

Chyby v obraze, typy šumu, reštaurácia obrazu a optimálna filtrácia

Typy šumu

- Základné delenie:
 - o závislý šum
 - o nezávislý šum
- Ďalšie delenie (toto rozdelenie koreluje s predošlým rozdelením)
 - o aditívny šum (aj nezávislý šum) – obraz f a šum v nie sú závislé

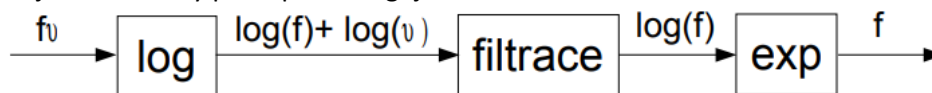
$$g(x, y) = f(x, y) + v(x, y)$$

- o multiplikatívny šum (aj závislý šum) – veľkosť šumu závisí na veľkosti obrazového signálu

$$g = f + vf = f(1 + v) \approx fv$$

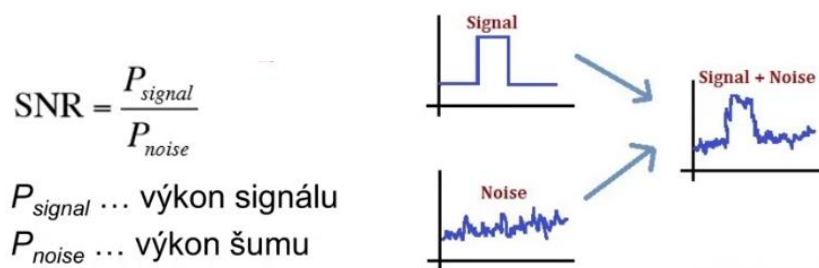
Homomorfná filtrácia

- Odstránenie multiplikatívneho šumu z obrázka
- Nie je to dokonalý prístup ale funguje



Signal-to-Noise Ratio (SNR)

- Typicky sa uvádza v decibeloch
- Čím vyššia hodnota SNR, tým čistší obrázok
- Meranie pomeru šumu voči obrazovému signálu
- Obrázok bez šumu / šum
- Ak má obrazový signál vyššie amplitúdy než šum nie až tak viditeľné poškodenie, ale ak má obraz približne rovnaké amplitúdy ako šum, obraz sa začne javiť značne poškodený

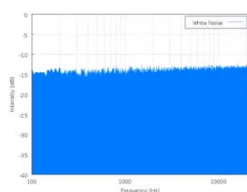


- Výkon signálu – suma vzoriek grafu, avšak každá vzorka je umocnená na druhú, aby sa pri sume neodčítali záporné vzorky (potrebujem ako to lieta hore aj dolu.)
 - o napr. sčítam všetky hodnoty pixelov na druhú

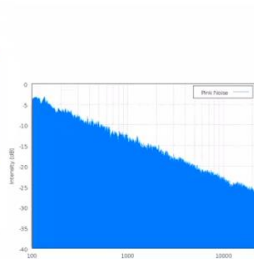
Biely šum

- nekolorovaný šum (náhodný)
- šum má konštantné výkonové spektrum. Intenzita šumu neklesá s rastúcou frekvenciou
- aproximácia najhorších šumov

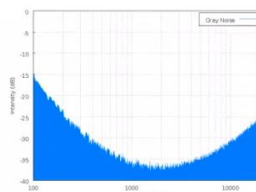
„Farby“ šumu



↑ **Bílý šum**
(rovnoměrné
výkonové
spektrum).



↑ **Růžový šum** (intenzita
šumu klesá s rostoucí
frekvencí – rovnoměrná v
logaritmickém měřítku).



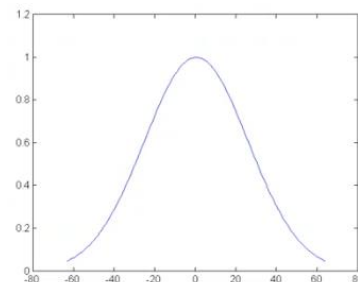
↑ **Šedý šum**
(definice v
terminologii
zpracování
signálu viz.
Wikipedia)

Gaussov šum

- šum pstihuje všetky pixely obrazu
- amplitúda šumu má normálne (gaussovo) rozloženie pravdepodobnosti

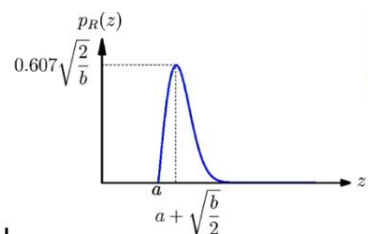
$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

- o stredná hodnota „mí“ je obvykle rovná 0
 - o najvyššia pravdepodobnosť je, že pixel nezmení vôbec, vrchol gaussovky je v bode 0
 - o málo pravdepodobné je podľa obrázku úprava o 60
- Biely gaussov šum vo veľa situáciách veľmi dobre aproximuje reálny šum
 - o napr. šum v prenosovom kanáli
- Síce je vhodný ale často sa používa aj tam, kde by nemal



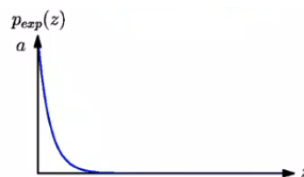
Rayleigh šum

- Tvar rozloženia je trochu iný než gaussov
- Range images – intenzita odpovedá vzdialenosti od kamery, napr. kinect ho využíva



Exponenciálny šum

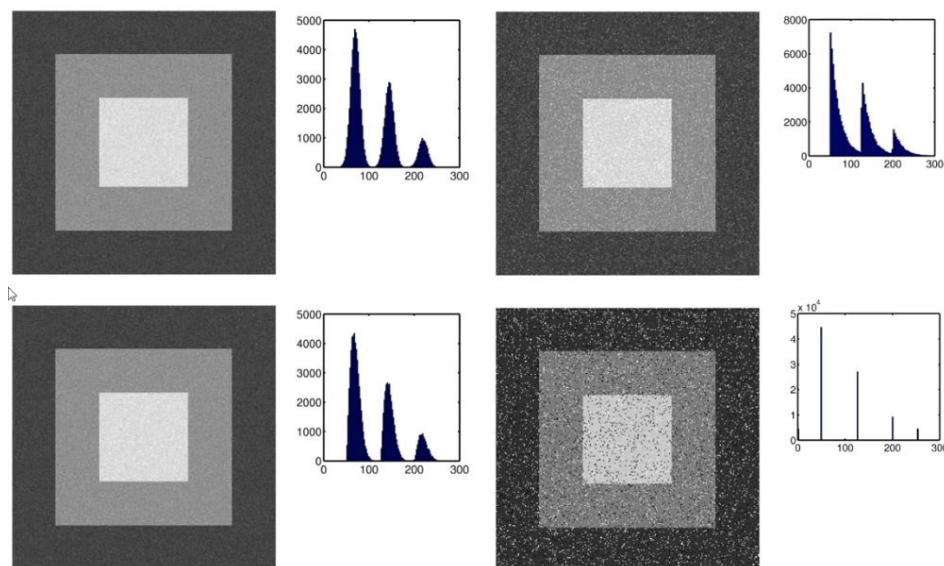
- Laser imaging – ležrové simanie obrazu



„Pepř a sůl“

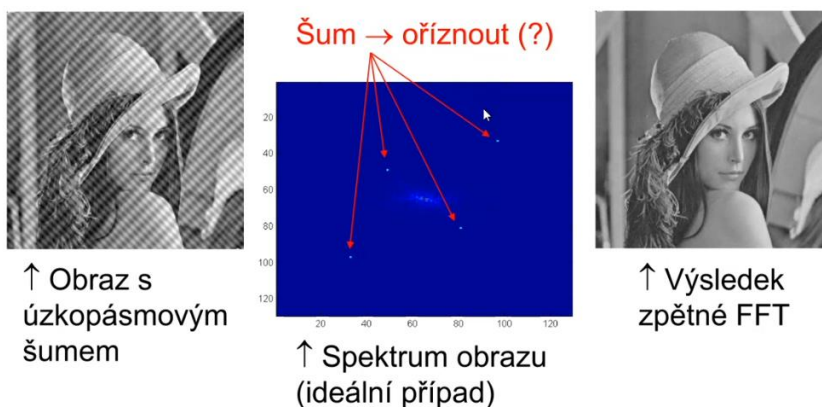
- Často nazývaný aj impulzný šum alebo výstrelový
- Napadne len niektoré pixle obrazu a zmení ich na extrémne hodnoty napr. biela, alebo čierna
- Je nutné na opravu detekovať tieto extrémne body a ich hodnotu upraviť pomocou okolných pixlov
 - o napr. vyhľadanie mediánovým filtrom

Porovnanie šumov



Úzkopásmový šum

- postihuje len niektoré frekvencie obrázku (na rozdiel od bieleho, ktorý postihuje všetky frekvencie)
- riešením odstránenia tohto šumu je filtrácia frekvenčnej oblasti



Štatistický prístup k filtrácii šumu

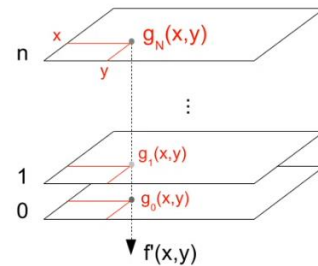
- metódy predpokladajú, že je šum aditívny a strednú hodnotu má nula a je nezávislý na signále
 - o priemerovanie z viacero snímkov (ale bez rozmazania)
 - o priemerovanie z opakujúcich sa oblastí

- priemerovanie vo vnútri snímku

Priemerovanie z viacerých snímkov

- Ak máme n realizácií rovnakej nemenej scény
- Realizácie však nemôžu obsahovať „pohyb“ pixelov, ak to obsahujú je nutné obrázky najprv predspracovať (stabilizácia obrázu)
- Vtedy môžeme spriemerovať pixely ktoré sú nad sebou

$$f'(x, y) = \frac{1}{n} \sum_{i=1}^n g_i(x, y)$$



Priemerovanie jedného snímku

- Často potrebujeme filtrovať šum z jediného obraz -> problem
- **Priemerovanie opakujúcich sa oblastí**
 - V podstate niekoľko realizácií v jednom obraze
- **Priemerovanie vo vnútri snímku**
 - Susedné pixely majú prevažne rovnakú alebo podobnú hodnotu jasu
- Representant okolia alebo priemer z niekoľkých hodnôt
- Problém rozmazávania hrán v obraze

Lokálne predspracovanie obrázu

- Rozdelenie:
 - **linéarne filtre** (obyč vyhladzovanie) – možná ľubovoľná kombinácia filtrov či aplikácia v rôznom poradí, výsledok bude vždy rovnaký
 - **nelineárne filtre** – tu záleží na poradí či kombinácií
- **Konvolúcia** – konvolučným oknom prechádzam obrázok, a prenasobím jednotlivé hodnoty konvolučného jadra s obrázok a sčítam ich dokopy
- **Filtrácia vo frekvenčnej oblasti** – prevediem obraz do spektra a tam na ňom pracujem

Priemerovanie

- Ak chceme zachovať intenzitu obrázku, suma filtru musí byť jedna, jednoducho dosiahnuteľné, zvoliť si váhy aké chcem a podeliť ich súčet.
- Príklad priemerovania:

$$h_9 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_{25} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Zváženie pixlov v strede masky

- Stále zachovávam sumu váh filtru rovnú jednej, nechcem zvyšovať intenzitu obrázu

$$h_{10} = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_{16} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Priemerovanie Gaussian filtrom

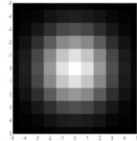
- Lineárny filter, jeho jadro je definované vzťahom

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$\frac{1}{273}$$

| | | | | |
|---|----|----|----|---|
| 1 | 4 | 7 | 4 | 1 |
| 4 | 16 | 26 | 16 | 4 |
| 7 | 26 | 41 | 26 | 7 |
| 4 | 16 | 26 | 16 | 4 |
| 1 | 4 | 7 | 4 | 1 |

↑ Konvoluční jádro
Gausiánu, $\sigma = 1$



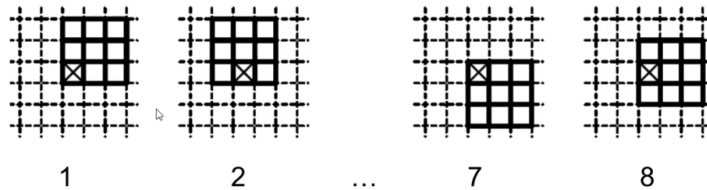
- Chceme aby v strede filtra boli hodnoty najvyššie a v okolí postupne klesali, napr. podľa gaussovho rozloženia

Nelineárne vyhladzovanie

- Nechceme vyhladzovať aj hrany ako do teraz
- Stredná hodnota je zlý odhad, pokiaľ existujú vychýlené hodnoty -> medián

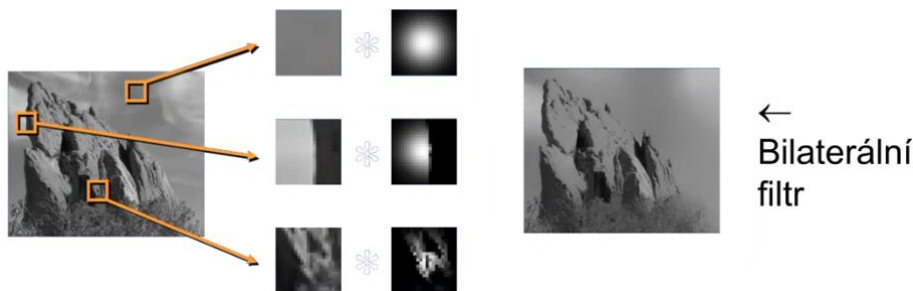
Filtrácia pomocou rotujúcej masky

- V okolí 5x5 vyhľadáva homogénnu časť rotujúca maska
- Hľadá pozíciu masky v ktorej je najmenší rozptyl pixelov
- Pre tento prípad mám 9 rôznych polôh



Bilaterálny filter

- Lepšia varianta rotujúcej masky, rozširuje klasický gaussov filter
- Využíva gaussovo jadro, ale je to nelineárny filter!
- Upravuje gaussovo jadro na základe toho ako pod jadrom obraz vyzerá



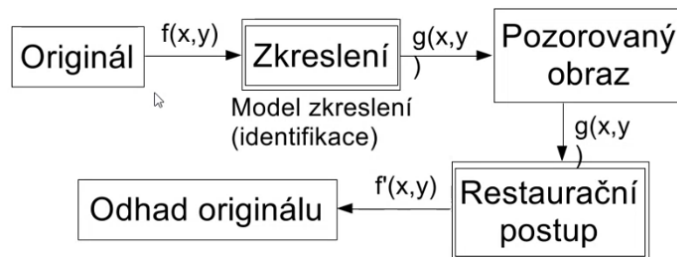
- Váhy okolných pixlov:
 - o Maska sa generuje podľa vzdialenosti od stredu okolia (typický gaussov filter)
 - o a ešte podľa rozdielu intenzít pixelov od stredu okolia

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} \underbrace{G_{\sigma_s}(\|p - q\|)}_{\text{space weight}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{range weight}} I_q$$

normalization factor
space weight
range weight

Reštaurácia (obnovenie) obrazu

- Rekonštrukcia originálneho obrazu z poškodeného obrazu
- Predpokladá sa známa degradácia obrazu
- Je nutný model zkeslenia, pre nás konvolúcia



- Najčastejší lineárny model poruchy obrazu:

$$g(x, y) = h(x, y) * f(x, y) + v(x, y)$$

- Prístupy:
 - o **deterministické metódy** – predpokladá známu degradáciu, slabý šum, najčastejšie sa použije inverzná transformácia k degradácii obrazu
 - o **štatistické metódy** – optimalizačné kritéria, napr suma najmenších štvorcov

Identifikácia zkeslenia

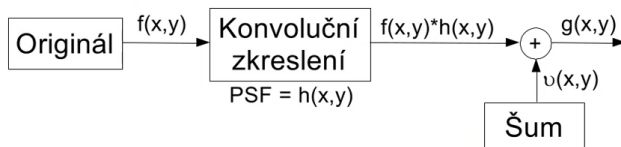
- Analiticky – spočíta sa impulzná charakteristika
- Experimentálne – hviezda (stálica) veľmi ďaleko
- Aposteriorná identifikácia – zo skreslených obrázkov, používajú sa líniové a hranové ciele
 - o Line Spread Function
 - o Step Spread Function

Rozmazanie pohybom

- Objekt sa pohybuje konštantnou rýchlosťou V v smere osy x vzhľadom ku kamere v dobe otvorenia závierky po dobu T
- Analytické riešenie modelu poruchy je: $H(u, v) = \frac{\sin(\pi VTu)}{\pi Vu}$
- Takto odstránim rozmazanie pohybom

Inverzná filtrácia

- Inak povedané **dekonvolúcia**
- Lineárny model zkeslenia obrazu



$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

- Pri prevedení do spektra sa mi z konvolúcie stane násobenie
- Rekonštrukčný filter, ak je $N(u, v)$ zanedbateľné

$$M(u, v) = H^{-1}(u, v) = \frac{1}{H(u, v)}$$

- Lenže ten šum tam je, takže uvedený rekonštrukčný filter nestačí (ak je zanedbateľný tak nejak to fungovať bude)

Optimálna filtrácia obrazu

- Je nutné o šume niečo vedieť, inak to nejde
- Jednotlivé metódy sa líšia tým, aké vlastnosti poznáme
- Metódy:
 - o Wienerov filter
 - o Viazaná dekonvolúcia – celková energia šumu
 - o Slepá dekonvolúcia – nepoznáme nič (dnes je možné natrénovať neuronovkou)

Wienerov filter

- Pracuje pre nezanedbateľný šum, ktorý má odhadnuteľné štatistické vlastnosti
- Šum je nezávislý na signále
- Filter:

$$M(u, v) = \underbrace{\frac{1}{H(u, v)}}_{\text{Prosty inverzný filter}} \underbrace{\frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{S_{vv}(u, v)}{S_{ff}(u, v)}}}_{\text{Wienerov korekčný faktor}}$$

- Tá veľká časť je wienerov korekčný faktor
 - o je v intervale 0 až 1
 - o pozerá sa na to, či vo frekvenciách $f(u, v)$ sa nachádza šum
 - $S_{vv}(u, v)/S_{ff}(u, v) \rightarrow 0$... nízky pomer šumu a signálu.
WKF $\rightarrow 1$
 - $H(u, v) \rightarrow 0$... zkrátením sa stratila niejaká frekvencia, preto nemá cenu zesilovať šum.
WKF $\rightarrow 0$
- Nepoužíva sa tento tvar filtru (len na vysvetlenie sa používa)
- V reálnom vystupuje len výkonové spektrum šumu a obrazu:

$$M(u, v) = \frac{1}{H(u, v)} \frac{S_{gg}(u, v) - S_{vv}(u, v)}{S_{gg}(u, v)}$$