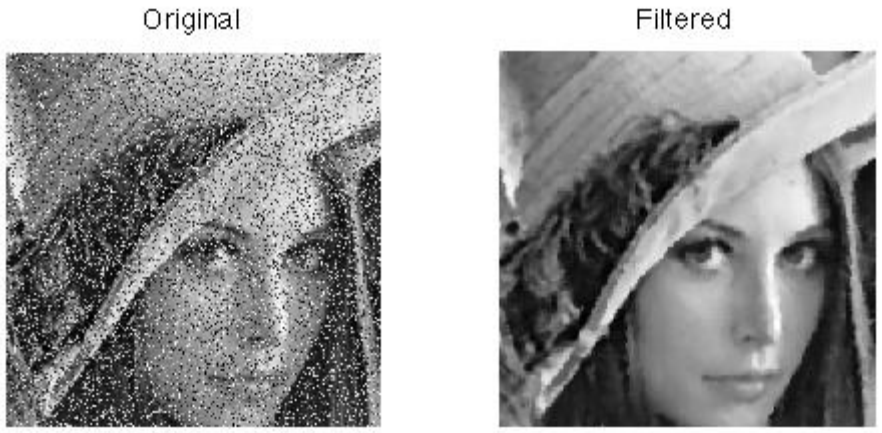
Filtrace v prostorové a frekvenční oblasti

Filtrace v prostorové a frekvenční oblasti

- Filtrace
  - soubor lokálních transformací obrazu
  - úrovně jasu se převádí na jiné
  - cílem je potlačit nežádoucí jasové složky



- Filtrace
  - v prostorové (obrazové) nebo časové oblasti - využití konvoluce
  - ve frekvenční oblasti – využití Fourierovy transformace

Filtrace v prostorové oblasti

- Filtry
  - Pracují s více pixely v obraze (nejsou zaměřeny na jeden pixel), tj. počítají novou hodnotu pixelu na základě hodnot více pixelů v obraze.
  - Díky tomu umožňují takové operace, jako je např. zaostření nebo rozmazání obrazu
  - Filtry potlačují část neužitečné informace a ponechávají (zdůrazňují) tu užitečnou.
  - Filtrace nebývá vratnou operací, tj. dochází díky ní ke ztrátě informace.

- Příklad průměrovacího filtru (rozmazání)

$$I'(u,v) \leftarrow \frac{p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{9}$$

  - Kde p_i jsou sousední pixely středového pixelu p_0.

Filtrace v prostorové oblasti

- Jiný zápis téhož, ale pomocí relativních odkazů na sousední pixely

$$I'(u,v) \leftarrow \frac{1}{9} \cdot [\, I(u-1,v-1) + I(u,v-1) + I(u+1,v-1) + I(u-1,v) + I(u,v) + I(u+1,v) + I(u-1,v+1) + I(u,v+1) + I(u+1,v+1) \,].$$

- resp. v kompaktní formě:

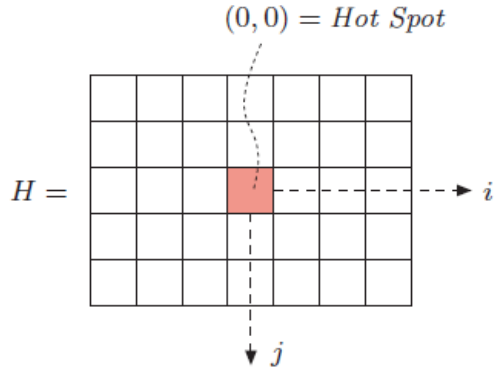
$$I'(u,v) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)$$

- (Obrazový) filtr je charakterizován zejména
  - množinou pixelů, které zohledňuje ve výpočtu
  - velikostí této množiny
  - tvarem oblasti, kterou tyto pixely vymezují (nemusí být spojitá)
  - vahami (mírou vlivu) jednotlivých pixelů

Filtrace v prostorové oblasti

- Lineární filtry
  - Lineární filtry jsou postaveny na váženém součtu hodnot uvažovaných pixelů
  - Velikost, tvar a váhy jsou určeny tzv. filtrační maticí, resp. filtrační maskou $H(i, j)$
  - Např. zmíněný průměrovací filtr velikosti 3x3 má masku:

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



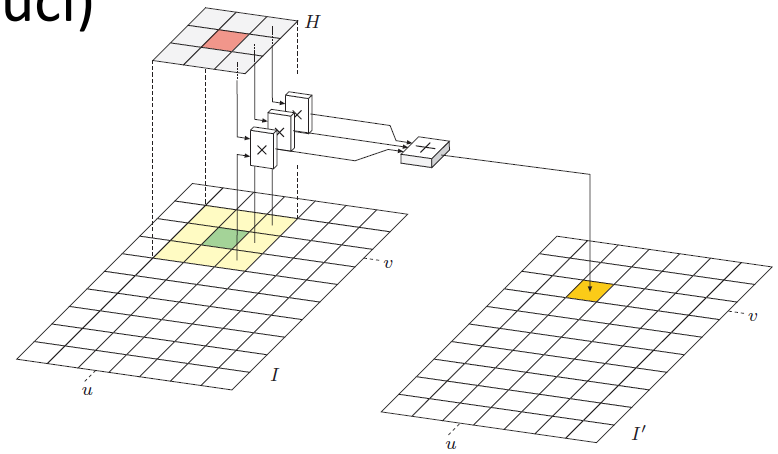
  - Filtr sám je dvourozměrnou diskrétní funkcí $H: Z \times Z \rightarrow R$ (pozice: $Z \times Z$, míra vlivu: R)
  - Filtr bývá zvykem indexovat od jeho středu.
  - Pro matematické operace se uvažuje automatické doplnění nulami okolo filtru.
  - Pro výpočty je výhodné mít v matici celá čísla.
  - Lineární filtry mají svá omezení při vyhlazování a odstraňování šumu (rozmazání).

Filtrace v prostorové oblasti

- Aplikace filtru
  - Maska filtru (dále jen filtr) je umístěna na uvažovanou pozici v obrázku.
  - Všechny hodnoty filtru *H*(*i, j*) jsou pronásobeny s odpovídající hodnotami v obraze *I*(*u*+*i, v*+*j*) a následně sečteny.
  - Výsledná hodnota je uložena do nového obrázku (nejsou přepisovány hodnoty ve stávajícím obrazu) a celý proces se opakuje pro další pixely v obrazu
  - Proces aplikace filtru lze vyjádřit vztahem (tzv. lin. konvolucí)

$$I'(u,v) \leftarrow \sum_{(i,j)\in R_H} I(u+i, v+j) \cdot H(i,j)$$

$$I'(u,v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i,j)$$

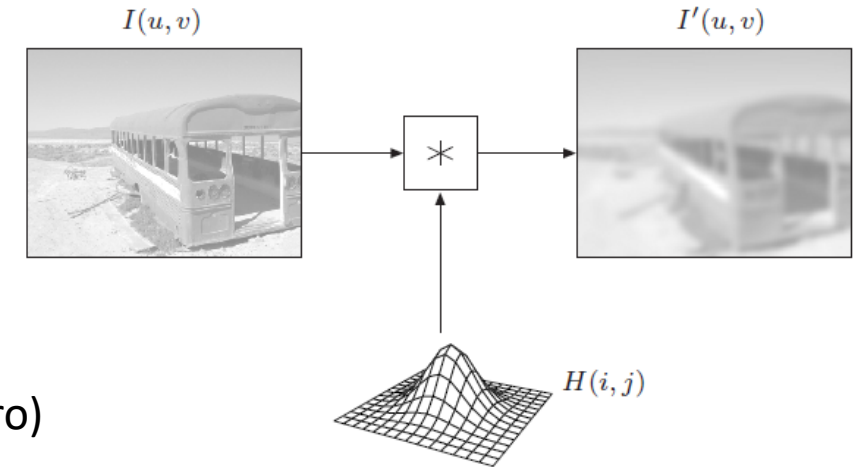


Filtrace v prostorové oblasti

- (Lineární) konvoluce
  - Kombinace (pronásobení) dvou diskrétních nebo spojitých funkcí mezi sebou při různém vzájemném překryvu a následné sečtení

$$I * H = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u-i, v-j) \cdot H(i,j)$$

symbol konvoluce



H – konvoluční matice (konvoluční jádro)

Filtrace v prostorové oblasti

- Vlastnosti konvoluce
  - Komutativita $I * H = H * I$
  - Linearita $(s \cdot I) * H = I * (s \cdot H) = s \cdot (I * H)$
  - Distributivita $(I_1 + I_2) * H = (I_1 * H) + (I_2 * H)$
  - Asociativita $A * (B * C) = (A * B) * C$
  - Separabilita $H = H_1 * H_2 * \ldots * H_n$

$$H_x = \begin{bmatrix} 1 & 1 & \mathbf{1} & 1 & 1 \end{bmatrix} \qquad H_y = \begin{bmatrix} 1 \\ \mathbf{1} \\ 1 \end{bmatrix}$$

Ušetření počtu operací, než u výpočtu s maticí

$$I' \leftarrow (I * H_x) * H_y = I * \underbrace{(H_x * H_y)}_{H_{xy}}$$

$$H_{xy} = H_x * H_y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

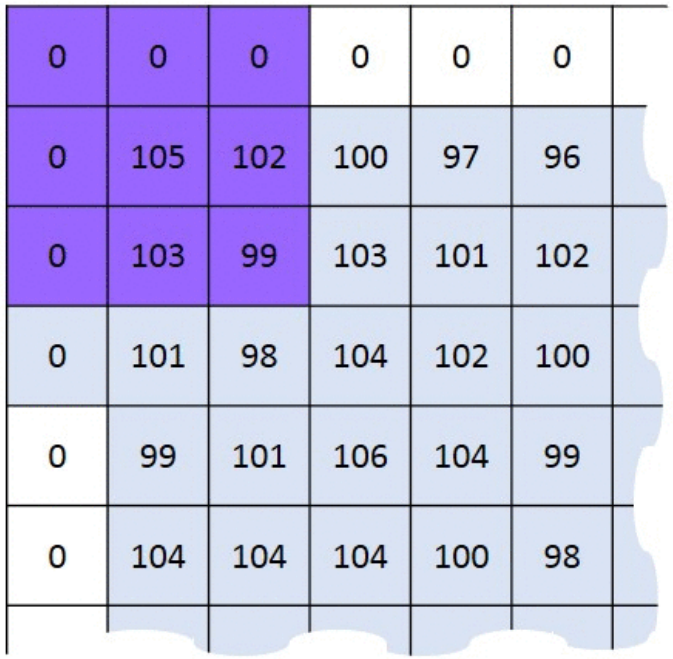
Filtrace v prostorové oblasti

- Ne každý filtr (reprezentovaný maticí) lze rozložit na dva dílčí filtry reprezentované vektory.
- Pokud má filtr (matice) hodnost 1, potom ho lze rozložit na součin dvou vektorů (pomocí SVD) a příslušný filtr je tak separovatelný.
- Separabilita sníží výpočetní složitost z $O(MNmn)$ na $O(MN(m + n))$, kde M, N jsou rozměry obrázku a m, n jsou rozměry filtru.

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} \\ h_1 \\ \\ \end{bmatrix} \begin{bmatrix} \quad h_2 \quad \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

```
[u,s,v] = svd(H);
s = diag(s); % prevod matice na vektor diagonalnich hodnot
hodnost = sum(s > 0); % nebo primo prikazem rank(H)
if (hodnost == 1)
   hcol = u(:,1) * sqrt(s(1));
   hrow = conj(v(:,1)) * sqrt(s(1));
else
   % filtr neni separabilni
end
```

Filtrace v prostorové oblasti



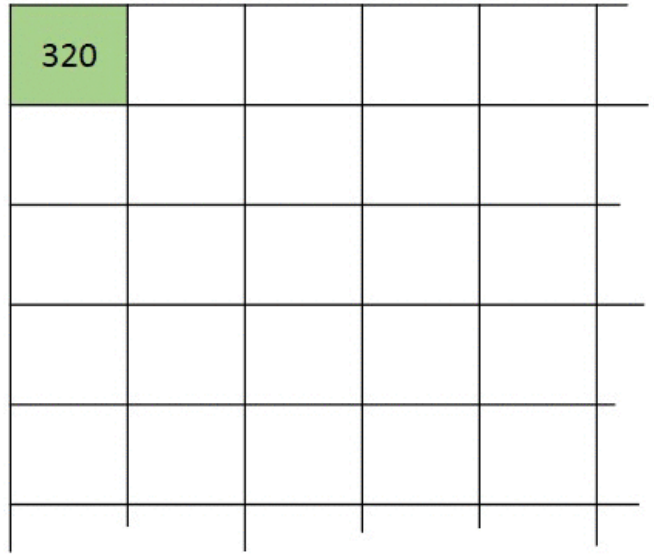
0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

$$0*0+0*-1+0*0$$
$$+0*-1+105*5+102*-1$$
$$+0*0+103*-1+99*0 = 320$$

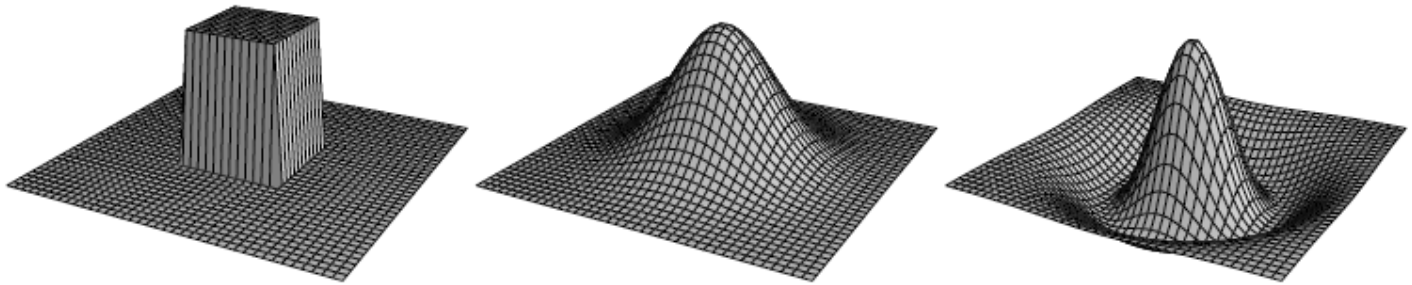


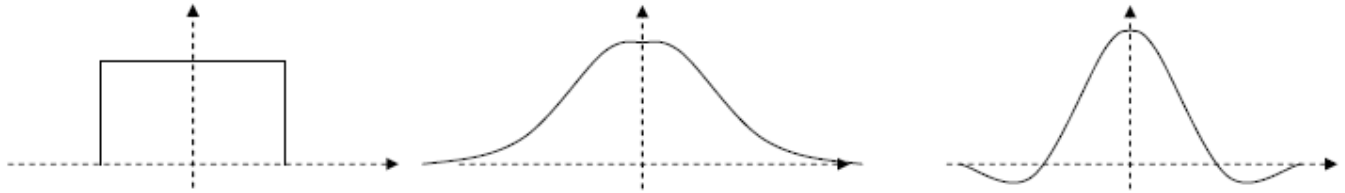
320					

Output Matrix

Filtrace v prostorové oblasti

- Tvary/typy filtrů (konvolučních jader)





0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Krabicový filtr

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Gaussův filtr

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Laplaceův filtr

Filtrace v prostorové oblasti

- Vyhlazovací filtry
  - Krabicový (průměrovací) filtr (Mean/Box filter)
    - Jednoduchá implementace
    - Nevýhodou jsou ostré přechody -> zvlnění obrazu
    - Neizotropní (nechová se stejně ve všech směrech)

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

  - Gaussův filtr (Gaussian filter)
    - Izotropní
    - „měkké“ okraje, bez zvlnění obrazu

$$G_\sigma(x,y) = e^{-\frac{r^2}{2\sigma^2}} = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

box filter gaussian

https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

Filtrace v prostorové oblasti

- Diferenční filtry
  - Některé koeficienty filtrační matice mohou být záporné
  - Výpočet lze interpretovat jako rozdíl dvou součtů: součet hodnot pro kladné hodnoty filtru mínus součet hodnot pro záporné hodnoty filtru

$$I'(u,v) = \sum_{(i,j)\in R_H^+} I(u+i, v+j) \cdot |H(i,j)| - \sum_{(i,j)\in R_H^-} I(u+i, v+j) \cdot |H(i,j)|$$

  - Příkladem je např. Laplaceův filtr
  - Zatímco vyhlazovací filtry potlačují výrazné osamělé hodnoty, diferenční filtry je naopak zvýrazňují -> zaostřování obrázku, detekce hran

Filtrace v prostorové oblasti

- Nelineární filtry
  - Hodnoty pixelů jsou kombinovány pomocí nelineární funkce.

- Minimální a maximální filtr

$$I'(u,v) \leftarrow \min\{I(u+i,v+j) \mid (i,j) \in R\}$$
$$I'(u,v) \leftarrow \max\{I(u+i,v+j) \mid (i,j) \in R\}$$

Min. filtr (potlačení bílé, zvýraznění černé)

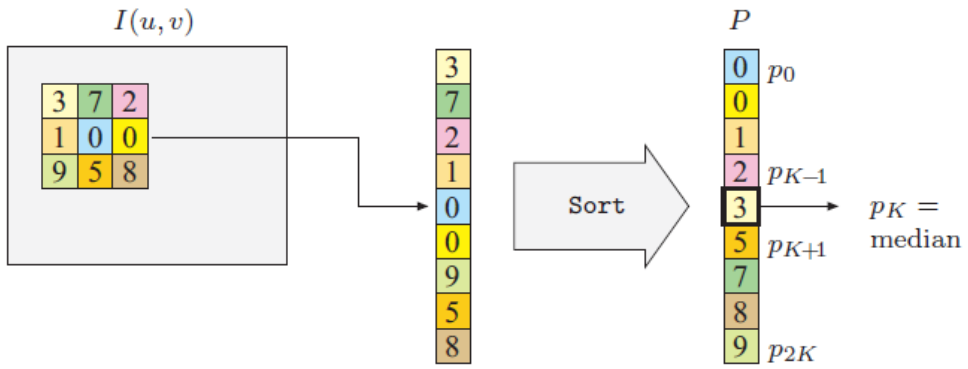
Max. filtr (potlačení černé, zvýraznění bílé)

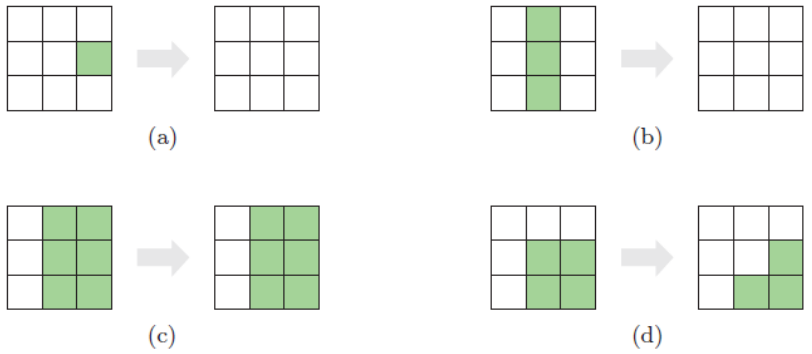
Př. Impulzní šum („pepř a sůl“)

Filtrace v prostorové oblasti

- Mediánový filtr
  - Nahrazuje hodnotu daného pixlu mediánem hodnot sousedních pixelů

$$I'(u,v) \leftarrow \text{median}\,\{I(u+i, v+j) \mid (i,j) \in R\}$$





  - Pro výpočet mediánu je potřeba provést seřazení hodnot → časově náročné.
  - Robustní – nerozhodí ho extrémní hodnoty.
  - Zachovává hrany, bohužel je někdy i vytváří.

Filtrace v prostorové oblasti

- Srovnání lineárního krabicového (průměrovacího) filtru 3x3 a mediánového filtru

Průměrovací lin. filtr 3x3

Mediánový filtr

Př. Impulzní šum („pepř a sůl“)

(a)

(b)

(c)

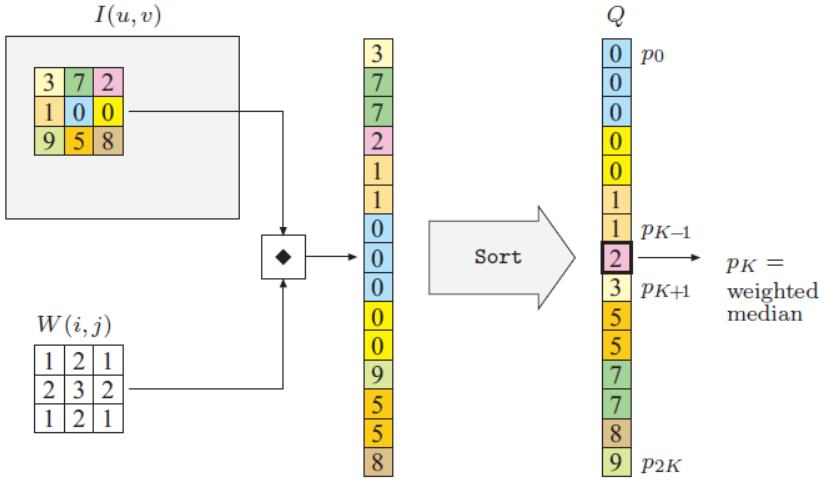
(d)

(e)

(f)

Filtrace v prostorové oblasti

- Vážený mediánový filtr
  - Jednotlivým pozicím ve filtru jsou přiřazeny váhy, tj. důležitosti jednotlivých hodnot
  - Většinou váhy blízko uvažovaného pixelu jsou větší, než vzdálenější pixely
  - Váhová matice $W(i,j) \in \mathbb{N}$
  - Výpočet probíhá tak, že každá hodnota obrazu je nakopírována do pomocného vektoru tolikrát, jaké je číslo na odpovídající pozici ve váhové matici.
  - Teprve poté je vypočten medián vektoru.



Filtrace v prostorové oblasti

- Hraniční oblasti
  - Problém: Jak počítat hodnoty na okrajích obrázku
  - Několik způsobů řešení:
    - Okrajové hodnoty doplnit nějakou konstantou, např. 0 = černá
    - Okrajové hodnoty doplnit hodnotami nejbližšího pixelu obrázku
    - Použít zrcadlově obrácený obraz
    - Periodicky rozšířit obraz (výhodné, pokud pracujeme ve frekvenční oblasti)

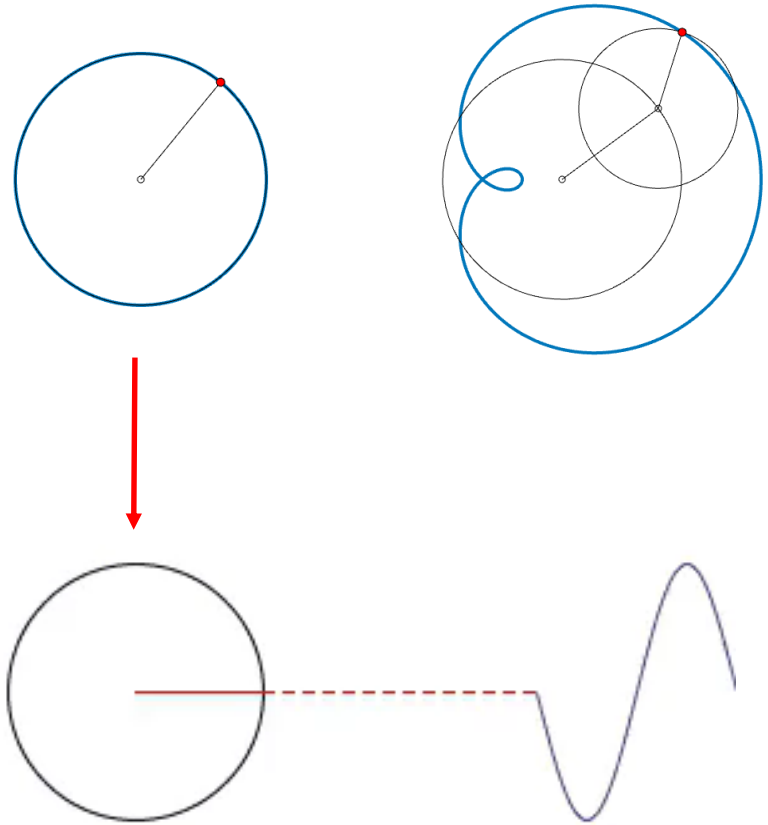
Filtrace ve frekvenční oblasti

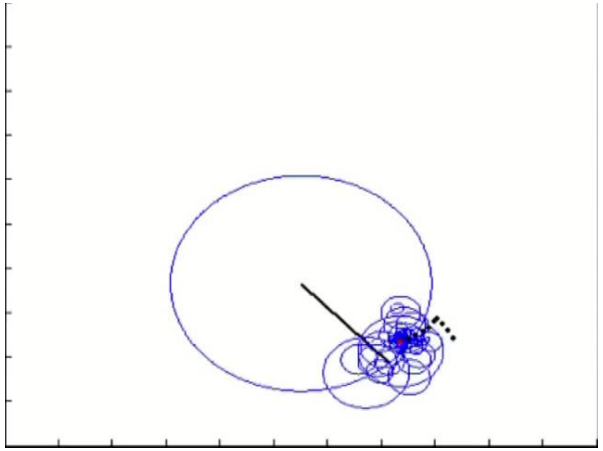
- Filtrace v prostorové oblasti
  - Snazší představivost, ale ne vždy výhodné, nemusí být vidět skryté závislosti

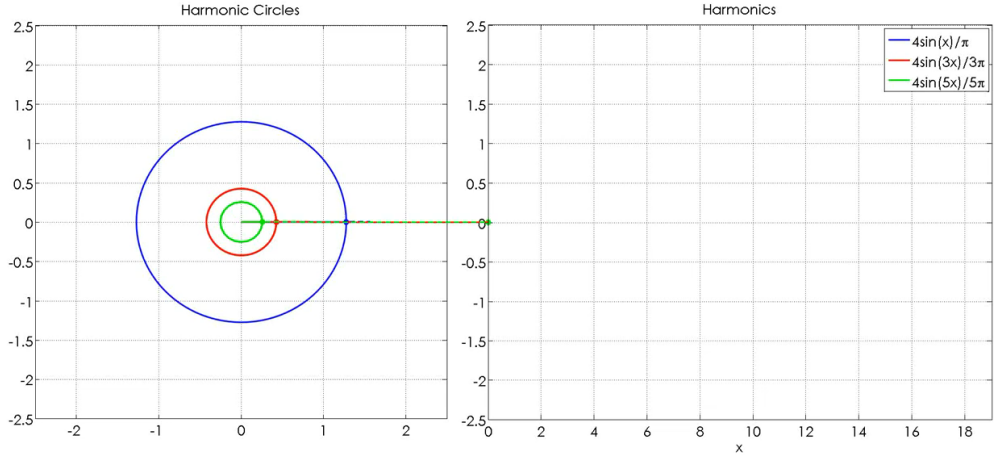
- Filtrace ve frekvenční oblasti
  - Převod obrazu do jiného prostoru, následná filtrace v tomto prostoru a převod zpět
  - Obraz je nejprve dekomponován na složky a filtrace se provádí nad těmito složkami
  - Existuje více způsobů, jak obraz (dvourozměrnou funkci) rozložit na složky, např.
    - na jednotlivé barevné kanály,
    - pomocí Taylorova rozvoje,
    - pomocí Ptolemaiových kružnic,
    - na harmonické složky pomocí Fourierovy transformace apod.

Ptolemaiovy kružnice

https://www.youtube.com/watch?v=8Q0NxFt-s7Y







https://www.youtube.com/watch?v=LznjC4Lo7lE

Fourierova transformace

- Fourierova analýza slouží ke zjištění, z jakých harmonických funkcí se daná funkce skládá.
- Harmonická funkce: $y(x) = A.\sin(\omega x + \varphi)$, kde
  - A je **amplituda**
  - ω je kruhová **frekvence**, $\omega = \frac{2\pi}{T}$, T je perioda
  - φ je **fázový posun** (fáze)
- Výsledkem Fourierovy analýzy je seznam (tabulka) amplitud a fázových posunů pro jednotlivé frekvence.
- Grafu závislosti amplitudy na frekvenci se říká **amplitudové spektrum**.
- Grafu závislosti fázového posunu na frekvenci se říká **fázové spektrum**.

Fourierova transformace

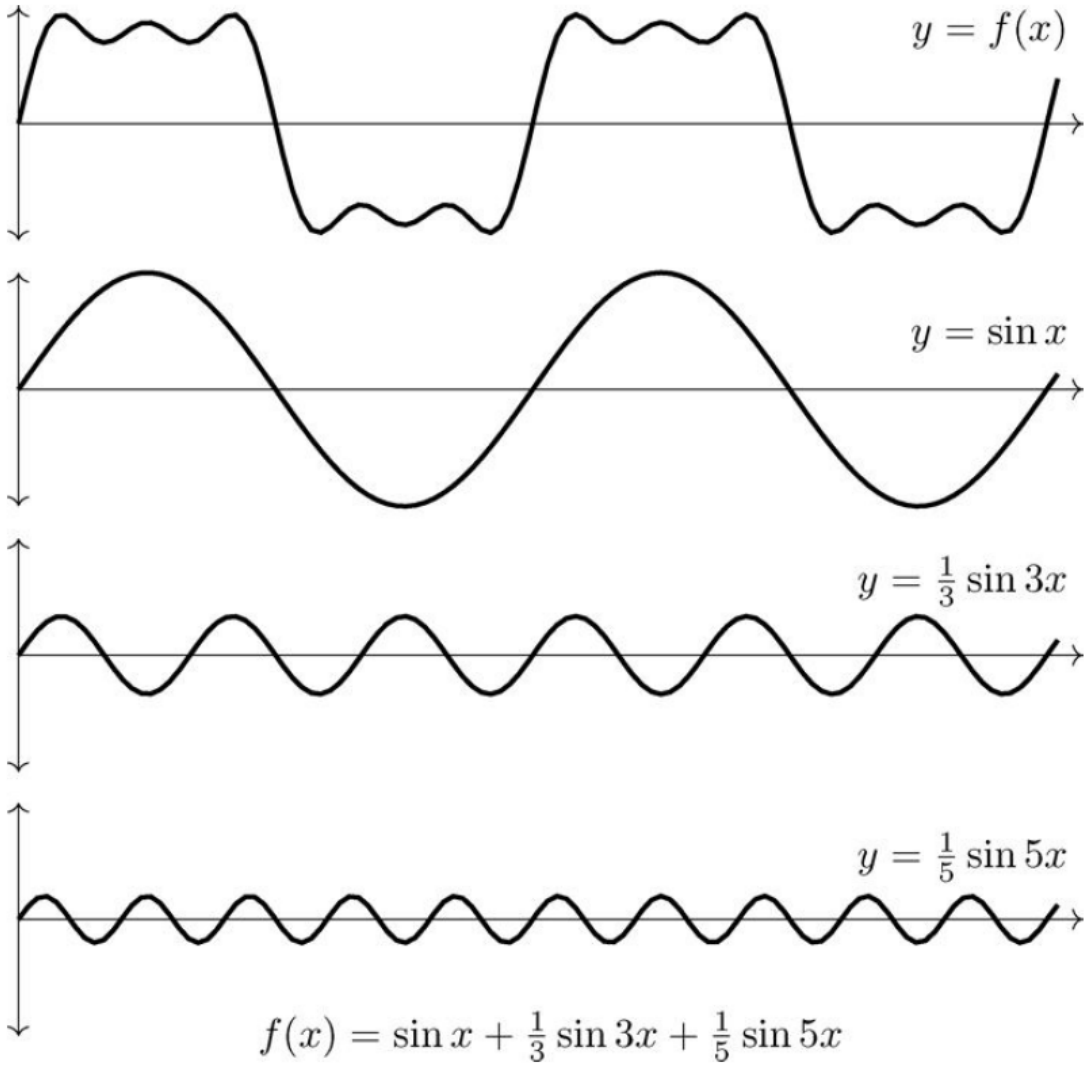
- Fourierova **řada** se používá pro vyjádření **periodického** signálu.
- Fourierova **transformace** (Fourierův integrál) se používá pro vyjádření **neperiodického** signálu.

- Dále se rozlišuje, zda pracujeme se **spojitou** nebo **diskrétní funkcí** (posloupností čísel).

Fourierova transformace

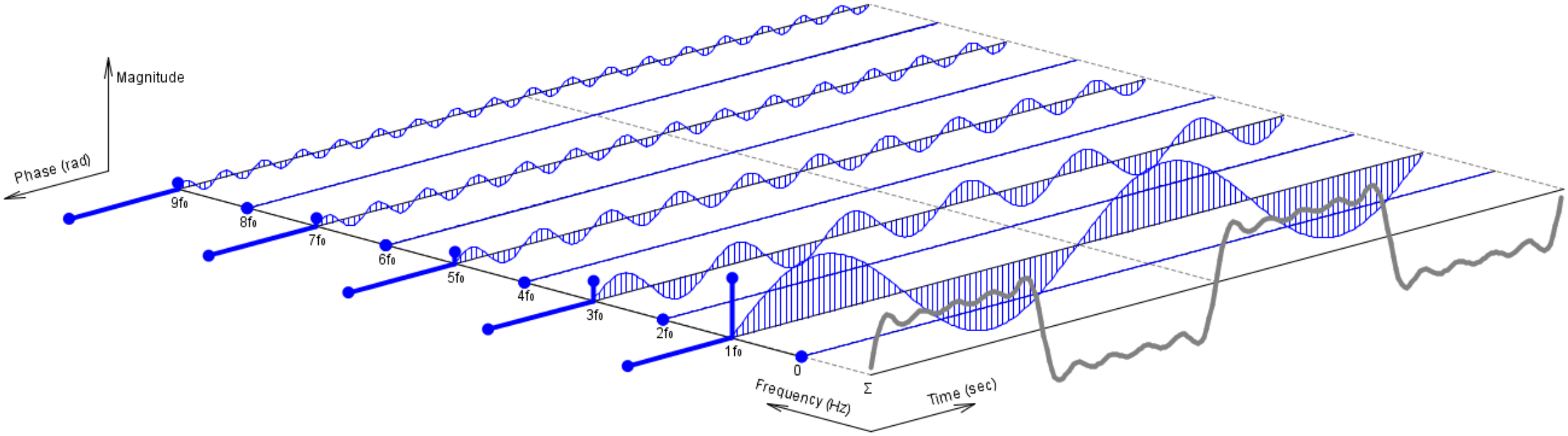
Příklad Fourierovy řady periodického signálu

$$f(x) = \sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x + \frac{1}{9}\sin 9x + \cdots$$



Fourierova transformace

- Rozklad 1D funkce na jednotlivé harmonické složky



http://www.tomasboril.cz/fourierseries3d/cz/

Fourierova transformace

- Rozvoj **Fourierovy řady** (**periodické funkce**) může zapsat i v komplexní podobě

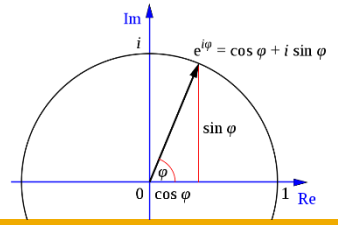
$$f(x) = \sum_{n=-\infty}^{\infty} c_n \exp\left(\frac{in\pi x}{T}\right) dx \quad \text{kde} \quad c_n = \frac{1}{2T}\int_{-T}^{T} f(x)\exp\left(\frac{-in\pi x}{T}\right) dx$$

Ukázka odvození komplexního tvaru jednoduché harmonické funkce

$$f(x) = \cos x = \cos \omega x\Big|_{\omega=1} = \underbrace{\frac{1}{2}\cos\omega x + i\frac{1}{2}\sin\omega x}_{\frac{1}{2}e^{i\omega x}} + \underbrace{\frac{1}{2}\cos\omega x - i\frac{1}{2}\sin\omega x}_{\frac{1}{2}e^{-i\omega x}} = \sum_{n\in\{1,-1\}} \frac{1}{2} e^{in\omega x} = \sum_{n=-\infty}^{\infty} c_n \cdot e^{in\omega x}, \text{ kde}$$

$$c_n = \begin{cases} 1 & \text{pro } n=1 \\ -1 & \text{pro } n=-1 \\ 0 & \text{jinde} \end{cases}$$

Eulerův vzorec: $e^{i\varphi} = \cos\varphi + i\sin\varphi$



$$\sin x = \cos\left(x - \frac{\pi}{2}\right)$$
$$\cos x = \sin\left(x + \frac{\pi}{2}\right)$$
$$\omega = \frac{2\pi}{T}$$

Fourierova transformace

- **Fourierova transformace** pro **neperiodickou funkci** lze také vyjádřit v **komplexní podobě**:

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega,$$

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{i\omega x} dx.$$

- Využití komplexních čísel je výhodné z důvodu kompaktnějšího zápisu a snazších matematických operací.

Fourierova transformace

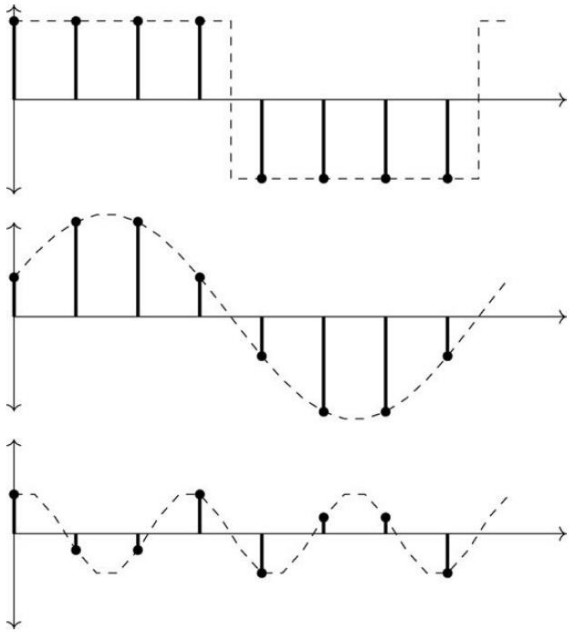
- Protože nás zajímá využití FT při zpracování obrazu, který je reprezentován maticí a je tedy diskrétní, omezíme se na **diskrétní Fourierovu transformaci** (**DFT**)
- Pro diskrétní **funkce jedné proměnné** (např. časové řady) je DFT dána těmito vztahy:

$$F_u = \frac{1}{N} \sum_{x=0}^{N-1} \exp\left[-2\pi i \frac{xu}{N}\right] f_x$$

DFT

$$x_u = \sum_{x=0}^{N-1} \exp\left[2\pi i \frac{xu}{N}\right] F_u$$

inverzní DFT (IDFT)



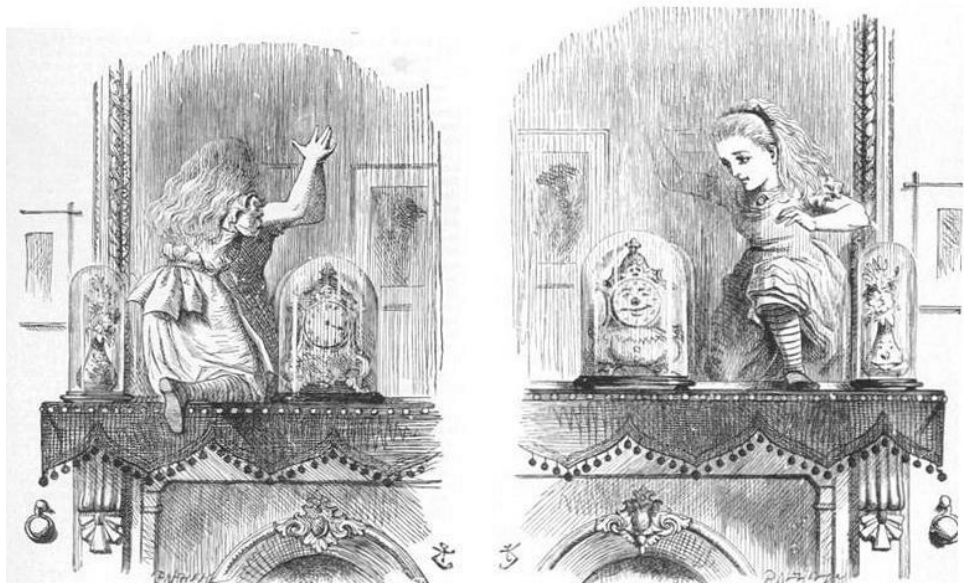
Fourierova transformace

- FT 2D obrázku (spektrum obrázku)

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-2\pi i \left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$

FT

FT^{-1}

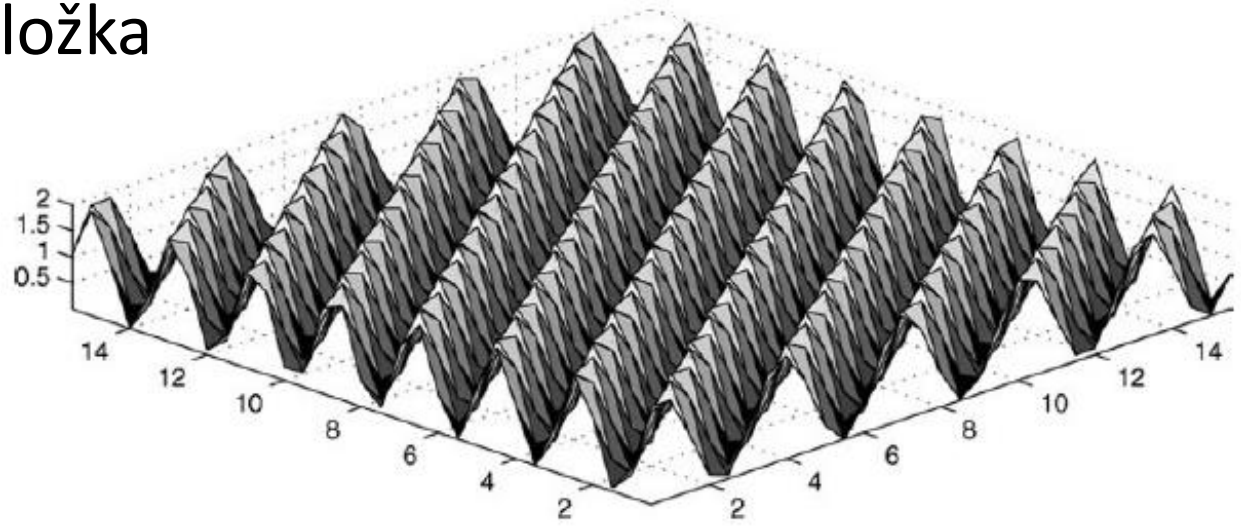


- Zpětná FT 2D obrázku (rekonstrukce obrázku ze spektra)

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$

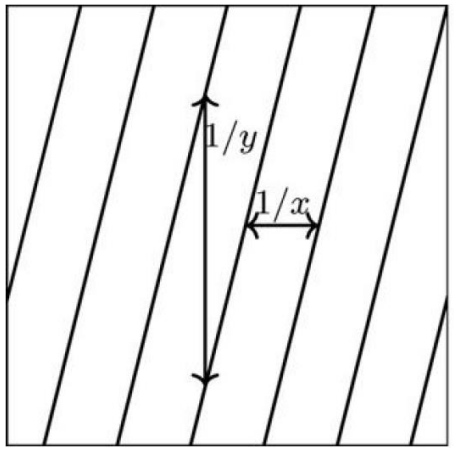
Fourierova transformace

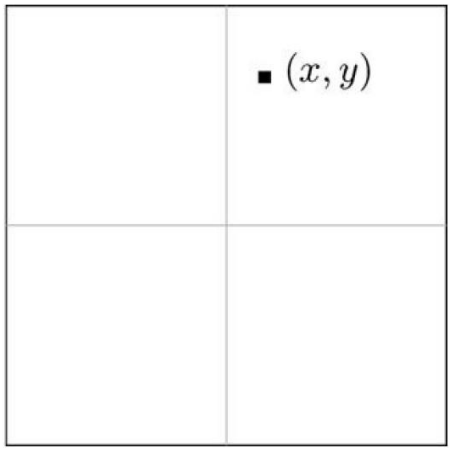
- 2D harmonická složka



$$z = a\sin(bx + cy)$$

- Pozice ve spektru





Fourierova transformace

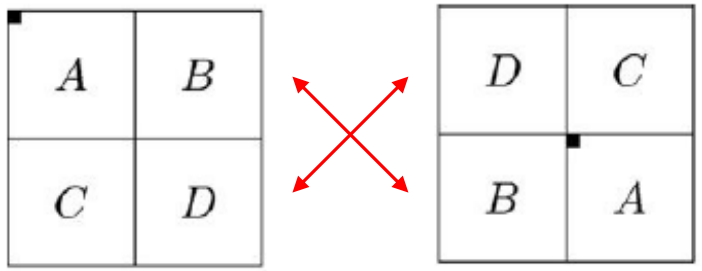
- Vlastnosti Fourierovy transformace
  - Linearita: $F(f+g) = F(f) + F(g)$ - výhodné pro odstranění známého šumu
  - Konvoluce: $F(M*S) = F(M)\cdot F(S)$ - výhodné pro filtraci
  - Posun: Spektrum se posune do středu, pokud pronásobíme obraz maticí $\{(-1)^{x+y}\}$

- Vizualizace spektra
  - Snížení vlivu stejnosměrné složky, např. $\log(1+|F(u,v)|)$

- FFT (Fast Fourier Transform)
  - Efektivní implementace DFT (2^{2n} -> $n2^n$ operací násobení)
  - Postavena na rekurzivním dělení dat -> výhodné pro matice o rozměrech 2^k

Fourierova transformace

- Posun spektra
  - Po FT je standardně stejnosměrná složka v levém horním rohu matice spektra
  - Okolí stejnosměrné složky odpovídá složkám s nižší frekvencí, která se zvyšuje jak se od levého horního roku vzdalujeme
  - Pro snazší interpretaci je výhodné provést posun spektra tak, aby stejnosměrná složka byla uprostřed a okolní hodnoty FT odpovídající složkám o nižších frekvencích. Větší vzdálenosti od středu odpovídají složkám o vyšších frekvencích.
  - Posun spektra využívá skutečnosti, že pokud obrázek pronásobíme stejně velkou maticí ve tvaru šachovnice s hodnotami +1 a -1, dojde k žádanému posunu.



Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 1: Mějme obrázek o rozměru 8x8 pixelů, který obsahuje samé jedničky (jednolitá plocha)

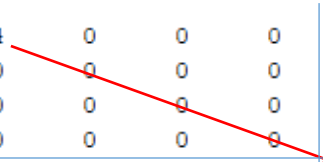
```
a1 =

     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     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1
```

obrázek jako matice

```
ans =

    64     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```



spektrum obrázku (neposunuté)

- Pokud dosadíme do vztahu pro F(u,v) za u=0 a v=0, vypadne exponenciální člen (e^0=1) a zůstane součet prvků matice, tj. hodnota 64.
- Pokud dosadíme do vztahu pro F(u,v) např. za u=0 a v=1, výpočtem zjistíme, že se vyruší a bude platit, že F(0,1)=1.
- F(u,v)=0 dostaneme pro všechny hodnoty u a v vyjma prvního případu, kdy u=0 a v=0.

Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 1: Mějme obrázek o rozměru 8x8 pixelů, který obsahuje samé jedničky (jednolitá plocha)

```
a1 =

     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     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1
```

obrázek jako matice

```
ans =

     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0    64     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

spektrum obrázku (posunuté)

- Pokud dosadíme do vztahu pro F(u,v) za u=0 a v=0, vypadne exponenciální člen (e^0=1) a zůstane součet prvků matice, tj. hodnota 64.
- Pokud dosadíme do vztahu pro F(u,v) např. za u=0 a v=1, výpočtem zjistíme, že se vyruší a bude platit, že F(0,1)=1.
- F(u,v)=0 dostaneme pro všechny hodnoty u a v vyjma prvního případu, kdy u=0 a v=0.

Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 2: Mějme obrázek o rozměru 8x8 pixelů, který obsahuje následující hodnoty:

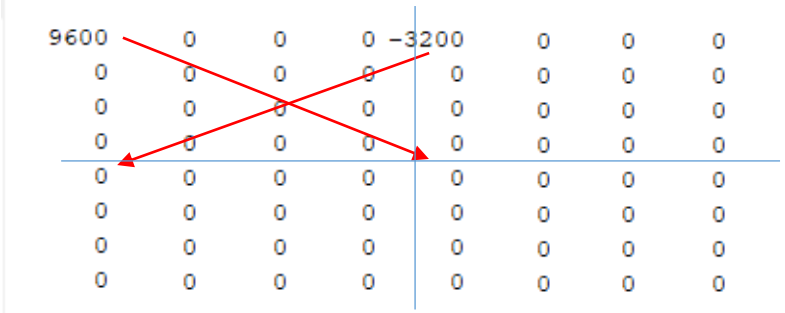
```
a2 =

   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
```

obrázek jako matice

```
af2 =

  9600     0     0     0 -3200     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```



spektrum obrázku (neposunuté)

- Pokud dosadíme do vztahu pro F(u,v) za u=0 a v=0, vypadne exponenciální člen (e^0=1) a zůstane dvojnásobný součet prvků matice, tj. hodnota 9600.
- Pokud dosadíme do vztahu pro F(u,v) např. za u=4 a v=0, výpočtem zjistíme, že F(0,1)=-3200.
- Pro ostatní hodnoty u a v dostaneme F(u,v)=0.
- Matice F(u,v) obsahuje v tomto případě reálná čísla. To však nemusí být vždy pravda. V tom případě je F(u,v) matice komplexních čísel a vyjadřujeme ji dvěma maticemi: maticí amplitud a maticí fází.

Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 2: Mějme obrázek o rozměru 8x8 pixelů, který obsahuje následující hodnoty:

```
a2 =

   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
   100   200   100   200   100   200   100   200
```

obrázek jako matice

```
af2 =

     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
 -3200     0     0     0  9600     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

spektrum obrázku (posunuté)

- Pokud dosadíme do vztahu pro F(u,v) za u=0 a v=0, vypadne exponenciální člen (e^0=1) a zůstane dvojnásobný součet prvků matice, tj. hodnota 9600.
- Pokud dosadíme do vztahu pro F(u,v) např. za u=4 a v=0, výpočtem zjistíme, že F(0,1)=-3200.
- Pro ostatní hodnoty u a v dostaneme F(u,v)=0.
- Matice F(u,v) obsahuje v tomto případě reálná čísla. To však nemusí být vždy pravda. V tom případě je F(u,v) matice komplexních čísel a vyjadřujeme ji dvěma maticemi: maticí amplitud a maticí fází.

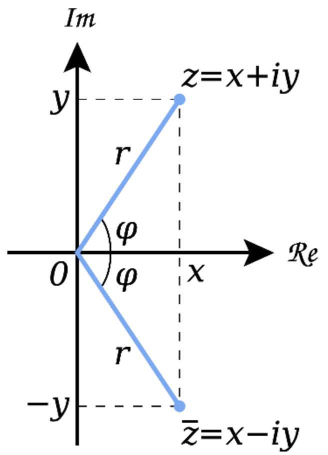
Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 3: Mějme obrázek jednotkového skoku, tj. matici:

a3 =

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

obrázek jako matice



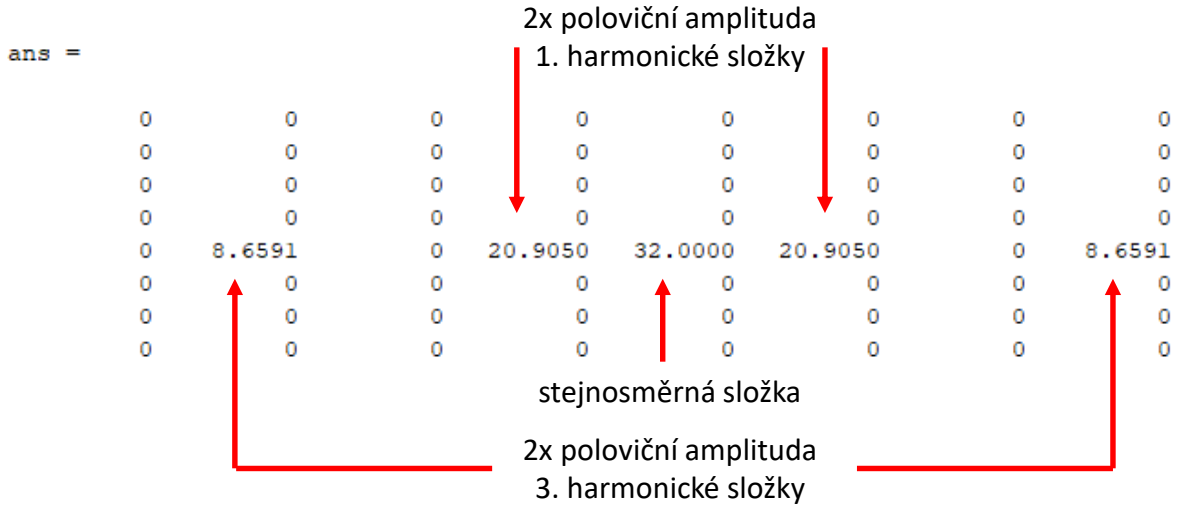
af3 =

spektrum obrázku (tentokrát už obsahuje komplexní čísla a je posunuté)

0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	-8.0000 - 3.3137i	0.0000 + 0.0000i	-8.0000 -19.3137i	32.0000 + 0.0000i	-8.0000 +19.3137i	0.0000 + 0.0000i	-8.0000 + 3.3137i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i

Fourierova transformace

- Jednoduché příklady DFT
  - Příklad 3: FT obrázku s posunem stejnosměrné složky na střed (shift):



amplitudové spektrum (posunuté)

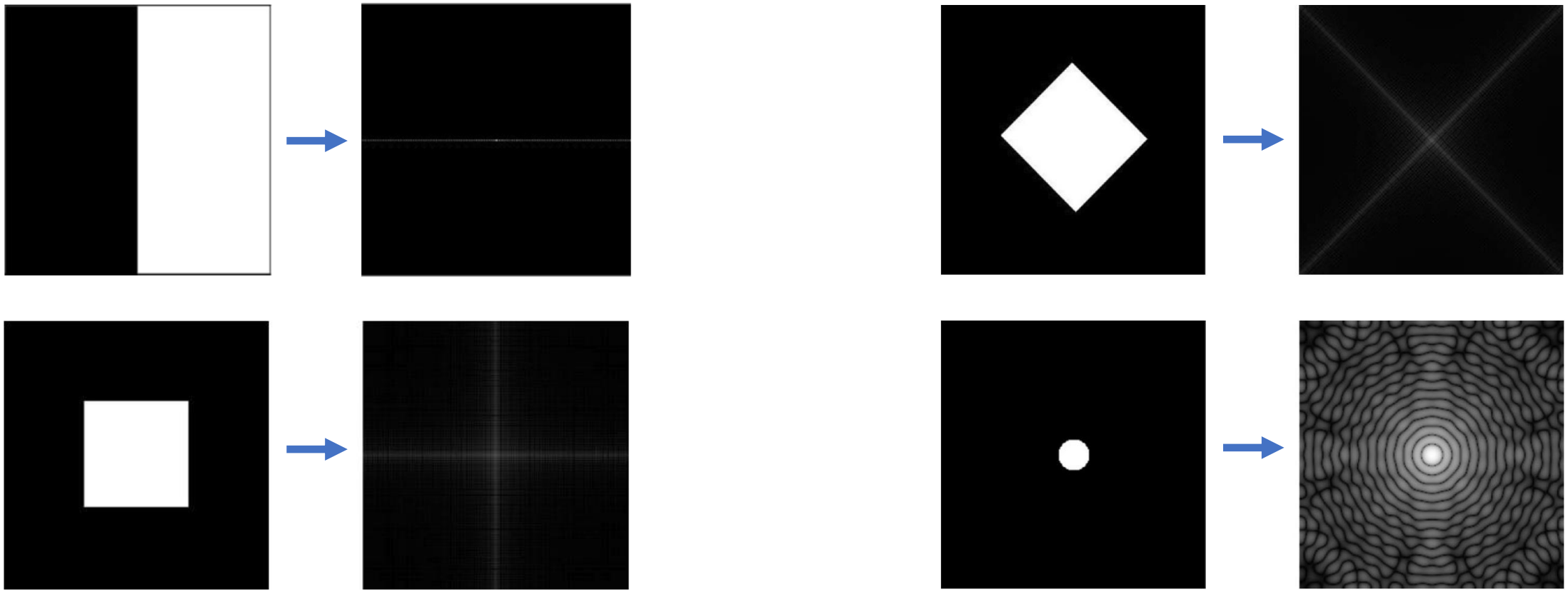
```
ans =

         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0 -157.5000         0 -112.5000         0  112.5000         0  157.5000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
```

fázové spektrum (ve stupních, posunuté)

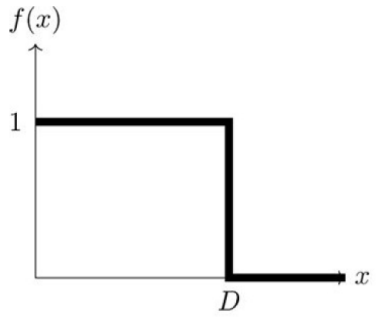
Fourierova transformace

- Jednoduché příklady DFT
  - Příklady umělých obrázků (všechna spektra jsou již po posunu)

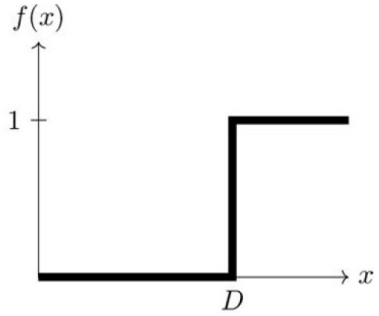


Filtrace ve frekvenční oblasti

- Ideální filtr
  - Low-pass filter (dolnofrekvenční propusť)

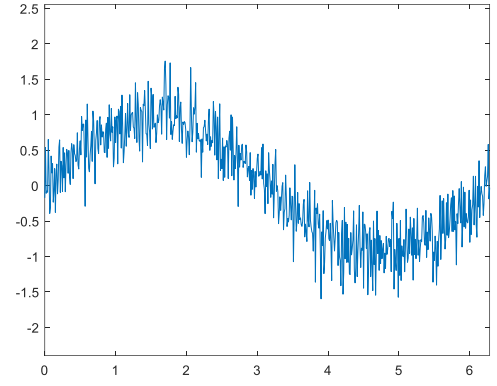


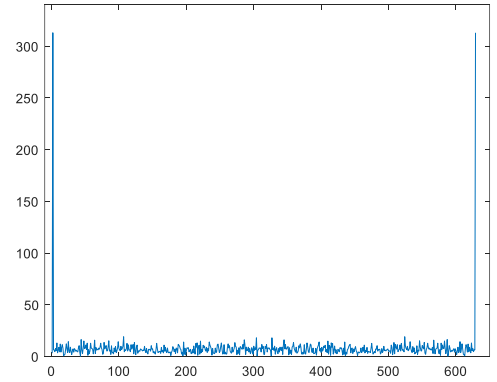
  - High-pass filter (hornofrekvenční propusť)

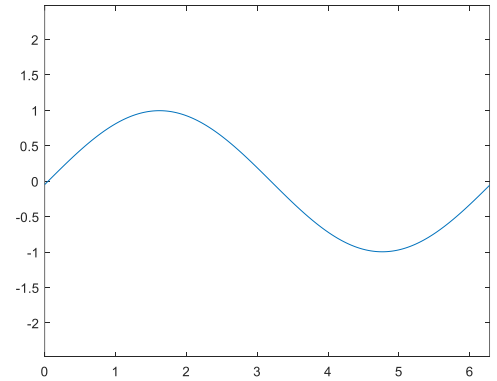


stejnosměrná složka

3.5610	0	0
315.3958	1	315.3958
4.6472	0	0
5.6448	0	0
4.6713	0	0
5.8070	0	0
9.1820	0	0
9.6404	0	0
12.6469	0	0
5.3635	0	0
.	.	.
. x	. =	.
.	.	.
4.8689	0	0
7.6546	0	0
5.3635	0	0
12.6469	0	0
9.6404	0	0
9.1820	0	0
5.8070	0	0
4.6713	0	0
5.6448	0	0
4.6472	0	0
315.3958	1	315.3958

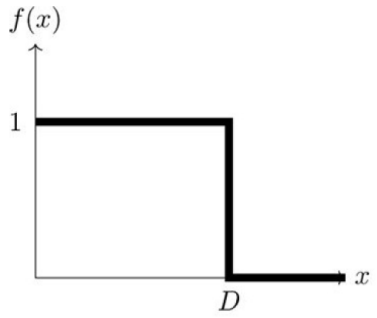






Filtrace ve frekvenční oblasti

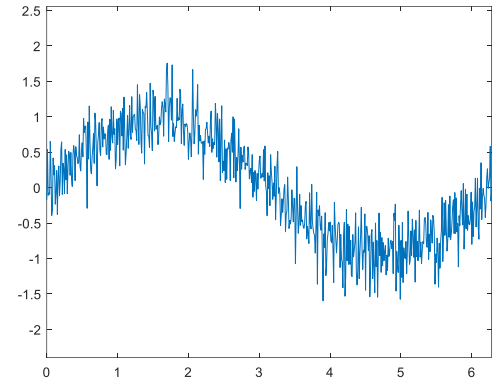
- Ideální filtr
  - Low-pass filter (dolnofrekvenční propusť)

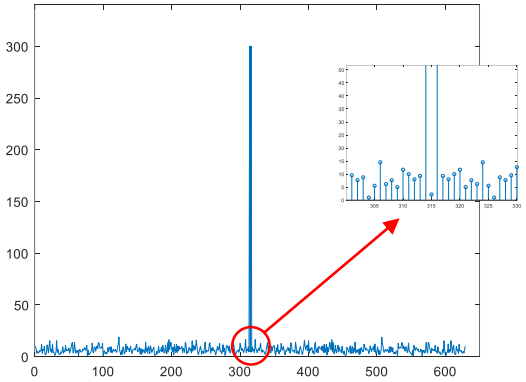


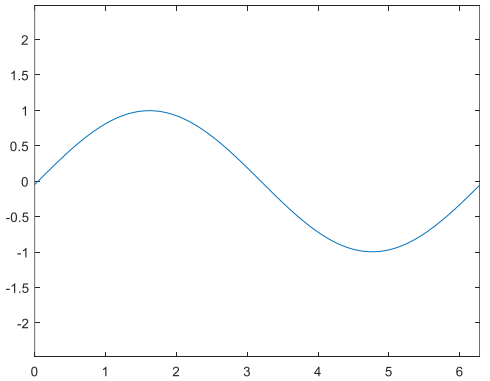
stejnosměrná složka

.	.	.	
.	.	.	
.	.	.	
4.8689	0	0	
7.6546	0	0	
5.3635	0	0	
12.6469	0	0	
9.6404	0	0	
9.1820	0	0	
5.8070	0	0	
4.6713	0	0	
5.6448	0	0	
4.6472	0	0	
315.3958	1	315.3958	
3.5610	0	0	
315.3958	1	315.3958	
4.6472	0	0	
5.6448	0	0	
4.6713	0	0	
5.8070	0	0	
9.1820	0	0	
9.6404	0	0	
12.6469	0	0	
5.3635	0	0	
.	.	.	
.	.	.	
.	.	.	

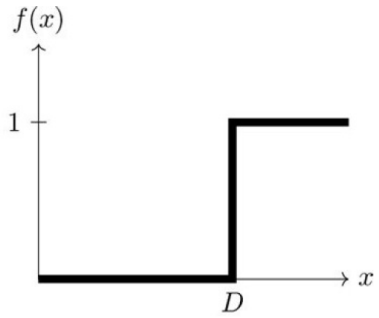
x =







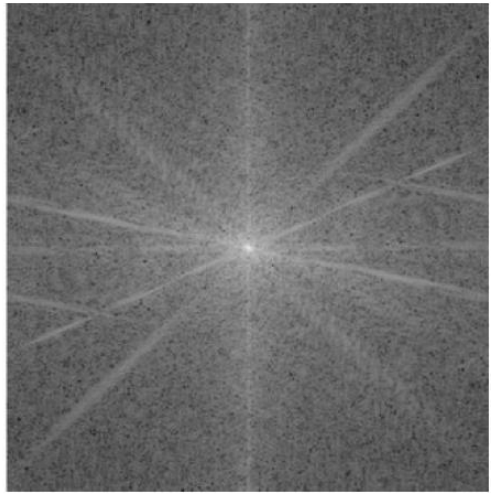
  - High-pass filter (hornofrekvenční propusť)



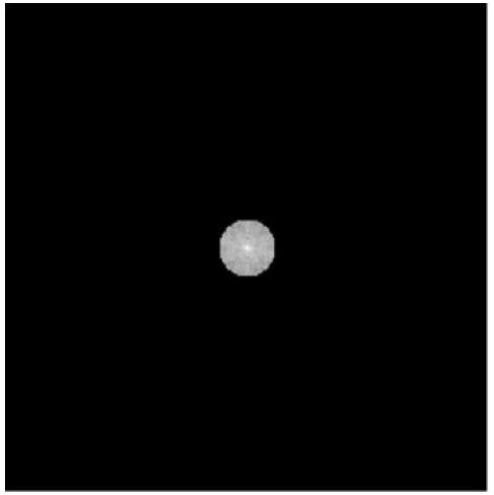
Filtrace ve frekvenční oblasti

- Ideální filtr – Low-pass filter (dolnofrekvenční propusť)

(amplitudové) spektrum původního obrázku



dolnofrekvenční ideální propusť, $D = 15$



zpětná transformace z vyfiltrovaného spektra

Filtrace ve frekvenční oblasti

- Ideální filtr – High-pass filter (hornofrekvenční propusť)

Filtrace ve frekvenční oblasti

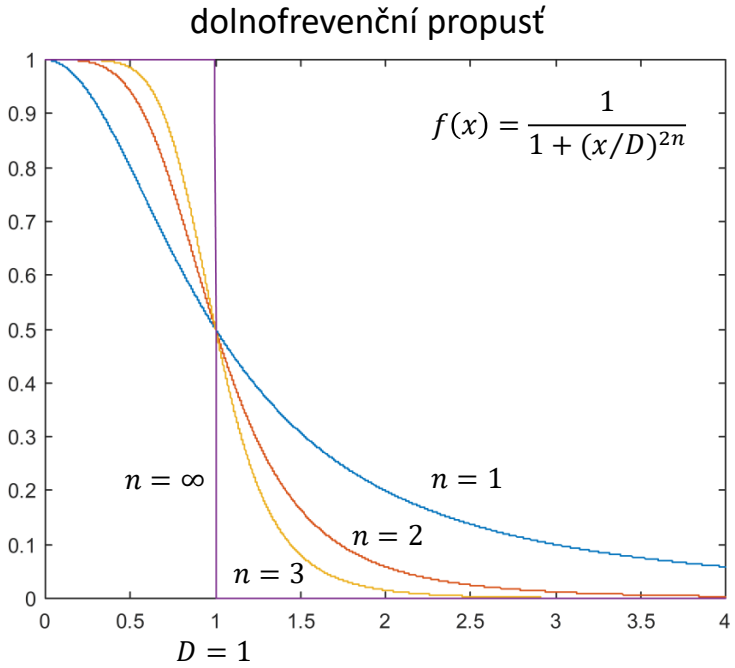
- Butterworthův filtr
  - Ideální filtr – snadná SW implementace, artefakty zvlnění obrazu
  - BF – postupný přechod -> snížení zvlnění obrazu

dolnofrekvenční propusť

$$f(x) = \frac{1}{1 + (x/D)^{2n}} \qquad f(x,y) = \frac{1}{1 + \left(\frac{x^2 + y^2}{D^2}\right)}$$

hornofrekvenční propusť

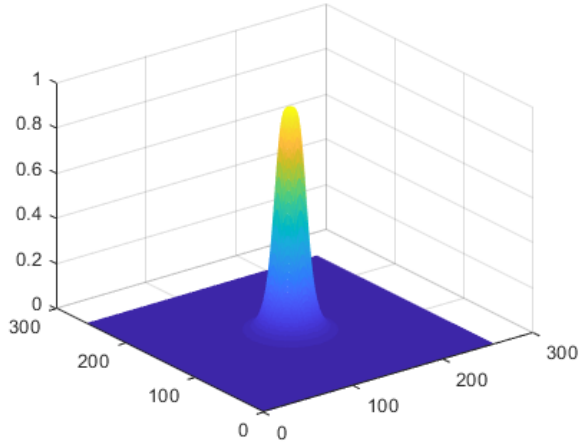
$$f(x) = \frac{1}{1 + (D/x)^{2n}} \qquad f(x,y) = \frac{1}{1 + \left(\frac{D^2}{x^2 + y^2}\right)}$$



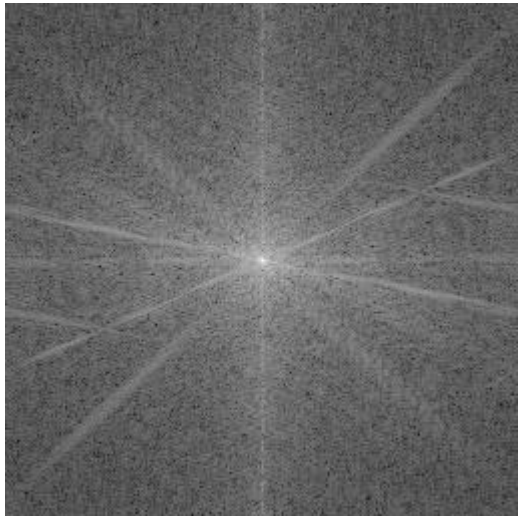
n je řád filtru, *D* je místo přechodu přes 0,5 (odpovídá hraniční frekvenci)

Filtrace ve frekvenční oblasti

- Butterworthův filtr

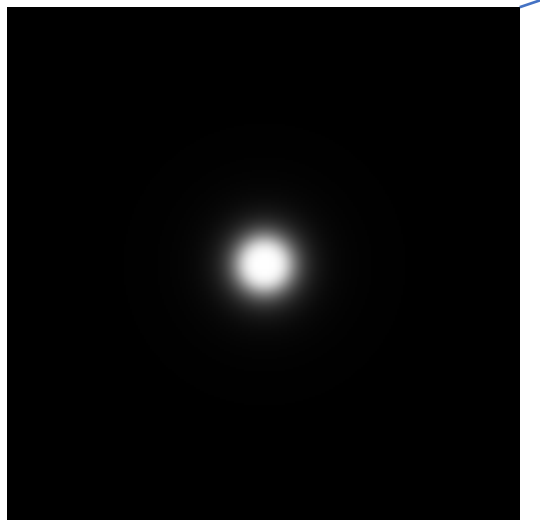


(amplitudové) spektrum původního obrázku

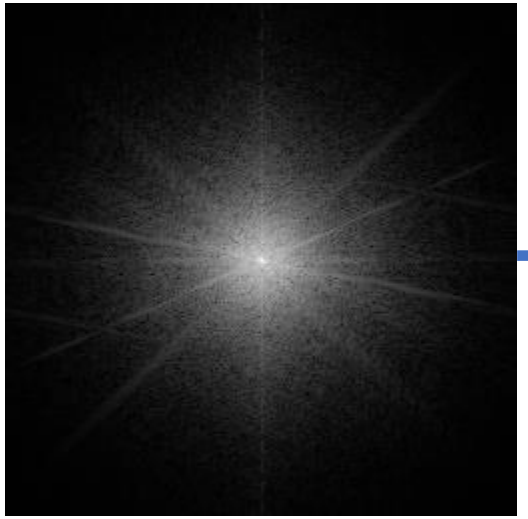


.

dolnofrekvenční propusť
$D = 15$, $n = 2$



=

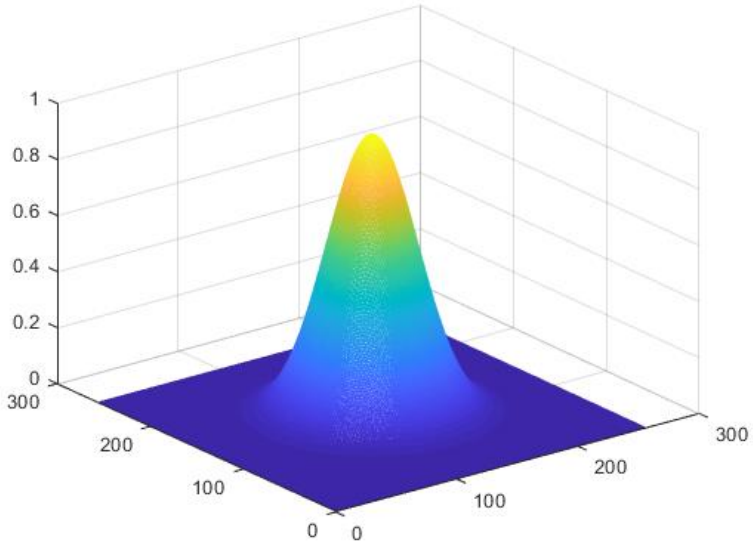


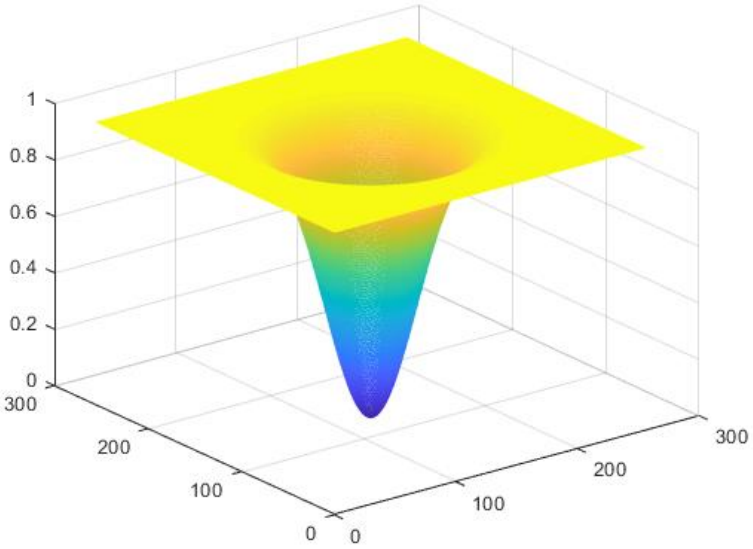
Filtrace ve frekvenční oblasti

- Gaussův filtr
  - „Hladký“ filtr
  - Tvar Gaussovy funkce (pro 2D tvar „klobouku“)

$$f(x,y) = A\exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right)$$

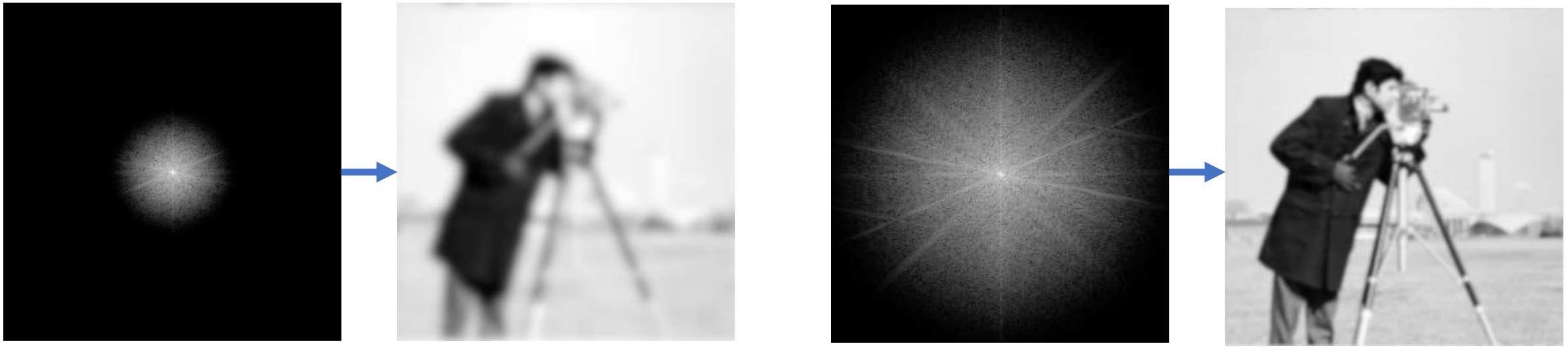
  - FT obrázku po prvcích pronásobíme hodnotami filtru a na výsledek aplikujeme zpětnou FT





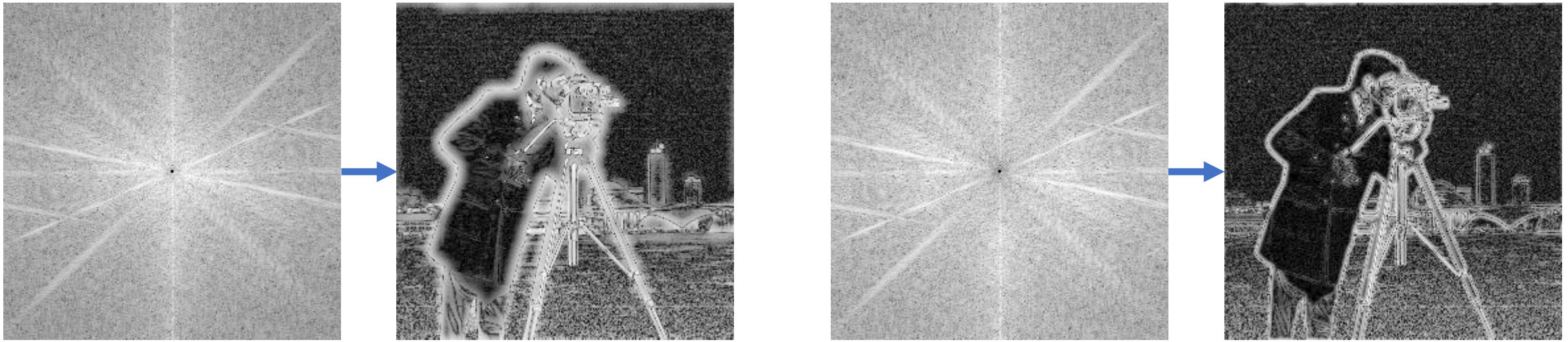
Filtrace ve frekvenční oblasti

- Gaussův filtr – low-pass filtr



Filtrace ve frekvenční oblasti

- Gaussův filtr – high-pass filtr



Literatura

- McAndrew A., Computational Introduction to Digital Image Processing, CRC Press, 2. vydání, 2016
- Sundararajan D., Digital Image Processing: A Signal Processing and Algorithmic Approach, Springer, 2017
- Birchfield S., Image Processing and Analysis, Cengage Learning, 2016
- Acharya T., Ray A. K., Image Processing: Principles and Applications, Wiley, 2005
- Burger W., Burge M. J., Principles of Digital Image Processing: Fundamental Techniques, Springer-Verlag, 2009
- https://docs.opencv.org/master/de/dbc/tutorial_py_fourier_transform.html
- https://docs.opencv.org/master/d8/d01/tutorial_discrete_fourier_transform.html