



Úvod do analýzy obrazu

Pokročilá analýza obrazu

Mgr. Markéta Trnečková, Ph.D.

Obraz

- **Zrak** (Human vision)

- pomáhá lidem vnímat svět okolo a rozumět mu

- **Počítačové vidění** (Computer vision)

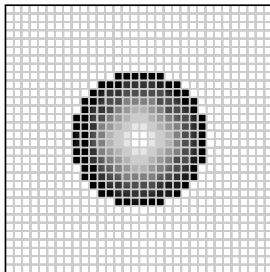
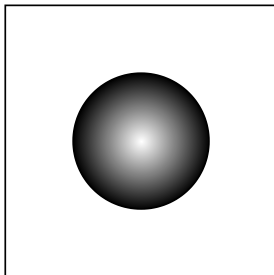
- napodobuje lidský zrak
- vstupem je digitální obraz, výstupem je nějaká smysluplná informace

Digitální obraz

- **Matematický model:** $g = f(x, y)$ (obrazová funkce)
- **Prostorové souřadnice:** $x \in \langle x_{min}, x_{max} \rangle$ a $y \in \langle y_{min}, y_{max} \rangle$
- **Hodnoty funkce g :**
 - jedno číslo
 - trojice (čtveřice) čísel
 - celá kolekce dat
- **Rastr**

Digitalizace

- Převod spojitého světa (spojité obrazové funkce) do rastru
- **Digitalizace:**
 - vzorkování
 - kvantování



Digitální obraz

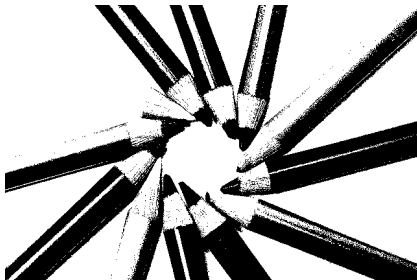
Příklad

Co všechno se ztratí při převodu reálného světa do digitálního obrazu reprezentovaného funkcí $f(x, y)$?

Typy obrazů

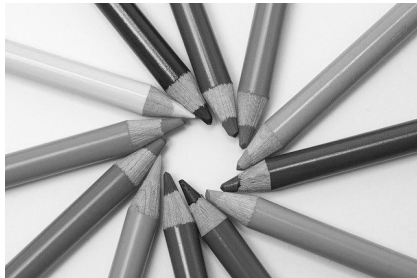
- **Binární obraz**

- hodnoty 0 nebo 1



- **Šedotónový obraz / Monochromatický obraz** (gray scale)

- 1 hodnota – většinou 1 byte (0 do 255)



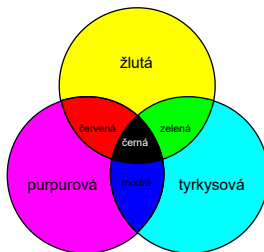
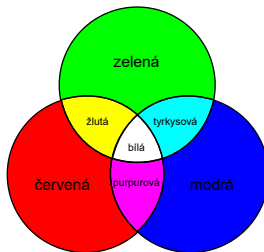
Typy obrazů

- **Barevný obraz**
- pixel = trojice (nebo čtveřice) hodnot
- velikost:
 - low color (15 bit)
 - high color (16 bit)
 - true color (24 bit)
 - super true color (32 bit)
 - deep color (48 bit)



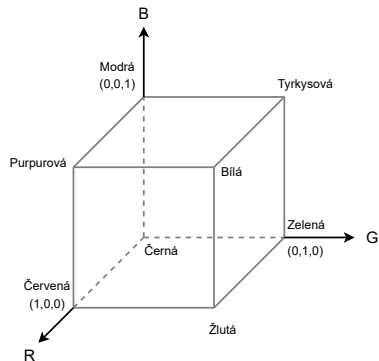
Barevné modely

- Základ v lidském vizuálním systému
- **Primární barvy:**
 - červená (red) – 700 nm
 - zelená (green) – 546.1 nm
 - modrá (blue) – 435.8 nm
- **Aditivní skládání barev**
- **Sekundární barvy** (primární barvy pigmentů):
 - tyrkysová (cyan)
 - purpurová (magenta)
 - žlutá (yellow)
- **Subtraktivní skládání barev**
- **Barevný prostor**, barevný model



Barevný model RGB

- **3 složky:** červená (R), zelená (G) a modrá (B)
- **Barvy:** body (r, g, b)
- Rozšíření **RGBA**



Barevný obraz



Červená složka



Zelená složka



Modrá složka

Barevný model RGB – stupně šedi

Průměr ze složek

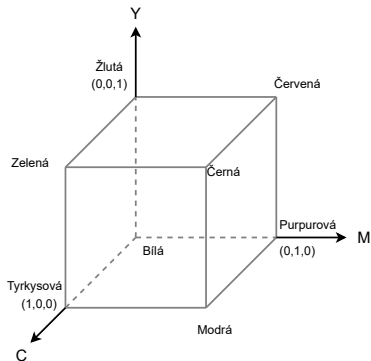


$$I = 0.299R + 0.587G + 0.114B$$

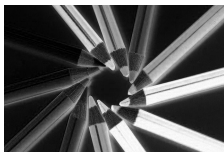


Barevný model CMY

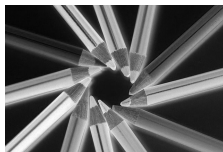
- **3 složky:** tyrkysová (C), purpurová (M) a žlutá (Y)
- **Barvy:** body (c, m, y)
- Rozšíření **CMYK**



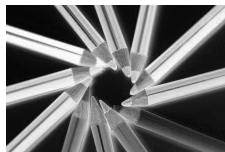
Barevný obrázek



Tyrkysová složka



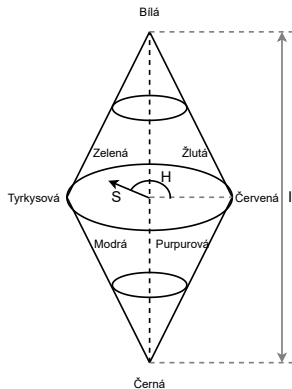
Purpurová složka



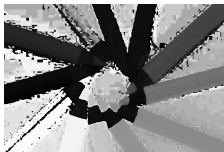
Žlutá složka

Barevný model HSI

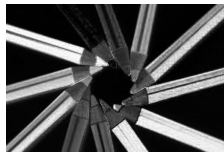
- Intuitivnější pro člověka
- **3 složky**: odstín barvy (hue H), saturace – čistota barvy (S) a intenzita (I)
- **Barvy**: body (h, s, i)



Barevný obraz



Hue



Saturation



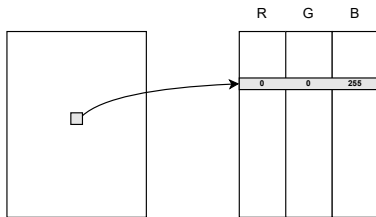
Intensity

Další modely

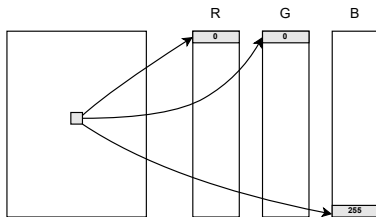
- **Podobné HSI:** HSV, HSL
- **Oddělení barevné a jasové složky:** YCbCr
- **Nezávislé na zobrazovacím zařízení:** CIE

Typy obrazů

- **Indexový obraz**
- jeden index do palety
- pseudo color



- **Indexový obraz**
- 3 indexy
- direct color



Příklad

V čem spočívá výhoda a nevýhoda indexovaného (pseudo color) obrazu oproti barevnému obrazu?

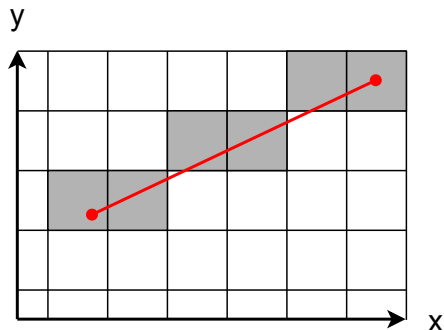
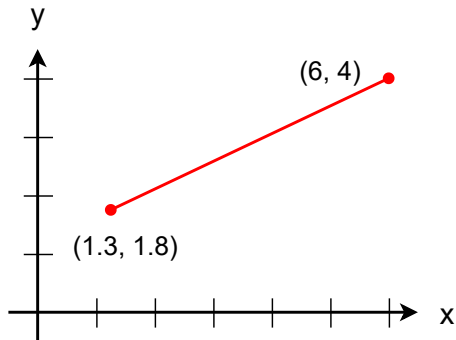
Zpracování digitálního obrazu

- Akvizice
- Vylepšení obrazu
 - subjektivní vylepšení
 - úprava jasů, zvýraznění detailů a pod.
- Rekonstrukce obrazu
 - vylepšení obrazu, objektivní na základě matematických principů
 - například potlačení šumu
- Získání informace z obrazu
- Komprese

Příklad

Máme obraz velikosti 1024×683 pixelů. Každý je reprezentován 3 byty. Kolik zabírá celkem paměti?

Spojité a rastrové zobrazení úsečky



Otázka je, jak zjistit, zda pixely tvoří souvislý objekt, nebo ne.

Vztahy mezi pixely

Sousednost

- pixel p ($[x, y]$)
 - **4-sousedé** ($N_4(p)$) – $[x-1, y]$, $[x+1, y]$, $[x, y-1]$ a $[x, y+1]$
 - **8-sousedé** ($N_8(p)$) – $N_4(p)$ a $[x-1, y-1]$, $[x-1, y+1]$, $[x+1, y-1]$ a $[x+1, y+1]$
 - **m-sousedé** ($N_m(p)$) – q , pro které platí $q \in N_4(p)$ nebo $q \in N_8(p)$, ale neexistuje r takové, že $r \in N_4(p) \wedge r \in N_4(q)$
mixed adjacency
-
- pixely p , q , definovaná množina intenzit
 - **4-sousední** – $q \in N_4(p)$
 - **8-sousední** – $q \in N_8(p)$
 - **m-sousedé** – $q \in N_m(p)$
-
- **Digitální cesta** z pixelu p ($[x, y]$) do q ($[s, t]$) – sekvence sousedních pixelů
 - **4-cesta**, **8-cesta**, **m-cesta**

Vztahy mezi pixely

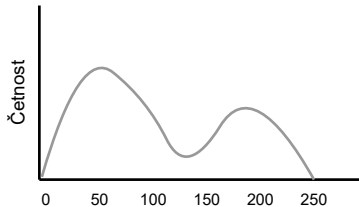
- **Spojité** pixely $p ([x, y])$, $q ([s, t])$ – existuje cesta
- **4-spojité, 8-spojité, m-spojité**
- **Komponenta** – množina spojitých pixelů
- **4-komponenta, 8-komponenta, m-komponenta**

- **Oblast** (region) – množina pixelů, která je tvořena jednou komponentou (bereme v úvahu pouze 4- a 8- komponenty)
- **Sousední oblasti** – oblasti, jejichž sjednocení tvoří oblast
- **Disjunktní oblasti** – oblasti, které nejsou sousední

Jak popisovat oblasti, jejich okraje a podobné se budeme věnovat více později

Histogram

- **Histogram:** $h(r_k) = n_k$

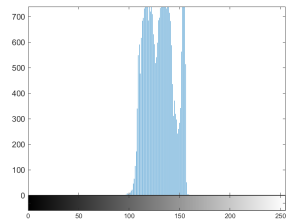
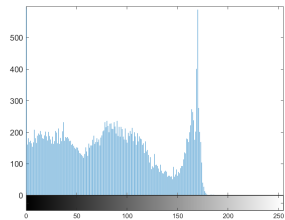
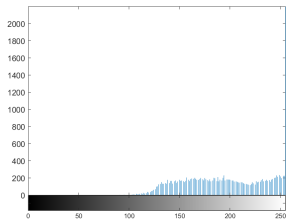


- **Normalizovaný histogram:** $p(r_k) = h(r_k)/M \cdot N$
 - **Kumulativní histogram:** $H(r_k) = \sum_{j=0}^k h(r_j)$
- Co takový histogram představuje?**

Příklad

Mohou mít dva různé obrázky stejný histogram?

Histogram



C

Operace s obrázky

■ Aritmetické operace (prvek po prvku)

- sčítání
- odčítání
- násobení
- dělení

■ Množinové operace

- univerzum Ω
- prvek množiny $a \in A$
- prázdná množina \emptyset
- podmnožina $B \subseteq A$
- komplement A^C
- průnik množin $A \cap B$
- sjednocení množin $A \cup B$
- rozdíl $A - B$

■ Logické operace

- pravda (true, nebo 1) a nepravda (false, 0)
- binární obrázky
- **negace** (NOT), **logický součin** (AND), **logický součet** (OR) a **exkluzivní součet** (XOR)

Operace

Součet



Obraz 1.



Obraz 2.



Morfing.



Jeden z obrazů obsahující šum.



Zprůměrovaný obraz.

Operace

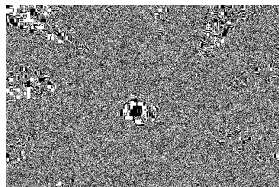
Rozdíl



Původní obraz.



Upravený obraz.



Rozdíl.

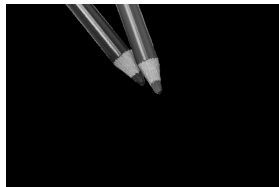
Součin



Původní obraz.



ROI.



Součin ROI a obrazu.

Operace

Součin



Původní obraz.



Obraz stínu.



Součin stínu a obrazu.

Podíl



Obraz obsahující stín.



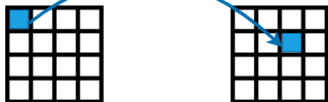
Obraz stínu.



Obraz bez stínu.

Úpravy obrazu

■ Geometrické transformace



■ Jasové transformace



■ Operace s okolím



Geometrické transformace

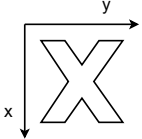
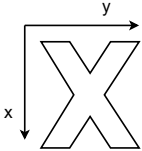
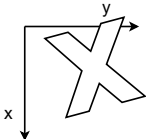
- transformace souřadnic: $(x, y) = T(v, w)$
- **afinní transformace**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

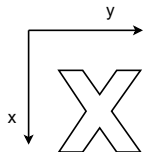
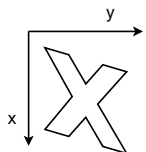
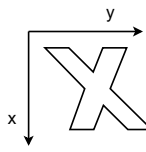
- výpočet nových souřadnic

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

Geometrické transformace

Název	Afinní matice	Rovnice	Ukázka
Identita	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Zvětšení	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Otočení	$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	

Geometrické transformace

Název	Afinní matice	Rovnice	Ukázka
Posunutí	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Zkosení (vertikální)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Zkosení (horizontální)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Geometrické transformace – aplikace

■ Dopředné mapování

– $(x, z) = T^1(v, w)$

■ Zpětné mapování

– $(v, w) = T^{-1}(x, y)$

Příklad

Máme funkci $(x, z) = T^1(v, w) = (3v, 2w)$. Jakou transformaci tato operace představuje? Jak by vypadala funkce zpětného mapování? Jak by afinní matice tohoto zobrazení?

Příklad

Máme funkci $(x, z) = T^1(v, w) = (v + 0.4w, w)$. Jakou transformaci tato operace představuje? Jak by vypadala funkce zpětného mapování? Jak by afinní matice tohoto zobrazení?

Interpolace – demonstrace

- obrázek 1×6 pixelů chceme ho zvětšit $1.5 \times$ v ose x



- zvětšení $1.5 \times$ v ose x (výsledný obrázek 1×9)



- zvětšení pixelů na původní velikost – výběr barvy
 - nejbližší soused



- bilineární interpolace



- bikubická interpolace



Další témata spojená s geometrickými transformacemi

Registrace

- vstupní a referenční obraz
- geometrická transformace?
- **kontrolní body**

$$x = a_1 v + a_2 w + a_3 vw + a_4,$$

$$y = a_5 v + a_6 w + a_7 vw + a_8,$$

Warping

- Jednoúčelový
- Obecný

Warping – síť



Původní obraz.



Magnet.



Výsledný obraz.



Původní obraz.



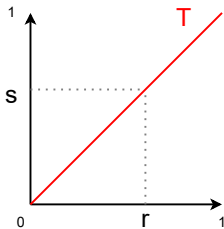
Modifikovaná síť.



Výsledný obraz.

Jasové transformace

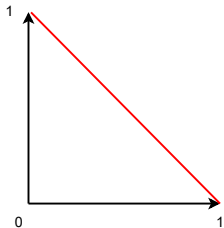
- $f(x, y)$ – jasová hodnota vstupního obrazu
- $g(x, y)$ – jasová hodnota výstupního obrazu
- **Transformační funkce T :**
 - $s = g(x, y)$ závisí na $r = f(x, y)$
 - $s = T(r)$
 - **lookup tabulka**



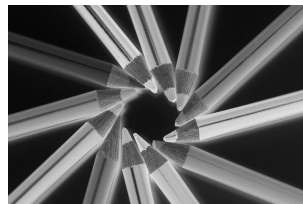
Příklady jasových operací

Negativ obrázku

$$T(r) = 1 - r$$



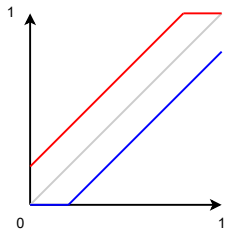
Původní obraz.



Negativ.

Změna jasu

$$T(r) = r + c$$



Zvýšení jasu.

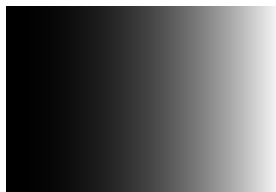
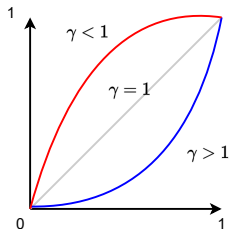


Snížení jasu.

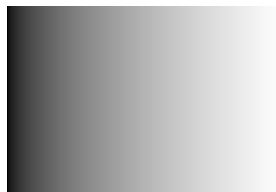
Příklady jasových operací

Gamma transformace

$$T(r) = c \cdot r^\gamma$$



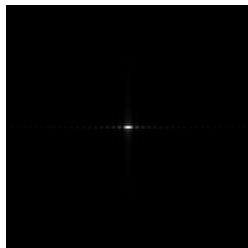
$\gamma > 1$



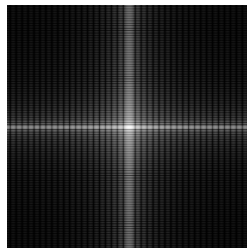
$\gamma < 1$

Logaritmická transformace

$$T(r) = c \cdot \log(1 + r)$$



Před aplikací.



Po aplikaci.

Příklady jasových operací

Změna kontrastu

$$T(r) = c \cdot r$$

$$T(r) = c \cdot (r + c_1) + c_2$$

Prahování

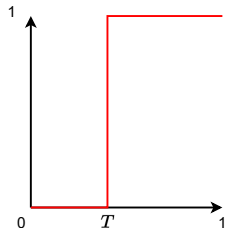
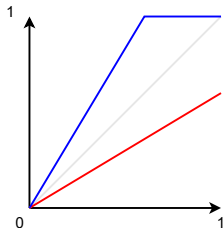
$$T(r) = \begin{cases} s_0 & \text{pro } r < \text{prah} \\ s_1 & \text{pro } r \geq \text{prah} \end{cases}$$

■ Volba prahu:

- experimentálně
- matematicky

■ Adaptivní prahování

■ Více se k němu vrátíme v segmentaci



Další probírané operace

- Ořezávání barevných intenzit
- Ořezávání bitů
- Operace založené na histogramu
 - Ekvalizace histogramu
 - Specifikace histogramu

Jasové operace barevných obrázků

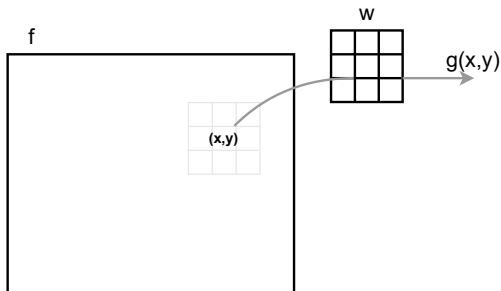
- $s_j = T_j(r_j)$,
- $j = 1, 2, \dots, n$ (n počet složek)
- Často
 - V **RGB** všechny složky
 - $s_j = r_j + k$
 - V **HSI** pouze jasová složka I
 - $s_i = r_i + k, s_j = r_j$, pro $j \in \{h, s\}$
- Operace mohou být stejné, nebo pro každou složku zvlášť
- Také jsme si ukazovali operace, kde výsledek jedné složky závisí i na hodnotách ostatních složek

Příklad

Jaká je výhoda provádění jasových transformací v HSI modelu oproti RGB?

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



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 $n = 2b + 1$)

- $g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t)$

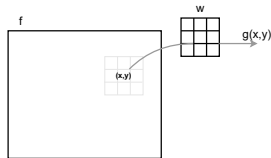
- **zero padding**: zvětšení o a na obou koncích ve směru x a o b ve směru y

- **Korelace** = 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°

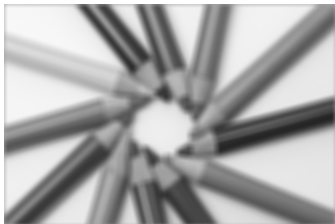
- $w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x - s, y - t)$



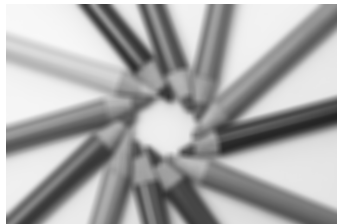
Padding



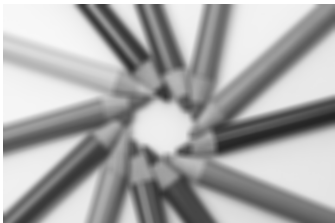
Zero padding



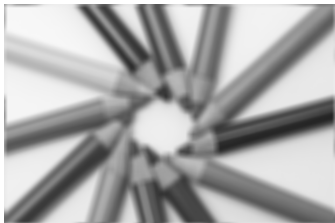
Constant padding



Replicate padding



Symetric padding



Circular padding

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
$$\begin{bmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{bmatrix}$$

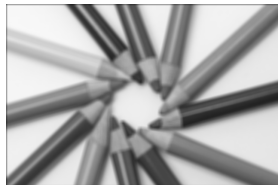
- **Gaussovský filtr** – prvky kopírují Gaussovskou křivku

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

Hodnoty se normalizují.



Maska velikosti 3×3 .



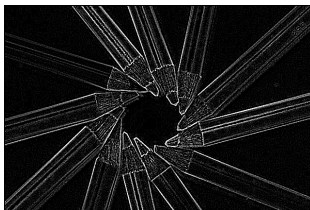
Maska velikosti 10×10 .

Ostřicí filtry

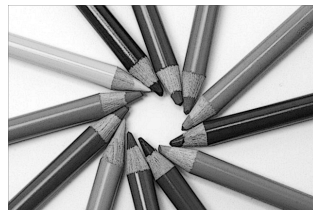
- založené na derivaci – 1. a 2. derivace
- 1. derivace: **Robertsův**, **Sobelův**, **Prewitové**
- 2. derivace: **Laplaceův**



Původní obraz.



Hrany v obraze.



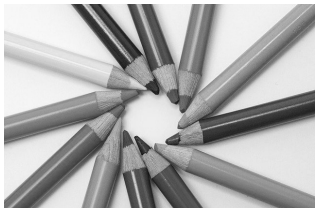
Zvýrazněné hrany.

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*)



Původní obraz.



Min filtr.



Max filtr.

Filtrování barevných obrázků

- nepracujeme s intenzitami, každý pixel = vektor hodnot
- Často:
 - V RGB filtrujeme zvlášť jednotlivé složky
 - V HSI filtrujeme jen I složku

Průměrování



Původní obraz.



Filtrace v RGB.



Filtrace v HSI.

Co více jsme probírali?

- Redukce počtu barev v obraze
- Matematická morfologie
 - Binárních obrázků
 - Šedotónových obrázků
- Frekvenční doména – budeme se jí věnovat více