

# Pokročilé spracovanie obrazu - Farebné priestory

Ing. Viktor Kocur  
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

9.10.2019

# HSV

## HSV

HSV je farebný model orientovaný na intuitívne využitie (tzv. užívateľský model). Hue predstavuje pozíciu na farebnom kruhu (odtieň), saturation je sýtosť a value je jas.

## rgb2hsv

`rgb2hsv(I)` - vráti obraz zapísaný v podľa HSV modelu, má rovnaké rozmery ako `I`, tj. tiež má tri kanály.

## hsv2rgb

`hsv2rgb(I_hsv)` - urobí to isté ale naopak.

# HSV - Úloha

## HSV

Použite GUI z minula. Ale namiesto R, G a B sliderov to budú H, S, V slide. Telo funkcie bude skoro rovnaké obrázok, ale musíte najprv premeniť na HSV, upraviť a potom konvertovať na RGB.

## GUI

Ak nemáte uložené GUI, tak si môžete stiahnuť súbory z minulého cvika a upraviť `gui_sliders.m`.

# CIE Lab

## CIE Lab

CIE  $L^*$   $a^*$   $b^*$  je model s tromi zložkami.  $L$  predstavuje jas (luminance),  $a$  je pozícia na ose zelená-červená a  $b$  je pozícia na ose modrá-žltá.

## rgb2lab

`rgb2lab(I)` - vráti obraz zapísaný v podľa CIE Lab modelu, má rovnaké rozmery ako  $I$ , t.j. tiež má tri kanály.

## lab2rgb

`lab2rgb(I_lab)` - urobí to isté ale naopak.

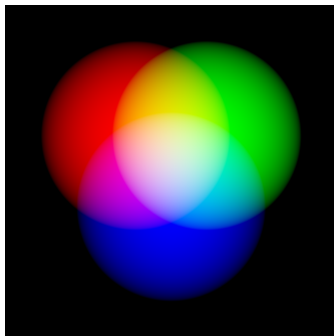
## Úloha

Upravte gui, tak aby slidre reprezentovali zložky  $L$ ,  $a$ ,  $b$ .

# RGB

## RGB

RGB je aditívny model, tj. na čierne pozadie pridávame farby. Ak prídáme všetky tri farby, tak vznikne biela farba.



# CMY

## CMY

CMY je substraktívny model, tj. na biele pozadie pridávame farby. Ak pridáme všetky tri farby, tak vznikne čierna farba.

## CMYK

CMYK má navyše čiernu farbu. To je vhodné pre tlačiarne.



# CMY vs. RGB

## CMY

Pre farby podľa CMY a RGB platí prechod  $C = 255 - R$ ,  
 $M = 255 - G$  a  $Y = 255 - B$

# CMY vs. RGB

## CMY

Pre farby podľa CMY a RGB platí prechod  $C = 255 - R$ ,  
 $M = 255 - G$  a  $Y = 255 - B$

## RGB

Naopak platí  $R = 255 - M - Y$ ,  $G = 255 - C - Y$ ,  $B = 255 - C - M$



# Euklidovská vzdialenosť

## Úloha

Napíšte skript ktorý zobrazí obrázok farby.png. Pomocou funkcie ginput vyberte tri body a zistite aká je vzdialenosť medzi prvým a zvyšnými dvoma pre rôzne farebné modely. Porovnajte, či podobné farby sú pri sebe bližšie ako zdanlivo rôzne farby.

## Euklidovská vzdialenosť

$$\rho_e(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

## ginput

[x ,y] = ginput(n) - vráti vektory x-ových a y-ových súradníc bodov na ktoré užívateľ po zadaní príkazu klikne vo figure

# Pseudofarby

## Pseudofarby

Pseudofarby využijeme na zafarbenie šedotónového obrázka, tak aby sa zvýraznili niektoré detaily.

## colormap

`colormap(mapa)` - zmení farby ktorými sa vykresluje obrázok na tie ktoré sú v mape. Mapa môže byť buď prednastavená (jet, hsv, winter, copper, bones, gray...), alebo matica  $n \times 3$ , kde na každom riadku je RGB trojica, ktorá predstavuje farbu.

## Kód

```
BW = imread('medical.pgm');  
imagesc(BW);  
colormap(hsv);
```

# Indexované obrázky

## Indexovaný obrázok

Indexovaný obrázok je taký, v ktorom každý pixel nieje reprezentovaný vlastnou RGB farbou, ale indexom. Tento index ukazuje ktorá farba sa v danom pixely nachádza. Na to však aby bolo jasné ktorý index korešponduje k akej farbe je nutné mať mapu farieb, tj. maticu  $n \times 3$ , kde na každom riadku je RGB trojica a  $n$  je počet indexovaných farieb.

## rgb2ind

`[X, map] = rgb2ind(I,n)` - vráti indexovaný obraz `X` (podobný label matici) s `n` farbami a mapu  $n \times 3$  tj. zoznam trojíc farieb v poradí podľa ktorého sa indexuje.

`X = rgb2ind(I,map)` - vráti indexovaný obraz `X` pre danú mapu.

## imhist

`hist = imhist(X,map)` - vráti histogram, tj. vektor početností pre jednotlivé indexy. V prípade, že si výstup nikam neuložíme, tak sa histogram nakreslí.

## Kód

```
[X, map] = rgb2ind(I,30);  
imagesc(X);  
colormap(map);  
figure;  
imhist(X,map);
```

# Úloha

## Úloha

Obrázok zátišia prekonvertujte na indexovaný pre 20 farieb a zistite, ktorá z týchto farieb je dominantná (nájdite jej RGB trojicu). K tomu aj zistite koľko percent pixelov má práve túto farbu.

## Hint

Pomôže vám príkaz `max`, pozrite si ho v helpe.

## Pozor!

Pre prípad ak nepoužijete funkciu `imhist`, tak je dôležité si uvedomiť, že indexy v obrázku začínajú na 0, ale indexy v mape začínajú 1!