

NPGR032 – CVIČENÍ III.

Šum a jeho odstranění – teorie&praxe

Adam Novozámský (novozamsky@utia.cas.cz)



TEORIE

Šum a jeho odstranění

ŠUM

- Co je to šum v obrázku a jak vzniká?
- Jaké známe typy šumu?



ŠUM

- V obrázku je přidána falešná informaci nahodilého původu
- Jak vzniká v digitální fotografii?
 - Přímou na senzoru (CCD nebo CMOS)
 - Tepelné zahřívání - Dark Current
 - Prachové částice
 - Při zpracování ve fotoaparátu
 - Kvantizační šum
 - Umělým přidáním
- Typy:
 - Aditivní náhodný šum: $g = f + n$
 - Gaussovský bílý šum (AGWN)
 - Impulsní šum (sůl a pepř)



ŠUM

- Co je to bílý šum a čím se vyznačuje?

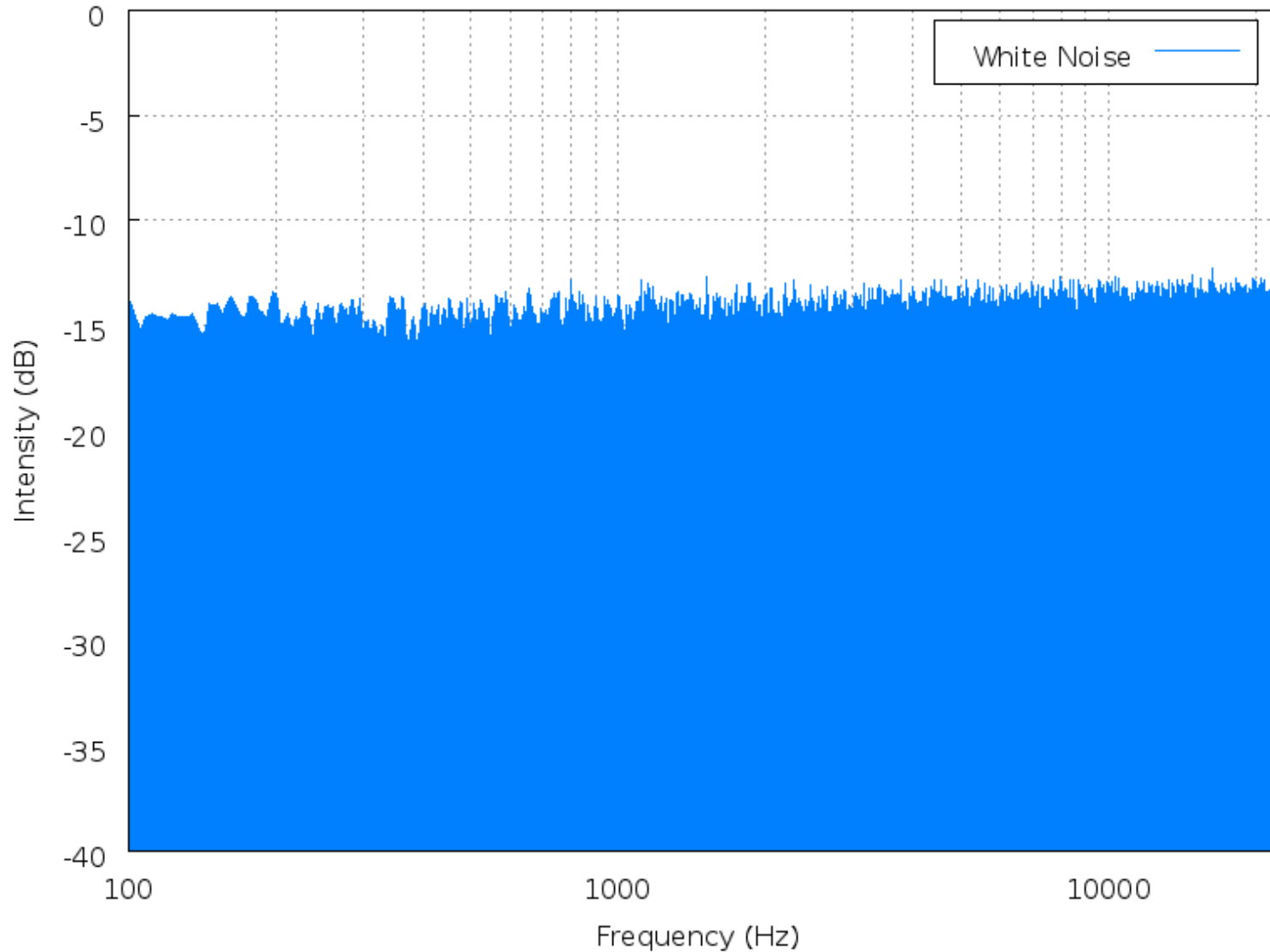


ŠUM

- **Gaussovský bílý šum** má normální rozložení
 - míra šumu stejná na všech pixelech
- Pokud něco nazýváme bílým, myslíme tím:
 - že dvě náhodné veličiny jsou navzájem nekorelované
 - míra šumu je pixel od pixelu na sobě nezávislá. Jedná se třeba o tepelný šum na CCD.
 - že střední hodnota je rovna nule
- Značíme AGWN = Adding Gaussian White Noise



SPEKTRUM BÍLÉHO ŠUMU



Nekorelované x Nezávislé se u gaussovských veličin rovná



ŠUM

- Šum sůl a pepř – jak se dá popsát?



ŠUM

○ Impulsní šum (sůl a pepř)

- náhodné veličiny šumu nabývají tří hodnot:

Hodnota	Pravděpodobnost	V obrázku
$+\infty$	p	(bílý)
$-\infty$	p	(černý)
0	$1-2p$	(nemění se)

- Čím se „ p “ zvětšuje \Rightarrow více zašumněné



MODELY ŠUMU

- **Gaussovský bílý šum**



- **Šum (sůl a pepř)**



ŠUM

- Jak měříme šum v obrázku?



ŠUM

- Signal-to-noise ratio (**SNR**) - míra šumu v obraze
 - $\text{SNR} = 10 \log (D(f)/D(n)) \text{ [dB]}$
 - $D(f)$... rozptyl nezašuměného signálu
 - $D(n)$... rozptyl šumu
 - Ve frekvenční oblasti je SNR definována takto:
 - $\frac{|N|^2}{|F|^2} (u, v)$
 - Kdyby šum byl bílý $\Rightarrow |N|^2 = \sigma_n^2$
 - Pokud je signál nekorelovaný $\Rightarrow |F|^2 = \sigma_f^2$
 - Což jsou ty rozptyly:
 - $\frac{\sigma_n^2}{\sigma_f^2}$
- Protože ty rozptyly v praxi moc neznáme, tak to odhadujeme většinou jako celek



ŠUM

- Máme dva obrázky – jeden má míru šumu 30, druhý 100, který je více zašuměný?
- Co se odstraňuje lépe – Impulsní šum nebo AGWN?
- Které výroky ohledně náhodných veličin (NV) jsou pravdivé?
 - **Nezávislé NV nemusí být nekorelované.**
 - **Nezávislé NV jsou vždy nekorelované.**
 - **Nezávislé NV jsou vždy korelované.**
 - **Nekorelované NV nemusí být nezávislé.**
 - **Nekorelované NV jsou vždy nezávislé.**



ŠUM

- Máme dva obrázky – jeden má míru šumu 30, druhý 100, který je více zašuměný?
- Co se odstraňuje lépe – Impulsní šum nebo AGWN?
- Které výroky ohledně náhodných veličin (NV) jsou pravdivé?
 - Nezávislé NV nemusí být nekorelované.
 - **Nezávislé NV jsou vždy nekorelované.**
 - Nezávislé NV jsou vždy korelované.
 - **Nekorelované NV nemusí být nezávislé.**
 - Nekorelované NV jsou vždy nezávislé.



SNR U LENY



20 dB

10 dB

0 dB



ODSTRANĚNÍ ŠUMU

- Jaké známe metody na odstranění šumu?



ODSTRANĚNÍ ŠUMU

○ Lineární metody

- Průměrování v čase
- Konvoluční filtry (rozmazání)
- Filtry ve frekvenční oblasti

○ Nelineární metody

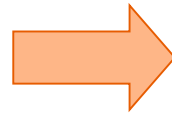
- Mediánový filtr
- *Minimalizace funkcionálu (neprobíráno)*
- *Splajnové metody (neprobíráno)*



ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

○ Průměrování v čase

- Scéna je statická (nehýbe se)
- Nafotím ji vícekrát
- Sečtu v jednotlivých pixelech
- Vydělím počtem snímků
- Šum klesá s hodnotou σ^2/N
- Tato metoda nepřináší žádné degradace



ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

- Průměrování v čase

šum

4

8



16

32

64



ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

○ Konvoluční filtry

- Lokální průměrování s maskou (konvoluce)
- Odstranění šumu → ← Rozmazání obrázku
 - Šum je vysokofrekvenční = > Potlačení vysokých frekvencí = > ztráta hran

- Jaké máme konvoluční filtry?



ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

○ Konvoluční filtry - Průměrování (prosté a vážené):

- $C = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

$$C = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

šum



3x3

5x5



7x7



ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

○ Konvoluční filtry – **Podél hran:**

- Pokud víme kde jsou hrany a jakým směrem jdou
- Měníme masku podle toho a průměrovat jen podél hran
- Problém:
 - hranový detektor detekuje stejně hrany jako šum
 - nutnost apriorní informace, kde jsou hrany



ODSTRANĚNÍ ŠUMU - ROTUJÍCÍ OKNO

- Vylepšuje průměrování – „respektuje“ směr a umístění hran v obrázku
- Docela dobré výsledky, ale časově náročná na výpočet
- Princip:
 - Pracuje na okolí 5x5 bodů
 - V 8 směrech od středního bodu počítá rozptyl
 - Vybere oblast s nejmenším rozptylem
 - Spočítá průměr a nahradí jím bod uprostřed masky 5x5

$$\begin{bmatrix} \cdot & O & O & O & \cdot \\ \cdot & O & O & O & \cdot \\ \cdot & \cdot & O & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & O & O \\ \cdot & \cdot & O & O & O \\ \cdot & \cdot & O & O & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & O & O \\ \cdot & \cdot & O & O & O \\ \cdot & \cdot & \cdot & O & O \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$



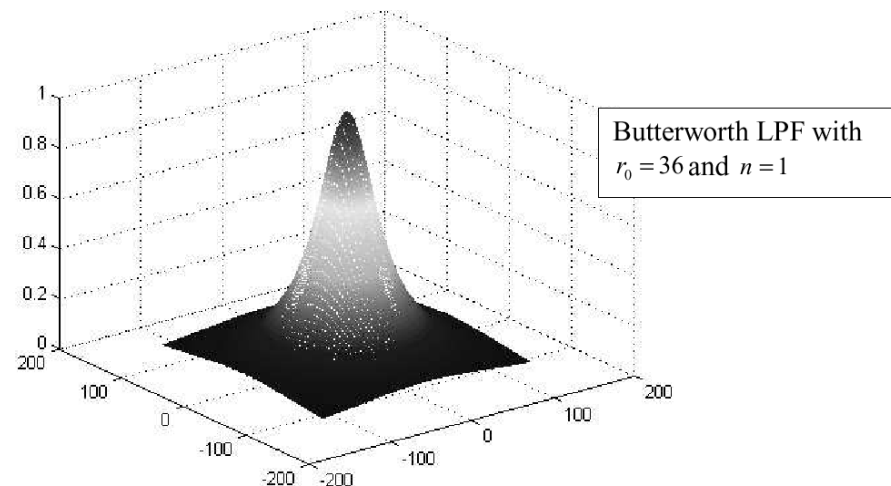
ODSTRANĚNÍ ŠUMU - LINEÁRNÍ METODY

○ Filtry ve frekvenční oblasti:

- Podíváme se do frekvenční oblasti
- Odstraníme nebo utlumíme vysoké frekvence pomocí hladkých **low-pass filtrů** (...to jsme dělali minule...)
- Může nastat podobný problém jako u konvolučních filtrů



LPF image, $r_0 = 13$



ODSTRANĚNÍ ŠUMU - NELINEÁRNÍ METODY

- **Mediánový filtr:**
 - Jak funguje?



ODSTRANĚNÍ ŠUMU - NELINEÁRNÍ METODY

○ Mediánový filtr:

- Posouváme okno jako při konvoluci
- V každém posunutí spočítáme medián
- Dosadím ho do středového bodu

- Jak spočteme medián v masce 3x3?

- $A = \begin{pmatrix} 7 & 4 & 6 \\ 2 & 1 & 4 \\ 5 & 3 & 9 \end{pmatrix}; med(A) = ?$

- Na co funguje mediánový filtr lépe:
 - AWGN
 - „Pepř a sůl“



ODSTRANĚNÍ ŠUMU - NELINEÁRNÍ METODY

- $A = \begin{pmatrix} 7 & 4 & 6 \\ 2 & 1 & 4 \\ 5 & 3 & 9 \end{pmatrix}$

- $\{1 \ 2 \ 3 \ 4 \ 4 \ 5 \ 6 \ 7 \ 9\}$

- $med(A) = 4$

- Na „pepř a sůl“ funguje dobře

- Co když je výskyt šumu v daném vybrání větší než 50% ?

- Jaký to má vliv na hrany?

- Co když máme jednopixelovou čáru?



ODSTRANĚNÍ ŠUMU - NELINEÁRNÍ METODY

- Šum $> 50\%$
 - Pokud je cca $\frac{1}{2}$ poškozených pixelů bílá a $\frac{1}{2}$ poškozených černá, tak medián je hodí na protilehlé strany a stejně si vezme v prostředku ty nepoškozené pixely. Takže i tak to může vypadat poměrně slušně – v tom je kouzlo mediánového filteru
 - Pokud bude více jak $\frac{1}{2}$ všech pixelů v dané oblasti bílá nebo černá, tak originální signál se při počítání mediánu posune ze středu a šum bude vybrán jako výsledný signál.
- Pokud je obrázek hodně zašuměn
 - „okusuje“ okraje a rohy (zakulacuje pravoúhlé hrany)
 - Řešení: jako výběrové okno mít třeba kříž
- Jednapixelovou čáru to „sežere“



MEDIÁN

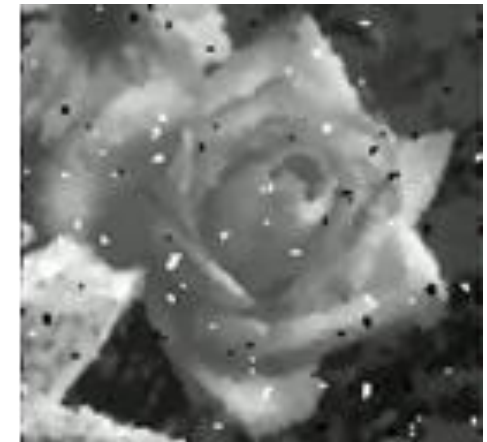
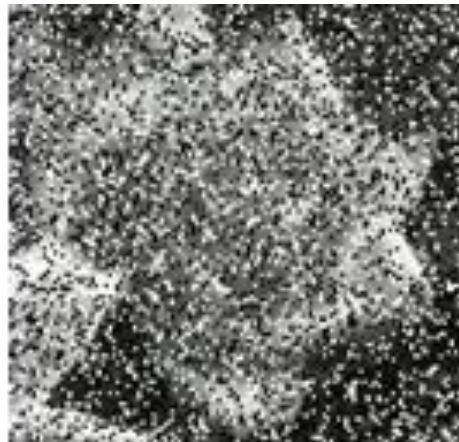
Originál



Poškozený



Filtrovaný





PRAXE

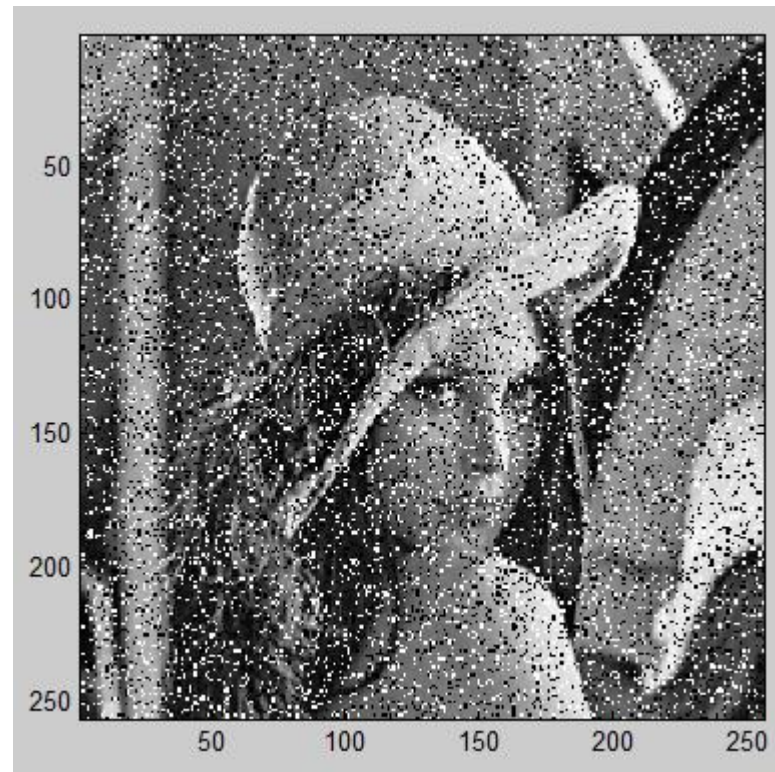
Šum a jeho odstranění

ŠUM – ÚLOHA I.

- Vytvořte funkci na zašumění snímku solí a pepřem:
- `function I = sulPepr(I, Pomer)`
 - Náповěda: `r = rand(n)`



Originál



Poškozený
Pomer = 0,3



ŠUM – ÚLOHA I. - ŘEŠENÍ

```
function I = sulPepr(I, Pomer)
% I = sulPepr(I, Pomer) - degrades image I with noise
salt and pepper
```

```
A = rand(size(I));
I(A<=Pomer/2) = 0;
I(A>=1-Pomer/2) = 255;
```

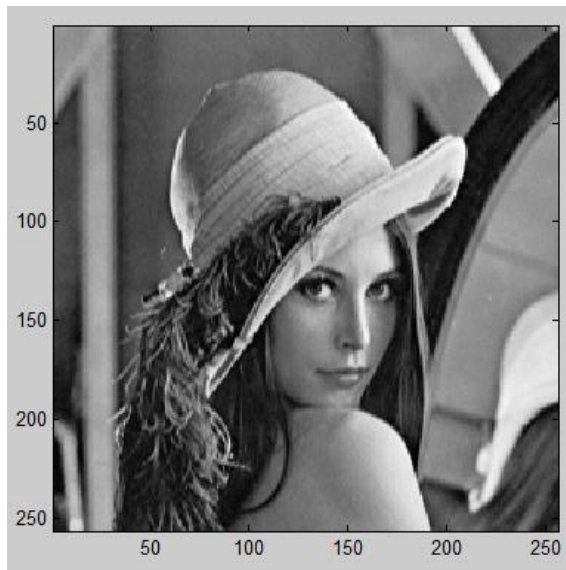
```
-----

function I = sulPepr(I, Pomer)
A = rand(size(I));
I(A<=Pomer/2) = min(I(:));
I(A>=1-Pomer/2) = max(I(:));
```

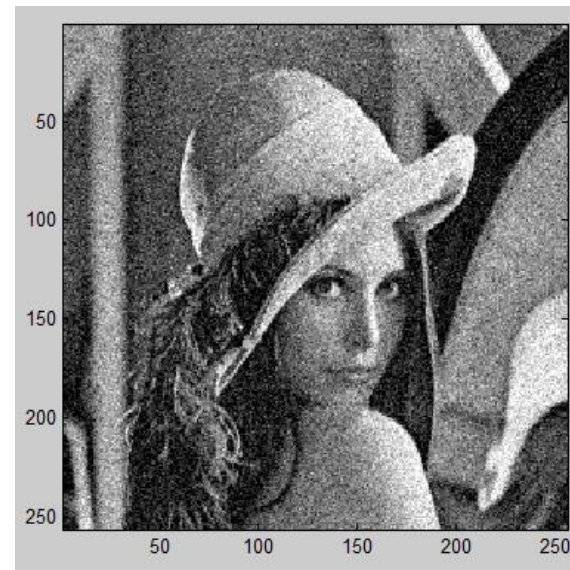


ŠUM – ÚLOHA II.

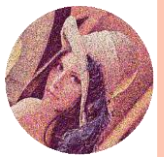
- Vytvořte funkci na zašumění snímku bílým šumem o daném SNR:
- `function R = bilySum(I, SNR)`
 - Náповěda: $r = \text{randn}(n)$



Originál



Poškozený



ŠUM – ÚLOHA II. - ŘEŠENÍ

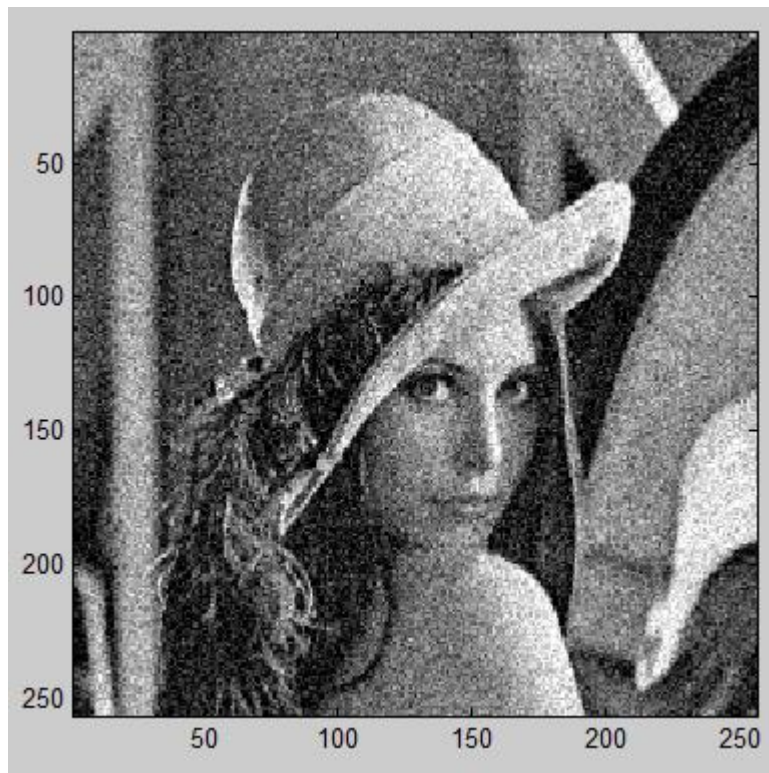
```
function R = bilySum(I, SNR)
% R = bilySum(I, SNR) - adds white noise of SNR to
image I

MinI = min(I(:));
MaxI = max(I(:));
S = sqrt(var(I(:))/(10^(SNR/10)));
R = I + S*randn(size(I));
R(R<MinI) = MinI;
R(R>MaxI) = MaxI;
```

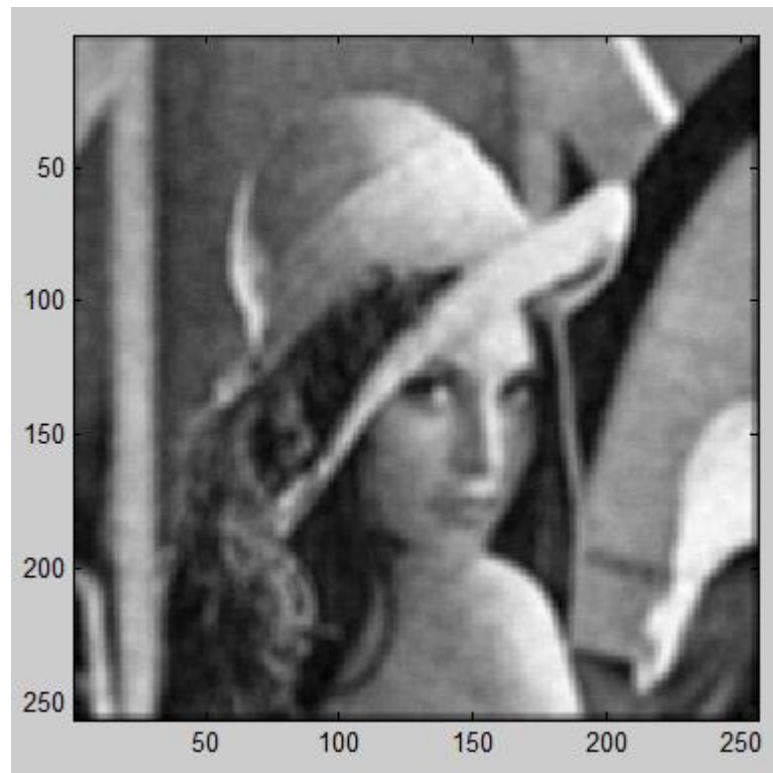


ŠUM – ÚLOHA III.

- Odstranit šum rozmazáním:
 - Náповěda: **conv2** s parametry 'full', 'same', 'valid'



Poškozený



Po filtraci



ŠUM – ÚLOHA III. - ŘEŠENÍ

```
zobr(conv2(B, kruh(3, 20), 'same'));
```

```
-----  
function K = kruh (R, N)
```

```
% K = kruh(R, N) - vraci kruhovou masku o polomeru R v  
matici NxN
```

```
[X, Y] = meshgrid(-(N-1)/2:(N-1)/2, -(N-1)/2:(N-1)/2);
```

```
K = double(X.^2 + Y.^2 < R^2);
```

- Co se stalo s intenzitou snímku?



ŠUM – ÚLOHA III. - ŘEŠENÍ

```
zobr(conv2(B, kruh(3, 20), 'same'));
```

```
-----  
function K = kruh (R, N)
```

```
% K = kruh(R, N) - vraci kruhovou masku o polomeru R v  
matici NxN
```

```
[X, Y] = meshgrid(-(N-1)/2:(N-1)/2, -(N-1)/2:(N-1)/2);
```

```
K = double(X.^2 + Y.^2 < R^2);
```

○ Co se stalo s intenzitou snímku?

- posunula se směrem nahoru, díky tomu, že jsme nenormovali po konvoluci

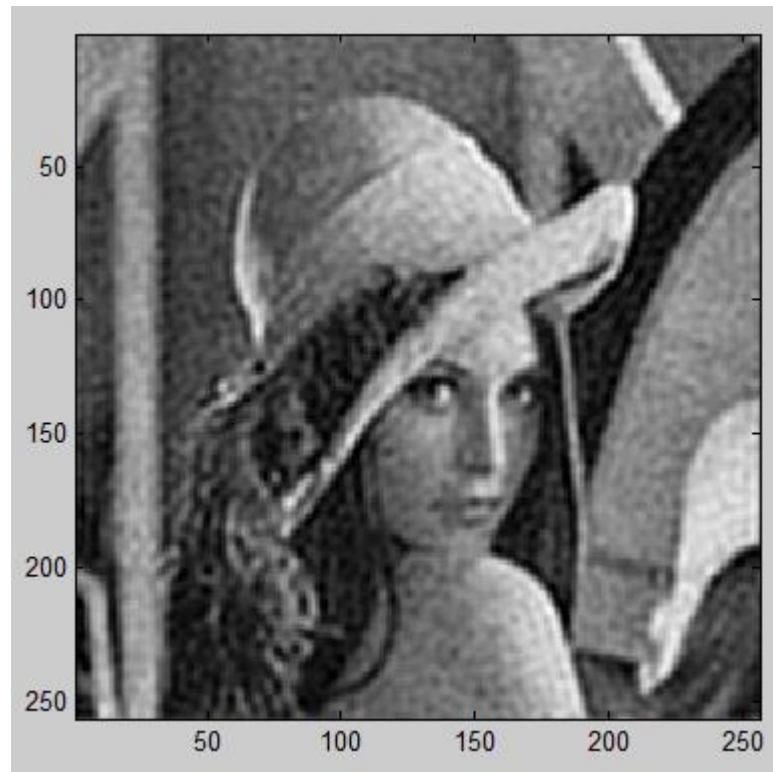


ŠUM – ÚLOHA IV.

- Odstranit šum low-pass filtrem:



Poškozený



Po filtraci

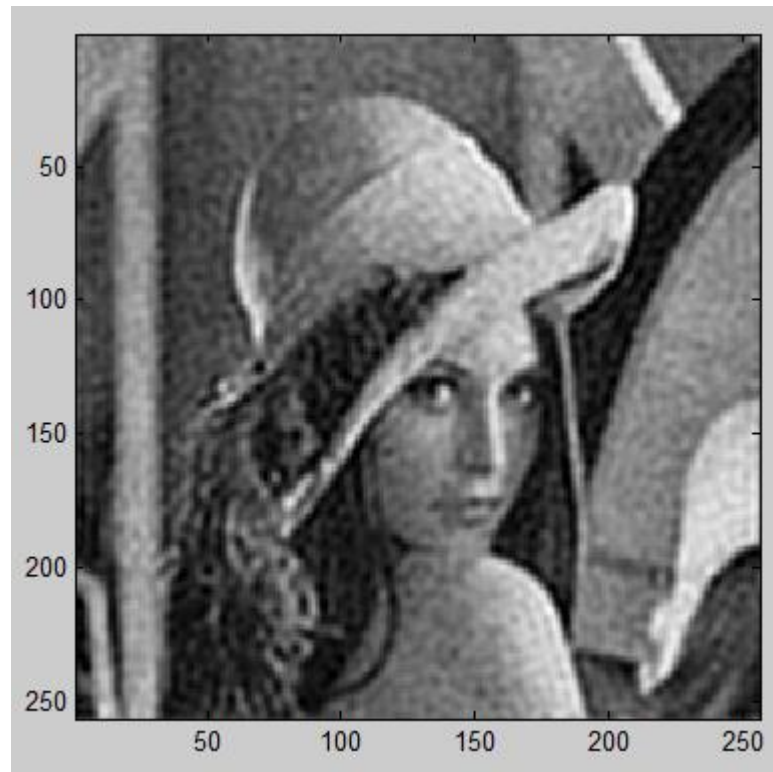


ŠUM – ÚLOHA IV. - ŘEŠENÍ

```
zobr(abs(ifft2(fft2(B).*fftshift(kruh(50,size(B,1))))));
```



Poškozený

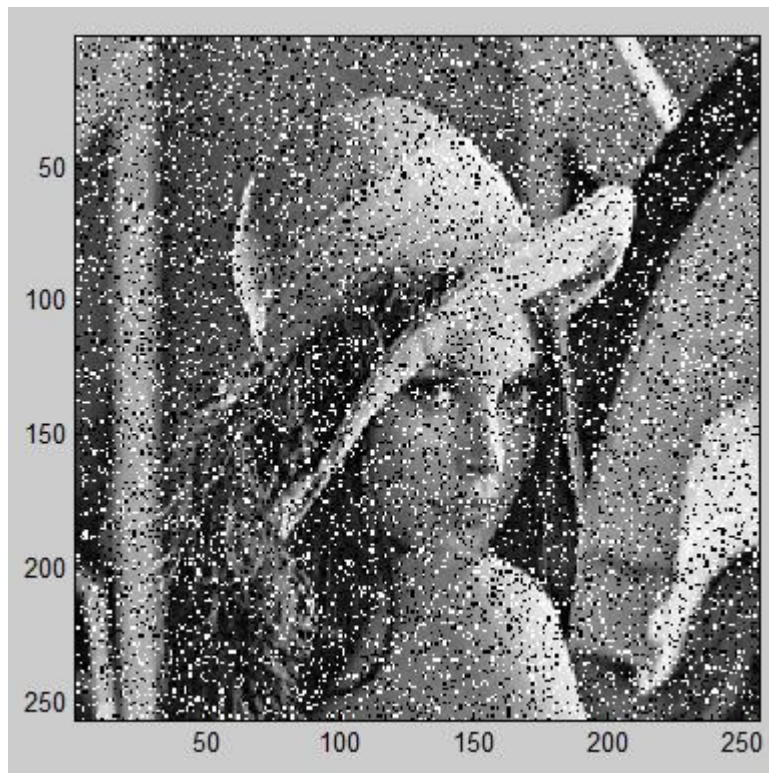


Po filtraci

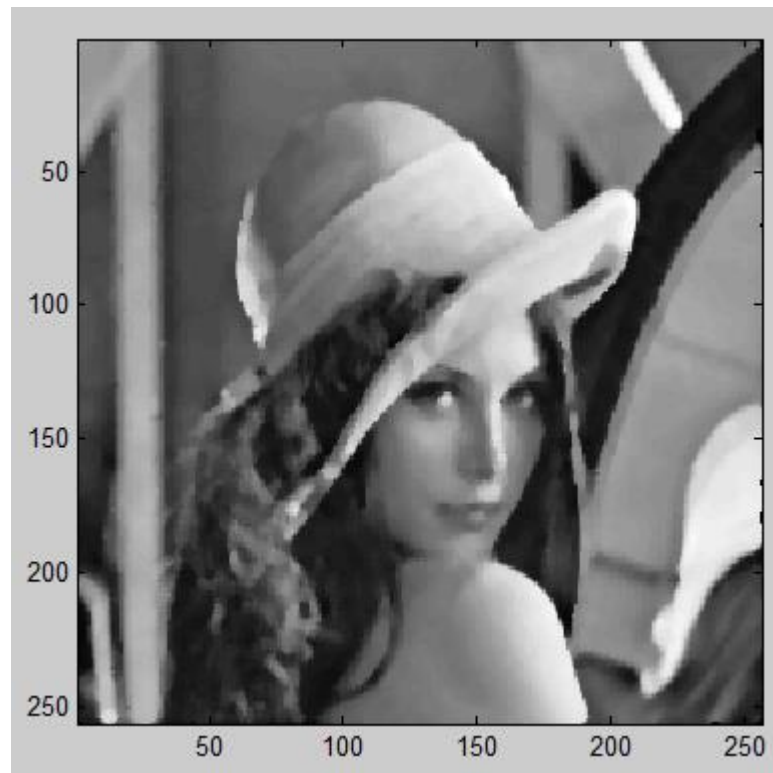


ŠUM – ÚLOHA V.

- Odstranit šum mediánovým filtrem:
 - Náповěda: `medop.m` (*v balíku zadání*)



Poškozený

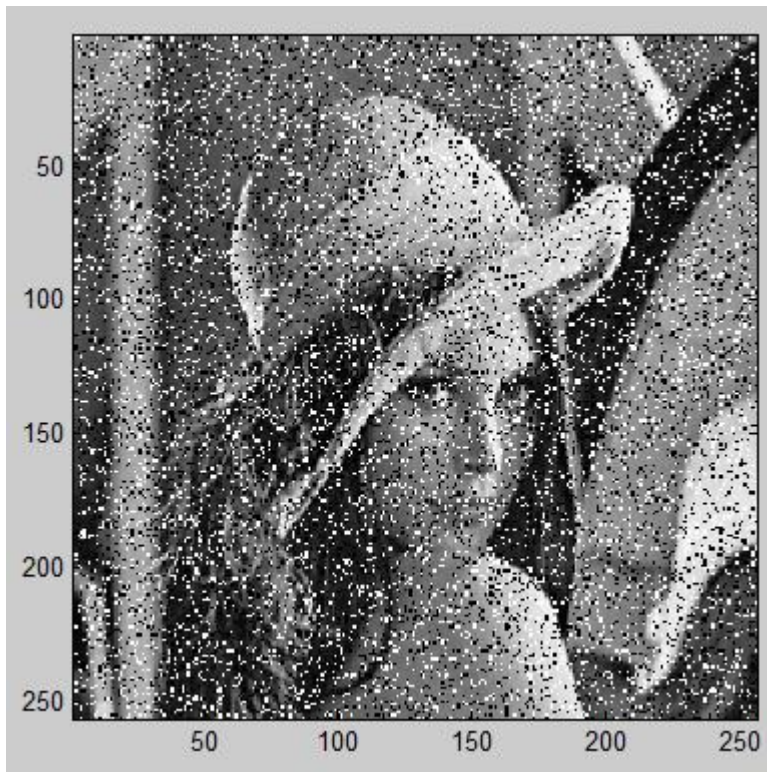


Po filtraci

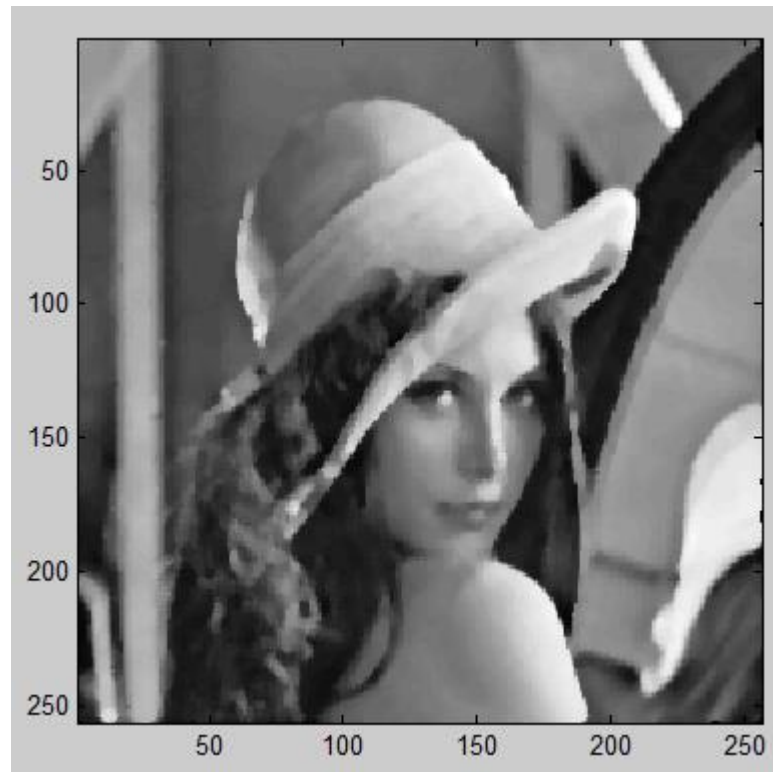


ŠUM – ÚLOHA V. - ŘEŠENÍ

`zobr (medop (J, ones (5))) ;`



Poškozený



Po filtraci



ŠUM – ÚLOHA PRO RYCHLÍKY

- Pro rychlíky: odstranit artefakty u low-pass filtru
zachovat roh čtverce u mediánového filtru
- Napsat vlastní mediánový filtr

