Morfologie matematică pentru imagini binare şi în tonuri de gri

M. Ivanovici, A. Căliman, B. Budescu

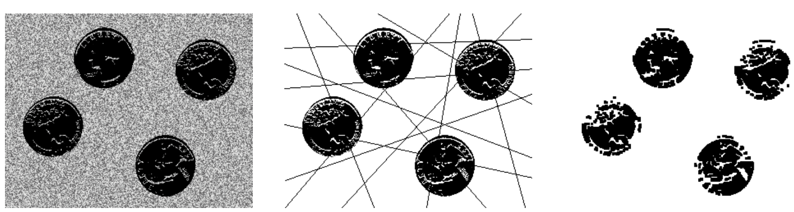
19 decembrie 2011

BREVIAR TEORETIC

Etimologia cuvântului morfologie are la bază cuvintele greceşti *morphos* care înseamnă formă şi *logos* care înseamnă ştiinţă, prin urmare morfologia este ştiinţa care se ocupă de analiza formelor. Morfologia matematică este o ramură specială a matematicii care se ocupă de analiza formelor reprezentate ca mulţimi de puncte. Morfologia clasică propusă de G. Matheron şi J. Serra operează cu mulţimi binare, prin urmare formele analizate şi prelucrate sunt forme ce rezultă ca urmare a unei operaţii de binarizare sau de segmentare pe imagini.

Prelucrările de tip morfologic vor viza modificarea formei obiectelor, nu a poziţiilor sau valorilor pixelilor - ca în cazul prelucrărilor de pñă acum.

Aplicaţia clasică a morfologiei matematice este de a construi filtre morfologice pentru post-procesarea hărţilor de segmentare.



imagine + zgomot gaussian binarizată	imagine + zgomot structural binarizată	....

Figure 1: .

Vom defini noţiunile următoare în contextul morfologiei pentru obiecte continue, urmând a ne concentra apoi pe morfologia pentru obiecte discrete (utilă în cazul prelucrării imaginilor digitale).

În continuare vom prezenta câteva noţiuni elementare de teoria mulţimilor, obligatorii pentru înţelegerea morfologiei matematice. Vom considera o mulţime A continuă, formată din puncte într-un spaţiu bidimensional E al tuturor punctelor. Oricare ar fi A, $A \in E$. E este o mulţime dotată cu o structură topologică şi considerăm că este un spaţiu vectorial pentru care sunt definite operaţiile de adunare între vectori şi înmulţire cu un scalar, λ:

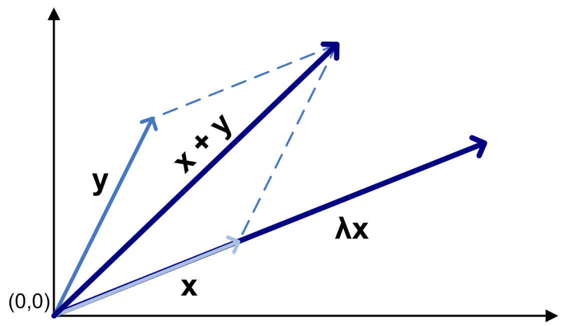


Figure 2: Ilustrarea operaţiilor de adunare de vectori şi înmulţire cu un scalar.

Pentru mulţimea A, se defineşte translatarea mulţimii cu vectorul t:

$$A_t = \{y \mid \exists x \in A, y = x + t\} \tag{1}$$

Ilustrarea translatării unei mulţimi A cu vectorul t într-un spaţiu bi-dimensional este prezentată în Figura 5

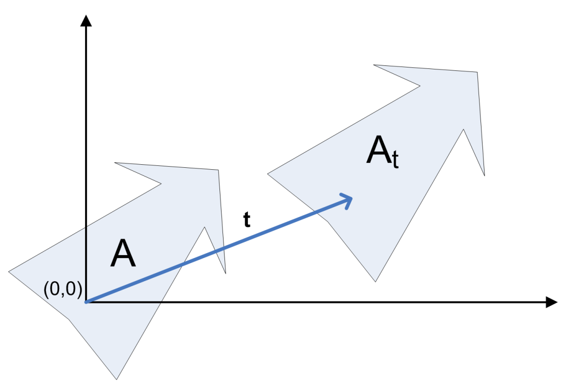


Figure 3: Translatarea mulţimii A cu vectorul t.

Mulţimea simetrică mulţimii A faţă de origine:

$$A^S = \{y \mid \exists x \in A, y = -x\} \tag{2}$$

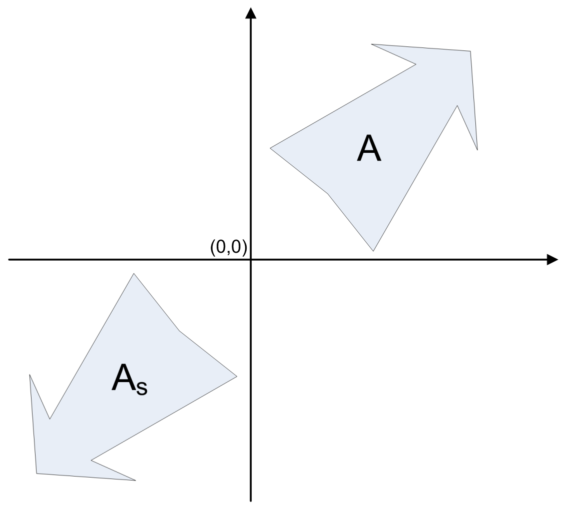


Figure 4: Mulţimea simetrică mulţimii A.

Complementul sau mulţimea complementară mulţimii A:

$$C_A = \{y \mid y \notin A\} \tag{3}$$

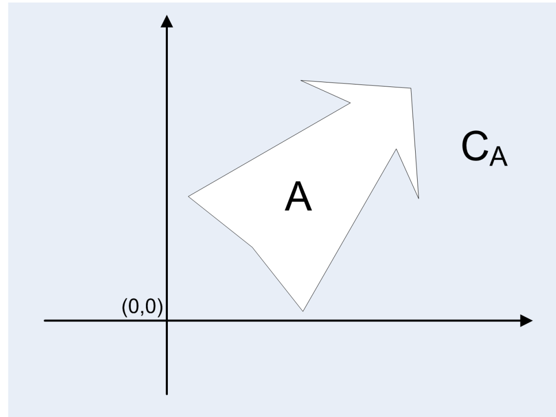
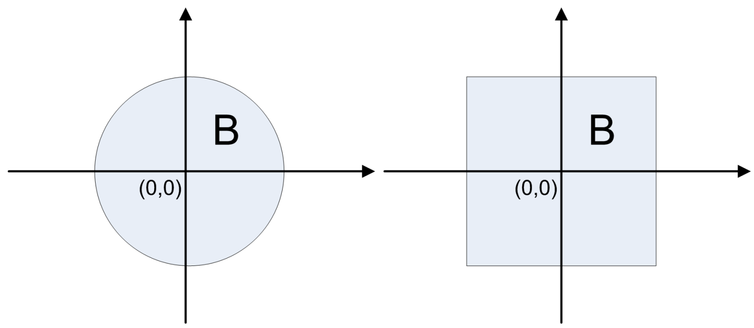


Figure 5: Mulţimea complementară mulţimii A.

Ideea: considerarea imaginii ca o mulţime asupra căreia se aplică transformări (comparaţii cu mulţimi mai simple → *elemente structurante* (ES))



(a) ES disc (b) ES pătrat

Figure 6: Elemente structurante.

1 Operaţii de bază în morfologia matematică

1.1 Definiţii

Erodarea morfologică a mulţimii A prin elementul structurant B se defineşte ca mulţimea punctelor (elementelor) în care se poate translata ES astfel încât acesta să fie inclus în mulţimea de prelucrat A:

$$A \ominus B = \{x \mid B_x \subseteq A\} \tag{4}$$

Punctele în care se translatează ES reprezintă de fapt punctele în care se translatează originea ES. În Figura 7 este ilustrată operaţia morfologică de erodare, pentru o muţime A de formă dreptunghiulară şi un element structurand B de tip disc.

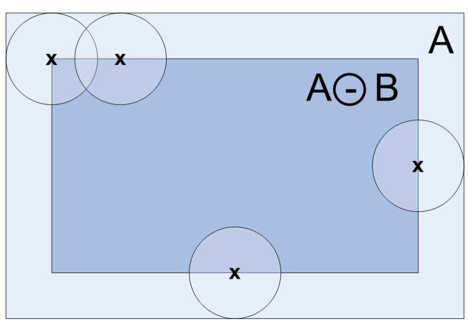


Figure 7: Operaţia morfologică de erodare.

Dilatarea morfologică a mulţimii A prin elementul structurant B se defineşte ca mulţimea punctelor (elementelor) în care se poate translata ES astfel încat acesta să aibă puncte comune cu mulţimea de prelucrat A:

$$A \oplus B = \{x \mid B_x \cap A \neq \emptyset\} \tag{5}$$

În Figura 8 este ilustrată operaţia morfologică de dilatare, pentru o muţime A de formă dreptunghiulară şi un element structurand B de tip disc.

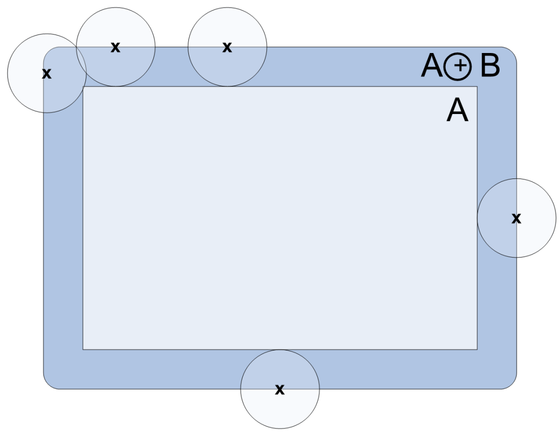


Figure 8: Operaţia morfologică de dilatare.

Pentru o imagine binară oarecare, observaţi efectele operaţiilor de erodare şi dilatare morfologică în Figura 9

(a) Imagine binară (b) Erodata imaginii (c) Dilatata imaginii

Figure 9: Erodarea şi dilatarea unei imagini binare.

1.2 Proprietăţi

- Dualitatea:

$$C_{(A \oplus B)} = C_A \oplus B \tag{6}$$
$$C_{(A \ominus B)} = C_A \ominus B \tag{7}$$

- Translaţia elementului structurant:

$$A \oplus B_t = (A \oplus B)_t \tag{8}$$

$$A \ominus B_t = (A \ominus B)_{-t} \tag{9}$$

- Translaţia mulţimii A:

$$A_t \oplus B = (A \oplus B)_t \tag{10}$$

$$A_t \ominus B = (A \ominus B)_t \tag{11}$$

- Mulţimea λA unde λ este real:

$$\frac{1}{\lambda}(\lambda A \oplus B) = A \oplus \frac{1}{\lambda}B \tag{12}$$

$$\frac{1}{\lambda}(\lambda A \ominus B) = A \ominus \frac{1}{\lambda}B \tag{13}$$

- Monotonie:

$$A_1 \subset A_2 \implies A_1 \oplus B \subset A_2 \oplus B \tag{14}$$

$$A_1 \ominus B \subset A_2 \ominus B \tag{15}$$

$$B_1 \subset B_2 \implies A \oplus B_1 \subset A \oplus B_2 \tag{16}$$

$$A \ominus B_1 \supset A \oplus B_2 \tag{17}$$

$$O \in B \implies A \subset A \oplus B \tag{18}$$

$$A \ominus B \subset A \tag{19}$$

- Pseudo-comutativitate:

$$A \oplus B = (B \oplus A)^S \tag{20}$$

- Asociativitate:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C^S \tag{21}$$

$$A \ominus (B \ominus C) = (A \ominus B) \ominus C^S \tag{22}$$

- Operaţii cu mulţimi:

$$(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C) \tag{23}$$

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C) \tag{24}$$

  Observatie: dacă elementul structurant e complex, îl putem descompune în mai multe elemente şi aplicăm relaţia de mai sus.

$$(A \cap B) \ominus C = (A \ominus C) \cap (B \ominus C) \tag{25}$$

$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C) \tag{26}$$

$$A \oplus (B \cap C) \subset (A \oplus B) \cap (A \oplus C) \tag{27}$$
$$A \ominus (B \cap C) \supset (A \ominus B) \cup (A \ominus C) \tag{28}$$

$$(B \cap C) \oplus \subset (B \oplus A) \cap (C \oplus A) \tag{29}$$
$$(B \cap C) \ominus \supset (B \ominus A) \cap (C \ominus A) \tag{30}$$

2 Operaţii morfologice derivate

2.1 Definiţii

Deschiderea morfologică a mulţimii A prin elementul structurant B se defineşte ca erodarea mulţimii cu elementul structurant respectiv, urmată de dilatarea cu simetricul elementului structurant:

$$A \circ B = (A \ominus B) \oplus B^S \tag{31}$$

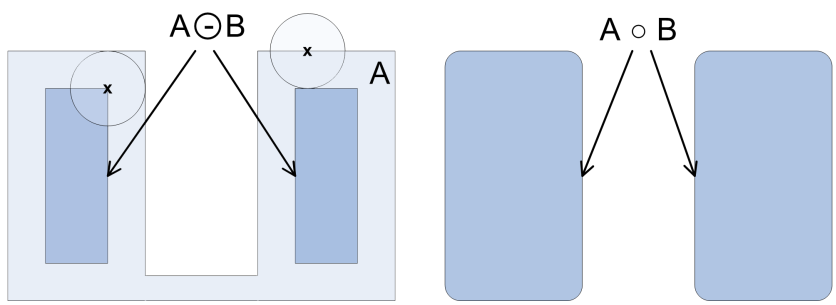


Figure 10: Deschiderea morfologică.

Închiderea morfologică a mulţimii A prin elementul structurant B se defineşte ca dilatarea mulţimii cu elementul structurant respectiv, urmată de erodarea cu simetricul elementului structurant:

$$A \bullet B = (A \oplus B) \ominus B^S \tag{32}$$

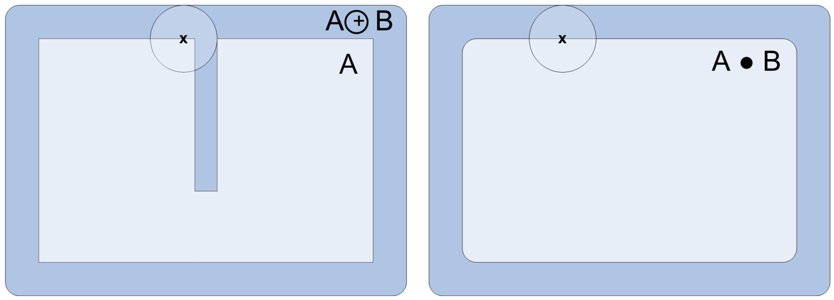


Figure 11: Închiderea morfologică.

În general, aa aplicarea operaţiilor de închidere şi de deschidere colţurile obiectelor se netezesc. La închidere canalele strâmte se unesc, iar obiectele care aparent erau separate devin un singur obiect, cu condiţia ca raza elementrului structurant să fie cel puţin egală cu jumătatea lăţimii canalului. La deschidere istmurile strâmpte dispar dând naştere la "insule", cu alte cuvinte se vor separa obiecte care iniţial erau unite.

(a) Închiderea imaginii (b) Deschiderea imaginii

Figure 12: Închiderea şi deschiderea pentru imaginea binară din Figura 9.

2.2 Proprietăţi

- Dualitatea:

$$C_{(A \circ B)} = C_A \bullet B \tag{33}$$

$$C_{(A \bullet B)} = C_A \circ B \tag{34}$$

- Idempotenţa:

$$(A \circ B) \circ B = A \circ B \tag{35}$$

$$(A \bullet B) \bullet B = A \bullet B \tag{36}$$

- Translaţia:

$$A_t \circ B = (A \circ B)_t \tag{37}$$

$$A_t \bullet B = (A \bullet B)_t \tag{38}$$

- Mulţimea λA:

$$\frac{1}{\lambda}(\lambda A \circ B) = A \circ \frac{1}{\lambda}B \tag{39}$$

$$\frac{1}{\lambda}(\lambda A \bullet B) = A \bullet \frac{1}{\lambda}B \tag{40}$$

- Monotonia:

$$A_1 \subset A_2 \implies A_1 \circ B \subset A_2 \circ B \tag{41}$$

$$A_1 \bullet B \subset A_2 \bullet B \tag{42}$$

$$B_1 \subset B_2 \implies A \circ B_1 \supset A \oplus B_2 \tag{43}$$

$$A \bullet B_1 \subset A \oplus B_2 \tag{44}$$

$$A \circ B \subset A \subset A \bullet B, \forall B \tag{45}$$

3 Aplicaţii ale operaţiilor morfologice

3.1 Extragere de contur morfologic

Pentru contur există mai multe definiţii din punct de morfologic:

- Conturul interior:

$$\delta A = A - (A \ominus B) \tag{46}$$

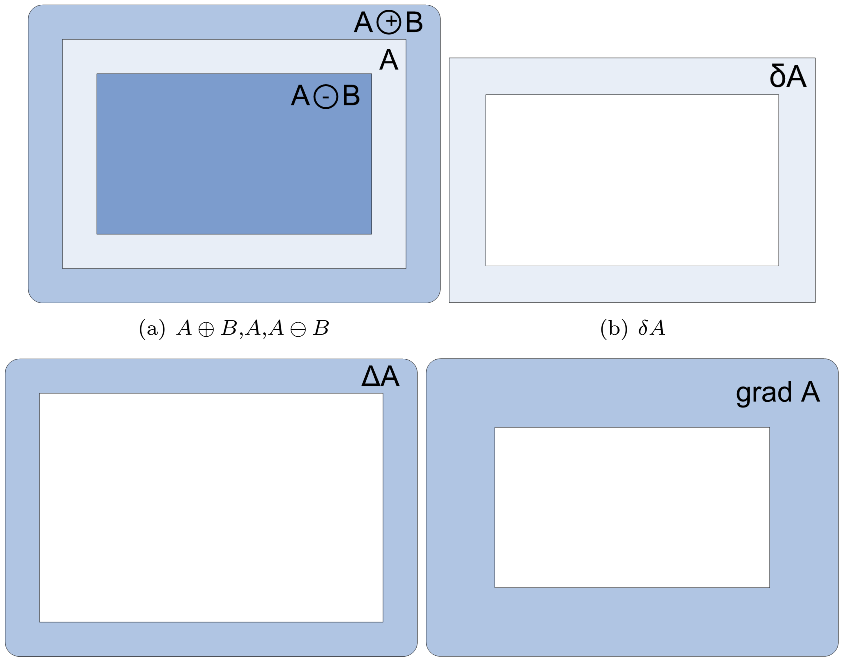
- Conturul exterior:

$$\Delta A = (A \oplus B) - A \tag{47}$$

- Gradientul morfologic:

$$gradA = (A \oplus B) - (A \ominus B) \tag{48}$$

Pentru ilustrare, în Figura 13 sunt prezentate mulţimea A de formă dreptunghiulară, mulţimea dilatată $A \oplus B$ şi mulţimea erodată $A \ominus B$, pentru un element structurant B de tip disc.



(a) $A \oplus B$,A,$A \ominus B$ (b) δA

(c) ΔA (d) $grad\ A$

Figure 13: O mulţime A, mulţimile dilatată şi erodată, conturul interior, cel exterior şi gradientul simetric.

Figure 14: Gradient morfologic simetric pentru imaginea binară din Figura 9.

3.2 Filtre alternate secvenţiale

$$(((((A \circ B) \bullet B) \circ 2B) \bullet 2B) \circ 3B)... \tag{49}$$

$$(((((A \bullet B) \circ B) \bullet 2B) \circ 2B) \bullet 3B)... \qquad (50)$$

4 Extinderea morfologiei matematice la imagini în nivele de gri

În cazul imaginilor în nivele de gri, mulţimile de prelucrat sunt reprezentate de punctele situate pe suprafaţa determinată de funcţia imagine, împreună cu umbra acestei funcţii.

Umbra unei funcţii $f : B \subseteq \mathbb{Z}^{n-1} \rightarrow C \subseteq \mathbb{Z}$ oarecare:

$$U(f) = \{(z, y) | z \in B, y \in \mathbb{Z}, y \leq f(z)\} \qquad (51)$$

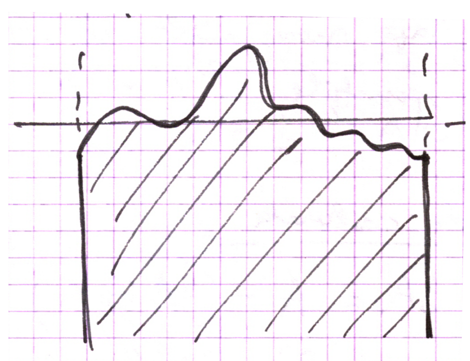


Figure 15: Umbra unei funcţii.

Definirea operaţiilor morfologice de bază:

- Erodarea:
$$f \ominus g = \min_{y \in Supp(g)} \{f(x - y) - g(y)\} \qquad (52)$$

- Dilatarea:
$$f \oplus g = \max_{y \in Supp(g)} \{f(x - y) + g(y)\} \qquad (53)$$

Dacă $g(y) = 0, \forall y \in Supp(y)$, g se numeşte element structurant plat (*flat*). În cazul în care se utilizează elemente structurante de tip *flat*, cele 2 operaţii morfologice devin identice cu filtrările de ordine de rang minim şi respectiv maxim.

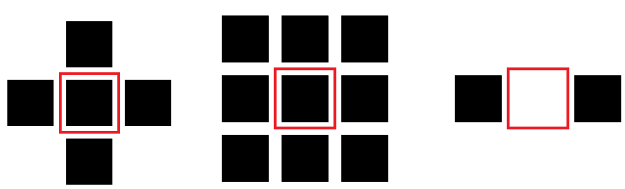
(a) Imagine originală (b) Erodata imaginii (c) Dilatata imaginii

Figure 16: Erodata şi dilatata unei imagini în tonuri de gri.

(a) Închiderea imaginii (b) Deschiderea imaginii (c) Gradient simetric

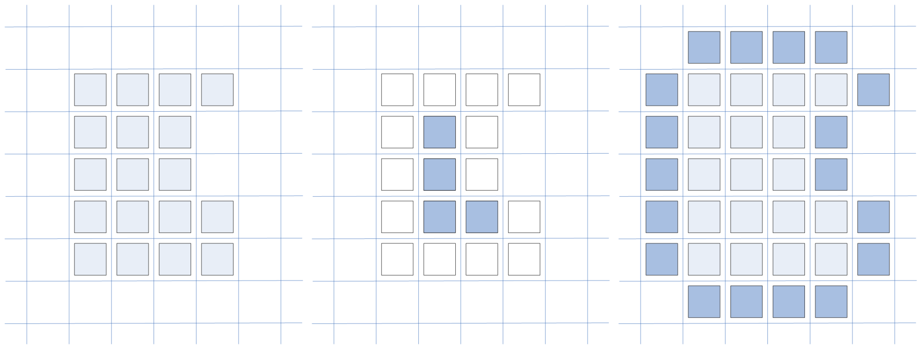
Figure 17: Deschiderea, închiderea şi gradientul morfologic pentru imaginea în tonuri de gri din Figura 16.

5 Morfologie matematică pentru mulţimi binare discrete



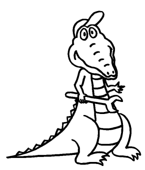
ES - cruce ES - patrat ES care nu îşi conţine originea

Un exemplu de dilatare şi erodare pentru o mulţime binară discretă, folosind un element structurand de tip cruce, este prezentat în Figura 18:



(a) A (b) $A \ominus B$ (c) $A \oplus B$

Figure 18: Erodata şi dilatata unei imagini binare discrete.



EXERCIŢII

Biblioteca Intel OpenCV pune la dispoziţie următoarele funcţii pentru implementarea operaţiilor morfologice descrise anterior: `cvCreatingStructuringElementEx`, `cvReleaseStructuringElement`, `cvErode`, `cvDilate` şi `cvMorphologyEx`. Acestea sunt prezentare pe scurt în continuare.

5.1 Funcţia `cvCreateStructuringElementEx`

Această funcţie permite crearea unui obiect de tip element structurant. Declaraţia ei este următoarea:

```
   IplConvKernel* cvCreateStructuringElementEx( int nCols, int nRows,
int anchorX, int anchorY, CvElementShape shape, int* values );
```

Parametrii de intrare ai funcţiei au următoarea semnificaţie: `nCols` este numărul de coloane ale elementului structurant, `nRows` este numărul de linii ale elementului structurant, `anchorX` este deplasamentul relativ pe axa orizontală al punctului central al elementului structurant iar `anchorY` este deplasamentul relativ pe axa verticală al punctului central al elementului structurant.

Parametrul `shape` indică forma elementului structurant şi poate avea una din următoarele valori: `CV_SHAPE_RECT` pentru un element structurant dreptunghiular, `CV_SHAPE_CROSS` pentru un element structurant în formă de cruce, `CV_SHAPE_ELLIPSE` pentru un element structurant de formă eliptică sau `CV_SHAPE_CUSTOM` pentru o formă oarecare definită de către utilizator. În acest ultim caz parametrul `values` va specifica masca care defineşte elementul structurant, adică vecinătatea pixelului considerat a fi centrul elementului structurant. `values` este un pointer către o structură de date de tip şir (array), ce reprezintă matricea elementului structurant parcursă linie cu linie. Valorile diferite de zero indică puncte care aparţin elementului structurant. Dacă poniter-ul este NULL, atunci toate valorile sunt considerate diferite de zero, caz în care elementul structurant este de formă dreptunghiulară. Acest parametru este luat în consideraţie doar atunci când parametrul `shape` are valoarea `CV_SHAPE_CUSTOM`.

Funcţia `cvCreateStructuringElementEx` crează un obiect de tip `IplConvKernel`, care poate fi folosit ca element structurant în implementarea operaţiilor morfologice.

5.2 Funcţia `cvReleaseStructuringElement`

Această funcţie de-alocă memoria utilizată pentru un obiect de tip element structurant, iar declaraţia ei este:

```
void cvReleaseStructuringElement( IplConvKernel** ppElement );
```

Parametrul `ppElement` este un pointer către obiectul de tip element structurant ce se doreşte a fi "şters". Dacă acest pointer este NULL atunci funcţia nu are nici un efect.

5.3 Funcţia `cvErode`

Aceasta implementează operaţia de erodare cu un element structurant oarecare. Declaraţia acestei funcţii este următoarea:

```
void cvErode( const CvArr* A, CvArr* C, IplConvKernel* B=0, int
iterations=1 );
```

Parametrii de intrare ai funcţiei sunt: `A` reprezintă obiectul imagine sursă, `C` reprezintă obiectul imagine destinaţie, `B` reprezintă obiectul de tip element structurant folosit pentru erodare. Dacă acesta are valoarea NULL, atunci implicit este folosit un element structurant pătrat de dimensiuni 3x3, iar `iterations` indică de câte ori este aplicată asupra imaginii sursă operaţia de erodare.

Funcţia `cvErode` permite operarea "inline", adică permite ca imaginea sursă să fie aceeaşi cu imaginea destinaţie.

De menţionat faptul că erodarea unei imagini color presupune transformarea independentă a tuturor canalelor imaginii, ceea ce nu reprezintă o operaţie morfologică validă din punctul de vedere al prelucrării imaginilor color.

5.4 Funcţia `cvDilate`

Implementează operaţia de dilatare cu un element structurant oarecare. Declaraţia funcţiei este:

```
void cvDilate( const CvArr* A, CvArr* C, IplConvKernel* B=0, int
iterations=1 );;
```

`A` reprezintă obiectul imagine sursă, `C` reprezintă obiectul imagine destinaţie, `B` reprezintă obiectul de tip element structurant folosit pentru dilatare. Dacă acesta are valoarea NULL, atunci implicit este folosit un element structurant pătrat de dimensiuni 3x3, iar `iterations` indică de câte ori este aplicată asupra imaginii sursă operaţia de dilatare.

Funcţia `cvDilate` permite operarea "inline" (imaginea sursă să fie aceeaşi cu imaginea destinaţie). Dilatarea unei imagini color presupune transformarea independentă a tuturor canalelor imaginii, dar a se vedea menţiunea de mai sus cu privire la pertinenţa unei astfel de operaţii.

5.5 Funcţia `cvMorphologyEx`

Implementează operaţii morfologice avansate, utilizând operaţiile de erodare şi dilatare ca operaţii de bază.

```
void cvMorphologyEx( const CvArr* A, CvArr* C, CvArr* temp, IplConvKernel*
B, CvMorphOp op, int iterations );
```

`A` reprezintă imaginea sursă, `C` reprezintă imaginea destinaţie, `temp` reprezintă o imagine temporală, atunci când este necesară, `B` reprezintă elementul structurant, `op` indică tipul operaţiei morfologice. Valorile posibile ale acestui parametru sunt următoarele:

- `CV_MOP_OPEN` indicând o operaţie de deschidere: C=open(A,B)=dilate(erode(A,B),B),
- `CV_MOP_CLOSE` indicând o operaţie de închidere: C=close(A,B)=erode(dilate(A,B),B),
- `CV_MOP_GRADIENT` indicând o operaţie de gradient morfologic: C=morph_grad(A,B)=dilate(A,B)-erode(A,B),
- `CV_MOP_TOPHAT` indicând o operaţie "top hat": C=tophat(A,B)=A-erode(A,B),
- `CV_MOP_BLACKHAT` indicând o operaţie "black hat": C=blackhat(A,B)=dilate(A,B)-A.

`iterations` indică de câte ori sunt aplicate operaţiile de erodare şi dilatare.

Obiectul temporal de tip imagine, `temp`, este necesar pentru cazul în care operaţia este gradient morfologic sau atunci când imaginea sursă este aceeaşi cu imaginea destinaţie (operaţie "inplace") şi operaţia morfologică este "top hat" sau "black hat".

5.6 Probleme propuse

1. Să se implementeze operaţia de dilatare pentru o imagine binară şi să se aplice de un număr arbitrar de ori, folosind funcţia OpenCV `cvDilate`.

2. Să se implementeze operaţiile de deschidere şi închidere pentru o imagine binară şi să se aplice de un număr arbitrar de ori, folosind funcţiile OpenCV `cvErode` şi `cvDilate`.

3. Să se implementeze operaţia de dilatare pentru o imagine în tonuri de gri şi să se aplice de un număr arbitrar de ori, folosind funcţia OpenCV `cvDilate`.

4. Să se implementeze operaţia de erodare pentru o imagine în tonuri de gri şi să se aplice de un număr arbitrar de ori, folosind funcţia OpenCV `cvErode`.

5. Să se implementeze operaţiile de deschidere şi de închidere pentru o imagine în tonuri de gri.

6. Să se extragă cele trei tipuri de contururi morfologice pentru imagini binare şi în nivele de gri.

7. Să se repete exerciţiile de mai sus pentru diverse forme ale elementului structurand (cruce, disc, etc.)