

# Домашна работа

## Филтрирање на слика во фреквенциски домен

Да се креира јупитер тетратка со SoS кернел, во која ќе се прави филтрирање на слика во фреквенциски домен. Конечната функционалност на тетратката треба да биде dashboard со следниве карактеристики:

- Да содржи објаснување на процесот за филтрирање на слика.
  - Сите равенки во објаснувањето да бидат напишани во Latex.
- Да содржи интерактивна визуелизација:
  - Лизгач со кој ќе се овозможи избирање на праговите на ниска и висока фреквенција.
  - Со избирање на праговите, автоматски под лизгачот да се прикажат филтрираната слика во просторен домен и соодветниот кернел, кој одговара на праговите избрани со лизгачот, а со кој е филтрирана сликата во фреквенциски домен.

Ви оставаме голема слобода во имплементацијата на тетратката, она што е важно е да користите барем два програмски јазика (препорачуваме Python i Matlab/Octave) и да произведете интерактивна визуелизација како на долното видео:

<https://drive.google.com/open?id=10EVvbowidtGXILZIZrNlqIYGBxREEdh7>

### ТЕОРИЈА:

За да филтрираме слика во фреквенциски домен потребно е да ја помножимо сликата во фреквенциски домен со филтерот (кернел) кој е исто така во фреквенциски домен.

$$I_{f,filt} = H I_f \quad (1)$$

За таа цел потребно е да се вметне Octave функција која ќе креира “Band pass gaussian filter” во фреквенциски домен. Bandpass гаусовиот филтер во фреквенциски домен е дефиниран со равенката:

$$H(i, j) = e^{-\frac{d(i, j)^2}{f_h^2}} \left( 1 - e^{-\frac{d(i, j)^2}{f_l^2}} \right) \quad (2)$$

Каде што  $d(i, j)$  е евклидовото растојанието на даден пиксел од центарот на сликата, додека  $f_h$  и  $f_l$  се праговите на високите и ниските фреквенции кои сакаме да ги филтрираме.

Конверзија на 2D сликата од просторен во фреквенциски домен се прави со користење на Фуриева трансформација. Фуриевата трансформација е дефинирана како:

$$F(u, v) = \iint_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

А во дискретниот домен горната равенка може да биде преведена како

$$F(u, v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[m, n] \cdot e^{-i2\pi(umx_0+vn y_0)} \quad (3)$$

Конверзија на 2D сликата од фреквенциски во просторен домен се прави со користење на инверзна Фуриева трансформација. Инверзната Фуриева трансформација е дефинирана како:

$$f(x, y) = \iint_{-\infty}^{\infty} F(u, v) e^{i2\pi(ux+vy)} du dv$$

А во дискретниот домен горната равенка може да биде преведена како

$$f(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F(m, n) \cdot e^{i2\pi(xmu_0+ynv_0)} \quad (4)$$

## Имплементација

Octave функцијата која прави Фуриева трансформација на слика е *fft2*. За да ја вратите сликата од фреквенциски во просторен домен се користи инверзна Фуриева трансформација, која во Octave се постигнува со командата *ifft2*.

**Забелешка:** По конвенција, Octave функцијата *fft2* ги преставува фреквенциите од 0 до N, каде централната фреквенција се наоѓа на позиција (0,0) во матрицата. За поставување на централната фреквенција во центарот на сликата во фреквентен домен се користи командата *fftshift*. *fftshift* ја трансформира сликата така што фреквенциите се преставени од - N/2 -> N/2. За подетални информации може да ја погледнете документацијата на *fftshift*.

Потоа да се направи филтрирање на сликата *Barbara.tif* со неколку различни Bandpass Гаусови филтри со фреквенциски прагови *fl[1, 101]* и *fh[1, 101]* со чекор 10. Matlab/Octave кодот за создавање на филтер е даден на крајот од документот, види *create\_filter (nx,ny,d0,d1)*.

## Визуелизација

Поречно е да се направи интерактивна визуелизација на филтрираните слики со користење на *plotly widgets*. Визуелизацијата треба да овозможи да се изберат

праговите на ниска и висока фреквенција, а потоа да се прикаже филтрираната слика во просторен домен и кернелот во фреквенциски домен кои соодветствуваат со дадените фреквенциски прагови. Визуелизацијата треба да изгледа како на видеото кое може да го најдете на следниот линк:

<https://drive.google.com/open?id=10EVvbowidtGXILZIZrNlqIYGBxREEdh7>

## Екстензии

Доколку сакате да експериментирате со можностите на Plot.ly и Jupyter Notebooks, тогаш еве неколку предлози за екстензии. Иако нема да даваме екстра поени за имплементација на екстензиите, никогаш не знаете како тоа може да ви се исплати:)

- Направете опција да се филтрира било која слика, од фајл или од интернет
- Овозможете ја истава функционалност за аудио или видео
- Имплементирајте библиотека на стандардни филтри, Гаусов, Чебишев, елиптичен и опција за споредување преку dashboard
- Бидете креативни, додадете функционалноста на која не сме помислиле

## Извештај

Вметнете го текстот од секцијата со поднаслов 'ТЕОРИЈА' на почетокот од тетратката. Сите равенки треба да бидат напишани во Latex. Под текстот вметнете го кодот за филтрирањето и интерактивната визуелизација.

Домашната треба да ја прикачите на вашиот GitHub репозиториум, а линкот до GitHub репозиториумот треба да го прикачите на courses најдоцна до 09.12.2019 во 23:59 ч.

## Корисни функции

```
function filter = create_filter(nx,ny,d0,d1)

filter = ones(nx,ny);

for i = 0:nx-1
    for j = 0:ny-1
        dist= sqrt((i-nx/2)^2 + (j-ny/2)^2);
        filter(i+1,j+1) = exp(-(dist^2)/(d1^2)).*(1.0-exp(-(dist^2)/(d0^2)));
    end
end
filter(nx/2+1,ny/2+1)=1;
end
```

Еве уште неколку функции што ќе помогнат со имплементацијата

%Трансформација на сликата од просторен во фреквенциски домен:

```
im_fft = fftshift(fft2(im));
```

% Филтрираната слика се добива како инверзна Фуриева трансформација на производот од филтерот и сликата во фреквенциски домен

```
filtered_image = abs(ifft2(ifftshift(create_filter(nx,ny,i,j).*im_fft)));
```