WS 2019/20	Exercises Digital Image Processing	Task No. 12

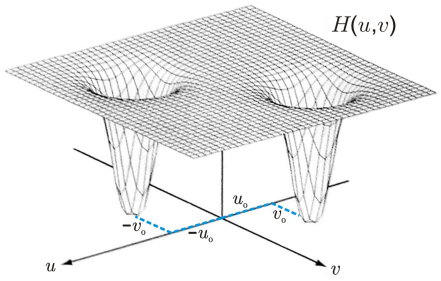
Task 12.1 *Image restoration in the frequency domain*

There are some given pictures with different sizes and a superimposed wave. The parameters for the wave are $k_x = 0.05$ and $k_y = 0.15$. Your task now is to restore the original image again.

a) Write a function that display the Fourier spectrum of an input image. The difference between the original and the distorted image is then visible.

Tip: To get a reasonable Fourier spectrum, you should multiply all the values with 255 and then clip the output image.

a) Write a function that eliminate the disturbance using the following *Butterworth Notch filter* (see Figure).



$$H(u,v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^n} \qquad (1)$$

with D_0 as a Cut-Off frequency, u and v as center of the fourier transformed images, $D_1(u,v)$ and $D_2(u,v)$ with

$$D_1(u,v) = \sqrt{(u - \frac{M}{2} - u_0)^2 + (v - \frac{N}{2} - v_0)^2} \qquad (2)$$

$$D_2(u,v) = \sqrt{(u - \frac{M}{2} + u_0)^2 + (v - \frac{N}{2} + v_0)^2} \qquad (3)$$

where (u_0, v_0) and $(-u_0, -v_0)$ are the distance from the disturbance centers to the image center (u, v). see exercise 10.1b.

Tip: To get the coordinates of the disturbance centers, you can take the Fourier spectra of the input images in a graphics program like GIMP to load and count the pixels.

Tip: There following functions are available in DIPLib.

`void fourier_transform(const ComplexImage& input, ComplexImage& output)`
Performs a Fourier transform of the complex-image `<input>` and place it in `<output>`.

`void inverse_fourier_transform(const ComplexImage& input, ComplexImage& output)`
Performs an inverse Fourier transform of the complex-image `<input>` and place it in `<output>`.

Continued →

In the DIP library, there is no function to load an image into a ComplexImage. To do that you could use a GrayImage to load an image and then copy that image in a ComplexImage as follows:

```
GrayImage temp;
temp.load();
ComplexImage cImage;
cImage.copy(temp);
```