

Izveštaj prvog domaćeg zadatka

iz predmeta

Digitalna obrada slike

Katarina Petrović 2021/0068

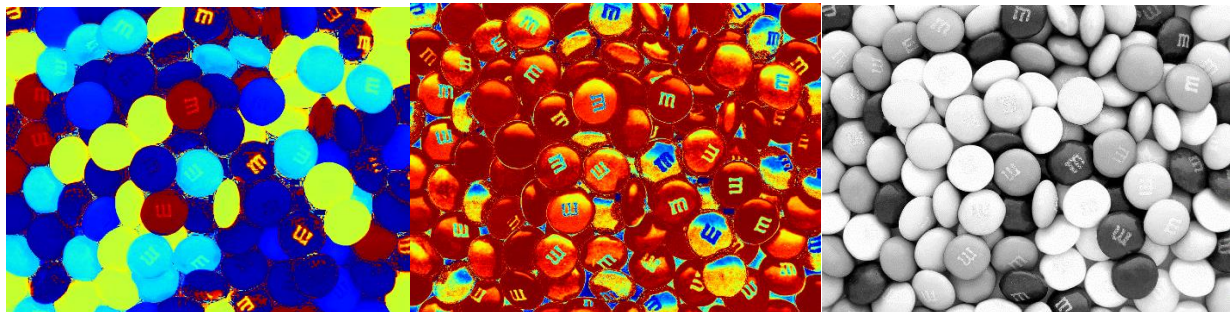
1. Zadatak:

S obzirom da je u RGB formatu (u kom su slike generalno zadate) nemoguće na odgovarajući način detektovati plave M&M bombone jer su beli i svetliji delovi na slici problematični zbog visokih vrednosti svih komponenti boje, neophodno je sliku konvertovati u drugi kolor sistem koji razlikuje informaciju o osvetljaju i o boji. Konkretno, u ovom slučaju odabran je HSV kolor sistem u kome su od značaja:

- Hue komponenta - predstavlja informaciju o boji (opseg od 0.5 do 0.7 se ispostavlja da dobro obuhvata plave vrednosti na slici)
- Saturation komponenta – predstavlja informaciju o osvetljaju (niske vrednosti Saturation predstavljaju bele piksele)

Postupak obrade slike podrazumeva prvobitno prebacivanje slike iz RGB formata u HSV format, a zatim selektovanje piksela koji imaju visoku vrednosti Saturation (ne želimo da selektujemo bele piksele), i krajnje – od selektovanih piksela biramo one kojima je Hue u opsegu od 0.5 do 0.75. Svim pikselima koji su zadovoljili ove uslove postavljamo Hue komponentu na 0.85 što odgovara ljubičastoj boji. Konačno, sliku vraćamo u RGB format, prikazujemo je i čuvamo.

Sledeće slike prikazuju ovaj postupak:



Hue, Saturation i Value komponente ulazne slike (respektivno)

```
#glavna obrada slike - za sve piksele posmatramo saturaciju i hue
for i in range(0,M):
    for j in range(0,N):
        if imghsv[i,j,1] > 0.3: #ako je saturacija veca od praga 0.3 (izuzimamo bele delove slike)
            if imghsv[i,j,0] > 0.5 and imghsv[i,j,0] < 0.75: #ako je hue u opsegu [0.5, 0.75] - plava boja na slici
                imghsv[i,j,0] = 0.85 #postavimo hue na 0.85 - ljubicasta boja
```

Glavna obrada slike



Prikaz slike pre (levo) i posle (desno) obrade

2. Zadatak:

Na osnovu pravilnosti koje se javljaju u zadatim slikama, moguće je odrediti postupak pomoću kog se kvalitet svih slika može popraviti.

- 1) Naime, funkcija `restore(img)` prvobitno treba da ukloni šum sa slike koji je u ovom slučaju impulsni koristeći median filter (traži medijanu neke okoline piksela i nju dodeljuje centralnom pikselu)
- 2) Zatim, funkcija treba da invertuje vrednosti u slici, tako da svi pikseli sa vrednošću `val` dobiju vrednost `255-val`
- 3) S obzirom da je izlazna slika nakon ovog postupka lošeg kontrasta, neophodno je razvući histogram slike i time poboljšati kontrast (funkcija `rescale_intensity` kojoj su automatski izračunati početak i kraj histograma)

Funkcija vraća sliku koja je prošla kroz opisani prvi, drugi i treći korak obrade. U nastavku su prikazani međukoraci u obradi slike `corrupted1.png`:



Ulazna slika – corrupted1.png



Slika nakon primene median filtra



Slika nakon invertovanja vrednosti



Slika nakon popravke kontrasta, izlazna slika original1.png

Takođe, u nastavku su prikazane slike nakon obrade fajlova corrupted2.png i corrupted3.png



Slika corrupted2.png pre i nakon obrade funkcijom restore



Slika corrupted3.png pre i nakon obrade funkcijom restore

Vidimo i u ova dva slučaja da je izlazna slika dobro obrađena.

3. Zadatak

Funkcija `bilateral_filter` realizovana je na sledeći način:

- 1) Prvobitno se unutar funkcije proverava neispravan unos vrednosti (negativne vrednosti radijusa, σ_s i σ_r , slika u pogrešnom opsegu).
- 2) Zatim se formira maska Gausovog filtra koja je invarijantna duž slike sa vrednostima definisanim parametrima radius i σ_s – adekvatno Gaussian ugrađenom filtru
- 3) Nakon toga, slika se proširuje za (radius) piksela sa svake strane ugrađenom funkcijom `np.pad()`
- 4) Konačno, u glavnoj obradi slike iteriramo po celoj slici i formiramo matricu maske tako što napravimo masku koja se odnosi na Gausovu raspodelu koja uzima u obzir osvetljaje piksela unutar maske i zatim je pomnožimo sa već napravljenom maskom Gausovog filtra za usrednjavanje (zamućivanje) slike. Takođe, u ovom koraku skaliramo masku tako da je suma njenih koeficijenata 1 (filtri za usrednjavanje) i primenjujemo masku na sliku (korelacija slike i maske).

Postupak bilateralnog filtra testiran je sa različitim parametrima i u nastavku slede rezultati:



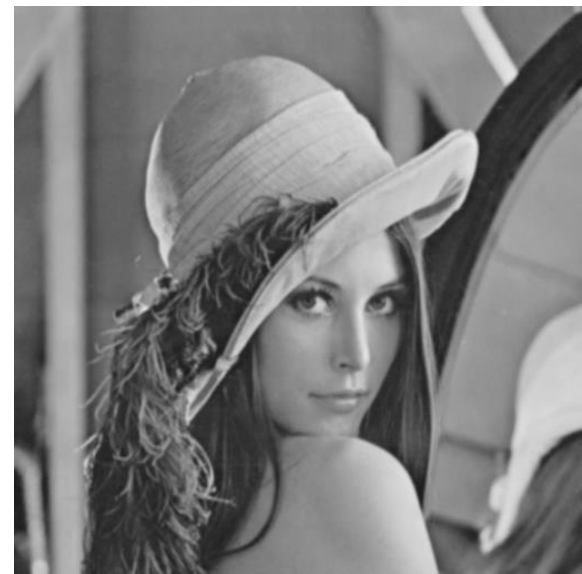
Slika pre i nakon obrade funkcijom `bilateral_filter(img, radius=3, sigma_s=1, sigma_r=0.4)`.
Ovaj filter deluje da dobro obavlja i funkciju zamućivanja homogenih površina i čuvanje značajnih ivica.



Slika pre i nakon obrade funkcijom `bilateral_filter(img, radius=3, sigma_s=3, sigma_r=0.4)`.
Ovaj filter ima prenaplašeno zamućenje (preveliko `sigma_s`) i samim tim iako je šum u velikoj meri potisnut, slika je izgubila informacije o detaljima.



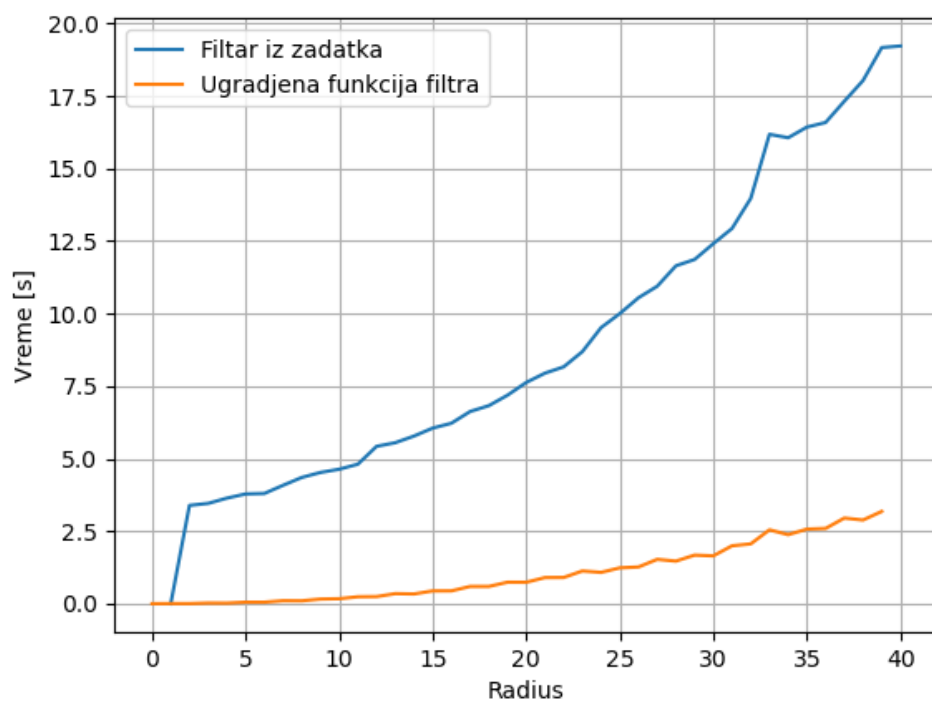
Slika pre i nakon obrade funkcijom `bilateral_filter(img, radius=3, sigma_s=3, sigma_r=0.1)`. Ovaj filter iako ima prenaplašeno zamućenje po homogenim površinama, ima značajno bolje informacije o ivicama (na ovo utiče malo σ_r koje velikim odstupanjima od vrednosti piksela daje minimalno na značaju). Osim očuvanja oštarih ivica, ovaj filter ima neke neprirodne delove (na ovoj slici konkretno nije dobro održana ivica brade, oka, delovi perja na šeširu)



Slika pre i nakon obrade funkcijom `bilateral_filter(img, radius=3, sigma_s=1, sigma_r=5)`. Ovaj filter ima preveliko σ_r i samim tip uopšte „ne vidi“ ivice, odnosno slika sa proastom σ_r sve više liči na klasičan Gaussian filter.

Sa povećanjem radijusa, primećeno je povećanje zamućenja (obuhvatamo veći broj piksela čije vrednosti usrednjavamo).

Naredni korak predstavlja praćenje povećanja vremena izvršavanja filtra sa porastom radijusa. Takođe, ovom prilikom rad funkcije je upoređen sa ugrađenom `skimage.restoration.bilateral_filter` funkcijom. Rezultat je predstavljen na sledećem grafiku:



Sa grafika je uočljivo da je ugrađeni bilateralni filter dosta brži u odnosu na ručno iskucani filter.