

Seminář 8

Práce s obrázky

- Obrázek intenzit - obrázek je reprezentován jako matice intenzit (šedotónový obraz).
- Barevný obrázek - matice $m \times n \times 3$, každý pixel představuje barvu v matici.
- Indexovaný obrázek - datová matice, a barevná mapa, data obsahují indexy do palety barev.

Načtení obrázku

`imread()`

```
obrazek_rgb = imread('pastelky.png');
```

Převod na šedotónový

`rgb2gray()`, `im2gray()`

```
obrazek_gray = rgb2gray(obrazek_rgb);
```

Zobrazení obrázku

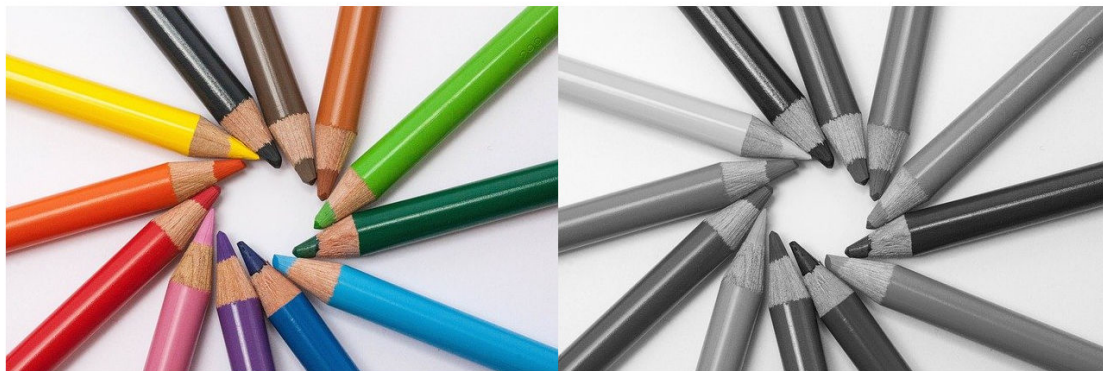
`imshow()`

```
figure,  
subplot(1,2,1), imshow(obrazek_gray);  
subplot(1,2,2), imshow(obrazek_rgb);
```



`imshowpair()`

```
figure, imshowpair(obrazek_rgb,obrazek_gray,"montage");
```



Prozkoumejte další parametry `imshowpair()`

Uložení obrázku

`imwrite()`

```
imwrite(obrazek_gray,'pastelky_gray.png');
```

Jak zjistíme velikost obrázku?

```
[m1, n1, o1] = size(obrazek_gray)
```

```
m1 = 426  
n1 = 640  
o1 = 1
```

```
[m2, n2, o2] = size(obrazek_rgb)
```

```
m2 = 426  
n2 = 640  
o2 = 3
```

Barevné roviny R, G, B

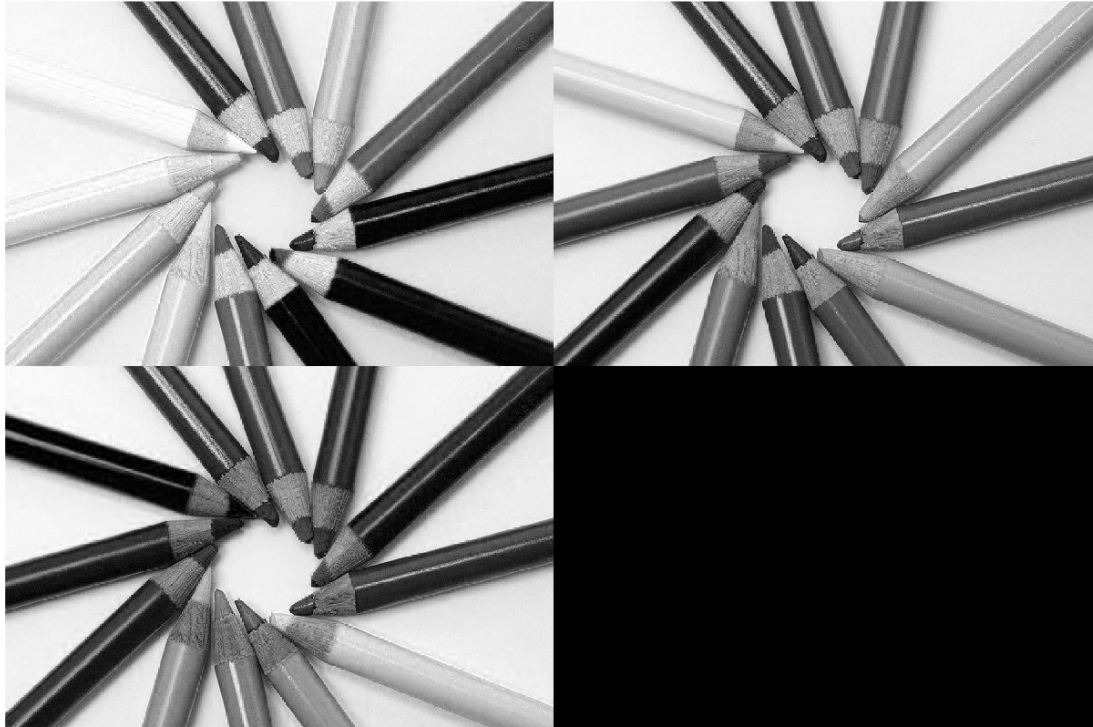
```
R = obrazek_rgb(:,:,1);  
G = obrazek_rgb(:,:,2);  
B = obrazek_rgb(:,:,3);
```

```
figure,  
subplot(1,3,1), imshow(R);  
subplot(1,3,2), imshow(G);  
subplot(1,3,3), imshow(B);
```



```
[R1, G1, B1] = imsplit(obrazek_rgb);  
figure,  
subplot(1,3,1), imshow(R1);
```

```
subplot(1,3,2), imshow(G1);  
subplot(1,3,3), imshow(B1);
```



Základní práce s obrázky

```

for i = 1 : m1
    for j = 1 : n1
        % prace s pixelem na souradnicich [i,j]
    end
end
end

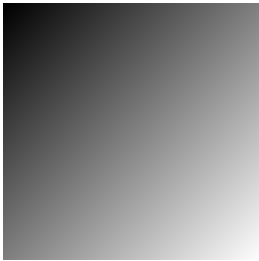
```

Vytvořte nový obrázek představující negativ obrázku `obrazek_gray`. Pixel na souřadnicích `[i,j]` bude mít hodnotu `255 - obrazek_gray(i,j)`.

Nezapomínejme na optimalizaci - prealokace paměti. (`uint8(zeros(m,n))`)

Tato operace jde udělat i maticově. Jak?

Vytvořte následující obrázek.



Na vedlejší diagonále jsou intenzity od 0 do 254.

Toolboxy

- MathWorks: <https://uk.mathworks.com/products.html>
- Toolboxy 3. stran

Image Processing Toolbox

https://uk.mathworks.com/products/image.html?s_tid=srchtitle_toolbox_12

Funkce pro geometrické transformace:

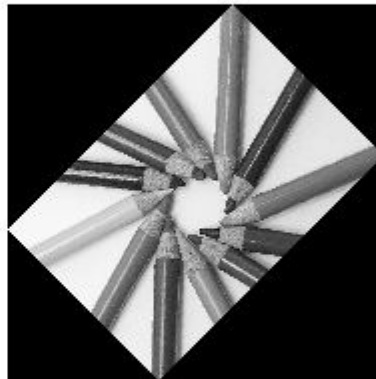
`imcrop()`, `imresize()`, `imrotate()`, `imtranslate()`, `imwarp()`

```

f1 = imread('pastelky_gray.png');
g1 = imrotate(f1,45);

figure,
subplot(1,2,1), imshow(f1);
subplot(1,2,2), imshow(g1);

```



Aplikace

Registration estimator

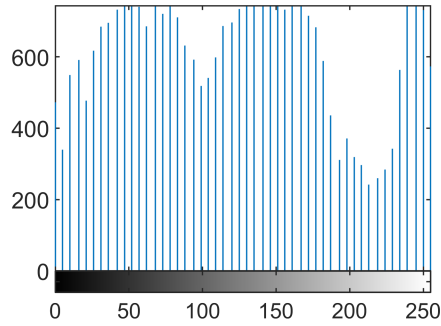
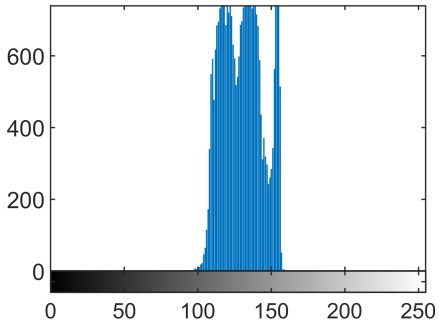
- App - Registration estimator
- load images from workspace (f1, g1)
- register

```
registrationEstimator
```

Funkce pro jasové transformace:

```
imadjust(), imlocalbrighten(), histeq()
```

```
I = imread('p2.png');  
  
figure,  
subplot(2,2,1), imshow(I);  
subplot(2,2,2), imshow(imadjust(I));  
  
% histogram  
subplot(2,2,3), imhist(I,256);  
subplot(2,2,4), imhist(imadjust(I));
```

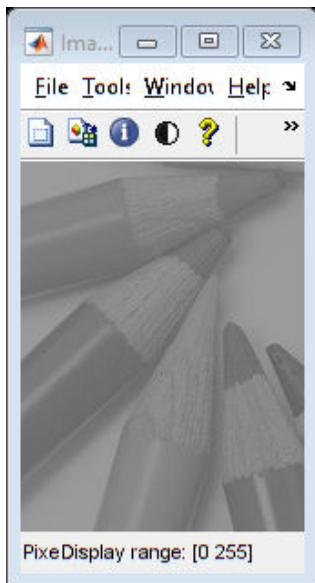


Nástroj pro interaktivní práci s obrázky

`imtool`

(příkaz `imtool(obrazek)`, `imtool`, Apps -> Image viewer)

```
imtool(I);
```



Info ikona - co zde vyčteme?

Tools -> pixel region

Tools -> Adjust contrast

Pomocí Adjust contrast upravte obrázek I a vyexportujte výsledek do workspace jako proměnnou I2.

Segmentace

Různé metody:

- hledání hran (hranice objektu)
- segmentace dle textury
- hledání specifických tvarů (kružnice)
- segmentace dle barvy (zelené pozadí)

Pracujeme s ROI - binární obrázek (maska)

Prahování

```
I = imread('figurky/figurka1a.jpg');  
I2 = imread('figurky/figurka1b.jpg');
```

```
ROI = I < 110; % 240  
ROI2 = I2 < 110;
```

```
figure,  
subplot(2,2,1), imshow(I);  
subplot(2,2,2), imshow(ROI);  
subplot(2,2,3), imshow(I2);  
subplot(2,2,4), imshow(ROI2);
```




`imbinarize()`

```
ROIa = 1 - imbinarize(I);  
ROI2a = 1 - imbinarize(I2);  
  
figure,  
subplot(2,2,1), imshow(I);  
subplot(2,2,2), imshow(ROIa);  
subplot(2,2,3), imshow(I2);  
subplot(2,2,4), imshow(ROI2a);
```



Datastore

Datový typ obsahující informace o obrázcích, ale ne samotné obrázky.

`imageDatastore(složka)`

```
data = imageDatastore("figurky");
```

Ve workspace si prohlédněte proměnnou data.

Cesty k souborům jsou uloženy v položce Files.

```
files = data.Files;
```

Kolik je souborů ve složce?

Práce s obrázky

```
I = imread(files{1});  
figure, imshow(I);
```



Načítání n-tého obrázku z datastore

```
readimage(datastore,n)
```

```
I = readimage(data,6);  
figure, imshow(I);
```



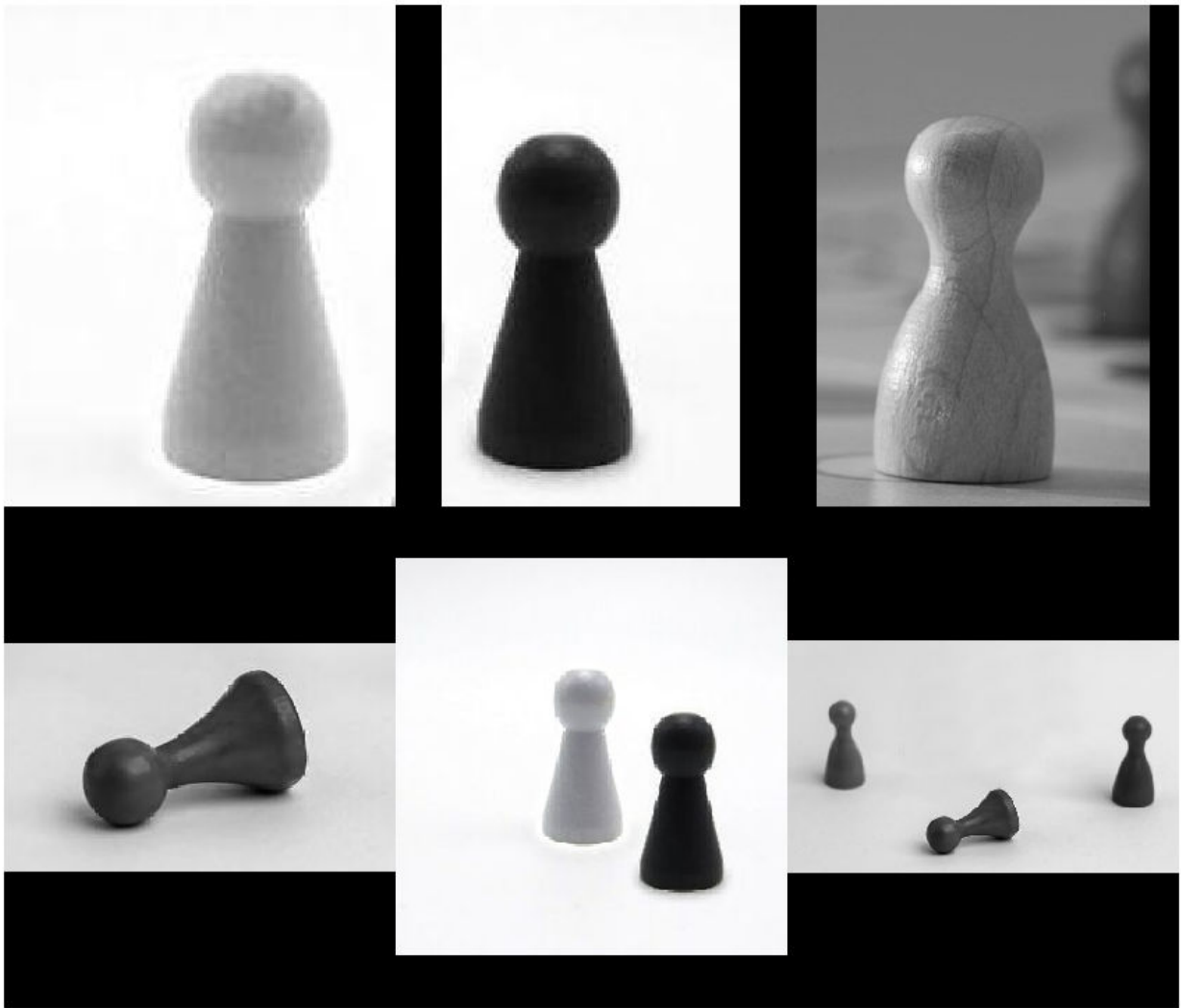
Načtěte postupně všechny soubory z datastore data a zobrazte je.

```
figure,
```

```
for i = 1 : size(files,1) % pripadne numel(files)
    I = readimage(data,i);
    subplot(1,size(files,1), i), imshow(I);
end
```



```
figure,  
montage(files);
```



Načtení všech obrázků

```
readall(datastore)
```

```
Images = readall(data);
```

Jak jsou obrázky uloženy a jak bychom s nimi pracovali? Zobrazte první z nich.

Funkce pro analýzu obrazu:

entropie: `entropy()`, `entropyfilt()`

analýza okolí pixelu: `rangefilt()` - nejmenší a největší hodnota, `stdfilt()` - standardní odchylka

kvalita obrazu: `immse()`, `psnr()`,

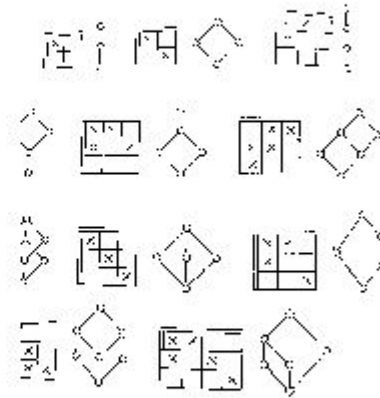
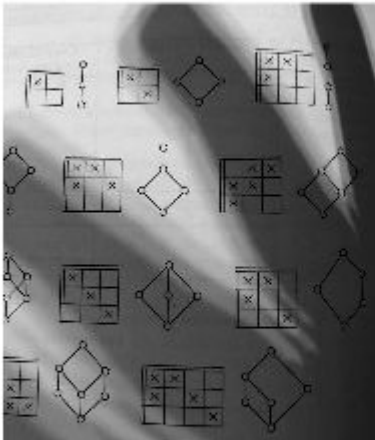
Využití například pro hledání lokální prahové hodnoty.

```
f3 = imread('lokalni.jpg');
f4=im2double(f3);
nhood = ones(3)/9;
g_std = stdfilt(f4);
g_mean = imfilter(f4,nhood);

a=0.3;
b=1-a;
T = a*g_std + b*g_mean;

g4 = imbinarize(f4,T);

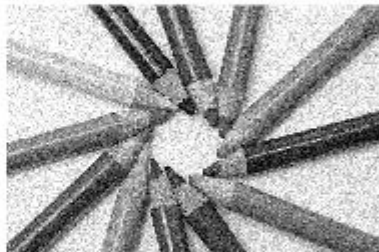
figure,
subplot(1,2,1), imshow(f4);
subplot(1,2,2), imshow(g4);
```



Porovnání dvou obrázků

```
obrazek_gray = imread('pastelky_gray.png');

% pridani sumu do obrazu
obrazek_noise = imnoise(obrazek_gray, 'gaussian');
figure,
subplot(1,2,1), imshow(obrazek_gray);
subplot(1,2,2), imshow(obrazek_noise);
```

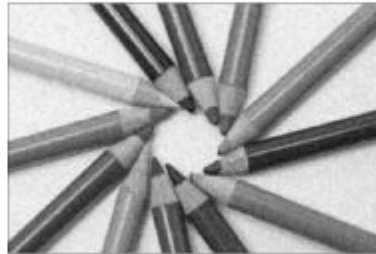
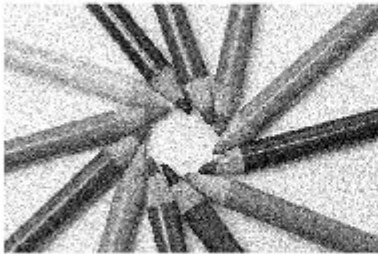


```
% RMSE cim je mensi, tim si jsou obrazky podobnejsi  
fprintf('\n Mean-squared error je %0.4f\n', immse(obrazek_gray, obrazek_noise));
```

Mean-squared error je 569.9688

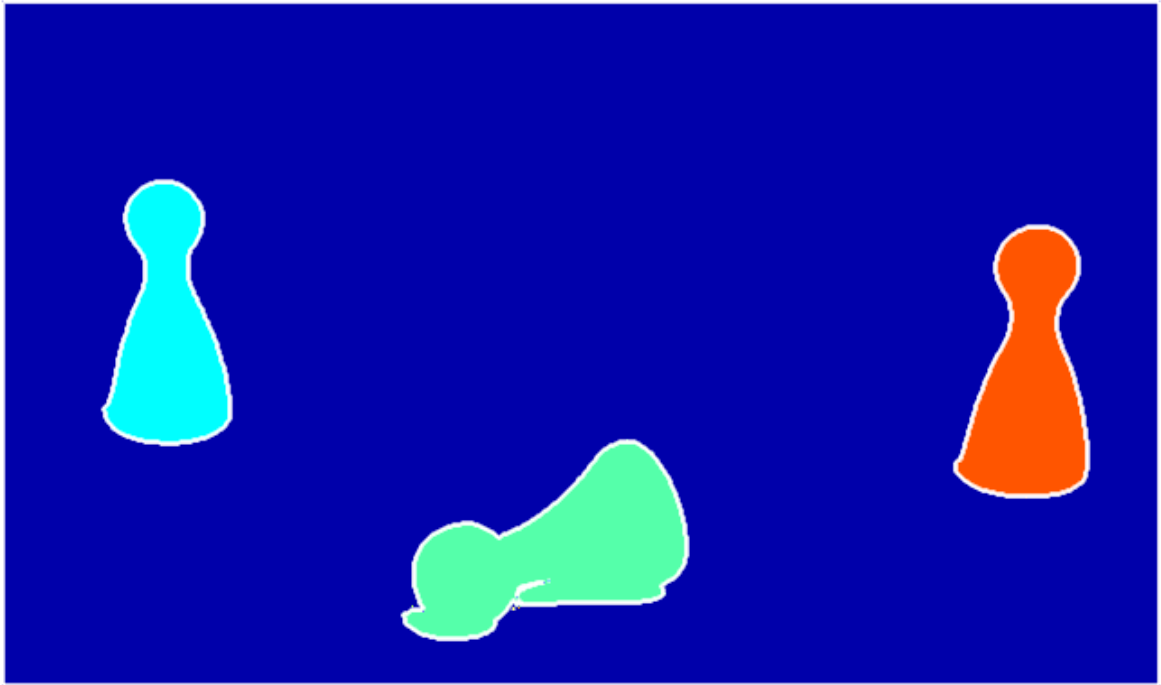
Filtrování obrázků

```
% vytvoreni filtru  
H = fspecial('average',5);  
obrazek_filtrovany = imfilter(obrazek_noise,H);  
  
figure,  
subplot(1,2,1), imshow(obrazek_noise);  
subplot(1,2,2), imshow(obrazek_filtrovany);
```



Ukázka segmentace

```
obrazek_gray = imread('figurky/figurky3.jpg');  
obrazek_bw = imbinarize(obrazek_gray);  
[hrany,L] = bwboundaries(obrazek_bw);  
  
figure,  
imshow(label2rgb(L, @jet, [.5 .5 .5]))  
hold on  
for k = 1:length(hrany)  
    hrana = hrany{k};  
    plot(hrana(:,2), hrana(:,1), 'w', 'LineWidth', 2)  
end  
hold off;
```

App - image segmenter

```
I = imread('figurky/figurky3.jpg');  
imageSegmenter(I);
```



Načtěte obrázek figurky3.jpg a vyzkoušejte si práci se segmenterem.

```
BW1 = imread('masks/mask1.png');  
BW2 = imread('masks/mask2.png');  
BW3 = imread('masks/mask3.png');  
  
B = imoverlay(I,BW1,'yellow');  
C = imoverlay(B,BW2,'red');  
D = imoverlay(C,BW3,'green');  
  
figure, imshow(D);
```

