

Vježba 8.

Segmentacija

Za segmentaciju u MATLAB-u može se koristiti funkcija iz IPT `im2bw` čija je osnovna sintaksa

```
BW = im2bw(I, level)
```

gdje je `I` ulazna intenzitetska slika, a `level` globalni prag. U rezultujućoj, binarnoj, slici `BW` vrijednost 1 će biti na koordinatama svih piksela slike `I` čiji je intenzitet veći od praga, a vrijednost 0 na ostalim. Ovu funkcionalnost je moguće postići i korištenjem vektorskih logičkih operacija:

```
BW = I > level
```

U IPT je ugrađen jedan algoritam za određivanje globalnog praga za segmentaciju slike. Radi se o Otsuovom algoritmu u kojem se prag za segmentaciju određuje maksimiziranjem vrijednosti varijanse između klasa. Polazna tačka za ovaj algoritam je normalizovani histogram

$$p_r(r_q) = \frac{n_q}{n}, \quad q = 0, 1, \dots, L-1,$$

gdje je n ukupan broj piksela slike, n_q je broj piksela čiji je intenzitet r_q , a L je ukupan broj mogućih nivoa intenziteta na slici. Pretpostavimo da je prag k izabran tako da je C_0 skup piksela sa nivoima intenziteta $[0, 1, \dots, k-1]$, a C_1 skup piksela sa nivoima intenziteta $[k, k+1, \dots, L-1]$.

Otsuovim metodom vrijednost praga k se bira tako da se maksimizira varijansa između klasa σ_B^2 , definisana sa

$$\sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2$$

gdje su

$$\begin{aligned}\omega_0 &= \sum_{q=0}^{k-1} p_q(r_q) \\ \omega_1 &= \sum_{q=k}^{L-1} p_q(r_q) \\ \mu_0 &= \sum_{q=0}^{k-1} q p_q(r_q) / \omega_0 \\ \mu_1 &= \sum_{q=k}^{L-1} q p_q(r_q) / \omega_1 \\ \mu_T &= \sum_{q=0}^{L-1} q p_q(r_q)\end{aligned}$$

Funkcija kojom je implementiran ovaj metod je `graythresh`. Njena sintaksa je

```
T = graythresh(a)
```

gdje je `a` ulazna slika, a `T` rezultujući prag. Vrijednost vraćenog praga je normalizovana na interval između 0.0 i 1.0. Kako bi se ova vrijednost iskoristila za segmentaciju neophodno ju je skalirati na odgovarajući opseg.

Segmentacija slika u boji

Slike u boji je moguće segmentirati klasterizacijom kolor-vektora piksela. Klasterizacija predstavlja grupisanje kolor-vektora pri čemu će se vektori „slične“ boje nalaziti u istom klasteru, a vektori različitih boja u različitim klasterima. Jedan od najčešće korištenih algoritama za klasterizaciju je *k-means* algoritam koji klastere određuje tako da se minimizira ukupna suma udaljenosti vektora od centroida klastera kojima su ti vektori dodijeljeni (distorzija).

S obzirom na uticaj načina izračunavanja udaljenosti kolor-vektora na rezultat segmentacije posebnu pažnju treba posvetiti izboru funkcije kojom se računa udaljenost kolor-vektora. Najčešće se koristi Euklidova udaljenost između kolor-vektora u RGB ili CIE $L^*a^*b^*$ prostoru koji se smatra perceptualno uniformnim.

U MATLAB-ovom Statistical Toolboxu se nalazi implementacija *k-means* algoritma čija je osnovna sintaksa:

```
idx = kmeans(x, k),
```

gdje je `x` matrica dimenzija $N \times P$, tj. sadrži N P -dimenzionalnih vektora koje treba rasporediti u k klastera.

Problem sa *k-means* algoritmom je mogućnost upadanja u lokalne minimume. Jedan od načina da se ovo izbjegne je da se algoritam pokrene više puta sa različitim početnim uslovima i da se kao konačni rezultat usvoji klasterizacija koja rezultuje minimalnom ukupnom distorzijom. Ovo je u pomenutoj imlementaciji *k-means* algoritma moguće postići opcijom `Replicate` (vidjeti dokumentaciju za funkciju `kmeans` u MATLAB-u).

Drugi problem je što je potrebno unaprijed poznavati broj klastera, tj. regiona na koje želimo da segmentiramo sliku. Iako postoje metodi za procjenu ovog broja još uvijek se on vrlo često određuje na osnovu znanja o konkretnom domenu primjene.

Slika u boji je predstavljena trodimenzionalnim nizom dimenzija $M \times N \times 3$ i za primjenu *k-means* algoritma za klasterizaciju boja potrebno je ovaj niz transformisati u matricu u kojoj se u svakom redu nalazi boja jednog piksela. Ova matrica je dimenzija $MN \times 3$. Ova transformacija se može postići korištenjem funkcije `reshape(X, M, N, P, ...)`, koja vraća višedimenzionalni niz koji ima iste elemente kao `X` ali preuređene tako da bude dimenzija $M \times N \times P \dots$, pri čemu proizvod dimenzija rezultujućeg niza mora biti jednak proizvodu dimenzija originalnog niza `X`.

Za ovu vježbu su još od značaja i funkcije:

```
c = makecform('srgb2lab');
```

kojom se formira struktura za konverziju iz sRGB kolor-prostora u CIE $L^*a^*b^*$ kolor-prostor;

```
b = applycform(a, c);
```

kojom se primjenjuje kolor-transformacija u promjenljivoj `c` na sliku `a`.

Vizuelizacija segmentiranih slika može se realizovati pomoću funkcije `label2rgb`, čija je sintaksa:

```
rgb = label2rgb(L);
```

Ova naredba konvertuje matricu oznaka regiona `L` u RGB matricu `rgb` u kojoj su regionima dodijeljene različite boje kako bismo mapu regiona mogli lako da prikažemo.

Zadaci

1. Učitati sliku `coins.png` (sastavni dio IPT). Nacrtati njen histogram i na osnovu histograma i slike odrediti prag za segmentaciju slike na novičiće i pozadinu. U rezultujućoj binarnoj slici novičići treba da budu objekti. Komentarisati izbor praga i rezultat segmentacije.
2. Napisati MATLAB program za automatsko određivanje praga ISODATA algoritmom. Pomoću ovog programa odrediti prag za segmentaciju slike `coins.png`. Segmentirati sliku na objekte i pozadinu. Da li je dobijen zadovoljavajući rezultat?
3. Odrediti globalni prag za segmentaciju slike `coins.png` Otsuovim algoritmom i segmentirati sliku na objekte i pozadinu. Uporediti dobijenu vrijednost praga sa vrijednošću dobijenom u prethodnoj tački.
4. Učitati sliku `rice.png` (sastavni dio IPT). Segmentirati sliku na zrna riže i pozadinu. Isprobati različite algoritme za određivanje praga i različite vrijednosti za prag. Komentarisati izbor praga i rezultat.
5. Ukloniti sjenčenje sa slike `rice.png` niskopropusnim filtriranjem i ponoviti određivanje praga i segmentaciju. Komentarisati izbor praga i rezultat. Uporediti rezultate dobijene primjenom automatskih metoda za određivanje praga dobijene u prethodnoj i ovoj tački.
6. Učitati sliku `lena.jpg`. Prikazati histogram ove slike i komentarisati zašto je teška za segmentaciju. Odrediti prag za segmentaciju nekim od algoritama za automatsko određivanje praga. Segmentirati sliku i komentarisati rezultat.
7. Napisati funkciju u MATLAB-u za segmentaciju kolor slika korištenjem klasterizacije. Zaglavlje funkcije treba da ima sledeći izgled:

```
function lab = kmeans_segmentacija(im, no_reg, no_iter)
% kmeans segmentacija
% lab = kmeans_segmentacija(im, k)
%
% im          - ulazna slika u boji
% no_reg      - broj regiona
% no_iter     - broj ponavljanja kmeans algoritma
% lab        - oznake regiona
```

8. Pomoću funkcije iz prethodne tačke segmentirati sliku `epithel.jpg` korištenjem klasterizacije u RGB kolor-prostoru.

9. Ponoviti prethodni zadatak korištenjem $L^*a^*b^*$ kolor-prostora. Uporediti rezultate.
10. Uporediti rezultate kolor-segmentacije sa segmentacijom grayscale verzije iste slike. Prag za segmentaciju odrediti pomoću funkcije `graythresh`.