

# **Pokročilé spracovanie obrazu**

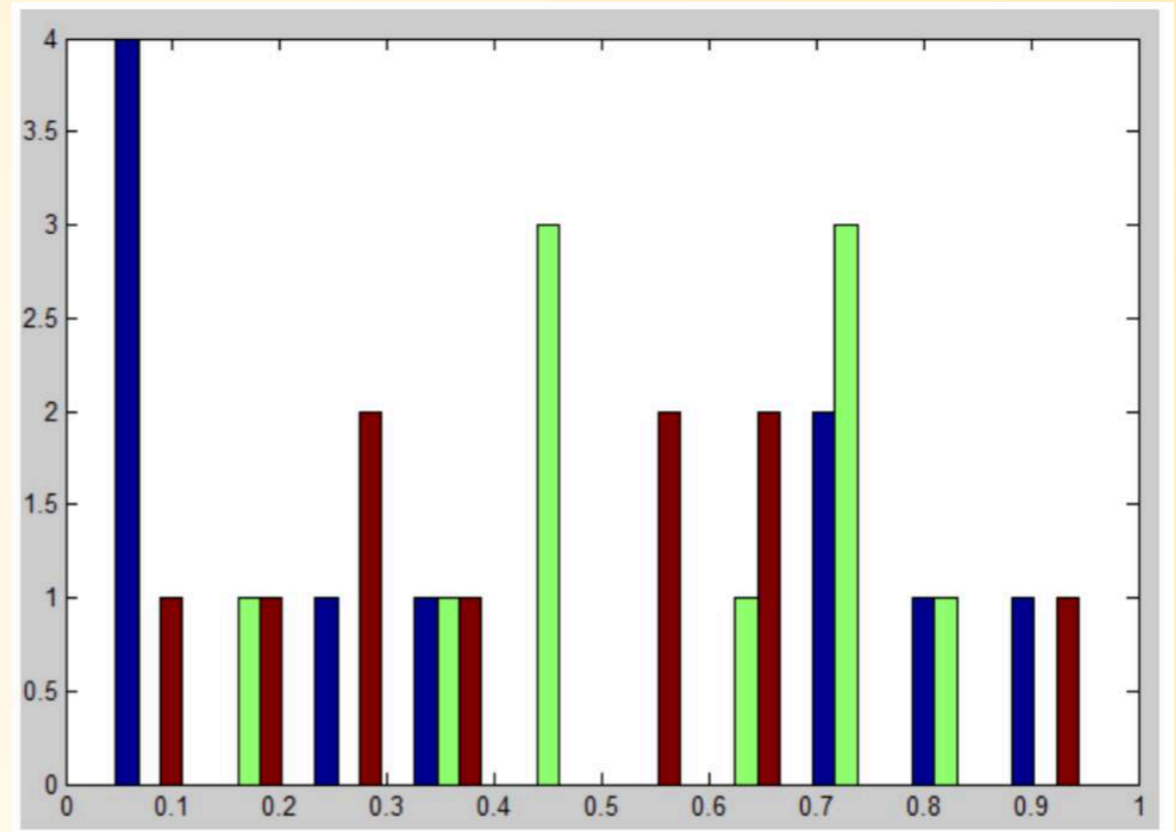
**Spracovanie obrazu**  
Cvičenia - 19.10.2016

# Spracovanie obrazu

- Histogram
- Vyhľadzovanie
  - priemer, medián
- Prahovanie
- Detekcia hrán
  - Sobel, Roberts,...

# Histogram

- Histogram početnosti
  - Stípcový diagram
  - os x – délka zvolených intervalov
  - os y – veľkosť absolútnych/relatívnych početností tried



# Histogram

```
n = hist(Y) %% Matlab 2015 n = histogram(Y)
```

```
n = hist(Y,x) %% n = histogram2(Y) -3D histogram
```

```
n = hist(Y,nbins)
```

```
[n,xout] = hist(...)
```

```
hist(...)
```

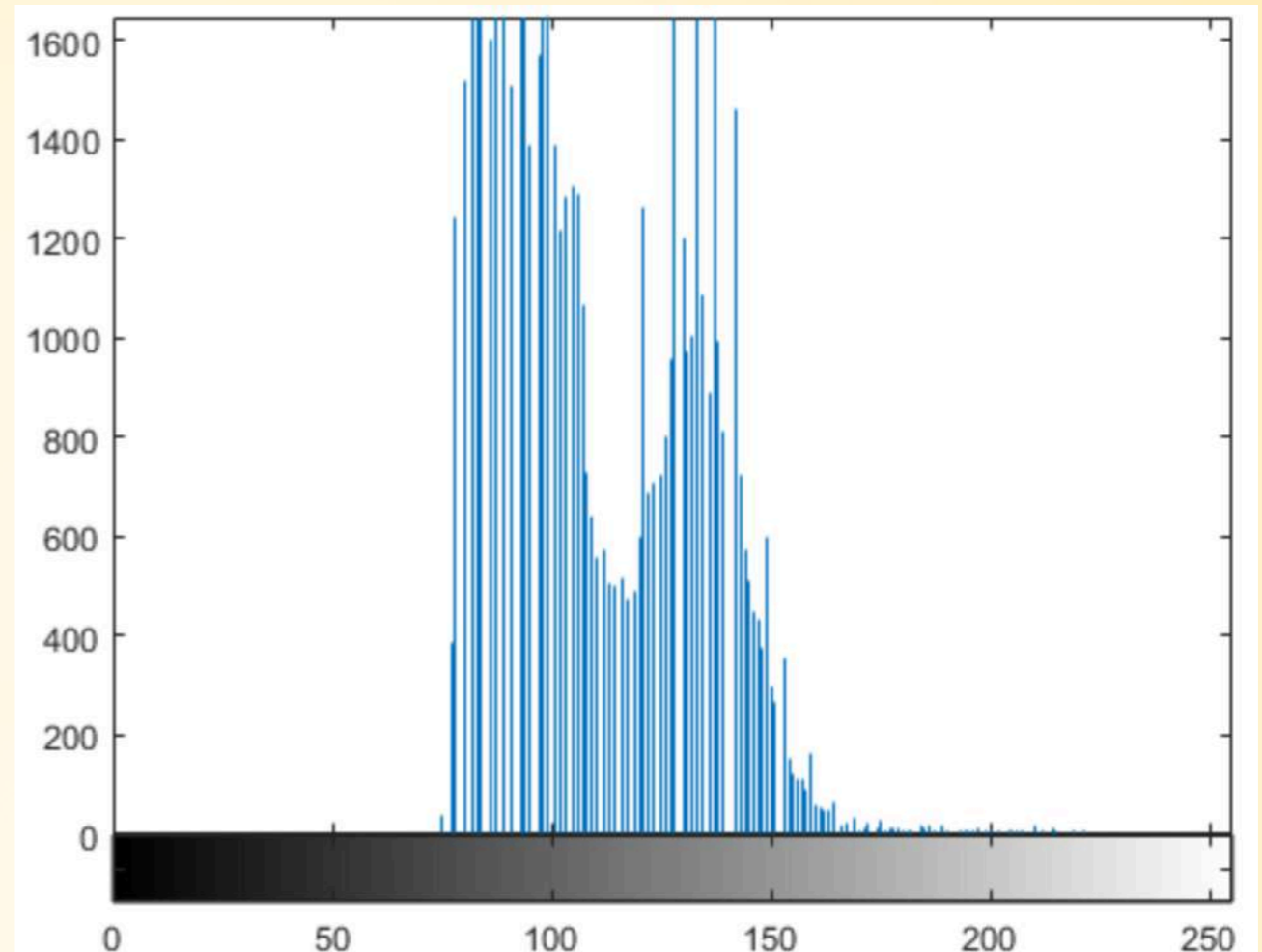
- nbins – počet tried histogramu
- pri N-D vráti histogram pre každý stĺpec spoločne v jednom grafe

# Histogram

`imhist(I);`

`imhist(I,n);`

`imhist(X,map);`



- histogram intenzity obrázka
- **I** - obrázok    **X** – indexový obrázok
- **n** je počet tried histogramu (default 256)

# Ekvalizácia histogramu

- Zvýšenie kontrastu obrazu
- Snaha dosiahnuť ideálny histogram

Čo je to ideálny histogram?

# Ekvalizácia histogramu

- Zvýšenie kontrastu obrazu
- Snaha dosiahnuť ideálny histogram
  - Obsahuje rovnaký počet z každej zastúpenej jasovej hodnoty

```
J = histeq(I, n)    %% zvyšuje na n diskrétnych úrovni  
šedej default 64
```

# Ekvalizácia histogramu

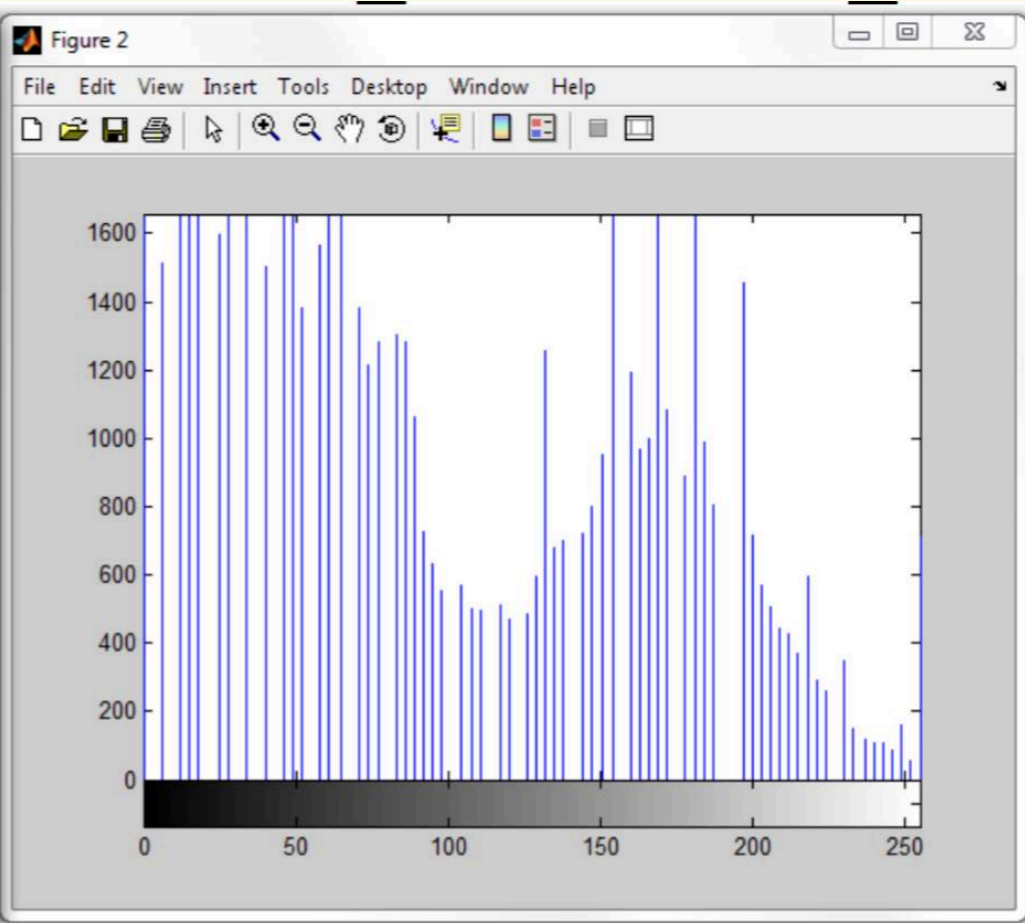
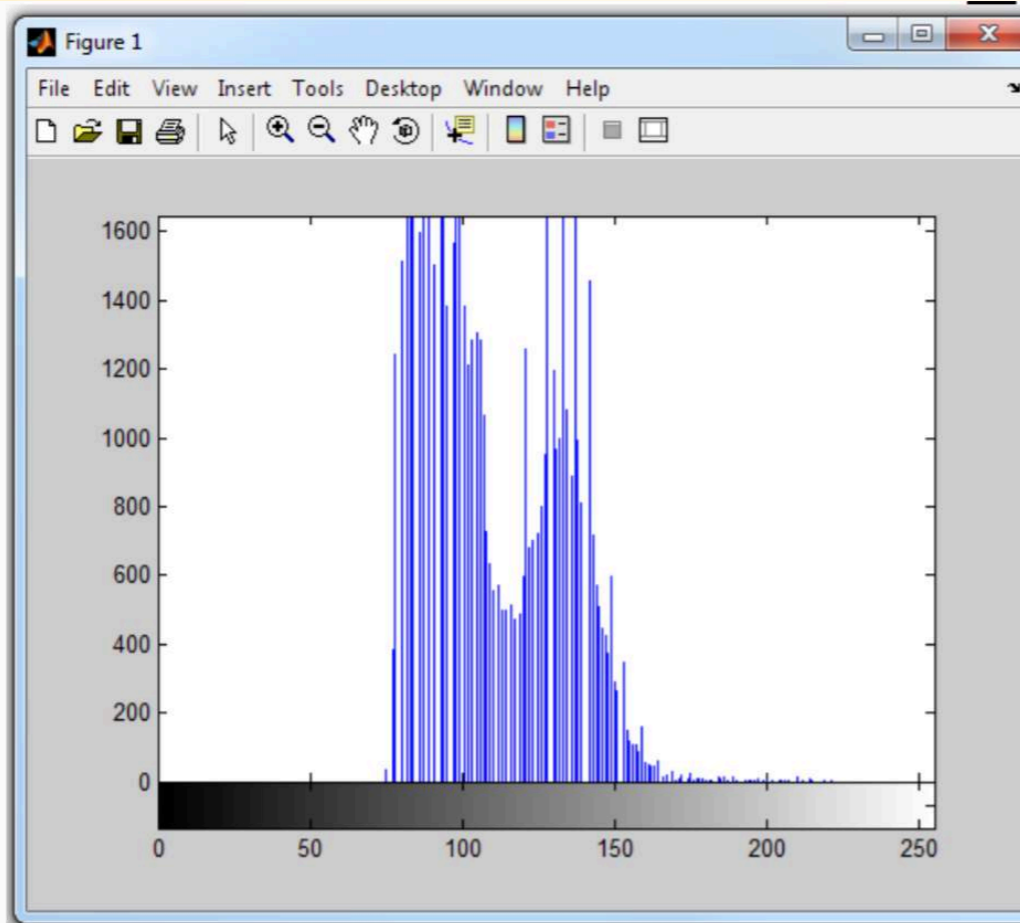
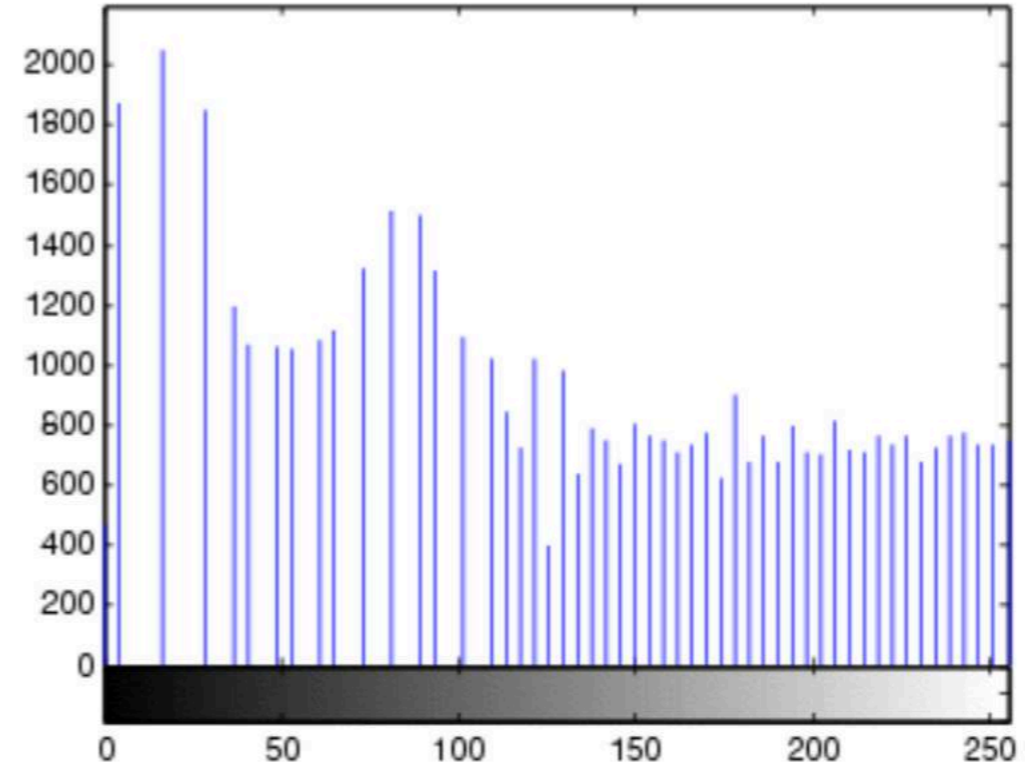
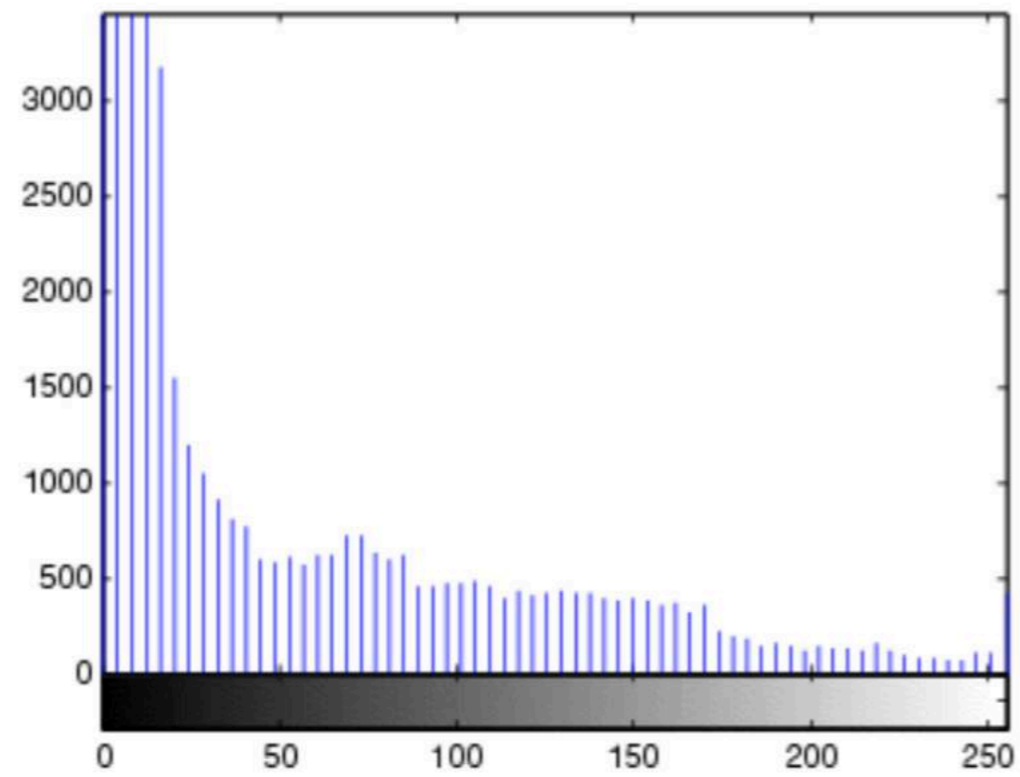
```
I = imread('tire.tif');  
J = histeq(I);  
imshow(I)  
figure, imshow(J)  
figure; imhist(I,64)  
figure; imhist(J,64)
```



# Ekvalizácia histogramu

```
J = imadjust(I) %% mapujú I na nové hodnoty  
J = imadjust(I, [low_in; high_in], [low_out; high_out])  
RGB2 = imadjust(RGB, [], []); %% RGB úpravy
```

Čo sa stane ak high out < low out ?



# Ekvalizácia histogramu

```
I = imread('pout.tif');  
J = imadjust(I);  
imshow(I), figure, imshow(J);  
K = imadjust(I, [0.3 0.7], []);  
figure, imshow(K);  
L = histeq(I);  
figure, imshow(L);
```

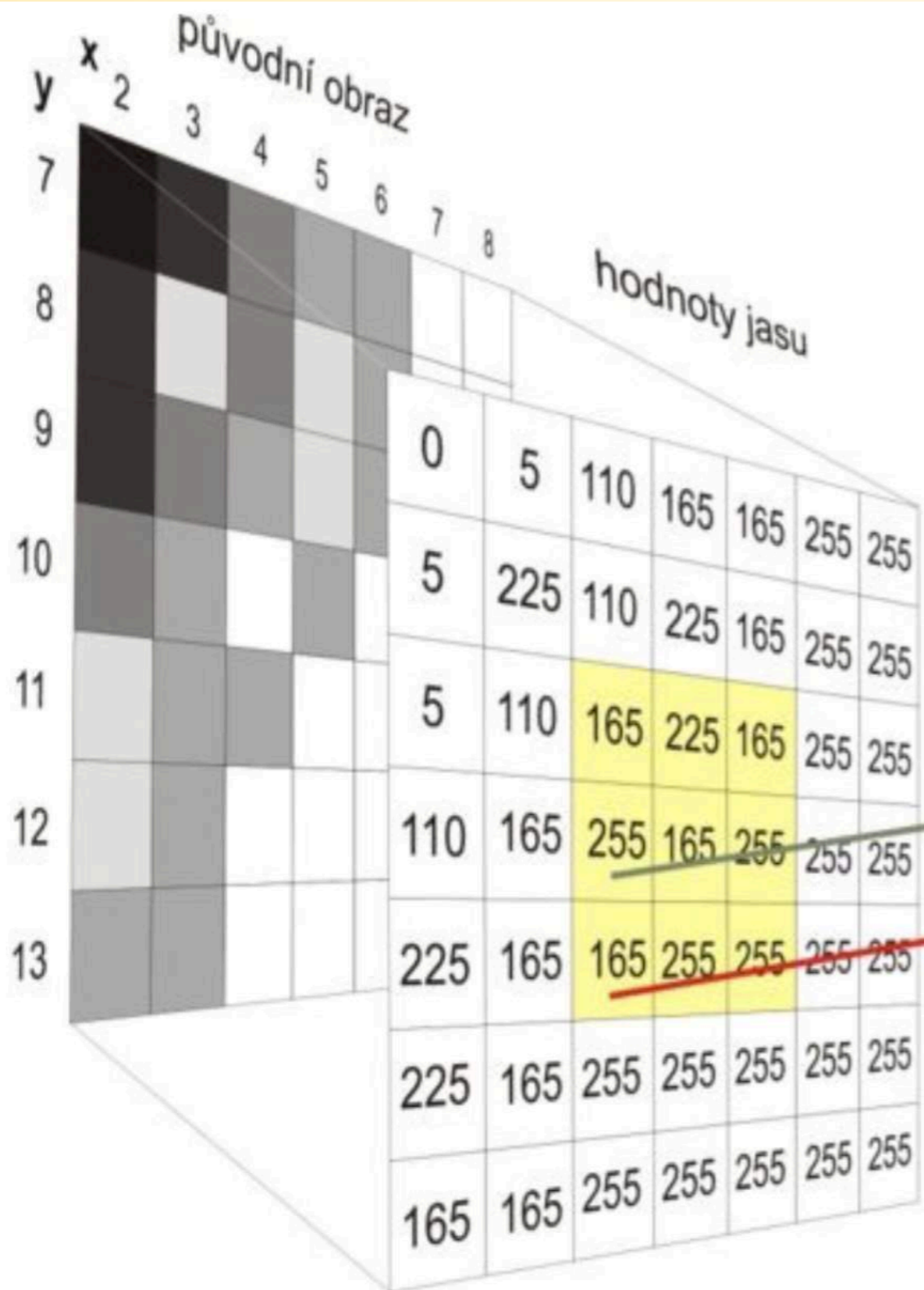
# Konvolúcia a korelácia 2D obrázka

- Korelácia: 
$$F \circ I(x, y) = \sum_{j=-N}^N \sum_{i=-N}^N F(i, j) I(x+i, y+j)$$

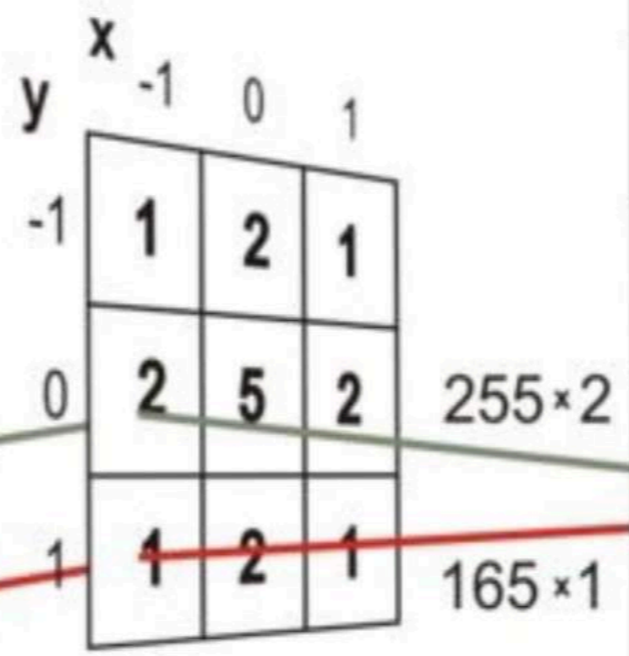
- Konvolúcia: 
$$F * I(x, y) = \sum_{j=-N}^N \sum_{i=-N}^N F(i, j) I(x-i, y-j)$$

- asociatívna

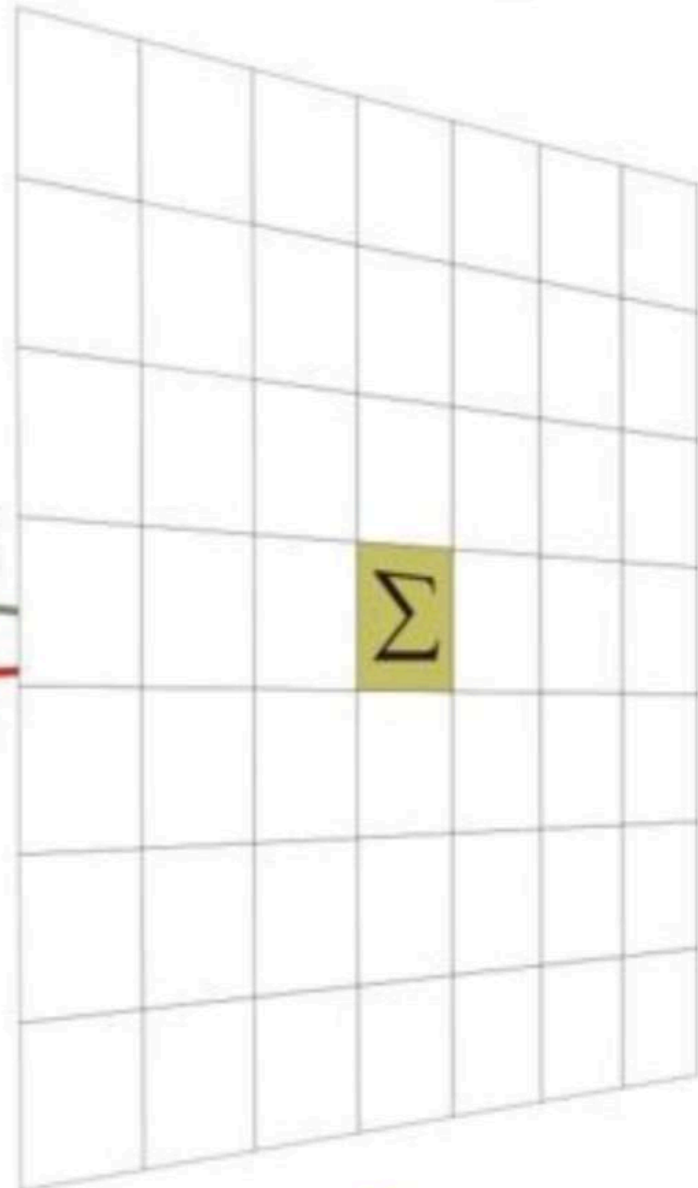
- Totožne pri symetrických filtroch



konvoluční maska



výsledný pixel



6	5	13 <sup>0,3</sup>	25 <sup>0,5</sup>	24 <sup>0,3</sup>	9
5	6	19 <sup>0,5</sup>	26 <sup>1</sup>	20 <sup>0,5</sup>	5
5	9	24 <sup>0,3</sup>	25 <sup>0,5</sup>	13 <sup>0,3</sup>	4
5	15	25	23	7	4
9	24	26	17	5	5

12,8	24,7	47,1	68	60,3	33,2
17,7	39,4	72,7	92,7		



12,8	24,7	47,1	68,2	60,3	33,2
17,7	39,4	72,7	92,7	72,9	37,3
20,8	50,2	84	89,3	58,4	27,8
29,4	65,7	91,5	80,4	44,8	22,6
46,6	87,8	97,3	69	35,2	22,5

# Konvolúcia

```
conv2(I,h,'same')
```

```
conv2(I,h,'full') = conv2(h,I,'full')
```

```
conv2(I,h,'valid')
```

```
filter2(h,I,'full') = conv2(h,I,'full')
```

```
%%pre symetrické h
```

- Otočí h o  $180^\circ$  a zavolá conv2

# Konvolúcia

```
A = rand(3);  
B = rand(4);  
C = conv2(A,B)    % C is 6-by-6
```

```
C =  
    0.1838    0.2374    0.9727    1.2644    0.7890    0.3750  
    0.6929    1.2019    1.5499    2.1733    1.3325    0.3096  
    0.5627    1.5150    2.3576    3.1553    2.5373    1.0602  
    0.9986    2.3811    3.4302    3.5128    2.4489    0.8462  
    0.3089    1.1419    1.8229    2.1561    1.6364    0.6841  
    0.3287    0.9347    1.6464    1.7928    1.2422    0.5423
```



# Dôležité konvolučné masky

- **Vyhladzovacie filtre**
  - Priemerovací filter
  - Gaussov filter
- **Hranové filtre**
  - Prewitt
  - Sobel
  - Roberts
  - ...

# Šum

```
J = imnoise(I, type)  
'gaussian', 'salt & pepper', 'speckle'
```



# Typy filtrov

```
fspecial(typ, parametre)
```

```
h = fspecial('average', hsize)
```

```
h = fspecial('disk', radius)
```

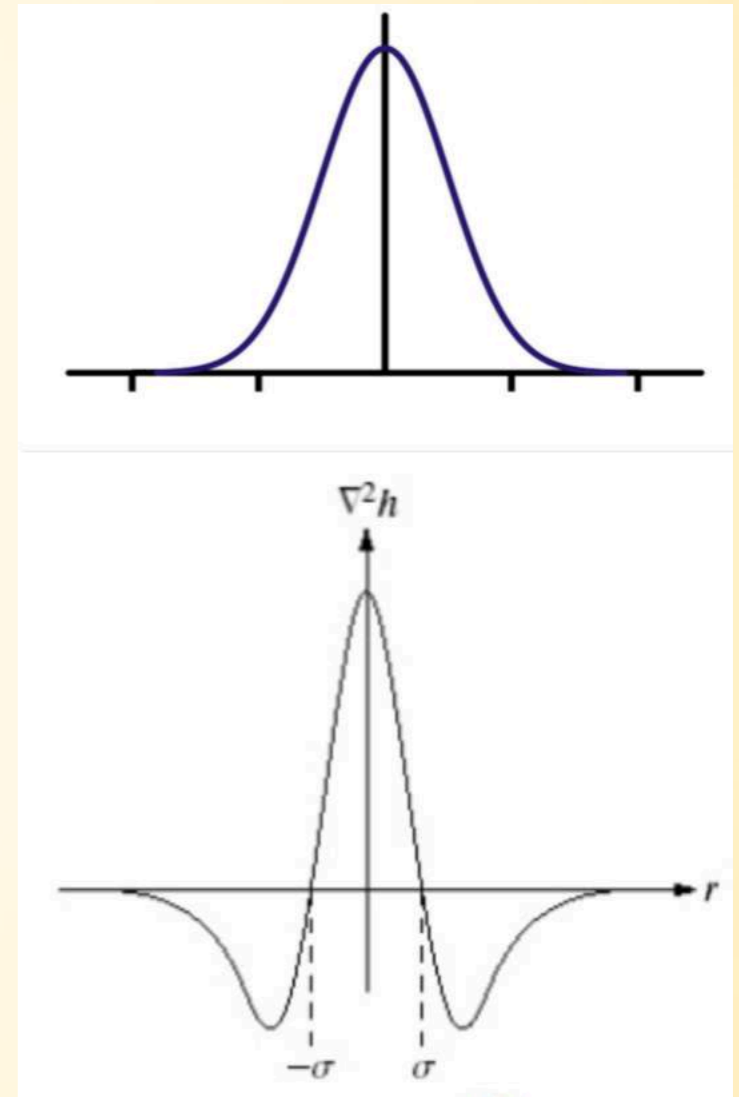
```
h = fspecial('gaussian', hsize, sigma)
```

```
h = fspecial('log', hsize, sigma)
```

```
h = fspecial('prewitt')
```

```
h = fspecial('sobel')
```

```
image(h*255)
```



# Priemerovací filter

- Výsledkom aritmetický priemer okolia bodu
- Priemerovací filter 7x7:

$$\frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} .$$

# Priemerovací filter

```
h = fspecial('average', 3) %% vytvorí 2D filter  
imfilter(I,h);
```



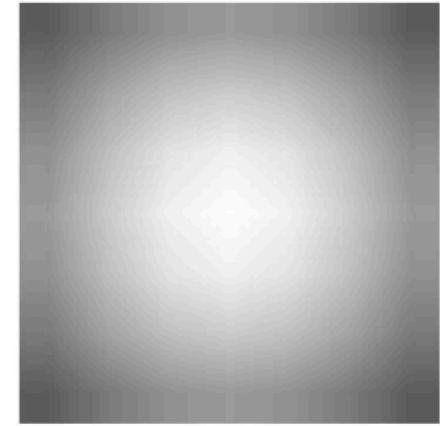
**Obrázok 2.2:** Vstupný obraz – detail nohy vtáka.



**Obrázok 2.3:** Príklad obrazu filtrovaného priemerovacím filtrom.

# Gaussov filter

- Filtračné jadro odvodené Gaussovej 2D funkcie



Obrázok 2.4: Gaussov filter  $h$  veľkosti  $7 \times 7$ ,  $\sigma = 3$ .



Obrázok 2.5: Vstupný obraz – detail nohy vtáka.



Obrázok 2.6: Príklad obrazu filtrovaného Gaussovým filtrom.

# Mediánový filter

- Zorad'ovací filter ( nepočítame konvolúciu)
- Výborným nástrojom na odtránenie šumu  
`medfilt2(I, [3, 3], 'symmetric')`



**Obrázok 2.16:** Vstupný zašumený obraz.



**Obrázok 2.17:** Mediánovo filtrovaný obraz.

# imfilter

```
B = imfilter(A, h, option1, option2,...)
X=0, 'symmetric', 'replicate', 'circular'
'same', 'full',
'corr', 'conv'
```



# Úloha na záver

- Načítajte obrázok
- Zašumte rôznym šumom

`J = imnoise(I, 'salt & pepper', d) % d = hustota`

- Odstráňte šum pomocou rôznych filtrov  
priemer, median, gaussov..