

Lucrarea 10

Segmentarea imaginilor

BREVIAR TEORETIC

Segmentarea reprezintă împărțirea imaginii în zone de interes, după anumite criterii. Fiecărui pixel i se va atribui o valoare, 0 sau 1, reprezentând apartenența acestuia la o anumită zonă sau regiune de interes. De regulă, segmentarea urmărește extragerea, identificarea sau recunoașterea unui anumit obiect dintr-o imagine. Zonele sau regiunile care alcătuiesc o imagine poartă numele de segmente. Pentru o imagine $f(m, n)$, segmentarea reprezintă împărțirea lui f într-un număr N de zone $f_i(m, n)$, cu $i = 1..N$, ca în Figura 10.1. Aceste segmente se numesc complete, dacă au următoarele proprietăți:

- $f_i \cap f_j = \emptyset$ pentru $i \neq j$,
- $\bigcup_{i=1}^N f_i = f$,
- segmentul f_i să fie compact, pentru $\forall i$,
- pentru $\forall i$, un anumit criteriu de uniformitate $E(f_i)$ este satisfăcut,
- pentru $\forall i, j$, criteriul de uniformitate pentru $f_i \cup f_j$ nu este satisfăcut.

Metodele de segmentare a imaginilor se pot clasifica în:

- metode de segmentare orientate pe regiuni
- metode de segmentare orientate pe contururi

10.1 Segmentarea orientată pe regiuni

În general, operația de segmentare orientată pe regiuni urmărește extragerea din imagine a zonelor (regiunilor) ocupate de diversele obiecte prezente

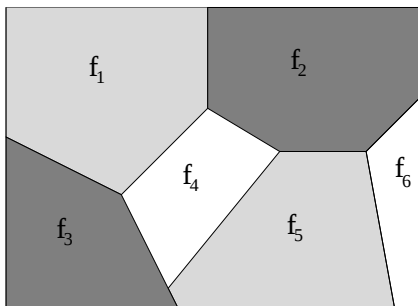


Figura 10.1: Exemplu teoretic de segmentare.

în scenă. Un obiect se definește ca o entitate caracterizată de un set de parametri ale căror valori nu se modifică în diferitele puncte ce aparțin entității considerate. Unul dintre cei mai simpli parametri de definiție este nivelul de gri al pixelului. Dacă nivelul de gri caracterizează în mod suficient obiectele din imagine, atunci histograma imaginii va prezenta o structură de moduri dominante - adică de intervale de nivele de gri ce apar cu probabilitate mai mare. Fiecare mod al histogramei va reprezenta câte un obiect sau o categorie de obiecte.

10.1.1 Prăguirea histogramei

Separarea modurilor histogramei, și deci identificarea obiectelor din imagine, se face prin alegerea unor nivele de gri, numite praguri de segmentare. De obicei aceste praguri se aleg ca fiind corespunzătoare minimelor locale ale histogramei. Din imaginea inițială f se construiește o imagine de etichete g (imagine etichetată), conform unei transformări de forma:

$$g(m, n) = \begin{cases} E_0, & 0 \leq f(m, n) < T_{K-1} \\ E_K, & T_{K-1} \leq f(m, n) < L \end{cases} \quad (10.1)$$

În cazul unei histogramă bimodale (care conține două moduri dominante), ca cea din Figura 10.2(a), transformarea de mai sus devine:

$$g(m, n) = \begin{cases} E_0, & 0 \leq f(m, n) < T \\ E_1, & T \leq f(m, n) < L \end{cases} \quad (10.2)$$

asemănătoare cu operația de binarizare. O astfel de histogramă bimodală este caracteristică imaginilor ce conțin un singur obiect sau mai multe obiecte de același fel, pe un fundal uniform. De exemplu, o imagine care conține un scris negru pe un fond alb.

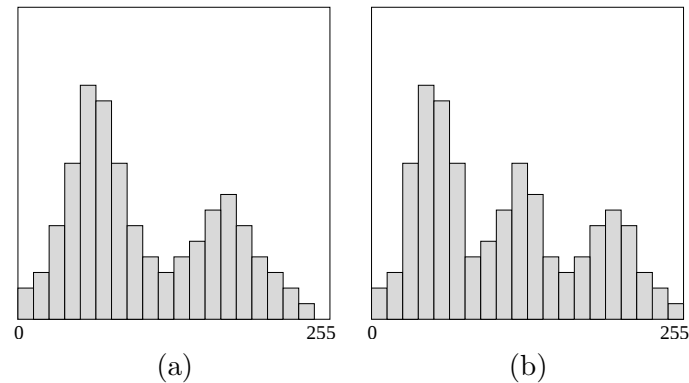


Figura 10.2: Histogramă: (a) bimodală; (b) cu trei moduri dominante.

10.1.2 Segmentarea prin creștere de regiuni

Principiul pe care se bazează creșterea regiunilor este următorul: se aleg în imagine pixeli reprezentativi pentru fiecare obiect individual, pe baza cărora are loc un proces de aglomerare a pixelilor vecini acestora, ce au aceleași proprietăți cu pixelii reprezentativi. În urma acestui proces de aglomerare se obțin zone de pixeli cu aceleași caracteristici.

Procesul se oprește în momentul în care fiecare pixel al imaginii a fost alocat unei regiuni. Metoda are două etape esențiale: alegerea punctelor de start (puncte inițiale), numite *germeni* sau *semințe*, și creșterea propriu-zisă a regiunilor. Numărul final de regiuni rezultate este egal cu numărul de germeni aleși inițial pentru creștere. În principiu, este de dorit ca fiecare obiect individual aflat în imagine, să fie marcat de un germene. Dacă în interiorul unui obiect se găsesc mai mulți germeni, pentru fiecare dintre ei va fi crescută o regiune, fapt ce duce la o segmentare artificială nedorită a obiectului respectiv. Acest neajuns poate fi corectat printr-o etapă de fuziune a regiunilor adiacente ce au proprietăți asemănătoare.

Dacă în interiorul unui obiect nu este ales nici un germene, atunci obiectul respectiv va fi inclus în regiunile ce cresc pornind de la germeni din vecinătatea sa.

10.2 Segmentarea orientată pe contururi

Într-o imagine, variațiile de nivel ale pixelilor reprezintă schimbări ale proprietăților fizice sau geometrice ale obiectelor ce compun scena. Într-un număr mare de cazuri, aceste variații de intensitate corespund frontierelor (contururilor) regiunilor determinate de obiectele dintr-o imagine.

10.2.1 Tehnicile de gradient

Principiul acestor metode constă în definirea punctelor de contur ca fiind acei pixeli ai imaginii pentru care apar schimbări abrupte ale nivelului de gri. Măsurarea acestor variații se face prin operatori derivativi de tip gradient. Derivata imaginii pe direcția r , ce face unghiul θ cu orizontala, este dată de combinația liniară a derivatelor parțiale pe direcțiile orizontală și verticală:

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = \frac{\partial f}{\partial x} \cos\theta + \frac{\partial f}{\partial y} \sin\theta \quad (10.3)$$

$$\frac{\partial f}{\partial r} = f_x \cos\theta + f_y \sin\theta \quad (10.4)$$

Valoarea maximă a acestei derivate, calculate după unghiul θ este determinată de ecuația:

$$\frac{\partial}{\partial \theta} \left(\frac{\partial f}{\partial r} \right) = -f_x \sin\theta + f_y \cos\theta = 0 \quad (10.5)$$

care are soluția:

$$\theta = \arctan \left(\frac{f_x}{f_y} \right) \quad (10.6)$$

Pe această direcție, modulul gradientului este:

$$\left(\frac{\partial f}{\partial r} \right)_{max} = \sqrt{f_x^2 + f_y^2} \quad (10.7)$$

Din punct de vedere practic, implementarea acestei metode implică calcularea, pentru fiecare pixel al imaginii, a derivatelor parțiale f_x și f_y , calcularea modulului gradientului maxim și a direcției acestuia. Valoarea gradientului maxim din fiecare pixel al imaginii este apoi comparată o valoare de prag: dacă pragul este depășit, atunci pixelul este considerat a fi pixel de contur. Realizarea derivatelor parțiale după direcțiile orizontală și verticală implică formularea discretă a lui f_x și f_y :

$$f_x = \frac{\partial f}{\partial x} = \frac{\Delta f(m, n)}{\Delta m} \quad (10.8)$$

$$f_y = \frac{\partial f}{\partial y} = \frac{\Delta f(m, n)}{\Delta n} \quad (10.9)$$

Aceste derivate parțiale discrete pot avea mai multe implementări:

$$f_x = f(m, n) - f(m + 1, n) \quad f_y = f(m, n) - f(m, n + 1) \quad (10.10)$$

$$f_x = f(m-1, n) - f(m, n) \quad f_y = f(m, n-1) - f(m, n) \quad (10.11)$$

$$f_x = f(m-1, n) - f(m+1, n) \quad f_y = f(m, n-1) - f(m, n+1) \quad (10.12)$$

Aceste expresii nu reprezintă altceva decât combinații liniare ale valorilor unor pixeli din imagine, situați în vecinătatea pixelului curent din poziția (m, n) . Prin urmare, se pot implementa folosind următoarele măști de filtrare:

$$W_x = \begin{pmatrix} 1^* & -1 \end{pmatrix} \quad W_y = \begin{pmatrix} 1^* \\ -1 \end{pmatrix}$$

$$W_x = \begin{pmatrix} 1 & -1^* \end{pmatrix} \quad W_y = \begin{pmatrix} 1 \\ -1^* \end{pmatrix}$$

$$W_x = \begin{pmatrix} 1 & 0^* & -1 \end{pmatrix} \quad W_y = \begin{pmatrix} 1 \\ 0^* \\ -1 \end{pmatrix}$$

unde prin $*$ am marcat originea măștii de filtrare. O simplificare uzuală practică este dată de aproximarea:

$$\left(\frac{\partial f}{\partial r} \right)_{max} \approx |f_x| + |f_y| \quad (10.13)$$

Folosirea măștilor de derivare pe verticală și orizontală prezentate are însă serioase neajunsuri: dimensiunea lor mică face ca rezultatele să fie extrem de sensibile la zgomote. În aceste condiții a apărut ideea naturală de a combina filtrarea de derivare cu o filtrare de netezire, care să reducă efectele zgomotului. Ceea ce rezultă pentru operatorii de derivare orizontală și verticală sunt măștile:

$$W_x = \begin{pmatrix} 1 & 0 & -1 \\ c & 0^* & -c \\ 1 & 0 & -1 \end{pmatrix} \quad W_y = \begin{pmatrix} 1 & c & 1 \\ 0 & 0^* & 0 \\ -1 & -c & -1 \end{pmatrix} \quad (10.14)$$

Prin particularizarea valorilor constantei c se pot obtine diverse tipuri de operatori de extragere de contur clasici: Prewitt ($c = 1$), Izotrop ($c = \sqrt{2}$) sau Sobel ($c = 2$). În Figura 10.3 puteți observa efectele extragerii de contur folosind operatorul Sobel.



Figura 10.3: Filtrul Sobel: (a) imaginea originală; (b) imaginea rezultată.

10.2.2 Operatorii compas

Un operator compas este definit de un număr de măști de derivare, corespunzătoare unor filtrări liniare, pe direcțiile principale (verticală, orizontală și cele două diagonale), în cele două sensuri. Compasul clasic are $D = 8$ măști de filtrare, fiecare dintre ele realizând o derivare după o direcție multiplu de 45° . Un exemplu de măști de derivare direcțională sunt măștile următoare, indexate după direcția geografică pe care se calculează derivata:

$$\begin{aligned}
 W_N &= \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} & W_{NV} &= \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \\
 W_V &= \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} & W_{SV} &= \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \\
 W_S &= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} & W_{SE} &= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} \\
 W_E &= \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} & W_{NE} &= \begin{pmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}
 \end{aligned}$$

Dupa ce se vor calcula, pentru fiecare pixel în parte, cele opt valori ale gradientilor, corespunzătoare celor opt măști de derivare:

$$f_1(m, n) \quad f_2(m, n) \quad \dots \quad f_8(m, n) \quad (10.15)$$

se va determina valoarea maximă dintre aceste opt valori. Această valoare va fi comparată cu valoarea de prag, iar în cazul în care este mai mare, pixelul respectiv va fi considerat pixel de contur.

10.2.3 Identificarea trecerilor prin zero ale celei de-a doua derivate

Unul dintre principalele dezavantaje ale metodelor de gradient este precizia slabă de localizare a conturului în condițiile unei pante puțin abrupte (tranziții slabe, graduale) a acestuia. Derivata a doua poate fi însă folosită pentru a marca centrul tranziției (trecerea sa prin zero). Operatorul bazat pe trecerea prin zero a derivatei secunde este operatorul “zero-crossing”. În cazul imaginilor, derivata secundă trebuie luată în considerare după ambele direcții, combinate în laplacian:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (10.16)$$

În continuare sunt prezentate trei măști ce implementează o derivată secundă bidirecțională (operator Laplace):

$$\begin{pmatrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{8} & 1 & -\frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{pmatrix}$$

Pentru fiecare pixel din imagine se va calcula derivata a doua și dacă aceasta este zero, atunci pixelul este considerat ca fiind pixel de contur.

DESFĂȘURAREA LUCRĂRII

Problema 1. Implementați segmentarea prin praguirea histogramei. Pentru aceasta veti folosi codul de la lucrarea nr. 4 (egalizarea histogramei) și imaginea `tools.bmp`.

Problema 2. Observați extragerea contururilor prin tehnici de gradient, folosind un operator Prewitt.

```
void ImageViewer :: extragere_contururi( void )
{
    int i, j;
    int k, l;
    int w, h;
```

```

int g, g1, g2;
int h1[ 3 ][ 3 ], h2[ 3 ][ 3 ];

//operatori Prewitt
h1[0][0] = -1;  h1[0][1] = 0;  h1[0][2] = 1;
h1[1][0] = -1;  h1[1][1] = 0;  h1[1][2] = 1;
h1[2][0] = -1;  h1[2][1] = 0;  h1[2][2] = 1;

h2[0][0] = -1;  h2[0][1] = -1; h2[0][2] = -1;
h2[1][0] = 0;   h2[1][1] = 0;   h2[1][2] = 0;
h2[2][0] = 1;   h2[2][1] = 1;   h2[2][2] = 1;

w = image.width();
h = image.height();

QImage image_ext( w, h, 32, 0, QImage::IgnoreEndian );

for( i = 1; i < w - 1; i++ )
    for( j = 1; j < h - 1; j++ )
    {
        g1 = 0;
        g2 = 0;

        for( k = -1; k < 2; k++ )
            for( l = -1; l < 2; l++ )
            {
                g1 += h1[ k + 1 ][ l + 1 ]
                    * qRed( image.pixel( i+k, j+l ) );
                g2 += h2[ k + 1 ][ l + 1 ]
                    * qRed( image.pixel( i+k, j+l ) );
            }

        g = abs( g1 ) + abs( g2 );

        if( g > 100 )
            image_ext.setPixel( i, j, qRgb(255,255,255) );
        else
            image_ext.setPixel( i, j, qRgb(0,0,0) );
    }

image = image_ext;
pm = image;
update();
}

```

Problema 3. Implementați operatorul Izotrop de extragere a contururilor.

Problema 4. Implementați operatorul Sobel de extragere a contururilor.

Problema 5. Implementați operatorul compas dat drept exemplu.

Problema 6. Implementați extragerea de contururi prin identificarea trecerilor prin zero ale derivatei secunde.

Problema 7. Implementați creșterea de regiuni, pornind de la un singur germene. Veți folosi pentru aceasta imaginea `tools.bmp`.

