4. Trăsături geometrice ale obiectelor binare

4.1. Introducere

În această lucrare sunt prezentate câteva trăsături importante ale obiectelor binare precum și modul de calcul al acestora. Trăsăturile prezentate sunt aria, centrul de masă, axa de alungire, perimetrul, factorul de subțiere al obiectului, elongația și proiecțiile obiectului binar.

4.2. Considerații teoretice

În urma operațiilor de segmentare și etichetare a obiectelor din imaginile digitale se obține o imagine cu obiecte care pot fi referite distinct.

Obiectul 'i' poate fi descris în imagine de următoarea funcție:

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

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

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

- de poziționare și orientare: centrul de masă, aria, perimetrul, axa de alungire;
- de formă: factorul de aspect, factorul de subțiere, numărul Euler, proiecțiile, 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 și 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 și respectiv coloanei centrului obiectului. Această proprietate ajută la localizarea obiectului într-o imagine bidimensională.

4.2.3. Axa de alungire (axa de inerție minimă / momentul de inerție 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 și numitorul fracției 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 funcția arctangentă. Această funcție este definită pe domeniul (-∞, +∞) și are valori în domeniul (-π/2, π/2). Evaluarea acestei funcții devine instabilă atunci când numitorul fracției se apropie de 0. Totodată, semnele numitorului și numărătorului sunt importante pentru determinarea cadranului corect în care se află rezultatul, dar funcția arctangentă nu face diferența între direcții diametral opuse. Din acest motiv se recomandă folosirea funcției „atan2", funcție care ia ca argumente numărătorul și respectiv numitorul unei astfel de fracții, și returnează un rezultat în domeniul (-π, π).

Această proprietate dă informații despre orientarea obiectului. Axa corespunde direcției în jurul căreia obiectul (văzut ca o suprafață plană de grosime constantă) se poate roti cel mai ușor (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 poziția obiectului în spațiu și detalii despre forma lui. Perimetrul poate fi determinat prin numărarea pixelilor de pe contur (pixeli cu valoarea 1 care au cel puțin un pixel vecin de valoare 0).

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

Altă metodă de detecție a conturului unui obiect ar fi detecția muchiilor sale folosind un algoritm de detecție consacrat, subțierea acestora la grosime unitară și numărarea pixelilor de muchie rezultați.

Metodele de tip "chain-codes" sunt metode mai complexe de detecție a conturului și oferă acuratețe mai mare.

4.2.5. Factorul de subțiere al obiectului (thinness ratio)

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

Această funcție 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 și informații despre regularitatea obiectului.

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

4.2.6. Elongația (factorul de aspect) ("aspect ratio"/elongație/excentricitate)

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

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

4.2.7. Proiecțiile obiectului binar

Proiecțiile ne dau informații despre forma obiectului. Proiecția orizontală este egală cu suma pixelilor pe fiecare linie, iar proiecția verticală este egală cu suma pixelilor pe fiecare coloană.

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

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

4.3. **Detalii de implementare**

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

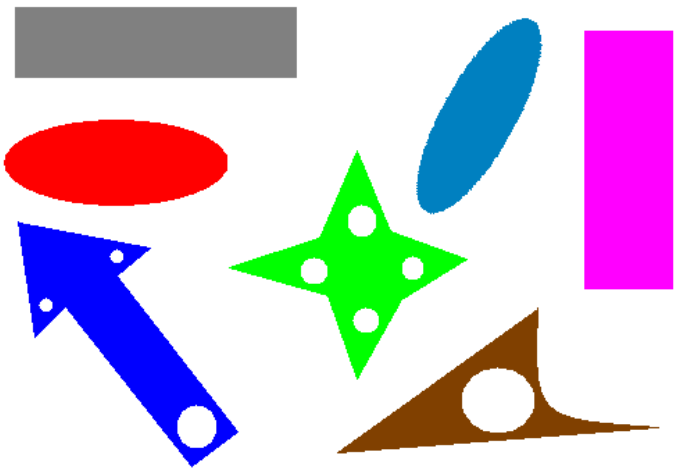

Fig. 4.1 Exemplu de imagine etichetată pe care pot fi testați algoritmii descriși

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 acțiune 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 funcției *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 trăsăturilor.

```
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* 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. **Activități 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 subțiere și elongația (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. **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!!!**

4.5. Referințe

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