

Seminář 1

Matlab: více seminář PVM

<https://www.marketa-trneckova.cz/matlab.html>

Nápověda k příkazům

help prikaz

help **sum**

sum Sum of elements.

`S = sum(X)` is the sum of the elements of the vector `X`. If `X` is a matrix, `S` is a row vector with the sum over each column. For N-D arrays, `sum(X)` operates along the first non-singleton dimension.

`S = sum(X,"all")` sums all elements of `X`.

`S = sum(X,DIM)` sums along the dimension `DIM`.

`S = sum(X,VECDIM)` operates on the dimensions specified in the vector `VECDIM`. For example, `sum(X,[1 2])` operates on the elements contained in the first and second dimensions of `X`.

`S = sum(...,OUTTYPE)` specifies the type in which the sum is performed, and the type of `S`. Available options are:

- "double" - `S` has class double for any input `X`
- "native" - `S` has the same class as `X`
- "default" - If `X` is floating point, that is double or single, `S` has the same class as `X`. If `X` is not floating point, `S` has class double.

`S = sum(...,NANFLAG)` specifies how NaN values are treated:

- "includemissing" / "includenan" -
(default) The sum of a vector containing NaN values is also NaN.
- "omitmissing" / "omitnan" -
The sum of a vector containing NaN values is the sum of all its non-NaN elements. If all elements are NaN, the result is 0.

Examples:

```
X = [0 1 2; 3 4 5]
sum(X, 1)
sum(X, 2)
```

```
X = int8(1:20)
sum(X)           % returns double(210), accumulates in double
sum(X,'native')  % returns int8(127), because it accumulates in
                  % int8 but overflows and saturates.
```

See also `prod`, `cumsum`, `diff`, `accumarray`, `isfloat`.

Documentation for `sum`
Other uses of `sum`

Práce s obrázky

Načtení obrázku:

```
imread(cesta);
```

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

Zobrazení obrázku:

```
imshow(obrazek);
```

```
imshow(I);
```



Roztažení intenzit:

všechny hodnoty menší než low budou nastaveny na 0, všechny větší než height na 255. Hodnoty mezi low a height budou roztaženy pravidelně mezi 0 a 255

```
imshow(obrazek, [low,high]);
```

Při vynechání low a high je na 0 je nastavena nejnižší hodnota a na 255 nejvyšší

```
imshow(obrazek, []);
```

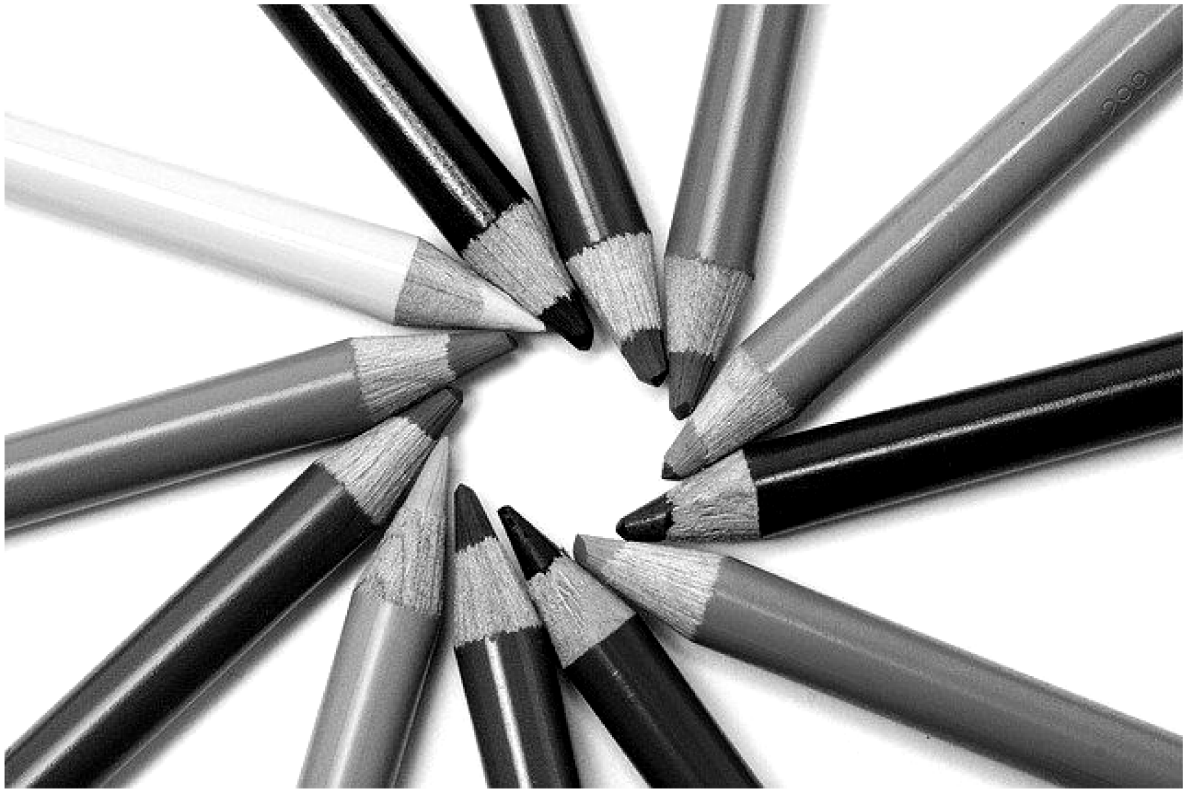
```
low = 48
```

```
low = 48
```

```
high = 212
```

```
high = 212
```

```
imshow(I,[low,high]);
```



Pokud za příkaz nepřidáme ; (středník) vypíše se výsledná hodnota

Hodnota pixelu na souřadnicích:

$I(x,y)$

```
I(1,1)
```

```
ans = uint8
```

```
248
```

pozor indexuje se od 1!

Pro výpis hodnoty můžeme použít funkci `disp`, případně `fprintf` se syntaxí podobnou jako má funkce `printf` v jazyce C

```
disp(I(1,1));
```

248

Velikost obrázku:

h - výška obrázku (počet řádků), w - šířka obrázku (počet sloupců), o - dimenze určující kolik hodnot je potřeba k reprezentaci informace o obraze (barevný = 3, šedotónový = 1)

size(promenna)

```
[h,w,o] = size(I);  
fprintf(" Vyska je %i \n Sirka je %i \n Pocet slozek je %i", h, w, o);
```

```
Vyska je 426  
Sirka je 640  
Pocet slozek je 1
```

Barevný obrázek

```
I_rgb = imread("pastelky.png");  
figure, imshow(I_rgb);
```




```
disp(I_rgb(1,1,:));
```

```
(:,:,1) =
```

```
248
```

```
(:,:,2) =
```

```
248
```

```
(:,:,3) =
```

```
250
```

```
[h,w,o] = size(I_rgb);
```

```
fprintf(" Vyska je %i \n Sirka je %i \n Pocet slozek je %i", h, w, o);
```

```
Vyska je 426
```

```
Sirka je 640
```

```
Pocet slozek je 3
```

EM spektrum

```
addpath 'spectral_color'
```

```
lambda = 380:0.2:780;
```

```
rgbValues = spectrumRGB(lambda);
```

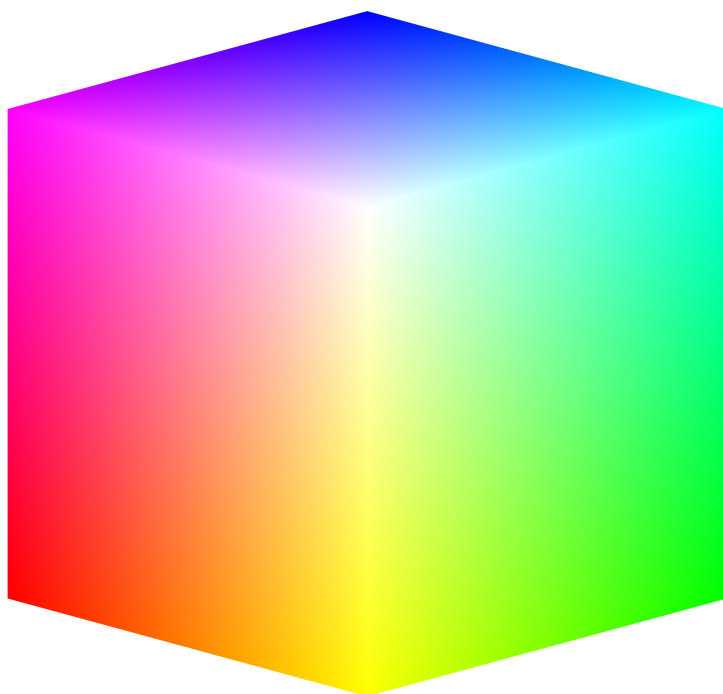
```
image = repmat(rgbValues, 100, 1, 1);
```

```
figure, imshow(image)
```



RGB

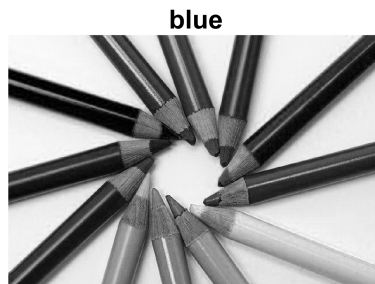
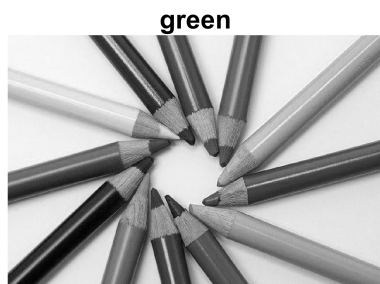
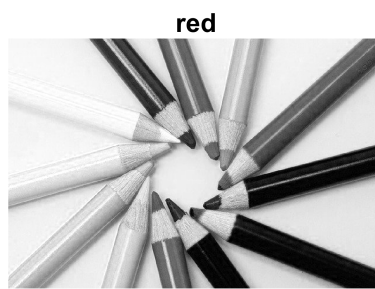
```
figure, RGBcube();
```



RGB

Zobrazení jednotlivých složek obrazu:

```
I_rgb = imread("pastelky.png");  
figure;  
subplot(2,2,1), imshow(I_rgb);  
title('original');  
subplot(2,2,2), imshow(I_rgb(:,:,1));  
title('red');  
subplot(2,2,3), imshow(I_rgb(:,:,2));  
title('green');  
subplot(2,2,4), imshow(I_rgb(:,:,3));  
title('blue');
```



RGB to GRAY

```
figure, subplot(1,3,1), imshow(I_rgb);
title('original');

% prumerovanim slozek
gray1 = (1/3)*I_rgb(:,:,1) + (1/3)*I_rgb(:,:,2) + (1/3)*I_rgb(:,:,3);
subplot(1,3,2), imshow(gray1,[]);
title('prumerovani');

% vazeny prumer
gray2 = rgb2gray(I_rgb);
subplot(1,3,3), imshow(gray2,[]);
title('vazene prumerovani');
```



RGB - CMY

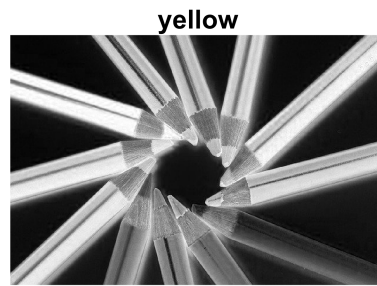
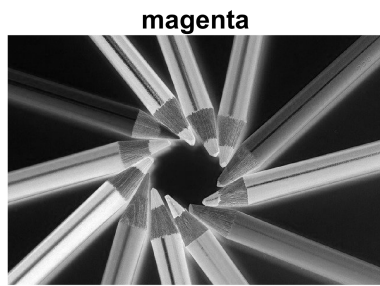
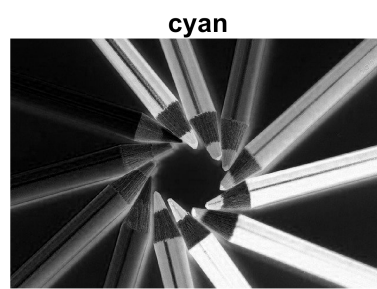
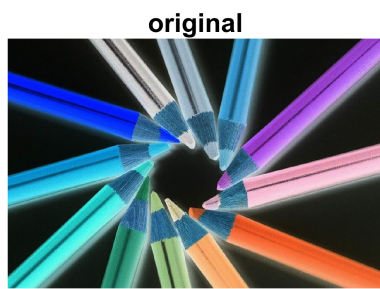
rgb2cmy(), cmy2rgb()

```
I_cmy = rgb2cmy(I_rgb);  
%I_cmy = 255 - I_rgb;  
I_rgb2 = cmy2rgb(I_cmy);  
  
figure, subplot(1,3,1), imshow(I_rgb);  
title('RGB');  
subplot(1,3,2), imshow(I_cmy);  
title('CMY');  
subplot(1,3,3), imshow(I_rgb2);  
title('RGB');
```



Zobrazení jednotlivých složek obrazu:

```
figure;  
subplot(2,2,1), imshow(I_cmy);  
title('original');  
subplot(2,2,2), imshow(I_cmy(:,:,1));  
title('cyan');  
subplot(2,2,3), imshow(I_cmy(:,:,2));  
title('magenta');  
subplot(2,2,4), imshow(I_cmy(:,:,3));  
title('yellow');
```



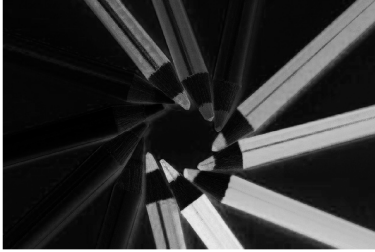
CMYK

```
C = makecform('srgb2cmyk');    %srgb (standard rgb) to cmyk

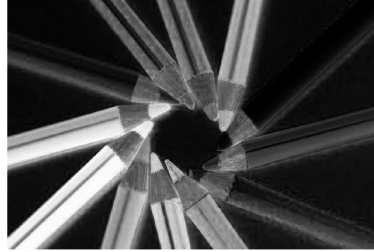
% aplikace transformace
I_cmyk= applycform(I_rgb,C);

figure;
subplot(2,2,1), imshow(I_cmyk(:,:,1));
title('Cyan');
subplot(2,2,2), imshow(I_cmyk(:,:,2));
title('Magenta');
subplot(2,2,3), imshow(I_cmyk(:,:,3));
title('Yellow');
subplot(2,2,4), imshow(I_cmyk(:,:,4));
title('black');
```

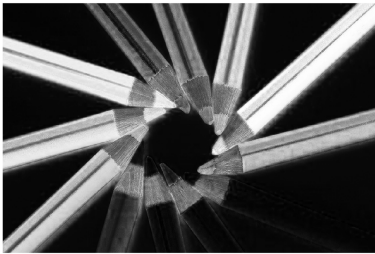

Cyan



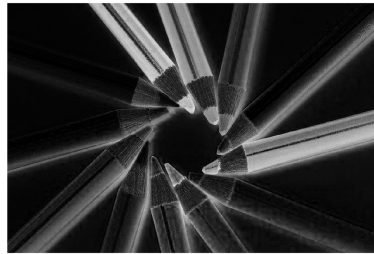
Magenta



Yellow



black



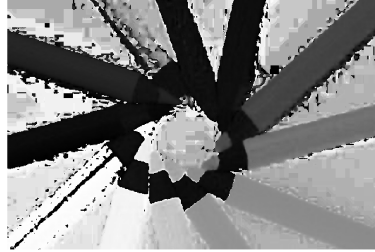
RGB to HSV

```
I_hsv = rgb2hsv(I_rgb);  
  
figure, subplot(2,2,1), imshow(I_rgb);  
title('RGB');  
subplot(2,2,2), imshow(I_hsv(:,:,1));  
title('hue');  
subplot(2,2,3), imshow(I_hsv(:,:,2));  
title('saturation');  
subplot(2,2,4), imshow(I_hsv(:,:,3));  
title('value');
```

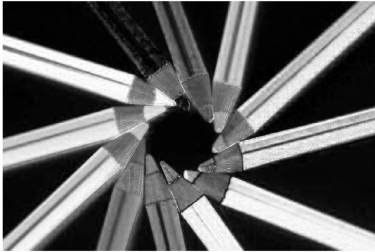
RGB



hue



saturation



value



HSV to RGB

```
I_rgb2 = hsv2rgb(I_hsv);  
figure, imshow(I_rgb2);
```



RGB to YCbCr

```
I_ycbcr = rgb2ycbcr(I_rgb);  
  
figure, subplot(2,2,1), imshow(I_rgb);  
title('RGB');  
subplot(2,2,2), imshow(I_ycbcr(:,:,1));  
title('Y');  
subplot(2,2,3), imshow(I_ycbcr(:,:,2));  
title('Cb');  
subplot(2,2,4), imshow(I_ycbcr(:,:,3));  
title('Cr');
```

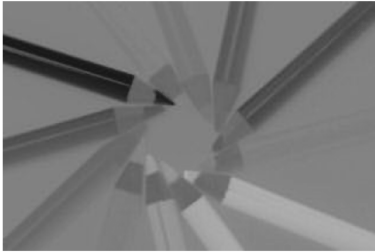
RGB



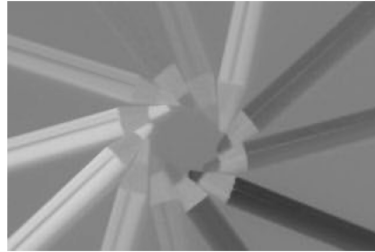
Y



Cb



Cr



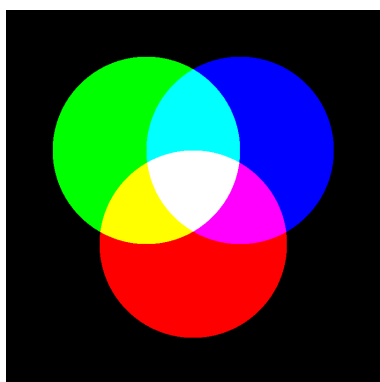
YCbCr to RGB

```
I_rgb2 = ycbcr2rgb(I_ycbcr);  
figure, imshow(I_rgb2);
```



Úkol 2

Načrtněte, jak by vypadaly jednotlivé barevné složky následujícího obrázku v RGB modelu.



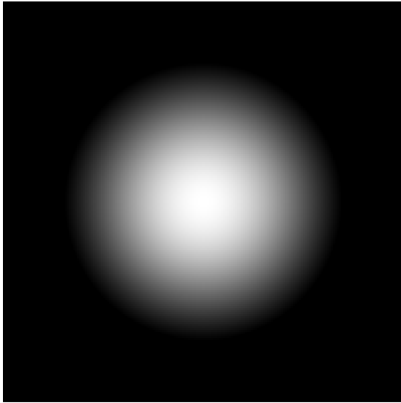
Řešení ověřte pomocí Matlab

```
I = imread('barvy.png');
```

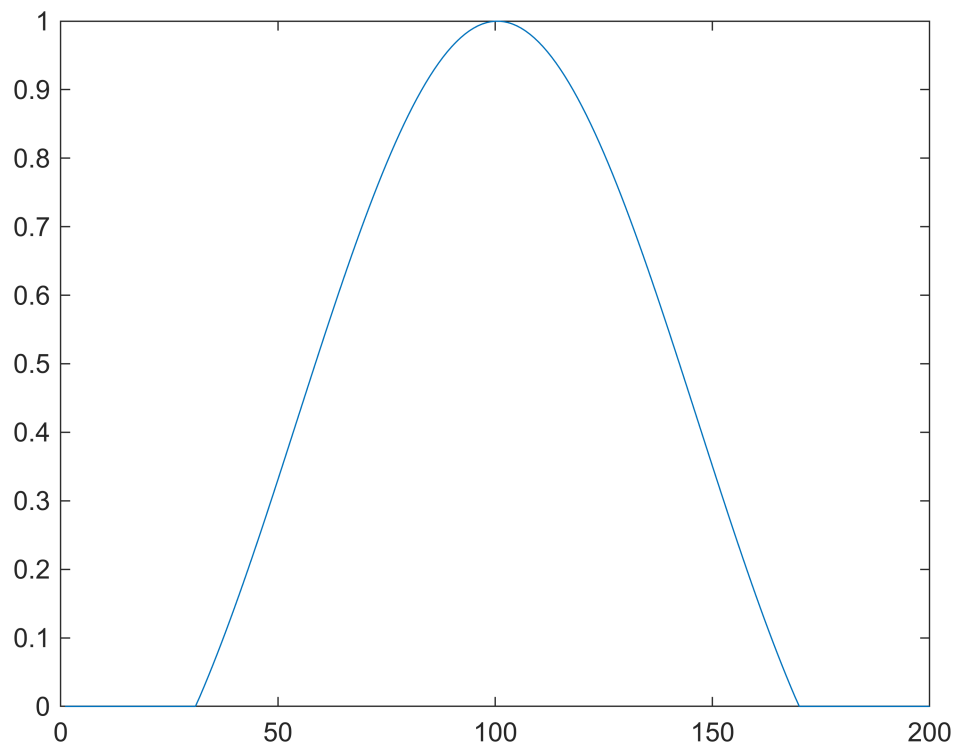
Digitalizace

```
tx = linspace (-4.5, 4.5, 200)';  
ty = linspace (-4.5, 4.5, 200)';
```

```
[xx, yy] = meshgrid (tx, ty);  
r = sqrt (xx .^ 2 + yy .^ 2) + eps;  
I = sin (r) ./ r;  
I(I<0) = 0;  
figure, imshow(I);
```

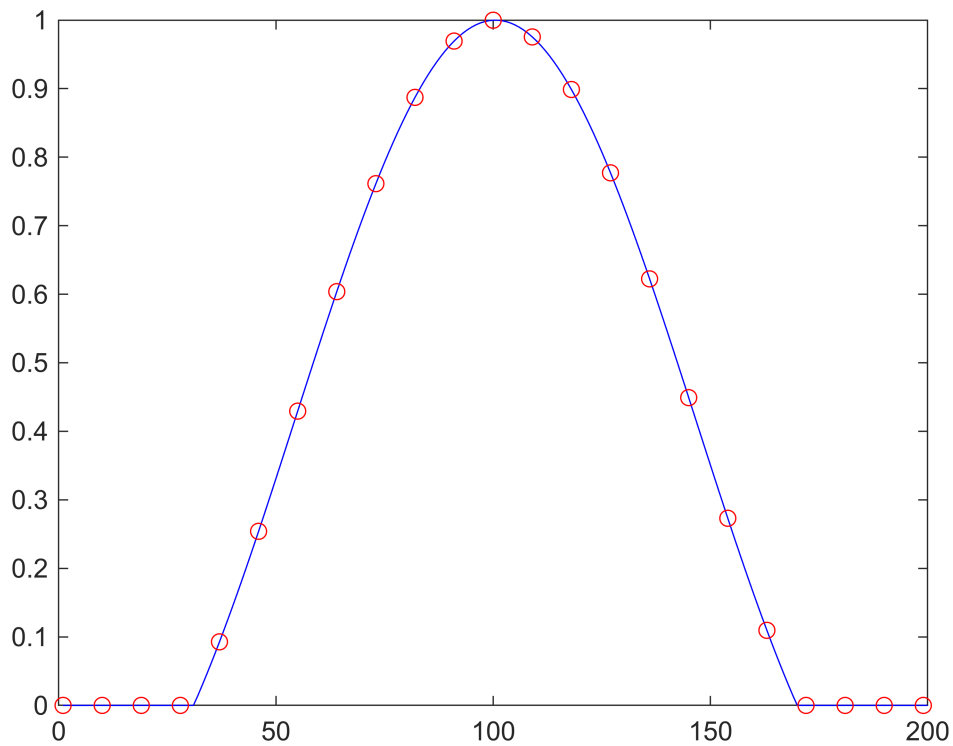


```
J = I(100,:);  
x = 1 : 200;  
figure, plot(x, J);
```



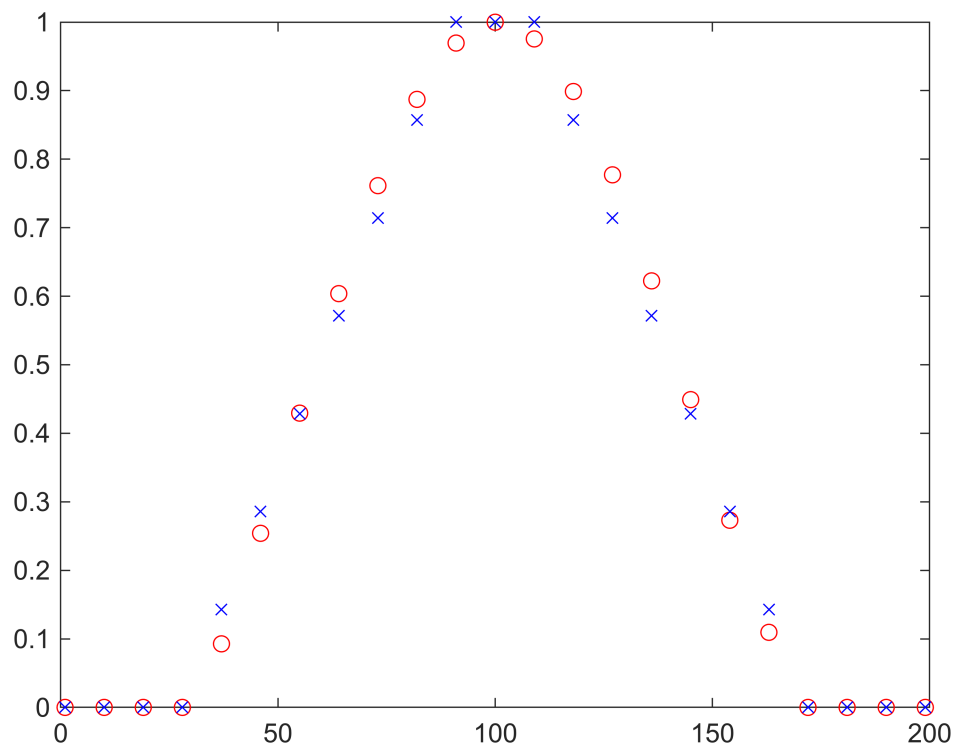
Vzorkování

```
delta = 9;  
J2 = J(1:delta:end);  
x2 = x(1:delta:end);  
  
figure,  
plot(x,J,'b-');  
hold on  
plot(x2,J2,'ro');  
hold off
```



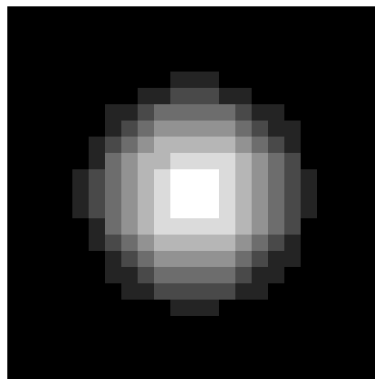
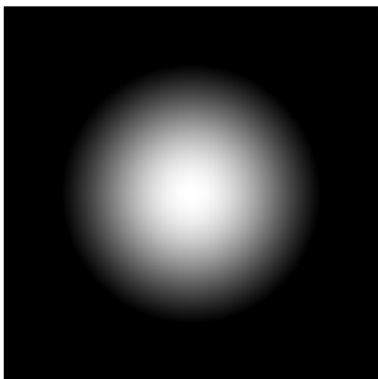
Kvantování

```
pocet_barev = 8;  
  
J3 = round(J2 * (pocet_barev-1))/(pocet_barev-1);  
  
figure,  
plot(x2,J2,'ro');  
hold on  
plot(x2,J3,'bx');  
hold off
```



```
I_vysledek = round(I(1:delta:end,1:delta:end)* (pocet_barev-1))/(pocet_barev-1);

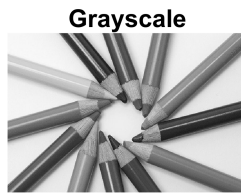
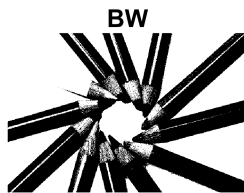
figure,
subplot(1,2,1), imshow(I);
subplot(1,2,2), imshow(I_vysledek);
```



Typy obrázků

```
I_rgb = imread('pastelky.jpg');  
I_gray = rgb2gray(I_rgb);  
I_bw = imbinarize(I_gray);
```

```
figure,  
subplot(1,3,1), imshow(I_bw);  
title('BW');  
subplot(1,3,2), imshow(I_gray);  
title('Grayscale');  
subplot(1,3,3), imshow(I_rgb);  
title('RGB');
```



Indexové obrázky

```
figure,
map = colormap('hot');
% vytvoreni obrazku palety
obr = cat(3, map(:,1)',map(:,2)', map(:,3)');
% zvetseni vysky pro lepsi vizualizaci
paleta = imresize(obr,[10,256]);

imshow(paleta);
```



```
figure, imshow(I_gray,map);
```



Rozlišení

Ukládání obrázku

```
A = imread('pastelky.jpg');  
B = rgb2gray(A);  
imwrite(B, 'pastelkygray.jpg');
```

```
imfinfo('pastelkygray.jpg')
```

```
ans = struct with fields:  
    Filename: 'C:\Users\marke\OneDrive\Skola\vyuka2024-2025\ZS\XVIT\matlab\pr1\pastelkygray.jpg'  
    FileModDate: '07-Sep-2024 13:30:00'  
    FileSize: 50729  
    Format: 'jpg'  
    FormatVersion: ''  
    Width: 640  
    Height: 426  
    BitDepth: 8  
    ColorType: 'grayscale'  
    FormatSignature: ''  
    NumberOfSamples: 1  
    CodingMethod: 'Huffman'  
    CodingProcess: 'Sequential'  
    Comment: {}
```

```
B = imread('pastelkygray.jpg');
```

Dle zvoleného formátu lze zadat další vlastnosti uloženého obrázku. Například kvalitu nebo rozlišení

```
imwrite(B, 'p1.tif', 'resolution', [1000, 1000]);  
imwrite(B, 'p2.tif', 'resolution', [500, 500]);  
imwrite(B, 'p3.tif', 'resolution', [250, 250]);
```

Oba obrázky mají stejnou velikost.

```
A1 = imfinfo('p1.tif');  
A2 = imfinfo('p3.tif');  
  
fprintf("Obrázek %s ma vysku %i a sirku %i.", A1.Filename, A1.Width, A1.Height);
```

Obrázek C:\Users\marke\OneDrive\Skola\vyuka2024-2025\ZS\XVIT\matlab\pr1\p1.tif ma vysku 640 a sirku 426.

```
fprintf("Obrázek %s ma vysku %i a sirku %i.", A2.Filename, A2.Width, A2.Height);
```

Obrázek C:\Users\marke\OneDrive\Skola\vyuka2024-2025\ZS\XVIT\matlab\pr1\p3.tif ma vysku 640 a sirku 426.

Liší se ale rozlišením pixelů

```
fprintf("Obrázek %s ma x-rozliseni %i a y-rozliseni %i.", A1.Filename,  
A1.XResolution, A1.YResolution);
```

Obrázek C:\Users\marke\OneDrive\Skola\vyuka2024-2025\ZS\XVIT\matlab\pr1\p1.tif ma x-rozliseni 1000 a y-rozliseni 1000

```
fprintf("Obrázek %s ma x-rozliseni %i a y-rozliseni %i.", A2.Filename,  
A2.XResolution, A2.YResolution);
```

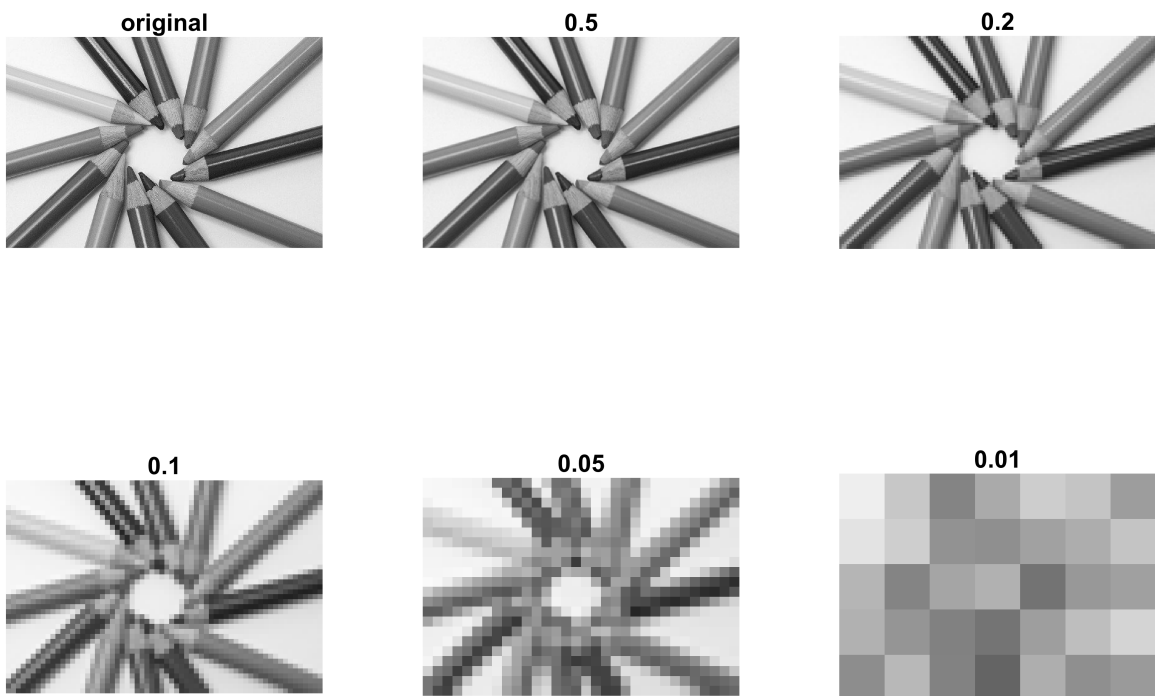
Obrázek C:\Users\marke\OneDrive\Skola\vyuka2024-2025\ZS\XVIT\matlab\pr1\p3.tif ma x-rozliseni 250 a y-rozliseni 250

Rozlišení ve smyslu počtu pixelů

```
B1 = imresize(B, 0.5);  
B2 = imresize(B, 0.2);  
B3 = imresize(B, 0.1);  
B4 = imresize(B, 0.05);  
B5 = imresize(B, 0.01);  
  
subplot(2,3,1), imshow(B);  
title('original');  
subplot(2,3,2), imshow(B1);  
title('0.5');  
subplot(2,3,3), imshow(B2);  
title('0.2');  
subplot(2,3,4), imshow(B3);  
title('0.1');  
subplot(2,3,5), imshow(B4);  
title('0.05');
```



```
subplot(2,3,6), imshow(B5);
title('0.01');
```

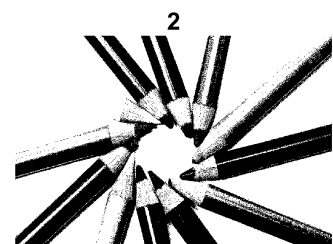
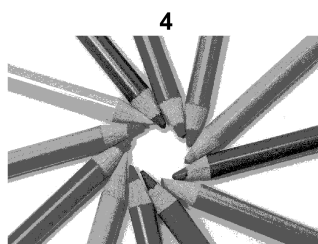
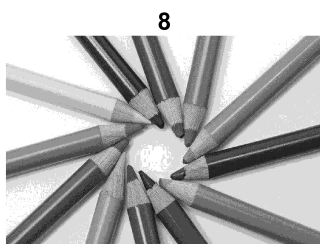
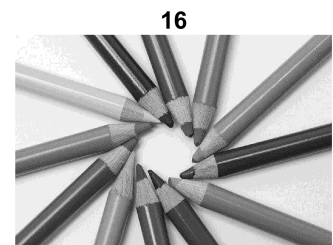
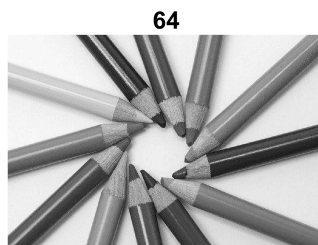
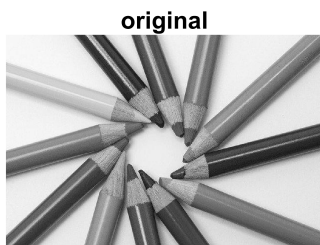


Barevná hloubka

```
Bi = gray2ind(B,256);
Bi1 = gray2ind(B,64);
Bi2 = gray2ind(B,16);
Bi3 = gray2ind(B,8);
Bi4 = gray2ind(B,4);
Bi5 = gray2ind(B,2);

subplot(2,3,1), imshow(Bi,[]);
title('original');
subplot(2,3,2), imshow(Bi1,[]);
title('64');
subplot(2,3,3), imshow(Bi2,[]);
title('16');
subplot(2,3,4), imshow(Bi3,[]);
title('8');
subplot(2,3,5), imshow(Bi4,[]);
title('4');
subplot(2,3,6), imshow(Bi5,[]);
```

```
title('2')
```



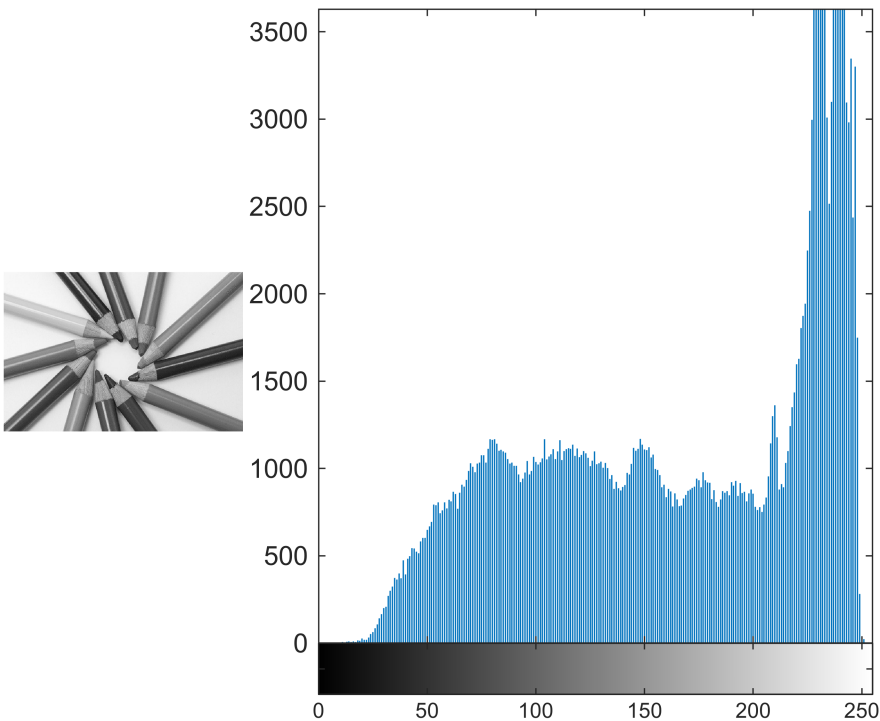
Histogram

Funkce

`imhist()`

`histogram()`

```
figure,  
subplot(1,3,1), imshow(I_gray);  
subplot(1,3,2:3), imhist(I_gray);
```



Úkol 4

Mohou mít dva různé obrázky stejný histogram?

Vzdálenosti

```
M = zeros(9);
M(5,5) = 1;
% city block
D_4 = bwdist(M, 'cityblock')
```

```
D_4 = 9x9 single matrix
    8     7     6     5     4     5     6     7     8
    7     6     5     4     3     4     5     6     7
    6     5     4     3     2     3     4     5     6
    5     4     3     2     1     2     3     4     5
    4     3     2     1     0     1     2     3     4
    5     4     3     2     1     2     3     4     5
    6     5     4     3     2     3     4     5     6
    7     6     5     4     3     4     5     6     7
    8     7     6     5     4     5     6     7     8
```

```
% D8
D_8 = bwdist(M, 'chessboard')
```

```
D_8 = 9x9 single matrix
    4     4     4     4     4     4     4     4     4
    4     3     3     3     3     3     3     3     4
    4     3     2     2     2     2     2     3     4
```

4	3	2	1	1	1	2	3	4
4	3	2	1	0	1	2	3	4
4	3	2	1	1	1	2	3	4
4	3	2	2	2	2	2	3	4
4	3	3	3	3	3	3	3	4
4	4	4	4	4	4	4	4	4

```
% Euklidovska
```

```
D_euclid = bwdist(M, 'euclidean')
```

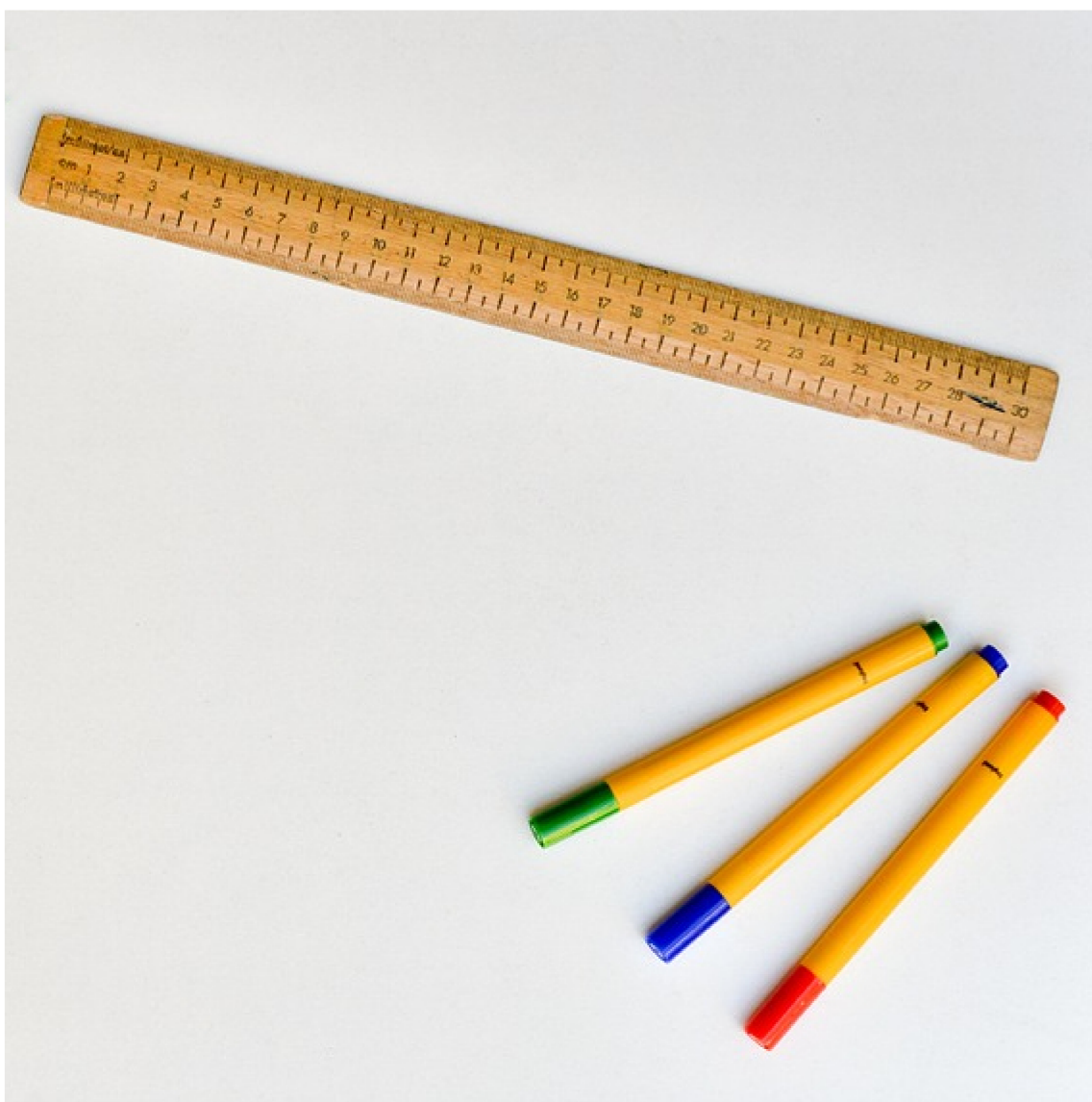
```
D_euclid = 9x9 single matrix
```

5.6569	5.0000	4.4721	4.1231	4.0000	4.1231	4.4721	5.0000	...
5.0000	4.2426	3.6056	3.1623	3.0000	3.1623	3.6056	4.2426	
4.4721	3.6056	2.8284	2.2361	2.0000	2.2361	2.8284	3.6056	
4.1231	3.1623	2.2361	1.4142	1.0000	1.4142	2.2361	3.1623	
4.0000	3.0000	2.0000	1.0000	0	1.0000	2.0000	3.0000	
4.1231	3.1623	2.2361	1.4142	1.0000	1.4142	2.2361	3.1623	
4.4721	3.6056	2.8284	2.2361	2.0000	2.2361	2.8284	3.6056	
5.0000	4.2426	3.6056	3.1623	3.0000	3.1623	3.6056	4.2426	
5.6569	5.0000	4.4721	4.1231	4.0000	4.1231	4.4721	5.0000	

Úkol 8

Určete skutečnou velikost fix na obrázku.

```
I = imread('priklad1.jpg');  
figure, imshow(I);
```



K měření vzdáleností v obraze je možné využít nástroj `imageView(I)`