

# Semnale și sisteme

**Lucrare de laborator nr. 12**

## **FILTRAREA SECVENTELOR BIDIMENSIONALE IN DOMENIUL FRECVENTEI**

- Transformata Fourier discreta bidimensională (2D-DFT) a unui semnal bidimensional  $f(x, y)$  de dimensiune finită  $M \times N$ :

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

- Transformata Fourier discreta bidimensională inversă:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

- Pentru  $u=v=0$  se obține:  $F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$ . Prin urmare, componenta de ordinul zero este valoarea medie a imaginii.

- Pași necesari filtrării în domeniul frecvenței:

1. Se înmulțește imaginea de intrare cu  $(-1)^{x+y}$  pentru a centra transformata.
2. Se calculează 2D DFT  $F(u, v)$
3. Se înmulțește  $F(u, v)$  cu o funcție de filtrare  $H(u, v)$ :  $G(u, v) = F(u, v)H(u, v)$
4. Se calculează transformata Fourier discreta inversă a lui  $G(u, v)$
5. Se extrage partea reală a lui  $g(x, y)$
6. Se înmulțește  $g(x, y)$  cu  $(-1)^{x+y}$

- De ce  $(-1)^x$ ?

$$(-1)^x = (\cos \pi + j \sin \pi)^x = (e^{j\pi})^x = e^{j\pi x}$$

$$\begin{aligned} \frac{1}{M} \sum_{x=0}^{M-1} (-1)^x f(x) e^{-j2\pi ux/M} &= \frac{1}{M} \sum_{x=0}^{M-1} e^{j\pi x} f(x) e^{-j2\pi ux/M} = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{j\pi x - j2\pi ux/M} \\ &= \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-2j\pi x(u - \frac{M}{2})/M} = F\left(u - \frac{M}{2}\right) \end{aligned}$$

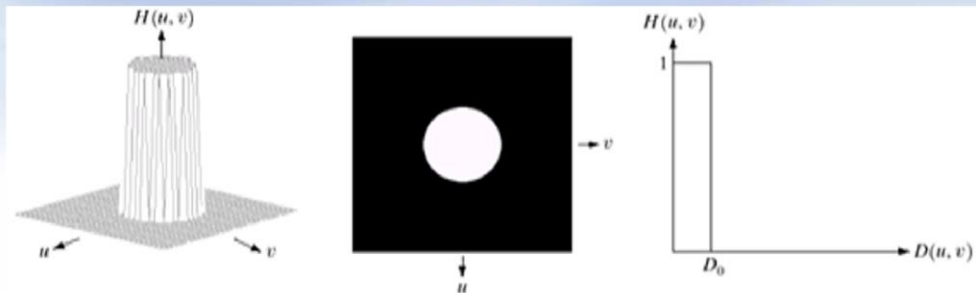
### Filtre trece-jos

#### □ Filtru ideal

$$H(u, v) = \begin{cases} 1, D(u, v) \leq D_0 \\ 0, D(u, v) > D_0 \end{cases}$$

Centrul este la  $(u, v) = (M/2, N/2)$ ,  $D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$

Frecventa de taiere este  $D_0$

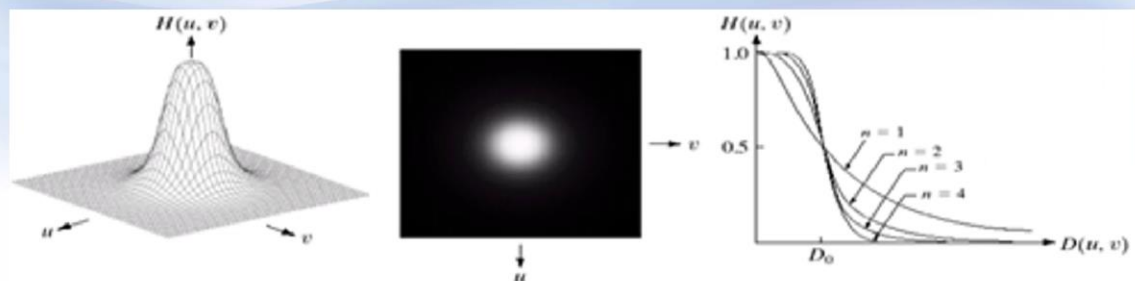


### Filtre trece-jos

#### □ Filtru trece-jos Butterworth

Filtru trece-jos Butterworth de ordin  $n$ :  $H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$

La frecventa  $D_0$ ,  $H(u, v) = 0.5$ .

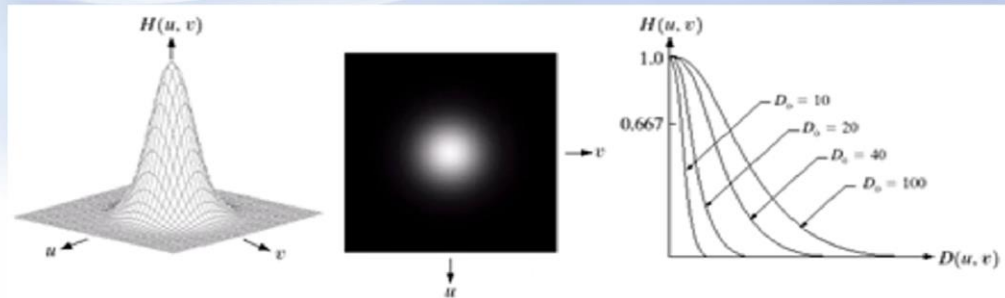


### Filtre trece-jos

□ Filtru trece-jos Gaussian  $H(u, v) = e^{-D^2(u, v) / 2\sigma^2}$

Pentru  $\sigma = D_0$ ,  $H(u, v) = e^{-D^2(u, v) / 2D_0^2}$

Daca  $D(u, v) = D_0$ ,  $H(u, v) = 0.667$



### Filtre trece-sus

□ Filtru ideal

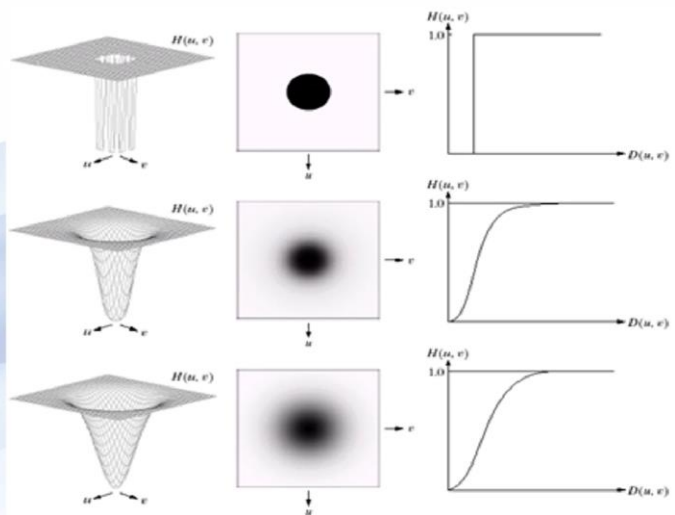
$$H(u, v) = \begin{cases} 0, & D(u, v) \leq D_0 \\ 1, & D(u, v) > D_0 \end{cases}$$

□ Filtru trece-sus Butterworth

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

□ Filtru trece-sus Gaussian

$$H(u, v) = 1 - e^{-D^2(u, v) / 2\sigma^2}$$



```

function H = imfreqfilter(type, M, N, D0, n)
u=0:(M-1); v=0:(N-1); [V,U]=meshgrid(v,u);
D = sqrt((U-M/2).^2+(V-N/2).^2);
switch type
    case 'LPF'
        H=double(D<=D0); % lowpass filter
    case 'BLPF'
        H=1./(1.+(D./D0).^(2*n)); % Butterworth lowpass filter of order n
    case 'GLPF'
        H=exp((-D.^2)/(2*D0.^2)); % Gaussian lowpass filter
    case 'HPF',
        H=double(D>D0); % highpass filter
    case 'BHPF'
        H=1./(1.+(D0./D).^(2*n)); % Butterworth highpass filter of order n
    case 'GHPF'
        H=1-exp((-D.^2)/(2*D0.^2)); % Gaussian highpass filter
end

img = imread('peppers.png');
if (ndims(img)==3) img = rgb2gray(img); end
F = fftshift(fft2(double(img)));
figure; imshow(img);
figure; imshow(log(abs(F)),[]); title('Magnitude')
figure; imshow(angle(F)); title('Phase')
filtertype = 'LPF'; %BLPF, GLPF, HPF, BHPF, GHPF
D0 = 27;
n = 1; % order for Butterworth filter
[M, N] = size(img);
H = imfreqfilter(filtertype, M, N, D0, n);
figure; surf(H), title(filtertype);
G=H.*F;
figure; imshow(log(abs(G)),[]); title('Filterred Magnitude')
figure; imshow(angle(G),[]); title('Filterred Phase')
imgf=real(ifft2(ifftshift(double(G))));
figure,imshow(imgf,[ ]); title('Filterred Image')

```

## Exercitii

### 1. Utilizarea filtrelor din domeniul spatial in domeniul frecventei

- a. Incarcati imaginea 'peppers.png' in Matlab
- b. Transformati imaginea intr-o imagine cu nuante de gri
- c. Creati un filtru in domeniul spatial pentru detectia muchiilor orizontale (ex. filtrul Sobel)
- d. Creati un filtru in domeniul spatial pentru detectia muchiilor verticale (cititi documentatia MATLAB a functiei *fspecial*).
- e. Transformati ambele filtre in domeniul frecventei utilizand functia *freqz2*
- f. Transformati imaginea in domeniul frecventei
- g. Realizati procesarile necesare in domeniul frecventei
- h. Transformati rezultatele in domeniul spatial
- i. Insumati imaginile obtinute
- j. Afisati imaginea rezultata

## Exercitii

### 2. Inversarea fazei transformatelor Fourier discrete 2D

- a. Incarcati imaginile 'saturn.png' si 'cameraman.tif'
- b. Transformati imaginile in imagini cu nuante de gri
- c. Redimensionati imaginea 'saturn.png' la dimensiunea imaginii 'cameraman.tif' utilizand functia *imresize*
- d. Calculati transformatele Fourier discrete ale celor doua imagini
- e. Afisati amplitudinea si faza transformatelor
- f. Inversati fazele celor doua transformate
- g. Calculati transformata Fourier discreta inversa
- h. Afisati imaginile rezultate