

Segmentarea imaginilor (1)

Scopul segmentarii imaginilor este de a separa obiectele dintr-o imagine (scena) de fundal. In general, algoritmii utilizati in segmentare se bazeaza pe doua caracteristici ale intensitatii pixelilor: *discontinuitate si similaritate*.

In prima categorie, algoritmii realizeaza partitionarea unei imagini folosind variatiile bruste ale intensitatii (ex. segmentarea prin detectarea muchiilor).

Algoritmii din cea de-a doua categorie partitioneaza imaginea in regiuni care sunt asemanatoare, pe baza unor criterii prestabilite. Dintre metodele din a doua categorie putem aminti:

- segmentarea cu prag
- dezvoltarea regiunilor (segmentarea prin extindere)
- segmentarea prin divizare si unificare (split and merge)

1. Segmentarea prin delimitarea cu prag

Delimitarea cu prag este utila cand caracteristicile de intensitate sunt suficiente pentru a individualiza un obiect dintr-o imagine. Valorile caracteristicilor de intensitate se aleg astfel incat un interval de intensitati sa fie caracteristic pentru un singur obiect din imagine.

Metodele de segmentare cu prag se folosesc in cazul imaginilor cu niveluri de gri, imagini binare (documente tiparite, desene, grafice etc), in cazul imaginilor color si a celor multispectrale.

Metodele de prag pot fi :

- **globale** (un singur prag pentru toata imaginea)
- **locale** (pragul variaza in cadrul unei imagini)

In cazul unor imagini cu niveluri de gri, alegerea unui prag inseamna impunerea unui nivel de gri care separa obiectele de fundal. In cazul unui singur prag T:

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & f(x, y) \leq T \end{cases}$$

1 – semnifica obiecte, 0 - fundal

Se pot impune:

- *praguri de banda*

$$g(x, y) = \begin{cases} 1, pt. f(x, y) \in D \\ 0 \quad in rest \end{cases}$$

- *praguri multiple*

Prelucrarea Imaginilor - Laborator 10

$$g(x, y) = \begin{cases} r_1, & \text{pt. } f(x, y) \in D_1 \\ r_2, & \text{pt. } f(x, y) \in D_2 \\ \vdots \\ r_n, & \text{pt. } f(x, y) \in D_n \\ 0, & \text{in rest} \end{cases}$$

- *semipraguri* - elimină pixelii din imagine cu un nivel de gri situat sub un anumit prag, punând 0 (negru) în locul acestora. Pixelii ce depășesc nivelul stabilit rămân neschimbați (“mascare” a fundalului):

$$g(x, y) = \begin{cases} f(x, y), & \text{pt. } f(x, y) \geq T \\ 0, & \text{pt. } f(x, y) < T \end{cases}$$

Alegerea pragului se poate face astfel:

- se examinează histograma imaginii pentru localizarea varfurilor și adânciturilor acesteia
- se face delimitarea adaptiva, prin examinarea histogramelor vecinătăților locale

Să considerăm o histogramă a unei imagini în nuante de gri, imagine ce conține obiecte luminoase pe un fond intunecat (fig. 1).

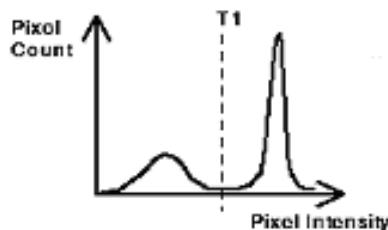


Fig. 1

Pentru a extrage obiectele de fundal se selectează pragul de valoare T . Se consideră ca obiectele și fundalul au, fiecare, mai multe nuante de gri.

In figura 2 se arată cazul a două tipuri de obiecte luminoase pe un fond intunecat:

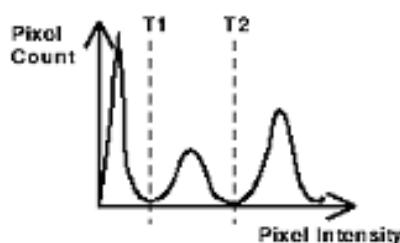


Fig. 2

In acest caz avem două praguri. Pentru $f(x, y) \leq T_1$ avem fundalul, pentru $T_1 < f(x, y) \leq T_2$ un obiect, iar pentru $T_2 < f(x, y)$ al doilea obiect.

Prelucrarea Imaginilor - Laborator 10

Funcții OpenCV utile în segmentare:

- binarizarea imaginii – **threshold**, cu parametrul **cv::THRESH_BINARY**, și o valoare prag aleasa de utilizator
- calcularea pragului folosing metoda Otsu – **threshold**, cu parametrul **cv::THRESH_BINARY + cv::THRESH_OTSU**, si valoarea prag 0
-
- etichetarea componentelor conectate intr-o imagine alb-negru – **connectedComponents**

Observatie:

Operatiile morfologice ce se pot aplica imaginilor binare: umplere, eroziune, dilatare, deschidere (eroziune + dilatare), închidere (dilatare + eroziune):

- *umplere* (eng. *fill*) – operatia de colorare a unor zone de imagine ce contin pixeli cu acelasi nivel de gri;
- *eroziune* – operatie morfologica ce realizeaza scaderea ariei unei zone de 1 dintr –o imagine binara urmarind frontiera acesteia, pe interior, cu un element morfologic dat;
- *dilatare* – operatia inversa eroziunii, realizata prin urmarirea frontierei cu elementul model pe exteriorul suprafetei;
- *deschidere* – operatie realizata prin eroziune si dilatare succesiva cu acelasi element – are ca efect eliminarea zonelor albe dintr-o imagine binara, de dimensiuni de pâna la dublul dimensiunilor elementului morfologic folosit.
- *închidere* – operatie realizata prin dilatare si eroziune succesiva cu acelasi element – are un efect analog *deschiderii*, dar elmina din zonele albe „insulele” de negru de dimensiuni mici.

Aplicatii

Implementați următoarele operații:

1. Determinati pragul de segmentare prin urmatoarele metode:

- a. Calculati pragul de segmentare folosind metoda Otsu (functia **threshold**);
- b. Calculati pragul de segmentare folosind urmatoarea metoda iterativa:

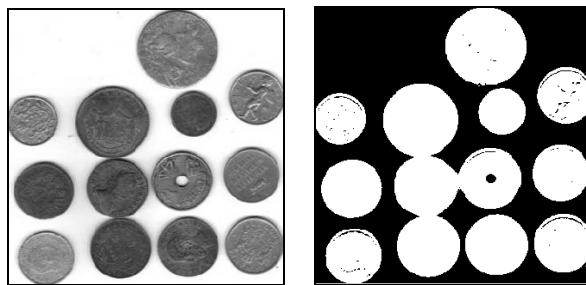
Prelucrarea Imagineilor - Laborator 10

Algorithm 1 Iterative threshold selection

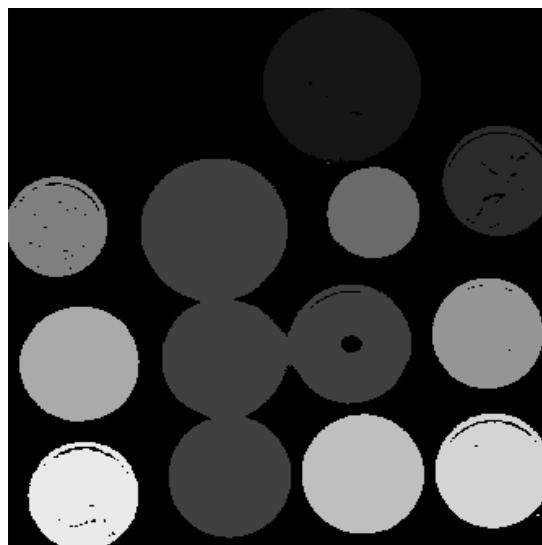
- 1: Select an initial estimate for T
 - 2: Partition the image using T into two regions: a group G_1 of pixels with gray level values $\leq T$ (object) and a group of pixels, G_2 , with gray level values $> T$ (background)
 - 3: Compute mean intensity values, μ_1 and μ_2 , for the two groups
 - 4: Compute the new threshold $T = \frac{\mu_1 + \mu_2}{2}$
 - 5: Repeat steps 2 to 4 until there is no change in T (or the difference in successive iterations is smaller than a predefined threshold)
-

Pentru valoarea initială a pragului se poate utiliza valoarea medie a intensității din imagine (atunci cand obiectul și fondul ocupă regiuni de dimensiuni similare) sau valoarea mediana între intensitățile minima și maxima din imagine (atunci cand obiectele de interes sunt mici în raport cu aria acupată de background – în acest caz nu se poate utiliza valoarea medie deoarece un grup de pixeli va domina histograma imaginii).

2. Afisati imaginea initiala si rezultatul segmentarii. Ex:

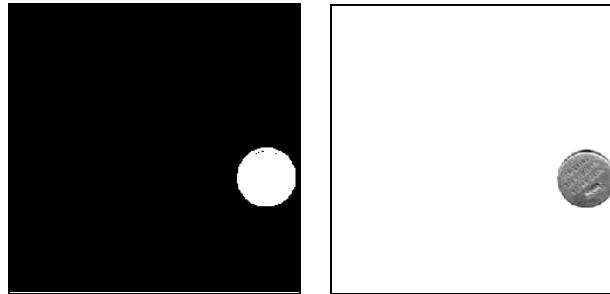


3. Afisati o imagine ce conține componentele etichetate din imaginea binarizată (rezultatul funcției **connectedComponents**, adus în intervalul 0-255) :



Prelucrarea Imaginilor - Laborator 10

4. Identificati obiectele individuale din imagine folosind rezultatul anterior, apoi afisati, pentru cateva din aceste obiecte, atat variantele lor segmentate, cat si cele din imaginea initiala. Ex:

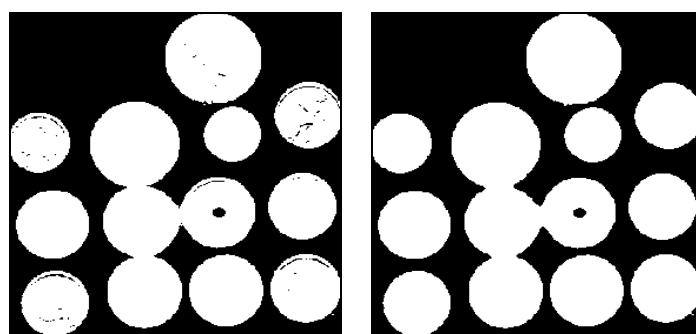


4. Utilizati operatiile morfologice de dilatare, eroziune, deschidere si/sau inchidere pentru a incerca sa corectati erorile care apar la segmentarea cu prag (regiunile negre din interiorul monedelor). Pentru operatiile mentionate, incercati elemente structurale de diverse forme (dreptunghi, cerc etc).

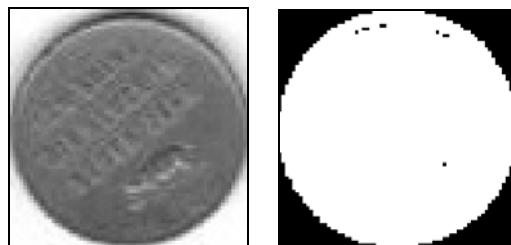
Un element structural se genereaza folosind functia **getStructuringElement**, pentru care parametrii importanti sunt tipul elementului (**MORPH_ELLIPSE**, **MORPH_RECTANGLE** etc.) si dimensiunea sa (Ex. **Size(10, 10)**)

Operațiile morfologice se aplică folosind functia **morphologyEx**, cu parametrul corespunzător operatiei dorite (**MORPH_DILATE**, **MORPH_ERODE**, **MORPH_OPEN** sau **MORPH_CLOSE**)

De exemplu, operatia de deschidere genereaza urmatorul rezultat:



5. Izolati obiectele din imagine si afisati-le (extrageti din imagine cel mai mic dreptunghi care incadreaza obiectele individuale). Ex:



Prelucrarea Imagineilor - Laborator 10

6. Pentru obiectele astfel extrase, realizati segmentarea utilizand un prag calculat individual (local) pentru fiecare obiect separat. Verificati daca exista diferente fata de segmentarea realizata la punctele anterioare.