ROZ1 - Cv. 2 - Fourierova transformace

ÚTIA - ZOI

Co to je FT?

Co to je FT?

- Transformace signálu z časové (resp. obrazové) reprezentace $f(t)$ do frekvenční reprezentace $F(\psi)$ a zpět.
- Díky ní můžeme signál analyzovat ve frekvenční oblasti
- Zápis FT v 1D:

Co to je FT?

- Transformace signálu z časové (resp. obrazové) reprezentace $f(t)$ do frekvenční reprezentace $F(\psi)$ a zpět.
- Díky ní můžeme signál analyzovat ve frekvenční oblasti
- Zápis FT v 1D:
- $F(\xi) = \int_{-\infty}^{\infty} f(t)\epsilon^{-2\pi i\xi t}\,\mathrm{d}t \quad \overset{\mathrm{FT}}{\Longleftrightarrow} \quad f(t) = \int_{-\infty}^{\infty} F(\xi)\epsilon^{2\pi i\xi t}\,\mathrm{d}\xi$
- Zápis DFT v 1D:

Co to je FT?

- Transformace signálu z časové (resp. obrazové) reprezentace $f(t)$ do frekvenční reprezentace $F(\psi)$ a zpět.
- Díky ní můžeme signál analyzovat ve frekvenční oblasti
- Zápis FT v 1D:
- $F(\xi) = \int_{-\infty}^{\infty} f(t)\epsilon^{-2\pi i \xi t}\,\mathrm{d}t \quad \overset{\mathrm{FT}}{\Longleftrightarrow} \quad f(t) = \int_{-\infty}^{\infty} F(\xi)\epsilon^{2\pi i \xi t}\,\mathrm{d}\xi$
- Zápis DFT v 1D:
- $F(k) = \sum_{n=0}^{N-1} f(n)\epsilon^{\frac{-2\pi i k n}{N}} \quad \overset{\mathrm{DFT}}{\Longleftrightarrow}$

  $f(n) = \frac{1}{N}\sum_{k=0}^{N-1} F(k)\epsilon^{\frac{2\pi i k n}{N}}$

K čemu je FT dobrá při digitálním zpracování obrazu?

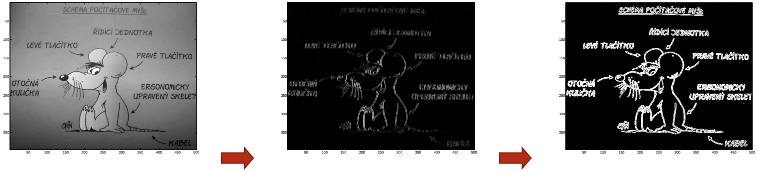
K čemu je FT dobrá při digitálním zpracování obrazu?

- základní matematický nástroj
- odstranění šumu
- detekce hran
- segmentace
- rekonstrukce
- komprese obrazu
- detekce objektů
- atd.

Detekce hran

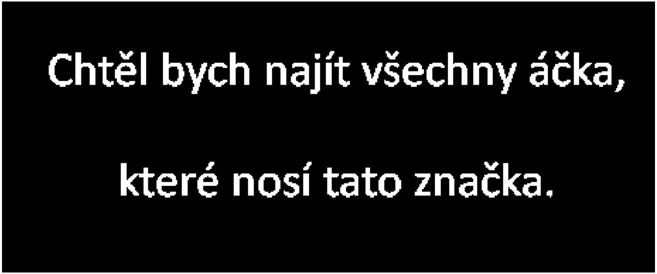
- Aplikace hranových detektorů ve frekvenční oblasti (filtr Prewittové)

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



Detekce objektů v obraze

- Jde o aplikaci konvolučního teorému: $(f * g)(t) = F(\psi)G(\psi)$

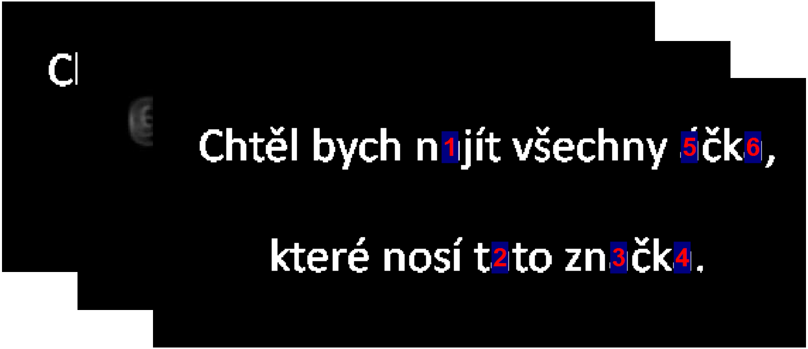


Detekce objektů v obraze

- Jde o aplikaci konvolučního teorému: $(f * g)(t) = F(\psi)G(\psi)$

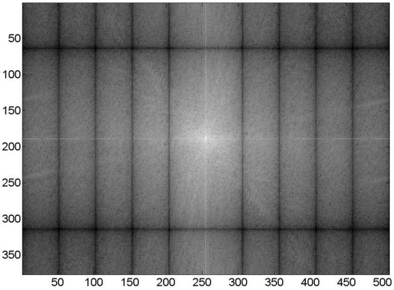
Detekce objektů v obraze

- Jde o aplikaci konvolučního teorému: $(f * g)(t) = F(\psi)G(\psi)$



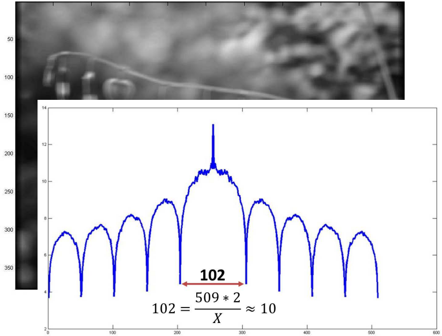
Doostření scény

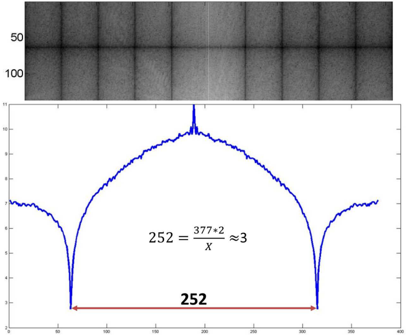
- I zde jde o aplikaci konvolučního teorému:



Doostření scény

- I zde jde o aplikaci konvolučního teorému:





Doostření scény

- I zde jde o aplikaci konvolučního teorému:

Náročnost FT na výpočet

- Výpočetní náročnost klasické DFT je:

Náročnost FT na výpočet

- Výpočetní náročnost klasické DFT je:
- $\mathcal{O}(N^2)$
- V čem spočívá FFT?

Náročnost FT na výpočet

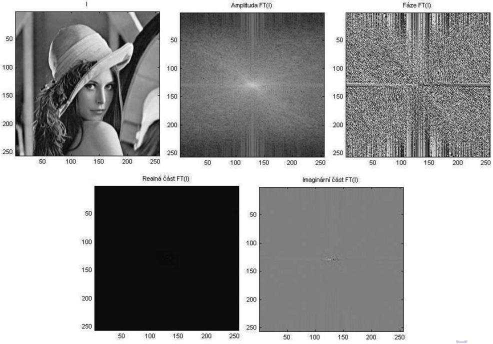
- Výpočetní náročnost klasické DFT je:
- $\mathcal{O}(N^2)$
- V čem spočívá FFT?
- (Danielson, Lanczos, 1942): DFT posloupnosti délky N lze vyjádřit jako součet dvou DFT posloupností délky $\frac{N}{2}$ - v první jsou liché a ve druhé sudé vzorky
- Výpočetní náročnost takové FFT je:

Náročnost FT na výpočet

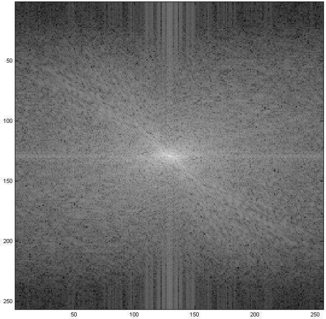
- Výpočetní náročnost klasické DFT je:
- $\mathcal{O}(N^2)$
- V čem spočívá FFT?
- (Danielson, Lanczos, 1942): DFT posloupnosti délky N lze vyjádřit jako součet dvou DFT posloupností délky $\frac{N}{2}$ - v první jsou liché a ve druhé sudé vzorky
- Výpočetní náročnost takové FFT je:
- $\mathcal{O}(N \log_2 N)$

Cvičení I.

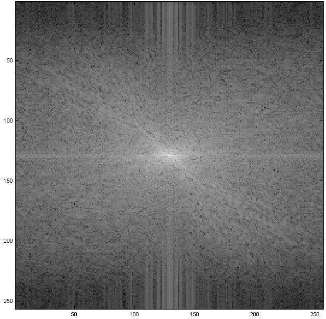
- Zobrazte amplitudu, fázi, reálnou i imaginární část - *nápověda:* `fft2()`, `ifft2()`, `fftshift()`, `abs()`, `angle()`, `real()`, `imag()`, `log()`, `exp()`



Řešení - Cvičení Ia. - amplituda

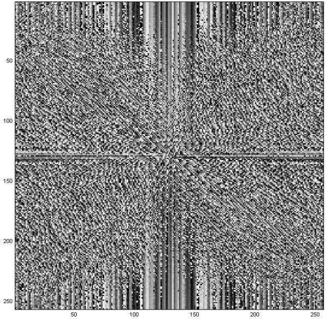


Řešení - Cvičení Ia. - amplituda

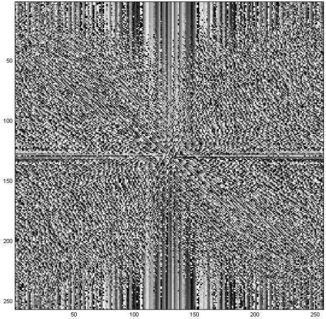


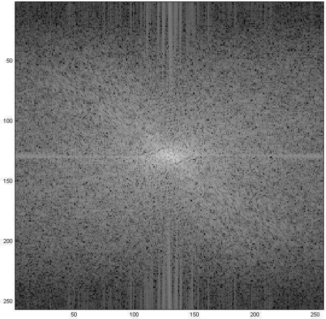
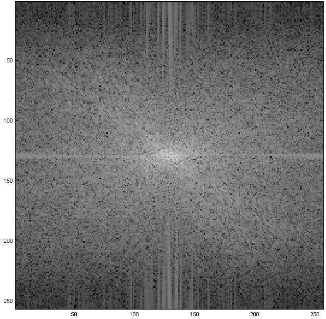
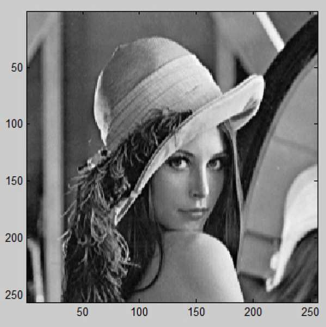
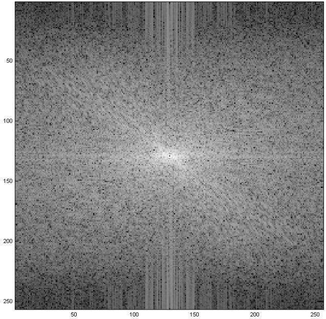
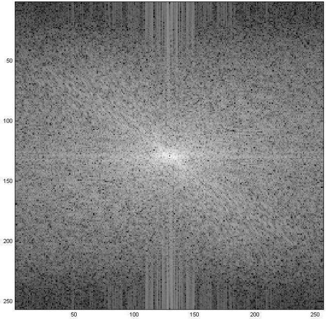
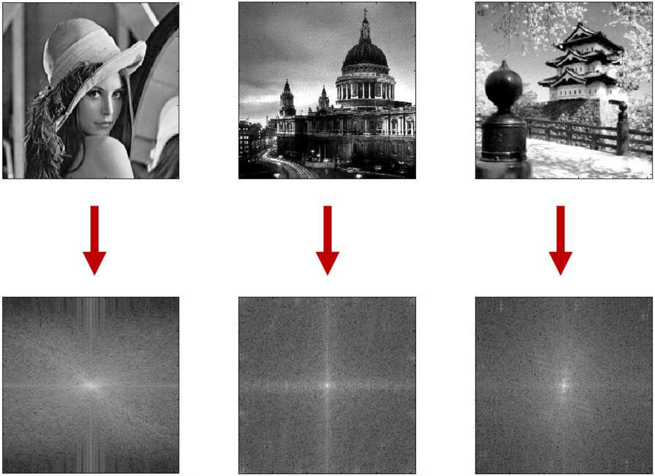
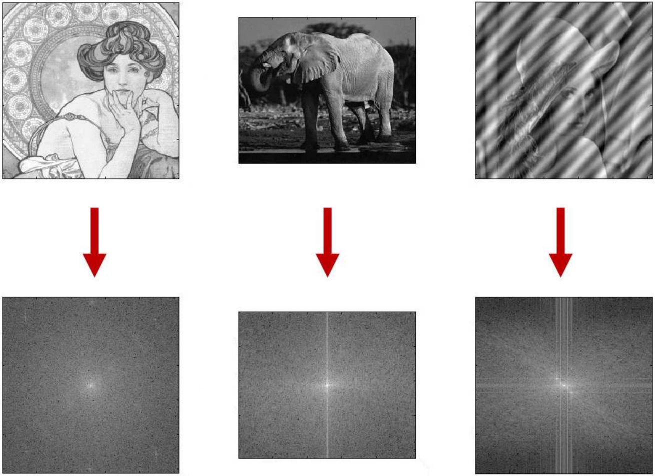
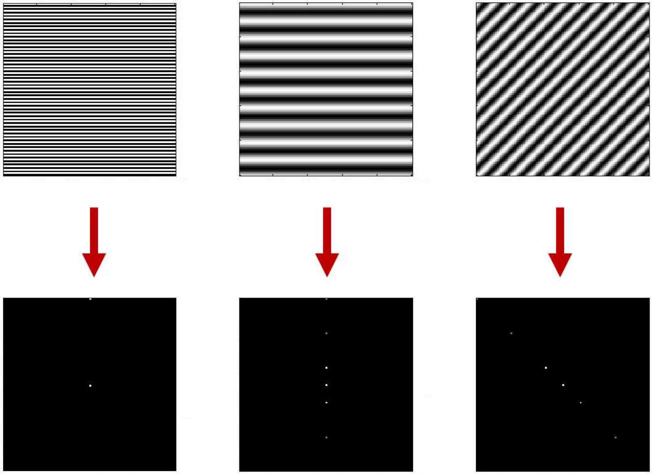
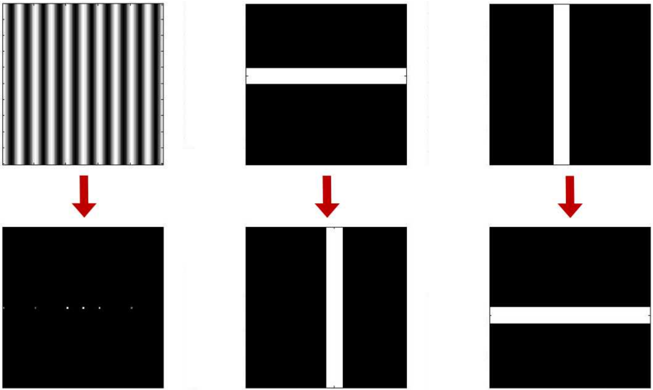
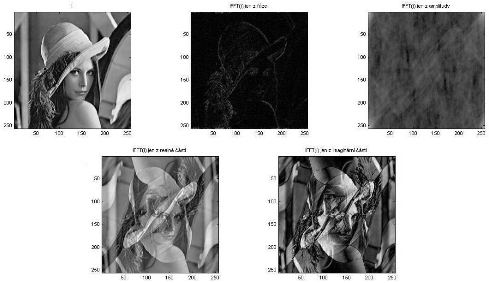
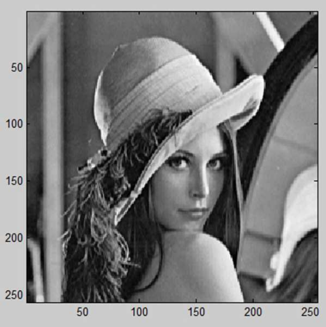
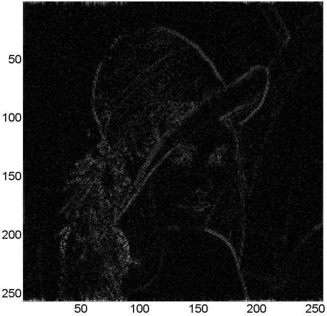
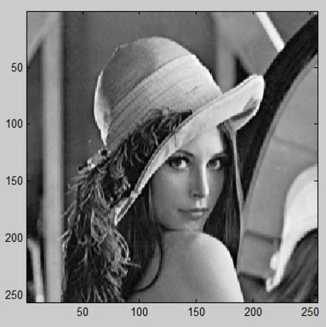
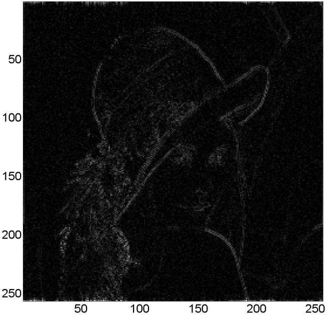
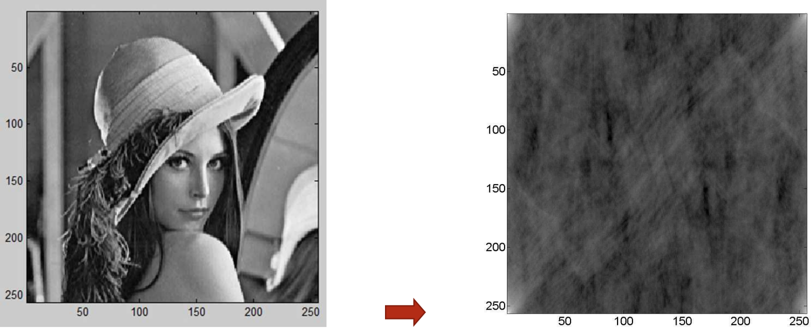
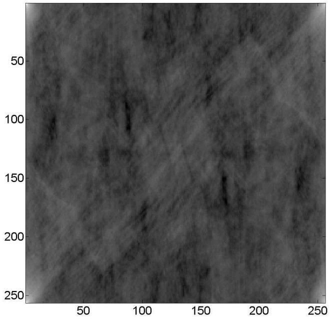
- `F=fft2(I);`
  `zobr(fftshift(log(abs(F)+1)));`

Řešení - Cvičení Ib. - fáze

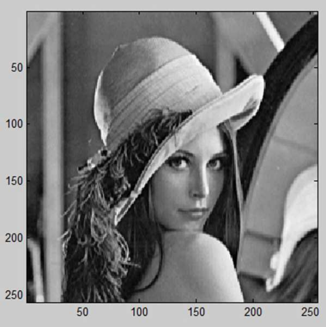


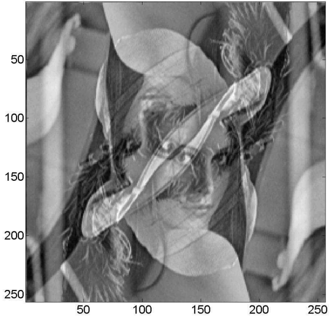
Řešení - Cvičení Ib. - fáze



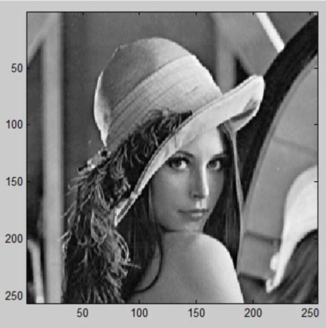

- ```
  F=fft2(I);
  zobr(fftshift(angle(F)));
  ```
  ```

Řešení - Cvičení Ic. - reálná část



Řešení - Cvičení Ic. - reálná část



- `F=fft2(I);`
  `zobr(fftshift(log(abs(real(F))+1)));`

Řešení - Cvičení Id. - imaginární část





Řešení - Cvičení Id. - imaginární část



- F=fft2(I);
  zobr(fftshift(log(abs(imag(F))+1)));

Cvičení II.

- Zobrazte amplitudu FT ostatních snímků
  - *co lze vizuálně vysledovat?*

Řešení - Cvičení II.a



Řešení - Cvičení II.b



Řešení - Cvičení II.c



Řešení - Cvičení II.d



Čemu se rovná `fft2(Img)` v bodě `(1,1)`?

Čemu se rovná `fft2(Img)` v bodě `(1,1)`?

- `fft2(Img) == sum(Img(:))`
- Proč?

Čemu se rovná `fft2(Img)` v bodě `(1,1)`?

- `fft2(Img) == sum(Img(:))`
- Proč?
- $F(0) = \sum_{n=0}^{N-1} f(n) \epsilon^{\frac{-2\pi i 0 n}{N}} = \sum_{n=0}^{N-1} f(n)$

Cvičení III.

- Zrekonstruujte snímek jen z jeho fáze, amplitudy, reálné a imaginární části



Řešení - Cvičení IIIa. - z fáze





Řešení - Cvičení IIIa. - z fáze





- F=fft2(I);
  zobr(abs(ifft2(exp(1i*angle(F)))));

Řešení - Cvičení IIIb. - z amplitudy



Řešení - Cvičení IIIb. - z amplitudy




- ```
  F=fft2(I);
  zobr(log(ifft2(abs(F))+1));
  ```
  ```

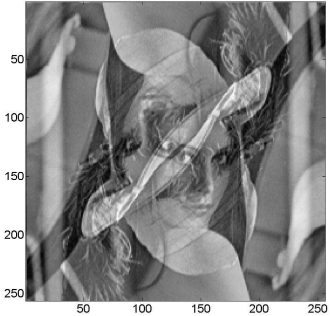
Řešení - Cvičení IIIc. - z reálné části

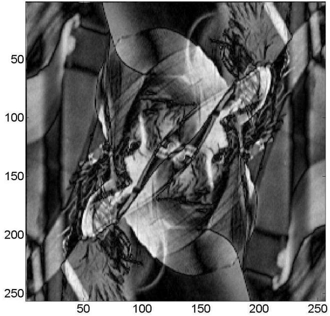
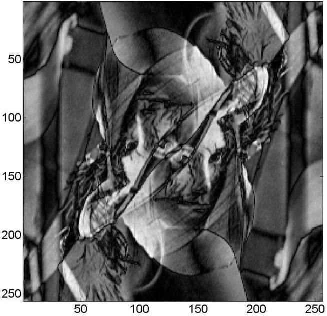




Řešení - Cvičení IIIc. - z reálné části





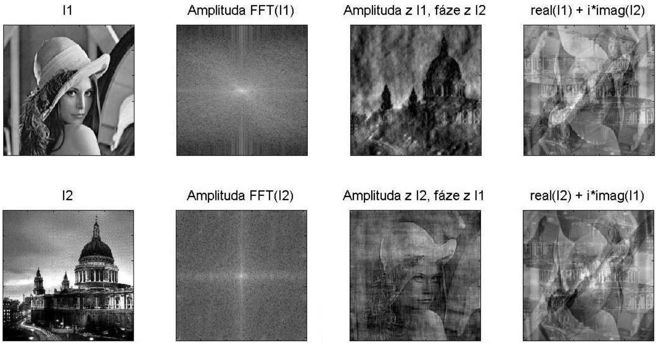

- ```
  F=fft2(I);
  zobr(ifft2(real(F)));
  ```
  ```

Řešení - Cvičení IIId. - z imaginární část



Řešení - Cvičení IIId. - z imaginární část




- ```
  F=fft2(I);
  zobr(abs(ifft2(imag(F))));
  ```
  ```

Cvičení IV.

- Zrekonstruujte snímek ze dvou původních snímků ($I1$, $I2$):
  1. nakombinujete amplitudu z $I1$ a fázi z $I2$
  2. nakombinujete reálnou část z $I1$ a imaginární část z $I2$

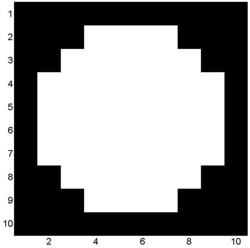
Řešení - Cvičení IV.

```
F1=fft2(I1);      F2=fft2(I2);
ad1.: zobr(abs(ifft2(abs(F1).*exp(1i*angle(F2)))));
ad2.: zobr(ifft2(real(F1)+(1i*imag(F2))));
```

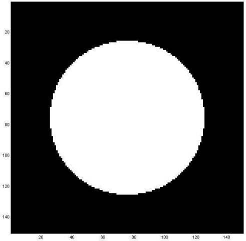


Cvičení V.

- Vytvořte fci vracející kruh:
  funciton K = kruh (R, N)
  % vrací binární kruhovou masku o poloměru R v matici NxN



(1.56) `kruh(4,10)`



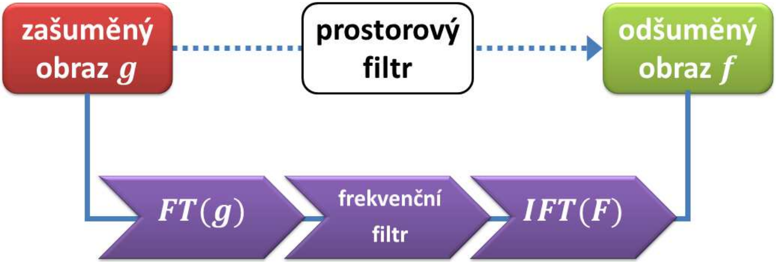
(1.57) `kruh(50,150)`

Řešení - Cvičení V.

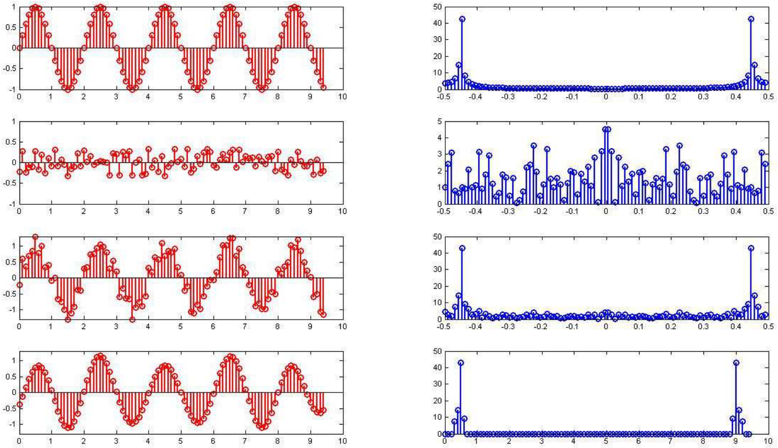
```
funciton K = kruh (R, N)
% vrací binární kruhovou masku o poloměru R v matici
NxN
[X,Y]=meshgrid(-(N-1)/2:(N-1)/2, -(N-1)/2:(N-1)/2);
K = double(X.^2 + Y.^2 < R^2);
end
```

Číslicová filtrace - jaký je její princip?

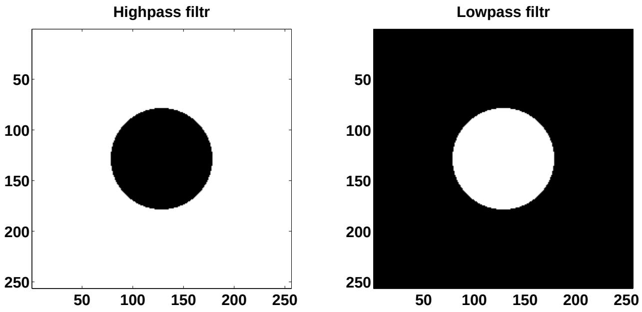
- $g = f + n$



Číslicová filtrace - jaký je její princip?



Obrazová filtrace - Highpass & lowpass filtry

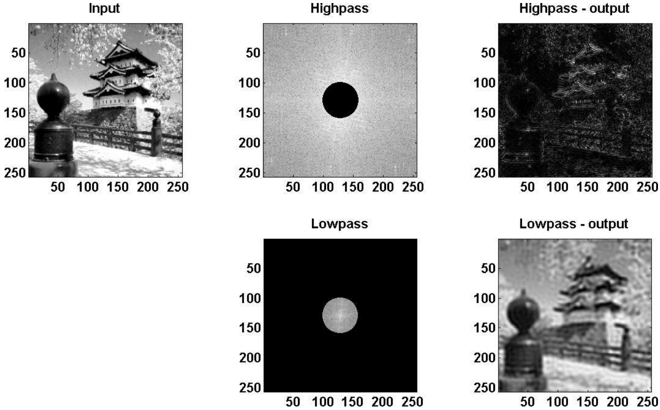


Cvičení VI.

- Vyzkoušejte Highpass & Lowpass filtr na `house.png`:

Cvičení VI.

- Vyzkoušejte Highpass & Lowpass filtr na `house.png`:



Řešení - Cvičení VI.

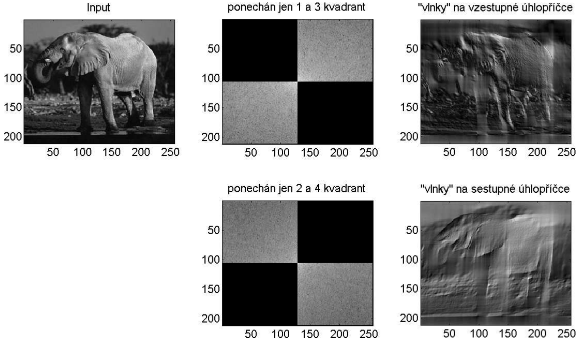
```
I=double(imread('house.png'));
M=kruh(30,size(I,1));
M1=fftshift(~M);
M2=fftshift(M);
FI=fft2(I);
K1=FI.*M1;
K2=FI.*M2;
zobr(abs(ifft2(K1)));
zobr(abs(ifft2(K2)));
```

Cvičení VIII. - Vynulování kvadrantů

- vynulujte u FT kvadrant (1.+3.) a (2.+4.) u slona
co se stane?

Cvičení VIII. - Vynulování kvadrantů

- vynulujte u FT kvadrant (1.+3.) a (2.+4.) u slona
co se stane?

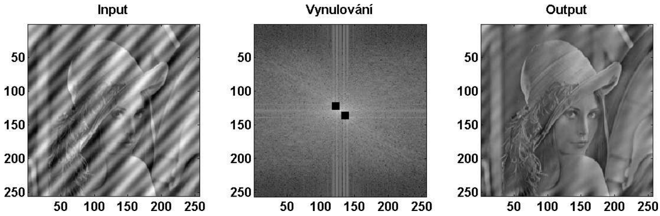


Cvičení IX. - Odstraňte poškození

- SVlnkama.pgm

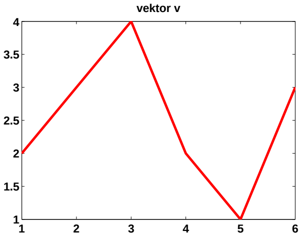
Cvičení IX. - Odstraňte poškození

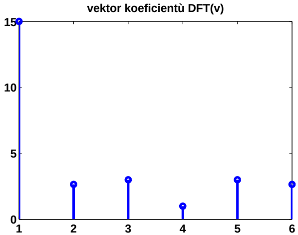
- `SVlnkama.pgm`



Cvičení VII.

- Vytvořte fci počítající DFT:
  funciton V = dft(v)
  % vrací vektor koeficientů DFT(v) o délce length(I)

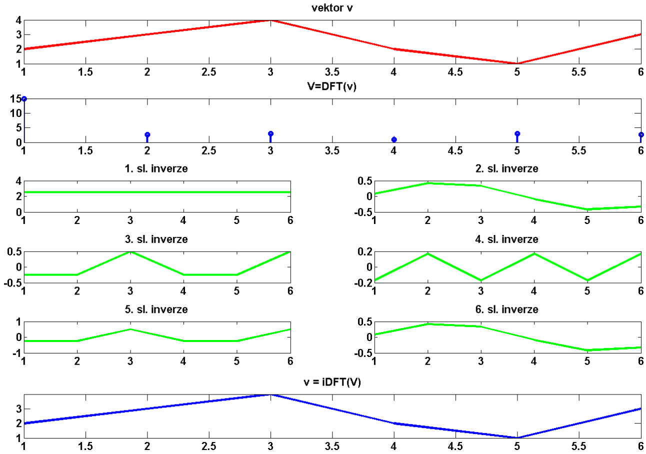




Řešení - Cvičení VII.

```
funciton K = dft(v)
% V - vektor koeficientů DFT(v) o délce length(I)
  N = length(v);
  forK = 1:N
    F(K)=sum(P.*exp(-2*pi*i/N*(K-1)*[0:N-1]));
  end
end
```

Řešení - Cvičení VII: `dft([2 3 4 2 1 3])`



Co jsme se dnes naučili:

- Výpočet a zobrazení FT - amplitudy, fáze, reálné a imaginární části
- umíme filtrovat ve frekvenční oblasti
- naprogramovali jsme si DFT

KONEC

Děkuji za pozornost !