Advanced Image Processing - Fourier transformation

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

20.11.2019

Discrete Fourier transformation - 1D

Definition

$$F_k = \mathscr{F}[\vec{f}]_k = \sum_{n=0}^{N-1} f_n \cdot e^{\frac{-2\pi i n k}{N}}$$

Inverse

$$f_n = \mathscr{F}^{-1}[\vec{F}]_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k \cdot e^{\frac{2\pi i n k}{N}}$$

2D FFT in Matlab

fft2

fft2(I) - returns the Fourier transform of grayscale image I.

ifft2

ifft2(F) - returns the inverse Fourier transform of F.

fftshift

fftshift(fft2(I)) - shifts the output of fft2 so that the zero-frequency is in the middle of the image. Calling this twice returns the original F.

Úloha

Display the Fourier transform of zatisie.

Complex numbers

Problem I - Complex numbers

FFT in matlab returns complex numbers! IFFT also returns complex numbers. We therefore have to learn how to work with them.

Working with complex numbers

- i, j - constants representing the complex number - watch out for shadowing
- real(c) - real part of the complex number c
- imag(c) - imaginary part of the complex number c
- abs(c) - absolute value of the number c
- angle(c) - angle of the complex number c

Exercise

Problem II

It is necessary to work in double and deal with the fact that the values of transform are too big in the middle.

Display function as a solution

```
function F = zobrfft(I)
    F = fftshift(fft2(im2double(I)));
    imagesc(log(abs(F)+1));
    colormap(gray);
end
```

Exercise

Exercise

Display the absolute values of Fourier transform for the images pruhyhoriz.pgm, pruhyvert.pgm, pruhysikme.pgm, zatisie.pgm, boat.pgm, waveboat.pgm. Display the angles for some of them.

Úloha

Transform the image boat, use the inverse transform, but only keep the information about the angle or information about the absolute value, or just the real part and just the imaginary part. Try to keep only one of the quadrants of the transformed image and display the image after inverse transform.

Spectral domain filtration - fundamentals

Positions of the frequencies in the spectral domain

In the spectral domain (after transformation) the lower frequencies are closer to the center and higher ones are at the edges.

Meaning

High frequencies carry details. If we only keep the high frequencies we will obtain mostly edges. If we only keep the low frequencies we will get a blurred image.

Ideal highpass a lowpass

Highpass

We transform the image and set the low frequencies to zero and then we perform the inverse transform.

Lowpass

The same process but with high frequencies.

Cut-off frequency

We call the frequency at which we set our limit the cut-off frequency

Butterworth filter

Butterworth filter

$$H = \frac{1}{1 + (\sqrt{2} - 1)\left(\frac{D}{D_0}\right)^{2n}}$$

D is the Euclidian distance from the center.

Function for the butterworth filter

Open the butterhigh.m from the zip file.

Úloha

Display this filter using imagesc. Use this filter on some image in the spectral domain. Modify the function so we can use it for lowpass filtering.

Ideal highpass and lowpass

Úloha

Create functions which generate the ideal lowpass and ideal highpass filters. You can use parts of butterhigh.m.

Exercise

Apply filters with various cut-off frequencies on boat.pgm and zatisie.pgm.

Filtration of periodic noise

Periodic noise

Periodic noise can be seen in the spectral domain as a pair or multiple pairs of areas with higher amplitude. We can therefore use change the representation in the spectral domain and perform the inverse transformation. This will suppress the noise in the original image.

Exercise

Suppress the noise the images tree.png, waveboat.pgm and periodic.png.