

## Segmentarea imaginilor

O alta metodă de extragere a caracteristicilor din imagini se bazează pe determinarea zonelor din imagine care conțin pixeli cu caracteristici similare de strălucire, culoare sau textură.

Metodele de segmentare se împart în următoarele categorii:

### 1. Segmentarea prin divizare („splitting ”)

Se pleacă de la o zonă mare, de exemplu întreaga imagine care se sparge în mai multe regiuni. Dacă regiunile obținute nu sunt omogene, se continuă spargerea până când se obțin regiuni omogene (care satisfac un anumit criteriu de omogenitate).

### 2. Segmentarea prin divizare și unificare („split & merge ”)

Se pleacă de la regiuni de mărime medie, de exemplu pătrate de mărime fixă, se unifică regiunile similare și apoi se divizează regiunile care nu au fost unificate (nesatisfăcând criteriul de unificare).

### 3. Segmentarea prin extindere („merging”)

Pornind de la o regiune mică, de exemplu de un pixel, se dezvoltă regiunea treptat, până când criteriul de extindere nu mai este satisfăcut.

### 4. Segmentarea prin extindere și unificare

O imagine segmentată conține mai multe regiuni, pixelii fiecărei regiuni având aceeași intensitate, culoare sau textura.

### Algoritm de segmentare prin divizare recursivă

Se pornește de la premisa că întreaga imagine este omogenă. Dacă nu este adevărat, atunci imaginea este divizată în patru regiuni. Procedura de divizare este aplicată recursiv până când se obțin numai regiuni omogene.

### Implementare in C

```
void divideReg(imagine a, imagine b, int N1, int M1, int N2, int M2, int T, int *N)
{ // a este imaginea de intrare iar b este imaginea de iesire
  // (N1, M1) – (N2,M2) sunt colturile stanga –sus si dreapta-jos ale dreptunghiului care
  // incadreaza imaginea segmentata
  // T este pragul folosit in criteriul de omogenitate
  // in N se memoreaza numarul de regiuni rezultate din divizarea recursiva
  long s;
  if(!omogen(a,N1, M1, N2, M2,T) && (N2-N1) > 1 && (M2-M1) >1)
  { // se divide regiunea in 4 zone egale si se apeleaza functia pentru cele 4 zone
    divideReg(a, b, N1, M1, (N1+N2)/2, (M1+M2)/2, T, N);
    divideReg(a, b, (N1+N2)/2, M1, N2, (M1+M2)/2, T, N);
```

```

divideReg(a, b, N1, (M1+M2)/2, (N1+N2)/2, M2, T, N);
divideReg(a, b, (N1+N2)/2, (M1+M2)/2, N2, M2, T, N);
}
else
// regiune omogena; calculeaza media intensitatilor pixelilor regiunii si o asigneaza
// pixelilor regiunii in imaginea de iesire
{ (*N)++;
s=0;
for(int i=M1; i<M2; i++)
    for(int j =N1; j<N2; j++)
        s+ = a[i][j];
s/ =(N2-N1)*(M2-M1);
for(i=M1; i<M2; i++)
    for(j=N1; j<N2; j++)
        b[i][j] = (unsigned char) s;
}
return;
}

```

```

int omogen( imagine a, int N1, int M1, int N2, int M2, int T)
{ int max =0, min = Lmax; // nivelul maxim de intensitate
for(int i=M1; i<M2; i++)

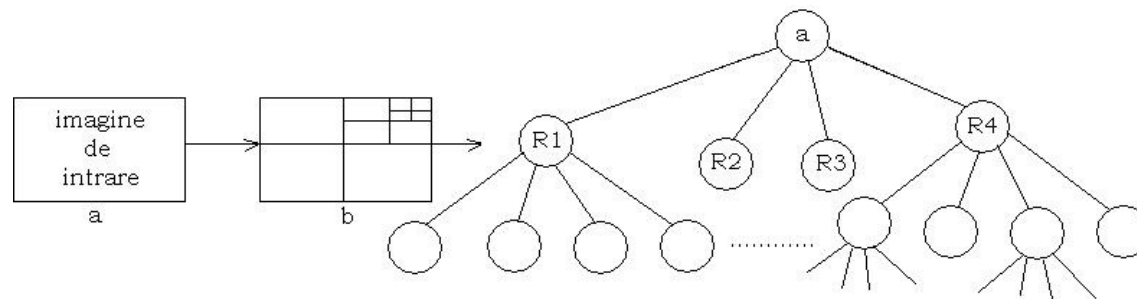
```

```

for(int j=N1; j< N2; j++)
    { if(a[i][j] < min) min = a[i][j];
      if(a[i][j] > max) max = a[i][j];
    }
if(max – min < T) return 1; // criteriul de omogenitate
else return 0;
}

```

Algoritmul de segmentare prin divizare recursivă produce o imagine segmentată care poate fi reprezentată printr-un arbore cuadric :



Fiecare nod are patru fii, corespunzători celor patru regiuni în care se face divizarea de fiecare dată. Este suficient să se memoreze în fiecare nod al arborelui intensitatea care a fost atribuită pixelilor regiunii corespunzătoare nodului.

Algoritmul are însă un **dezavantaj** major : **poate produce regiuni adiacente similare**, care nu corespund aceluiași nivel de divizare.

Astfel:

*dacă*  $P(R)$  este un predicat logic care definește regula de segmentare (de exemplu  $I_{max} - I_{min} < \text{prag}$  ),

*atunci*,

*dacă*  $P(R_i \cup R_j) = \text{TRUE}$

$\Rightarrow R_i$  și  $R_j$  sunt regiuni similare.

Regiunile adiacente similare trebuie să fie unificate.

### Segmentarea prin divizare și unificare („split & merge”)

Algoritmul „split & merge” este o procedură iterativă care include în fiecare iterație divizare și unificare :

- dacă o regiune este neomogenă ( $P(R) = \text{FALSE}$  ), ea este spartă în 4 regiuni.
- dacă două regiuni adiacente  $R_i, R_j$  sunt similare ( $P(R_i \cup R_j) = \text{TRUE}$  ), ele sunt unificate.
- algoritmul se termină atunci când nu mai este posibilă nici spargere, nici unificare.

Algoritmul este recursiv:

- dacă regiunea curentă nu este omogenă, ea este divizată în 4 și procedura este apelată recursiv.
- dacă regiunea este omogenă, se calculează media sa și se compară cu mediile regiunilor vecine (pe cele 4 laturi ale regiunii), verificând posibilele unificări. Astfel, dacă diferența dintre media regiunii curente și media unei regiuni adiacente este sub o valoare de prag, atunci cele două regiuni se unifică: se calculează media valorilor pixelilor celor două regiuni și se atribuie pixelilor celor două regiuni în imaginea de ieșire.

- dacă nici o unificare nu este posibilă, se atribuie pixelilor regiunii din imaginea de ieșire valoarea medie a pixelilor săi
- dacă mai multe unificări sunt posibile, se alege cea mai bună, adică unificarea cu regiunea vecină care are valoarea medie cea mai apropiată de valoarea medie a regiunii curente.

**Avantaj:**

Algoritmul produce regiuni mai compacte decât cel prezentat anterior (bazat numai pe divizare).

**Dezavantaj :**

Algoritmul nu produce o reprezentare prin arbore cuadric.

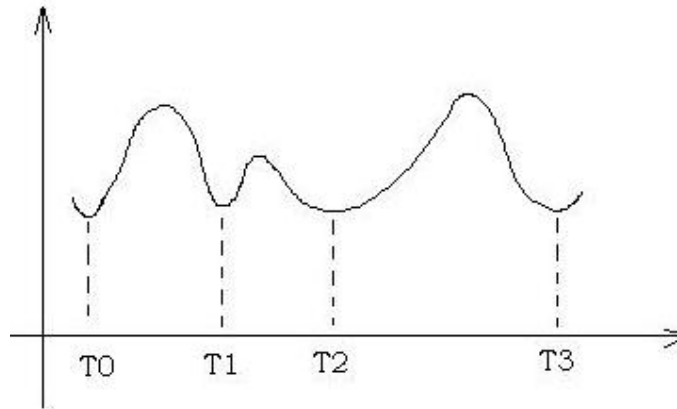
**O posibilitate :**

Se aplică algoritmul de segmentare prin divizare obținându-se arborele cuadric al imaginii.

Se unifică regiunile adiacente similare.

## Segmentarea prin divizare bazată pe praguri

1. Se aleg valorile de prag ca puncte de minim local ale funcției histogramă :



Dacă  $f(x, y)$  este funcția imagine de intrare, atunci imaginea segmentată este definită de :

$$g(x, y) = R_i \text{ dacă } T_{i-1} \leq f(x,y) < T_i \text{ cu } i = 1, N$$

$R_i$  poate fi media intensitatii pixelilor regiunii.

În multe cazuri, „lobii” histogramei nu sunt clari datorită contribuției pixelilor aflați la frontiera dintre două regiuni. Dacă se elimină sau se moderează contribuția acestor pixeli în calculul histogramei, atunci lobii vor fi mai clari. Metoda cea mai simplă de modificare a histogramei în acest scop este de a detecta fronturile din imaginea de intrare, excluzând apoi în calculul histogramei pixelii de front.

2. Divizarea în  $N$  regiuni distincte, fiecareia corespunzându-i un domeniu de intensități.

$f(x, y)$  – imaginea de intrare

$g(x,y)$  – imaginea segmentată

$$g(x,y) = \begin{array}{l} R_i, \text{daca } i * \left(\frac{L}{N}\right) \leq f(x, y) < (i+1) * \frac{L}{N} \\ R_{N-1}, \text{daca } (N-1) * \frac{L}{N} \leq f(x, y) < L \end{array}$$

$$R_i = i * \frac{L}{N}, i = 0, 1, \dots, N-2$$

De exemplu,

$$N = 4, L = 256, L/N = 64$$

$$g(x,y) = \begin{array}{l} 0, \text{daca } 0 \leq f(x, y) \leq 63 \\ 64, \text{daca } 64 \leq f(x, y) \leq 127 \\ 128, \text{daca } 128 \leq f(x, y) \leq 191 \\ 192, \text{daca } 192 \leq f(x, y) \leq 255 \end{array}$$

Dacă histograma imaginii este concentrată într-un domeniu mic de intensități, atunci segmentarea prin aplicarea unui prag uniform (metoda a 2-a) nu dă rezultate bune deoarece majoritatea pixelilor vor fi repartizați într-un număr foarte mic de regiuni.



În astfel de cazuri, segmentarea prin aplicarea de praguri se poate face după ce asupra imaginii de intrare se aplică transformarea de egalizare a histogrammei.

Dacă  $G(f(x,y))$  este transformarea imaginii de intrare prin egalizarea histogrammei, atunci imaginea segmentată este

$$g(x,y) = \begin{cases} R_i, & \text{daca, } i * \left\lceil \frac{L}{N} \right\rceil \leq G(f(x,y)) < (i+1) * \left\lceil \frac{L}{N} \right\rceil \\ R_{N-1}, & \text{daca, } (N-1) * \left\lceil \frac{L}{N} \right\rceil \leq G(f(x,y)) < L \end{cases}$$

- Tehnicile de segmentare bazate pe aplicarea de praguri garantează omogenitatea regiunilor, dar nu și conectivitatea lor, deoarece aplicarea pragurilor se face pixel cu pixel, fără să se țină cont de vecinătatea fiecărui pixel.

Vecinătatea geometrică joacă un rol important în segmentarea imaginilor. Pixelii dintr-o vecinătate au în general proprietăți statistice similare și deci aparțin aceleiași regiuni. De aceea, algoritmi de segmentare trebuie să încorporeze, dacă este posibil, atât criterii de similaritate cât și criterii de vecinătate.

- Algoritmi de spargere recursivă adaptivă precum și algoritmi de spargere și unificare produc regiuni conectate deoarece ele provin din divizarea unei regiuni.

## Segmentarea prin extindere

Se pleacă de la pixeli „germen” în jurul cărora se vor forma regiunile. Aceștia sunt de regulă aleși de către utilizator într-un mod controlat.

Plecând de la pixelii germen se formează în jurul lor regiuni adăugând iterativ pixelii vecini care satisfac criteriul de similaritate.

Segmentarea prin extindere se bazează pe :

- O regulă care descrie mecanismul de extindere
- O regulă de verificare a omogenității regiunilor după fiecare pas de extindere

Mecanismul de extindere este simplu :

În fiecare pas K:

Pentru fiecare regiune  $R_i^{(k)}$  cu  $i = 1, \dots, N$

1. Se verifică dacă există pixeli neclasificați în vecinătatea de 8 pixeli a fiecărui pixel de pe frontiera regiunii.
2. Un pixel vecin, X, este adăugat la regiune (clasificat) dacă  $P(R_i^{(k)} \cup \{X\}) = \text{TRUE}$ , unde P este criteriul de similaritate:

$$P = |\text{intensitate pixel} - \text{intensitate medie regiune}| < T \text{ unde } T \text{ este un prag dat}$$

3. Atunci cand un pixel este adaugat la regiune, se recalculeaza media regiunii

Algoritmul se termină atunci când într-o iterație numărul de pixeli clasificați nu s-a modificat (nici o regiune nu s-a mai putut extinde).

Pixelii germen pot fi aleși interactiv de către utilizator sau automat ca pixeli a căror intensitate corespunde unuia dintre vârfurile histogramei.

**Observație:**

Performanța algoritmului depinde mult de alegerea pixelilor germen.

### Unificarea regiunilor

In urma segmentării prin extindere („creșterea” regiunilor) pot rezulta regiuni adiacente cu proprietăți statistice similare. Ele pot fi reunite printr-un algoritm de unificare („merging”).

Ca proprietăți statistice se pot folosi media și deviația standard. Pentru o regiune  $R_i$  de  $n$  pixeli, acestea sunt definite astfel :

$$m_i = \frac{1}{n} \sum_{(k,l) \in R_i} f(k,l) \quad - \text{ media}$$

$$\sigma_i = \sqrt{\frac{1}{n} \sum_{(k,l) \in R_i} (f(k,l) - m_i)^2} \text{ - deviația standard}$$

Două regiuni adiacente, R1 și R2, pot fi unificate dacă mediile lor aritmetice sunt suficient de apropiate

$$|m_1 - m_2| < K * \sigma_i, \text{ cu } i = 1, 2$$

Unificarea poate fi înglobată în mecanismul de extindere.

### Algoritm de extindere si unificare

Nu sunt disponibile informații despre imagine (nu se dau germei). Imaginea este parcursă pe rânduri și fiecare pixel este atribuit unei regiuni. Atunci când s-a ajuns în pixelul (k,l), toți pixelii de pe rândurile anterioare,  $0 \leq j \leq l-1$  au fost deja atribuiți. Deci pixelul (k,l) are regiuni înconjurătoare dacă nu este primul ( $k = 0, l = 0$ ).

Mai întâi se încearcă atribuirea pixelului (k, l) la una dintre regiunile vecine. Dacă nu poate fi atribuit unei regiuni existente, este atribuit unei regiuni noi (se creează o nouă regiune).

Performanța algoritmului este dependentă de regula de unificare, adică de modul de atribuire a pixelului curent la o regiune existentă :  $P(R_i \cup (k,l))$

Regula de asignare se poate baza pe media aritmetică și deviația standard. Astfel, asignarea este permisă dacă intensitatea pixelului este apropiată de media  $M_i$  a unei regiuni vecine:

$$|f(k,l) - M_i| \leq T_i(k,l)$$

Pragul  $T_i$  poate fi diferit de la regiune la regiune și de la pixel la pixel. El poate fi ales astfel :

$$T_i(k,l) = (1 - \frac{\sigma'_i}{M_i}) * T$$

unde  $M'_i$  și  $\sigma'_i$  sunt media și deviația standard ale regiunii  $i$  unificate cu pixelul  $(k,l)$ , iar  $T$  este pragul global.

Dacă pentru nici o regiune nu este satisfăcută condiția de asignare, pixelul  $(k,l)$  este atribuit unei regiuni noi.

Dacă condiția este satisfăcută pentru mai multe regiuni vecine, pixelul este atribuit regiunii  $R_i$  pentru care diferența

$$|f(k,l) - M_i| \text{ este minimă.}$$

- **Creșterea regiunilor este dependentă de parametrul  $T$ .** Dacă este mic, pragul  $T_i(k,l)$  este mic pentru toate regiunile și extinderea regiunilor este dificilă. Vor rezulta multe regiuni mici. Dacă este mare, pot rezulta regiuni puțin omogene și cu deviații standard mari. Pragul  $T_i(k,l)$  depinde de asemenea de raportul  $\frac{\sigma'_i}{M_i}$ . Pentru regiuni omogene, acest raport tinde către zero și  $T_i$  tinde către  $T$ . Deci,  $T$  desemnează diferența maximă permisă între intensitatea unui pixel și media unei regiuni foarte omogene. Pentru regiuni mai puțin omogene, raportul  $\sigma'_i / M_i$  este mai mare, pragul  $T_i$  devine mai mic și extinderea unor astfel de regiuni este restricționată.