

## 6. Algoritmul de urmărire a conturului

### 6.1. Obiective:

Obiectivele acestui laborator sunt următoarele:

- extragerea conturului obiectelor folosind algoritmul de urmărire a conturului;
- reprezentarea eficientă a conturului extras folosind coduri înălțuite;
- exploatarea avantajelor utilizării codurilor înălțuite în reprezentarea conturilor obiectelor (reconstrucția conturului, potrivire, unire, etc.).

### 6.2. Fundamente teoretice

#### 6.2.1. Algoritmul de urmărire a conturului

Algoritmul de urmărire a conturului este folosit pentru extragerea conturului obiectelor dintr-o imagine. La aplicarea acestui algoritm presupunem că imaginea este binară sau că obiectele din imagine au fost etichetate în prealabil.

Pașii algoritmului:

1. Se scanează imaginea din colțul stânga sus până când se găsește un pixel care aparține unei regiuni; acest pixel  $P_0$  reprezintă pixelul de start al conturului regiunii. Se definește o variabilă  $dir$  în care se reține direcția mutării anterioare de-a lungul conturului de la elementul anterior spre elementul curent. Se inițializează:
  - (a)  $dir = 0$  dacă conturul este detectat folosind vecinătate de 4 (Fig. 6.1a)
  - (b)  $dir = 7$  dacă conturul este detectat folosind vecinătate de 8 (Fig. 6.1b)
2. Se parcurge vecinătatea de 3x3 a pixelului curent în sens invers acelor de ceasornic, începând cu pixelul corespunzător poziției:
  - (a)  $(dir + 3) \bmod 4$  (Fig. 6.1c)
  - (b)  $(dir + 7) \bmod 8$  dacă  $dir$  este par (Fig. 6.1d)
  - (c)  $(dir + 6) \bmod 8$  dacă  $dir$  este impar (Fig. 6.1e)

Primul pixel găsit care are aceeași valoare ca și pixelul curent este noul element  $P_n$  al conturului. Se actualizează valoarea lui  $dir$ .
3. Dacă elementul curent  $P_n$  al conturului este egal cu al doilea element  $P_1$  din contur și dacă elementul anterior  $P_{n-1}$  este egal cu primul element  $P_0$ , atunci algoritmul se încheie. Altfel se repetă pasul (2).
4. Conturul detectat este reprezentat de pixelii  $P_0 \dots P_{n-2}$ .

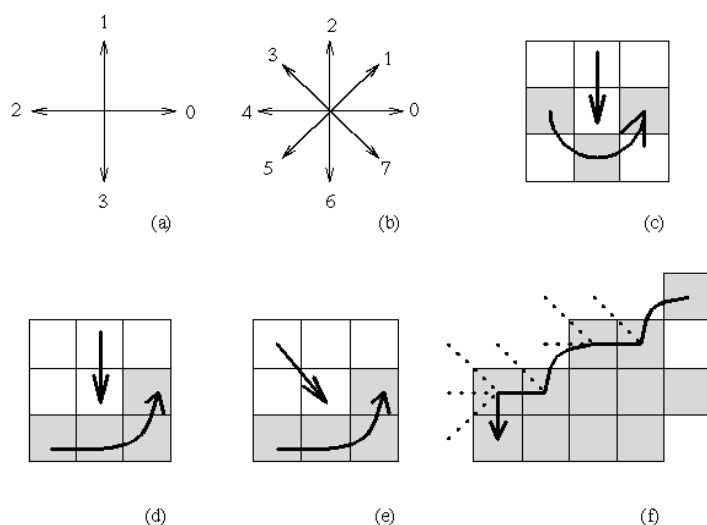


Fig. 6.1(a) Reprezentarea direcției, vecinătate de 4, (b) vecinătate de 8, (c) secvența de căutare în cazul vecinătății de 4 pixeli, (d),(e) secvența de căutare în cazul vecinătății de 8 pixeli, (f) urmărirea conturului pentru vecinătate de 8 (liniile întrerupte arată pixelii care au fost testați în timpul algoritmului de urmărire a conturului).



Fiind dat punctul de referință și codul înlănțuit, conturul triunghiului poate fi reconstruit complet. Codul înlănțuit reprezintă o modalitate eficientă de a stoca informația de contur deoarece în reprezentarea lui sunt necesari 3 biți ( $2^3 = 8$ ) pentru a determina oricare din cele 8 direcții de deplasare.

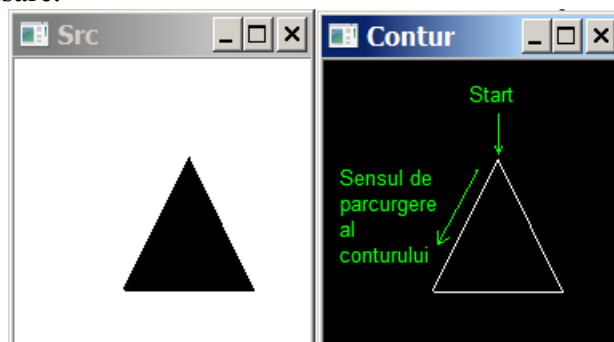


Fig. 6.3 Direcțiile codului înlănțuit cu numerele corespunzătoare

Codurile înlănțuite pot fi reprezentate independent de poziție prin ignorarea „punctului inițial” („punctul de început”). În cazul conturilor închise codurile înlănțuite pot fi normalizate în raport cu punctul de start prin alegerea acestuia astfel încât secvența rezultată la reprezentarea codului înlănțuit să formeze un număr întreg cu valoare absolută minimă.

„Derivata” codului înlănțuit este o reprezentare utilă deoarece este invariantă la rotația conturului. Derivata (o diferență *modulo* 4 sau 8) este o altă secvență de numere care indică direcția relativă a segmentelor codului înlănțuit; ele reprezintă numărul de rotații cu 90 de grade sau cu 45 de grade necesare pentru a obține direcția următorului segment din codul înlănțuit. O diferență *modulo* 4 sau 8 se numește derivata codului înlănțuit (vezi Fig. 6.4!).

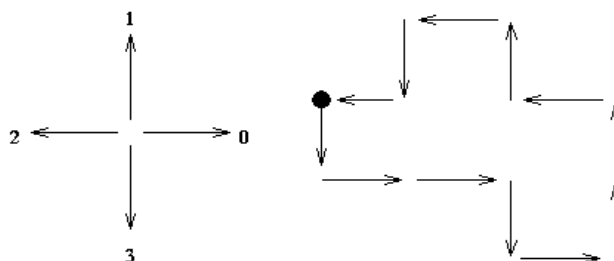


Fig. 6.4 Cod înlănțuit pentru vecinătate de 4 și derivata codului înlănțuit.

Cod: 3, 0, 0, 3, 0, 1, 1, 2, 1, 2, 3, 2

Derivata: 1, 0, 3, 1, 1, 0, 1, 3, 1, 1, 3, 1

***Proprietăți ale codului înlănțuit:***

- Codurile înălțuite descriu un obiect folosind o secvență de segmente de dimensiune unitate având orientări date (vecinătate de 4)
- Primul element al unei astfel de secvențe trebuie să conțină informații despre poziția primului pixel pentru ca regiunea să poată fi reconstruită.
- Codurile pare  $\{0, 2, 4, 6\}$  corespund direcțiilor verticale și orizontale; codurile impare  $\{1, 3, 5, 7\}$  corespund direcțiilor diagonale.
- Fiecare cod poate fi considerat ca fiind direcția unghiulară, în multipli de 45 de grade, în care trebuie să fie parcurși pixelii succesivi ai conturului.
- Coordonatele absolute ale primului pixel din contur (cel mai de sus, din stânga) împreună cu codul înălțuit al conturului reprezintă informația completă despre conturul regiunii.
- O schimbare a două elemente consecutive din codul înălțuit marchează o schimbare în direcția conturului. Punctul în care apare aceasta schimbare se numește *colt*.

### 6.3. Activități practice

Utilizând OpenCV Application și fișierele adiționale laboratorului:

1. Implementați algoritmul de urmărire a conturului și desenați conturul obiectului dintr-o imagine având un singur obiect.
2. Folosind algoritmul de urmărire a conturului scrieți algoritmul care construiește codul înlănțuit și derivata codului înlănțuit al unui obiect. Calculați și afișați (la linia de comanda sau într-un fișier text) ambele coduri (codul înlănțuit și derivata) pentru o imagine având un singur obiect.
3. Implementați o funcție care reconstruiește (afișează) conturul unui obiect peste o imagine, având ca și date de intrare coordonatele punctului de start și secvența de cod înlănțuit utilizând vecinătate de 8 (*reconstruct.txt*). Încărcați imaginea *gray\_background.bmp* și apelați funcția care reconstruiește conturul. Trebuie să obțineți conturul cuvântului „EXCELLENT” (având literele unite între ele).
4. **Salvați-vă ceea ce ați lucrat. Utilizați aceeași aplicație în laboratoarele viitoare. La sfârșitul laboratorului de procesare a imaginilor va trebui să prezentați propria aplicație cu algoritmii implementați!!!**

#### Informații adiționale:

Imaginile de test care conțin un singur obiect au:

- 8 biți/pixel;
- indexul 0 pentru pixelii obiect (pixeli negri)
- altă valoare a indexului pentru pixelii de fundal (pixeli albi)

Fișierul *reconstruct.txt* este un fișier text care conține:

- pe prima linie coordonatele punctului de start (rând și coloană) separate printr-un spațiu;
- pe a doua linie numărul de coduri înlănțuite;
- pe a treia linie codurile înlănțuite (secvența de direcții pentru vecinătate de 8 pixeli) separate printr-un spațiu (nu este trecut codul implicit de start (7) aferent primului punct).

### Bibliografie

- [1] Border Tracing – Digital Image Processing lectures, The University of Iowa, <http://www.icaen.uiowa.edu/~dip/LECTURE/Segmentation2.html#tracing>
- [2] Contour Representations – Quantitative Imaging Group, Delft University <http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Contour.html#Heading27>
- [3] G.X. Ritter, J.N. Wilson – Handbook of Computer Vision Algorithms in Image Algebra Second Edition – Chapter 10.4 Chain Code Extraction and Correlation, CRC Press, New York 2001
- [4] Chain Codes – Digital Image Processing lectures, The University of Iowa <http://www.icaen.uiowa.edu/~dip/LECTURE/Shape2.html#chaincodes>
- [5] Representation of Two-Dimensional Geometric Structures, [http://homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB/LIB/bandb8\\_12.pdf](http://homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB/LIB/bandb8_12.pdf)