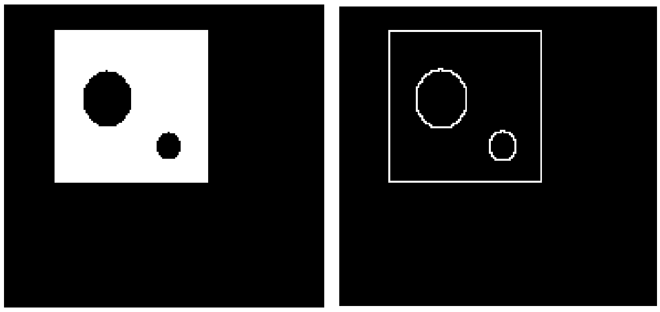
Extragerea conturului poate fi realizată și prin operația de XOR între imaginea binară conținând obiectele originale și imaginea binară conținând obiectele erodate.

$$contur(A) = A \text{ XOR } (A \ominus B))$$

În ambele cazuri conturul obținut este alcătuit din pixeli aparținând obiectului din imagine, dat fiind că obiectul binar erodat este mai mic decât cel original.

Dacă se realizează diferența sau operația de XOR între imaginea dilatată și imaginea originală se obține de asemenea un contur al obiectului, dar aflat pe exteriorului obiectului original.

Exemplu: exemplu de extragere a contururilor utilizând operația de XOR între imaginea originală și imaginea erodată.



a) b)

5 Operații morfologice pentru imagini în tonuri de gri

În cazul imaginilor în tonuri de gri, elementele structurante folosite pentru dilatare respectiv erodare sunt de două tipuri. și anume

(1) Elemente structurante care doar definesc o vecinătate a pixelului procesat, la fel ca în cazul operațiilor morfologice pentru imagini binare, numite elemente structurante plate - *flat structuring elements*

(2) Elemente structurante reprezentate de măc sti ce conțin coeficienți care contribuie la modificare imaginii în tonuri de gri. Acestea se numesc elemente structurante ne-plate -*nonflat structuring elements*

5.1 Dilatare și erodare

În cazul elementelor structurante plate operațiile de dilatare, respectiv erodare se definesc în modul următor:

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

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

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

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

În cazul elementelor structurante ne-plate, cele două operații devin:

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

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

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

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

Considerăm în acest caz toate valorile din mască pozitive. De obicei măștile de acest fel sunt simetrice relativ la toate axele, pentru ca vecinii aflați la aceeași distanță față de pixelul central să contribuie în aceeași măsură la rezultat.

Exemplu: de dilatare a unei imagini în tonuri de gri cu un element structurant plat.

Exemplu: de erodare a unei imagini în tonuri de gri cu un element structurant plat.

Proprietăți

1. **Dilatarea**:
    - Dacă toate valorile din masca b sunt pozitive, imaginea rezultat este mai deschisă decât imaginea originală.
    - Detalii închise sunt reduse sau eliminate.
2. **Erodarea**
    - Dacă toate valorile din masca b sunt pozitive, imaginea rezultat est mai închisă decât imaginea originală.
    - Detalii deschise sunt reduse sau eliminate.

5.2 Opening și closing

Operațiile de opening și closing pentru imagini în tonuri de gri se definesc în mod similar ca cele pentru imagini binare, ținând desigur cont de definițiile operațiilor de dilatare, repectiv erodare. Astfel:

- **Opening:** $f \circ b = (f \ominus b) \oplus b$
- **Closing:** $f \bullet b = (f \oplus b) \ominus b$

Operația de opening elimină mici detalii deschise, respectiv operația de closing elimină mici detalii închise, păstrând nivelul mediu de gri și detalii deschise mai mari neschimbate. Efectul acestor operații asupra imaginilor binare este ilustrat în figura 3 În primul rând din figură sunt reprezentate imaginea originală, rezultatul operației de opening și rezultatul operației de closing. În a rândul al doilea este reprezentată curba de variație a nivelului de gri pe aceeași linie pentru fiecare dintre cele trei imagini. Pe aceste curbe se observă faptul că, pentru imaginea rezultată după operația de opening, s-au redus maximele locale aparținând unor regiuni mici de culoare deschisă, iar minimele locale au rămas aproape nemodificate. În cazul imaginii rezultate din closing efectul este exact invers. Minimele locale corespunzătoare unor regiuni reduse (ca suprafață) din imagine sunt afectate, pe când regiunile deschise rămân practic nemodificate.

5.3 Aplicații ale operațiilor morfologice pentru imagini în tonuri de gri

Operațiile morfologice pentru imagini în tonuri de gri pot fi folosite pentru a construi diferiți algoritmi, precum *netezirea morfologică* [**?**] sau *gradientul morfologic* [**?**], pe care îi vom prezenta în continuare.

Netezirea morfologică

Netezirea morfologică se referă la un algoritm care combină operațiile de opening și closing pentru a obține o netezire a imaginii. Întâi se eliminară detaliile deschise minore, considerate zgomot, prin opening, iar pe imaginea rezultată se aplică operația de closing

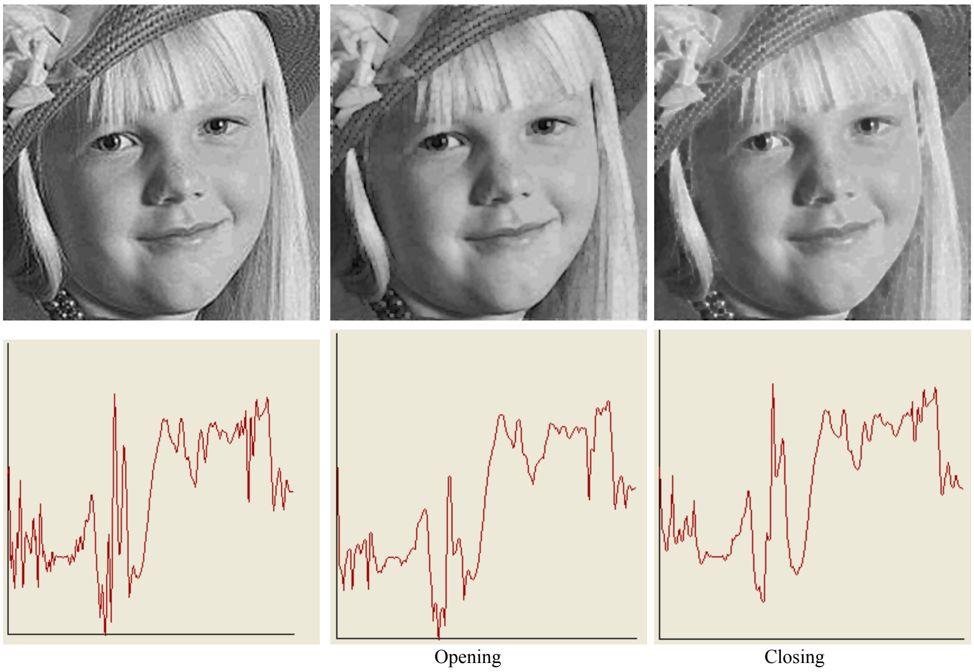


Figura 3: Efectele operațiilor de opening și closing asupra unei imagini în tonuri de gri, împreună cu curbele valorilor de gri pe un același rînd din fiecare imagine reprezentată.

cu aceeași mască utilizată pentru opening. Formula matematică corespunzătoare poate fi exprimată prin [?]:

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

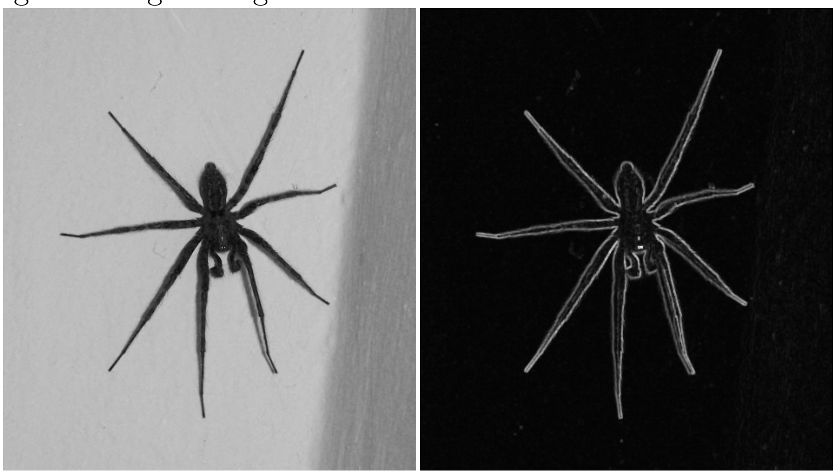
Exemplu: de netezire morfologică a unei imagini în tonuri de gri. Imaginea din stânga este cea originală, iar cea din dreapta este rezultatul netezirii morfologice a imaginii originale cu un element structurant 3×3.

Gradientul morfologic Prin combinarea operației de dilatare cu cea de erodare se poate obține o imagine similară ceu imaginea de gradient obținută cu ajutorul unui filtru trecesus. Formula asociată este dată prin [?]:

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

și reprezintă imaginea obținută prin scăderea imaginii erodate din imaginea dilatată. Obținerea marginilor (*edges*) se realizează datorită proprietăților dilatării și erodării, care determină extinderea, respectiv reducerea zonelor deschise, regiunile uniforme nefiind modificate.

Exemplu: de gradient morfologic al unei imagini în tonuri de gri. Imaginea din stânga este cea originală, iar cea din dreapta este rezultatul gradientului morfologic al imaginii originale cu un element structurant 3×3.



6 Operații morfologice pe imagini color

Așa cum s-a remarcat în secțiunile anterioare, dilatarea și erodarea pot fi exprimate prin operații de maxim și respectiv minim pe o vecinătate delimitată de un element structurant. Astfel este necesară o ordonare a valorilor din vecinătate. În cazul imaginilor RGB, culoarea fiecărui pixel este reprezentată printr-un vector tridimensional. Pentru a putea determina o *culoare minimă* sau una *maximă* pe o vecinătate, este necesară stabilirea unui criteriu de comparare, respectiv ordonare a doi vectori de culoare.

Există o serie de abordări pentru rezolvarea acestei probleme.Una dintre cele mai utilizate clasificări a metodelor de ordonare pentru vectori este prezentată în [?]. Tipurile de metode considerate sunt:

1. **Metode marginale (*M-ordering*)** , în care se consideră separat fiecare componentă. O astfel de metodă poate considera de exemplu operații morfologice separat pe cele 3 canale de culoare. Astfel de operați însă tind să produce culori false (inexistente în imaginea originală).

2. **Metode de ordonare condițională (*C-ordering*)**, în care vectorii sunt ordonați pe baza uneia / unora dintre componente și a unor condiții de selectare în cadrul

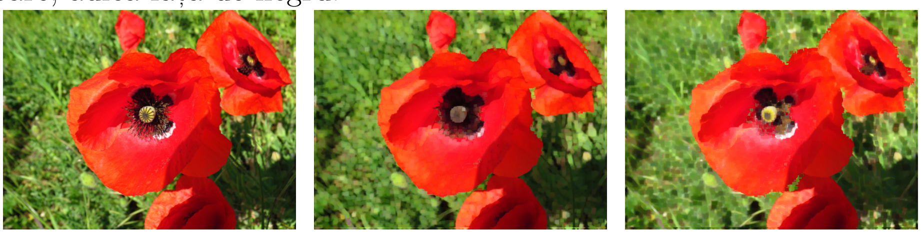
setului de date. Exemple de acest tip sunt:

- **Ordonarea lexicografică**. Acest tip de ordonare are dezavantajul că atribuie o prioritate crescută anumitor componente ale vectorului și ar trebui utilizată doar atunci, când acest lucru este dezirabil. În cazul imaginilor color de exemplu, dacă una dintre componentele de culoare este mai importantă dintr-un anumit motiv.
- **Ordonarea în spațiul HSV**, în care se ignoră componenta de culoare H (*Hue*).

3. **Metode de ordonare parțială (*P-ordering*)**, care partiționează setul de vectori în clase de echivalență relativ la o anumită ordonare.
4. **Metode de ordonare redusă (*R-ordering*)**. Acestea 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, $n = 3$. Iar ordonarea vectorilor este dată de ordonarea valorilor scalare asociate. Un exemplu de astfel de corespondență este aceea de asocia unui vector distanța față de un element de referință, de exemplu originea sistemului de coordonate. Este posibil ca funcția f să nu fie injectivă, caz în care este vorba de o *pre-ordonare* [?].

În [?] se propune clasificarea metodelor de ordonare a vectorilor în funcție de proprietățile algebrice ale acestora în *ordonări parțiale*, *pre-ordonări totale* și *ordonări totale*. Printre metodele prezentate în acest articol se află cele care ordonează în funcție de distanță (R-ordering), care realizează o pre-ordonare totală (funcțiile de distanță nefiind injective), ordonarea lexicografică, care este o ordonare totală precum și o metodă de ordonare bazată pe amestecarea biților fiecărui vector de culoare.

Exemplu: de operații morfologice asupra unei imagini RGB. Prima imaginea este cea originală, imaginea din mijloc reprezintă imaginea originală asupra căreia s-a efectuat o operație de opening, iar cea din dreapta este rezultatul unei operații de closing asupra imaginii originale. S-a folosit o metodă de tip R-ordering, în care valoarea scalară considerată pentru fiecare vector de culoare a fost distanța față de originea cubului de culoare, adică față de negru.



To Do: Implementați un operațiile de dilatare / erodare pentru imagini color, utilizând ordonarea în spațiul HSV în funcție ce canalul V, sau utilizând ordonarea vectorilor de culoare în raport cu distanța unei culori față de originea cubului de culoare RGB.

Să ne reamintim că

- operațiile morfologice au în general la bază operații de minim / maxim, deci presupun ordonarea culorilor după un anumit criteriu
- în cazul marginilor color, există diferite metode de ordonare a culorilor, de exemplu metodele marginale (M-ordering), care au la bază ordonarea după una sau mai multe componente ale vectorilor de culoare, sau metode de ordonare redusă (R-ordering), care asociază fiecărui vector o valoare scalară și realizează ordonarea după această valoare

Rezumat: în acest capitol

- a fost discutată problema operațiilor morfologice ca operații pe mulțimi de pixeli
- au fost discutați operatorii morfologici pe imagini binare: dilatare, erodare, opening și closing
- au fost prezentate câteva aplicații ale operatorilor morfologici pe imagini binare
- au fost discutați operatorii morfologici pe imagini în tonuri de gri: dilatare, erodare, opening și closing
- au fost prezentate: netezirea morfologică și gradientul morfologic
- s-a discutat problema operatorilor morfologici pentru imagini color