

Vježba 6.

Uvod u digitalnu obradu slike

Uvod

Cilj ove vježbe je upoznavanje sa osnovnim operacijama digitalne obrade slike: *operacijama u tački* kod kojih se svjetlina izlaznog piksela izračunava samo na osnovu svjetline ulaznog piksela i *lokalnim operacijama* kod kojih se svjetlina izlaznog piksela izračunava na osnovu svjetlina piksela iz njegove okoline. Pored toga, razmotrićemo i formiranje palete boja korištenjem k-means algoritma.

Operacije u tački

Histogram je funkcija koja za svaku boju, odnosno, vrijednost intenziteta daje broj piksela te boje, odnosno, vrijednosti intenziteta u nekom regionu. Histogram sive (grayscale) slike je u MATLAB-u moguće izračunati i nacrtati korištenjem funkcije Image Processing Toolboxa (IPT) `imhist`. Sintaksa ove funkcije je

```
h = imhist(a, n);
```

gdje je a ulazna slika, a n broj ćelija (intervala na koje je podjeljena skala intenziteta). Izlazni vektor h će imati n elemenata i u svakom elementu će biti broj piksela čiji intenzitet se nalazi u određenom intervalu. Ukoliko se izlazna promjenljiva ne zada, funkcija crta histogram slike.

Kada se želi postići poboljšanje kontrasta na slici često se koriste operacije zasnovane na transformacijama histograma. Jedna od često korištenih operacija je *razvlačenje histograma*.

Matematički se transformacija intenziteta može iskazati na sljedeći način. Ako je potrebno nivo svjetline piksela polazne slike $a(m, n)$ koje se nalaze između vrijednosti p_{low} i p_{high} preslikati u opseg svjetlina od s_{low} do s_{high} koristi se sledeća transformacija:

$$b(m, n) = \begin{cases} s_{low}, & a(m, n) < p_{low} \\ \left(s_{high} - s_{low} \right) \left(\frac{a(m, n) - p_{low}}{p_{high} - p_{low}} \right)^\gamma + s_{low}, & p_{low} \leq a(m, n) \leq p_{high} \\ s_{high}, & a(m, n) > p_{high} \end{cases}$$

Ako je *razvlačenje histograma* linearno γ je jednako 1. Druge vrijednosti γ daju nelinearno *razvlačenje histograma*.

Osnovna funkcija IPT koja se koristi za transformaciju intenziteta piksela je `imadjust`. Njena sintaksa je:

```
b = imadjust(a, [low_in high_in], [low_out, high_out], gamma);
```

Ova funkcija preslikava vrijednosti intenziteta na slici *a* u nove vrijednosti na slici *b*, tako da se vrijednosti između *low_in* i *high_in* preslikavaju u vrijednosti između *low_out* i *high_out*. Sve vrijednosti manje od *low_in* se preslikavaju u *low_out*, a vrijednosti veće od *high_in* se preslikavaju u *high_out*. Iako funkcija *imadjust* prihvata i ulazne podatke klase *uint8* i *uint16* ove vrijednosti moraju biti iz opsega [0, 1]. Ukoliko se ove vrijednosti izostave, tj. zada se [] koriste se predefinisane vrijednosti [0 1]. Parametar *gamma* određuje oblik krive koja određuje preslikavanje vrijednosti intenziteta. Ako je *gamma* manje od 1 veća težina se daje većim izlaznim vrijednostima intenziteta (svjetlije), a ako je *gamma* veće od 1 veća težina se daje manjim izlaznim vrijednostima intenziteta (tamnije).

Lokalne operacije

Transformacije histograma pripadaju grupi operacija u tački. Pored ovih operacija značajne su i lokalne operacije kod kojih se vrijednost intenziteta piksela izračunava korištenjem vrijednosti intenziteta piksela iz neke njegove okoline. Uobičajen termin za korištenje ovih operacija je *filtriranje slike*.

Za filtriranje slike u IPT koristi se funkcija *imfilter* čija je osnovna sintaksa:

```
b = imfilter(a,h, način_filtriranja, rubne_opcije, opcije_veličine);
```

gdje je *a* slika koja se filtrira, *h* maska filtra, *način_filtriranja* specificira da li se izračunava korelacija ili konvolucija, *rubne_opcije* način rješavanja problema rubnih piksela, a *opcije_veličine* određuje da li će biti vraćen cijeli rezultat konvolucije ili samo slika čija je veličina jednaka veličini polazne slike. Detalji specificiranja ovih parametara su dati u dokumentaciji.

U IPT postoji funkcija *fspecial* pomoću koje je moguće generisati predefinisane 2D filtre koji se često koriste u digitalnoj obradi slike. Osnovna sintaksa ove funkcije je:

```
h = fspecial('tip_filtra');
```

gdje je *tip_filtra* string koji sadrži naziv željenog filtra. U zavisnosti od tipa filtra ova funkcija može imati i dodatne argumente kojima se filter detaljnije definiše. Maske filtera koje se dobijaju korištenjem ove funkcije su već rotirane za 180 stepeni i mogu se direktno iskoristiti za izračunavanje konvolucije pomoću funkcija *imfilter* i *filter2*.

Na primjer, naredbom:

```
>> h = fspecial('average', 3)
```

```
h =
```

```
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
```

generisan je uniformni filter dimenzija 3x3, a naredbom:

```
>> w = fspecial('laplacian', 0)
```

```
w =
```

```
    0    1    0
```

```
1   -4   1
0    1   0
```

Laplasov filtar.

K-means klasterizacija

Paletu boja koja se koristi za indeksiranu reprezentaciju slike u boji je moguće formirati korištenjem k-means algoritma. U MATLAB-ovom Statistical Toolboxu se nalazi implementacija k-means algoritma čija je osnovna sintaksa:

```
[idx, c] = kmeans(x, k),
```

gdje je x matrica dimenzija $N \times P$, tj. sadrži N P -dimenzionalnih vektora koje treba rasporediti u k klastera. Vektor idx je dimenzija $N \times 1$ i sadrži oznake klastera kojim pripada svaki od vektora iz x . U redovima matrice c , dimenzija $K \times P$, se nalaze centri klastera. Kada se k-means koristi za formiranje palete boja centri klastera će odgovarati bojama u paleti.

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 implementaciji k-means algoritma moguće postići opcijom `Replicates` (vidjeti dokumentaciju za funkciju `kmeans` u MATLAB-u).

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 .

Napomena: U ovoj vježbi je pogodno raditi sa vrijednostima piksela u opsegu $[0, 1]$. Za konverziju možete iskoristiti funkciju `im2double`.

Zadaci

1. Proučiti dokumentaciju funkcije `imhist`. Koliko nivoa intenziteta Matlab koristi za određivanje histograma ukoliko se ne specificira taj podatak?
2. Učitati sliku `pout.tif` (sastavni dio IPT). Prikazati sliku i njen histogram. Kakva je pokrivenost dinamičkog opsega?
3. Formirati novu sliku tako što ćete opseg intenziteta $[0.3, 0.65]$ linearno preslikati u opseg $[0, 1]$. Prikazati dobijenu sliku i njen histogram. Šta se desilo sa kontrastom na slici? Kako se ta promjena ogleda na histogramu?
4. Učitati sliku `cameraman.tif` (sastavni dio IPT). Prikazati sliku i njen histogram.
5. Na osnovu slike i njenog histograma odrediti kojem opsegu intenziteta pripadaju pikseli kaputa? Preslikati linearno taj opseg u opseg $[0.5, 1]$. Prikazati dobijenu sliku. Šta se desilo

- sa slikom u cjelini? Šta se desilo sa opsegom intenziteta piksela kaputa? Šta je na taj način postignuto? Šta se desilo sa pozadinom? Zašto?
6. Formirati novu sliku koja predstavlja negativ slike iz tačke 4.
 7. Preslikati nelinearno cijeli opseg intenziteta sa vrijednostima parametra `gamma` 0.5 i 1.5. Prikazati dobijene slike. Šta se dešava sa slikom u ovim slučajevima? Objasniti nastale promjene?
 8. Učitati sliku `cameraman.tif` i prikazati je. Pomoću funkcije `fspecial` generisati uniformne filtre dimenzija 3x3, 5x5 i 7x7. Kako izgledaju ovi filtri? Filtrirati sliku korištenjem svakog od ovih filtara. Iskoristiti predefinisane opcije funkcije `imfilter`. Prikazati dobijene slike. Šta se dešava sa slikom kada se red filtra povećava? Kako se ta pojava može objasniti?
 9. Učitati i prikazati sliku `blurry_moon.tif`. Može se primijetiti da je ova slika blago zamučena. Primijeniti na ovu sliku Laplasov filter. Obratite pažnju da pošto Laplasov filter sadrži i negativan koeficijent može se očekivati da će rezultujuća slika sadržati piksele sa negativnom vrijednošću. Međutim, ako je slika klase `uint8` negativne vrijednosti će biti odsječene. Zbog toga je prije upotrebe Laplasovog filtra pogodno konvertovati sliku u klasu `double` korištenjem funkcije `im2double`. Pronaći sada razliku originalne slike i slike dobijene nakon filtriranja Laplasovim filtrom. Prikazati dobijenu sliku. Kakav zaključak možete izvesti? Filter implementiran na ovaj način zove se *unsharp mask*.
 10. Učitati sliku [mandrill_small.png](#) i prikazati je. K-means algoritmom klasterizovati piksele ove slike u 256 klastera. Da biste izbjegli eventualne probleme sa praznim klasterima postavite opciju 'EmptyAction' na vrijednost 'singleton'. Korištenjem priložene funkcije `showPalette(c)` prikazati dobijenu paletu.
 11. Da biste vidjeli kako izgleda slika nakon prelaska na indeksiranu reprezentaciju potrebno je rekonstruisati sliku korištenjem samo centroida kojima su pridruženi pikseli. Napisati program u MATLAB-u koji za datu sliku i paletu formira rekonstruisanu sliku tako što svaki piksel mijenja najbližim centroidom. Pošto se paleta formira posebno za svaku sliku ovdje se može iskoristiti niz `idx` koji vraća funkcija `kmeans`. Prikazati rekonstruisanu sliku.
 12. Ponoviti prethodne dvije tačke za palete sa različitim brojem boja, kao i za različite slike. Kako biste smanjili vrijeme izvršavanja `kmeans` algoritma preporučljivo je smanjiti rezoluciju slike korištenjem funkcije `imresize`.