



Filtrování

Úvod do analýzy obrazu

Mgr. Markéta Trnečková, Ph.D.

Úpravy obrazu

■ Geometrické transformace



■ Jasové transformace

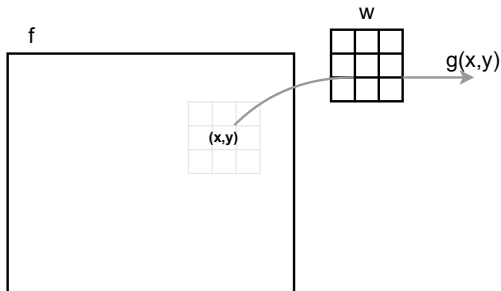


■ Operace s okolím



Filtrování

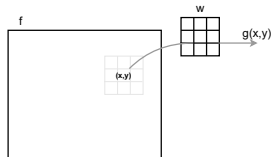
- **Operace s okolím**,
neighbourhood operations
- **Vstupní obraz f**
- **Výstupní obraz g**
- nahrazujeme hodnotu pixelu $f(x, y)$ výsledkem funkce aplikované na pixely v okolí
- **Filtr** (maska) w – určuje okolí
 - Lineární filtry
 - Nelineární filtry



Pojem filtr pochází ze zpracování obrazu ve frekvenční doméně, které se budeme věnovat později.

Lineární filtry

- w velikosti 3×3
- $g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(1, 1)f(x + 1, y + 1)$
- **Obecně** filtr velikosti $m \times n$ ($m = 2a + 1$ a $n = 2b + 1$)
- $g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t)$
- **zero padding**: zvětšení o a na obou koncích ve směru x a o b ve směru y



Korelace a konvoluce

- **Korelace** = postup posunu masky po obraze a součet prvků pod maskou vynásobených koeficienty v masce
- $w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$
- **Konvoluce** = korelace s maskou otočenou o 180°
- $w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$
- oba pojmy opět pocházejí z frekvenční domény
 - Korelace je vhodná pro hledání podobností, šablon (template matching)
 - Konvoluce je v matematice i ve zpracování signálů to ta „správná“ definice filtrování
 - Většinou jsou filtry symetrické a je jedno, co použijeme

Příklad

$$f(x) \quad \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

$$w \quad \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline \end{array}$$

Korelace a konvoluce

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

1	2	3
---	---	---

g(x)

0

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

1	2	3
---	---	---

g(x)

0	3
---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

1	2	3
---	---	---

g(x)

0	3	2
---	---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

1	2	3
---	---	---

g(x)

0	3	2	1
---	---	---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

1	2	3
---	---	---

g(x)

0	3	2	1	0
---	---	---	---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

3	2	1
---	---	---

g(x)

0

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

3	2	1
---	---	---

g(x)

0	1
---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

3	2	1
---	---	---

g(x)

0	1	2
---	---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

3	2	1
---	---	---

g(x)

0	1	2	3
---	---	---	---

f(x)

0	0	0	1	0	0	0
---	---	---	---	---	---	---

w

3	2	1
---	---	---

g(x)

0	1	2	3	0
---	---	---	---	---

Korelace a konvoluce

Padding

- pro efektivní práci s konvolucí a korelací je vhodné obraz rozšířit
- nejčastěji se okraje doplňují o 0 = **zero padding** (viz předchozí příklad)
 - Výhody: rychlé, snadná implementace
 - Nevýhody: okraje obrázku jsou tmavé



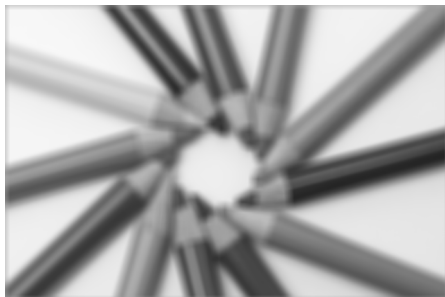
Příklad

Pokud máme velikost filtru $a \times a$, o kolik pixelů musíme obrázek zvětšit?

Korelace a konvoluce

Padding

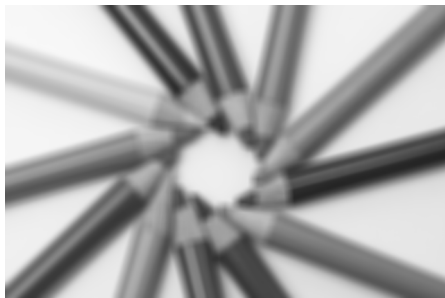
- **constant padding** – okraje se doplní konstantou (ne nutně nulou)
 - Výhody: flexibilita, doplnění například průměrnou jasovou hodnotou
 - Nevýhody: může působit nepřirozeně, pokud okraje nejsou průměrně jasné



Korelace a konvoluce

Padding

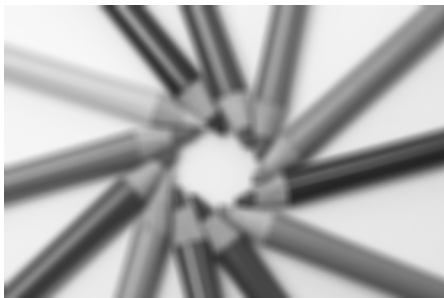
- **replicate padding** – okraje se doplní hodnotu nejbližšího krajního pixelu
 - Výhody: nevytváří se konstantní okraj okolo obrazu
 - Nevýhody: může tvořit silné pruhy okolo okraje



Korelace a konvoluce

Padding

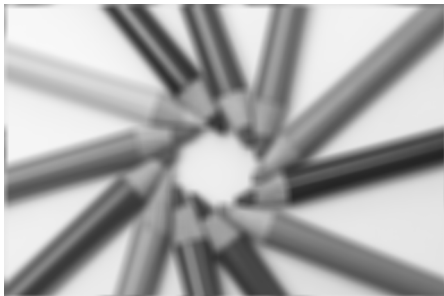
- **symetric padding** (mirror/reflect) – okraje se doplní jako zrcadlový odraz obrazu
 - Výhody: hladký přechod, nevytváří se artefakty
 - Nevýhody: může narušit periodicitu při některých typech zpracování



Korelace a konvoluce

Padding

- **circular padding** – okraje se doplní hodnotami z protilehlé strany obrazu (jako by byl obraz na toru)
 - Výhody: přirozené pro periodické signály, konzistence s Fourierovou transformací
 - Nevýhody: nevhodné pro běžné fotografie – objekty se „napojují“ z druhé strany



Korelace a konvoluce

Filtrování více filtry

- předpokládejme, že chceme aplikovat na obrázek filtr w_1 a na výsledek w_2, \dots
- konvoluce je komutativní, můžeme vše filtrovat najednou filtrem w
Konvoluce je asociativní a komutativní, takže matematicky nezáleží na pořadí jader; v praxi ale pořadí může ovlivnit okraje (padding) nebo numerickou přesnost.
- $W = w_1 \star w_2 \star \dots \star w_n$
- velikost nového filtru se postupně zvětšuje (nevracíme ořezaný filtr)

Separabilita filtrů

- 2D filtr nazýváme **separabilní**, pokud ho můžeme rozložit na součin dvou jednorozměrných filtrů
$$w(x, y) = w_1(x)w_2(y)$$

Příklad

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} [1 \quad 1 \quad 1] = w_1 w_2^T$$

- výpočet je jednodušší – rychlejší
- máme obrázek velikosti $M \times N$ a filtr velikosti $m \times n$
- konvoluce potřebuje $MNmn$ násobení a sčítání
- u separabilních filtrů nejprve aplikujeme jeden – MNm násobení a sčítání
- poté aplikujeme druhý – MNn násobení a sčítání
- celkem tedy $MN(m + n)$

Příklad

Spočítejte úsporu času (počet násobení a sčítání) pro filtr velikosti 11×11 .

Separabilita filtrů

- z teorie matic víme, že matice je separabilní, pokud je její hodnost rovna 1
- hodnost matice = dimenze prostoru generovaného řádky (resp. sloupci) matice (počet lineárně nezávislých řádků)

Příklad

Určete, zda jsou následující filtry separabilní.

$$W_1 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Co tyto filtry představují za operace si ukážeme později.

Separabilita filtrů

- Jak spočítáme na jaké vektory můžeme matici rozložit?
 - 1 vybereme si nenulový prvek matice (označme si jeho hodnotu E)
 - 2 jako vektor w_1 vezmeme sloupec příslušející k vybranému prvku
 - 3 jako vektor w_2^T vezmeme řádek příslušející k vybranému prvku jehož prvky vydělíme E

Příklad

Pro následující matici nalezněte vektory, které jí rozkládají.

$$W_1 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Vyhlazovací filtry – Průměrování

- **Box filtry** – prvky matice rovny $1/(m \cdot n)$

- maska 3×3
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- **Vážené průměrování** – prvky blíže mají vyšší váhu

- maska 3×3
$$\begin{bmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{bmatrix}$$

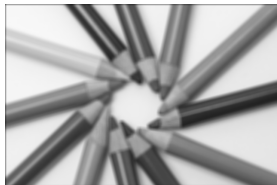
- **Gaussovský filtr** – prvky kopírují Gaussovskou křivku

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

Hodnoty se normalizují.



Maska velikosti 3×3 .



Maska velikosti 10×10 .

Vyhlazovací filtry – Průměrování

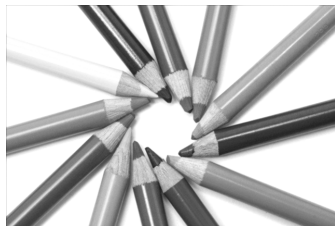
- Důležitá poznámka, hodnoty v matici jsou vždy normalizované (součet koeficientů je roven 1)
- Proč?



Normalizované váhy



Součet vah < 1

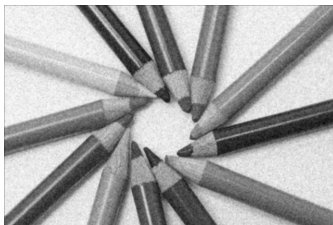
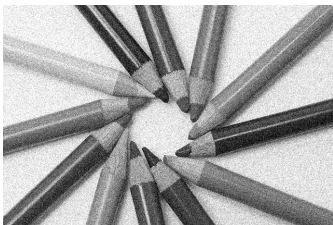


Součet vah > 1

Vyhlazovací filtry

Příklady využití

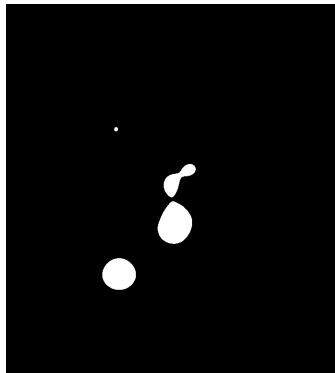
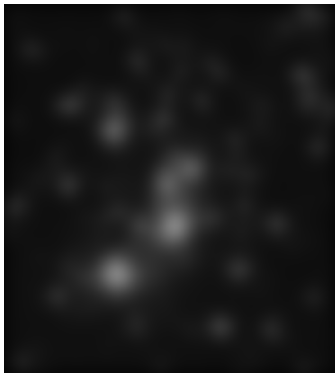
- Minimalizace šumu



Vyhlazovací filtry

Příklady využití

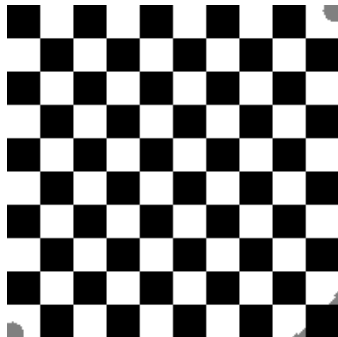
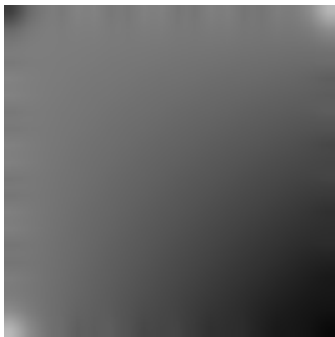
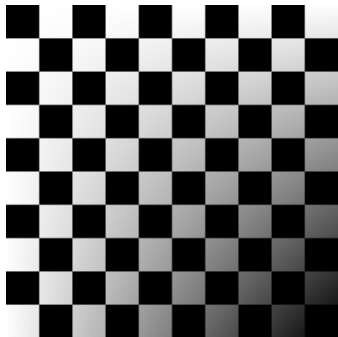
- V kombinaci s prahováním hledání větších světlých oblastí



Vyhlazovací filtry

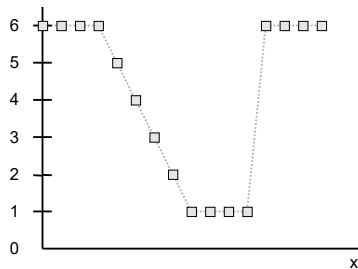
Příklady využití

- Odhad stínovací funkce



Ostřicí filtry

- založené na derivaci – 1. a 2. derivace
- **konstantní, přechod, skok**

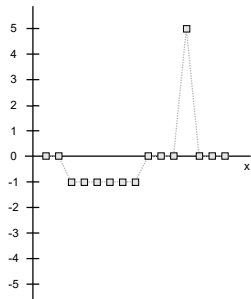
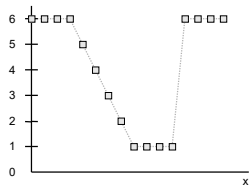


1. derivace

- je rovna 0 v oblastech s konstantní intenzitou
- nenulová na počátku na přechodech a skocích
- nenulová podél přechodu

■ **Příklad:**

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

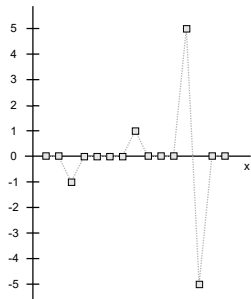
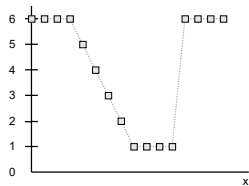


2. derivace

- je rovna 0 v oblastech s konstantní intenzitou
- nenulová na počátku a na konci přechodů a skoků
- nulová podél přechodu

■ **Příklad:**

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



Laplaceův filtr

- **Vertikální:** $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$
- **Horizontální:** $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$
- **Dohromady:** $\nabla^2 = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$

- **Filtr:**

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- **Diagonální:** $f(x+1, y-1) + f(x-1, y+1) - 2f(x, y)$ a $f(x-1, y-1) + f(x+1, y+1) - 2f(x, y)$

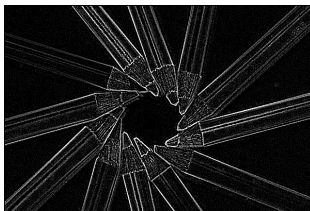
- **Filtr:**

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

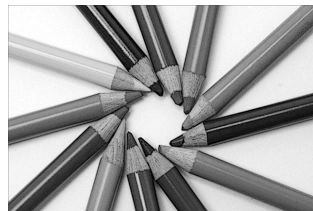
Laplaceův filtr



Původní obraz.



Hrany v obraze.



Zvýrazněné hrany.

Robertsův filtr

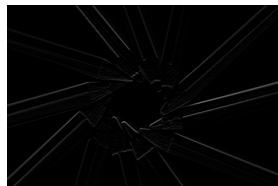
- diagonální hrany
- $\nabla f(x, y) = (f(x, y) - f(x - 1, y)) + (f(x, y + 1) - f(x + 1, y))$

- **Filtr:**

$$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$



Původní obraz.



Hrany v obraze.

Sobelův filtr

- **Vodorovné hrany:**

- $\frac{\partial f}{\partial x} = (f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)) - (f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1))$

- **Filtr:**

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- **Svislé hrany:**

- $\frac{\partial f}{\partial y} = (f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)) - (f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1))$

- **Filtr:**

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

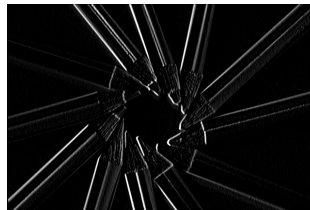
Sobelův filtr



Původní obraz.



Vodorovné hrany v obraze.



Svislé hrany v obraze.

Prewittové filtry

- **Vodorovné hrany:**

- $\frac{\partial f}{\partial x} = (f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)) - (f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1))$

- **Filtr:**

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- **Svislé hrany:**

- $\frac{\partial f}{\partial y} = (f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)) - (f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1))$

- **Filtr:**

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

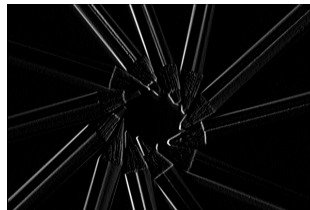
Prewittové filtr



Původní obraz.



Vodorovné hrany v obraze.



Svislé hrany v obraze.

Nelineární filtry

- nelze je zadat maskou
- volí se velikost okolí, na pixely v okolí je aplikována nelineární operace
- $g(x, y)$ a okolí 3×3 , vezmeme hodnoty v okolí $f(x, y) - f(x - 1, y - 1), f(x - 1, y), \dots, f(x + 1, y + 1)$

Percentilové filtry (statistické)

- 0. percentil – nejmenší hodnota v okolí (*min filtr*)
- 50. percentil – střední hodnota v okolí (*median filtr*)
- 100. percentil – největší hodnota v okolí (*max filtr*)

Percentilové filtry



Původní obraz.

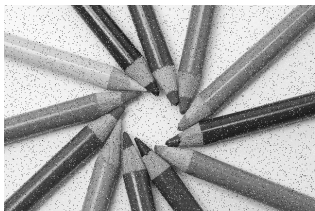


Min filtr.

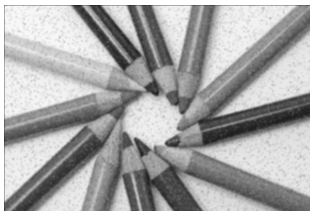


Max filtr.

Mediánový filtr



Původní obraz.



Filtrování průměrováním.



Mediánová filtrace.