4. Trăsături geometrice ale obiectelor binare

4.1. Introducere

În această lucrare sunt prezentate câteva trăsături importante ale obiectelor binare precum şi modul de calcul al acestora. Trăsăturile prezentate sunt aria, centrul de masă, axa de alungire, perimetrul, factorul de subţiere al obiectului, elongaţia şi proiecţiile obiectului binar.

4.2. Consideraţii teoretice

În urma operaţiilor de segmentare şi etichetare a obiectelor din imaginile digitale se obţine o imagine cu obiecte care pot fi referite distinct.

Obiectul ‘i’ poate fi descris în imagine de următoarea funcţie:

$$I_i(r,c) = \begin{cases} 1, & \text{dacă } I(r,c) \in \text{obiectului etichetat 'i'} \\ 0 & \text{în celelalte cazuri} \end{cases}$$

$$\text{unde } r \in [0...Height-1] \text{ şi } c \in [0...Width-1]$$

Trăsăturile geometrice ale obiectelor se pot clasifica în următoarele două categorii:

- de poziţionare şi orientare: centrul de masă, aria, perimetrul, axa de alungire;
- de formă: factorul de aspect, factorul de subţiere, numărul Euler, proiecţiile, diametrele Feret ale obiectelor

4.2.1. Aria

$$A_i = \sum_{r=0}^{H-1}\sum_{c=0}^{W-1} I_i(r,c)$$

Aria A_i este măsurată în pixeli şi indică mărimea relativă a obiectului.

4.2.2. Centrul de masă

$$\overline{r_i} = \frac{1}{A_i}\sum_{r=0}^{H-1}\sum_{c=0}^{W-1} rI_i(r,c)$$

$$\overline{c_i} = \frac{1}{A_i}\sum_{r=0}^{H-1}\sum_{c=0}^{W-1} cI_i(r,c)$$

Aceste formule corespund liniei şi respectiv coloanei centrului obiectului. Această proprietate ajută la localizarea obiectului într-o imagine bidimensională.

4.2.3. Axa de alungire (axa de inerţie minimă / momentul de inerţie de ordin 2)

$$\tan(2\varphi_i) = \frac{2\sum_{r=0}^{H-1}\sum_{c=0}^{W-1}(r-\overline{r_i})(c-\overline{c_i})I_i(r,c)}{\sum_{r=0}^{H-1}\sum_{c=0}^{W-1}(c-\overline{c_i})^2 I_i(r,c) - \sum_{r=0}^{H-1}\sum_{c=0}^{W-1}(r-\overline{r_i})^2 I_i(r,c)}$$

Dacă atât numărătorul cât şi numitorul fracţiei din formula de mai sus sunt zero atunci obiectul prezintă simetrie circulară, orice dreaptă ce trece prin centrul de greutate fiind axă de simetrie.

Astfel, pentru aflarea unghiului trebuie aplicată funcţia arctangentă. Această funcţie este definită pe domeniul (-∞, +∞) şi are valori în domeniul (-π/2, π/2). Evaluarea acestei funcţii devine instabilă atunci când numitorul fracţiei se apropie de 0. Totodată, semnele numitorului şi numărătorului sunt importante pentru determinarea cadranului corect în care se află rezultatul, dar funcţia arctangentă nu face diferenţa între direcţii diametral opuse. Din acest motiv se recomandă folosirea funcţiei „atan2", funcţie care ia ca argumente numărătorul şi respectiv numitorul unei astfel de fracţii, şi returnează un rezultat în domeniul (-π, π).

Această proprietate dă informaţii despre orientarea obiectului. Axa corespunde direcţiei în jurul căreia obiectul (văzut ca o suprafaţă plană de grosime constantă) se poate roti cel mai uşor (are momentul cinetic minim).

După ce unghiul φ_i a fost determinat, se verifică corectitudinea valorii găsite prin trasarea axei de alungire. Axa de alungire va corespunde liniei care trece prin centrul de masă al obiectului și care face unghiul φ_i cu axa Ox.

4.2.4. Perimetrul

Perimetrul obiectului ne ajută să determinăm poziţia obiectului în spaţiu şi detalii despre forma lui. Perimetrul poate fi determinat prin numărarea pixelilor de pe contur (pixeli cu valoarea 1 care au cel puţin un pixel vecin de valoare 0).

O primă variantă de detecţie a conturului ar fi scanarea imaginii, linie cu linie, şi numărarea pixelilor din obiect, care îndeplinesc condiţia mai sus menţionată. Ca principal dezavantaj al acestei metode este faptul că nu putem face distincţie între conturul exterior şi eventualele contururi interioare (datorate găurilor din obiect). Deoarece pixelii din imaginile digitale sunt distribuiţi după un rastru dreptunghiular, lungimile curbelor şi ale dreptelor oblice nu pot fi estimate corect prin numărarea pixelilor. O primă corecţie ar fi înmulţirea perimetrului rezultat la algoritmul precedent cu π/4. Există şi alte metode de corecţie a lungimilor care ţin seama de tipul de vecinătate folosit (V4, V8 etc.)

Altă metodă de detecţie a conturului unui obiect ar fi detecţia muchiilor sale folosind un algoritm de detecţie consacrat, subţierea acestora la grosime unitară şi numărarea pixelilor de muchie rezultaţi.

Metodele de tip "chain-codes" sunt metode mai complexe de detecţie a conturului şi oferă acurateţe mai mare.

4.2.5. Factorul de subţiere al obiectului (thinness ratio)

$$T = 4\pi\left(\frac{A}{P^2}\right)$$

Această funcţie are valoarea maximă 1, care corespunde cercului. Această proprietate poate fi folosită pentru a vedea cât de rotund este un obiect. Cu cât este mai aproape de 1 cu atât obiectul este mai rotund. Această valoare dă şi informaţii despre regularitatea obiectului.

Obiectele cu contur regulat au o valoare a raportului mai mare decât obiectele cu contur neregulat. Valoarea 1/T se numeşte factor de neregularitate al obiectului (sau factor de compactitate).

4.2.6. Elongaţia (factorul de aspect) ("aspect ratio"/elongaţie/excentricitate)

Această mărime se determină prin scanarea imaginii şi determinarea valorilor minime şi maxime ale liniilor şi coloanelor ce formează dreptunghiul circumscris obiectului.

$$R = \frac{c_{\max} - c_{\min} + 1}{r_{\max} - r_{\min} + 1}$$

4.2.7. Proiecţiile obiectului binar

Proiecţiile ne dau informaţii despre forma obiectului. Proiecţia orizontală este egală cu suma pixelilor pe linii, iar proiecţia verticală este egală cu suma pixelilor pe coloane.

$$h_i(r) = \sum_{c=0}^{W-1} I_i(r,c) \qquad v_i(c) = \sum_{r=0}^{H-1} I_i(r,c)$$

Proiecţia este utilă în programele de recunoaştere a scrisului unde obiectul care ne interesează poate fi normalizat.

4.3. Detalii de implementare

Pentru a face distincţie între diversele obiecte prezente în imagine vom considera că fiecare dintre ele este desenat cu o culoare diferită. Aceste culori pot fi rezultatul unei operaţii prealabile de etichetare sau pot fi generate manual (vezi Fig. 4.1).

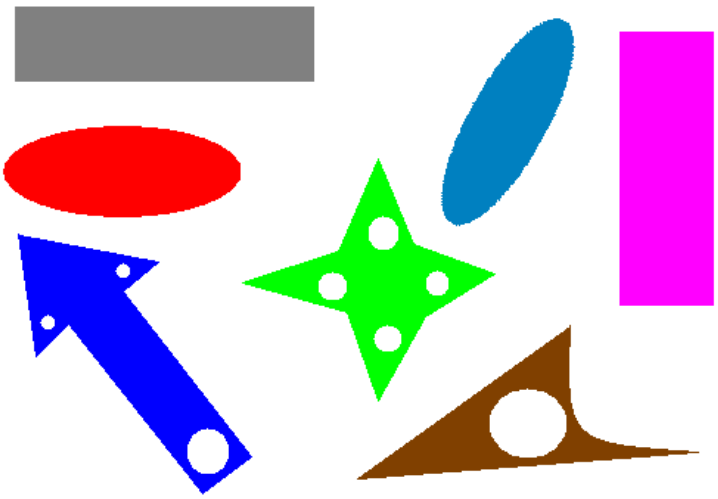

Fig. 4.1 Exemplu de imagine etichetată pe care pot fi testaţi algoritmii descrişi

Există mai multe abordări pentru implementarea extragerii proprietăților geometrice:

4.3.1. Calcularea proprietăților geometrice pentru toate obiectele din imagine, ca o singură operație

Se vor identifica pixelii fiecărui obiect pe baza etichetei (culorii) unice și se vor calcula trăsăturile geometrice. Se aplică acest procedeu pentru fiecare obiect etichetat din imagine.

4.3.2. Calcularea proprietăților geometrice a unui anumit obiect, selectat în prealabil cu ajutorul mouse-ului

Utilizatorul va selecta obiectul dorit printr-un *click* pe suprafața lui. Ca răspuns la această acţiune se vor afișa la consolă trăsăturile geometrice dorite.

Pentru a adăuga o metodă *handler* pentru evenimentul de *click* se va folosi funcția `setMouseCallback` din OpenCV care are rolul de a seta un *handler* pentru mouse pentru o anumită fereastră.

```
void setMouseCallback(const string& winname, MouseCallback onMouse, void* userdata=0)
```

`winname` – numele ferestrei,
`onMouse` – numele funției *callback* care va fi apelată în momentul când apare un eveniment de mouse in fereastra `winname`,
`userdata` – parametru opțional care poate fi transmis funcției *callback*

Se va implementa și funcția `onMouse` în care se vor implementa operațiile de calcul al trăsăsturilor.

```
void onMouse (int event, int x, int y, int flags, void* param)
```

`event` - este evenimentul de mouse, care poate avea următoarele valori:

- EVENT_MOUSEMOVE
- EVENT_LBUTTONDOWN
- EVENT_RBUTTONDOWN
- EVENT_MBUTTONDOWN
- EVENT_LBUTTONUP
- EVENT_RBUTTONUP
- EVENT_MBUTTONUP
- EVENT_LBUTTONDBLCLK
- EVENT_RBUTTONDBLCLK
- EVENT_MBUTTONDBLCLK

`x, y` – sunt coordonatele x și y la care s-a produs evenimentul,
`flags` – corespunde unor condiții specifice atunci când evenimentul se produce,
`param` – sunt parametrii utilizator care se transmit din funcția `setMouseCallback`.

În framework-ul *OpenCVApplication_basic* este prezentat un exemplu concret de *handler* la evenimentele de mouse în funcția `testMouseClick()`.

Pentru trasarea axei de alungire, folosiți funcția `line` din OpenCV:

```
void line( Mat img,  Point pStart, Point pEnd, Scalar color, int thickness )
```

`img` – imagine în care se face desenarea
`pStart`, `pEnd` –cele două puncte între care se trasează linia
`color` – culoarea liniei
`thickness` – grosimea liniei

4.4. Activităţi practice

1. Pentru un anumit obiect, selectat printr-un *click*, dintr-o imagine etichetată calculați aria, centrul de masă, axa de alungire, perimetrul, factorul de subţiere şi elongaţia (factorul de aspect).
    a. Afișați valorile obținute la consolă
    b. Într-o imagine separată (copie a imaginii sursă):
        - Desenați punctele de contur ale obiectului selectat
        - Desenați centrul de masă al obiectului selectat
        - Afișați axa de alungire a obiectului selectat folosind funcția *line* din OpenCV
    c. Calculați și afișați într-o imagine separată (copie a imaginii sursă) proiecțiile obiectului selectat
2. Creați o funcție nouă care, având ca și imagine sursă o imagine etichetată, să păstreze în imaginea destinație doar acele obiecte care:
    a. au *aria* < *TH_arie*
    b. și au o anumită orientare *phi*, unde *phi_LOW* < *phi* < *phi_HIGH*

    unde *TH_arie*, *phi_LOW*, *phi_HIGH* sunt citite de la tastatură.
3. **Salvaţi-vă ceea ce aţi lucrat. Utilizaţi aceeaşi aplicaţie în laboratoarele viitoare. La sfârşitul laboratorului de procesare a imaginilor va trebui să prezentaţi propria aplicaţie cu algoritmii implementaţi!!!**

Referinţe

[1]. Umbaugh Scot E, *Computer Vision and Image Processing*, Prentice Hall, NJ, 1998, ISBN 0-13-264599-8