

# PRELUCRAREA IMAGINILOR

## CURS 6

### *Transformări ale imaginilor*

6.04.2017

# Transformări ale imaginilor

În general, putem spune că transformarea unei imagini se adresează:

- **ochiului uman**, pentru a putea observa mai bine anumite caracteristici ale imaginii studiate, sau
- **prelucrării automate** în scopul *recunoașterii formelor*.

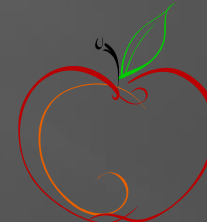
Scopul transformărilor descrise în acest paragraf este de a obține anumite **structuri formate din linii și curbe** (imagini de tip 3) necesare *recunoașterii formelor*.

În *analiza imaginilor*, o etapă importantă o constituie **extragerea caracteristicilor** în scopul descrierii sau interpretării scenelor, urmată de o altă etapă și anume **segmentare** care presupune printre altele **determinarea conturului**.

# Transformări ale imaginilor



1. *Determinarea conturului*



2. *Scheletizare*



3. *Subțiere*



# 1. *Determinarea conturului*

*Determinarea conturului* este transformarea unei imagini de tip 2 (alb-negru) în imagine de tip 3 (formată din linii și curbe), după clasificarea dată în [6].

Muchiile, care caracterizează *conturul obiectului* (conturul fiind descris prin muchii), sunt utile în recunoașterea obiectelor (clasificarea în cazul reprezentărilor codificate).

**Muchiile** sunt locații de pixeli cu salturi mari de nuanță (nivel) de gri.

În imaginile alb-negru, muchiile sunt formate din puncte (pixeli) de culoare neagră cu cel puțin un punct alb în vecinătate.

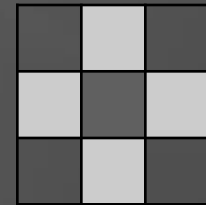
## ... 1. Determinarea conturului

În rezolvarea aceste probleme trebuie precizate următoarele:

a) **Conexitatea**, adică definirea vecinătății.

Putem considera că un punct  $P(j,i)$  are patru vecini ( pe cele patru direcții,  $r(P)$ ,  $u(P)$ ,  $l(P)$ ,  $d(P)$  ) din fereastra ecran ( $V_E$ ) :

$$V_4(P) = \{ Q \in V_E / \epsilon(P,Q) = 1 \}$$



sau **opt** vecini :

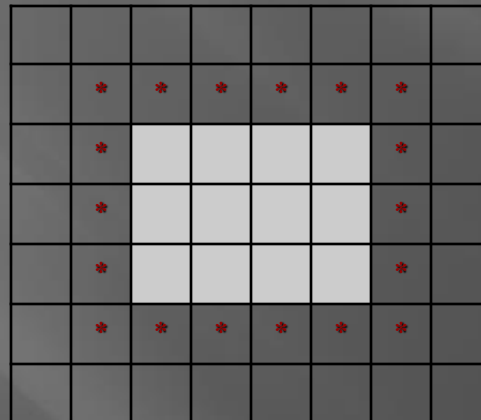
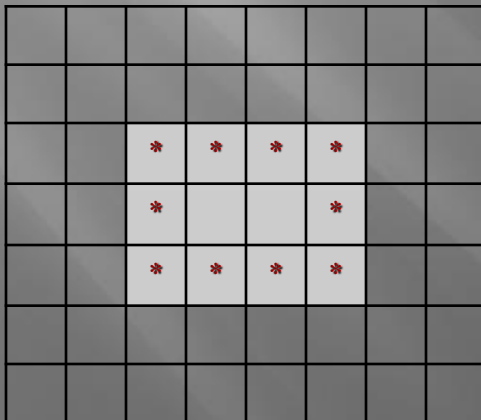
$$V_8(P) = \{ Q \in V_E / 1 \leq \delta(P,Q) < 2 \}$$



## ... 1. Determinarea conturului

### b) Precizarea apartenenței conturului (*Interior* sau *Exterior*).

Dacă imaginea este *alb-negru*, atunci vom preciza culoarea fondului și culoarea obiectului. Practic vom preciza o culoare (*alb* sau *negru*) care reprezintă culoarea punctelor pentru care se verifică apartenența la contur (conturul poate fi unul dintre următoarele două):



## ... 1. Determinarea conturului

De exemplu  $P \in \text{Conturului}$  dacă :

- $\text{Culoare}(P) = \text{Negru}$  și
- $|\{Q \in V_v(P) / \text{Culoare}(Q) = \text{Alb}\}| > 1$

Practic se verifică pentru fiecare punct  $P \in V_E$  condițiile de tipul celor de mai sus, sau mai simplu, putem spune:

$P \in \text{Conturului}$  dacă următoarea expresie este adevărată:

$(\text{Ob}(P) \text{ Xor } (\text{Ob}(u(P)) \text{ Or } (\text{Ob}(d(P)))))) \text{ Or } (\text{Ob}(P) \text{ Xor } (\text{Ob}(l(P)) \text{ Or } (\text{Ob}(r(P))))))$  ,

unde:

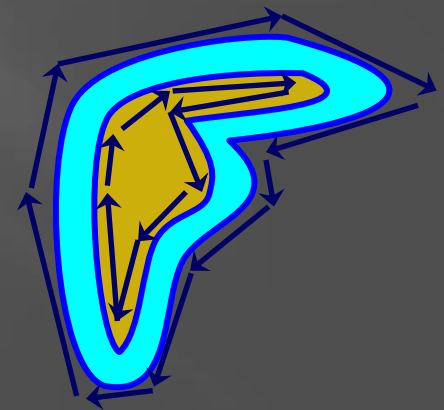
$\text{Ob}(P) = (\text{Culoare}(P) = \text{Culoare\_Obiect})$  ,

$\text{Culoare\_Obiect} \in \{ \text{Alb}, \text{Negru} \}$ .

c) **Obținerea descrierii conturului** prin traversarea punctelor determinate.

Mulțimea punctelor  $P$  determinate anterior se va ordona, prin parcurgerea acestei din vecin în vecin (rezultând și șirul comenzilor de descriere, adică  $\Pi$ -cuvântul corespunzător) începând cu un punct ales din contur (de exemplu cel mai din stânga-sus), până se revine la punctul inițial sau nu se mai poate deplasa.

Dacă mai există puncte din contur netraversate se construiește alt cuvânt de descriere și așa mai departe. În final vom avea o mulțime de  $\Pi$ -cuvinte de descriere, deci un  $\Pi$ -limbaj (așa cum se poate vedea în figura alăturată unde vor fi două cuvinte de descriere).





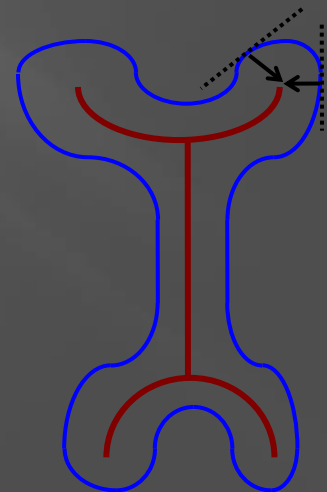
## 2. *Scheletizare*

Obiectele sau scenele pot fi descrise prin diverse *structuri* compuse din diferite elemente (*linii, curbe, etc*). De exemplu în *recunoașterea caracterelor, amprentelor, cromozomilor, a norilor, etc.*, sunt necesare transformări ale *axeii mediane* în scopul obținerii unei *descrieri a obiectului studiat*.

Vom studia două clase de algoritmi:

- de *scheletizare* și
- de *subțiere*.

Intuitiv, putem să definim *scheletul* ca fiind mulțimea punctelor în care se întâlnesc cel puțin două *tangente* la contur care pleacă cu aceeași viteză.



## ... 2. Scheletizare

Practic, *scheletul* unui obiect *Ob* este definit ca fiind mulțimea punctelor  $P \in Ob$  pentru care *distanța* până la cel mai apropiat punct de pe contur (notată cu  $\delta(P)$ ) realizează un *maxim local*.

Algoritmul de determinare a scheletului unui obiect este următorul:

- Calculează  $\delta(P)$  pentru toate punctele  $P \in Ob$ :

- $\delta_0(P) = Culoare(P) \in \{0, 1\}$  ,  $\forall P \in Ob$ ;

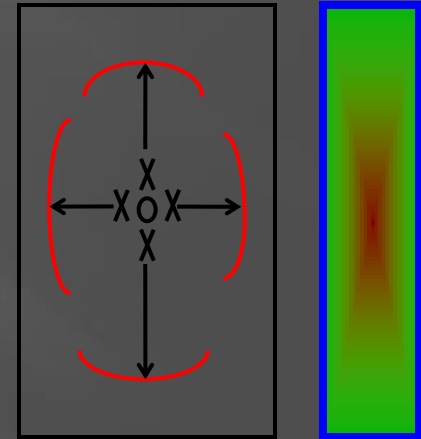
0 = *Negru* este culoarea fondului iar

1 = *Alb* este culoarea obiectului;

- $\delta_k(P) = Culoare(P) + \underset{Q \in V_4(P)}{\text{Min}} \delta_{k-1}(Q)$  ,  $\forall P \in Ob$ ,

$k=1,2,...,lățimea\ obiectului (\leq!)$ ;

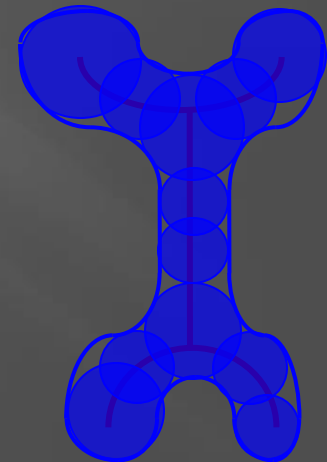
- Determină *Scheletul* =  $\{ S \in Ob / \delta(S) \geq \delta(P), \forall P \in V_4(S) \}$



Reconstituirea obiectului plecând de la *scheletul* acestuia se poate realiza utilizând formula :

$$Ob_r = \bigcup_{P \in Schel} \{ O \in V_E / d(O,P) < \delta(P) \}$$

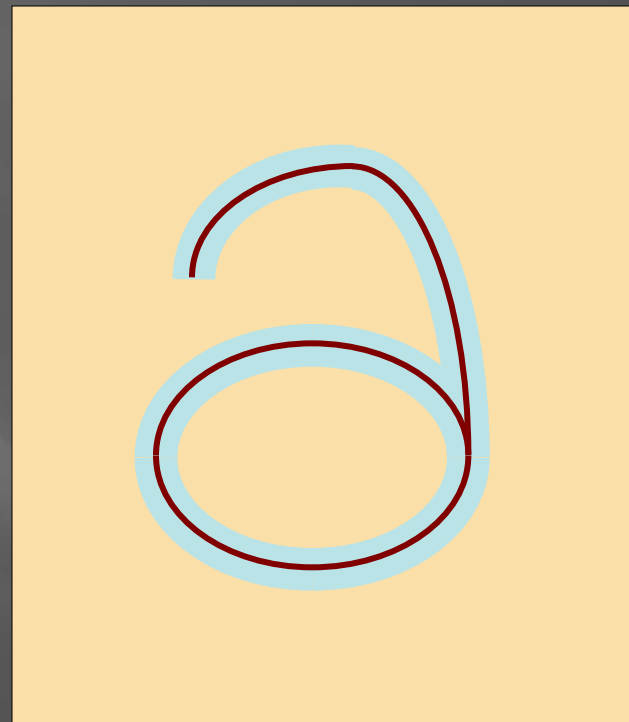
Reconstituirea se realizează practic prin *desenarea discurilor* de centru  $P$  și rază  $\delta(P)$ , pentru fiecare punct  $P \in Schel$ .



### 3. *Subțiere*

Algoritmii de *subțiere* urmăresc transformarea obiectelor într-un set de arce digitale aflate de-a lungul axelor mediane.

O caracteristică importantă a acestor algoritmi constă în faptul că structura obținută nu depinde de neregularitățile mici ale conturului.



### ... 3. Subțiere

Strategia acestor algoritmi poate fi următoarea: se elimină punctele  $P \in Ob \cap Contur(Ob)$ , adică acele puncte de pe marginea sau frontiera obiectului, care îndeplinesc următoarele condiții:

- au cel puțin doi vecini ( $\in Ob$ );

$$| V_8(P) \cap Ob | \geq 2 \quad ( N_v(P) > 1 )$$

- nu provoacă prin eliminare o deconectare a obiectului (rupere a legăturilor)

$Ob$  este conex  $\rightarrow Ob \setminus \{P\}$  este conex.

### ... 3. Subțiere

**Def.:**  $R$  este o *regiune conexă* dacă pentru  $\forall P, Q \in R$  există  $P_0=P, P_1, \dots, P_n=Q \in Ob$  astfel încât  $P_i \in V_8(P_{i-1}), \forall i=1, 2, \dots, n$ .

**Obs.:** Extremitățile arcelor subțiri nu trebuie eliminate prin acești algoritmi.

**Not.:** Numărul vecinilor punctului  $P$  (de culoare albă,  $\in Ob$ ) este :

$$Nv(P) ::= |\{Q \in Ob\} \cap V_8(P)|$$

**Not.:** Numărul tranzițiilor de la 0 la 1

în șirul punctelor  $P_1, P_2, \dots, P_8, P_9=P_1$  este:

3	2	1	1	1	0
4	P	8	1	1	1
5	6	7	0	0	0

$$Nt(P)=2$$

$$Nt(P) ::= |\{i \in \{1, \dots, 8\} / P_i \notin Ob \text{ și } P_{i+1} \in Ob\} \cap V_8(P)|$$

Punctul  $P \in Ob$  se poate elimina dacă următoarele condiții sunt îndeplinite:

- $2 \leq N_v(P) \leq 6$ 
  - dacă  $N_v(P)=1$  atunci P este extremitate deci nu se poate elimina, iar
  - dacă  $N_v(P)>6$  atunci P este punct interior, deci nu se poate elimina nici în acest caz;
- $N_t(P)=1$  ( pentru a nu deconecta obiectul ).

0	0	0
0	1	1
0	0	0

1	1	1
0	1	1
1	1	1

0	0	0
0	1	1
0	1	1

1	1	0
1	1	1
0	0	0

Algoritmul constă în determinarea și eliminarea punctelor care îndeplinesc aceste condiții, până când nu mai sunt modificări în imagine.

Algoritmul transformă o imagine într-un set de arce *conectate* pe patru direcții, deci pot fi descrise prin  $\Pi$ -cuvinte.

# Teme

Realizați următoarele *Transformări ale imaginilor*:

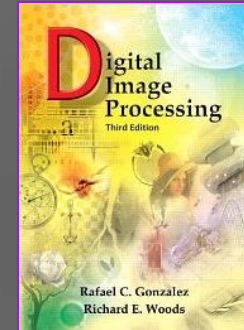
1. *Determinarea și descrierea (traversarea) conturului*
2. *Scheletizare și reconstrucție (verif.)*
3. *Subțiere și descriere (trav.)*



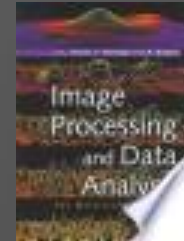


# References

- ▣ *Digital image processing,*  
Rafael C. González, Richard Eugene Woods



- ▣ *Image processing and data analysis: the multiscale approach,*  
Jean-Luc Starck, Fionn Murtagh, A. Bijaoui



- ▣ *Image Processing: Basics, Challenges and Perspectives*

[http://www.mas.ecp.fr/Personnel/lilla/classes/image\\_processing/index.html](http://www.mas.ecp.fr/Personnel/lilla/classes/image_processing/index.html)



École Centrale Paris

- ▣ *Lecture Notes in Digital Image Processing*

<http://eeweb.poly.edu/~onur/lectures/lectures.html>

- ▣ *Free ebook Image Processing and Data Analysis: The Multiscale Approach*

<http://www.onlinecomputerbooks.com/view.php?book=www.multiresolution.com/cupbook.pdf&t=Image+Processing+and+Data+Analysis:+The+Multiscale+Approach>

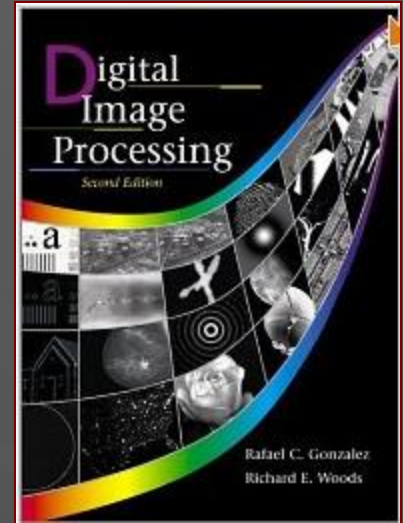
# ... References

## *Digital Image Processing*

Second Edition

- Rafael C. Gonzalez (University of Tennessee)
- Richard E. Woods

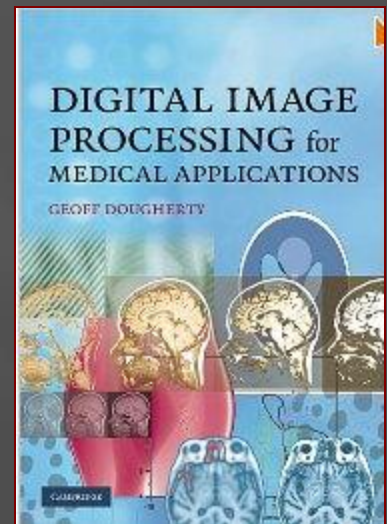
<http://www.amazon.com/Digital-Image-Processing-Rafael-Gonzalez/dp/0201180758>



## *Digital Image Processing for Medical Applications*

[Hardcover]

- Geoff Dougherty



▣ Contour Detection and Image Segmentation

by Michael Randolph Maire

B.S. (California Institute of Technology) 2003

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science in the GRADUATE DIVISION of the UNIVERSITY OF CALIFORNIA, BERKELEY

▣ Efficient, High-Quality Image Contour Detection

by Bryan Catanzaro, Bor-Yiing Su, Narayanan Sundaram, Yunsup Lee, ...

EECS Department, University of California at Berkeley, 573 Soda Hall, Berkeley, CA 94720

▣ Contour Detection and Hierarchical Image Segmentation

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik

▣ Progressive Conformal Snake Algorithm For Contour Detection

S. Y. LAM and C. S. TONG†

Department of Mathematics, Hong Kong Baptist University, Waterloo Road, Kowloon, Hong Kong

▣ Accurate Breast Contour Detection Algorithms in Digital Mammogram

Indra Kanta Maitra , Research Fellow, Department of CS & Engineering, University of Calcutta, India

Sanjay Nag Research Scholar Dep. of CS & E...,

Samir K Bandyopadhyay Professor Dep. of CS & E... .