

Univerzitet u Sarajevu
Elektrotehnički fakultet

Odsjek za računarstvo i informatiku
Digitalno procesiranje signala

Seminarski rad

Tema: Color image processing

Radili:
Dardagan Nađa (17953)
Brđanin Adnan (17796)

Profesorica:
Vanr.Prof.Dr.Amila Akagić

Sarajevo, maj 2018. god.

Sadržaj

1. Osnovni pojmovi.....	3
1.1 Hromatitet.....	4
1.2 Percepcijski uniformni prostori boja.....	5
2. Percepcija boje ljudskom oku.....	6
3. Modeli predstavljanja boja.....	7
3.1 RGB.....	7
3.2 CMY/CMYK.....	8
3.3 HSV.....	8
3.4 YIQ(NTSC).....	9
3.5 YcbCr (digitalni video).....	10
4. Predstavljanje slika u MATLAB-u.....	11
4.1 RGB slike.....	11
4.2. Indeksirane slike.....	12
5. Pseudocolor obrada slika.....	13
5.1 Intensity Slicing.....	13
5.2 Transformacija nijansi sive u boje.....	14
5.3 Pseudocoloring u frekventnom domenu.....	14
6. Full color obrada slika.....	15
6.1 Transformacija boja.....	16
6.2 Histogram Processing.....	17
6.3 Poboljšanje i izoštravanje slike u boji.....	18
6.4 Redukcija šumova boje.....	19
6.5 Segmentacija slike bazirana na boji.....	19
7. Literatura.....	28

1. Osnovni pojmovi

Percepcija boje je psihofizički fenomen koji se ogleda u dvije stvari:

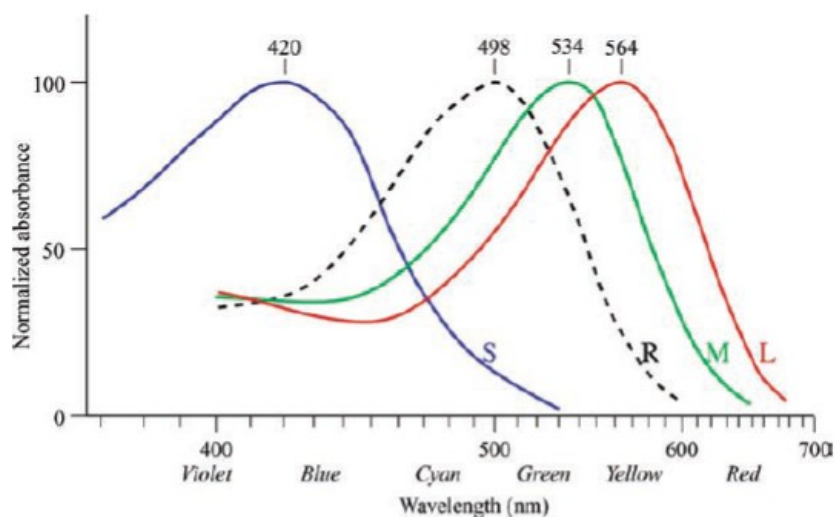
- fizička svojstva svjetlosnih izvora (obično se radi o površinama i spektralnoj distribuciji snage (SPD))
- fiziološki i psihološki aspekti ljudskog vizuelnog sistema (čula vida)

Ako govorimo o percepciji boje, moramo spomenuti i hromatski svjetlosni izvor, a to je izvor koji emitira elektromagnetnu radijaciju talasnih dužina između 400 i 700 nm. Odbijanje tog zračenja od površinu uzrokuje pojavu boje u našem čulu vida. Boje se na ovaj način razlikuju u količini odbijanja elektromagnetnog zračenja, tako da ako govorimo o bijeloj boji, govorimo o površini koja odbija sve talasne dužine u ranije navedenom spektru, dok je crna boja ustvari površina koja apsorbira sve (većinu) talasne dužine, bez obzira o istoj.

Također, mogu se definisati i ostale boje na isti način, npr. žuta se definiše kao površina koja apsorbira većinu energije u spektru od 565-590 nm. Izvor koji emituje ovakvo zračenje možemo definisati sa tri parametra:

- Intenzitet (Radijansa engl. Radiance) – ukupna količina energije koja protiče kroz svjetlosni izvor izmjerena u vatima (W).
- Osvjetljenje (engl. Luminance) – količina informacije koju posmatrač dobije iz svjetlosnog izvora, mjeri se lumenima (lm) te odgovara snazi radijacije svjetlosnog izvora mjenog spektralnom funkcijom senzitiviteta.
- Svjetlina (engl. Brightness) – subjektivna percepcija (ahromatskog) svjetlosnog intenziteta

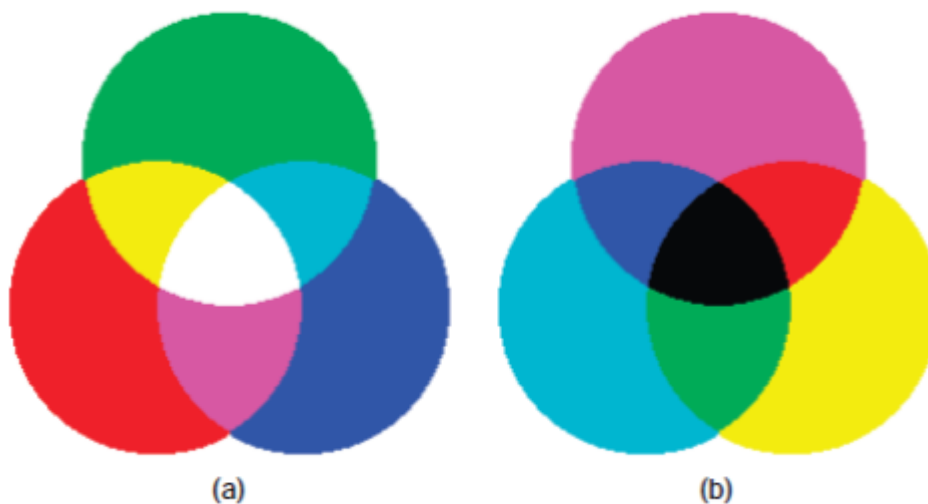
Ljudska retina (zadnja strana oka) sadrži dva receptora: *cones* i *rods*. Cones su uglavnom predviđeni za svjetliji spektar, djeluju u svjetlijim uslovima te postoje tri vrste: L, M i S (Long Medium i Short) koje odgovaraju talasnim dužinama ~610nm, ~560nm i ~430nm. Rods djeluju u tamnijim uslovima i ne mogu 'dekodirati' boju odnosno njen talas.



Commission Internationale de L'Eclairage (International Commission on Illumination) su definisali osnovne boje kao :

- crvena (700 nm)
- zelena (546.1 nm)
- plava (435.8 nm)

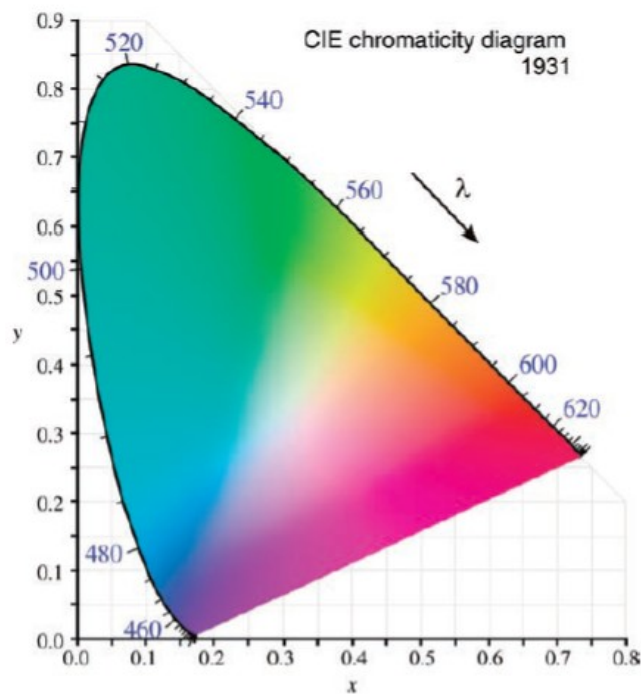
Sekundarne boje su definisane kao mješavina osnovnih boja: magenta (ljubičasta) = crvena + plava, cyan (tirkizna) = plava + zelena, žuta = zelena + crvena.



1.1 Hromatitet

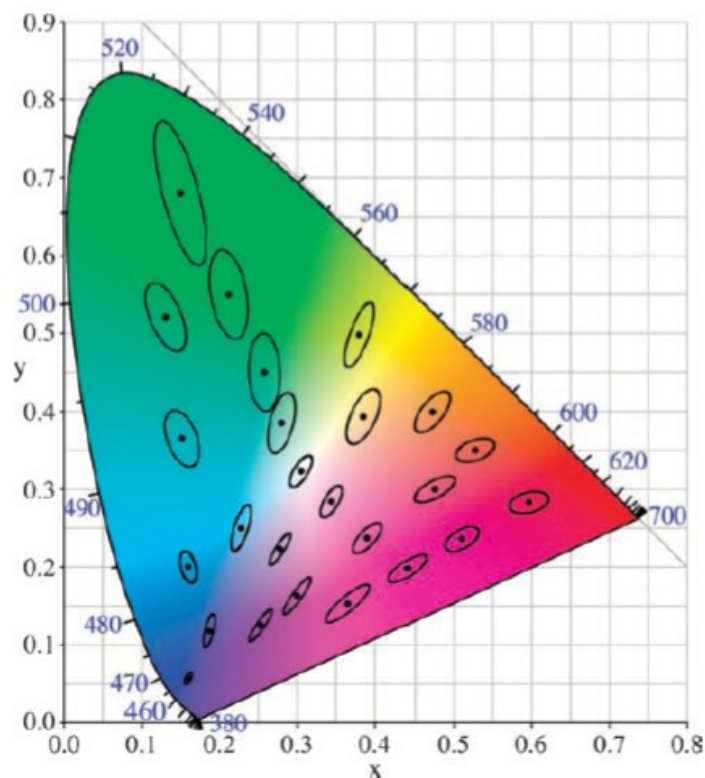
Hromatitet je objektivna specifikacija kvaliteta boje u ovisnosti od njene osvijetljenosti (engl. Luminance). Sastoji se od dva parametra: hue (h) i colorfulness (s), još se naziva i saturastion, chroma, intenzitet i sl. Ovaj broj parametra prati trihromatnost u čulu vida ljudi, o čemu je bilo ranije govora.

Bijela tačka osvijetljenja je neutralna referentna tačka koju karakteriše hromatitet, svi ostali hromatiteti moraju biti definisani u nekoj relaciji sa ovim referentnim koristeći polarne koordinate. Tako da hue predstavlja ugaonu komponentu a purity je radijalna komponenta, normalizovana. Purity je pojam ekvivalentan pojmu saturation u HSV modelu boja, o čemu će biti govora kasnije.



1.2 Percepcijski uniformni prostori boja

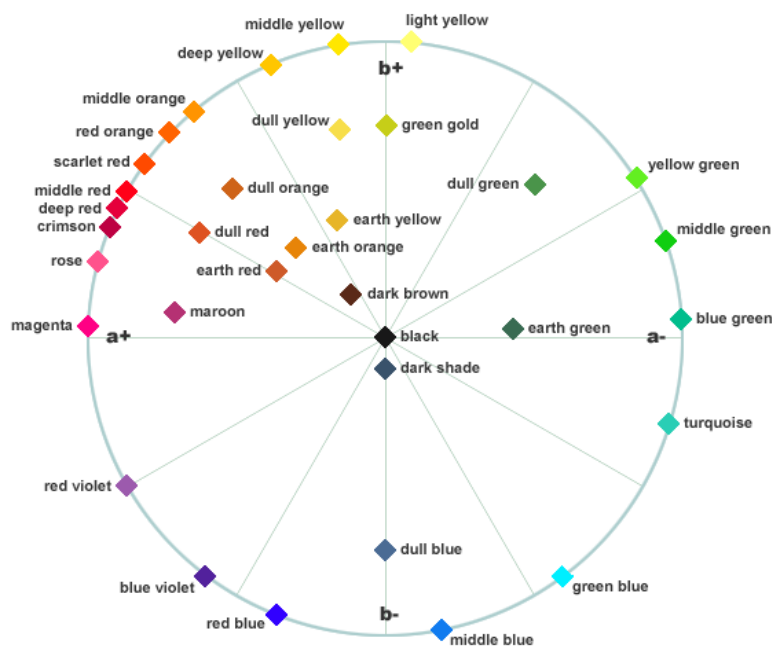
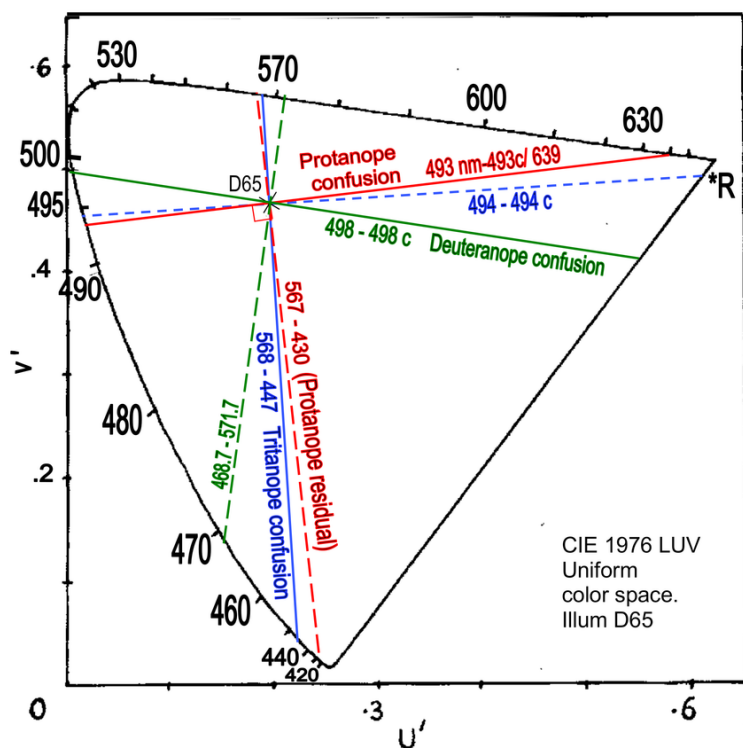
Ovaj pojam predstavlja sve boje na grafiku koje prosječnom ljudskom oku ne prave nikakvu razliku jedne između drugih, a ipak svojstveno se uniformno mijenjaju.



2. Percepcija boje ljudskom oku

Prethodni dijagram (iz poglavlja 1.2.) je dobijen iz eksperimenta; subjektima su date slike boja, počevši od jedne referentne pa nadalje, trebaju da prijave promjenu nakon što je uoče. Na taj način je primijećen 'pattern' u modelu boja tzv. *MacAdam-ove elipse*. Te elipse predstavljaju boje koje ljudsko oko ne razlikuje, ako posmatramo dijagram hromatiteta, počevši od boje označene tačkom u središtu elipse.

Na osnovu ovog istraživanja su napravljeni mnogi prostori boja, od kojih su najpoznatiji CIE $L^*u^*v^*$ (CIELUV) i CIE $L^*a^*b^*$ (CIELAB) sa ciljem ostvarenja takozvane perceptualne uniformnosti. To je jednostavno rečeno jednaka udaljenost između tačaka u prostoru boja koji daju jednake razlike u bojama.



3. Modeli predstavljanja boja

Modeli predstavljanja boja odnosno prostori boja su određen skup parametara tj. specifikacija posebno definisanog koordinatnog sistema i potprostora tog sistema gdje svaka boja odgovara jednoj i samo jednoj tački tog koordinatnog sistema. Mnogo je različitih modela definisano kroz historiju (u proteklih 400 godina!), definisanih za različite potrebe i namjene (fotografija, mjerenja boja, miješanja boja i sl.). Neki su potpuno odbačeni, a neki zadržani do dan danas. Neki od najpoznatijih koje ćemo razmatrati u ovom poglavlju su RGB, CMY/CMYK, HSV, YIQ(NTSC) i YcbCr. Ovi modeli se najčešće koriste u obradi slika, te zbog tog razloga će biti predstavljeni detaljnije.

3.1 RGB

Jedan od najjednostavnijih, najlogičnijih ali isto tako i najčešće korištenih modela je RGB (skraćeno od Red Green Blue). Zasniva se na Kartezijevom koordinatnom sistemu čije tri ose predstavljaju svaku od boja, normalizirane na interval $[0, 1]$.

RGB boje se obično predstavljaju u heksadecimalnom formatu sa 6 znakova, pri čemu prva dva predstavljaju crvenu, druga dva zelenu i posljednja dva plavu boju. Tako da bi crvena predstavljena na ovaj način bila FF0000. Također, ono što je interesantno za ovaj model je da vrijednosti RGB su funkcija *dubine piksela*, a ona se definiše kao broj bita potrebnih da se predstavi svaki od piksela. Tipična vrijednost je 24 bita : 3 koordinatne ose * 8 bita po svakoj osi. Na taj način, kocka koju dobijamo u prostoru sadrži oko 16 miliona različitih kombinacija boja.

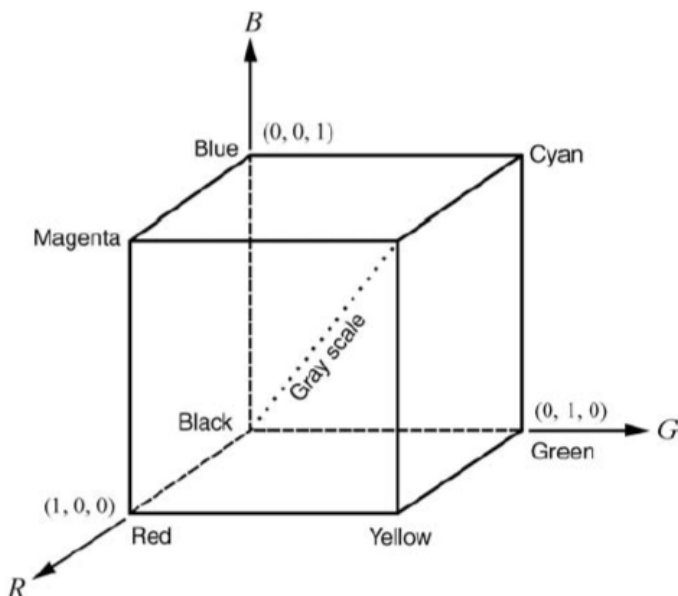


FIGURE 16.8 RGB color model.

MATLAB: ova kocka je generisana funkcijom `patch`. Ta funkcija pravi grafičke objekte koji se sastoje od jednog ili više poligona koji su specifikirani koordinatama njihovih tačaka te boje i osvjetljenja kao `patch` parametara.

3.2 CMY/CMYK

CMY (Cyan Magenta Yellow) je zapravo inverzija RGB sistema boja (koordinatni sistem je drugačije orijentisan). Obično se koristi u printerima, gdje svaka boja odgovara ketridžu tonera. Kako dodavanje svake od osnovnih boja daje *mutnu crnu*, često se ovaj model naziva i Cyan Magenta Yellow black.

Konverzija između RGB i CMY modela je prilično jednostavna, ali nema toliku praktičnu primjenu.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

MATLAB: pretvaranje između RGB i CMY se može uraditi funkcijom `imcomplement`.

3.3 HSV

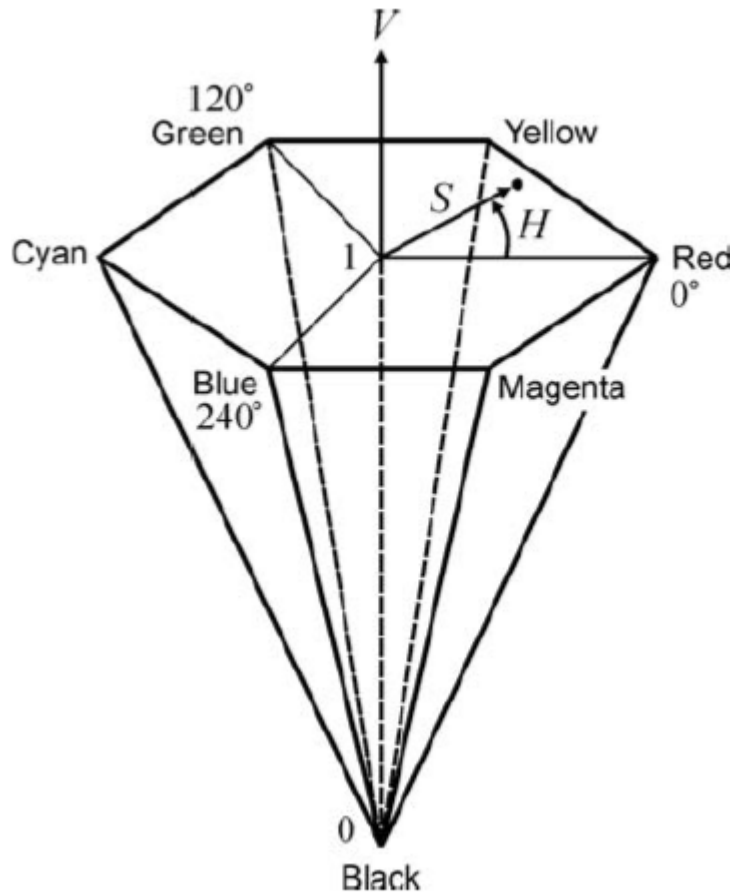
RGB i CMY modeli, iako dosta jednostavni i praktično primjenjivi u obradi slika i njihovom digitalnom predstavljanju, nisu način na koji ljudi vide boje, te ne bi mogli komunicirati u kodovima RGB ili CMY prostora jer jednostavno nije prihvatljivo. Zato je uveden HSV model (Hue Saturation Value). *Hue* predstavlja tip boje, ton, a često se naziva i imenom boje. *Saturation* predstavlja nivo zasićenosti odnosno čistoće (engl. purity), odnosno koliko je boja ‘razmućena’ u bijeloj boji. *Value* je zapravo svjetlina (engl. lightness) i predstavlja intenzitet odbitka svjetlosnih talasa o tu površinu.

Nažalost, ovaj model ima različite oznake i shvatanja, tako da HSV model objašnjen ovdje, ne mora biti nužno način kako se definiše u drugoj literaturi, jer postoje različita shvatanja pojmova hue saturation i lightness-a. Tako da kada posmatramo HSV model, zapravo posmatramo jednu familiju prostora boja.

Jedan način da dobijemo HSV model iz ranije navedenih modela je da posmatramo RGB kocku po glavnoj dijagonali (osa sive boje), pri čemu dobijamo šestougao paletu boja. Ako se krećemo po glavnoj osi piramide dobijene na ovaj način, šestougao postaje sve manji, te

Digitalno Procesiranje Signala - Color image processing

smanjujemo V . Za svaki šestougao, tri osnovne i sekundarne boje svjetla (svjetline) su definisane u njegovim čvorovima. Hue bi tada predstavljao ugao u odnosu na izvorište (origin), dok saturation predstavlja udaljenost od osa. Osim piramide prikazane na slici ispod, postoje i načini prikazivanja ovog modela u obliku cilindra i kupe, ali je šestostrana piramida znatno jednostavniji način vizualizacije svih boja i njihovih razgraničenja.



MATLAB: Pretvaranje između HSV modela i RGB se može izvršiti funkcijama `rgb2hsv` i `hsv2rgb`.

3.4 YIQ(NTSC)

NTSC model boja je standard definisan u Americi za analognu TV. Jedna od prednosti ovog modela je što je moguće potpuno razdvojiti grayscale komponentu od ostatka boja, što je bio veliki zahtjev u razvijanju televizora u boji. U NTSC modelu boja, tri komponente su osvjetljenje (engl. luminance, oznaka Y), te dvobojna razlika signala hue (I) i zasićenje (engl.

Digitalno Procesiranje Signala - Color image processing

saturation, oznaka Q). Oznake I i Q možda izgledaju nelogično, ali treba posmatrati sa ovog aspekta: i je faza, gdje je q (quadrature) pomjerena za 90 stepeni. Proces pretvaranja RGB u YIQ model je dat na sljedećoj slici:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

MATLAB: Pretvaranje između RGB i YIQ modela se može raditi funkcijama `rgb2ntsc` i `ntsc2rgb`.

3.5 YcbCr (digitalni video)

YcbCr model boja je dosta sličan prethodnom modelu, s tim da se ovaj koristi za digitalni video. Jedina razlika je što je Y osvijetljenje (luminance), Cb je razlika između plave komponente boje i referentne vrijednosti, Cr je razlika između crvene boje i referentne vrijednosti. Konverzija iz RGB u YcbCr:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

MATLAB: Pretvaranje između RGB i YCbCr modela se može vršiti funkcijama `rgb2ycbcr` i `ycbcr2rgb`

Kao zaključak za ovaj paragraf možemo izdvojiti činjenicu da se svako pretvaranje svodi na RGB model kao najzastupljeniji model u digitalnoj obradi slike. Najzastupljeniji je iz razloga što za svaku komponentu boje nam treba 8 bita da je jednoznačno predstavimo ($2^8 = 256$), te kako imamo tri komponente, 24 bita su dovoljna za predstavljanje 16 miliona različitih boja,

što nam znatno olakšava rad s njima, nego da ih predstavljamo u bilo kojem formatu od gore navedenih, iako i oni imaju velik značaj u nekim drugim sličnim oblastima.

4. Predstavljanje slika u MATLAB-u

Kao što smo ranije zaključili, slike u boji se obično predstavljaju kao RGB (24 bita po pikselu) ili indeksirane sa paletom boja (256). Ovi načini predstavljanja boja su neovisni od formata fajla, iako se GIF formati obično indeksiraju a JPEG ne. Mi ćemo se fokusirati na načine predstavljanja RGB sistema.

4.1 RGB slike

RGB slika u MATLAB-u predstavlja 3D niz dimenzija $M \times N \times 3$ gdje su M i N visina i širina slike (respektivno). Broj 3 predstavlja broj kanala za boje, odnosno kako se radi o RGB modelu, postoje 3 boje. Svaki individualni piksel je trojka komponenti R, G i B, te svaki niz veličine $M \times N$ se naziva komponenta slike i odgovara jednom od kanala. Ako govorimo o strukturi podataka koja je ove strukture i tipa double, raspon vrijednosti je između $[0.0, 1.0]$, dok ako se radi o uint8 ili uint16 opsezi su $[0, 255]$ odnosno $[0, 65535]$. Od ranije spomenute 3 komponente, svaka ima dubinu od 24 bita, tako da ukupno možemo predstaviti $24^3 = 16777216$ boja.

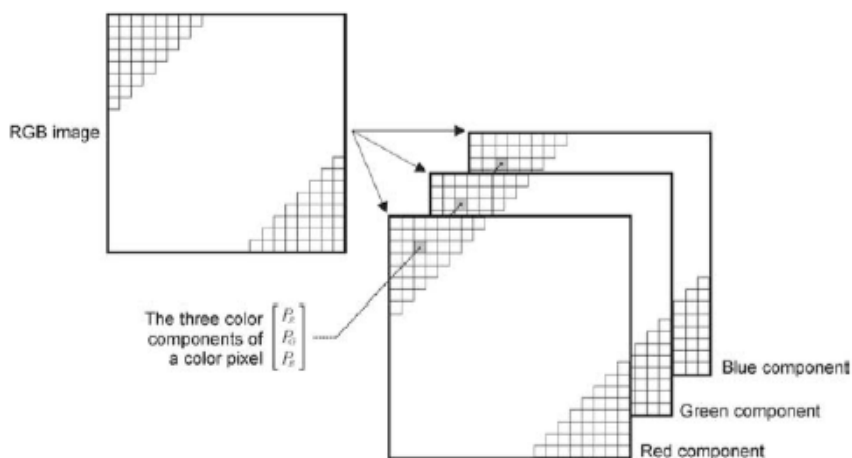


TABLE 16.4 Color Maps in MATLAB

Name	Description
hsv	Hue-saturation-value color map
hot	Black-red-yellow-white color map
gray	Linear gray-scale color map
bone	Gray scale with tinge of blue color map
copper	Linear copper-tone color map
pink	Pastel shades of pink color map
white	All white color map
flag	Alternating red, white, blue, and black color map
lines	Color map with the line colors
colorcube	Enhanced color-cube color map
vga	Windows color map for 16 colors
jet	Variant of HSV
prism	Prism color map
cool	Shades of cyan and magenta color map
autumn	Shades of red and yellow color map.
spring	Shades of magenta and yellow color map
winter	Shades of blue and green color map
summer	Shades of green and yellow color map

4.2. Indeksirane slike

Indeksirana slika je matrica cijelih brojeva X gdje svaki broj je *pokazivač* na neki red RGB vrijednosti u sekundarnoj matrici (mapi) koja se još naziva i mapa boja. Slika je predstavljena kao niz uint8, uint16 ili double vrijednosti. Mapa boja je niz $M \times 3$ doubleova gdje je svaki element u opsegu $[0.0, 1.0]$. Svaki red predstavlja jednu od boja, crvena zelena i plava, u tom poretku. Mehanizam indeksiranja funkcioniše po slici ispod: ako je X uint8 ili uint16, sve komponente sa vrijednosti 0 pokazuju na prvi red mape, sve sa 1 na drugi red itd. Međutim, ako se radi o double vrijednostima, sve komponente sa vrijednosti manjom od 1.0 pokazuju na prvi red itd.

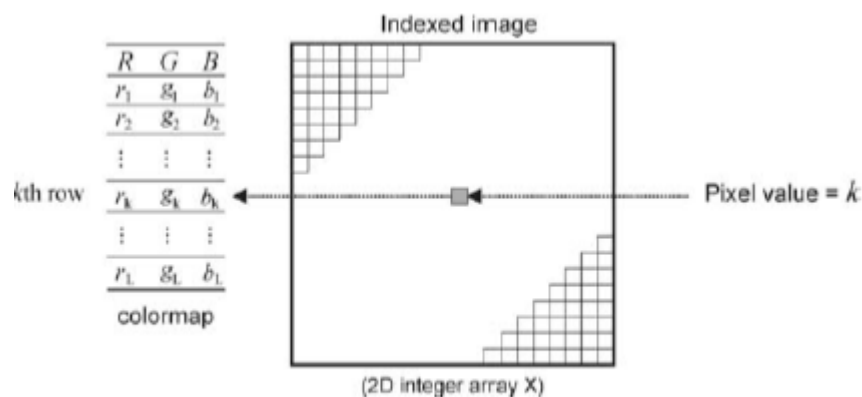
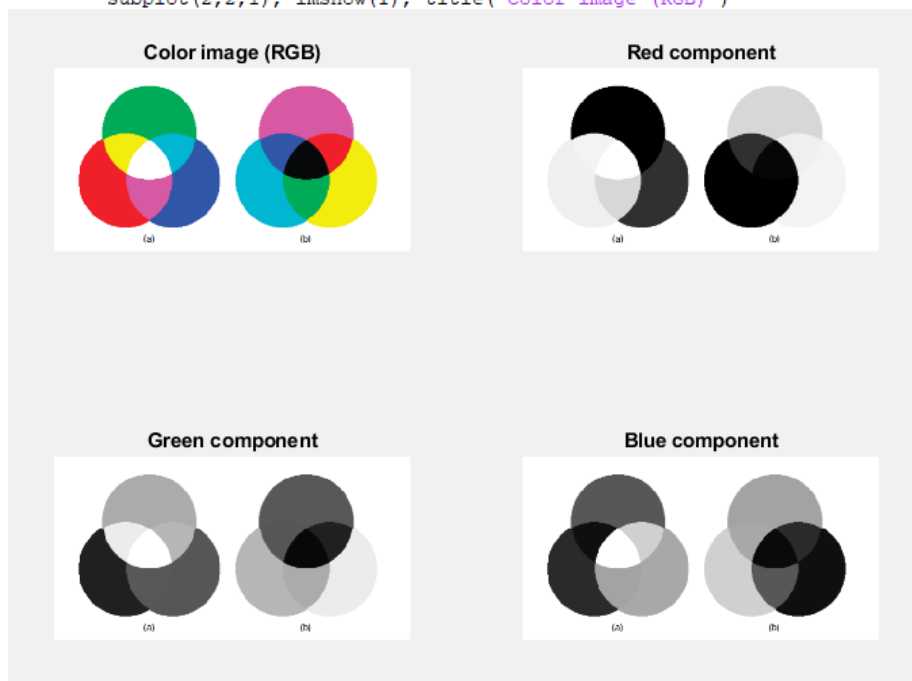


FIGURE 16.15 Indexed color image representation.

Primjer u MATLAB-u: rastavljanje slike po kanalima boja (rastavljanjePoRGB.m)

```
I = imread('slika2.PNG');
size(I)
class(I)
subplot(2,2,1), imshow(I), title('Color image (RGB)')
```



5. Pseudocolor obrada slika

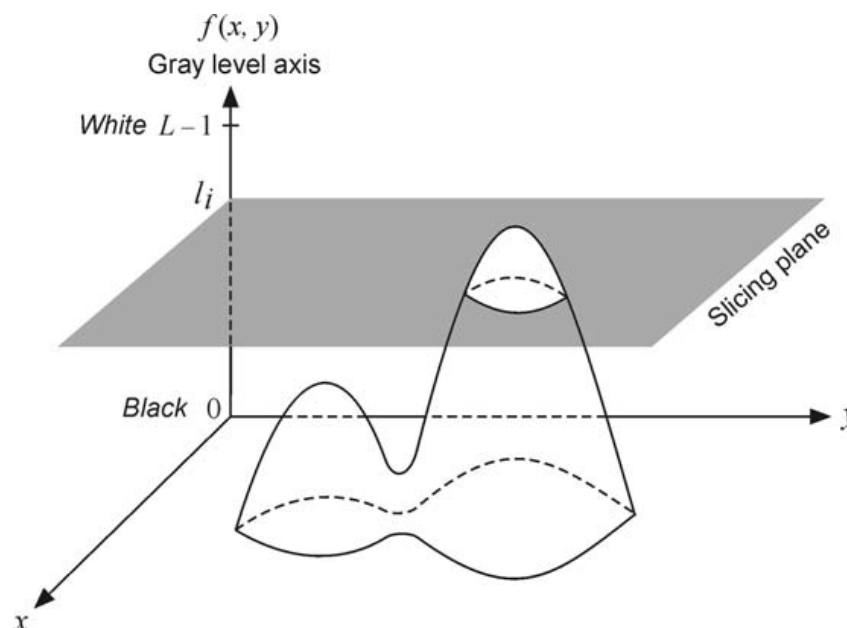
Svrha pseudocolor tehnologija za obradu slika je naglašavanje monohromatske slike radi prilagodbe ljudskom oku. Naime, suptilne varijacije nijansi sive često maskiraju skrivaju i maskiraju regije od interesa na slici. Zamjena nijansi sive bojama dovodi do bolje vizualizacije i bolje vizualizacije bitnih detalja, s obzirom da je ljudsko oko sposobno detektovati više hiljada različitih nijansi boja, a samo 100 ili manje nijansi sive. Ovo je naročito korisno u polju medicine za detekciju tumora na snimcima.

Tipično rješenje je korištenje color lookup tabele (CLU) dizajnirane za mapiranje kompletnog spektra nijansi sive na (mnogo) manji broj boja. Termin pseudo boja je iskorišten da naglasi da dodijeljena boja nema nikakve veze sa izvornom bojom.

5.1 Intensity Slicing

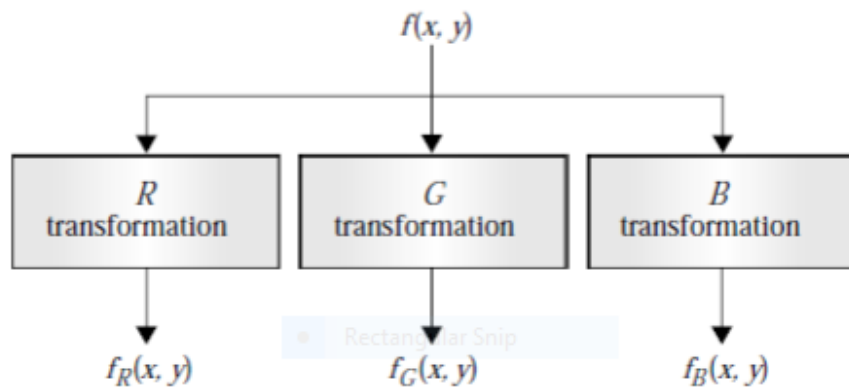
Intensity slicing je najjednostavnija i najbolje poznata pseudocolor tehnika. Ako monohromatsku sliku predstavimo kao 3D graf nijansi sive gdje najizraženiji vrhovi odgovaraju najsvjetlijim nijansama. Tehnika odgovara postavljanju nekoliko ravnih paralelnih sa koordinatnim sistemom slike. Svaka ravan siječe 3D funkciju a rezultat je nekoliko intervala nijansi sive. Svakoj strani grafa je onda dodijeljena različita boja.

U matlabu intensity slicing se vrši pomoću grayslice funkcije.



5.2 Transformacija nijansi sive u boje

Alternativni pristup pseudocoloring-u se koristi tri različite funkcije transformacije nad svakim pixelom slike dodjeljujući rezultat svake transformacije kanalu boje.

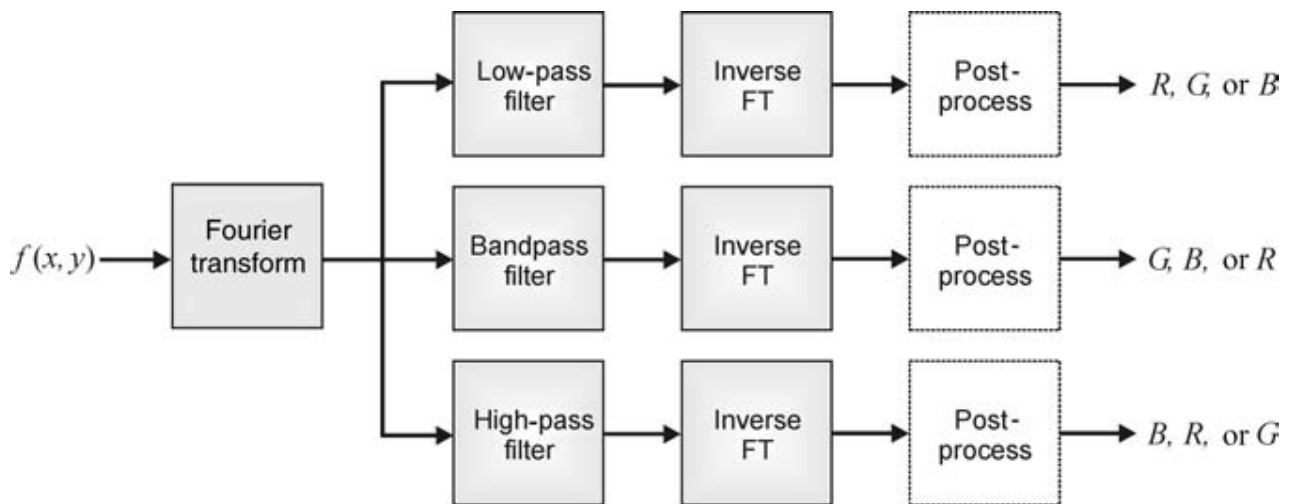


Dijagram za pseudocoloring koristeći tri funkcije transformacije

5.3 Pseudocoloring u frekventnom domenu

Pseudocoloring se također može vršiti u frekventnom domenu primjenom Fourierove transformacije na originalnu sliku, a onda primjenjujemo nisko-propusni, visoko-propusni i filtera propusnog opsega na transformisane podatke. Na izlaze tri različita filtra su onda primijenjene inverzne transformacije i rezultati se koriste kao R, B i G komponente slike.

Digitalno Procesiranje Signala - Color image processing



(a)



(b)

6. Full color obrada slika

Ovaj paragraf objašnjava obradu digitalnih slika u boji. Tehnika je relativno nova, ali postaje sve zastupljenija zahvaljujući jeftinom hardveru za čuvanje, prikaz i printanje slika u boji.

Postoji nekoliko tehničkih poteškoća prilikom ekstrapolacije tehnika za obradu monohromatskih slika na slike u boji. Jedna od njih koja se posebno ističe je izbor odgovarajućeg modela boja za zadatak. Važnost izbora ovih boja će biti očigledna kroz primjere prikazane u ovom poglavlju.

Postoje dva metoda procesiranja slika u boji.

- Rastavljanje po komponentama: Nakon selektovanog odgovarajućeg modela boja, svaka komponenta (R, B ili G) se individualno obrađuje.

Digitalno Procesiranje Signala - Color image processing

- Metoda vektora: Pikseli boja se tretiraju kao vektori

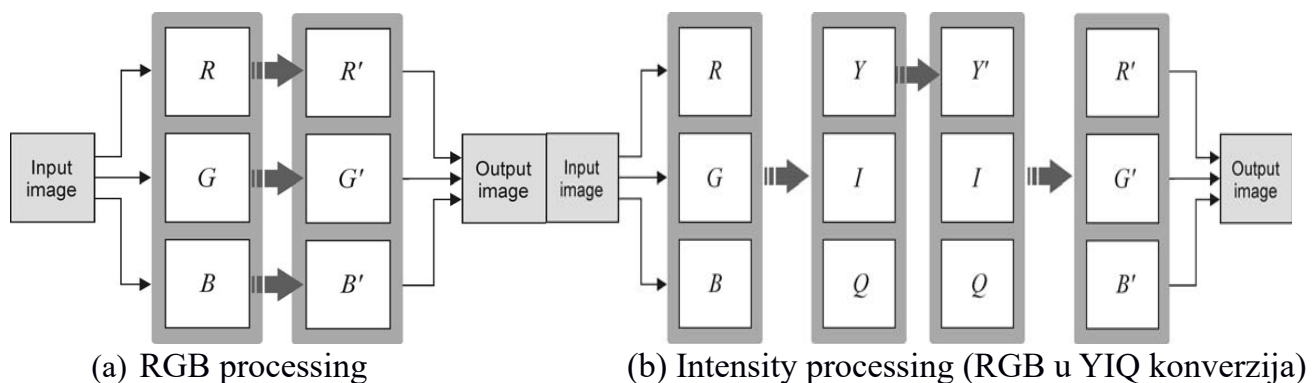
$$\mathbf{c}(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

Dvije metode su ekvivalentne ako i samo ako:

- Operacije nad vektorima komponenti su nezavisne od ostalih komponenti.
- Proces se može primijeniti i nad vektore i nad skalare.

U nastavku fokus će biti na rastavljanju po komponentama.

Nekoliko tehnika za procesiranje slika može biti primijenjeno na pojedinačnim R, B i G kanalima boja ulazne slike. Ove tehnike zahtijevaju pristup komponentama ekvivalentnim monohromatskoj verziji ulazne slike. Ovo se naziva Intensity processing, a prethodno RGB processing.



6.1 Transformacija boja

Koncept transformacije nijansi sive se može proširiti i na slike u boji.

$$g(x, y) = T[f(x, y)]$$

Originalna transformaciona jednačina se može adaptirati tako da input $g(x, y)$ i output $f(x, y)$ budu slike u boji gdje svaki individualni piksel nije više integer ili double nego uređena trojka (R, G i B vrijednost za svaki piksel).

Digitalno Procesiranje Signala - Color image processing

S obzirom da ova transformacija ne ovisi od pozicije piksela niti njegovog okruženja, možemo uvesti pojednostavljenu notaciju

$$s_i = T_i(r_1, r_2, \dots, r_n), \quad i = 1, 2, \dots, n$$

gdje su r_i i s_i komponente izlazne $f(x, y)$ i ulazne $g(x, y)$ slike, respektivno, n je broj zasebnih komponenti a T_i je skup transformacija (mapping function) za svaku komponentu. Npr. za RGB slike broj komponenti $n = 3$ a r_1, r_2 i r_3 odgovaraju R, G i B vrijednosti za svaki piksel slike.

Primjeri transformacija odnosno mapping funkcija:

- Transformacija intenziteta (Intensity mapping) predstavljena kao

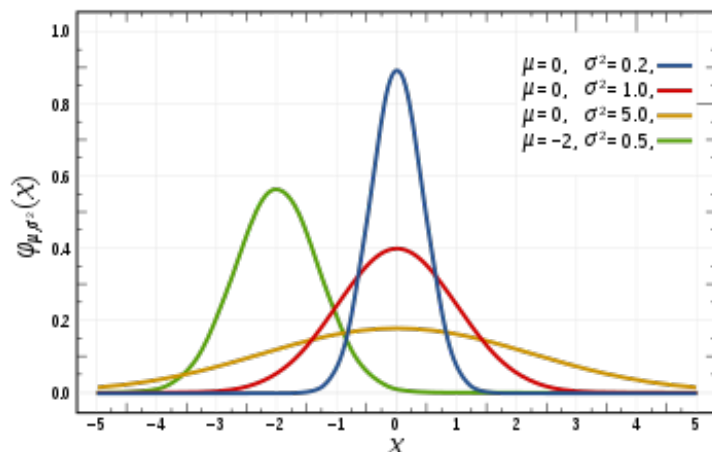
$$g(x, y) = kf(x, y)$$

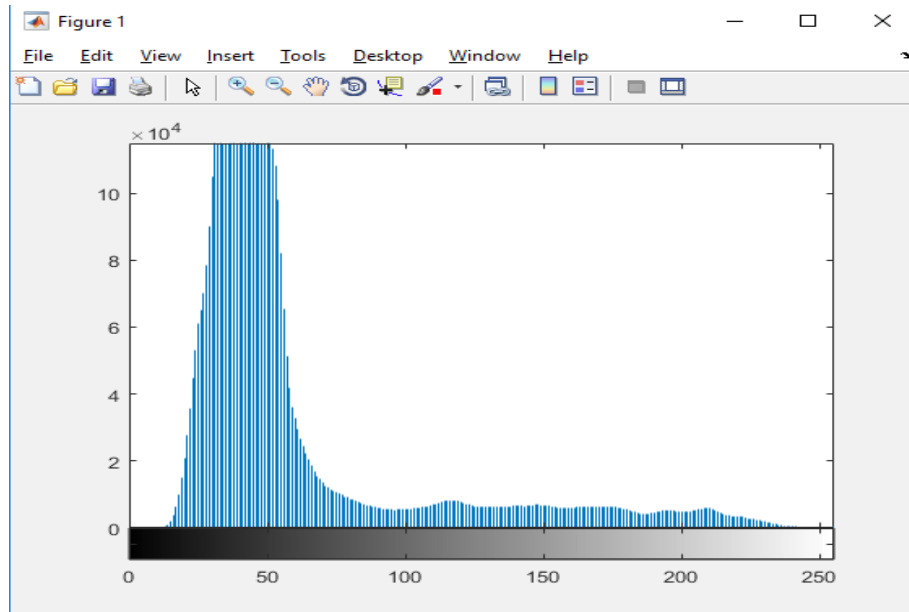
gdje je očito da ako je $k > 1$ izlazna slika će biti svjetlija od ulazne, a za $k < 1$ bit će tamnija.

- Komplement boje (Color Complements) svaku nijansu zamijeni njenim komplementom, ako je slika prikazana RGB modelom primijenimo trivijalnu transformaciju na svaki kanal boje, a u koliko je prikazan HVS modelom, trivijalna transformacija se primijeni nad V, nad H se primjenjuje netrivialna transformacija koja se brine o nekonzistentnosti a S kanal ostaje nepromjenjen.
- Color Slicing transformacija se primjenjuje tako što sve boje van opsega od interesa zamijenimo neutralnom bojom (npr. sivom) dok boje unutar opsega ostaju nepromijenjene.

6.2 Histogram Processing

Koncept histograma se može proširiti na slike koje se predstavljaju sa tri histograma od N ($4 \leq N \leq 256$) bita. Histogram tehnika se može primijeniti na slike predstavljene modelom koji dozvoljava rastavljanje na komponente osvjetljenja i hrominanse (*luminance and chrominance components*) npr. HSI model, I je komponenta osvjetljenja koja je obrađena, dok su H I S komponente ostale nepromijenjene.





Na prethodnom primjeru imamo jednu interesantnu situaciju; naime, portret poznatog matematičara Gaussa (Carl Friedrich Gauss 1777-1855) formira u histogramu upravo Gaussovu krivulju.

6.3 Poboljšanje i izoštravanje slike u boji

Linearno okruženjem orijentisano izoštravanje i poboljšavanje (*sharpening and smoothing*) slika može se proširiti na slike u boji uz izmjenu da originalni 3×3 kernel postaje niz vektora.

Formulacija vektora filtera prosječenja na RGB sliku koristeći okruženje S_{xy} oko koordinata x i y izgleda ovako:

$$\bar{c}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{(s,t) \in S_{xy}} R(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} G(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} B(s, t) \end{bmatrix}$$

Jednačina pokazuje da se rezultat može dobiti primjenom prosječenja okruženja na svakom kanalu boje koristeći standardni proces za prosječenje nijansi sive.

Izoštrenje slike kodirane RGB modelom može se postići primjenom operatora izoštravanja (Laplacian) posebno na svaku komponentu, a zatim iskombinovati rezultate.

Smoothing and sharpening opracije mogu biti primijenjene na slike predstavljene HSI ili YIQ modelom, procesiranjem komponente osvjetljenja i kombiniranjem rezultata s originalnim hrominanse komponentama.

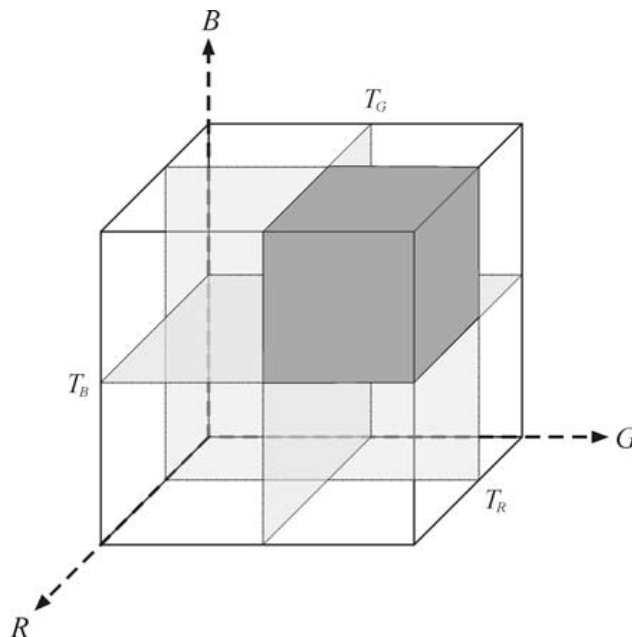
6.4 Redukcija šumova boje

Uticaj šuma na sliku u boju ovisi o modelu kojim je slika predstavljena. Čak i kada je samo jedan kanal (R, G ili B) pod uticajem šuma, konverzija u drugi model boja će proširiti šum na sve komponente. Tehnike za linearnu redukciju šuma mogu se primijeniti na svaki kanal zasebno s dobrim rezultatima.

6.5 Segmentacija slike bazirana na boji

Opisat ćemo dva načina segmentacije:

- *Thresholding* – postoji nekoliko načina da se thresholding namijenjen monohromatskim slikama proširi na slike u boji. Osnovna ideja je podijeliti obojeni dio u nekoliko regija, koje bi trebale da se sadrže smislene objekte sa slike, koristeći odgovarajući threshold. Jednostavna opcija je definiranje jednog ili više thresholda za svaku komponentu boje, što rezultuje u podjeli RGB kocke iz koje se može izolirati manja kocka s bojama iz regiona od interesa.



- *Segmentacija slike u boji u RGB vektorskom prostoru* – Moguće je specificirati threshold relativne distance između bilo koje boje i referentne boje u RGB prostoru. Ako je referentna boja (R_0, G_0, B_0) , threshold pravilo može biti predstavljeno kao :

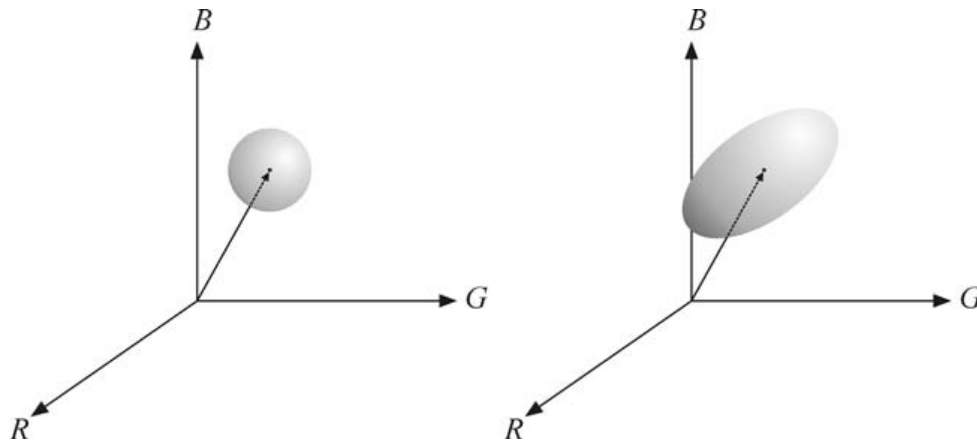
$$g(x, y) = \begin{cases} 1 & d(x, y) \leq d_{\max} \\ 0 & d(x, y) > d_{\max} \end{cases}$$

gdje je

$$d(x, y) = \sqrt{[f_R(x, y) - R_0]^2 + [f_G(x, y) - G_0]^2 + [f_B(x, y) - B_0]^2}$$

Threshold specificiran ovim jednačinama ustvari predstavlja sferu u RGB prostoru, čiji je centar referentna boja. Svi pikseli čija boja leži unutar ili na površini sfere će biti postavljeni na 1, svim ostalim ćemo dodijeliti vrijednost 0.

Ako jednačinu thresholda generalizujemo tako što uvedemo različitu vrijednost thresholda za svaku primarnu boju, rezultat će biti elipsoid u RGB prostoru.



Definiranje sfere ili elipsoida u RGB prostoru

Kodovi u MATLAB-u koji demonstriraju na konkretnim primjerima objasnjenje tehnike:

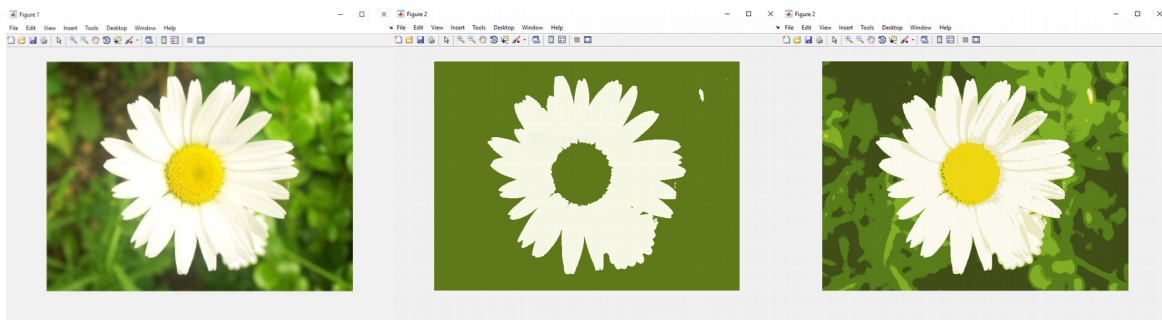
- *Segmentacija*

U ovom primjeru koristili smo sliku sa tri dominantne boje zelena, bijela i žuta.

Koristeći predefinisanu funkciju `rgb2ind` uradili smo segmentaciju. Rezultati značajno variraju od `n` i nejednake distribucije svjetla.

```
I = imread('cvijet-segmentacija.jpg');  
n = 6;  
[I2,map2] = rgb2ind(I,n,'nodither');  
imshow(I)  
figure, imshow(I2,map2)
```

Digitalno Procesiranje Signala - Color image processing



(a) Originalna slika

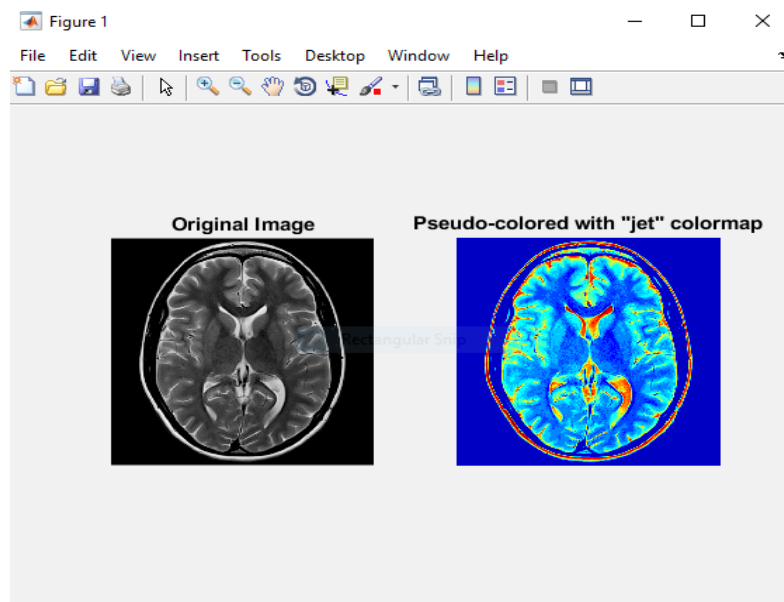
(b) Slika nakon segmentacije
za $n = 2$

(c) Slika nakon segmentacije
za $n = 6$

- Pseudocolor obrada slike

Koristimo MATLAB funkciju `grayscale` da bi nad slikom uradili pseudocoloring. S obzirom da je ova tehnika korisna za polje medicine (za detekciju tumora) kao primjer smo odabrali smo MRI snimak ljuskog mozga, MATLAB kod i rezultat su prikazani ispod.

```
I = imread('mri-pseudocoloring.jpg');  
subplot(1,2,1);  
imshow(I);  
title('Original Image');  
I2 = grayscale(I,16);  
subplot(1,2,2);  
imshow(I2, colormap(jet(16)));  
title('Pseudo-colored with "jet" colormap');
```



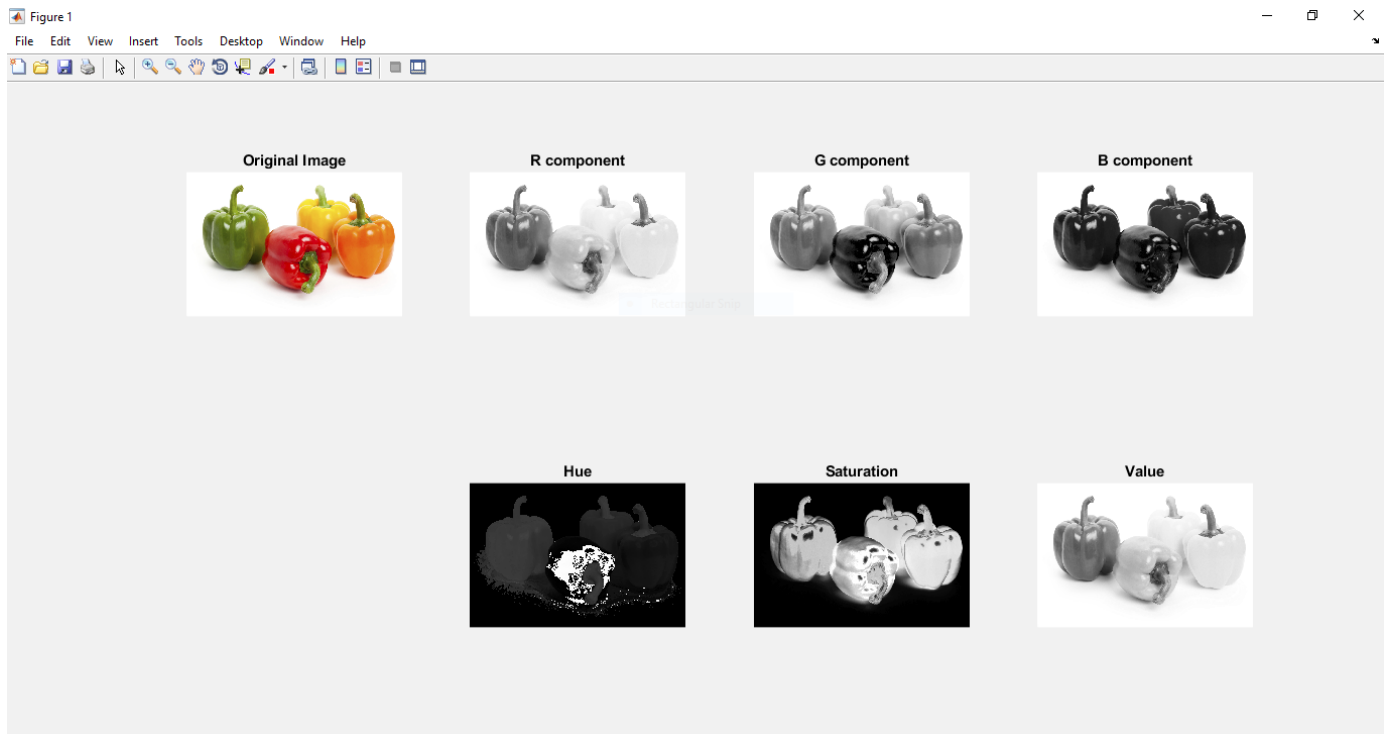
Digitalno Procesiranje Signala - Color image processing

- *Full-color obrada slike*

Prvo ćemo pokazati rastavljanje slike na R, G i B komponentu, vizualizacija svake od komponenti u nijansama sive je vrlo intuitivna jer predstavlja intenzitet komponente tj. koliko je koje komponente korišteno pri formiranju finalne boje.

Zatim smo uradili pretvaranje originalne slike u HSV i rastavili je na komponente. Vizualizacija ovih komponenti nije intuitivna, možemo primijetiti da na osnovu komponente H i S ne možemo zaključiti ništa o stvarnoj boji slike, dok V komponenta izgleda kao grayscale verzija originalne slike.

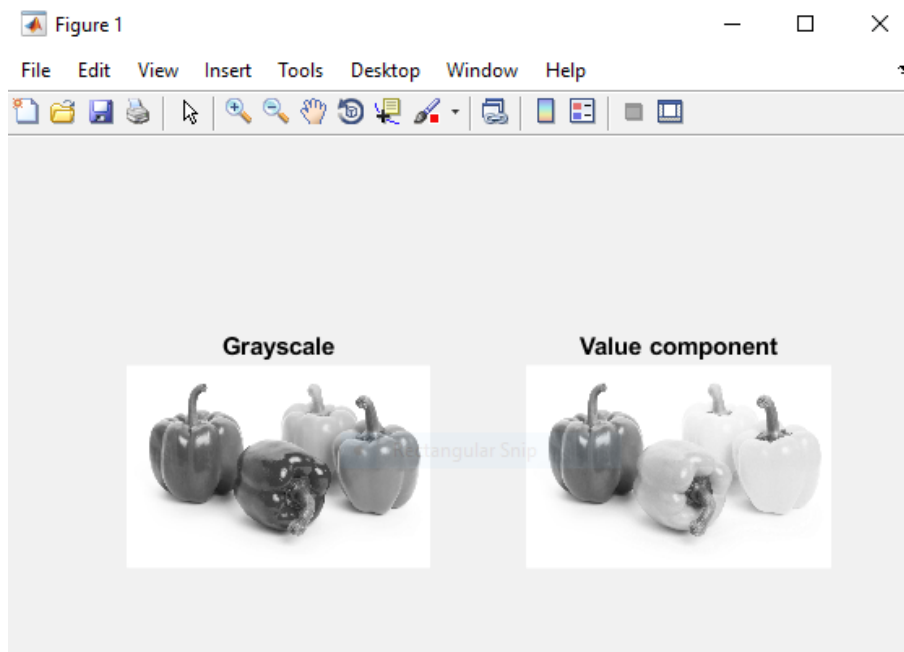
```
I = imread('peppers-fullcolor.jpg');
subplot(2,4,1), imshow(I), title('Original Image');
subplot(2,4,2), imshow(I(:,:,1)), title('R component');
subplot(2,4,3), imshow(I(:,:,2)), title('G component');
subplot(2,4,4), imshow(I(:,:,3)), title('B component');
Ihsv = rgb2hsv(I);
subplot(2,4,6), imshow(Ihsv(:,:,1)), title('Hue')
subplot(2,4,7), imshow(Ihsv(:,:,2)), title('Saturation');
subplot(2,4,8), imshow(Ihsv(:,:,3)), title('Value');
```



Samo ćemo sliku prikazati na skali sivih(kao grayscale) i to uporediti sa V komponentom HSV modela.

Digitalno Procesiranje Signala - Color image processing

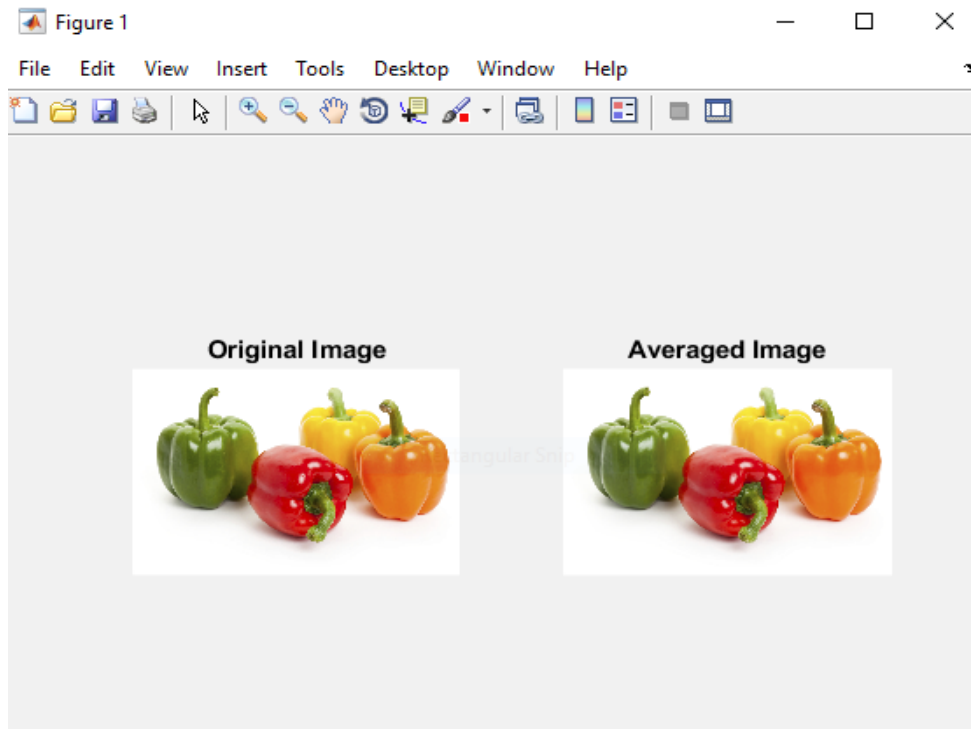
```
I = imread('peppers-fullcolor.jpg');
Igray = rgb2gray(I);
subplot(1,2,1), imshow(Igray), title('Grayscale');
subplot(1,2,2), imshow(Ihsv(:,:,3)), title('Value component');
```



Poboljšavanje slike radimo tako što primijenimo smoothing filter na svaku komponentu slike a zatim uradimo rekonstrukciju.

```
clear all;
I = imread('peppers-fullcolor.jpg');
fn = fspecial('average');
I2r = imfilter(I(:,:,1), fn);
I2g = imfilter(I(:,:,2), fn);
I2b = imfilter(I(:,:,3), fn);
I2(:,:,1) = I2r;
I2(:,:,2) = I2g;
I2(:,:,3) = I2b;
figure, subplot(1,2,1), imshow(I), title('Original Image');
subplot(1,2,2), imshow(I2), title('Averaged Image');
```

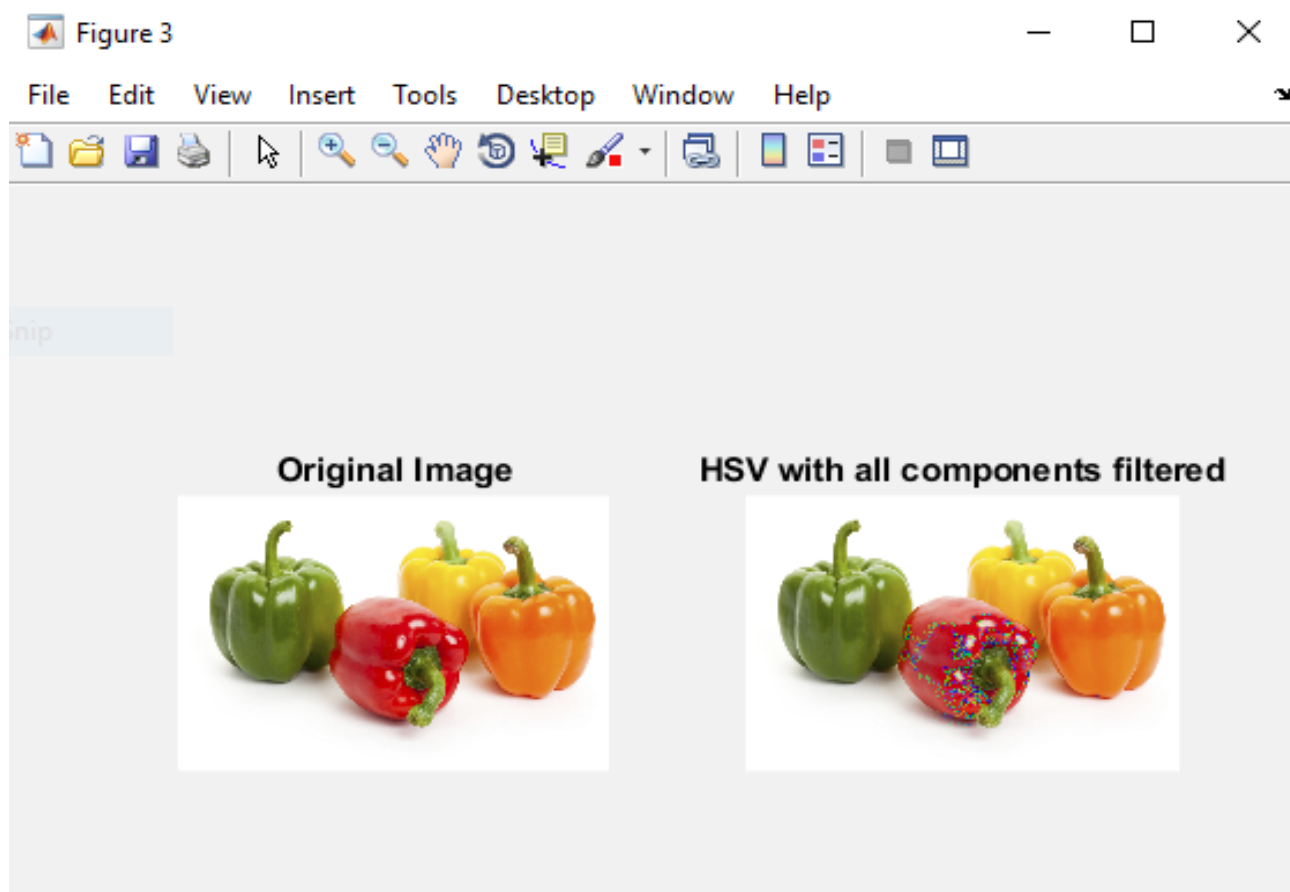
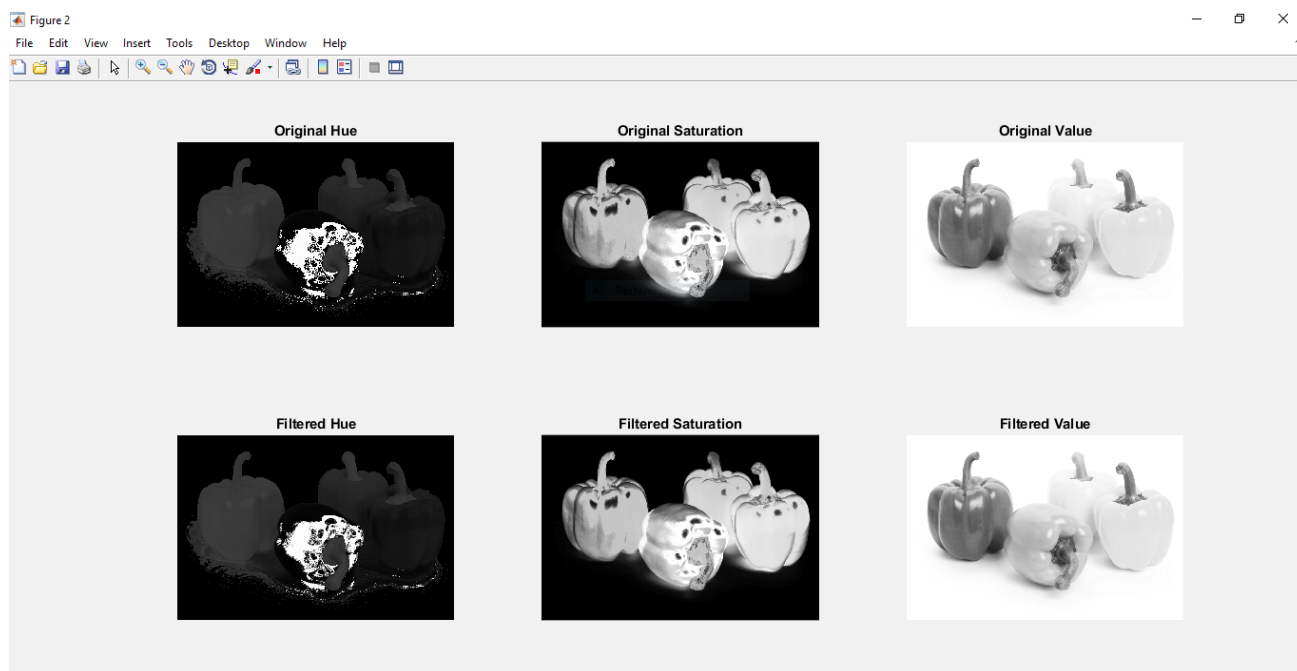
Digitalno Procesiranje Signala - Color image processing



Sad ćemo smoothing filter primijeniti na HSV model slike.

```
clf;
clear all;
I = imread('peppers-fullcolor.jpg');
fn = fspecial('average');
Ihsv = rgb2hsv(I);
Ihsv2h = imfilter(Ihsv(:,:,1), fn);
Ihsv2s = imfilter(Ihsv(:,:,2), fn);
Ihsv2v = imfilter(Ihsv(:,:,3), fn);
Ihsv2(:,:,1) = Ihsv2h;
Ihsv2(:,:,2) = Ihsv2s;
Ihsv2(:,:,3) = Ihsv2v;
figure, subplot(2,3,1), imshow(Ihsv(:,:,1)), ...
title('Original Hue');
subplot(2,3,2), imshow(Ihsv(:,:,2)), ...
title('Original Saturation');
subplot(2,3,3), imshow(Ihsv(:,:,3)), ...
title('Original Value');
subplot(2,3,4), imshow(Ihsv2(:,:,1)), ...
title('Filtered Hue');
subplot(2,3,5), imshow(Ihsv2(:,:,2)), ...
title('Filtered Saturation');
subplot(2,3,6), imshow(Ihsv2(:,:,3)), ...
title('Filtered Value');
figure, subplot(1,2,1), imshow(I), title('Original Image');
subplot(1,2,2), imshow(hsv2rgb(Ihsv2)), ...
title('HSV with all components filtered');
```


Digitalno Procesiranje Signala - Color image processing

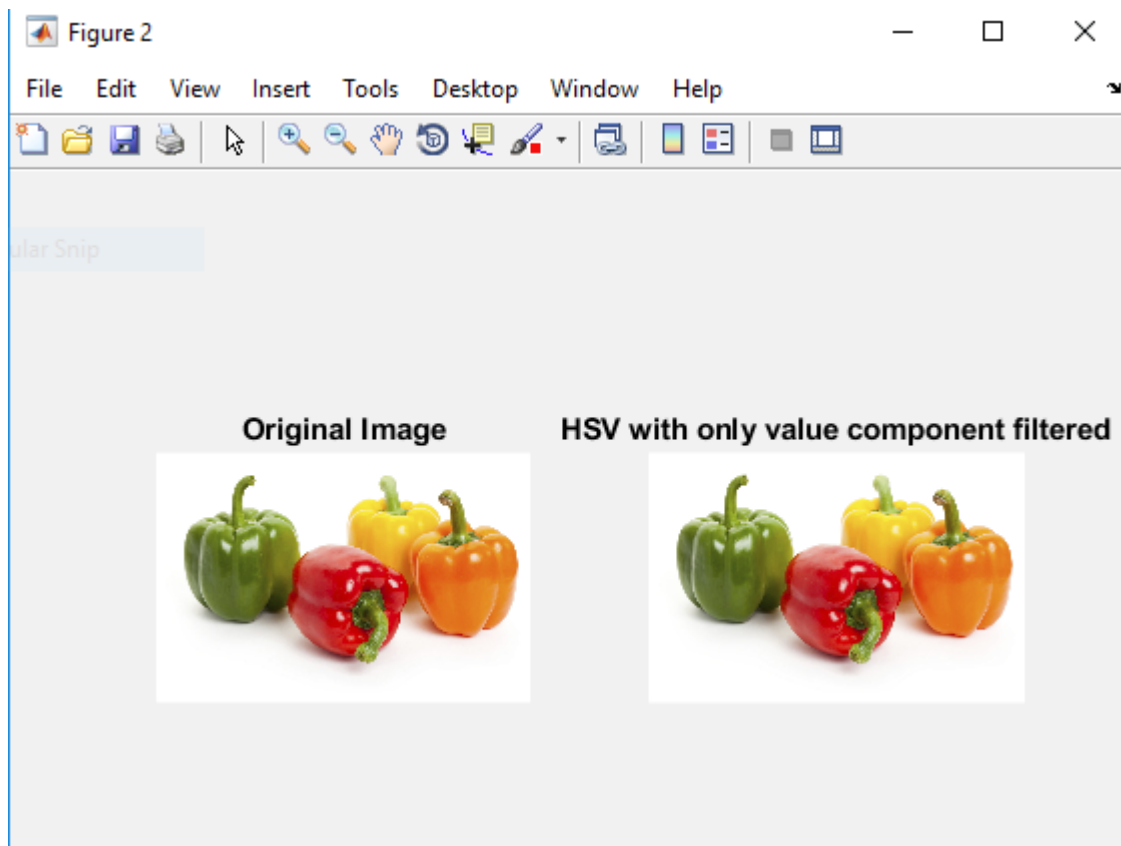


Digitalno Procesiranje Signala - Color image processing

Možemo primijetiti da nema smisla raditi filtriranje nad svim komponentama jer narušava rekonstrukciju slike, zato se u praksi obično radi samo filtracija V komponente.

Rezultati primjene filtra samo nad V komponentom:

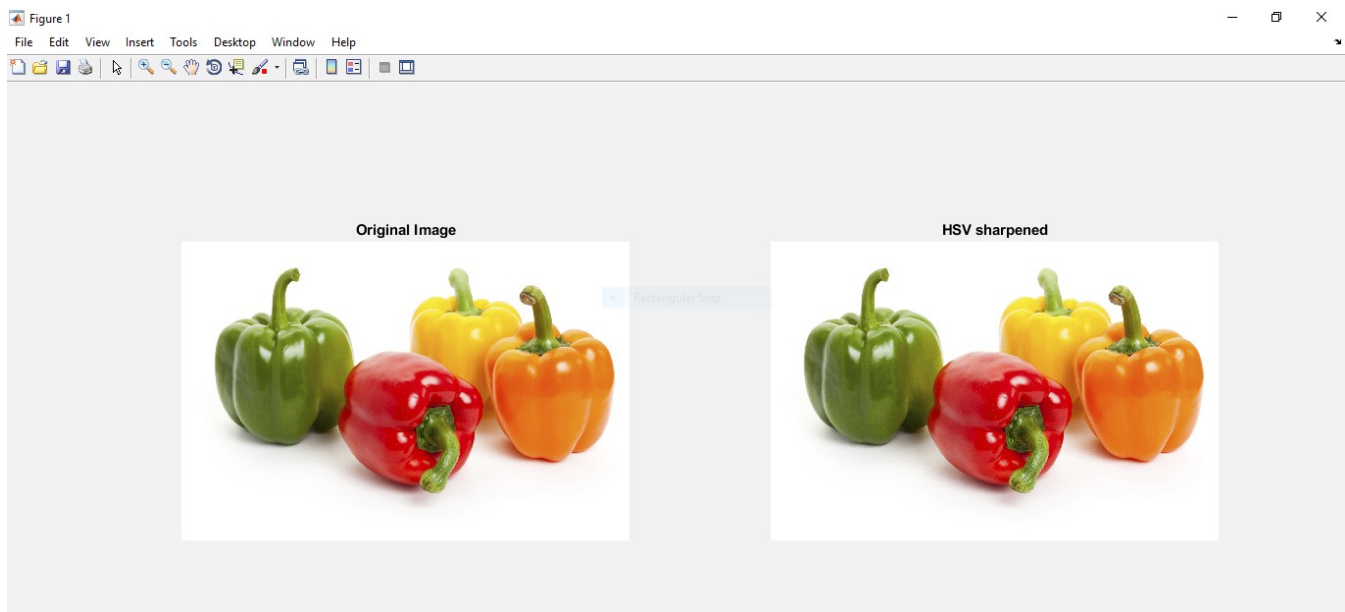
```
clf;
clear all;
I = imread('peppers-fullcolor.jpg');
fn = fspecial('average');
Ihsv = rgb2hsv(I);
Ihsv2v = imfilter(Ihsv(:,:,3), fn);
Ihsv3(:,:, [1 2]) = Ihsv(:,:, [1 2]);
Ihsv3(:,:,3) = Ihsv2v;
subplot(1,2,1), imshow(I), title('Original Image');
subplot(1,2,2), imshow(hsv2rgb(Ihsv3)), title('HSV with only value component
filtered')
```



Ostaje jos da damo primjer primjene Laplaciana tj. izoštravanja HSV slike te rekonstrukcija nazad u RGB.

Digitalno Procesiranje Signala - Color image processing

```
clear all;
I = imread('peppers-fullcolor.jpg');
Ihsv = rgb2hsv(I);
fn2 = fspecial('laplacian',0);
Ihsv4v = imfilter(Ihsv(:,:,3), fn2);
Ihsv4(:,:, [1 2]) = Ihsv(:,:, [1 2]);
Ihsv4(:,:,3) = imsubtract(Ihsv(:,:,3),Ihsv4v);
figure, subplot(1,2,1), imshow(I), title('Original Image');
subplot(1,2,2), imshow(hsv2rgb(Ihsv4)), title('HSV sharpened');
```



7. Literatura

1. Oge Marques - Practical image and video processing using MATLAB
2. Rafael Gonzales, Richard Woods, Steven Eddins – Digital image processing using MATLAB
3. Wikipedia članci
4. Jose Maria Giron-Sierra – Digital signal processing with MATLAB examples, volume 2