

Cvičení 6 - Filtrování II

Filtrace

Pro připomenutí, filtrování v prostorové doméně je realizováno korelací (konvolucí). V Matlabu k tomu slouží funkce `imfilter()`.

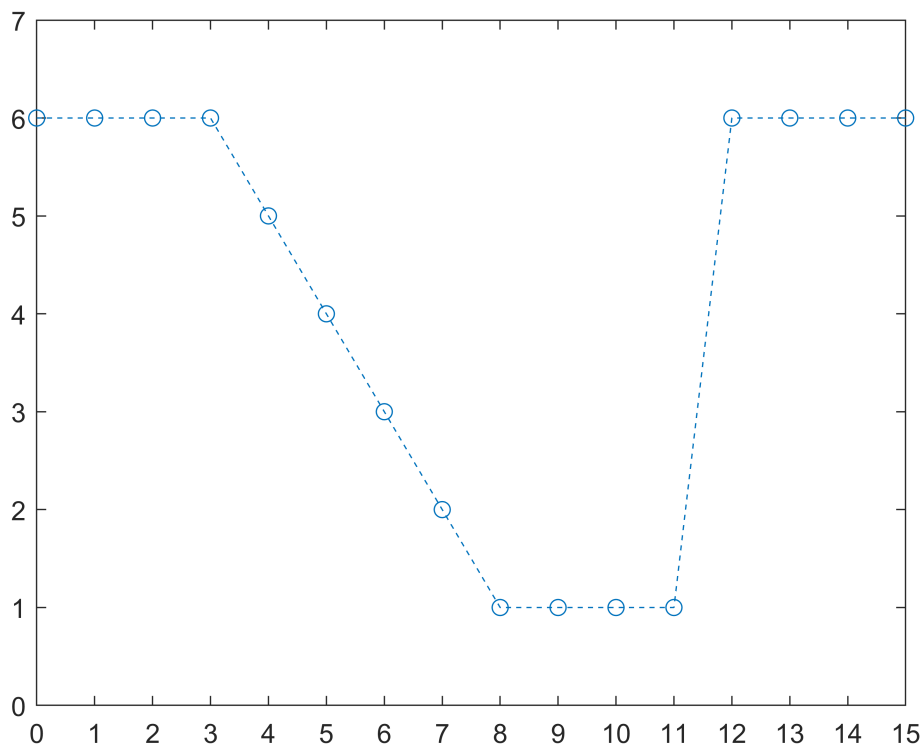
```
f = [ 0 0 0 0 0;
      0 0 0 0 0;
      0 0 1 0 0;
      0 0 0 0 0;
      0 0 0 0 0];
w = [ 1 2 3;
      4 5 6;
      7 8 9];
g = imfilter(f,w,'corr','same')
```

```
g = 5x5
    0     0     0     0     0
    0     9     8     7     0
    0     6     5     4     0
    0     3     2     1     0
    0     0     0     0     0
```

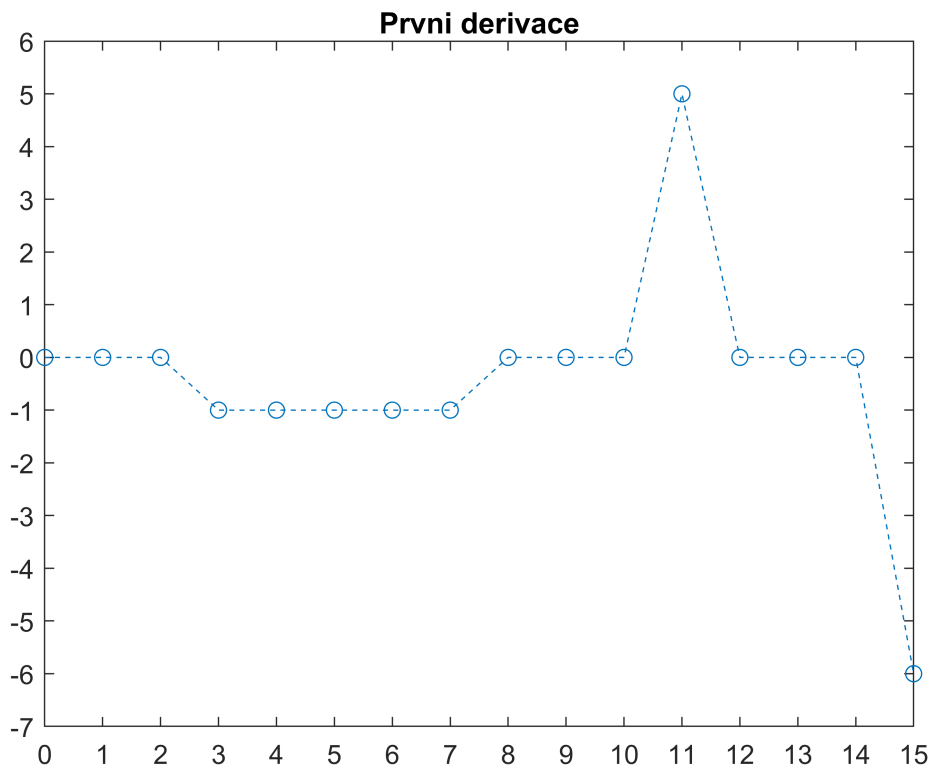
Ostřicí filtry

Principem ostřících filtrů je zvýraznění přechodu mezi intenzitami. Přechody se hledají pomocí derivací (první nebo druhé). Vlastnosti jednotlivých derivací (jejich aproximací znáte z přednášky).

```
data = [6 6 6 6 5 4 3 2 1 1 1 1 6 6 6 6];
figure, plot(0:size(data,2)-1, data, 'o--');
axis([0 15 0 7])
xticks([0 : 15])
```



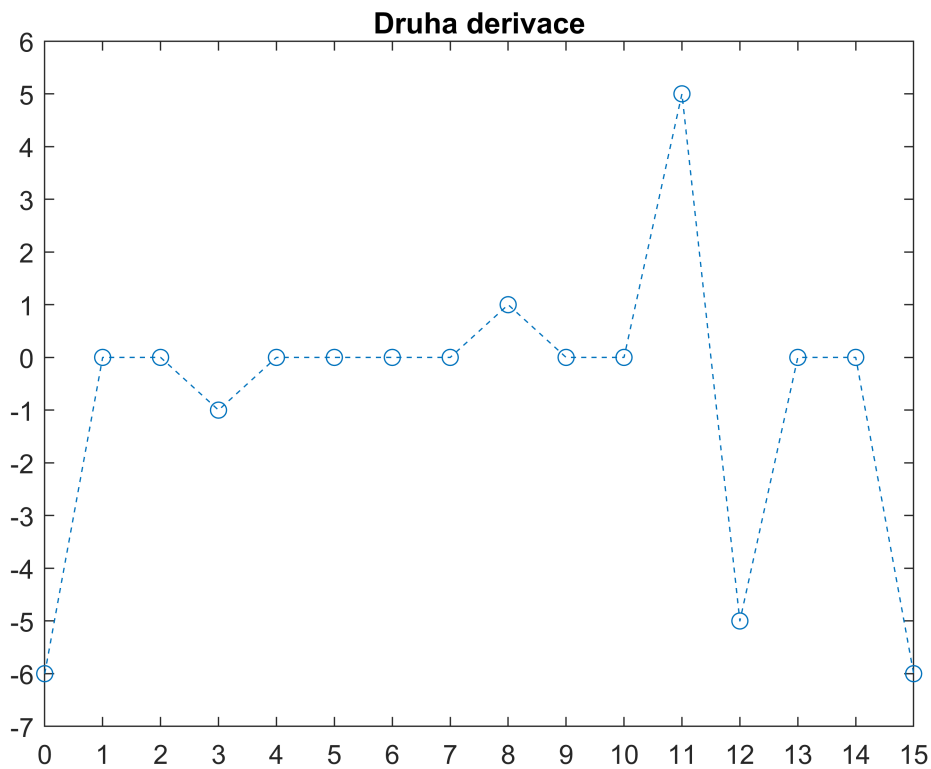
```
% prvni derivace f(x + 1) - f(x)
f = [-1 1];
data2 = imfilter(data,f,'corr','same');
figure, plot(0:size(data2,2)-1, data2, 'o--');
axis([0 15 -7 6])
xticks([0 : 15])
yticks([-7 : 6])
title('Prvni derivace');
```



```

% prvni derivace  $f(x + 1) + f(x-1) - 2f(x)$ 
f = [1 -2 1];
data3 = imfilter(data,f,'corr','same');
figure, plot(0:size(data3,2)-1, data3, 'o--');
axis([0 15 -7 6])
xticks([0 : 15])
yticks([-7 : 6])
title('Druha derivace');

```



1. derivace

Několik různých operátorů. Operátory sami o sobě hledají jen hrany. Ke zvýraznění hran v obraze je potřeba výsledný obraz přičíst k původnímu obrázku.

Robertsův

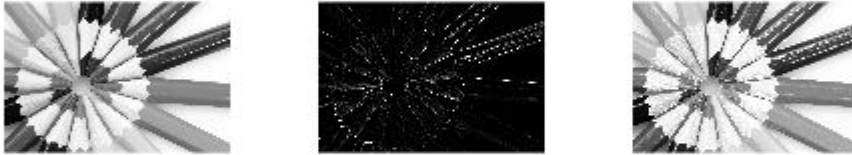
```
I = imread('pastelkygray.jpg');
% detekuje hlavne diagonalni hrany
w = [-1 -1; 1 1];
I2 = imfilter(I,w,'corr','same');
```

```
figure
subplot(1,3,1)
imshow(I,[])
subplot(1,3,2)
imshow(I2,[])
subplot(1,3,3)
imshow(I + I2)
```



Sobelův

```
I = imread('pastelkygray.jpg');  
  
% detekuje vodorovne hrany  
% w = fspecial('sobel')  
w = [1 2 1;  
      0 0 0;  
     -1 -2 -1];  
I2 = imfilter(I,w,'corr','same');  
  
figure  
subplot(1,3,1)  
imshow(I,[])  
subplot(1,3,2)  
imshow(I2,[])  
subplot(1,3,3)  
imshow(I + I2)
```

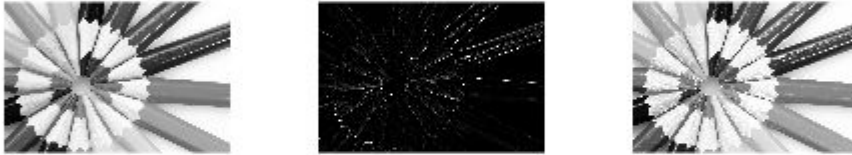


ÚKOL 1

Jak by vypadal sobelův operátor pro svislé a diagonální hrany?

Prewitové

```
I = imread('pastelkygray.jpg');  
  
% detekuje hlavne vodorovne hrany  
% w = fspecial('prewitt')  
w = [1 1 1;  
      0 0 0;  
     -1 -1 -1];  
I2 = imfilter(I,w,'corr','same');  
  
figure  
subplot(1,3,1)  
imshow(I,[])  
subplot(1,3,2)  
imshow(I2,[])  
subplot(1,3,3)  
imshow(I + I2)
```



ÚKOL 2

Jak by vypadal Prewitové operátor pro svislé a diagonální hrany?

2. derivace

Laplaceův

```
I = imread('pastelkygrayblurred.png');  
  
% zvyraznujici vodorovne, svisle i diagonalni hrany  
w = -[-1 -1 -1;  
      -1 8 -1;  
      -1 -1 -1];  
  
% zvyraznujici vodorovne a svisle hrany  
% w = fspecial('laplacian',0)  
w2 = -[0 -1 0;  
       -1 4 -1;  
       0 -1 0];  
  
C = imfilter(I,w,'corr','same');  
C2 = imfilter(I,w2,'corr','same');  
  
figure,  
subplot(2,3,1)
```

```

imshow(I,[]) % puvodni obrazek
subplot(2,3,2)
imshow(C,[]) % vysledek po aplikaci filtru
title('Vodorovne, svisle a diagonalni');
subplot(2,3,3)
imshow(I-C,[]) % vyuziti vysledku k ostreni

subplot(2,3,4)
imshow(I,[]) % puvodni obrazek
subplot(2,3,5)
imshow(C2,[]) % vysledek po aplikaci filtru
title('Vodorovne a svisle hrany');
subplot(2,3,6)
imshow(I-C2,[]) % vyuziti vysledku k ostreni

```



Upravený Laplaceův operátor, který i přičítá hodnotu původního obrázku.

```

w2 = -[0 -1 0;
      -1 3 -1;
       0 -1 0];

C2 = imfilter(I,w2,'corr','same');

figure
subplot(1,2,1)
imshow(I,[])
subplot(1,2,2)

```



```
imshow(C2,[])
```



Funkce fspecial

V předchozích příkladech byl zakomentován postup vytváření filtru pomocí funkce `fspecial()`. Je to funkce, která vrací matici pro zadaný typ filtru.

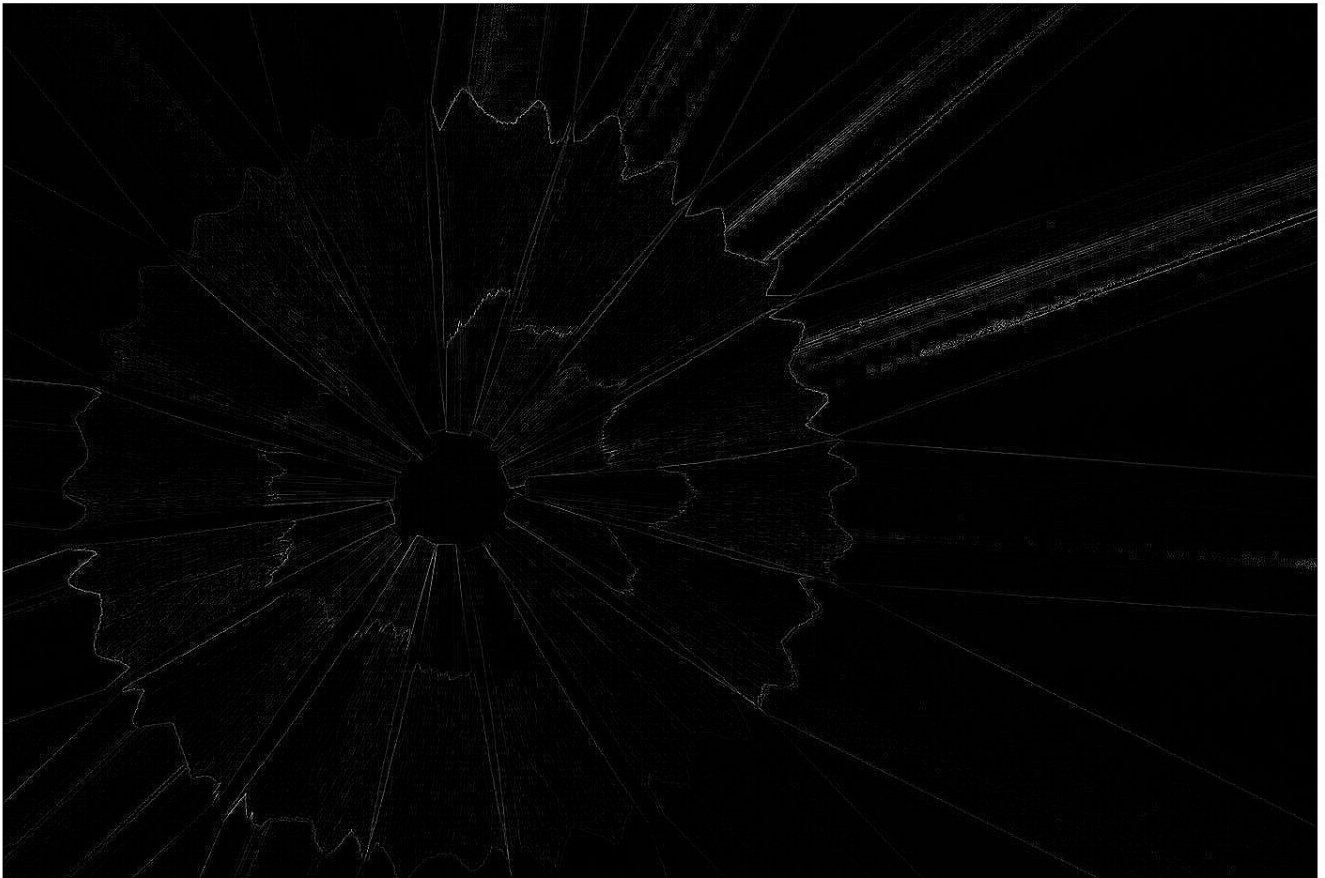
```
w = fspecial('typ','parametry');
```

Pro použití takto vytvořeného filtru musí být obrázek typu `double`. Pro převod používáme funkci `im2double()`.

```
w = fspecial('average',[5,5]);  
display(w); % vypise operator do command window
```

```
w = 5x5  
 0.0400    0.0400    0.0400    0.0400    0.0400  
 0.0400    0.0400    0.0400    0.0400    0.0400  
 0.0400    0.0400    0.0400    0.0400    0.0400  
 0.0400    0.0400    0.0400    0.0400    0.0400  
 0.0400    0.0400    0.0400    0.0400    0.0400
```

```
f = imread('pastelkygray.jpg');  
w = fspecial('laplacian',0);  
  
f2 = im2double(f);  
g2 = imfilter(f2,w);  
figure, imshow(g2);
```

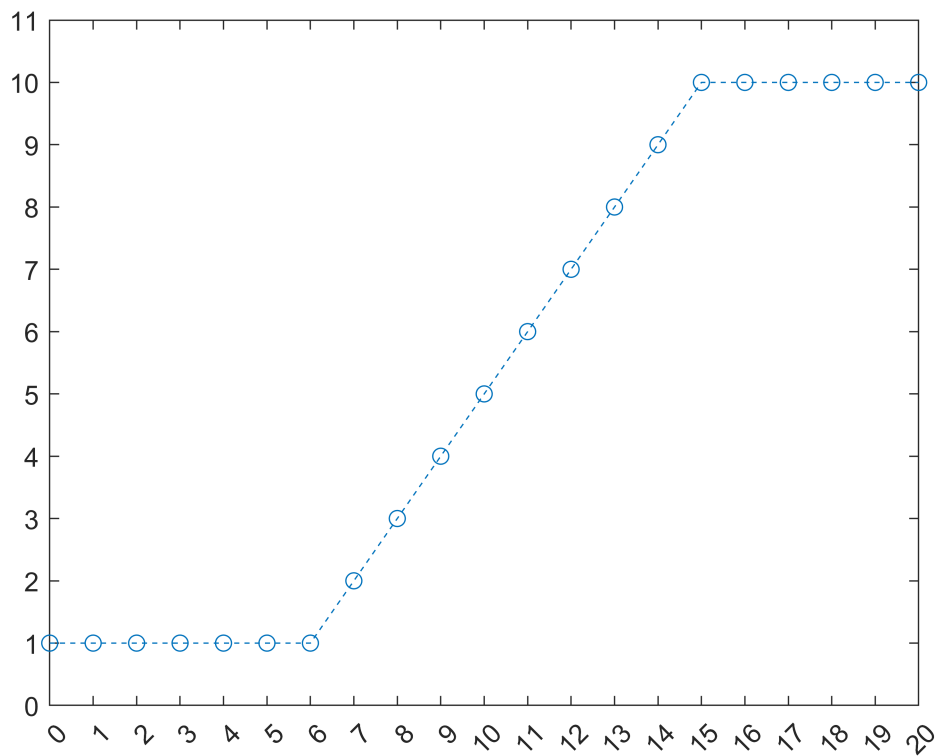


Jiný způsob ostření -- High boost ostření

Tento princip není založený ani na první ani na druhé derivaci.

Princip: Od původního obrázku odečteme rozmazaný obrázek (výsledek nazveme maskou). Tím získáme místa, kde jsou velké přechody. Přičtením nějakého násobku masky k původnímu obrázku dostaneme ostřejší obrázek.

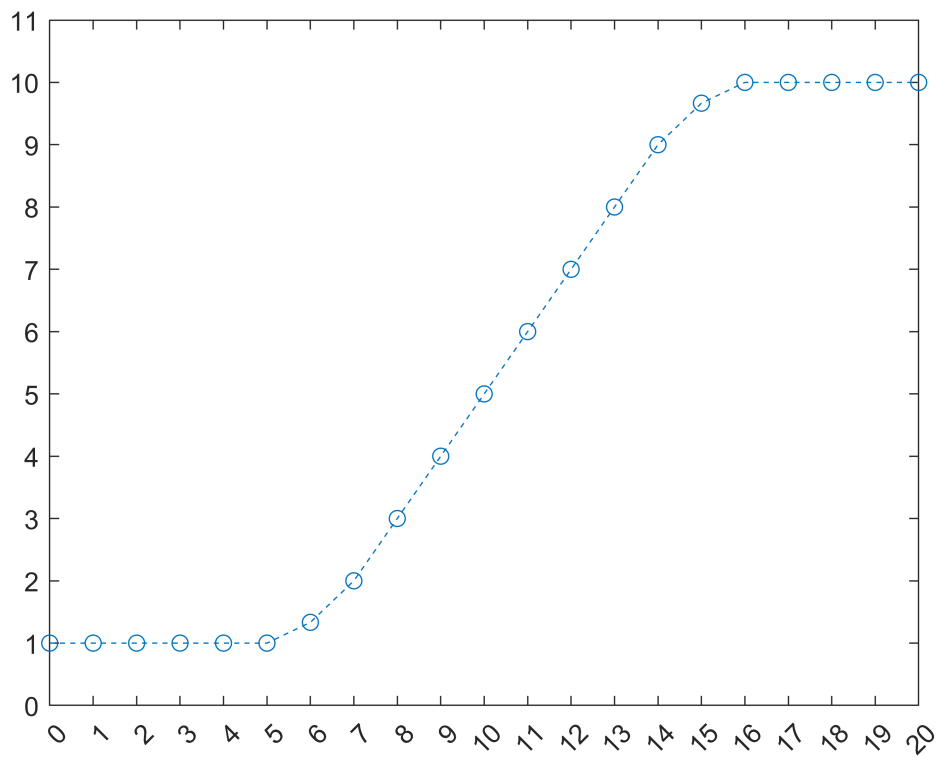
```
data = [1 1 1 1 1 1 1 2 3 4 5 6 7 8 9 10 10 10 10 10 10];  
figure, plot(0:size(data,2)-1, data, 'o--');  
axis([0 20 0 11])  
xticks([0 : 20])
```



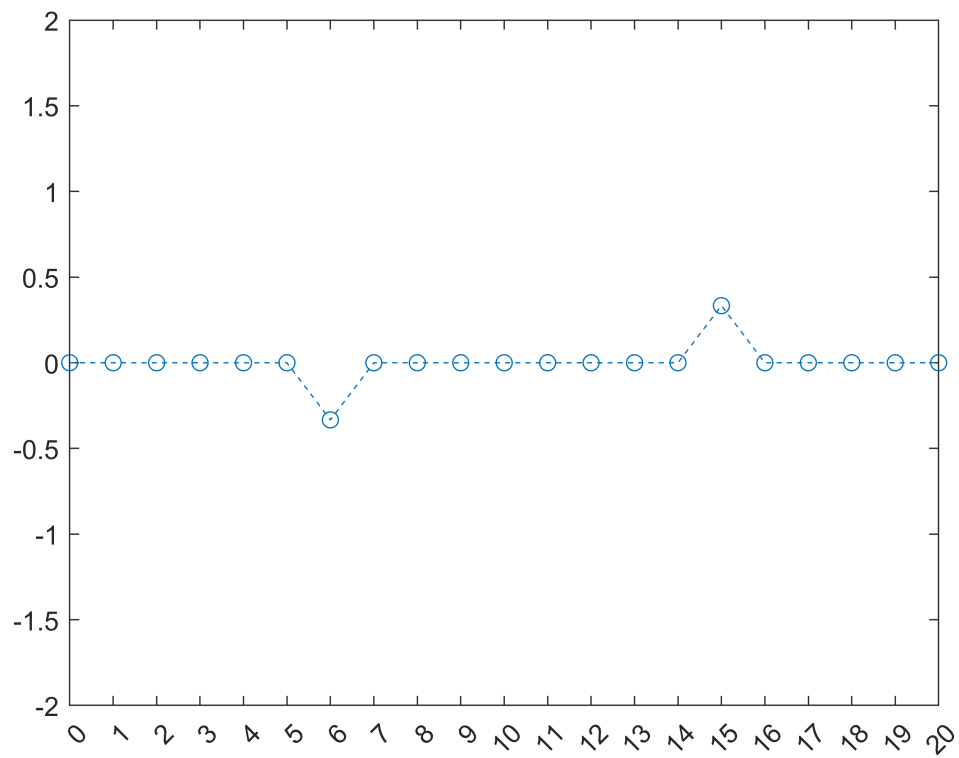
```

% rozmazana data
w = [1/3 1/3 1/3];
data2 = imfilter(data,w,'replicate');
figure, plot(0:size(data2,2)-1, data2, 'o--');
axis([0 20 0 11])
xticks([0 : 20])

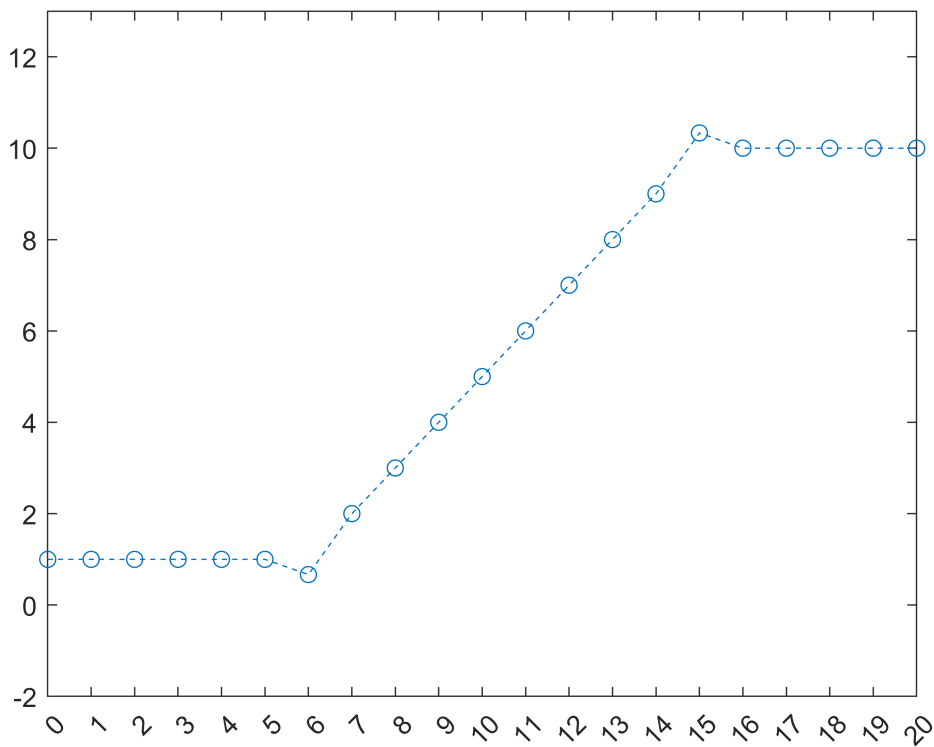
```



```
% maska  
maska = data - data2;  
figure, plot(0:size(maska,2)-1, maska, 'o--');  
axis([0 20 -2 2])  
xticks([0 : 20])
```



```
% zvyrazneny prechod  
data3 = data + maska;  
figure, plot(0:size(data3,2)-1, data3, 'o--');  
axis([0 20 -2 13])  
xticks([0 : 20])
```



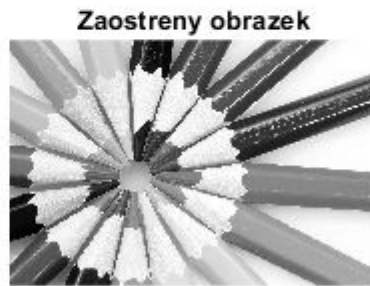
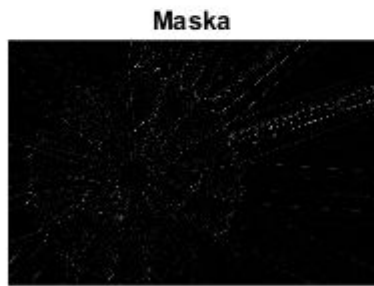
```

I = imread('pastelkygray.jpg');
Iblur = imread('pastelkygrayblurred.png');

mask = I - Iblur;
Isharp = I + 2* mask;

figure
subplot(2,2,1)
imshow(I,[])
title('Puvodni obrazek');
subplot(2,2,2)
imshow(Iblur,[])
title('Rozostreny obrazek');
subplot(2,2,3)
imshow(mask,[])
title('Maska');
subplot(2,2,4)
imshow(Isharp,[])
title('Zaostreny obrazek');

```



ÚKOL 3

V předchozím příkladu nahraďte

```
Iblur = imread('pastelkygrayblurred.png');
```

Iblur vypočítejte z I.

ÚKOL 4

Vyberte vhodnou metodu a najděte hrany v následujícím obrázku

```
I3 = rgb2gray(imread('cv6-img.png'));  
figure, imshow(I3);
```

ANALÝZA A ZPRACOVÁNÍ OBRAZU

