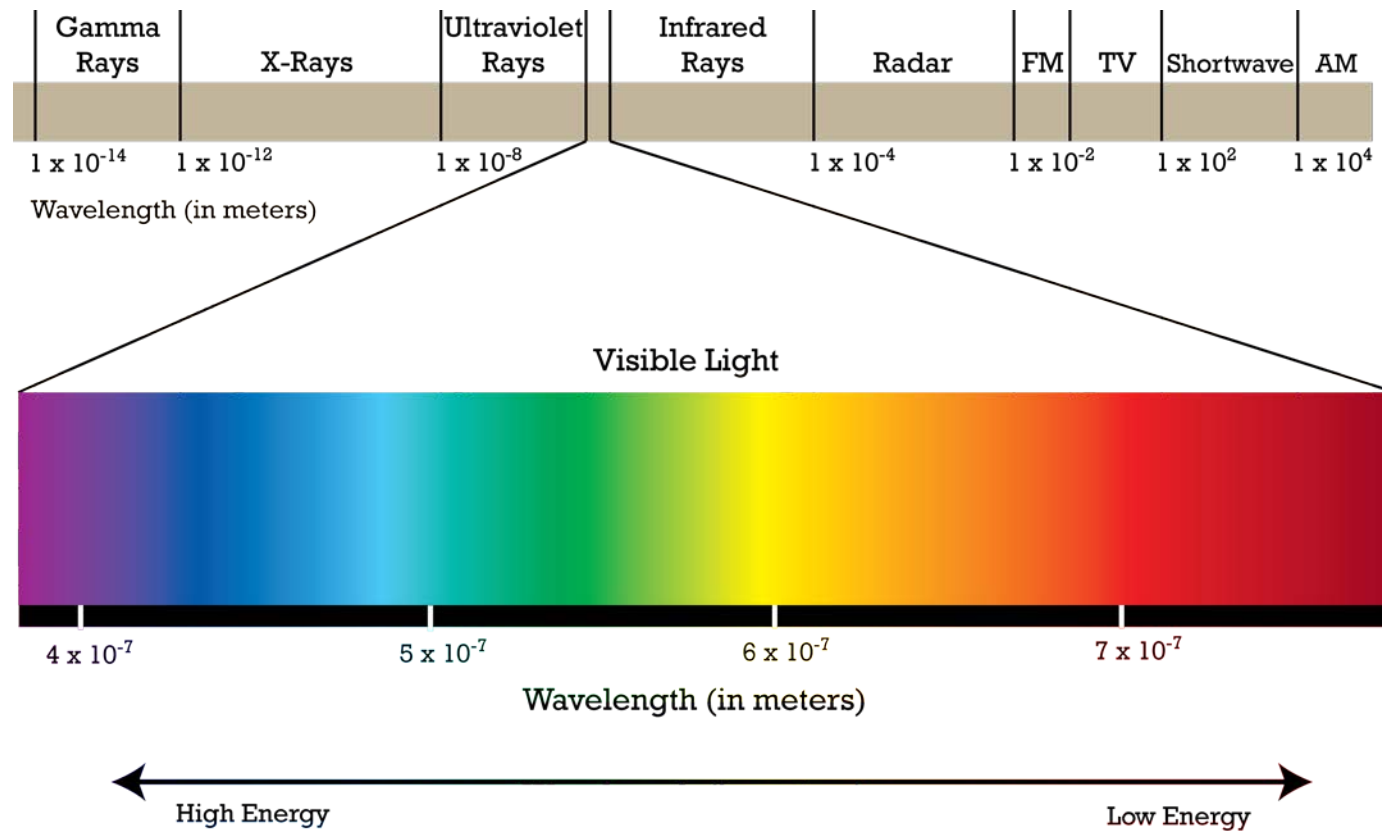




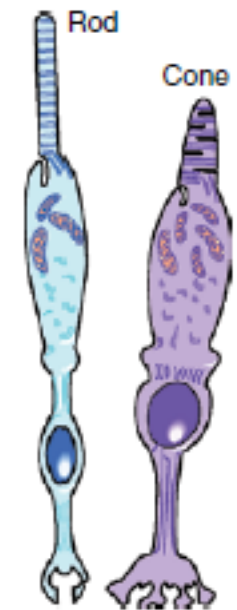
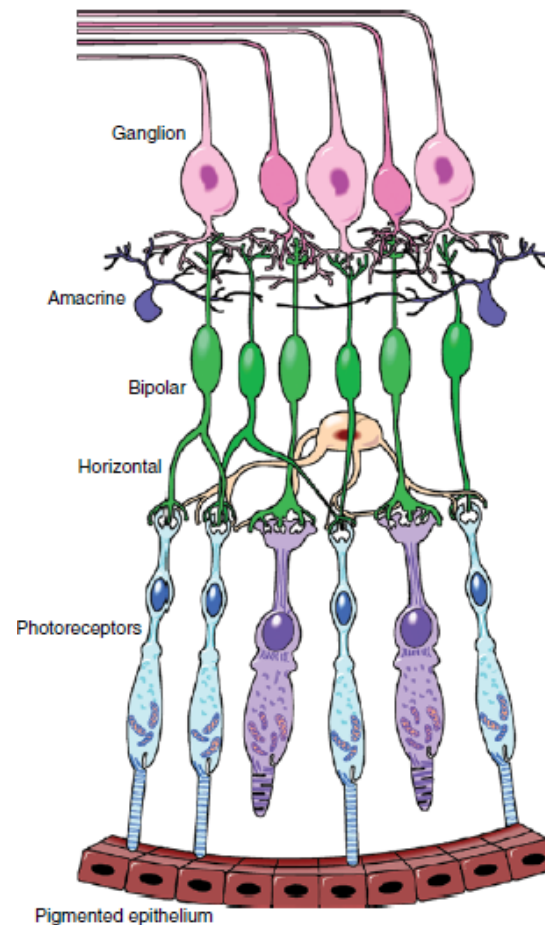
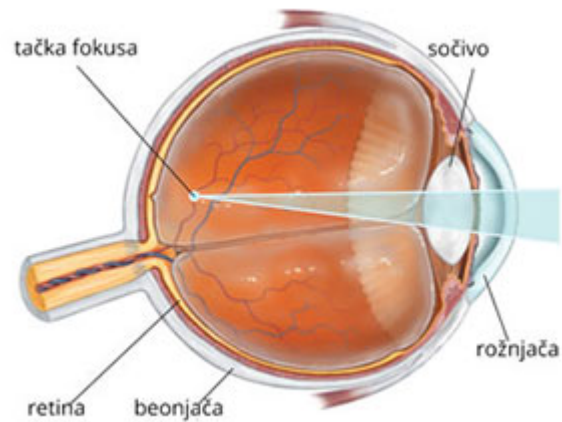
DIGITALNA OBRAĐA SLIKE

ČAS 2 – PERCEPCIJA SVETLOSTI, KOLOR SISTEMI

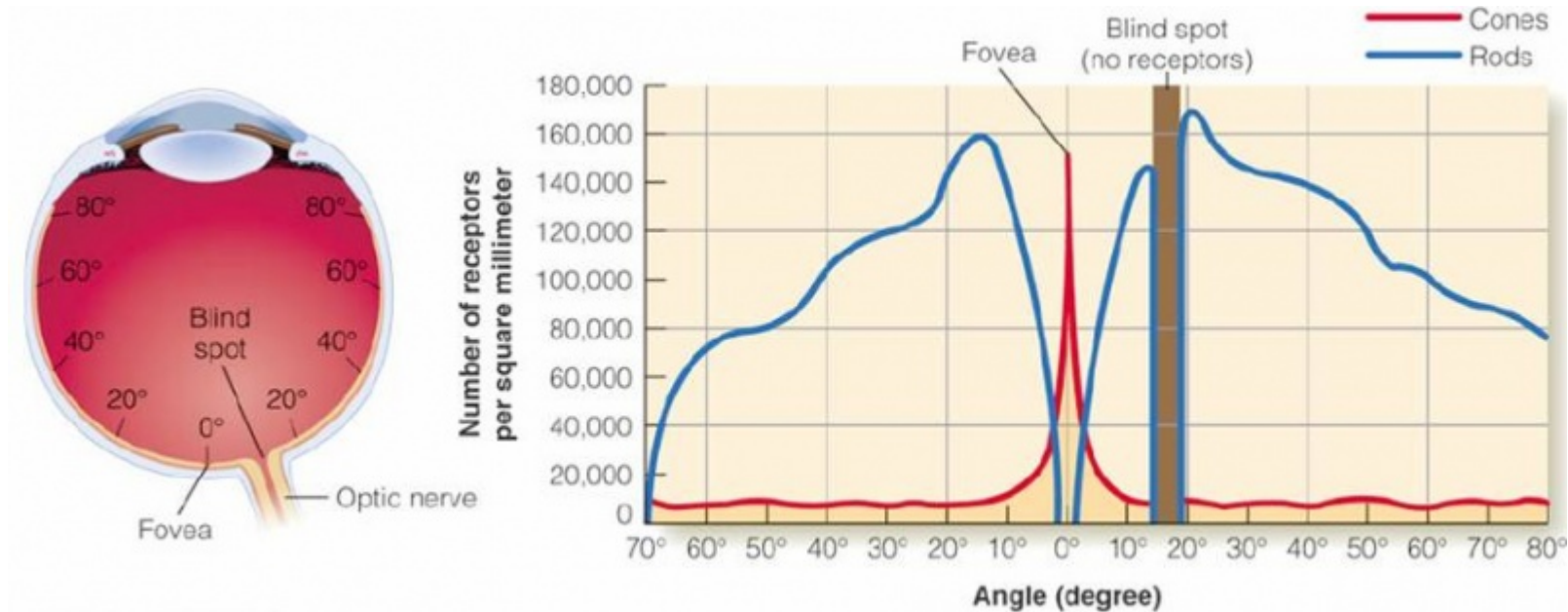
Spektar elektromagnetnog zračenja



Vizuelni sistem čoveka



Raspored čepića i štapića u retini



Distribucija fotoreceptora (čepića i štapića) u retini. Najveća gustina čepića je u žutoj mrlji (tačka jasnog vida) dok ih na periferiji skoro i nema. Štapići su uglavnom raspoređeni po perifernim delovima. Slepa tačka označava mesto na kojem optički nerv izlazi iz oka i na ovom mestu nema fotoreceptora.

Demonstracija postojanja slepe tačke



Zažmurite na desno oko i levim okom gledajte tačku. Polako menjajte položaj u odnosu na monitor i u jednom trenutku znak plus će nestati iz vidnog polja pošto zraci koji dolaze odatle padaju na slepu tačku.

Weber-Fechner-ov zakon

Najmanja razlika osvetljenosti koja se može vizuelno registrovati u određenom opsegu osvetljenja linearno zavisi od trenutne osvetljenosti.

$$\frac{\Delta L}{L} \approx 0.02$$

Zadatak:

Scena se snima kamerom koja ima linearnu funkciju odziva. Kontrast scene (odnos minimalne i maksimalne osvetljenosti) je 30:1. Za kvantizaciju informacije o osvetljaju se koristi 8 bita. Nakon kvantizacije scena se projektuje na displeju koji takođe ima linearnu prenosnu funkciju. Da li se u ovom slučaju može desiti da su neke dve tačke imaju različitu sjajnost na sceni a da izgledaju isto sjajne prilikom projekcije? Objasniti zašto i navesti primer. Koliko je minimalno bita potrebno za kodovanje osvetljenosti ove scene a da reprodukcija bude verna. Da li se ovaj broj bita može nekako smanjiti?

Weber-Fechner-ov zakon

Rešenje:

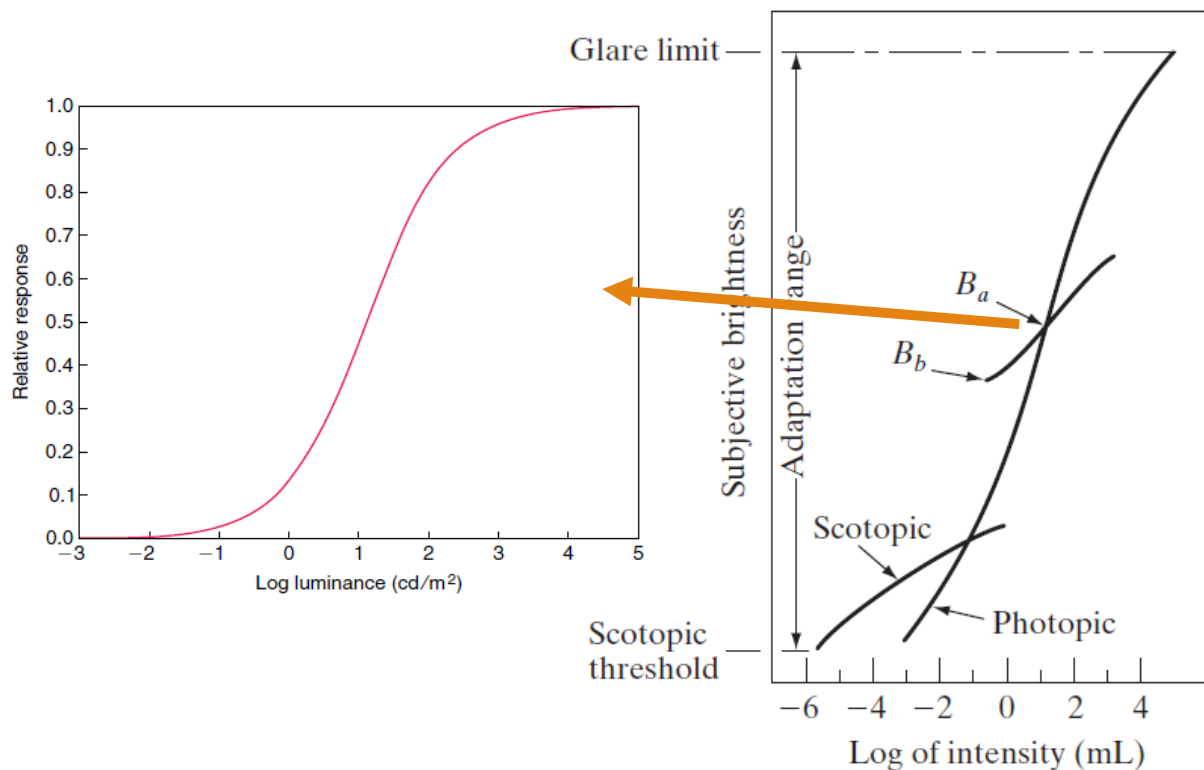
Ako je na raspolaganju 8 bita onda je korak između dve kodne reči $(301)/255 = 0.114$. Problem je što je ovakvim kodovanjem moguće predstaviti sledeće osvetljaje 1, 1.114, 1.228 itd međutim po Weber-ovom zakonu ljudsko oko može da percipira razliku i između 1 i 1.02 koji će u ovom slučaju biti predstavljeni identičnim kodom. Sa druge strane kodovi 20, 20.114, 20.228, 20.342 će predstavljati iste sjajnosti s obzirom da se razlika može uočiti tek između kodova 20 i 20.4.

Kako bi se izbegle greške pri reprodukciji potrebno je usvojiti minimalni korak od 0.02 što dovodi do $(29/0.02)+1=1451$ kodnih reči odnosno 11 bita.

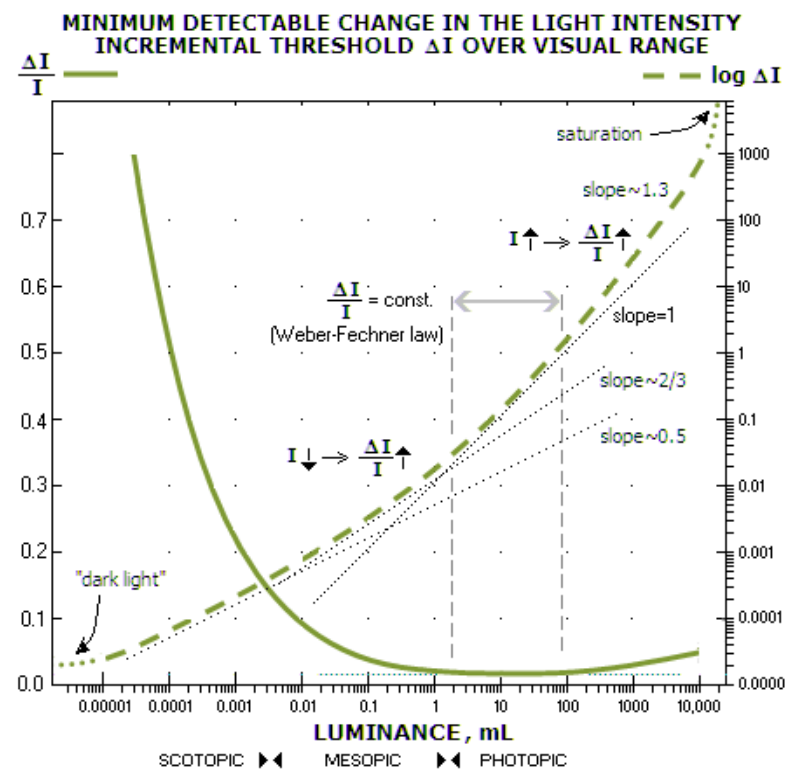
U slučaju da se informacije o osvetljaju pre kvantizacije prebace u logaritamski domen i tu kvantuju fiksnim korakom $\log(1.02)$, dobija se ispunjenost Weber-ovog zakona između svaka dva praga kvantizacije. Potreban broj kodnih reči je tada $\log(30)/\log(1.02) + 1 = 173$ odnosno sada je dovoljno 8 bita za kvantizaciju.

Sjajnost i kontrast

Adaptacija ljudskog vizuelnog sistema na različite nivoe osvetljenja



Zavisnost praga primetne razlike od osvetljenja (adaptacije)



Simultani kontrast

Pokrenuti demonstraciju `simultani_kontrast.m`

Dva prikazana objekta su iste osvetljenosti ali su im sjajnosti različite pošto se razlikuje pozadina.

- 1) Smanjivati osvetljenost levog objekta sve dok njegova sjajnost ne postane jednaka desnom objektu
- 2) Podesiti jednaku osvetljenost obe pozadine. Da li su sjajnosti objekata i dalje jednake?
- 3) Postaviti početno stanje kao sa slike i ponoviti eksperiment povećavajući osvetljenost desnog objekta.



Simultani kontrast

Pokrenuti demonstraciju `simultani_kontrast2.m`

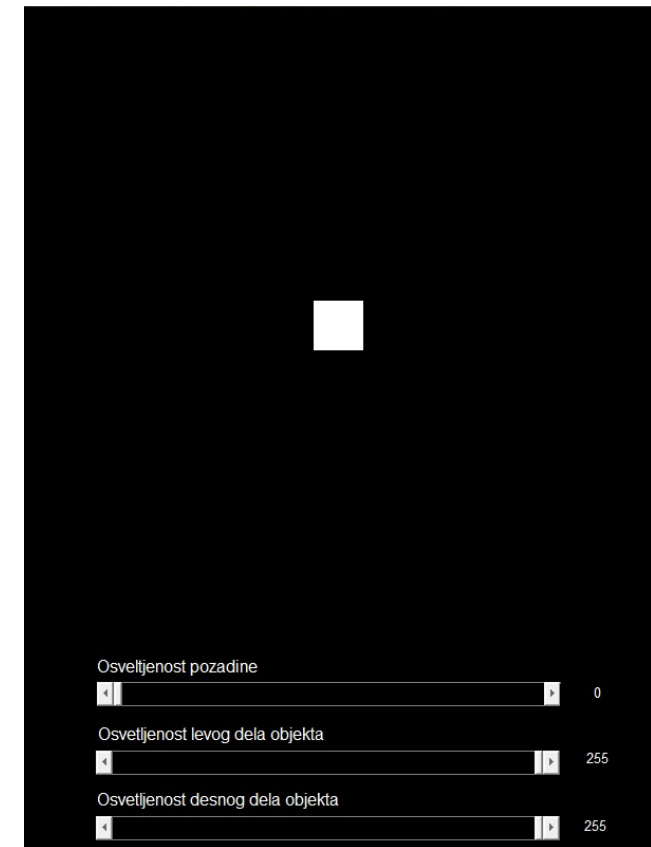
Povećati prozor aplikacije tako da zauzme ceo ekran. Sprovesti sledeći eksperiment. Podesiti osvetljenost levog dela objekta na 126 a osvetljenost desnog dela objekta na 130.

- 1) Podesiti osvetljenost pozadine na 0. Oceniti kontrast.
- 2) Podesiti osvetljenost pozadine na 255. Oceniti kontrast.
- 3) Podesiti osvetljenost pozadine na 128. Oceniti kontrast.

Podesiti osvetljenost objekta i pozadine na 0. Povećavati osvetljenost levog dela objekta dok ga ne uočite. Zabeležiti vrednost praga.

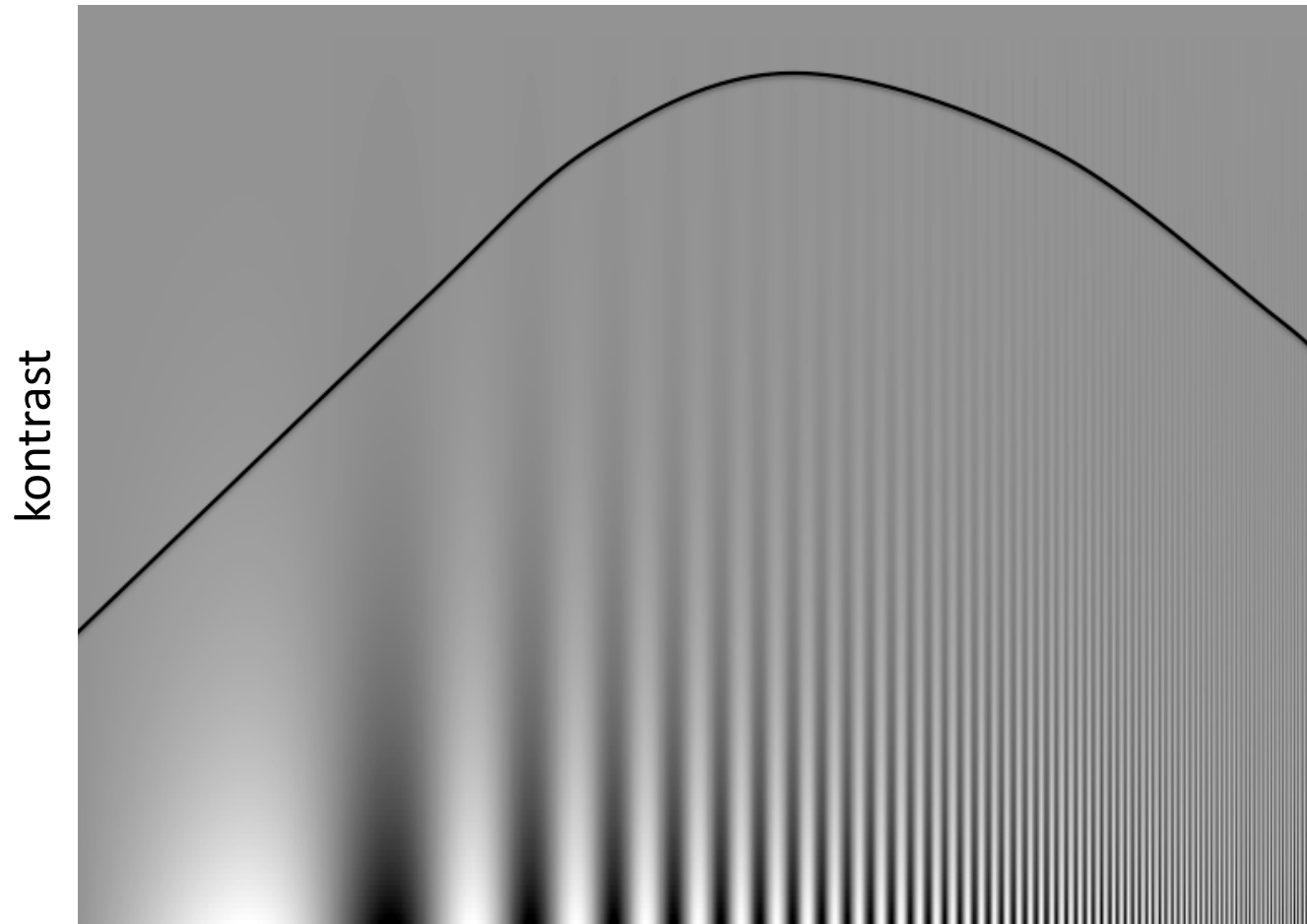
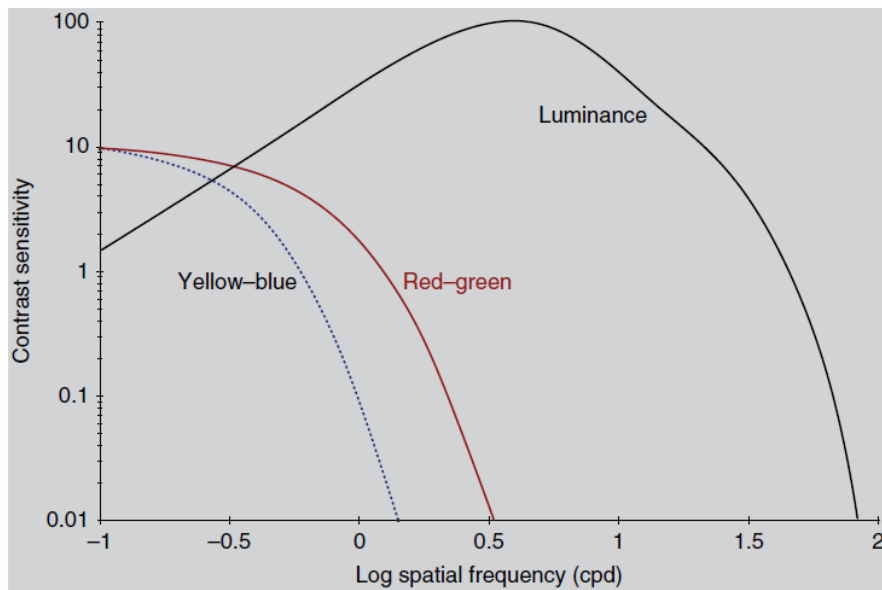
Podesiti osvetljenost objekta i pozadine na 255. Smanjivati osvetljenost levog dela objekta dok ga ne uočite. Zabeležiti vrednost praga.

Ponovite eksperiment u zamračenoj sobi.



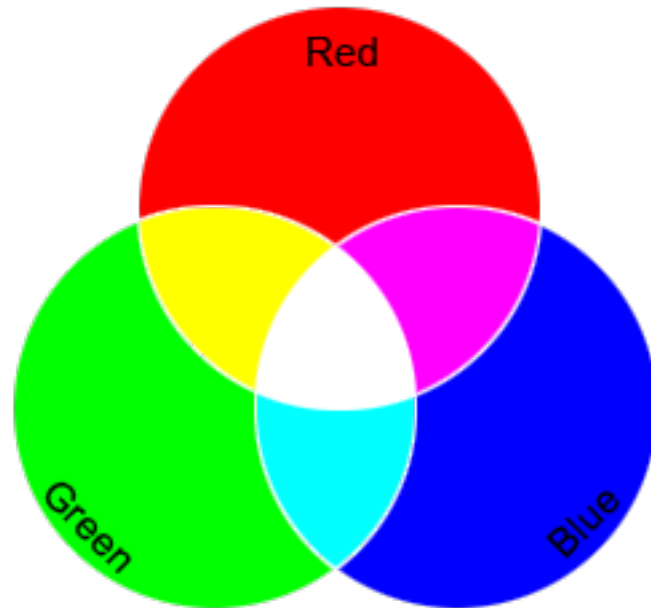
Osetljivost na kontrast (CSF)

Osetljivost na promenu kontrasta je manja za informaciju o boji u odnosu na luminansu.

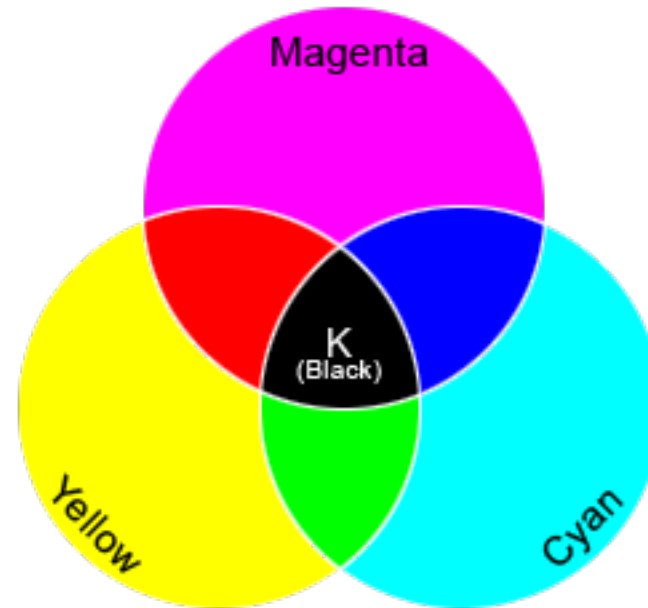


Predstavljanje boja (kolor sistemi)

Aditivni i subtraktivni kolor sistemi

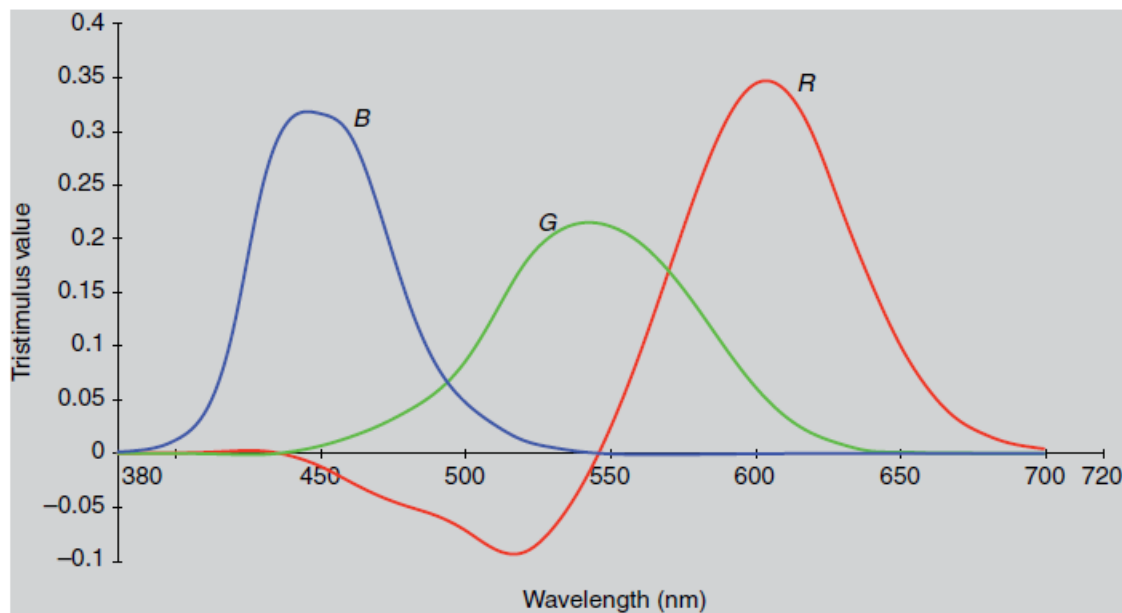


Additive Colors

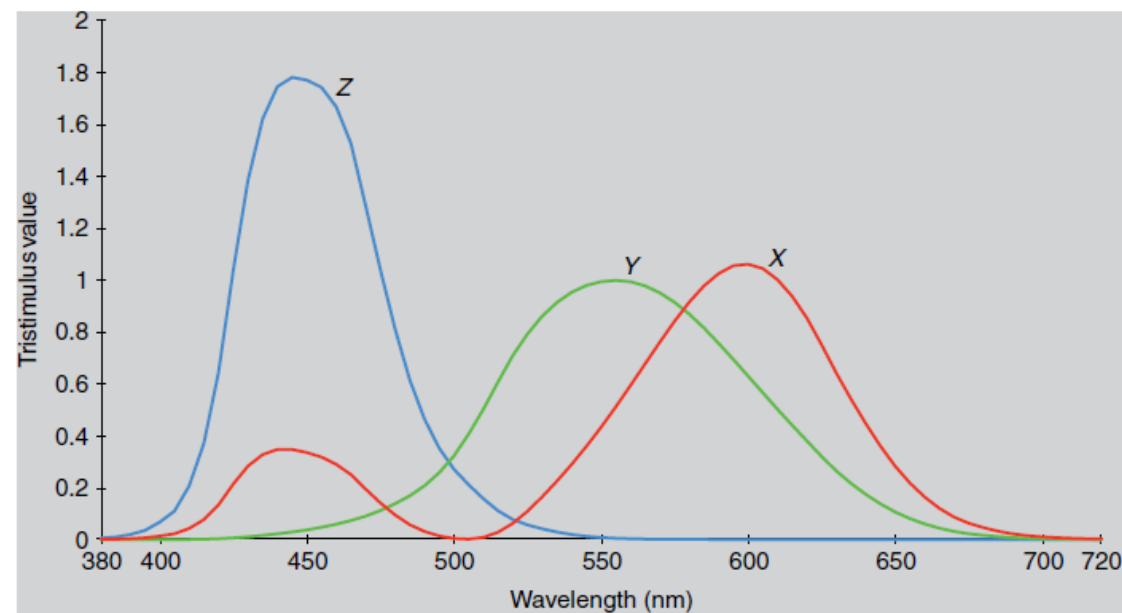


Subtractive Colors

Primarne komponente

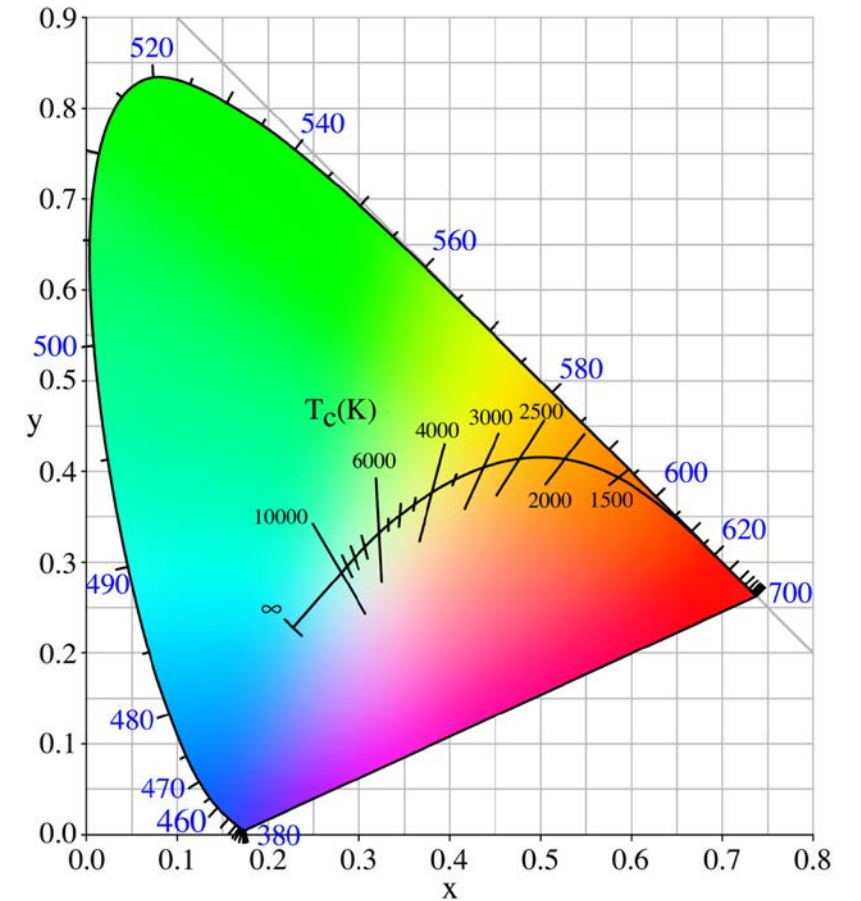
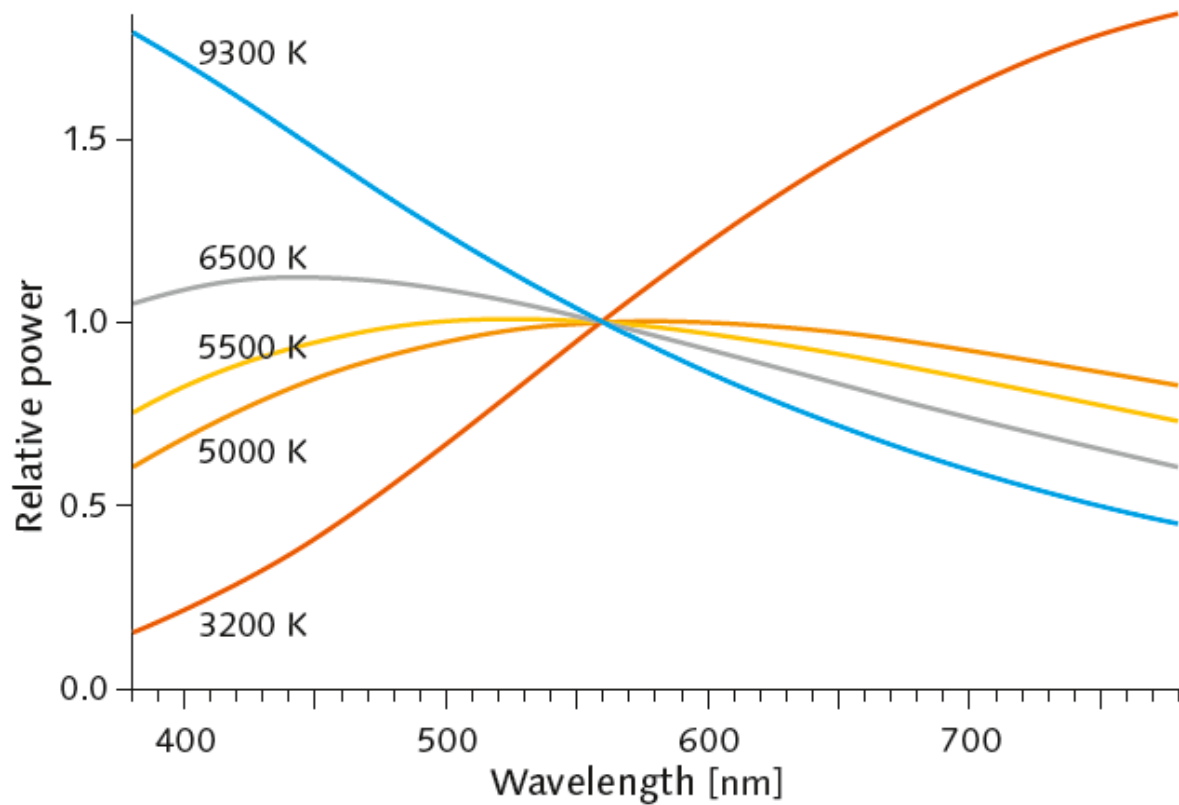


$$\lambda_B = 435.8\text{nm} \quad \lambda_G = 546.1\text{nm} \quad \lambda_R = 700\text{nm}$$



Imaginarne primarne komponente pomoću kojih je moguće generisati bilo koju boju. Izbegnute su negativne vrednosti. Sami primari ne mogu biti realizovani.

Spektar zračenja crnog tela



Hromatski dijagram

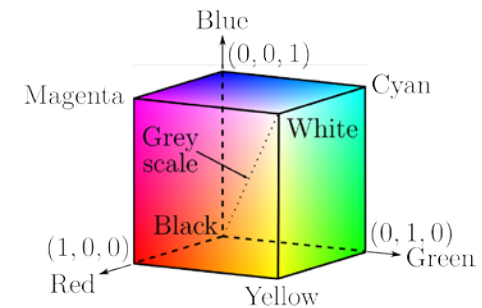
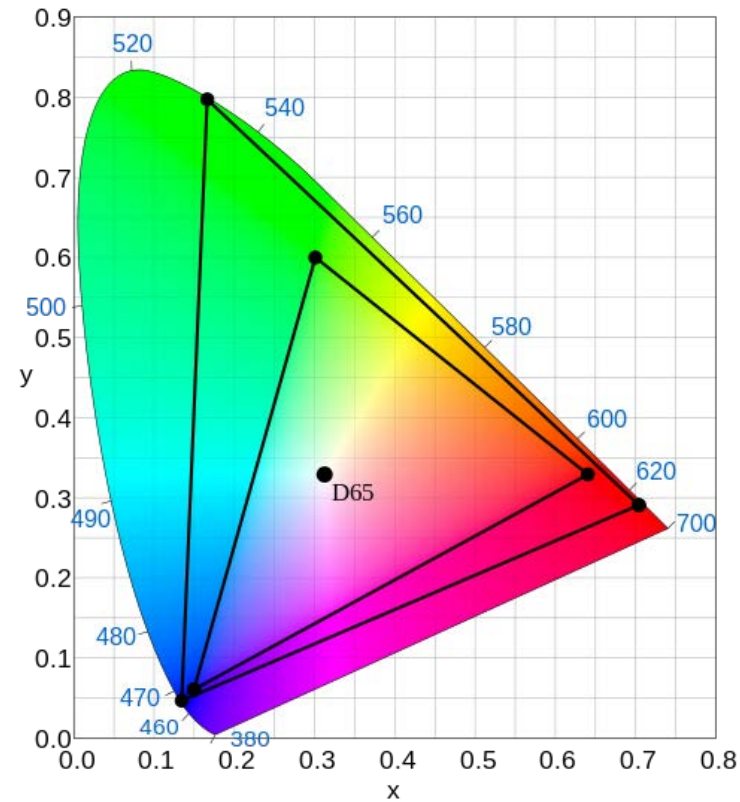
$$X = k \int_{\lambda} \Phi(\lambda) \bar{x}(\lambda) d\lambda$$

$$Y = k \int_{\lambda} \Phi(\lambda) \bar{y}(\lambda) d\lambda$$

$$Z = k \int_{\lambda} \Phi(\lambda) \bar{z}(\lambda) d\lambda$$

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$



Analiza spektra zračenja crnog tela

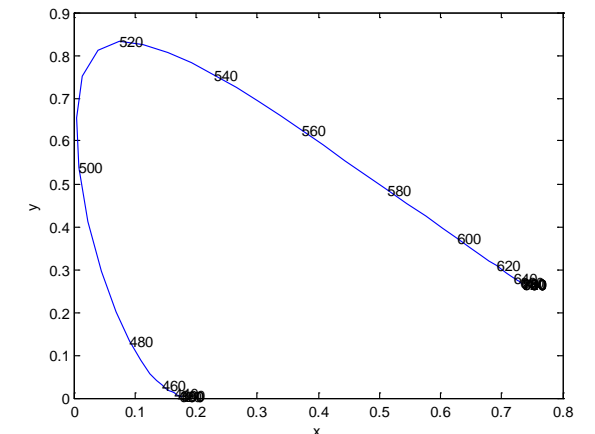
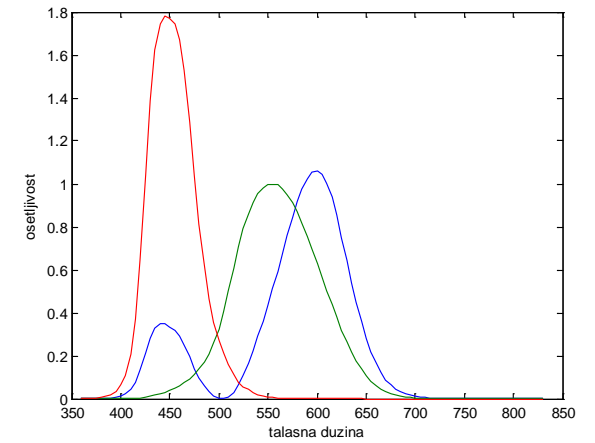
U tabeli StdObsFuncs.xls se nalaze odbirci funkcija standardnog posmatrača. Učitati ove vrednosti u Matlab i na jednom dijagramu iscrtati funkcije x_{bar} , z_{bar} i z_{bar} u zavisnosti od talasne dužine. Odrediti hromatske koordinate x i y i iscrtati hromatski dijagram (zavisnost y od x).

U tabeli blackbody.xls nalazi se skripta za računanje spektra zračenja crnog tela za različite temperature. Pomoću ove skripte izgenerisati spektar zračenja crnog tela na 6500K i učitati ove vrednosti u Matlab.

Odrediti hromatske koordinate spektra zračenja crnog tela na 6500K.

Analiza spektra zračenja crnog tela

```
wl = 360:5:830;  
xbar = []; %uneti podatke iz tabele  
ybar = []; %uneti podatke iz tabele  
zbar = []; %uneti podatke iz tabele  
  
figure; plot(wl, xbar, wl, ybar, wl, zbar);  
xlabel('talasna duzina'); ylabel('osetljivost');  
  
x = xbar./(xbar + ybar + zbar);  
y = ybar./(xbar + ybar + zbar);  
  
figure; plot(x, y);  
text(x(1:4:end), y(1:4:end), cellstr(num2str(wl(1:4:end)')));  
xlabel('x'); ylabel('y');
```



Analiza spektra zračenja crnog tela

```
wl_bb = 300:10:800;
bb_spec = []; %uneti podatke iz tabele

figure; plot(wl_bb, bb_spec); ylim([0 1]);
xlabel('talasna duzina'); ylabel('normalizovan intenzitet');

%zajednicki podaci samo u opsegu 360nm-800nm sa korakom 10nm
X_bb = sum(bb_spec(find(wl_bb==360):end).*xbar(1:2:find(wl==800)));
Y_bb = sum(bb_spec(find(wl_bb==360):end).*ybar(1:2:find(wl==800)));
Z_bb = sum(bb_spec(find(wl_bb==360):end).*zbar(1:2:find(wl==800)));

x_bb = X_bb./(X_bb + Y_bb + Z_bb)
y_bb = Y_bb./(X_bb + Y_bb + Z_bb)
```

Važni kolor sistemi (RGB, YCbCr, HSV, Lab)

Postoji jako veliki broj kolor sistema. Svaki je kreiran za određenu oblast primene i poseduje odgovarajuće prednosti i mane. Neophodno je poznavati karakteristike kolor sistema kako bi se odabrao odgovarajući za željenu primenu. Najbitniji kolor sistemi su:

1) Zavisni od uređaja

- **RGB** – Koristi se za prikaz slike na monitor i akviziciju slike sa senzora.
- **YCbCr** – Koristi se u digitalnoj televiziji, sistemima za obradu unutar kamere i pri kompresiji. Omogućava efikasniji prenos i skladištenje.
- **HSV, HSL, HSI** – Koriste se u računarskoj grafici za intuitivniju kontrolu boje preko „perceptualnih“ parametara nijanse i zasićenosti boje.

2) Nezavisni od uređaja

- **XYZ** – Osnovni kolor sistem koji omogućava specificiranje bilo koje boje korišćenjem pozitivnih koordinata. Predstavlja osnovu svih kolor sistema nezavisnih od uređaja.
- **Lab** – Razdvaja informaciju o sjajnosti L od informacija o boji a i b. Sistem je uniforman (jednake promene vrednosti Lab odgovaraju jednakim perceptualnim razlikama). Koristi se za specificiranje boje nezavisno od uređaja i za merenje razlike između dve boje.
- **LCh** – Transformacijom a i b koordnata u cilindrični sistem omogućava se intuitivnija kontrola nad bojama specificiranjem parametara čistoće boje (C) i nijanse (h)

CIE-LAB kolor sistem

XYZ nije uniforman – jednake promene vrednosti proizvode nejednake perceptualne razlike u boji. Zbog toga nije baš dobar za kvantifikovanje razlike između dve boje.

CIE-LAB (Lab) je sistem koji je nezavistan od uređaja i ima dovoljno dobru uniformnost da se može koristiti za definisanje razlike između boja. Sastoji se iz 3 komponente:

L – sjajnost

a – razlika između crvene i zelene

b – razlika između žute i plave

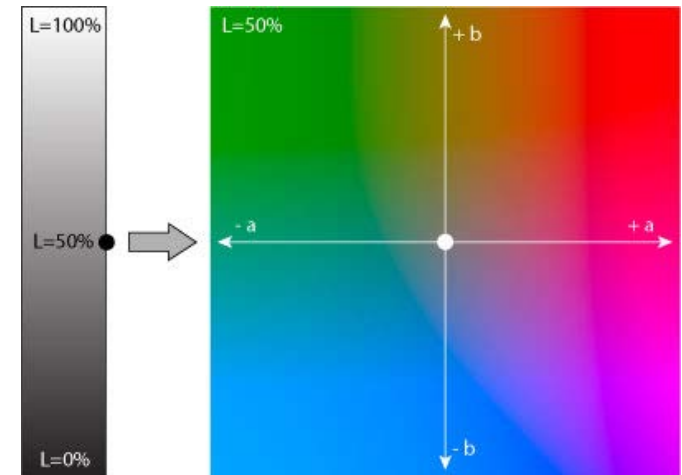
Uključuje nelinearnu karakteristiku ljudskog vizuelnog sistema.

$$L^* = 116 \cdot h\left(\frac{Y}{Y_w}\right) - 16$$

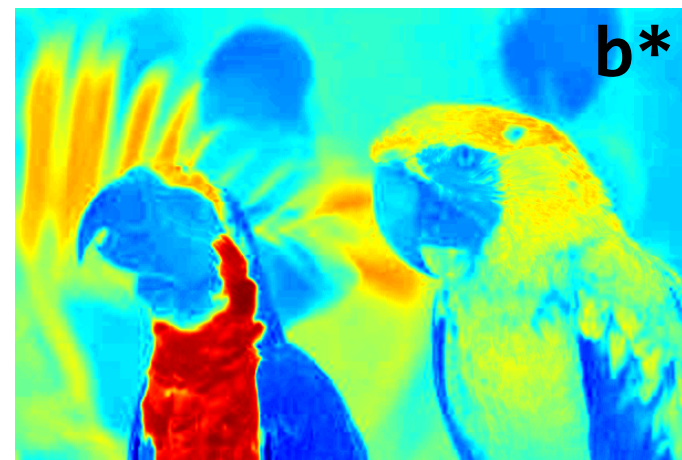
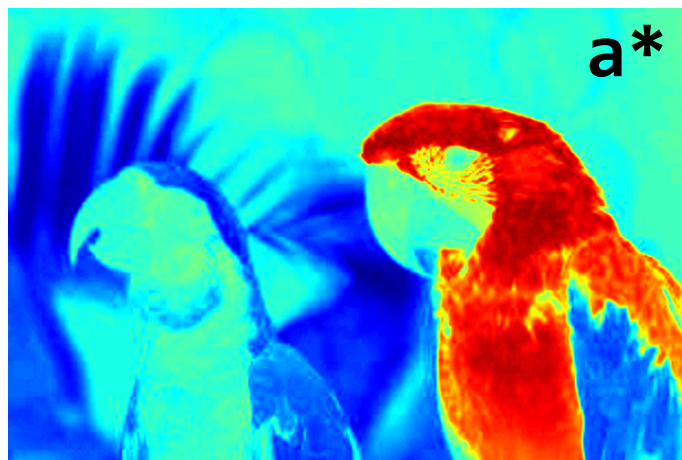
$$a^* = 500 \cdot \left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right) \right]$$

$$b^* = 200 \cdot \left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right) \right]$$

$$h(q) = \begin{cases} \sqrt[3]{q}, & q > 0.008856 \\ 7.787q + \frac{16}{116}, & q \leq 0.008856 \end{cases}$$



CIE-LAB primer



CIE-LCh kolor sistem

CIE-LAB je uniforman i može se koristiti za definisanje razlike između boja, međutim nije dovoljno intuitivan za specificiranje boje i manipulaciju.

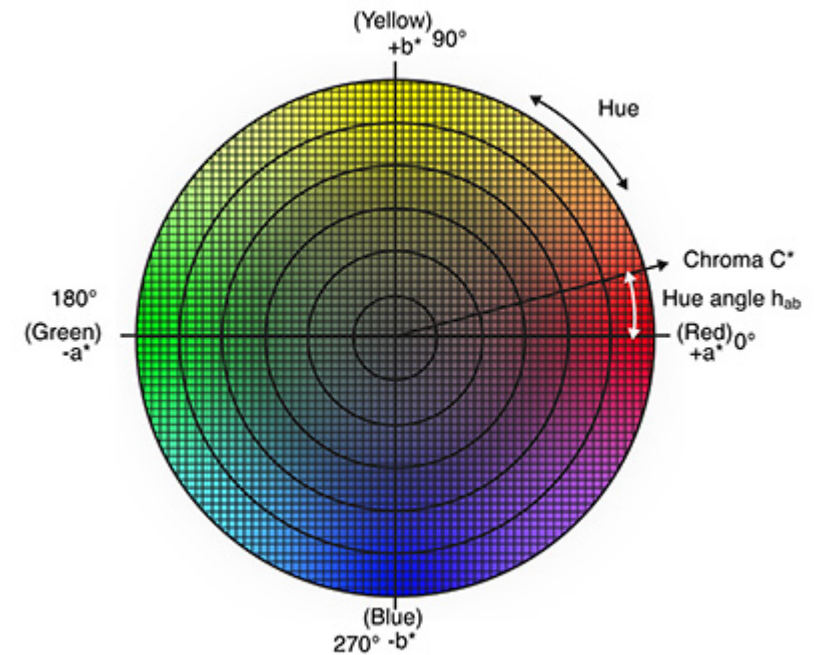
Specificiraju se polarne koordinate:

C – hroma (zasićenost, čistoća boje)
h – vrsta boje (nijansa)

Sjajnost je definisana komponentom L i ista je kao u originalnom Lab sistemu.

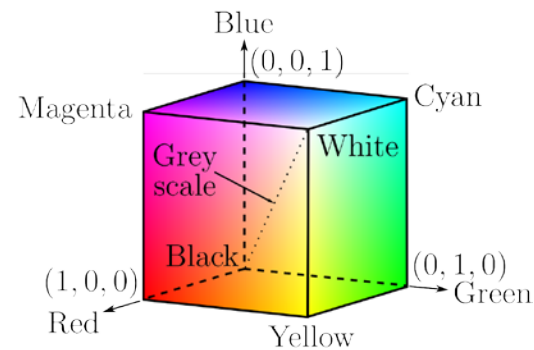
$$C^* = \sqrt{(a^*)^2 + (b^*)^2}$$

$$h_{ab} = \arctan\left(\frac{b^*}{a^*}\right)$$

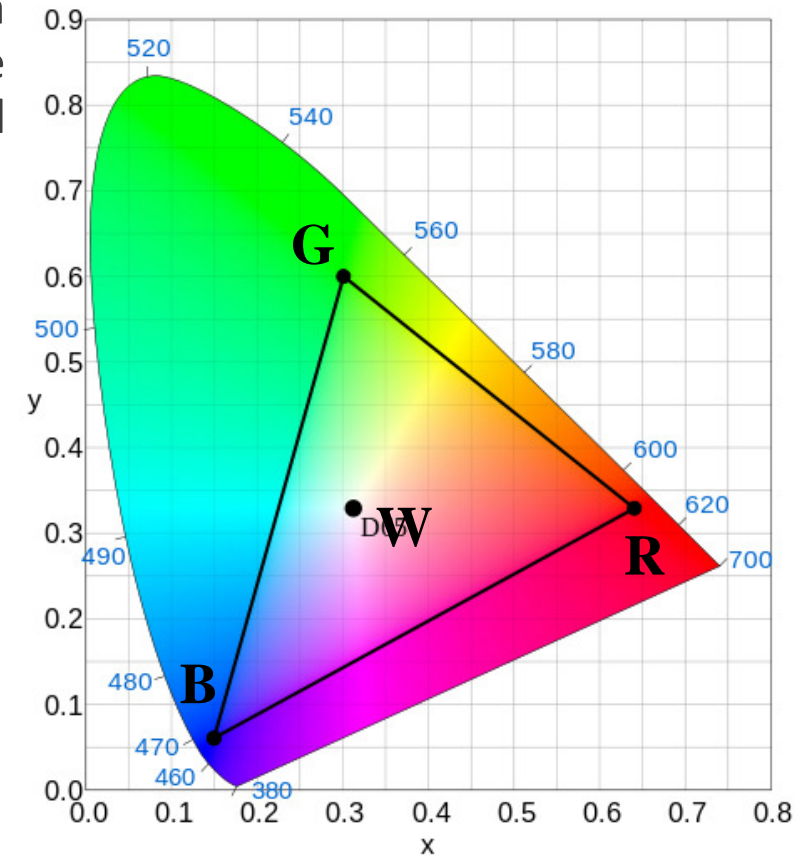


RGB kolor sistem

Svaka boja predstavljena trodimenzionalnim vektorom koji označava udeo svake od primarnih komponenti. Ukupan opseg boja se može predstaviti RGB kockom. Kako priroda primarnih komponenti zavisi od uređaja, tako je i predstava u RGB formatu vezana za uređaj.



$$\mathbf{C} = R_{LIN} \cdot \mathbf{R} + G_{LIN} \cdot \mathbf{G} + B_{LIN} \cdot \mathbf{B}$$

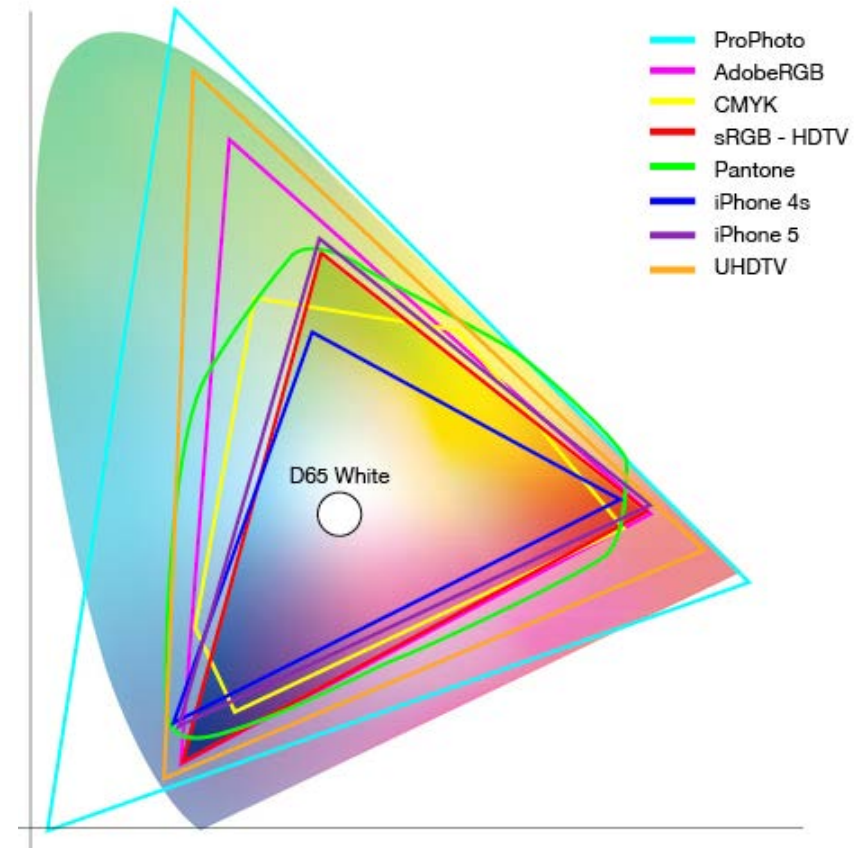


Zavisnost kolor sistema od uređaja

Različiti uređaji mogu imati različite primarne komponente. Kako se boja specificira kao linearna kombinacija ovih komponenti onda i ta informacija o boji zavisi od uređaja. Dve boje koje imaju istu vrednost (koordinate) mogu izgledati sasvim različito na različitim uređajima. Međutim ako je boja specificirana u nekom kolor sistemu koji ne zavisi od uređaja onda se ta informacija može jednostavno preneti na različite uređaje i obezbediti da prikaz bude identičan.

Kolor sistemi nezavisni od uređaja koji se najčešće koriste su: XYZ, Lab, LCh, Luv

XYZ predstavlja osnovni koordinatni sistem za predstavu boja nezavisno od uređaja i iz njega se izvode ostali kolor sistemi.



Transformacija iz RGB u XYZ

Potrebne informacije:

- 1) Hromatske koordinate primarnih komponenti
- 2) Hromatske koordinate bele (izvora svetla)
- 3) Nelinearna transformacija tf primenjena na RGB komponente

$$\mathbf{R} = X_R \cdot \mathbf{X} + Y_R \cdot \mathbf{Y} + Z_R \cdot \mathbf{Z}$$

$$\mathbf{G} = X_G \cdot \mathbf{X} + Y_G \cdot \mathbf{Y} + Z_G \cdot \mathbf{Z}$$

$$\mathbf{B} = X_B \cdot \mathbf{X} + Y_B \cdot \mathbf{Y} + Z_B \cdot \mathbf{Z}$$

$$\mathbf{1} = X_W \cdot \mathbf{X} + Y_W \cdot \mathbf{Y} + Z_W \cdot \mathbf{Z}$$

$$tf(L) = \begin{cases} (1+f) \cdot L^y - f & t < L \leq 1 \\ s \cdot L & 0 \leq L \leq t \end{cases}$$

$$\begin{bmatrix} R_{LIN} & G_{LIN} & B_{LIN} \end{bmatrix} = \begin{bmatrix} tf^{-1}(R) & tf^{-1}(G) & tf^{-1}(B) \end{bmatrix}$$

$$\mathbf{C} = R_{LIN} \cdot \mathbf{R} + G_{LIN} \cdot \mathbf{G} + B_{LIN} \cdot \mathbf{B} = \begin{bmatrix} \mathbf{R} & \mathbf{G} & \mathbf{B} \end{bmatrix} \times \begin{bmatrix} R_{LIN} \\ G_{LIN} \\ B_{LIN} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} & \mathbf{Z} \end{bmatrix} \times \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \times \begin{bmatrix} R_{LIN} \\ G_{LIN} \\ B_{LIN} \end{bmatrix}$$

$$\begin{bmatrix} S_R \\ S_G \\ S_B \end{bmatrix} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix}^{-1} \times \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} \Rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} S_R X_R & S_G X_G & S_B X_B \\ S_R Y_R & S_G Y_G & S_B Y_B \\ S_R Z_R & S_G Z_G & S_B Z_B \end{bmatrix} \times \begin{bmatrix} R_{LIN} \\ G_{LIN} \\ B_{LIN} \end{bmatrix}$$

Primer: transformacija sRGB u XYZ

Koordinate primarnih komponenti i bele boje iz specifikacije sRGB kolor sistema

	x	y
R	0.64	0.33
G	0.3	0.6
B	0.15	0.06
W(D65)	0.3127	0.3290

Nelinearna transfer funkcija iz specifikacije sRGB kolor sistema

$$t = 0.003138 \quad f = 0.055$$
$$\gamma = 1/2.4 \quad s = 12.92$$

Kako se vrednosti transfer matrice skaliraju tako da se RGB vrednost (1,1,1) slika u referentnu belu boju, pri izvođenju se mogu koristiti koordinate xyz umesto koordinata XYZ pošto su proporcionalne ($XYZ = xyz*(Y/y)$).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4123908 & 0.3578434 & 0.1804808 \\ 0.2126390 & 0.7151687 & 0.0721923 \\ 0.0193308 & 0.1191948 & 0.9505322 \end{bmatrix} \times \begin{bmatrix} tf^{-1}(R) \\ tf^{-1}(G) \\ tf^{-1}(B) \end{bmatrix}$$

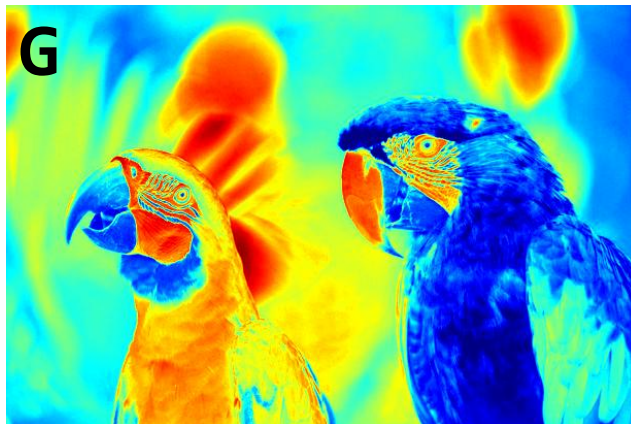
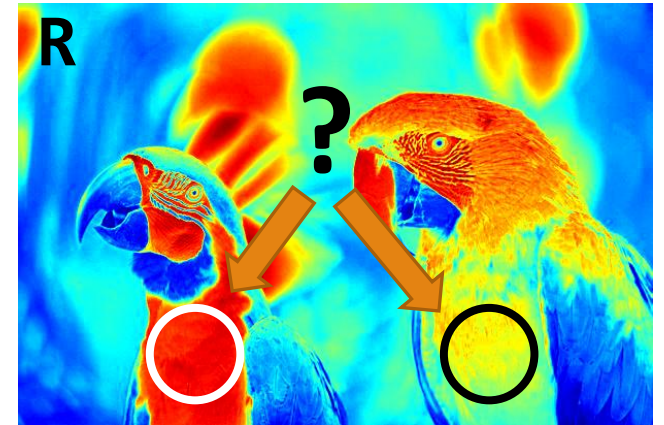
Primer: RGB kolor sistem

```
I = imread('parrots.jpg');
```

```
figure;  
imshow(I(:,:,1),[]);  
colormap('jet');
```

```
figure;  
imshow(I(:,:,2),[]);  
colormap('jet');
```

```
figure;  
imshow(I(:,:,3),[]);  
colormap('jet');
```



YCbCr kolor sistem

Koristi se u sistemima digitalne televizije. Razdvaja informaciju o osvetljenosti od informacije o boji uzimajući u obzir karakteristike ljudskog vizuelnog sistema.

REC. 601 – SDTV

$K_B=0.114$ $K_R=0.299$

$$Y = K_R \cdot R + (1 - K_R - K_B) \cdot G + K_B \cdot B$$

REC. 709 – HDTV

$K_B=0.0722$ $K_R=0.2126$

$$C_B = \frac{1}{2} \frac{B - Y}{1 - K_B} \quad C_R = \frac{1}{2} \frac{R - Y}{1 - K_R}$$

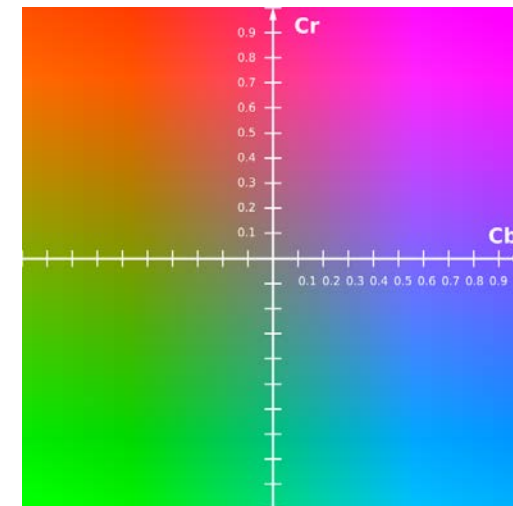
REC. 2020 – UHD TV

$K_B=0.0593$ $K_R=0.2627$

U digitalnoj televiziji se ne koristi pun opseg 0-255 već se ostavlja prostor na oba kraja opsega kako bi se povećala preciznost pri procesiranju:

Y 16-235 (mapira se na opseg 0 - 219)

Cb, Cr 16-240 (mapira se na opseg -112 - 112)



Prilikom obrade slike na računaru, preciznost se može povećati interno tokom procesiranja pa se koristi pun opseg 0 – 255 za lumu odnosno -128 – 127 za hromu

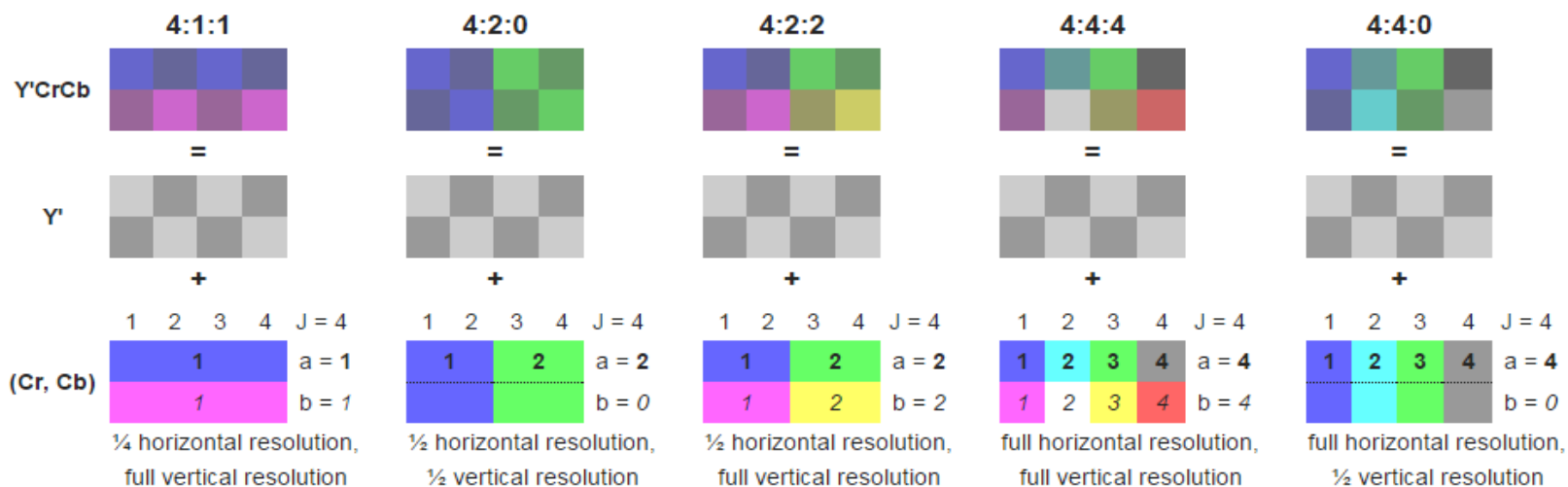
YCbCr – decimacija hrome

Informacija o osvetljenosti je sadržana u komponenti luma Y. Kako je ljudski vizuelni sistem osjetljiviji na promenu osvetljaja nego boje, komponente hrome Cb i Cr se mogu decimirati kako bi se uštedelo na protoku i prostoru potrebnim za njihovo skladištenje. Najčešće korišćeni formati su:

YCbCr444 – Y, Cb, Cr su punih dimenzija (nema decimacije)

YCbCr422 – Cb i Cr decimirani 2x po horizontali

YCbCr420 – Cb i Cr decimirani 2x po horizontali i vertikali



Ove funkcije ne vraćaju pun opseg 0 – 255!!!

Primer: YCbCr kolor sistem

`rgb2ycbcr(rgb_slika)`
`ycbcr2rgb(ycbcr_slika)`

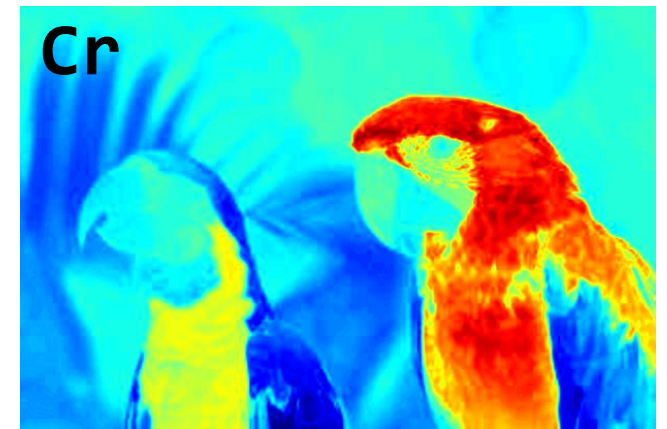
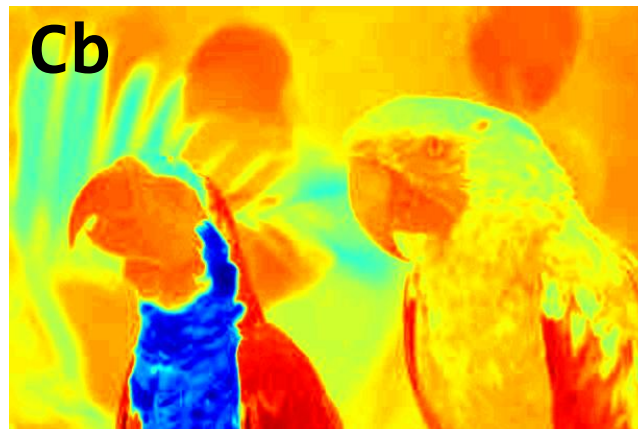
```
I = imread('parrots.jpg');
```

```
Iycbcr = rgb2ycbcr(I);
```

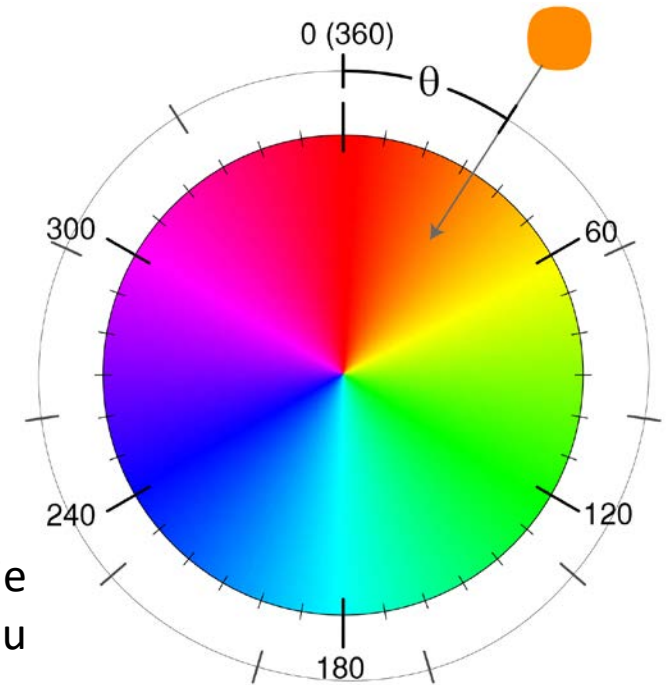
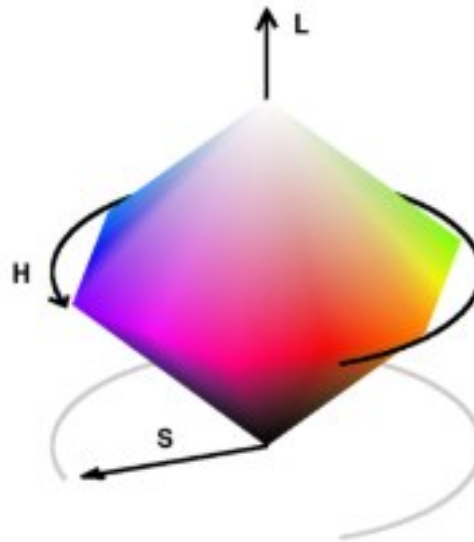
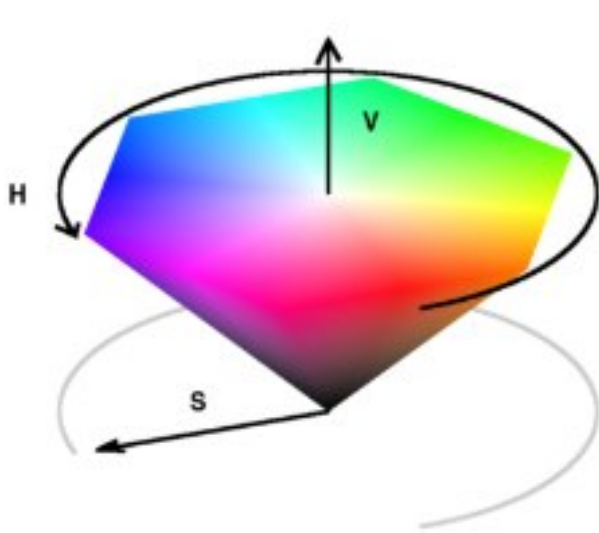
```
figure;  
imshow(Iycbcr(:,:,1),[]);  
colormap('gray');
```

```
figure;  
imshow(Iycbcr(:,:,2),[]);  
colormap('jet');
```

```
figure;  
imshow(Iycbcr(:,:,3),[]);  
colormap('jet');
```



HSV, HSL, HSI



Iako YCbCr kolor sistem uspešno razdvaja informaciju o osvetljaju od informacije o boji, hrominente komponente Cb i Cr nisu dovoljno intuitivne za manipulaciju potrebnu u računarskoj grafici.

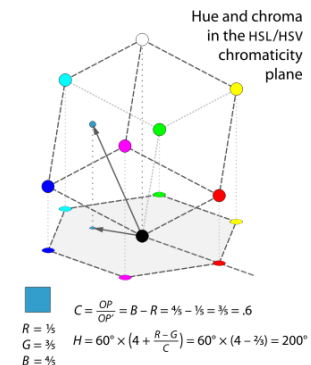
Uvode se novi sistemi koji informaciju o boji predstavljaju komponentama:
Nijansa (H) – vrsta boje i Saturacija (S) – čistoća boje

Konverzija iz RGB u HSV, HSL, HSI

$$M = \max(R, G, B)$$

$$m = \min(R, G, B)$$

$$C = M - m$$



$$H' = \begin{cases} 0, C = 0 \\ \frac{G - B}{C}, M = R \\ \frac{B - R}{C} + 2, M = G \\ \frac{R - G}{C} + 4, M = B \end{cases}$$

$$H = 60^\circ \cdot H'$$

$$V = M \quad S = \frac{C}{V}$$

$$L = \frac{M + m}{2} \quad S = \frac{C}{1 - |2L - 1|}$$

$$I = \frac{R + G + B}{3} \quad S = \begin{cases} 0, C = 0 \\ 1 - \frac{m}{I}, C \neq 0 \end{cases}$$

Transformacija iz HSV u RGB

$$H' = \frac{H}{60^\circ}$$

$$k = \lfloor H' \rfloor$$

$$p = H' - k$$

$$t = 1 - S$$

$$n = 1 - S \cdot p$$

$$r = 1 - S \cdot (1 - p)$$

$$[R' \quad G' \quad B']$$

$$[1 \quad r \quad t], k = 0$$

$$[n \quad 1 \quad t], k = 1$$

$$[t \quad 1 \quad r], k = 2$$

$$[t \quad n \quad 1], k = 3$$

$$[r \quad t \quad 1], k = 4$$

$$[1 \quad t \quad n], k = 5$$

$$[R \quad G \quad B] = V \cdot [R' \quad G' \quad B']$$

Primer: HSV kolor sistem

`rgb2hsv(rgb_slika)`
`hsv2rgb(hsv_slika)`

```
I = imread('parrots.jpg');
```

```
Ihsv = rgb2hsv(I);
```

```
hue = Ihsv(:,:,1);
```

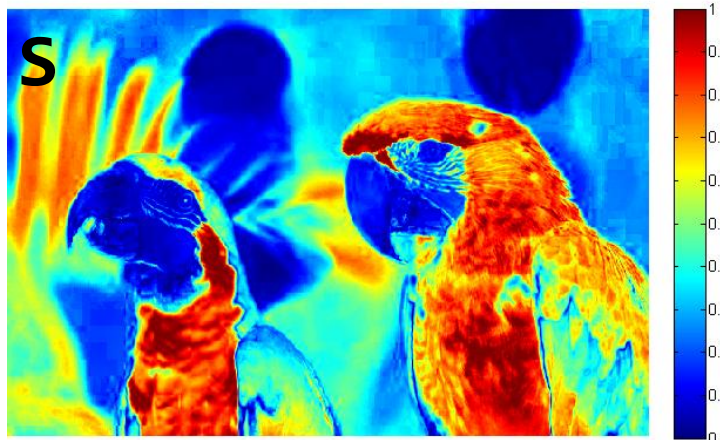
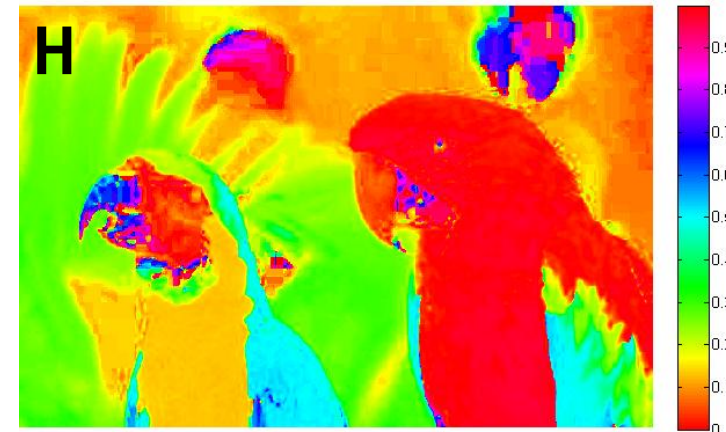
```
sat = Ihsv(:,:,2);
```

```
val = Ihsv(:,:,3);
```

```
figure; imshow(hue, []);  
colormap('hsv');
```

```
figure; imshow(sat, []);  
colormap('jet');
```

```
figure; imshow(val, []);  
colormap('gray');
```



Primer: Manipulacija u HSV kolor sistemu

```
hue_shift1 = hue - 0.2;
hue_shift1(hue_shift1<0)=hue_shift1(hue_shift1<0)+1;
hue_shift2 = hue + 0.2;
hue_shift2(hue_shift2>1)=hue_shift2(hue_shift2>1)-1;

Ihsv(:,:,1) = hue_shift1; Ihsv(:,:,2) = sat; Ihsv(:,:,3) = val;
figure; imshow(hsv2rgb(Ihsv));
Ihsv(:,:,1) = hue_shift2; Ihsv(:,:,2) = sat; Ihsv(:,:,3) = val;
figure; imshow(hsv2rgb(Ihsv));

Ihsv(:,:,1) = hue; Ihsv(:,:,2) = 0.8*sat; Ihsv(:,:,3) = val;
figure; imshow(hsv2rgb(Ihsv));
Ihsv(:,:,1) = hue; Ihsv(:,:,2) = 1.2*sat; Ihsv(:,:,3) = val;
figure; imshow(hsv2rgb(Ihsv));

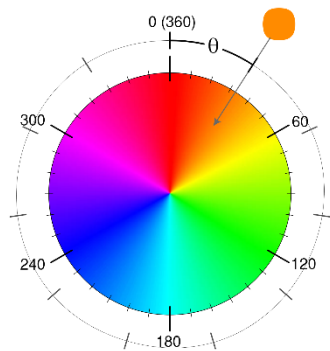
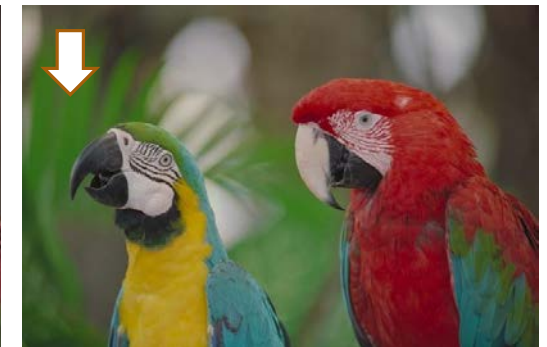
Ihsv(:,:,1) = hue; Ihsv(:,:,2) = sat; Ihsv(:,:,3) = 0.8*val;
figure; imshow(hsv2rgb(Ihsv));
Ihsv(:,:,1) = hue; Ihsv(:,:,2) = sat; Ihsv(:,:,3) = 1.2*val;
figure; imshow(hsv2rgb(Ihsv));
```

Primer: Manipulacija u HSV kolor sistemu

H

S

V



Poređenje osvetljaja kod različitih kolor sistema





keep staring at the black dot.