Képfeldolgozás a gyakorlatban

Munka színes képekkel

Szeghalmy Szilvia

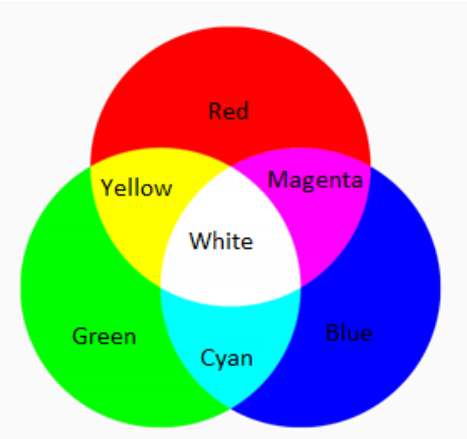
Néhány színmodell

- **RGB**
  - kijelzők, kamerák, tömörítettlen képek tárolása
- CMYK
  - nyomtatás
- HSI, **HSV**, HSL
  - képszerkesztő programok (pl. színkijelölés, átszínezés)
- YIQ, YCbCr, YUV:
  - tömörítés
- **CIE LAB**
  - eszközfüggetlen standardok
  - színkülönbség mérés
- Transzformációk: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html

RGB modell

- Additív színmodell: piros, zöld, kék keverése

https://isle.hanover.edu/Ch06Color/Ch06ColorMixer.html

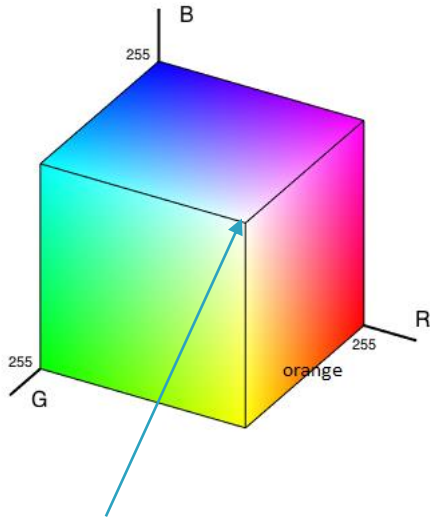


- RGB hullámhossz:700nm, 546nm, 435nm
- Elektronikai eszközök alkalmazzák: kijelzők, kamerák
- 16 millió szín kódolható
- Szürke árnyalat: R=G=B

- OPENCV: BGR
  - Mátrix típus: CV_8UC3
  - Pont típus: img.at<Vec3b>(i, j)
  - Tartomány: R, G, B ∈ [0, 255]

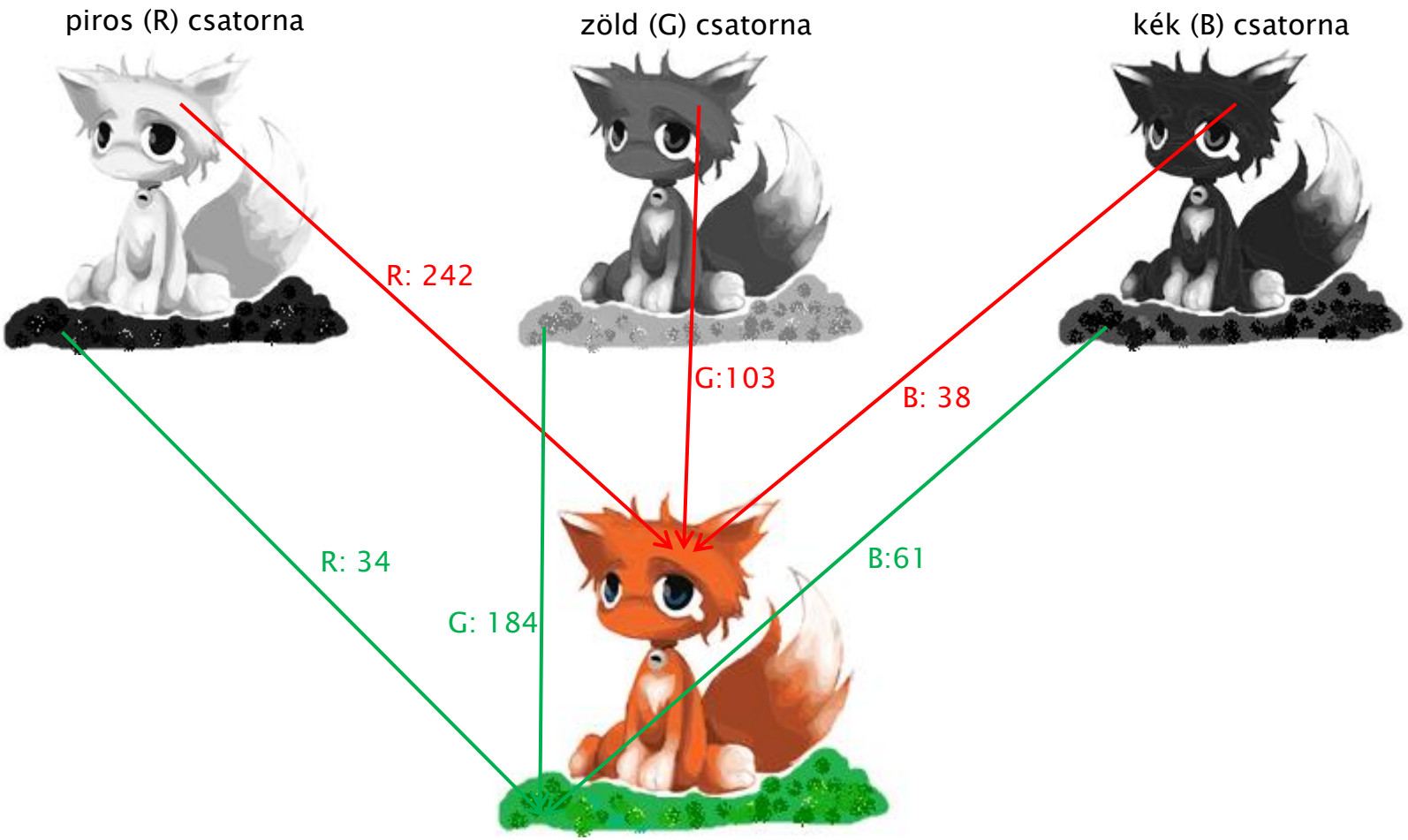
  - Tartomány váltás: r, g, b ∈ [0, 1]

    rgb.convertTo(norm_rgb, **CV_32FC3**, 1/255.0);



Tiszta fehér: (255, 255, 255)

RGB modell



fox image: https://commons.wikimedia.org/wiki/File:Firefox-Vectorized.svg

Feladat: nem fehér pontok másolása

- Töltse le Kuramát és az egyik háttérképet az e-learning rendszerből (gyak1).
- Olvassa be mindkét képet színesben.
- Készítsen egy függvényt, ami a tiszta fehértől eltérő pontokat átmásolja egy másik képre:

```
void eloterMasolo( const cv::Mat fg, cv::Mat& bg );
```

- Tesztelje.
- Emlékeztető
  - Kép beolvasása:

```
cv::imread( elérési_út, ImreadModes::IMREAD_COLOR )
```

  - Kép létrehozása:

```
img.create( méret, típus );
```

  - Kép másolása (deep copy):

```
img2 = img.clone();
```

  - Képpont elérése

```
img.at<cv::Vec3b>(sor, oszlop)[csatorna]
csatorna: 0 - kék, 1 - zöld, 2 - piros
```

Gyakorló feladat otthonra

- A háttér helyett Kurama képét méretezze át. A magasság és a szélesség is az eredeti negyede legyen.
- Számolja ki, hogy vízszintes irányban mennyi eltolás kell ahhoz, hogy Kurama a kép közepére kerüljön. A függőleges eltolást próba-hiba módon állítsa be úgy, hogy a kis róka "leüljön" a földre.
- Járja be a kicsinyített képet és másolja át a nem fehér pontokat a háttérképre.
- Tesztelje a munkáját, szükség esetén igazítson a kis kép pozícióján.

- *Tipp: csak akkor másoljon át egy pontot, ha a koordinátái a háttérkép koordinátatartományán belülre esnek. Ha így jár el, akkor sem akad ki a program, ha rosszul határozta meg elsőre az eltolást. Könnyebb lesz javítani.*

Feladat: narancsmásoló (RGB)

- Töltse le az alábbi képet:

  https://en.wikipedia.org/wiki/File:Indian_hybrid_Orange.jpg

- Készítsen egy függvényt, ami a narancsokat egy másik képre másolja.

```
void narancsmasolo( const cv::Mat src, cv::Mat& dest );
```

- Emlékeztető
  - Kép beolvasása:

```
cv::imread( elérési_út, ImreadModes::IMREAD_COLOR )
```

  - Kép létrehozása:

```
img.create( méret, típus );
```

  - Kép másolása (deep copy):

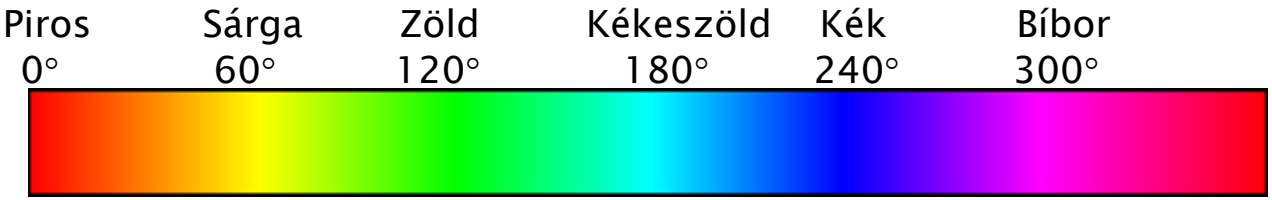
```
img2 = img.clone();
```

  - Képpont elérése

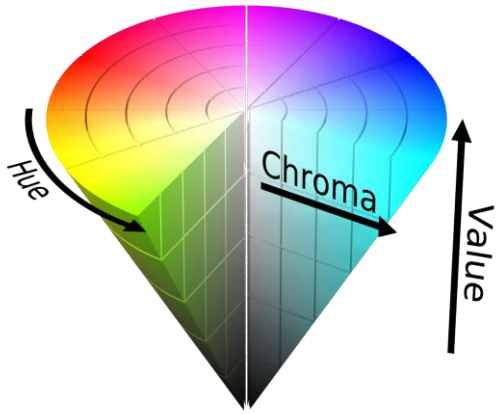
```
img.at<cv::Vec3b>(sor, oszlop)[csatorna]
csatorna: 0 - kék, 1 - zöld, 2 - piros
```

HSV

- Hue (színezet)



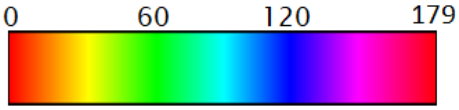
- Saturation/Chroma (telítettség)
  - Alap- és kiegészítőszínek telítettsége maximális
  - Szürke telítettsége nulla
- Value (max(R, G, B))
- 3 millió szín kódolható



- OpenCV
  - Mátrix típus pl.: CV_8UC3
  - Pont típus: img.at<Vec3b>(i, j)
  - Tartomány: H ∈ [0, 179], S, V ∈ [0, 255]
  - Áttérés BGR-ből HSV-be:

```
cvtColor( InputArray src, OutputArray dest,
          ColorConversionCodes::COLOR_BGR2HSV);
```

OpenCV Hue range: (CV_8U):



Feladat: Narancsmásoló (HSV)

- Töltse le az alábbi képet:

  https://en.wikipedia.org/wiki/File:Indian_hybrid_Orange.jpg

- Készítsen egy függvényt, ami a narancsokat egy másik képre másolja.

```
void narancsmasoloHSV( const cv::Mat src, cv::Mat& dest );
```

- Emlékeztető
  - Kép beolvasása:

```
imread( elérési_út, ImreadModes::IMREAD_COLOR )
```

  - Kép létrehozása:

```
img.create( méret, típus );
```

  - Áttérés HSV színtérbe:

```
cvtColor( InputArray src, OutputArray dest,
                ColorConversionCodes::COLOR_BGR2HSV);
```

  - Kép másolása (deep copy):

```
img2 = img.clone();
```

  - Képpont elérése

```
img.at<cv::Vec3b>(sor, oszlop)[csatorna]
                  csatorna: 0 - kék, 1 - zöld, 2 - piros
```

Feladat: Narancsmásoló (HSV)

- A feladat ugyanaz. Másolja át a narancsokat egy másik képre.

  https://en.wikipedia.org/wiki/File:Indian_hybrid_Orange.jpg

- Másolja át a narancsokat egy másik képre, beépített függvények használatával.
  - *inRange(img, lowerBound, upperBound, mask)* :

$$\text{mask}(i,j) = \begin{cases} 255, & if\ lowerBound \leq src(i,j) \leq upperBound \\ 0, & otherwise \end{cases}$$

  - *src.copyTo(dest [,mask])*:

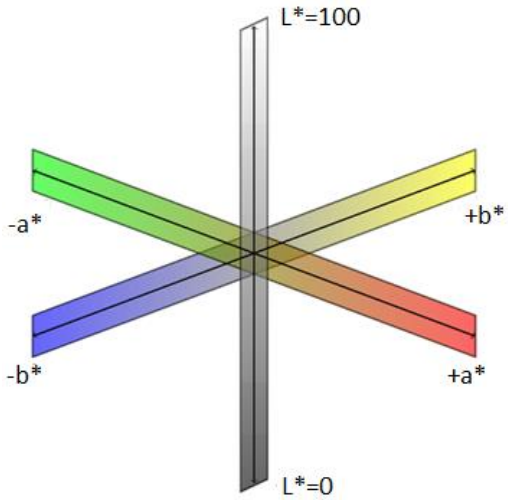
$$\text{dest}(\text{i},\text{j}) = \begin{cases} \text{src}(\text{i},\text{j}), & \text{if mask}(\text{i},\text{j}) > 0 \\ \text{dest}(\text{i},\text{j}), & \text{otherwise} \end{cases}$$

```
inRange(InputArray src, Scalar lowerBound, Scalar upperBound,
                                                    OutputArray dest);
img.copyTo(InputOutputArray dest, InputArray mask);
```

CIE Lab

- Szétválik a színezet (a*,b*) és a világosság (L*)
- Uniform színtér:

  Euklideszi távolság használható színkülönbség mérésre
- Tartományok:
  - L*: [0, 100]
  - a*: zöld-vörös tengely: [-100, 100]
  - b*: kék-sárga tengely: [-100, 100]



- Teljes színkülönbség:

$$\Delta E^* = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}$$

- Színezet változás:

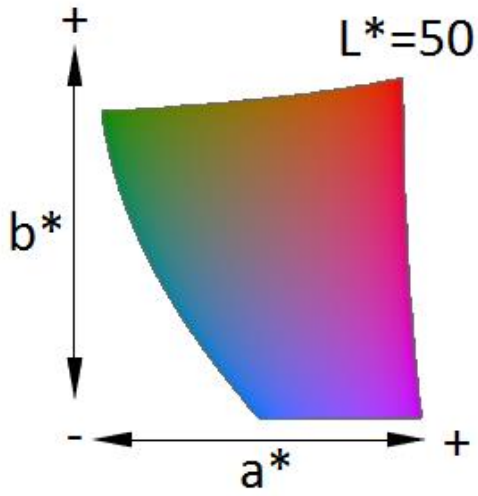
$$\Delta H^* = \sqrt{\Delta E^{*2} - \Delta L^{*2} - \Delta C^{*2}}$$

- Telítettség (chroma):

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

- Színezeti szög számítása:

$$h_{ab}{}^{\circ} = arctg\,\frac{b^*}{a^*}$$



Feladat: padlólapok etalontól való eltérése I.

- A cél egy etalon kép és más, azonos méretű képek teljes, átlagos színkülönbségének meghatározása. A fa erezete miatt viszonylag nagy különbségek várhatóak.

- Töltse le a padlolapok.zip fájlt és bontsd ki.

- Írja meg a képek beolvasását végző részt:

```
int main(){

    Olvassa fel az etalon képet.
    Olvassa fel egy ciklussal a többi képet egyesével.
            "padlolapok/fa_1.png", … "padlolapok/fa_8.png"
    Ellenőrzésként jelenítse meg a képeket.
    (A moveWindows(ablaknev, x, y)-nal elrendezheti az ablakokat,
       ha akarja.)
    …
}
```

Feladat: padlólapok etalontól való eltérése II.

- Írja meg a színtérváltó részt:

```
void convert(const Mat img, Mat& lab){
    Konvertálja az img képet valóssá, és a [0, 255] tartományról
      válts át a [0, 1]-re).
    Konvertálja át a valós képet Lab színtérbe.
}

Segítség: src.convertTo(dest, dest_type, scale_factor);
          cvtColor( src, dest, COLOR_BGR2Lab);
```

- Hívja meg a konvertálót az etalonképre.
- Hívja meg a konvertálót a ciklusban felolvasott képre.

Feladat: padlólapok etalontól való eltérése III.

- Írja meg az összehasonlítást végző függvényt, mely két kép pontonként vett teljes színkülönbségének **átlagát** adja vissza.
- A teljes színkülönbség képlete:

$$\Delta E^* = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}$$

```
double compareImg(const Mat etalon_lab, const Mat img_lab) {
    // Határozza meg a különbségképet:
    Mat diff = ________________;
    // Az eredménynek vegye a pontonkénti szorzatát: mat.mul(mat)
     Mat diff2 = ___________________;
    // Bontsa csatornákra
    vector<Mat> chs;
    split(diff2, chs);
    // Összegezze a csatornákat a pontonkénti gyökvonás előtt:
    Mat dist;
    cv::sqrt(_____________________, dist);
    // Átlagolja a teljes színkülönbség mátrix értékeit
    return mean(dist)[0];
}
```

Feladat

- Készítsen egy függvényt, ami a táblázat alapján visszaadja a kapott eredménynek megfelelő kategóriát!

```
string get_category(double diff);
```

Teljes színkülönbség	Kategória
0–0.5	Szemmel nem érzékelhető
0.5–1.5	Alig érzékelhető
1.5–3.0	Érzékelhető
3.0–6.0	Jól látható
6.0–	Nagy színinger különbség

- Jelenítse meg minden képre a mért színkülönbséget és a kategóriát.

Fontosabb eszközök

- Képpont elérés: `img.at<pixel_típus>(sor_idx, oszlop_idx)`
- Típus váltás: `img.convertTo(dest, dest_típus, skálázási_faktor)`
- Színtér váltás: `cvtColor(src, dest, konverziós_kód)`
- Pontonkénti mátrixszorzás: `img.mul(img2)`
- Pontonkénti hatványozás: `cv::pow( img, 2, dest)`
- Pontonkénti gyökvonás: `cv::sqrt( img, dest )`
- átlagszámítás: `Scalar s = mean(img);  //csatornánként`
- csatornák szétbontása: `split(src, chs)`

Köszönöm a figyelmet!

A diákon nem jelölt képek (vagy azok eredetiének) származási helye:

https://i.imgur.com/aoXWFkxl.jpg
https://en.wikipedia.org/wiki/File:HSV_color_solid_cone_chroma_gray.png
http://www.texample.net/media/tikz/examples/PDF/cielab.pdf
https://commons.wikimedia.org/wiki/File:Lab_color_space.png
https://pbs.twimg.com/media/DUJZVZvWkAE6Tar?format=jpg&name=900x900