

5. Etichetarea obiectelor din imagini binare

5.1. Introducere

În această lucrare este prezentat un algoritm care permite etichetarea obiectelor disjuncte dintr-o imagine binară. Acest algoritm este util pentru separarea obiectelor distincte, în scopul unor analize/măsurători ulterioare aplicate fiecărui obiect în parte.

5.2. Considerații teoretice

Rezultatul procesului de etichetare constă în asignarea unei etichete unice, reprezentate printr-un număr întreg, fiecărui obiect distinct din imagine. Toți pixelii unui obiect vor avea aceeași etichetă. În continuare se va descrie un algoritm performant pentru etichetarea obiectelor, care parcurge imaginea o singură dată.

Pașii algoritmului

1. Etichetarea pixelilor din imaginea sursă, printr-o singură trecere, și stabilirea perechilor de etichete echivalente;
2. Stabilirea claselor de etichete echivalente;
3. Înlocuirea etichetei fiecărui pixel cu eticheta atașată clasei de echivalență din care face parte.

Pasul 1. Etichetarea pixelilor din imaginea sursă și stabilirea perechilor de etichete echivalente.

Operația de etichetare, necesită mai întâi definirea tipului de conectivitate care se va folosi, iar apoi scanarea imaginii și etichetarea pixelilor conecși cu aceeași etichetă. Algoritmul prezentat folosește o conectivitate de 5 (Fig. 5.1) în care pixelul curent este în poziția X. Prin utilizarea unei astfel de conectivități imaginea va fi parcursă de sus în jos pe linii, iar în cadrul aceleiași linii de la stânga la dreapta pe coloane.

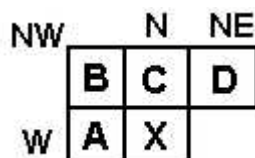


Fig. 5.1 Conectivitatea de 5 folosită

Cazul general al algoritmului de etichetare este prezentat în Fig. 5.2. Considerăm că pixelii fondului sunt albi și au valoarea diferită de "0". Pixelii aparținători obiectelor sunt negri și au valoarea egală cu "0". În Fig. 5.2 folosim următoarele notații:

- X_Label, A_Label, B_Label, C_Label, D_Label pentru etichetele atribuite pixelilor X,A,B,C respectiv D (vezi Fig. 5.1);
- NewPair(Label1,Label2) este o funcție care adaugă perechea de etichete (Label1,Label2) în lista perechilor de etichete echivalente;
- NewLabel() este o funcție care generează o nouă etichetă, respectiv incrementează cu 1 ultima etichetă generată (prima etichetă generată va avea valoarea 1).

Observație:

Etichetele pixelilor din imaginea sursă trebuie să fie memorate într-o matrice de aceeași dimensiuni cu sursa. În matricea etichetelor, pixelii fondului au valoarea etichetei egală cu 0 iar pixelii aparținători obiectelor au valorile etichetelor numere naturale mai mari ca 0.

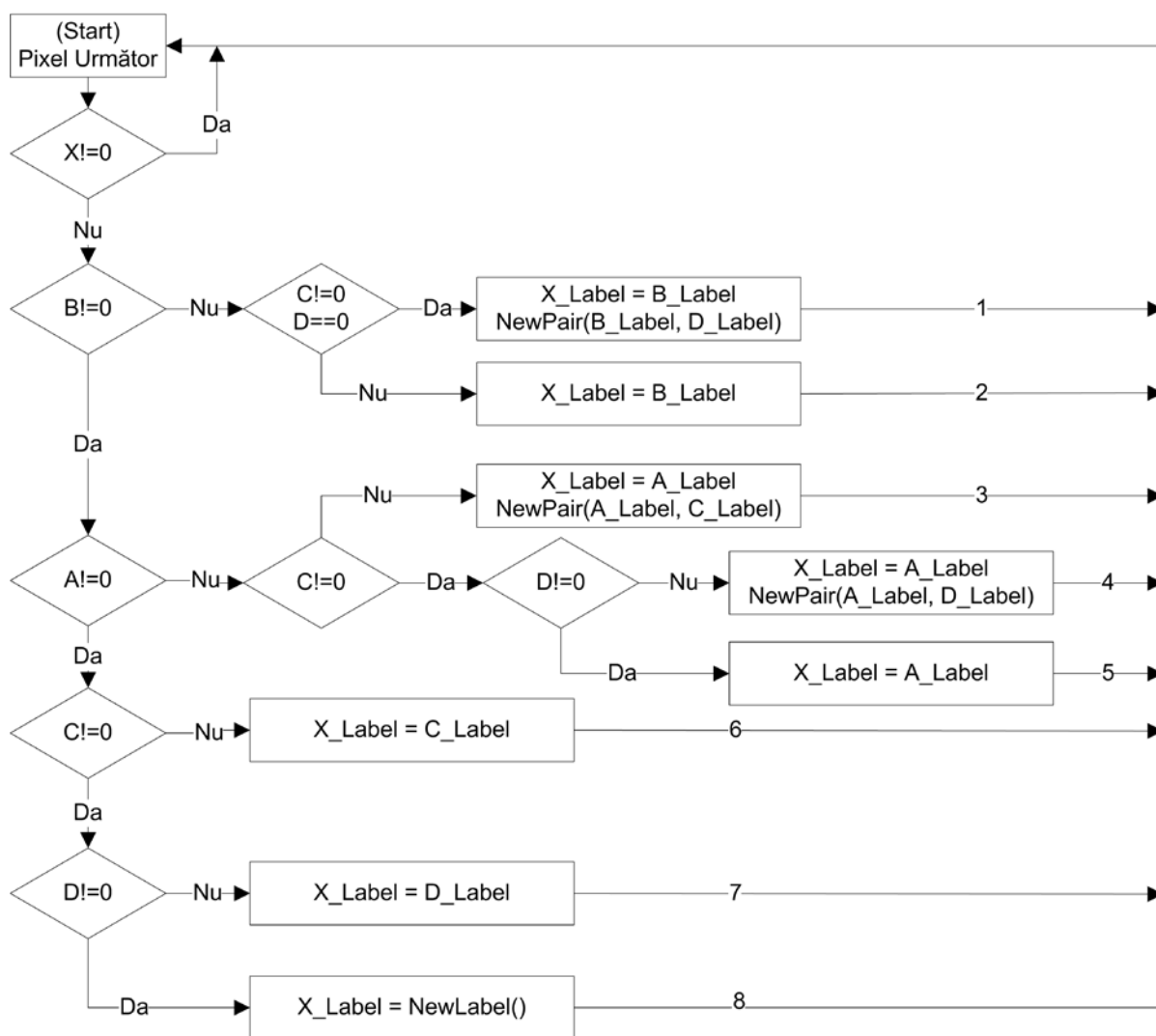


Fig. 5.2 Schema bloc a algoritmului de etichetare

Algoritmul de etichetare se aplică pentru fiecare pixel din imagine parcurgând imaginea de sus în jos pe linii și de la stânga la dreapta pe coloane. **Observație:** utilizând conectivitatea de 5 (Fig. 5.1) parcurgerea imaginii va începe de pe a doua linie și a doua coloană din partea de sus și se va termina pe ultima linie și penultima coloană întrucât se observă că pentru pixelul curent (pixelul X, Fig. 5.1) ales pentru etichetare trebuie să se poată defini (să fie în imagine) cei patru vecini din direcțiile W, NW, N, NE (pixelii A, B, C, D, Fig. 5.1). În caz contrar, vor apărea erori de depășire a capacității de memorie alocate.

Algoritmul de etichetare și de adăugare a perechilor de etichete echivalente pentru fiecare pixel din zona de aplicare este descris, în ordinea de mai jos, de următoarele cazuri:

- a) dacă pixelul curent aparține fondului, se trece la următorul pixel;
- b) dacă pixelul curent este pixel obiect, el trebuie să fie etichetat, și se disting situațiile:
 - dacă vecinii din direcțiile NW și NE sunt pixeli obiect și vecinul din direcția N nu este pixel obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția NW și se adaugă ca fiind echivalente perechea de etichete a vecinilor din direcțiile NW și NE (ramura 1, Fig. 5.2);
 - dacă vecinul din direcția NW este pixel obiect și fie vecinul din direcția N este pixel obiect sau vecinul din direcția NE nu este pixel obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția NW (ramura 2, Fig. 5.2);

- dacă vecinul din direcția NW nu este pixel obiect și vecinii din direcțiile W și N sunt pixeli obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția W și se adaugă ca fiind echivalente perechea de etichete a vecinilor din direcțiile W și N (ramura 3, Fig. 5.2);
- dacă vecinii din direcțiile NW și N nu sunt pixeli obiect și vecinii din direcțiile W și NE sunt pixeli obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția W și se adaugă ca fiind echivalente perechea de etichete a vecinilor din direcțiile W și NE (ramura 4, Fig. 5.2);
- dacă vecinul din direcția W este pixel obiect și vecinii din direcțiile NW, N și NE nu sunt pixeli obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția W (ramura 5, Fig. 5.2);
- dacă vecinii din direcțiile NW și W nu sunt pixeli obiect și vecinul din direcția N este pixel obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția N (ramura 6, Fig. 5.2);
- dacă vecinii din direcțiile NW, W și N nu sunt pixeli obiect și vecinul din direcția NE este pixel obiect atunci eticheta pixelului curent va fi egală cu eticheta vecinului din direcția NE (ramura 7, Fig. 5.2);
- dacă niciunul din vecini (din direcțiile NW, W, N și NE) nu sunt pixeli obiect atunci se va genera o etichetă nouă care se va atribui pixelului curent (ramura 8, Fig. 5.2).

Etichetarea multiplă a unui obiect poate să apară în cazul scanării secvențiale, așa cum se poate vedea în cazul unui obiect în forma de "J" (Fig. 5.3). Algoritmul descris etichetează două obiecte diferite cu etichetele 1 și 2, până când este atins pixelul notat cu X, unde se descoperă că obiectele 1 și 2 sunt conectate. În acest moment cele 2 etichete se declară echivalente prin adăugarea perechii (1,2) în lista perechilor de etichete echivalente (NewPair(1,2)).

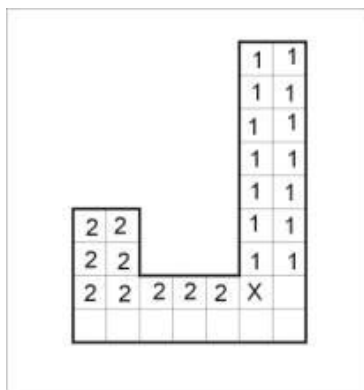


Fig. 5.3 Cazul obiectelor cu etichete multiple

Pasul 2. Stabilirea claselor de etichete echivalente.

După ce toți pixelii au fost etichetați și au fost stabilite perechile de etichete echivalente, trebuie găsite mulțimile (clasele) de etichete echivalente. Pentru a modela echivalența etichetelor putem folosi un graf neorientat în care nodurile sunt etichetele iar arcele sunt relațiile binare de echivalență stabilite în pasul anterior. Astfel, clasele de echivalență sunt subgrafurile conexe ale acestui graf.

Pentru a găsi un subgraf conex se poate folosi o căutare în lățime pornind dintr-un prim nod oarecare al subgrafului. Se pun toți vecinii primului nod într-o listă și se marchează ca fiind luați în considerare (în scopul evitării nodurilor ce au fost deja luate în considerare ca făcând parte din subgraful conex curent); pentru fiecare nod din lista respectivă, acesta este

extras din listă, și i se adaugă toți vecinii (neconsiderați încă) în lista și se marchează ca fiind luați în considerare. Procesul se repetă pentru fiecare nod adăugat în listă, până când lista devine goală. Toate nodurile trecute prin această listă vor fi etichetate cu aceeași etichetă, care va fi noua etichetă asociată acestei clase de etichete echivalente. În continuare se caută un alt nod care nu a fost considerat, și pornind de la acesta se va căuta o nouă clasă de echivalență. Procesul se continuă până când sunt considerate toate nodurile grafului. Fiecare nod singular reprezintă câte o clasă de echivalență cu un singur element.

Pentru re-etichetare, se poate utiliza un șir de întregi în care la intrarea i se găsește noua etichetă cu care va fi înlocuită vechea etichetă i , respectiv eticheta atribuită clasei de etichete echivalente. Inițial, elementele acestui șir au valoarea 0. Acest șir este utilizat și pentru ținerea evidenței etichetelor care au fost sau nu considerate: o valoare 0 înseamnă ca eticheta respectivă (nodul din graf) nu a fost luată încă în considerare.

Căutarea vecinilor unui nod presupune parcurgerea listei de perechi echivalente, și găsirea perechilor care conțin nodul curent, celălalt nod al perechii fiind vecinul căutat. Această formă de reprezentare a grafului este numită “reprezentare prin lista de perechi de adiacență”. Ea are marele dezavantaj că necesită un timp de căutare foarte mare. S-ar putea folosi o matrice de adiacență, dar aceasta va fi o matrice foarte rară deci va necesita multă memorie alocată și un timp de căutare destul de ridicat.

Pasul 3. Re-etichetarea tuturor etichetelor cu valorile corespunzătoare claselor de echivalență.

Se parcurge matricea etichetelor și se înlocuiește fiecare etichetă cu eticheta atribuită clasei respective de etichete echivalente din care aceasta face parte. Acest lucru se poate realiza pe baza șirului de întregi descris la pasul anterior.

5.3. Exemple de etichetări



Fig. 5.4 Exemple de etichetări

5.4. Detalii de implementare

Pentru etichetare, parcurgerea imaginii binare sursă se va face de sus în jos pe linii și de la stânga la dreapta pe coloane. Datorită utilizării conectivității de 5 (Fig. 5.1) parcurgerea imaginii va începe de pe a doua linie și a doua coloană din partea de sus și se va termina pe ultima linie și penultima coloană (vezi Pasul 1, secțiunea 5.2).

```
for (int i=dwHeight-2;i>=0;i--)  
    for (int j=1;j<dwWidth-1;j++)  
        if (lpSrc[i*w+j]==0)  
        {  
            // pixelul curent este negru  
            // se aplică algoritmul de etichetare și stabilire  
            // a perechilor de etichete echivalente  
            // se memorează valoarea în matricea de etichete  
            // se memorează perechile de etichete echivalente  
            // care pot apărea eventual  
        }
```

Observații:

1. Imaginea bitmap este memorată răsturnat (linia 0 din memorie corespunde primei linii din partea de jos a imaginii – așa cum este ea afișată pe ecran).
2. Pentru matricea de etichete se va aloca o matrice de tip *int* de dimensiunea imaginii.

După ce pixelii obiect au fost etichetați se determină clasele de echivalență pe baza perechilor de etichete echivalente memorate în pasul anterior (vezi Pasul 2, secțiunea 5.2).

Ultimul pas este cel al reetichetării pixelilor cu valoarea etichetei clasei de echivalență din care face parte vechea etichetă. Afișarea obiectelor etichetate cu diferite culori se poate realiza modificând paleta pentru intrările 1..254 (intrările 0 și 255 nu pot fi modificate deoarece aparțin pixelilor obiect respectiv de fond) și apoi atribui pentru pixelii aparținători fiecărui obiect o valoare din intervalul 1..254, rezultând astfel fiecare obiect colorat diferit.

```
// modificarea paletei pentru a afișa obiectele etichetate cu culori  
// diferite  
for (int k=1;k<=254;k++)  
{  
    // generarea unei nuanțe de culoare aleatoare în paletă la  
    // indexul k, 1≤k≤254  
    bmiColorsDst[k].rgbRed = randomValueRed;  
    bmiColorsDst[k].rgbGreen = randomValueGreen;  
    bmiColorsDst[k].rgbBlue = randomValueBlue;  
}
```

Observații:

1. Imaginea binară utilizată pentru etichetare trebuie să fie o imagine cu 8 biți/pixel cu paleta ordonată care conține doar pixeli negri (valoarea 0 – aparținători obiectelor) și pixeli albi (valoarea 255 – aparținători fondului)
2. Utilizând această metodă de colorare cu o nuanță diferită a obiectelor etichetate, permite existența a maxim 254 de nuanțe de culoare pentru afișarea obiectelor etichetate.

5.5. Activități practice

1. Implementați algoritmul de etichetare a obiectelor din imagini binare.
2. Adăugați la framework-ul DIBLook o funcție de procesare care să execute algoritmul de etichetare a obiectelor pe o imagine binară. Evidențiați obiectele etichetate prin afișarea lor în culori/nuanțe diferite.
3. **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 algoritmi implementați!!!**

Referințe

- [1]. Umbaugh Scot E, *Computer Vision and Image Processing*, Prentice Hall, NJ, 1998, ISBN 0-13-264599-8
- [2]. Robert M. Haralick, Linda G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Publishing Company, 1993.