

DSZOB, cvičenie 8.

Zadanie:

Úloha 1 Lineárna filtrácia obrazu v priestorovej doméne

Vykonajte filtráciu obrazu v priestorovej doméne s využitím konvolúcie. Použite nasledovné filtre:

- Gaussov filter
- Sobelov filter
- Laplacov filter

Postupujte nasledovne:

1. Načítajte si ľubovoľný obrázok (imread).
2. Prevedťte si obrázok na grayscale (rgb2gray).
 - a. Dobrovoľné: Môžete robiť aj s farebným rgb obrázkom (3 kanály).
3. Vytvorte si maticu filtra pre konvolúciu (pomôcka: fspecial).
4. Vykonajte konvolúciu (conv2).

V úlohe vizualizujte pôvodný obrázok a výstupy po filtriach.

Posudte ci pouzite filtracne jadra pri konvolucii su LP alebo HP! Zdôvodnite !

Postup vhodne dokumentuje (Code/Text bloky)!

Riešenie:

```
% Riesenie / Solution aaron_paul_1.0.jpg
% 1. Načítali sme ľubovoľný obrázok (pomocou funkcie: imread(img) ),%
% tento obrazok sme nasledne vizualizovali/zobrazily (pomocou funkcie: imshow() )
img = imread('aaron_paul_1.0.jpg');
figure
    imshow(img)
    title("Original image")
```

Original image



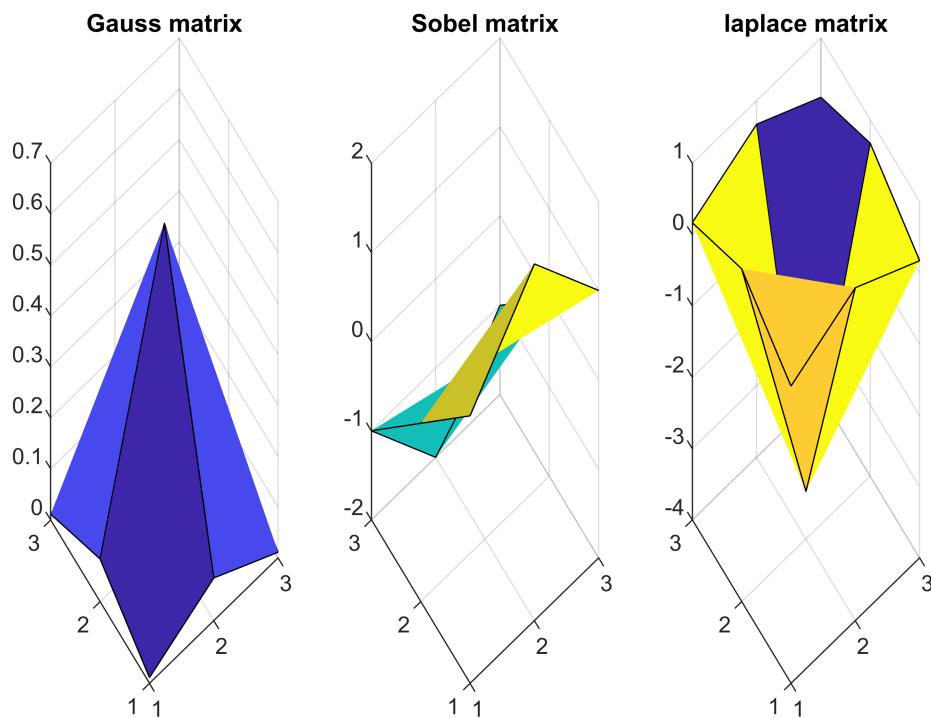
```
% 2. Previedli sme obrázok na grayscale (pomocou funkcie: rgb2gray(img) ).  
% tento obrazok sme nasledne vizualizovali/zobrazili (pomocou funkcie: imshow() )  
grayImg = rgb2gray(img);  
figure  
imshow (grayImg)  
title("Grayscale image")
```

Grayscale image



```
% 3. Vytvorili sme matice filtrov (Gauss, Sobel, Laplace) pre konvolúciu (pomocou
funkcie: fspecial() ).  
gaussMat = fspecial("gaussian");  
sobelMat = fspecial("sobel");  
laplaceMat = fspecial("laplacian");  
figure  
    layout1 = tiledlayout(1,3);  
    nexttile  
        surf(gaussMat)  
        title("Gauss matrix")  
    nexttile  
        surf(sobelMat)  
        title("Sobel matrix")  
    nexttile  
        surf(laplaceMat)  
        title("laplace matrix")  
    title(layout1,"Filter:")
```

Filter:



```
% 4. Vykonali sme konvolúciu (s pomocou funkcie: conv2() ).
```

```
% 4.1 Gauss
```

```
gaussConvImg = conv2(grayImg,gaussMat);  
% gaussToShow = log(abs(gaussConvImg));  
figure  
imshow(gaussConvImg, []) %[] su pre automatické preškalovanie  
visualizovaného obrázku  
title("Gauss convoluted img.")
```

Gauss convoluted img.



```
% 4.2 Sobel
sobelConvImg = conv2(grayImg,sobelMat);
% sobelToShow = log(abs(sobelConvImg));
figure
imshow(sobelConvImg, []) %[] su pre automatické preškalovanie
viualizovaného obrázku
title("Sobel convoluted img.")
```

Sobel convoluted img.



```
% 4.3 Laplace
laplaceConvImg = conv2(grayImg,laplaceMat);
% laplaceToShow = log(abs(laplaceConvImg));
figure
imshow(laplaceConvImg, []) %[] su pre automatické preškalovanie
viualizovaného obrázku
title("Laplace convoluted img.")
```

Laplace convoluted img.



Posudte ci pouzite filtracne jadra pri konvolucii su LP alebo HP! Zdôvodnite !

Použité filtre sú HP, alebo LP?:

- Gauss je LP filter, vzhľadom že to (vyhľadzuje obrazok, odstránením vysokých freq, a trochu je rozmažanejší od originálu) .
- Sobel je HP filter, vzhľadom že to (zobrazuje hlavne virazné okraje, s hlbkou) .
- Laplace je (HP) filter, (vzhľadom že to zobrazuje okraje, bez hlbky).

Úloha 2 Filtrácia obrazu v spektrálnej doméne

Vykonajte nasledovné filtrácie obrazu v spektrálnej doméne:

- Ideálny HP filter
- Ideálny LP filter
- Gaussov HP filter
- Gaussov LP filter

Postupujte nasledovne:

1. Načítajte si obrázok a preveďte ho na grayscale (dobrovoľné: RGB).
2. Vypočítajte Fourierovu transformáciu nad obrázkom (fft2).
3. Správne si posuňte spektrum (prehodenie kvadrantov) (fftshift).

4. Vytvorte si ideálny filter (pomôcka: definícia funkcie pre kruh s využitím polomeru a stredového bodu).
5. Vykonajte filtračiu. (násobenie)
6. Inverzne posuňte spektrum (ifftshift).
7. Vykonajte spätnú transformáciu (ifft2) a vizualizujte obrázok.

Rovnako postupujte aj pri ostatných filtroch. Pri vytvaraní Gaussovoho filtra cez fspecial si správne naškálujte filter (pozn. min = 0, max = 1). Vizualizujte aj jednotlivé filtre, ktoré ste vytvorili v tejto úlohe. **Porovnajte a zhodnodťte vplyv jednotlivých filtrov na výsledné vizualizácie.**

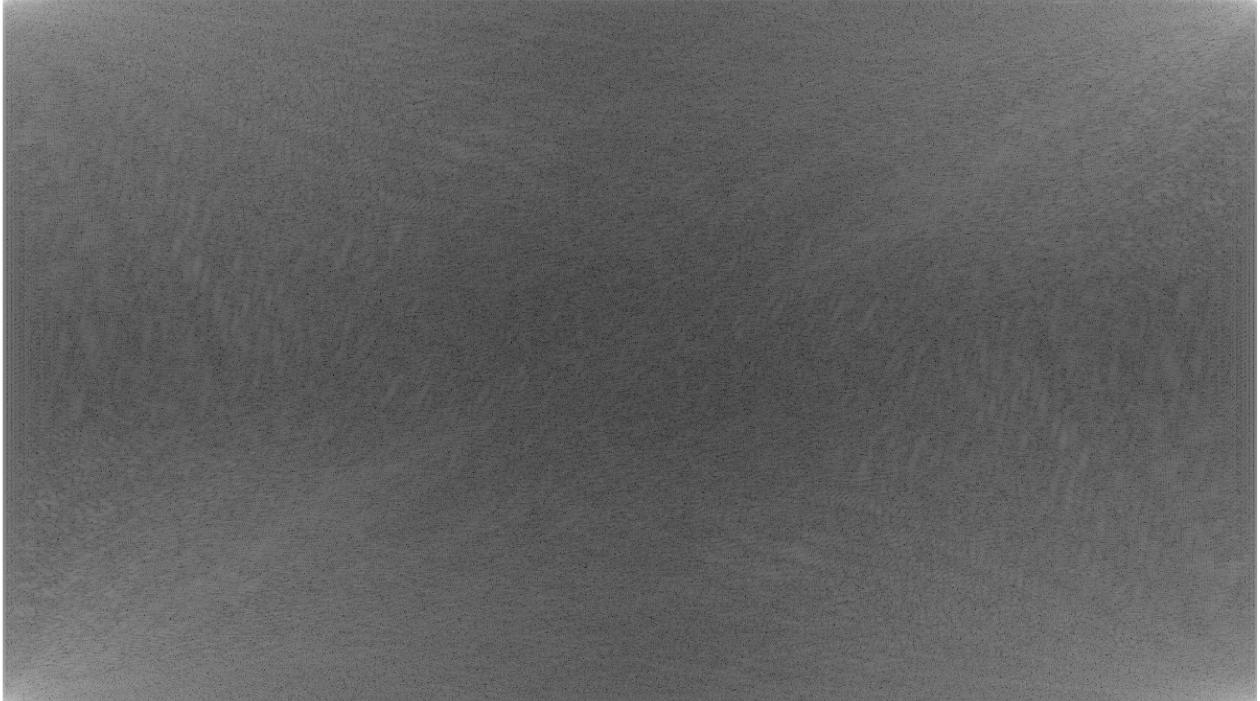
```
% Riesenie / Solution  
% 1. Načítajte si obrázok a preveďte ho na grayscale (dobrovoľné: RGB).  
secondImg = img;  
secondGrayImg = grayImg;
```

```
% 2. Vypočítajte Fourierovu transformáciu nad obrázkom (fft2).
```

```
fftGrayImg = fft2(secondGrayImg);  
toShowFftGrayImg = log(abs(fftGrayImg));
```

```
figure  
imshow(toShowFftGrayImg, [])  
title("Grayscale img after FFT.")
```

Grayscale img after FFT.



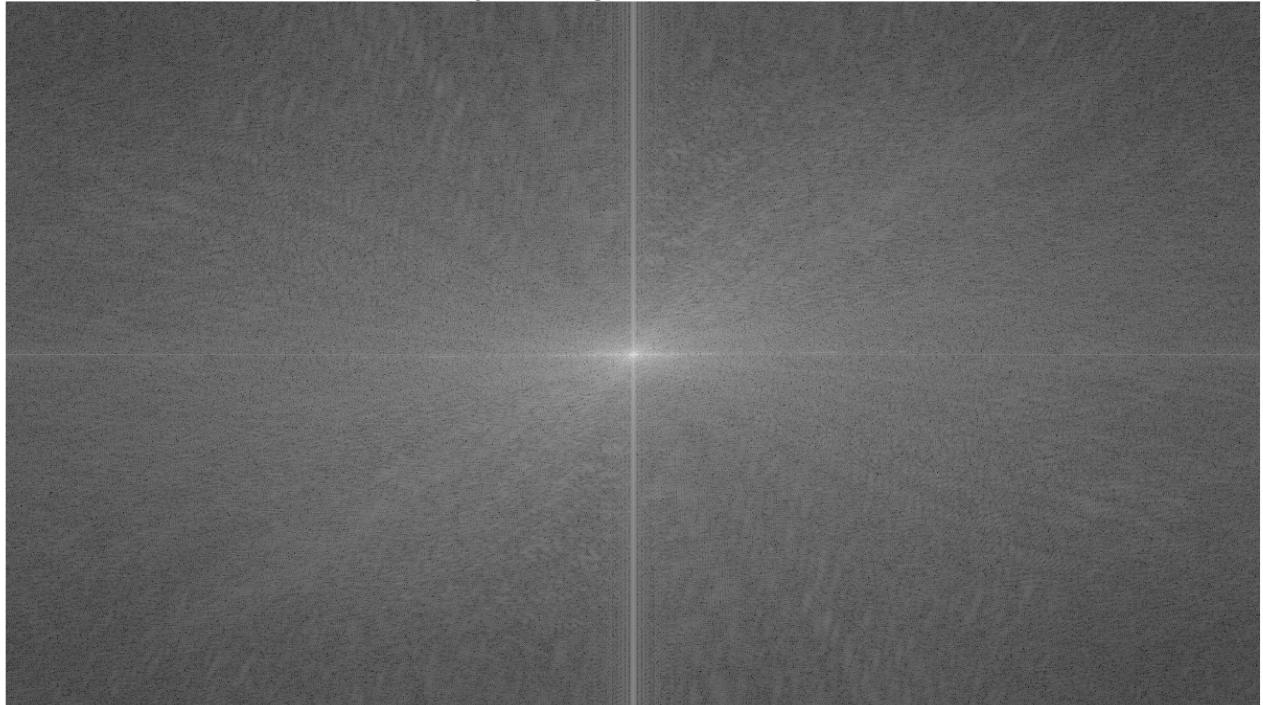
```
% 3. Správne si posuňte spektrum (prehodenie kvadrantov) (ifftshift).
```

```

shiftGrayImg = fftshift(fftGrayImg);
toShowShiftGrayImg = log(abs(shiftGrayImg));
figure
imshow(toShowShiftGrayImg, [])
title("Grayscale image after FFT and shift.")

```

Grayscale image after FFT and shift.



% 4. Vytvorili sme si ideálne filtre (jednak pomocou funkcie ffspec pre gaussov filter a vnoreným cyklom pre ideal filter).

```

imgSize = size(secondGrayImg);
%4.1 Ideálny LP filter
radius = 25;
center = round(imgSize / 2);
lpIdealfilter = zeros(imgSize);

for y = (center(1) - radius):(center(1) + radius) % s predpokladom toho ze
vzdialenosť od stredu by nemala presahovať radius bodov od stredu
    for x = (center(2) - radius):(center(2) + radius)
        dist = norm(center - [y,x]);
        if(dist <= radius)
            lpIdealfilter(y,x) = 1;
        end
    end
end
%4.2 Ideálny HP filter
hpIdealfilter = 1 - lpIdealfilter;
%4.3 Gaussov LP filter

```

```

lpGaussFilt = fspecial("gaussian", [imgSize], 25);
%4.4 Gaussov HP filter
hpGaussFilt = 1 - lpGaussFilt;
figure
layout1 = tiledlayout(1,4,"TileSpacing","compact");
nexttile
imshow(lpIdealfilter, [])
title("Ideal LP matrix")
nexttile
imshow(hpIdealfilter, [])
title("Ideal HP matrix")
nexttile
imshow(lpGaussFilt, [])
title("Gauss LP matrix")
nexttile
imshow(hpGaussFilt, [])
title("Gauss HP matrix")
title(layout1,"Filter:")

```

Filter:



% 5. Vykonali sme nasobenie/filtráciu source signalu a filtra, a vizualizovali sme obrázok.

```

idealLP = shiftGrayImg.*lpIdealfilter;
idealHP = shiftGrayImg.*hpIdealfilter;
gaussLP = shiftGrayImg.*lpGaussFilt;
gaussHP = shiftGrayImg.*hpGaussFilt;

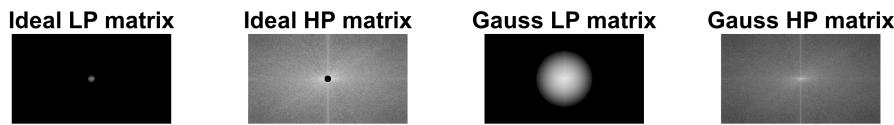
```

```

figure
    layout1 = tiledlayout(1,4,"TileSpacing","compact");
        nexttile
            imshow(log(abs(idealLP)), [])
            title("Ideal LP matrix")
        nexttile
            imshow(log(abs(idealHP)), [])
            title("Ideal HP matrix")
        nexttile
            imshow(log(abs(gaussLP)), [])
            title("Gauss LP matrix")
        nexttile
            imshow(log(abs(gaussHP)), [])
            title("Gauss HP matrix")
    title(layout1,"Images in spectral domain after filtering with:")

```

Images in spectral domain after filtering with:



% 6. Inverzne sme posunuly spektrum (s pomocou funkcie: ifftshift) a vizualizovali sme obrázok.

```

idealLP = ifftshift(idealLP);
idealHP = ifftshift(idealHP);
gaussLP = ifftshift(gaussLP);
gaussHP = ifftshift(gaussHP);

```

```

figure
    layout1 = tiledlayout(1,4,"TileSpacing","compact");
        nexttile

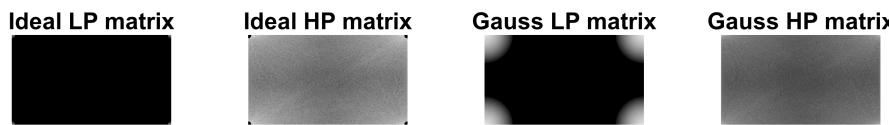
```

```

imshow(log(abs(idealLP)), [])
title("Ideal LP matrix")
nexttile
imshow(log(abs(idealHP)), [])
title("Ideal HP matrix")
nexttile
imshow(log(abs(gaussLP)), [])
title("Gauss LP matrix")
nexttile
imshow(log(abs(gaussHP)), [])
title("Gauss HP matrix")
title(layout1,"Images in spectral domain after back-shifting and filtering
with:")

```

Images in spectral domain after back-shifting and filtering with:



```

% 7. Vykonali sme spätnú transformáciu (s pomocou funkcie: ifft2) a vizualizovali
sme obrázok.
finalIdealLP = ifft2(idealLP);
finalIdealHP = ifft2(idealHP);
finalGaussLP = ifft2(gaussLP);
finalGaussHP = ifft2(gaussHP);

figure
imshow(finalIdealLP, [])

```

Warning: Displaying real part of complex input.

```
title("Images filtered with: Ideal LP matrix")
```

Images filtered with: Ideal LP matrix



```
figure  
imshow(finalIdealLP, [])
```

Warning: Displaying real part of complex input.

```
title("Images filtered with: Ideal HP matrix")
```

Images filtered with: Ideal HP matrix



```
figure  
    imshow(finalGaussLP, [])
```

Warning: Displaying real part of complex input.

```
title("Images filtered with: Gauss LP matrix")
```

Images filtered with: Gauss LP matrix



```
figure  
    imshow(finalGaussHP, [])
```

Warning: Displaying real part of complex input.

```
title("Images filtered with: Gauss HP matrix")
```



Zhodnotenie vplyvu jednotlivých filtrov na výsledné vizualizácie:

Ideal LP: Tento je pomerne najmenej rozoznatelný (je rozmazaný s vlnením), pričom sa v nom nachádza i vlnenie(ringing effect)

Ideal HP: Z tohto máme značne rozoznatelný obrazok, s najviac zvýrazneními črtamy a zvlením (vzhľadom na to že sme použily idealny filter).

Gauss LP: Tento obrazok, je roznamazanejší ako jeho HP spoločnýk, ale stále ma lepšie rozlišenie ako jeho idealny konkurent.

Gauss HP: obrazok filtrovaný tímto filtrom, je nabližšá k pôvodnému grayscale obrazku, avšak nie je v nom vidieť odfiltrovanie nízších frekvencií, predpokladám že tento filter vzhľadom na jeho radius odfiltroval limitujúcu skálu frekvencií.