

## Seminář 4

### Transformační funkce

$$g(x,y) = T[f(x,y)]$$

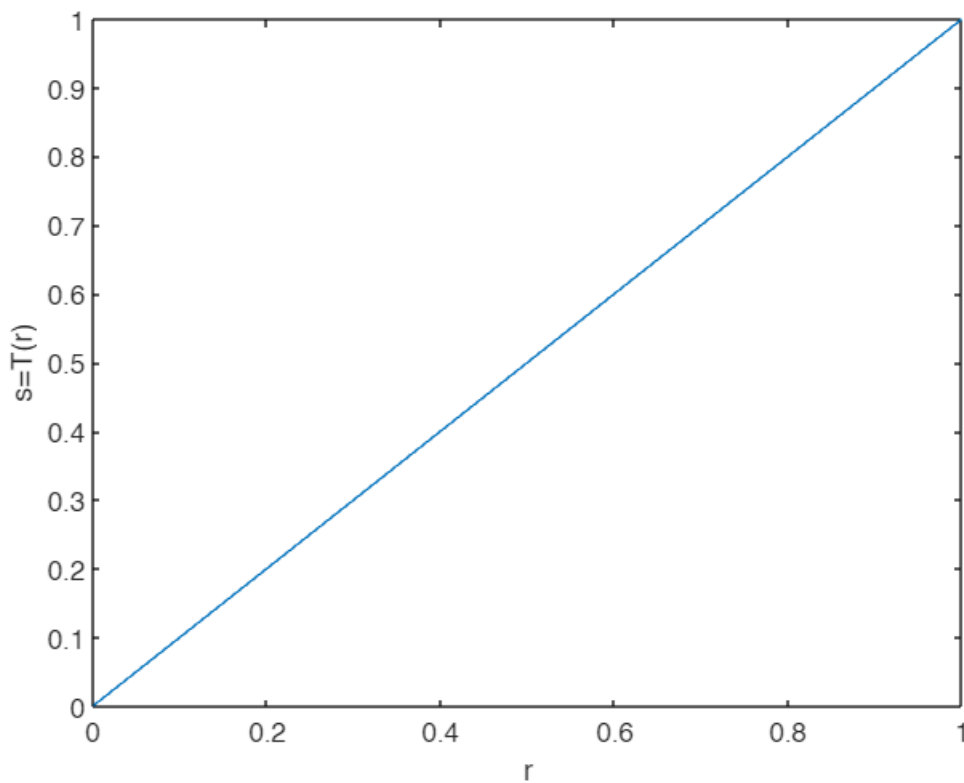
f ... vstupní obrázek

g ... výstupní obrázek

T - transformace barev

$$s = T(r)$$

```
figure,  
r = (0:255)/255;  
plot(r,r);  
xlabel('r');  
ylabel('s=T(r)');  
xlim([0,1]);
```



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

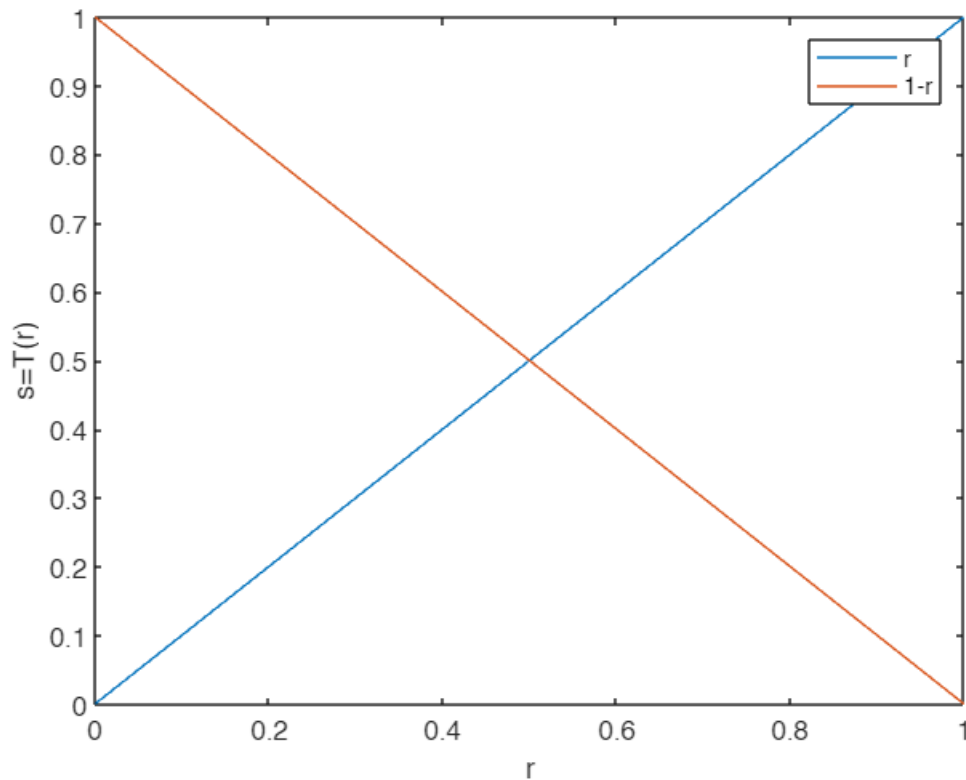
### Negativ obrazku

```
figure,  
s5 = 1-r;
```

```

plot(r,r);
xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s5);
legend('r','1-r','Location','northeast');

```



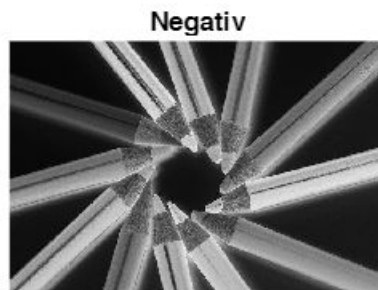
### Aplikace negativu

```

map5 = createMap(s5);

figure
subplot(1,2,1)
imshow(f)
title('Original')
subplot(1,2,2)
imshow(f,map5)
title('Negativ')

```



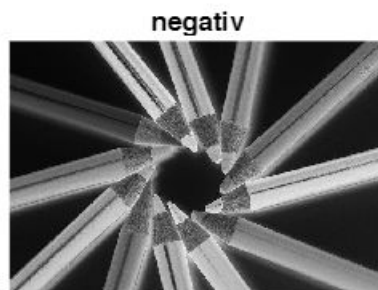
### Aplikace na každý pixel

```
neg = 255 - I;
```

```
neg = imcomplement(I);
```

```
g1 = imcomplement(f);
```

```
figure  
subplot(1,2,1)  
imshow(f)  
title('Original')  
subplot(1,2,2)  
imshow(g1)  
title('negativ')
```

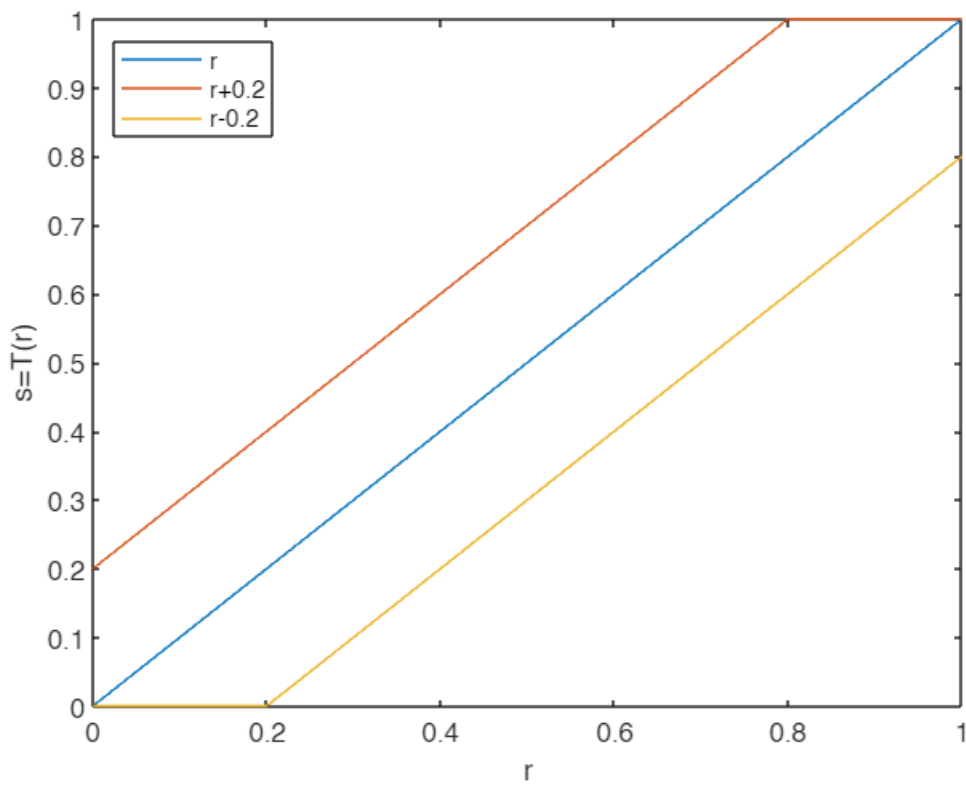


## Změna jasu

```
k1= 0.2;
k2 = -0.2;

s1 = clipMap(r + k1,0,1);
s2 = clipMap(r + k2,0,1);

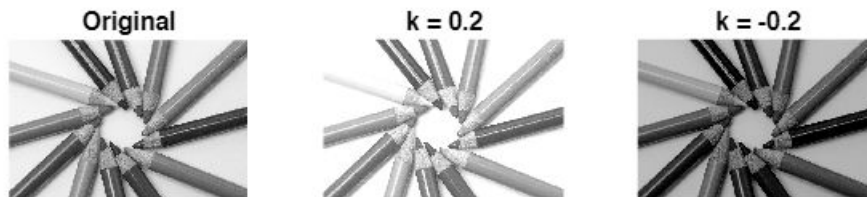
figure,
plot(r,r);
xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s1);
plot(r,s2);
legend('r','r+0.2','r-0.2','Location','northwest');
```



### Aplikace zmeny jasu

```
map1 = createMap(s1);
map2 = createMap(s2);

figure
subplot(1,3,1)
imshow(f)
title('Original')
subplot(1,3,2)
imshow(f,map1)
title(['k = ' num2str(k1)])
subplot(1,3,3)
imshow(f,map2)
title(['k = ' num2str(k2)]);
```



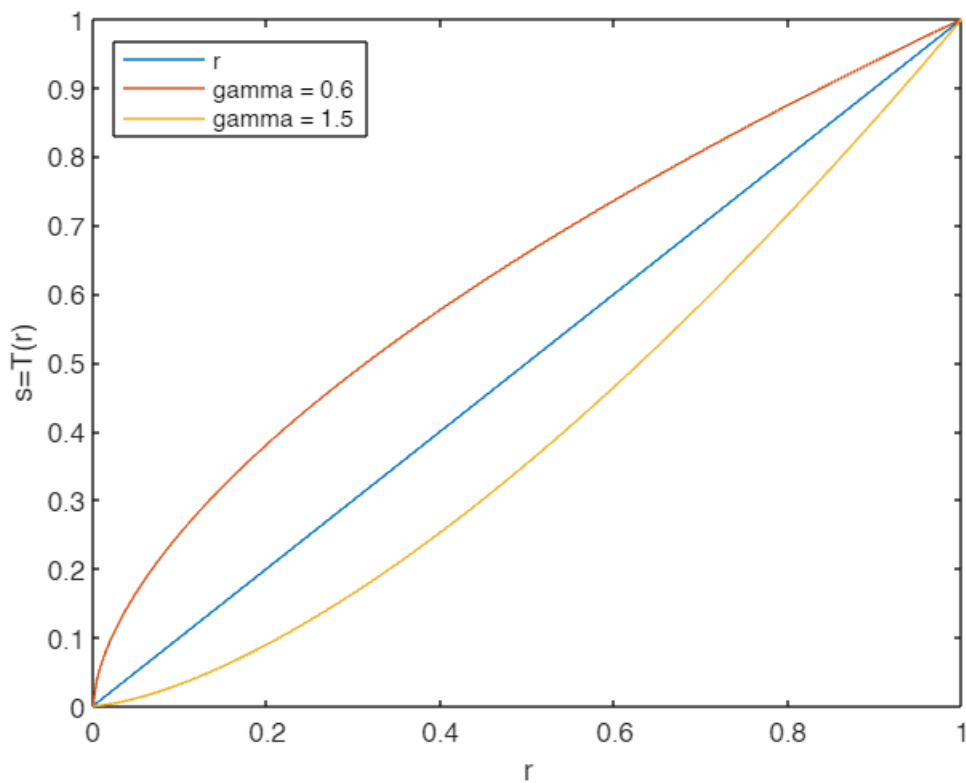
## Gamma transformace

```
f=imread('prechod.png');
c=1;
Gamma1=0.6;
Gamma2=1.5;

%x1=double(x);
% y=c*(x1.^Gamma);
% s=c*(r.^ ?)

s7 = clipMap(c*(r.^Gamma1),0,1);
s8 = clipMap(c*(r.^Gamma2),0,1);

figure,
plot(r,r);
xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s7);
plot(r,s8);
legend('r', ['gamma = ' num2str(Gamma1)], ['gamma = '
num2str(Gamma2)], 'Location', 'northwest');
```



### Aplikace na obrazek

```
map7 = createMap(s7);
map8 = createMap(s8);

figure
subplot(1,3,1)
imshow(f)
title('Original')
subplot(1,3,2)
imshow(f,map7)
title(['gamma = ' num2str(Gamma1)])
subplot(1,3,3)
imshow(f,map8)
title(['gamma = ' num2str(Gamma2)])
```

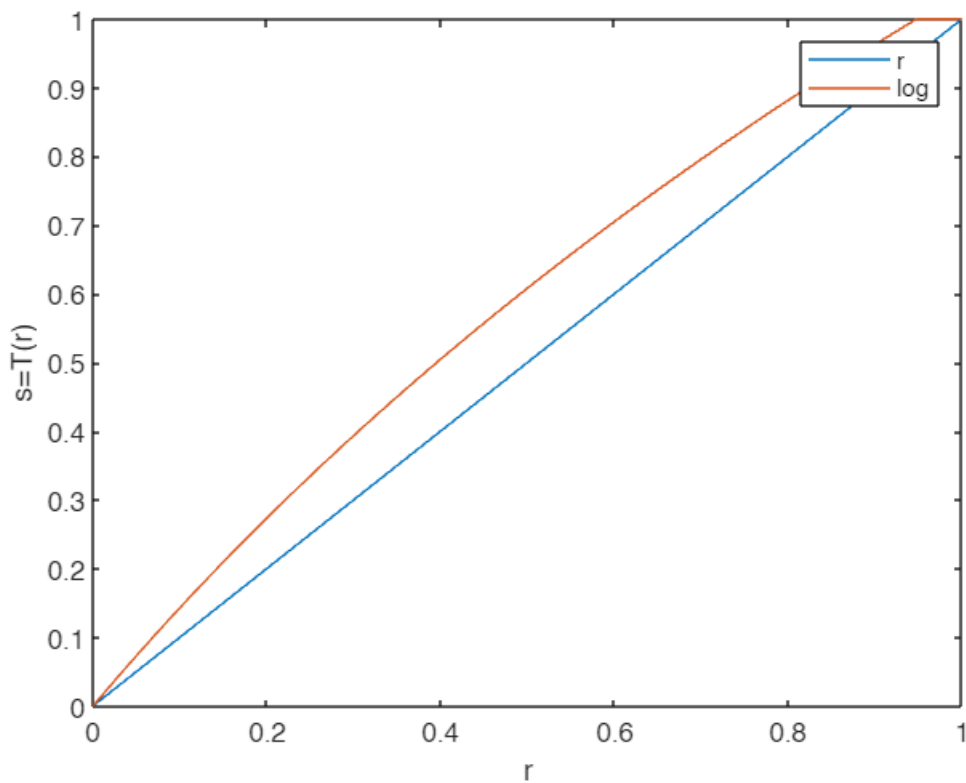


## Logaritmická transformace

```
c = 1.5;
s6 = clipMap(c*log(1+r),0,1);

figure,
plot(r,r);
xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s6);
legend('r','log','Location','northeast');
```



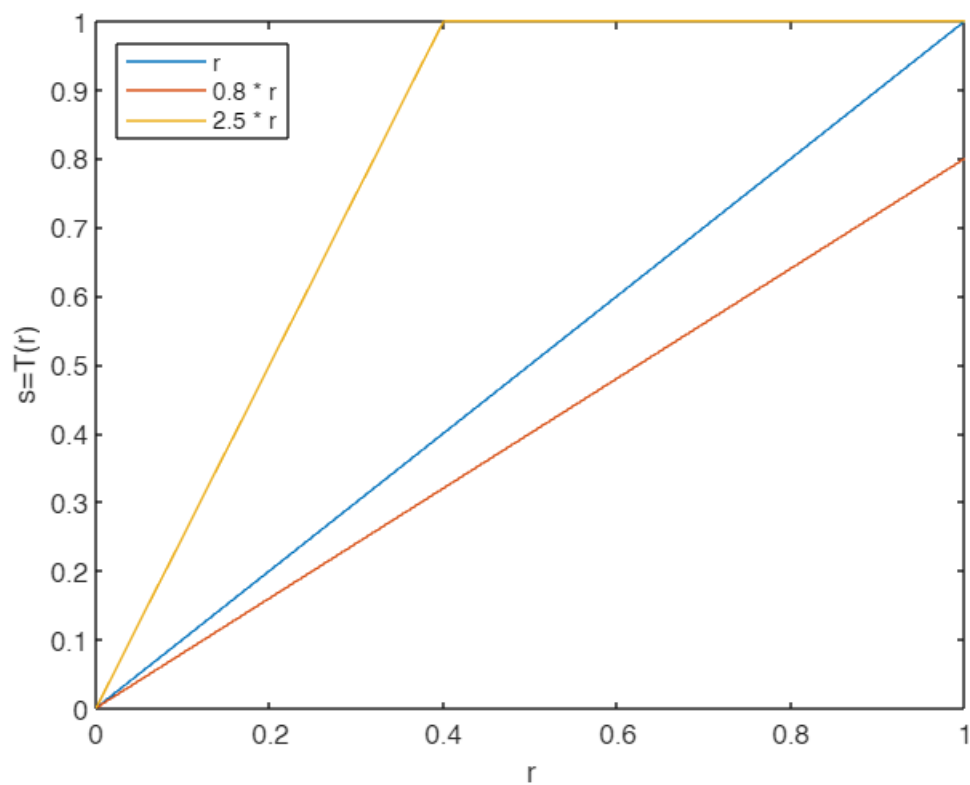


## Změna kontrastu

```
f = imread('pastelky_gray.png');
c1 = 0.8;
c2 = 2.5;

s3 = clipMap(c1*r,0,1);
s4 = clipMap(c2*r,0,1);

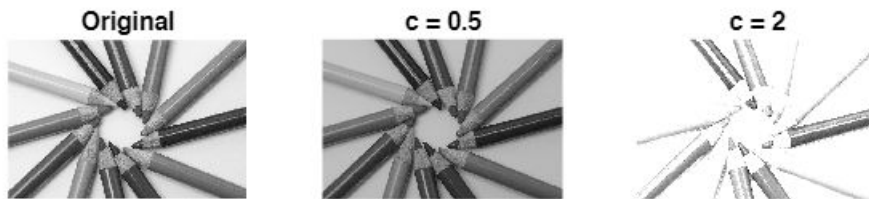
figure,
plot(r,r);
xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s3);
plot(r,s4);
legend('r',[num2str(c1) ' * r'],[num2str(c2) ' * r'],'Location','northwest');
```



### Aplikace změny kontrastu

```
map3 = createMap(s3);
map4 = createMap(s4);
```

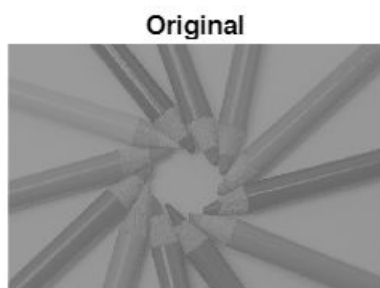
```
figure
subplot(1,3,1)
imshow(f)
title('Original')
subplot(1,3,2)
imshow(f,map3)
title('c = 0.5')
subplot(1,3,3)
imshow(f,map4)
title('c = 2')
```



## Roztažení kontrastu

```
%imadjust
I = imread('pastelky_lc.png');
J = imadjust(I);

figure
subplot(1,2,1)
imshow(I);
title('Original')
subplot(1,2,2)
imshow(J);
title('imadjust')
```

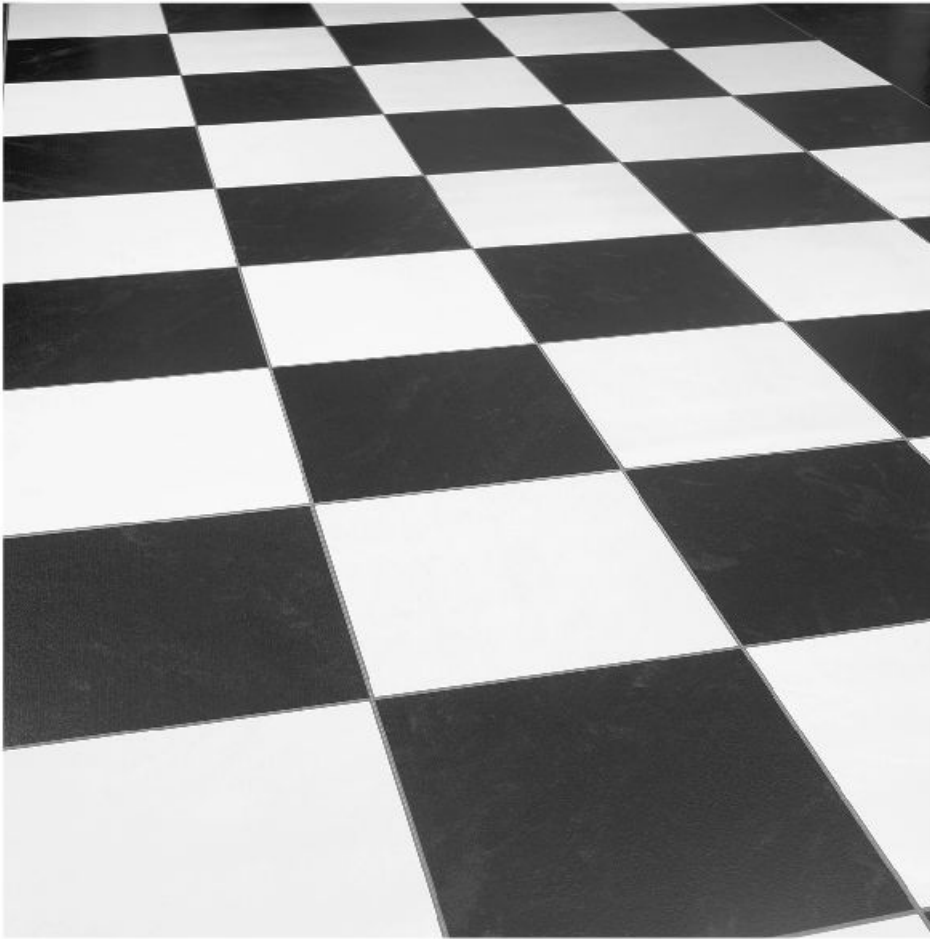


## Úkol 1

*V obrázku je nejnížší jasová hodnota rovna 100 a nejvyšší 156. Jak by vypadala funkce pro úpravu kontrastu, aby ve výsledném obrázku byla nejnížší hodnota rovna 0 a nejvyšší 255?*

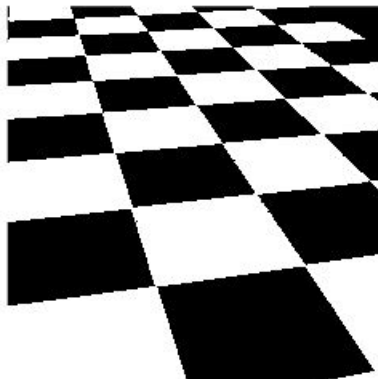
## Prahování

```
I = rgb2gray(imread('chess.jpg'));  
  
figure,  
imshow(I);
```



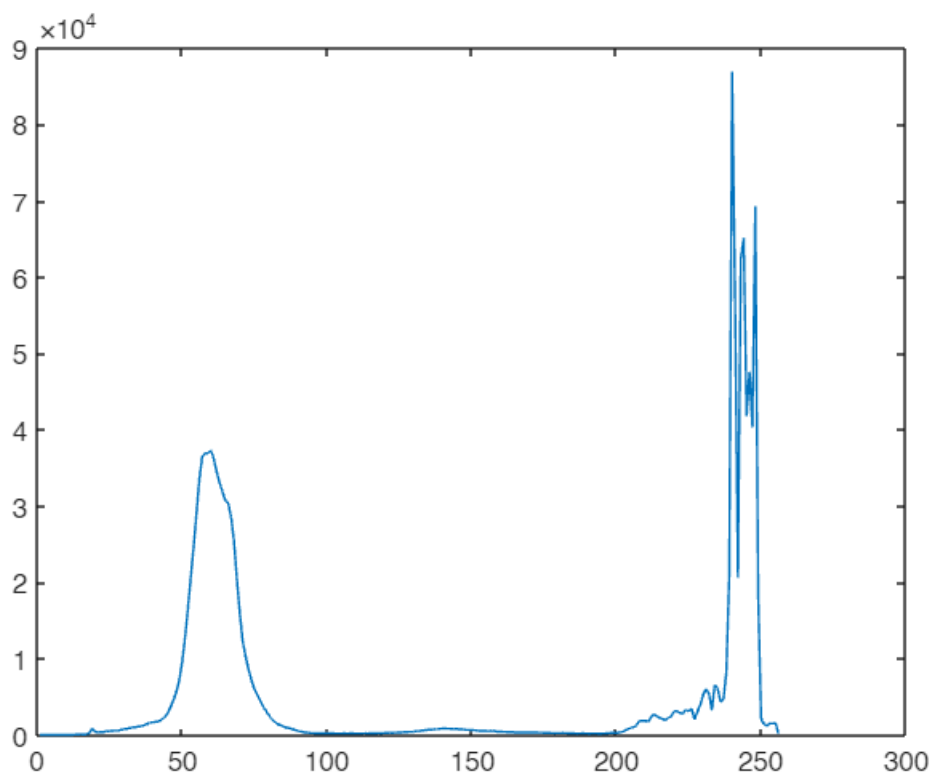
### Experimentální výběr prahu

```
prah = 120;  
J = I>prah;  
  
figure,  
subplot(1,2,1)  
imshow(I);  
subplot(1,2,2)  
imshow(J);
```



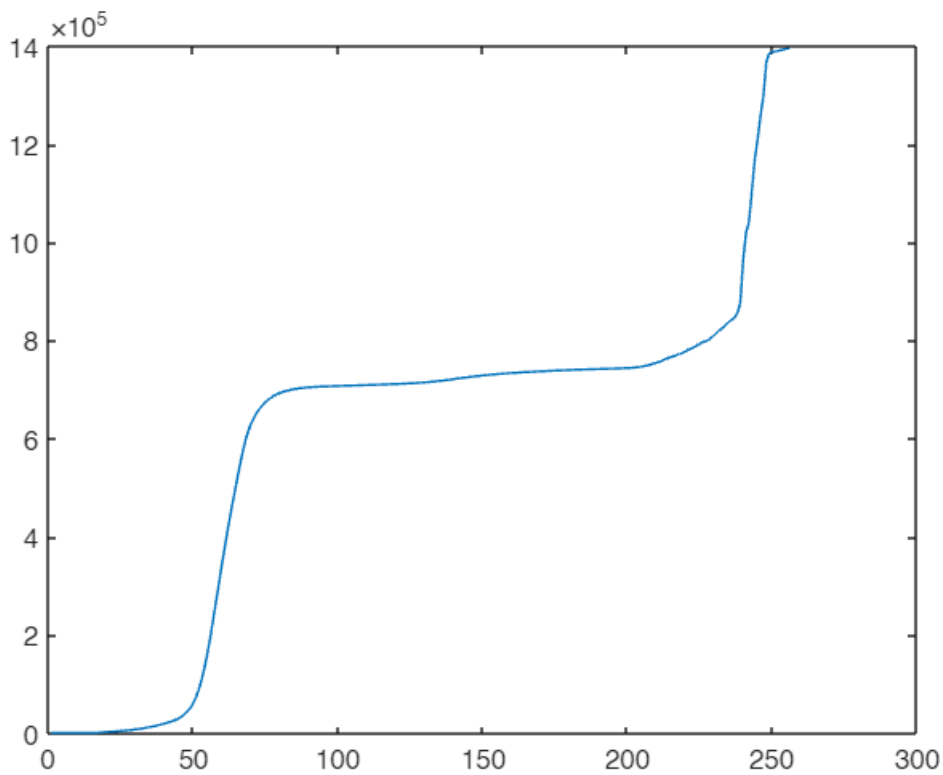
### Práh na základě znalosti histogramu

```
histogramI = imhist(I);  
  
figure,  
plot(histogramI);
```



### Kumulativní histogram

```
[pocet,X] = imhist(I);  
cumh = cumsum(pocet);  
  
figure,  
plot(cumh);
```



### Práh %

```
procento = 50
```

```
procento = 50
```

```
velikost = size(I,1) * size(I,2);
prah = round(velikost*(procento/100));

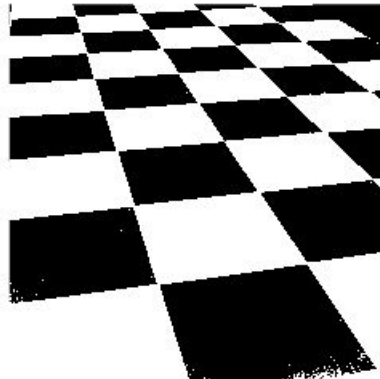
prah_index = find(cumh>=prah,1,'first');
display(prah_index);
```

```
prah_index = 83
```

```
J = I>prah_index;
```

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





## Vícenásobné prahování

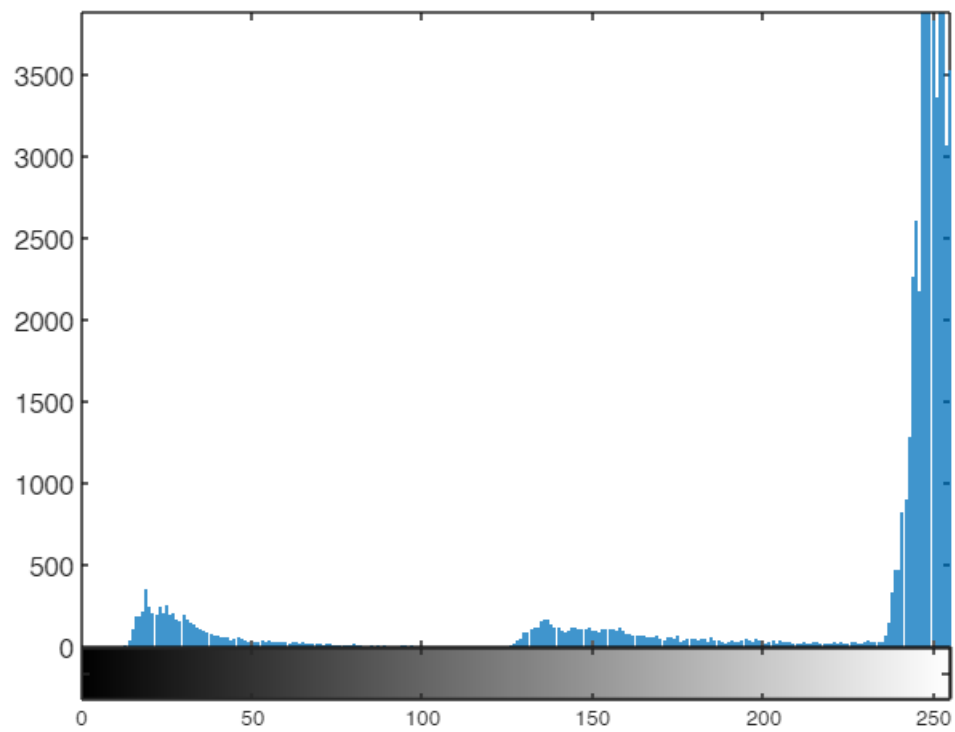
$g(x,y) = a$  pro  $f(x,y) > T_1$  (pozadí)

$= b$  pro  $f(x,y) \leq T_1$  a  $f(x,y) > T_2$  (objekt 1)

$= c$  pro  $f(x,y) \leq T_2$  (objekt 2)

většinou je  $a=1$ ,  $b = 0.5$  a  $c = 0$

```
f2 = rgb2gray(imread('figurka2.jpg'));  
figure, imhist(f2);
```



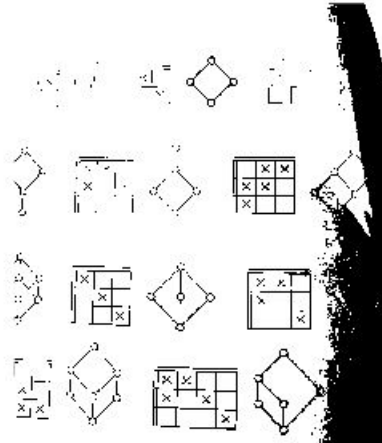
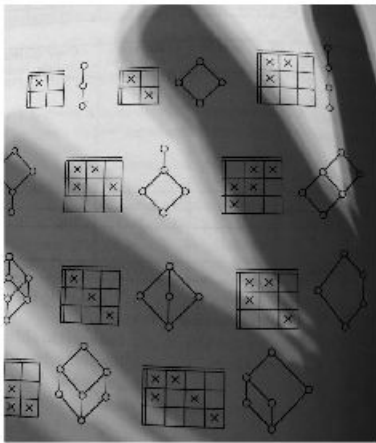
```
T1 = 220;  
T2 = 100;  
g2 = 0.5*(f2 >T1) + 0.5*(f2>T2);
```

```
figure,  
subplot(1,2,1)  
imshow(f2);  
subplot(1,2,2)  
imshow(g2);
```



## Lokální prahování

```
f3 = imread('lokalni.jpg');  
T = 46;  
  
g3 = f3 > T;  
  
figure,  
subplot(1,2,1), imshow(f3);  
subplot(1,2,2), imshow(g3);
```

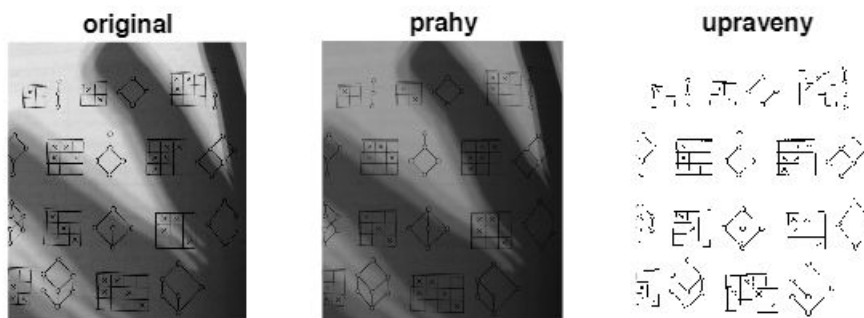


Pro každý bod počítáme prahovou hodnotu v závislosti na okolních bodech. Jak funguje výběr automatického prahu je nad rámec kurzu

```
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 = f4 > T;
subplot(1,3,1), imshow(f4);
title('original')
subplot(1,3,2), imshow(T);
title('prahy')
subplot(1,3,3), imshow(g4);
title('upraveny')
```



## Úkol 2

Jakým způsobem ovlivní následující transformace vzhled histogramu obrázku?

1. Změna jasu.
2. Změna kontrastu.
3. Negativ obrázku.
4. Gamma korekce s hodnotou  $< 1$ .
5. Gamma korekce s hodnotou  $> 1$ .

## Ořezávání hodnot intenzity

```
f = imread('pastelky_gray.png');
%J = I;
%J(J >= 140 & J<=240) = 255;

r1 = 0.5;
r2 = 0.9;

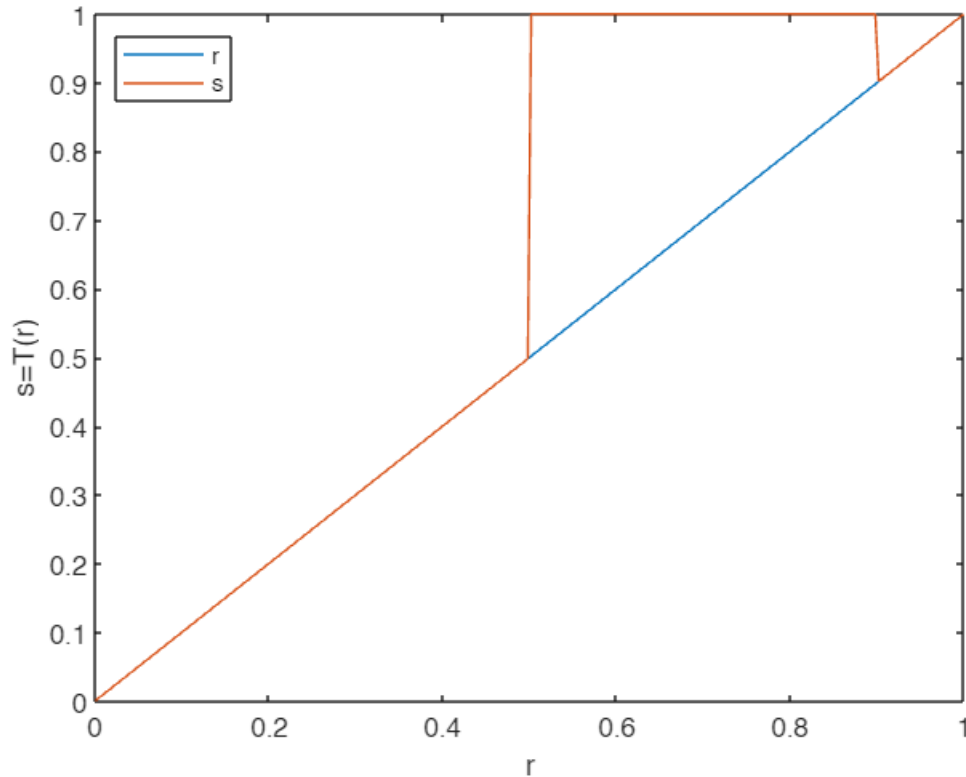
s13 = r;
s13(s13 >= r1 & s13<=r2) = 1;

figure,
plot(r,r);
```

```

xlabel('r');
ylabel('s=T(r)');
xlim([0,1]);
hold on;
plot(r,s13);
legend('r','s','Location','northwest');

```



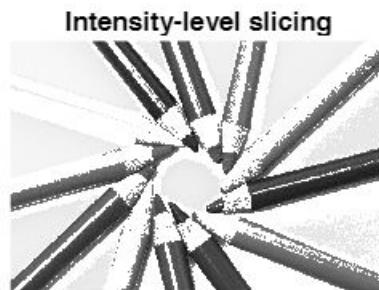
### Aplikace na obrázek

```

map13 = createMap(s13);

figure
subplot(1,2,1)
imshow(f);
title('Original')
subplot(1,2,2)
imshow(f,map13);
title('Intensity-level slicing')

```



## Ořezávání bitů

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

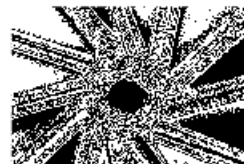
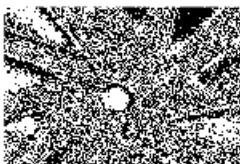
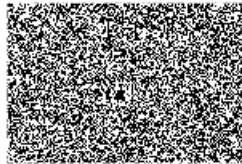
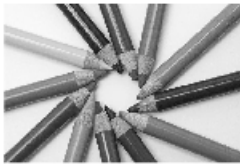
```
I1 = bitand(I,1);
I2 = bitand(I,2);
I3 = bitand(I,4);
I4 = bitand(I,8);
I5 = bitand(I,16);
I6 = bitand(I,32);
I7 = bitand(I,64);
I8 = bitand(I,128);
```

```
figure
subplot(3,3,1)
imshow(I,[])
subplot(3,3,2)
imshow(I1,[])
subplot(3,3,3)
imshow(I2,[])
subplot(3,3,4)
imshow(I3,[])
subplot(3,3,5)
imshow(I4,[])
subplot(3,3,6)
```

```

imshow(I5,[])
subplot(3,3,7)
imshow(I6,[])
subplot(3,3,8)
imshow(I7,[])
subplot(3,3,9)
imshow(I8,[])

```



### Skládání obrázků

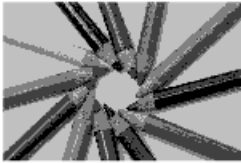
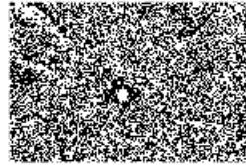
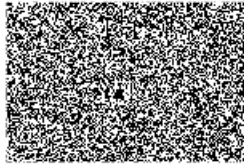
```

J1 = I8 + I7;
J2 = I8 + I7 + I6;
J3 = I8 + I7 + I6 + I5;

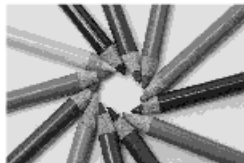
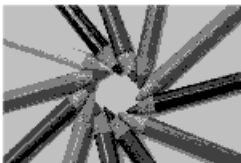
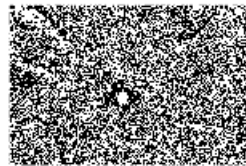
subplot(1,3,1)
imshow(J1);

```

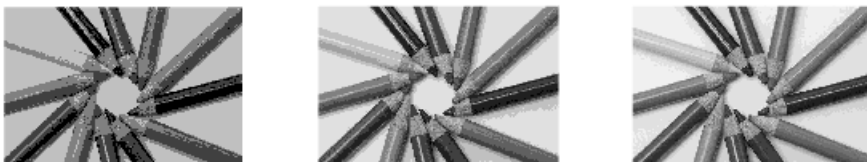




```
subplot(1,3,2)  
imshow(J2);
```



```
subplot(1,3,3)
imshow(J3);
```



## Úkol 3

Jakým způsobem ovlivní následující transformace vzhled histogramu obrázku?

1. Nastavení nejméně významného bitu u všech pixelů na 0.
2. Nastavení nejvíce významného bitu u všech pixelů na 0.

## Vyrovňávání histogramu

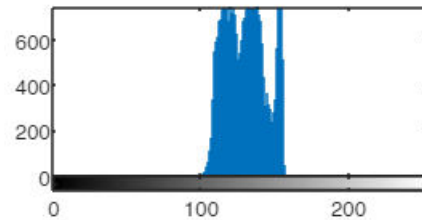
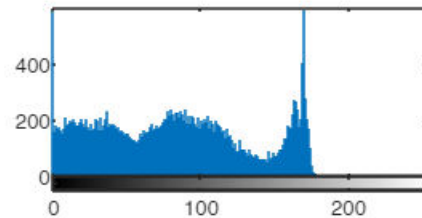
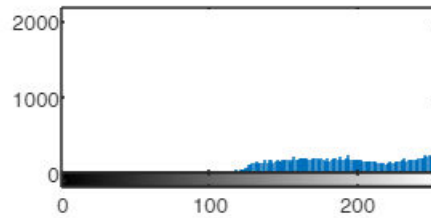
```
I1 = imread('p1.png');
I2 = imread('p2.png');
I3 = imread('p3.png');
```

```
figure,
subplot(3,2,1)
imshow(I1);
subplot(3,2,2)
imhist(I1)
```

```
subplot(3,2,3)
imshow(I2);
subplot(3,2,4)
```

```
imhist(I2)

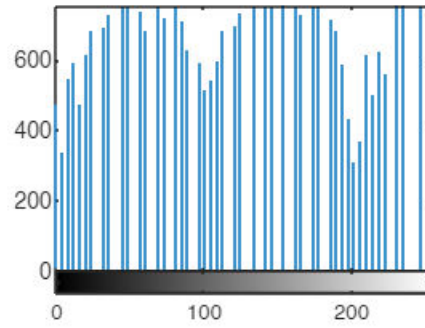
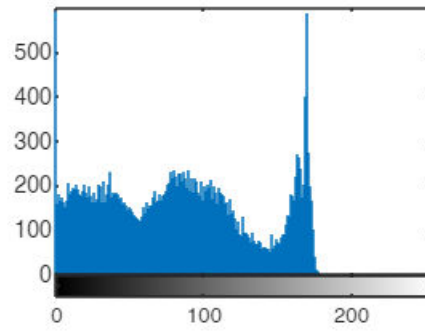
subplot(3,2,5)
imshow(I3);
subplot(3,2,6)
imhist(I3)
```



## Obrázek

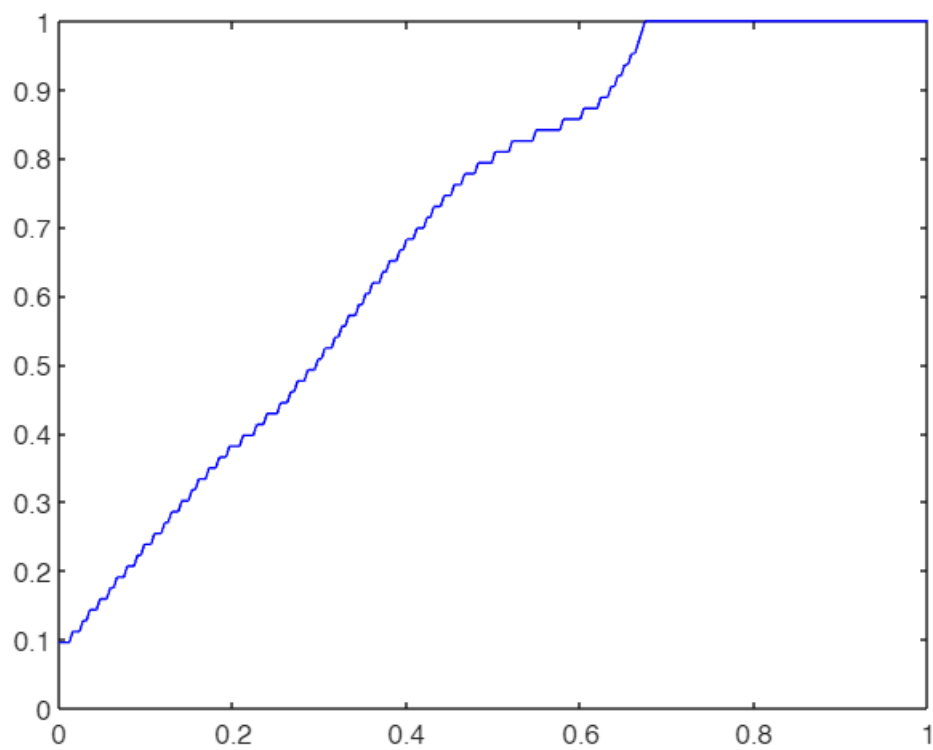
```
I = I2;
% J = histeq(I);
[J2,T] = histeq(im2double(I));

figure,
subplot(2,2,1)
imshow(I);
subplot(2,2,2)
imhist(I)
subplot(2,2,3)
imshow(J1);
subplot(2,2,4)
imhist(J1)
```



Transformační funkce

```
figure,  
plot(linspace(0,1,256), T, 'b-');
```

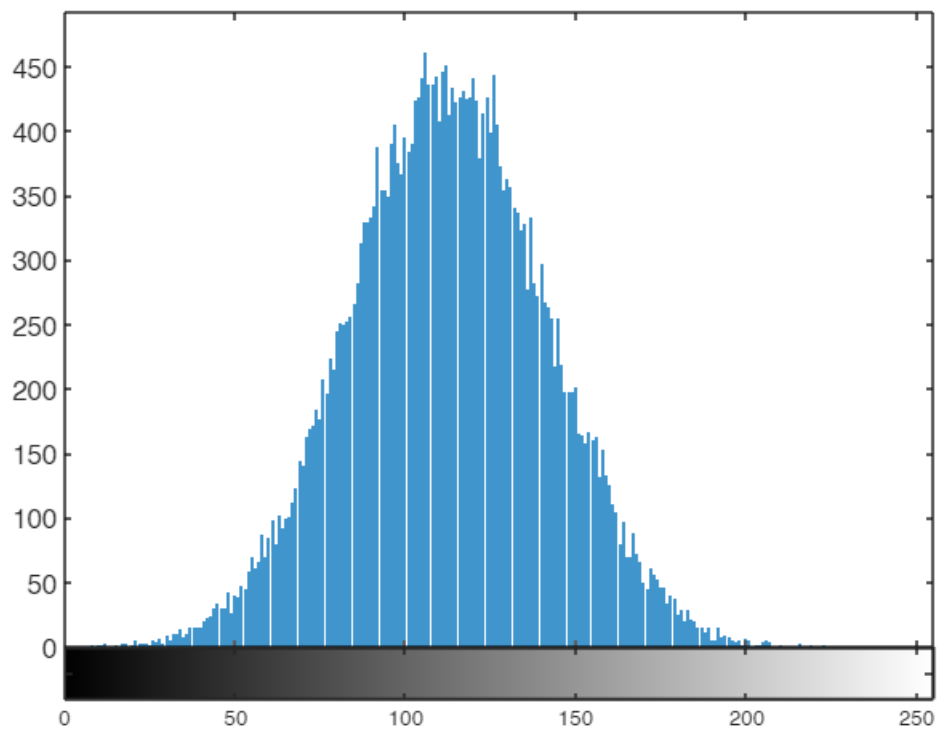


## Specifikace histogramu

Vytvoření histogramu

```
I = I2;  
spechistobr = randn(size(I));  
minimum = min(min(spechistobr));  
spechistobr = spechistobr + (0-minimum);  
maximum = max(max(spechistobr));  
spechistobr = 255*(spechistobr/maximum);  
spechistobr = uint8(round(spechistobr));
```

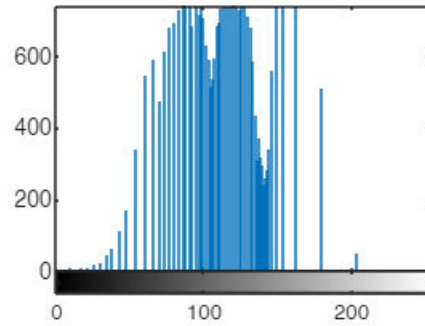
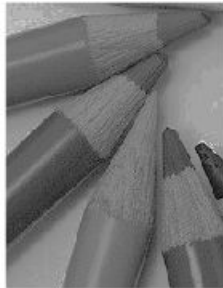
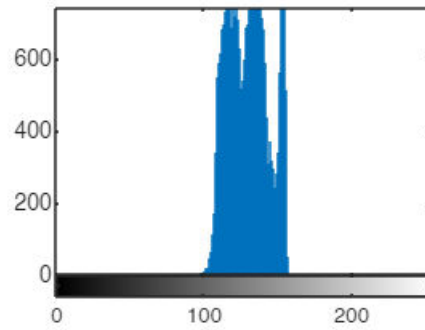
```
figure,  
imhist(spechistobr)
```



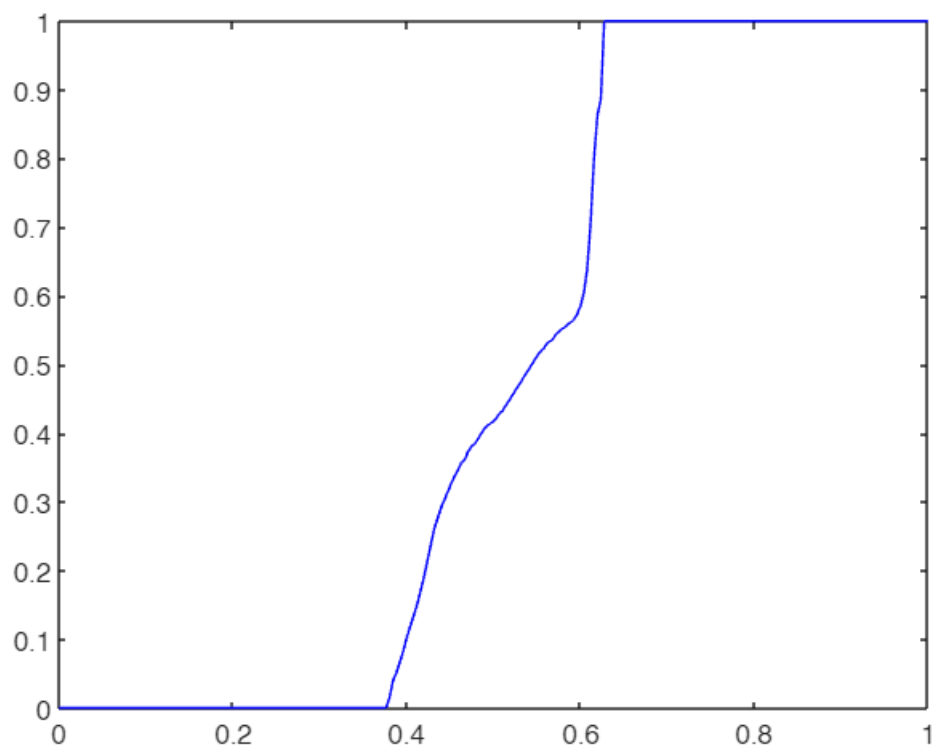
### Specifikace histogramu

```
[COUNTS,X] = imhist(spechistobr);
[Jspec, T] = histeq(I, COUNTS);
```

```
figure,
subplot(2,2,1)
imshow(I);
subplot(2,2,2)
imhist(I)
subplot(2,2,3)
imshow(Jspec);
subplot(2,2,4)
imhist(Jspec)
```



```
figure,  
plot(linspace(0,1,256), T, 'b-');
```



## Barevné obrázky

### Změna jasu

```
f_rgb = im2double(imread('pastelky.png'));
f_hsv = rgb2hsv(f_rgb);
c = 0.4;
g_rgb = f_rgb;
g_hsv = f_hsv;

g_rgb(:,:,1) = g_rgb(:,:,1) + c;
g_rgb(:,:,2) = g_rgb(:,:,2) + c;
g_rgb(:,:,3) = g_rgb(:,:,3) + c;

g_hsv(:,:,3) = f_hsv(:,:,3) + c;

figure,
subplot(1,3,1)
imshow(f_rgb);
title('original')
subplot(1,3,2)
imshow(g_rgb)
title('RGB')
subplot(1,3,3)
imshow(hsv2rgb(g_hsv));
```

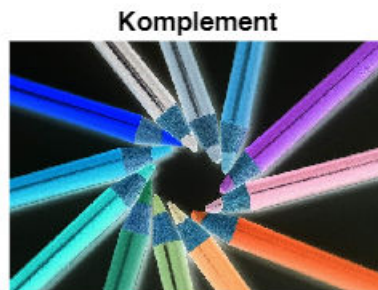


```
title('HSV')
```



## Barevný komplement

```
f_komplement = 1-f_rgb;  
  
figure,  
subplot(1,2,1)  
imshow(f_rgb);  
title('original')  
subplot(1,2,2)  
imshow(f_komplement);  
title('Komplement')
```



## Tónování

```
Gamma = 1.8;  
f_ton = f_rgb.^Gamma;  
  
figure,  
subplot(1,2,1)  
imshow(f_rgb);  
title('original')  
subplot(1,2,2)  
imshow(f_ton);  
title(['Tonovani gamma = ' num2str(Gamma)]);
```



## Barevná korekce

```
slozka = 3;  
Gamma = 0.4;  
  
f_kor = f_rgb;  
f_kor(:, :, slozka) = f_kor(:, :, slozka).^Gamma;  
  
figure,  
subplot(1,2,1)  
imshow(f_rgb);  
title('original')  
subplot(1,2,2)  
imshow(f_kor);  
title(['Barevna korekce gamma = ' num2str(Gamma)]);
```



## Vyrovňání histogramu

```
g_hsv = f_hsv;

g_rgb = histeq(f_rgb);

g_hsv(:, :, 3) = histeq(f_hsv(:, :, 3));

figure,
subplot(1,3,1), imshow(g_rgb);
title('RGB');
subplot(1,3,2), imshow(hsv2rgb(g_hsv));
title('HSV');
subplot(1,3,3), imshow(im2double(g_rgb)-hsv2rgb(g_hsv));
title('rozdíl');
```



## Maticové úpravy

```
% sepia
M_sepia = [0.393 0.349 0.272 0 0;
0.769 0.686 0.534 0 0;
0.189 0.168 0.131 0 0;
0 0 0 1 0;
0 0 0 0 1];

% swap
M_swap = [0 0 1 0 0;
0 1 0 0 0;
1 0 0 0 0;
0 0 0 1 0;
0 0 0 0 1];

% černá a bílá
M_bw = [1.5 1.5 1.5 0 0;
1.5 1.5 1.5 0 0;
1.5 1.5 1.5 0 0;
0 0 0 1 0;
-1 -1 -1 0 1];

% polaroid
M_polar = [1.438 -0.062 -0.062 0 0;
```

```

-0.122 1.378 -0.122 0 0;
-0.016 -0.016 1.483 0 0;
0 0 0 1 0;
-0.03 0.05 -0.02 0 1];

% červená složka
M_red = [1.5 0 0 0 0;
0 0.5 0 0 0;
0 0 0.5 0 0;
0 0 0 1 0;
0 0 0 0 1];

% Vyber efektu
M = M_sepia;

im = double(imread('pastelky.png'));

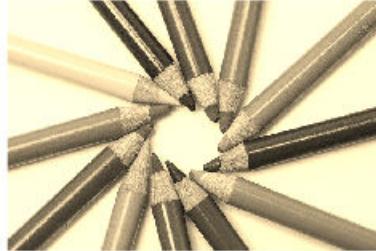
inputRed = im(:,:,1); %// barevná složka
inputGreen = im(:,:,2);
inputBlue = im(:,:,3);

% Aplikace na jednotlivé složky
outputRed = (inputRed * M(1,1)) + (inputGreen * M(2,1)) + (inputBlue * M(3,1)) +
M(5,1);
outputGreen = (inputRed * M(1,2)) + (inputGreen * M(2,2)) + (inputBlue * M(3,2)) +
M(5,2);
outputBlue = (inputRed * M(1,3)) + (inputGreen * M(2,3)) + (inputBlue * M(3,3)) +
M(5,3);

% Sjednocení složek
out = uint8(cat(3, outputRed, outputGreen, outputBlue));

figure,
subplot(1,2,1), imshow(uint8(im));
subplot(1,2,2), imshow(out);

```



## Color slicing

```
im = imread('pastelky.png');
im = im2double(im);
W = 0.2;
R = 0.1;
a = [0.81, 0.08, 0.11];

[m,n,o] = size(im);
out1 = im;
out2 = im;

c = zeros(1,3);
for i = 1 : m
    for j = 1 : n
        c(1) = im(i,j,1);
        c(2) = im(i,j,2);
        c(3) = im(i,j,3);
        if(any((abs(c-a))>(W/2)))
            %(abs(im(i,j,:)-a)>W/2
            out1(i,j,:) = [0.5,0.5,0.5];
        end
        if(sum((c-a).^2) > R^2)
            out2(i,j,:) = [0.5,0.5,0.5];
        end
    end
end
```

```
        end
    end

    %unique(out1)
    figure,
    subplot(1,3,1)
    imshow(im);
    title('original')
    subplot(1,3,2)
    imshow(out1)
    subplot(1,3,3)
    imshow(out2);
```