DIGITALNA OBRADA SLIKE

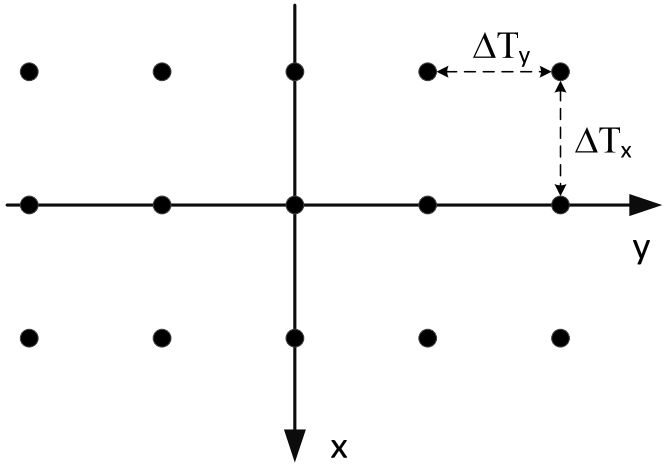
ČAS 4 – DISKRETNA FURIJEOVA TRANFSORMACIJA SLIKE, FILTRIRANJE SLIKE U FREKVENCIJSKOM DOMENU, RESTAURACIJA SLIKE

Diskretna Furijeova transformacija slike

Furijeova transformacija 2D signala

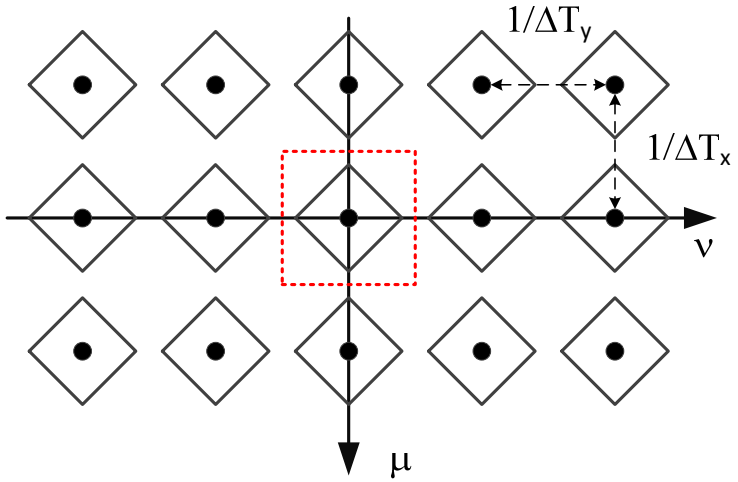
$$F(\mu,\nu)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}dtdz \quad \leftrightarrow \quad f(t,z)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t+\nu z)}d\mu d\nu$$

diskretna funkcija



$$f(x,y) \Leftrightarrow F(\mu,\nu)$$

periodičan spektar



Diskretna Furijeova transformacija 2D signala

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

$$u=0,1,2,\ldots,M-1,\quad v=0,1,2,\ldots,N-1 \qquad x=0,1,2,\ldots,M-1,\quad y=0,1,2,\ldots,N-1$$

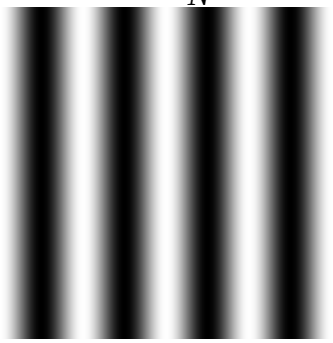
$$\mu=\frac{u}{M\Delta T_x} \qquad \nu=\frac{v}{N\Delta T_y}$$

$$\Delta\mu=\frac{1}{M\Delta T_x} \qquad \Delta\nu=\frac{1}{N\Delta T_y}$$

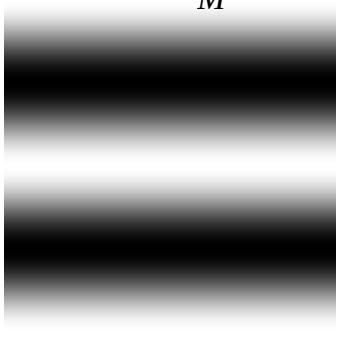
Primeri 2D kosinusnih funkcija

```
M = 200; N = 200;
[y x] = meshgrid(1:N, 1:M);

f1 = cos(8*pi*y/N);
figure; imshow(f1, []);

f2 = cos(4*pi*x/M);
figure; imshow(f2, []);

f3 = cos(16*pi*x/M + 4*pi*y/N);
figure; imshow(f3, []);

f4 = cos(8*pi*x/M + 8*pi*y/N);
figure; imshow(f4, []);

f5 = cos(8*pi*x/M) + cos(8*pi*y/N);
figure; imshow(f5, []);

f6 = cos(6*pi*x/M + 20*pi*y/M) +
cos(14*pi*x/M + 4*pi*y/M);
figure; imshow(f6, []);
```

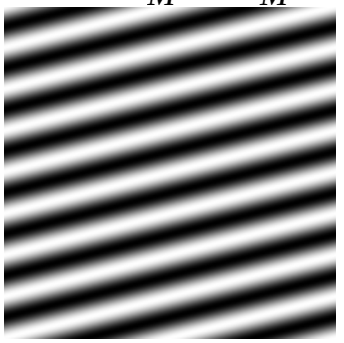
$\cos(8\pi \frac{y}{N})$

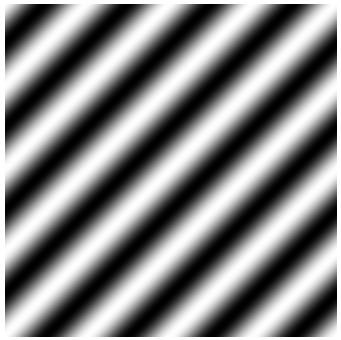


$\cos(4\pi \frac{x}{M})$

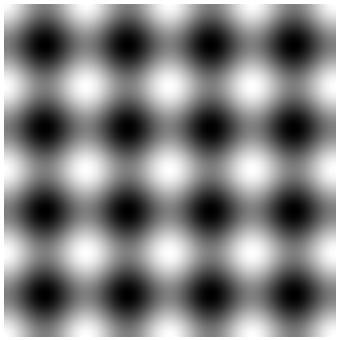


$\cos(16\pi \frac{x}{M} + 4\pi \frac{y}{M})$

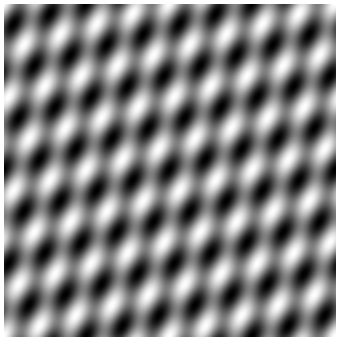




$\cos(8\pi \frac{x}{M} + 8\pi \frac{y}{M})$



$\cos(8\pi \frac{x}{M}) + \cos(8\pi \frac{y}{M})$

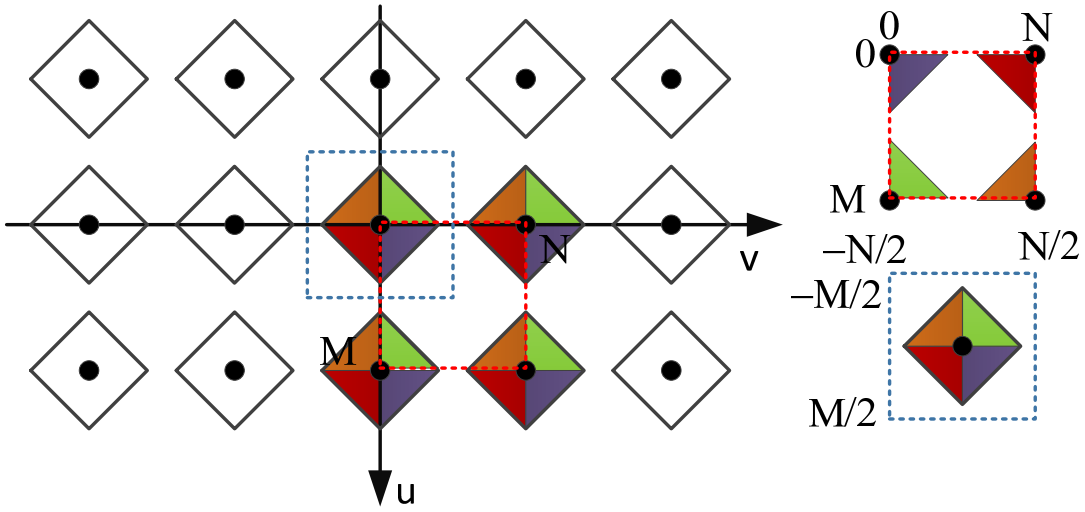


$\cos(6\pi \frac{x}{M} + 20\pi \frac{y}{M}) + \cos(14\pi \frac{x}{M} + 4\pi \frac{y}{M})$

Diskretna 2D Furijeova transformacija u Matlabu

`fft2(matrica_slike)`

`ifft2(matrica_spektra)`



Funkcija fft2 vraća diskretnu Furijeovu transformaciju slike, diskretizovane u opsegu 0-1/ΔT. Ova matrica je uokvirena crvenim kvadratom na slici. Vidi se da su učestanosti koje odgovaraju nižem delu spektra raspoređene po ćoškovima ove matrice. Kako bi se dobio spektar uokviren plavim kvadratom potrebno je izvršiti pomeranje spektra za -1/2ΔT po obe dimenzije. Ovo se može postići množenjem ulazne sekvence sa $(-1)^{x+y}$ pre transformacije.

`fftshift(matrica)`

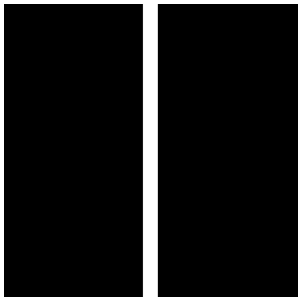
Obavlja potrebno preuređivanje izlaza fft2 funkcije.

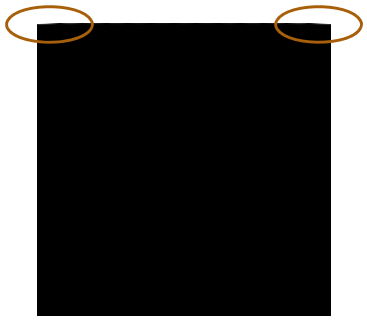
`ifftshift(matrica)`

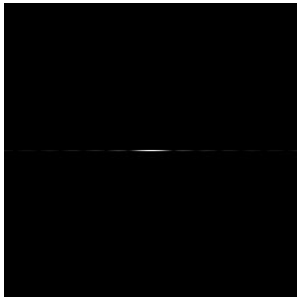
Vraća spektar u standarnu formu. Neophodno je pre inverzne transformacije u slučaju da je korišćen fftshift.

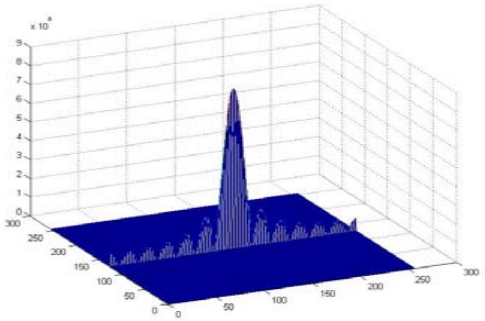
Primer – vertikalna linija

```
f = imread('stripe.png');
figure; imshow(f);

F = fft2(f);

figure; imshow(abs(F),[]);

F = fftshift(fft2(f));

figure; imshow(abs(F),[]);

figure; mesh(abs(F));
```



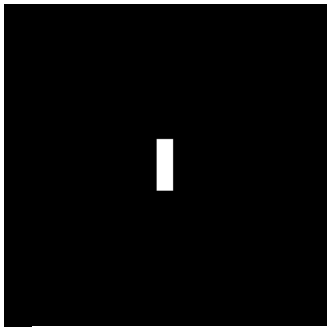


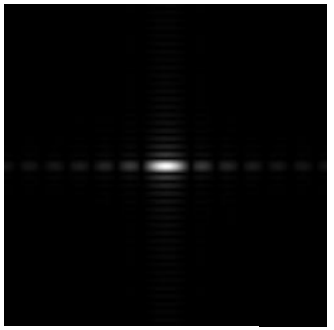


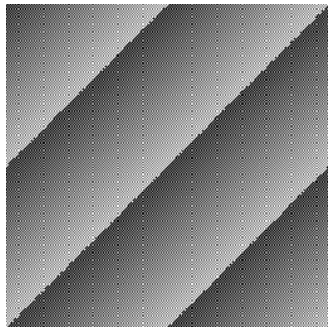


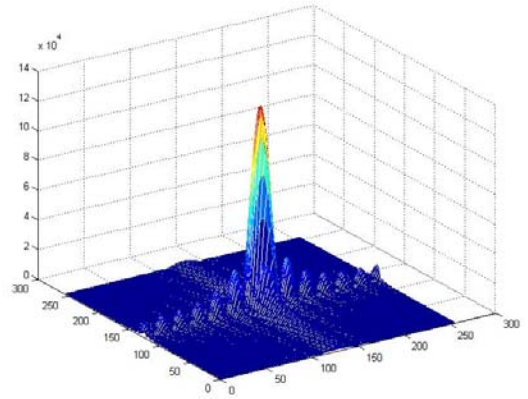
Primer - pravougaonik

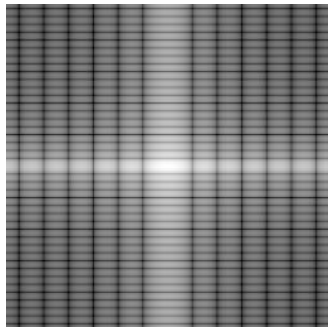
```
f = imread('rect.png');
figure; imshow(f);

F = fftshift(fft2(f));
Fmag = abs(F);
Fang = angle(F);

figure; imshow(Fmag,[]);
figure; imshow(Fang,[]);

figure; mesh(Fmag);
figure; imshow(log(1+Fmag),[]);
```





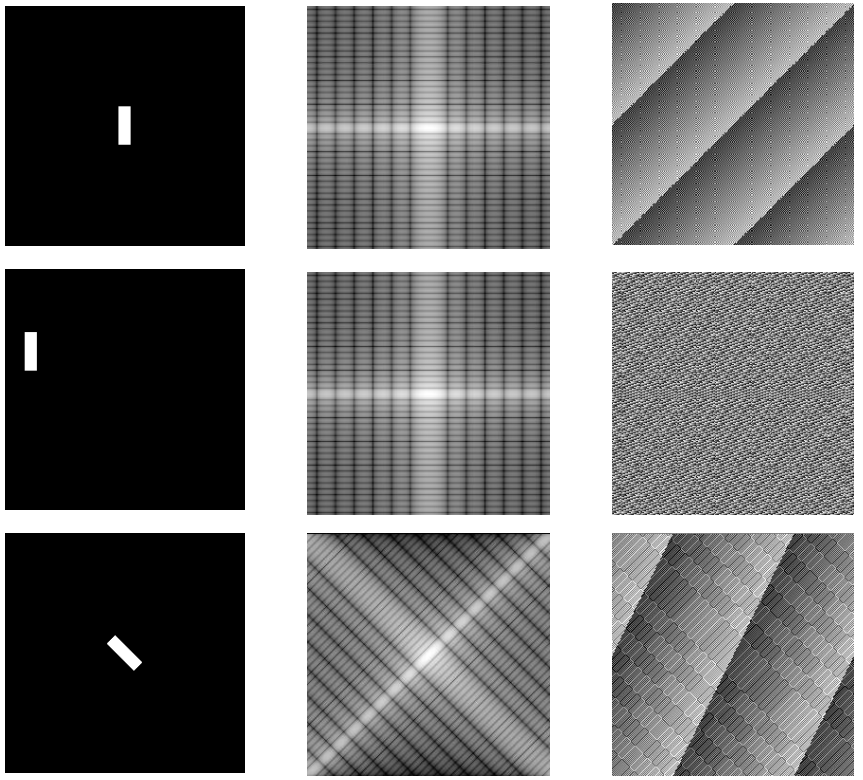






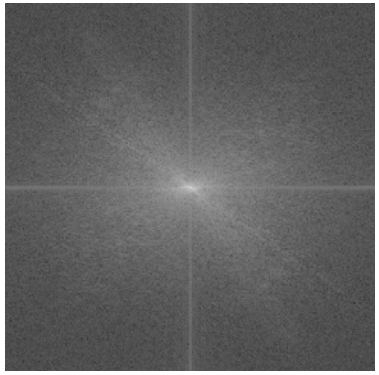
Uticaj translacije i rotacije na spektar

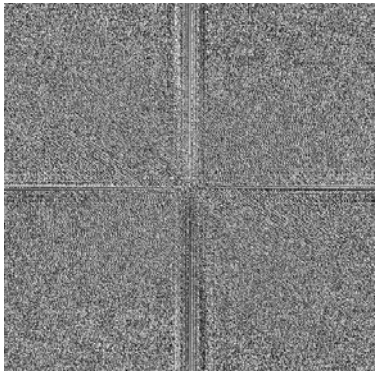
```
f = imread('rect_translated.png');
F = fftshift(fft2(f));
figure; imshow(log(1+abs(F)),[]);
figure; imshow(angle(F),[]);

f = imread('rect_rotated.png');
F = fftshift(fft2(f));
figure; imshow(log(1+abs(F)),[]);
figure; imshow(angle(F),[]);
```

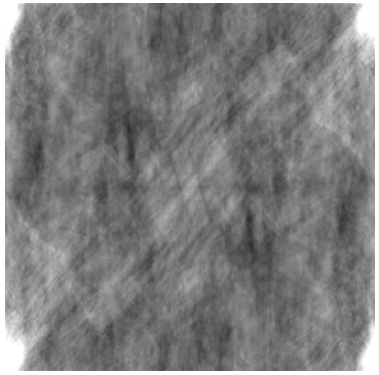


Značaj faze u slikama

```
f = im2double(imread('lena.tif'));
F = fftshift(fft2(f));

Fmag = abs(F);
Fang = angle(F);

figure; imshow(log(1+Fmag),[]);
figure; imshow(Fang,[]);

figure;
imshow(real(ifft2(ifftshift(Fmag.*exp(i*Fang)))));

figure; imshow(real(ifft2(ifftshift(Fmag))));

figure; imshow(ifft2(ifftshift(exp(i*Fang))),[]);
```





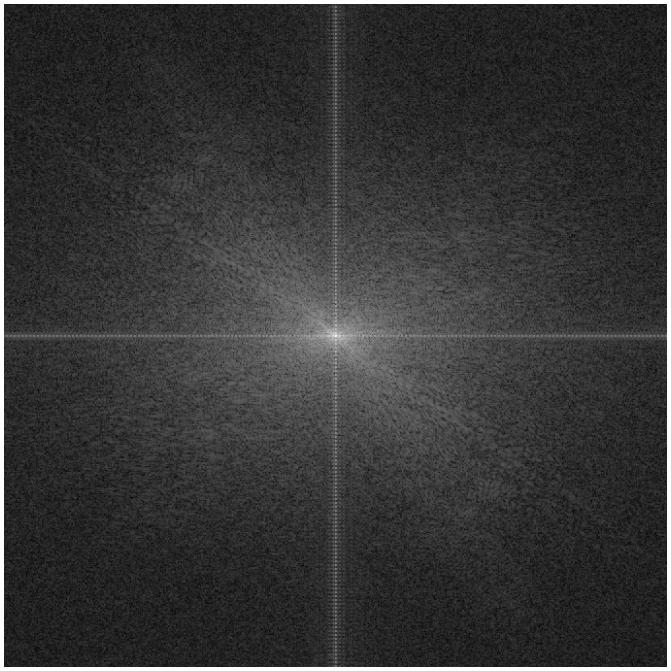


Spektar slike proširene nulama

```
fp = zeros(2*size(f));
fp(1:size(f,1), 1:size(f,2)) = f;
figure; imshow(fp);

Fp = fftshift(fft2(fp));
figure; imshow(log(1+abs(Fp)), []);

diff = abs(Fp(1:2:end, 1:2:end) - F);
max(diff(:))
```

$$\Delta T_{xp} = \Delta T_x \qquad \Delta T_{yp} = \Delta T_y$$

$$\Delta \mu_p = \frac{\Delta \mu}{2} \qquad \Delta \nu_p = \frac{\Delta \nu}{2}$$

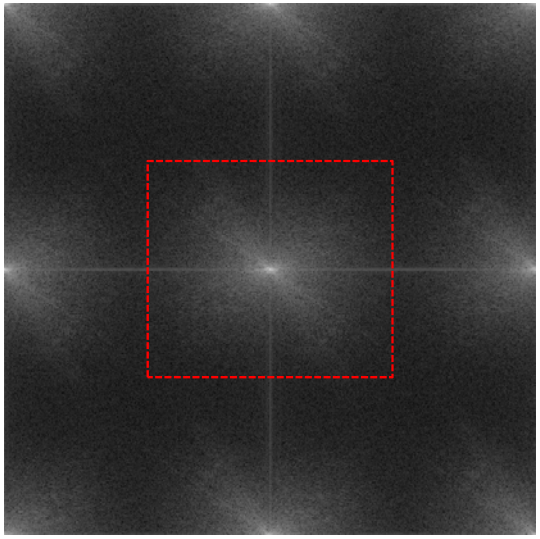


Spektar slike sa umetnutim nulama

```
fu = zeros(2*size(f));
fu(1:2:end, 1:2:end) = f;
figure; imshow(fu);

Fu = fft2(fu);
figure; imshow(log(1+abs(fftshift(Fu))), []);

diff = abs(fftshift(Fu(1:size(F,1), 1:size(F,2))) - F);
max(diff(:))
```

$$\Delta T_{xu} = 2\Delta T_x \qquad \Delta T_{yu} = 2\Delta T_y$$

$$\Delta \mu_u = \Delta \mu \qquad \Delta \nu_u = \Delta \nu$$



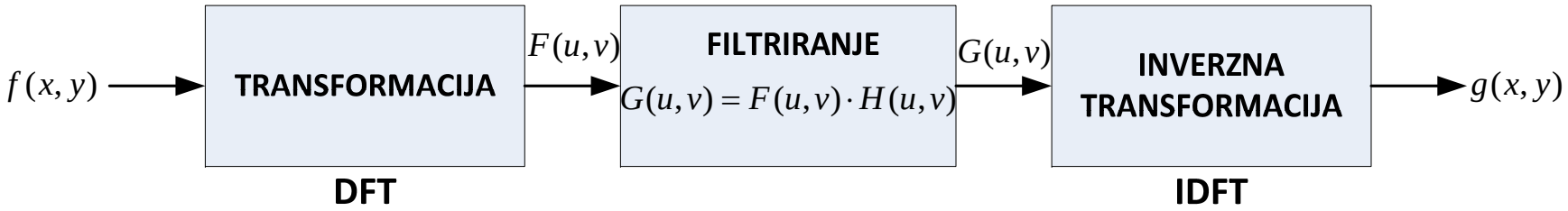
Filtriranje slike u frekvencijskom domenu

Filtriranje u frekvencijskom domenu

KONVOLUCIONA TEOREMA

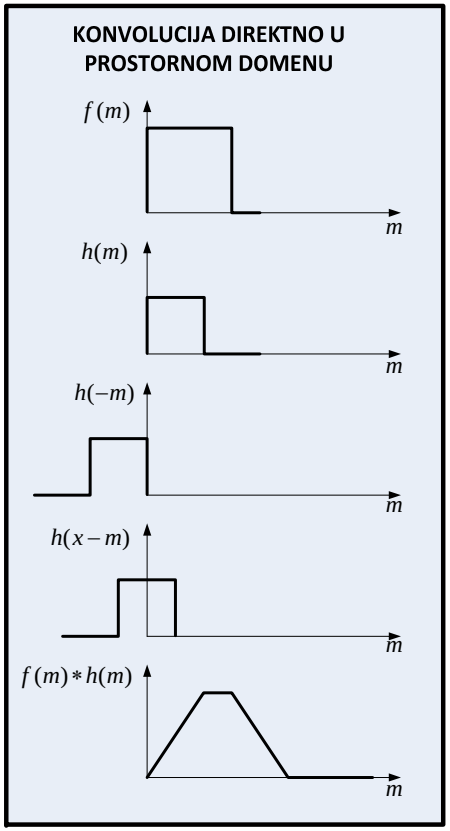
$$f(x,y) * h(x,y) \leftrightarrow H(u,v) \cdot F(u,v)$$
$$f(x,y) \cdot h(x,y) \leftrightarrow H(u,v) * F(u,v)$$

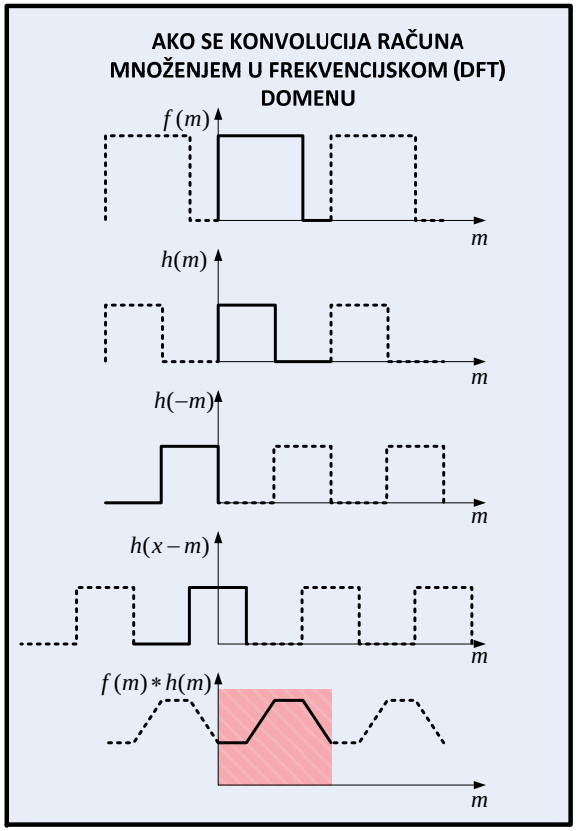
Kako je DFT diskretne funkcije periodična funkcija i kako je inverzna IDFT takođe periodična funkcija to ekvivalentna konvolucija koja odgovara množenju u frekvencijskom domenu odgovara cirkularnoj konvoluciji!!!

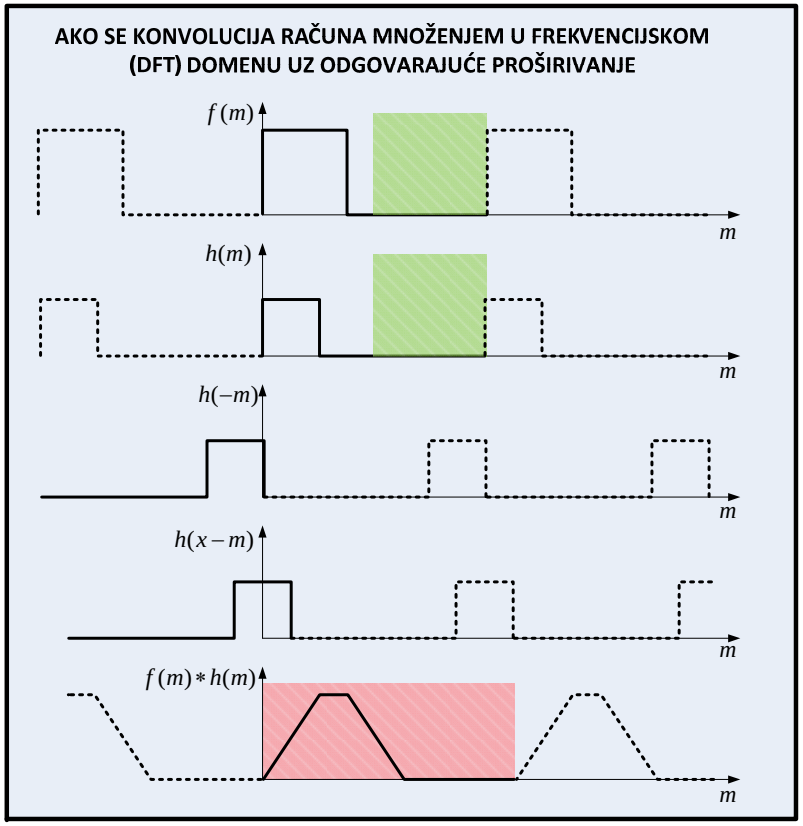


Funkcija filtra se obično specificira direktno u frekvencijskom domenu. F i H moraju imati isti broj odbiraka kako bi se obavilo skalarno množenje. Funkcija H se obično definiše tako da ne utiče na faznu karakteristiku.

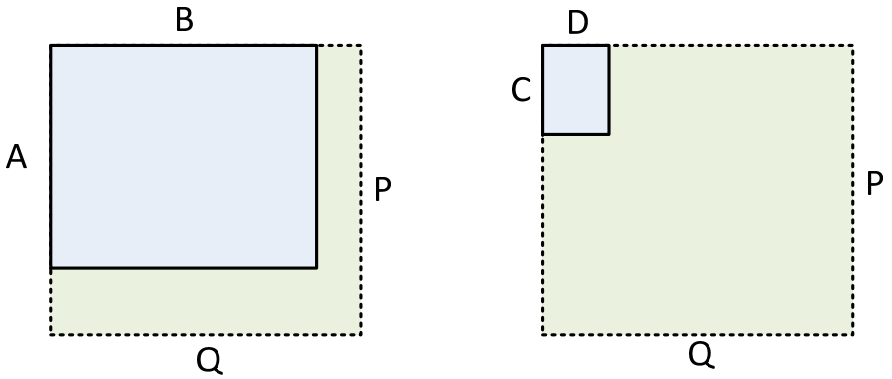
Značaj proširivanja sekvence







Proširivanje slike



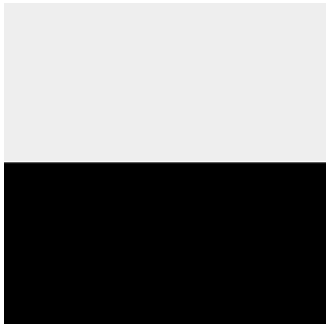
$$P \geq A + C - 1$$
$$Q \geq B + D - 1$$

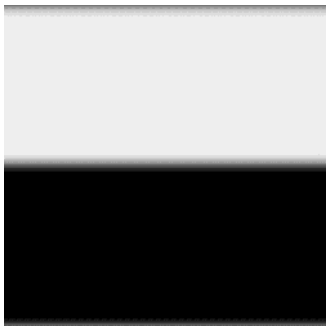
Proširivanje slike neophodno ako se konvolucija slike i prostorne maske obavlja množenjem u frekvencijskom domenu.

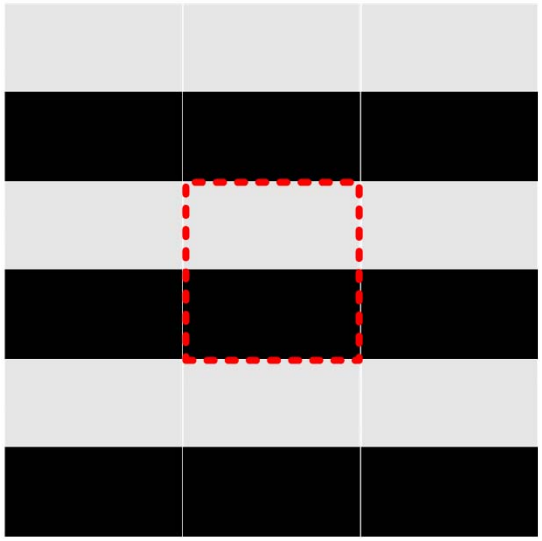
Proširivanje se obavlja dodavanjem nula.

Filtriranje bez proširivanja

```
f = double(imread('square.tif'));

[M, N] = size(f);

F = fft2(f);
H = lpfilter('gaussian', M, N, 10);

G = H.*F;

g = ifft2(G);

figure; imshow(uint8(f));
figure; imshow(uint8(g));
```

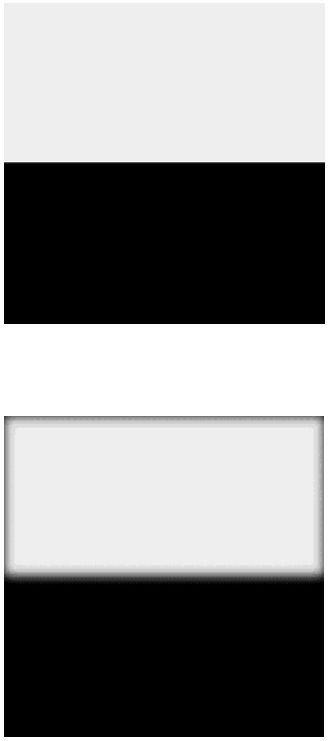


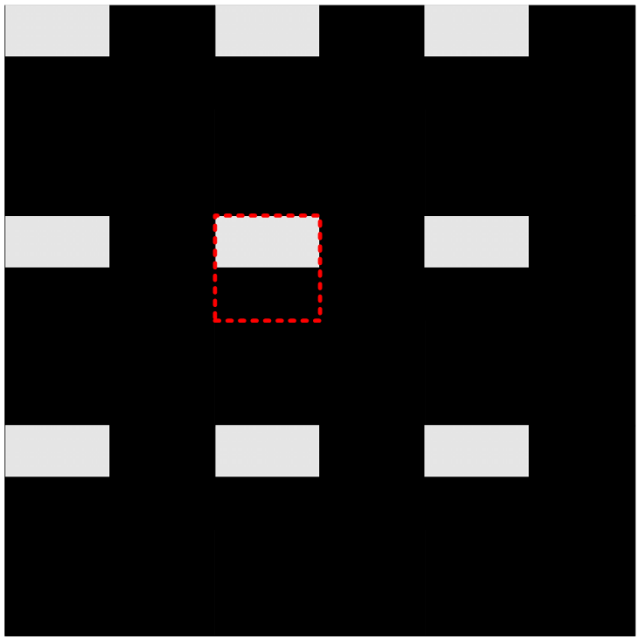




Filtriranje sa proširivanjem nulama

```
f = double(imread('square.tif'));

[M, N] = size(f);

P = 2*M-1; Q = 2*N-1;

F = fft2(f, P, Q);
H = lpfilter('gaussian', P, Q, 20);
G = H.*F;

g = ifft2(G);
g = g(1:M-1, 1:N-1);

figure; imshow(uint8(f));
figure; imshow(uint8(g));
```





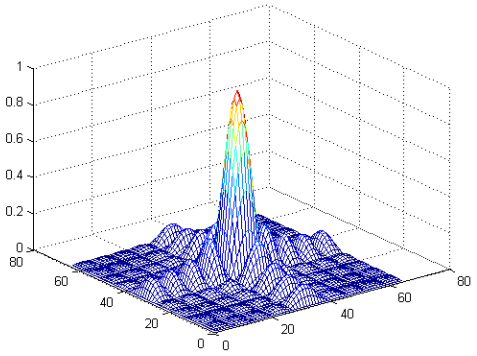
Osnovni koraci pri filtriranju

1) KONVERTOVATI SLIKU U ***FLOAT*** ILI ***DOUBLE***
2) PROŠIRITI SLIKU NULAMA
3) IZRAČUNATI FFT PROŠIRENE SLIKE
4) SPECIFICIRATI FILTARSKU FUNKCIJU U FREKVENCIJSKOM DOMENU
5) SKALARNO POMNOŽITI FFT PROŠIRENE SLIKE SA FILTARSKOM FUNKCIJOM
6) IZRAČUNATI IFFT REZULTATA
7) ISEĆI ODGOVARAJUĆI DEO IZLAZNE SLIKE
8) PO POTREBI KONVERTOVATI REZULTAT U IZVORNI FORMAT

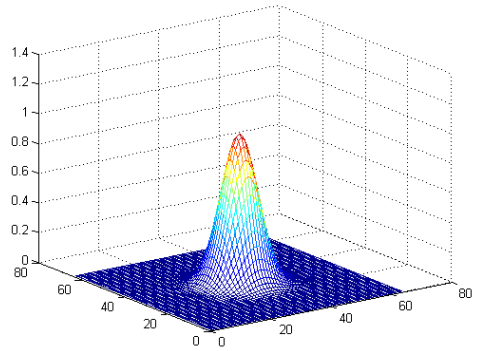
Vizualizacija prostornih filtara u frekvencijskom domenu

```
h_avg = fspecial('average', 10);
h_gauss = fspecial('gaussian', 20, 2);
h_lap = [1 1 1; 1 -8 1; 1 1 1];
h_sharp = [-1 -1 -1; ...
           -1 17 -1; ...
           -1 -1 -1]./9;

H_avg = freqz2(h_avg, 64, 64);
H_gauss = freqz2(h_gauss, 64, 64);
H_lap = freqz2(h_lap, 64, 64);
H_sharp = freqz2(h_sharp, 64, 64);

figure; mesh(abs(H_avg));
figure; mesh(abs(H_gauss));
figure; mesh(abs(H_lap));
figure; mesh(abs(H_sharp));
```

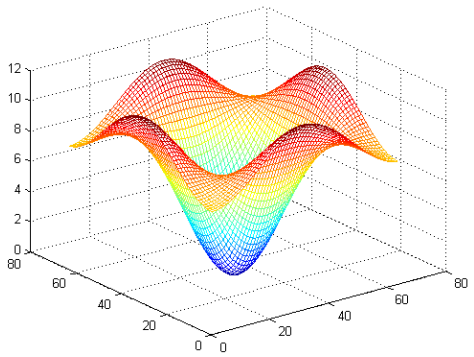
USREDNJAVANJE



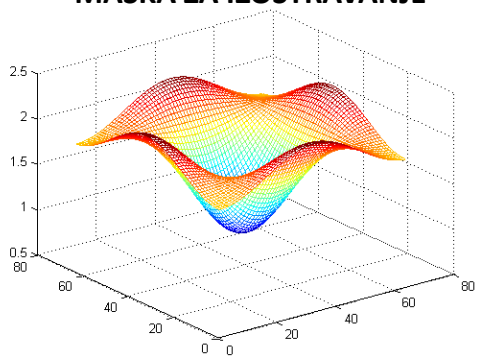
GAUSOV FILTAR



LAPLASIJAN



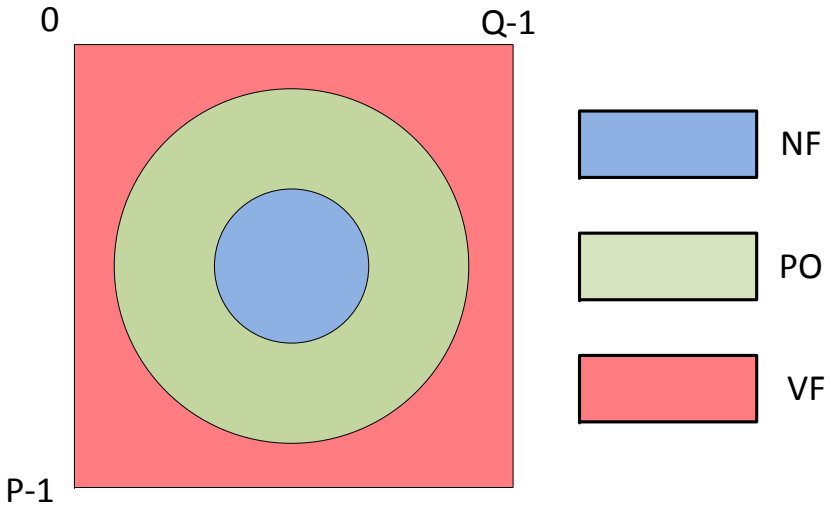
MASKA ZA IZOŠTRAVANJE



Specifikacija filtra u frekvencijskom domenu

$$D(u,v) = \sqrt{\left(u - P/2\right)^2 + \left(v - Q/2\right)^2}$$

SPECIFIKACIJA FILTRA U FREKVENCIJSKOM DOMENU



Niskofrekventno filtriranje

IDEALNI NF FILTAR

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

BATERVORTOV NF FILTAR

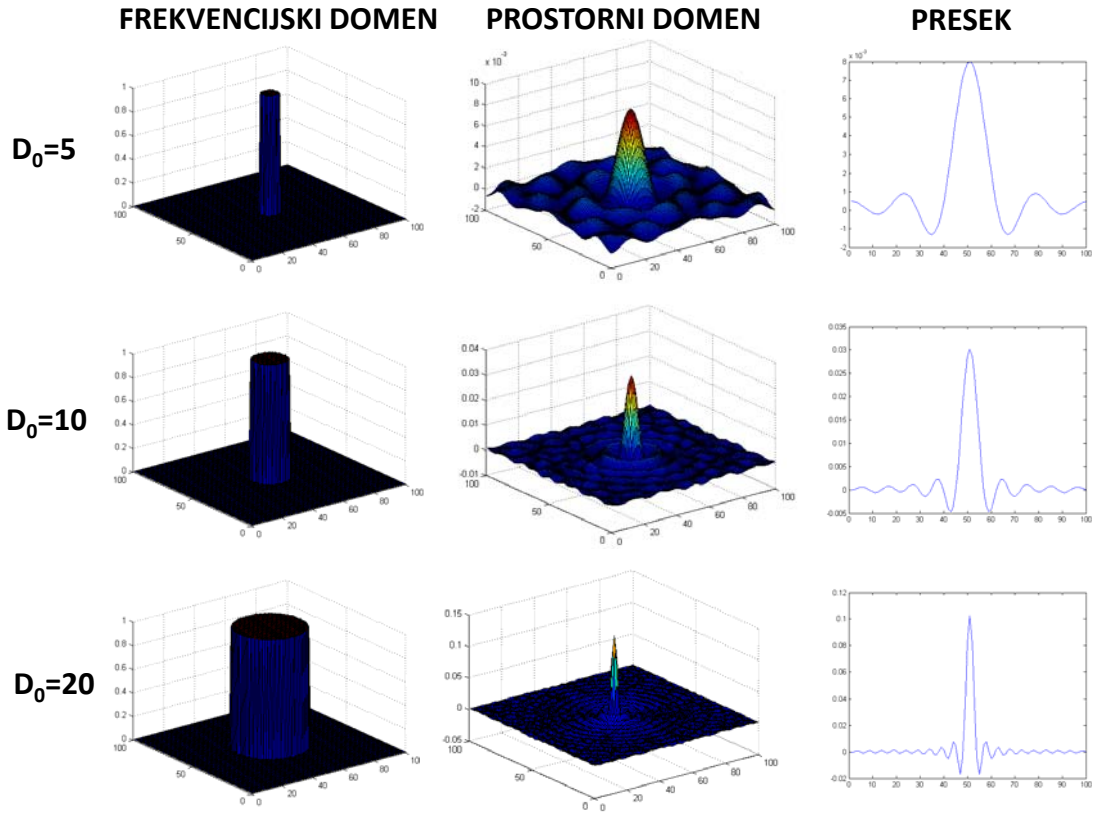
$$H(u,v)=\frac{1}{1+\left(\frac{D(u,v)}{D_0}\right)^{2n}}$$

GAUSOV NF FILTAR

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

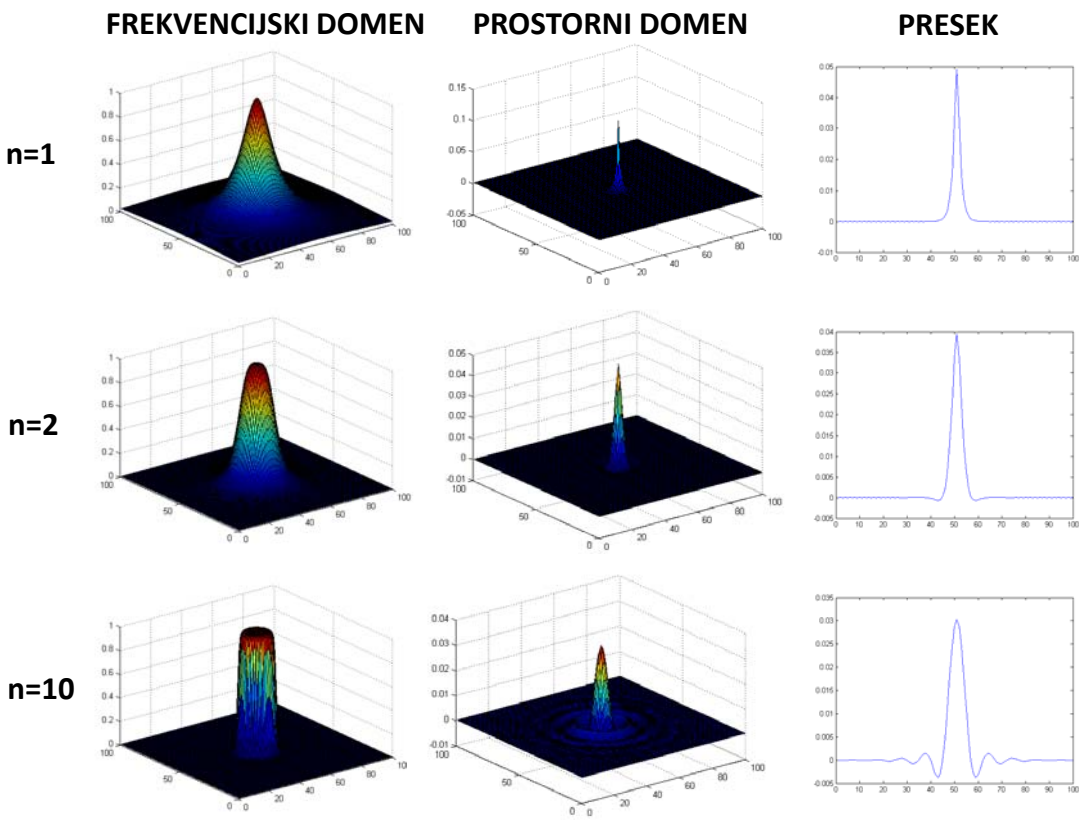
Idealni niskofrekventni filtar

```
H = lpfilter('ideal', 100, 100, 5);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('ideal', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('ideal', 100, 100, 20);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));
```



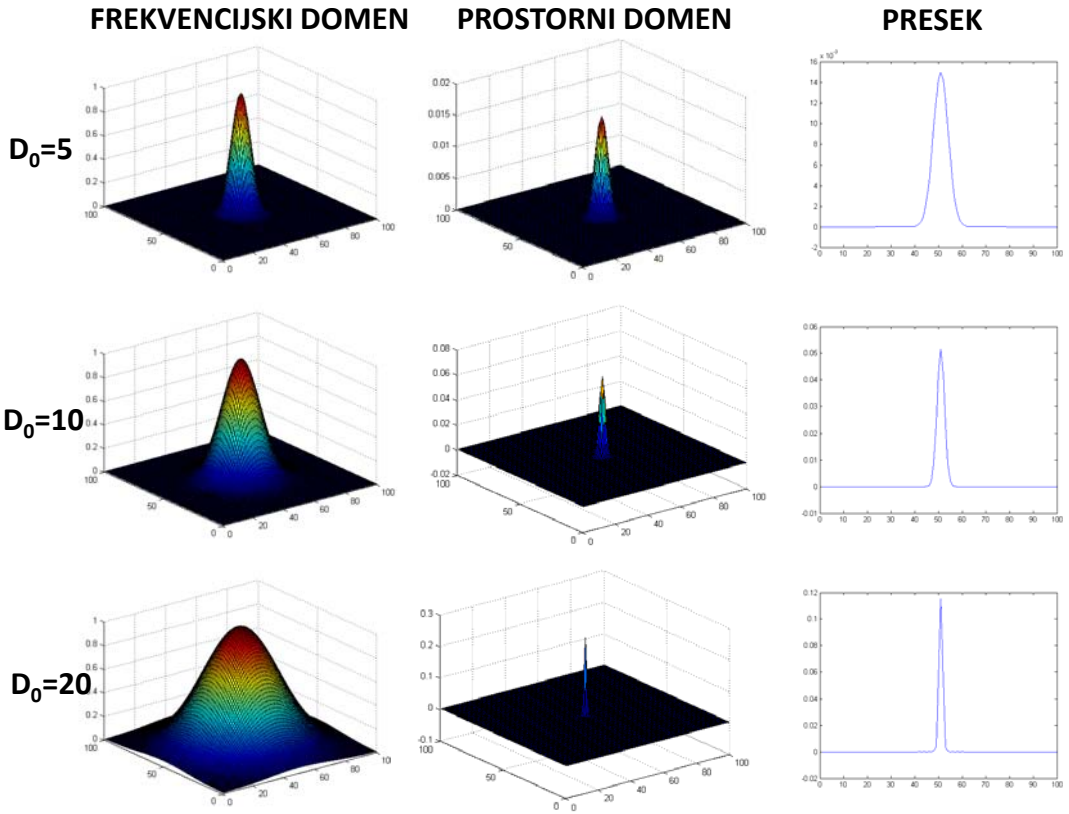
Batervortov niskofrekventni filtar

```
H = lpfilter('btw', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('btw', 100, 100, 10, 2);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('btw', 100, 100, 10, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));
```



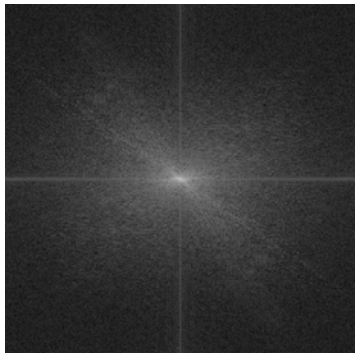
Gausov niskofrekventni filtar

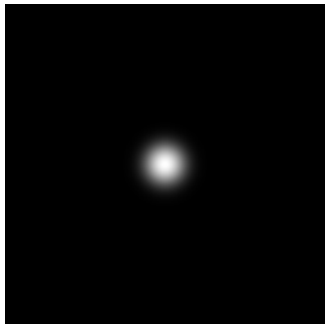
```
H = lpfilter('gaussian', 100, 100, 5);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('gaussian', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));

H = lpfilter('gaussian', 100, 100, 20);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:));
```

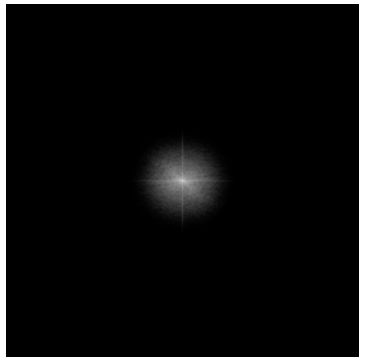


Niskofrekventni filtar - primer

```
f = im2double(imread('lena.tif'));
[M, N] = size(f);
P = 2*M-1; Q = 2*N-1;

Fp = fftshift(fft2(f, P, Q));

H = lpfilter('gaussian', P, Q, 40);
H = fftshift(H);
figure; imshow(log(1+abs(H)), []);

Gp = Fp.*H;
figure; imshow(log(1+abs(Gp)), []);

gp = ifft2(ifftshift(Gp));
g = gp(1:M, 1:N);
figure; imshow(g);
```







ISPROBATI RAZLIČITE PARAMETRE I TIPOVE FILTARA!

Visokofrekventno filtriranje

$$H_{VF}(u,v) = 1 - H_{NF}(u,v)$$

IDEALNI NF FILTAR

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

BATERVORTOV NF FILTAR

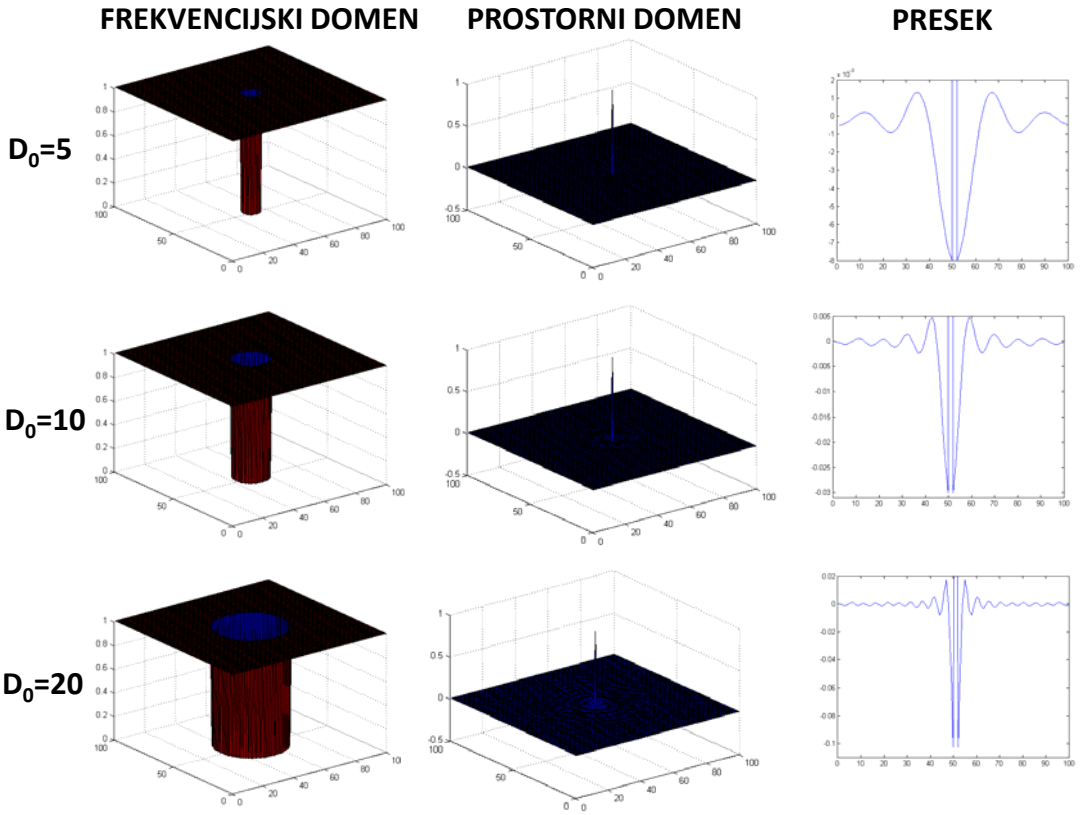
$$H(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)} \right)^{2n}}$$

GAUSOV NF FILTAR

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$

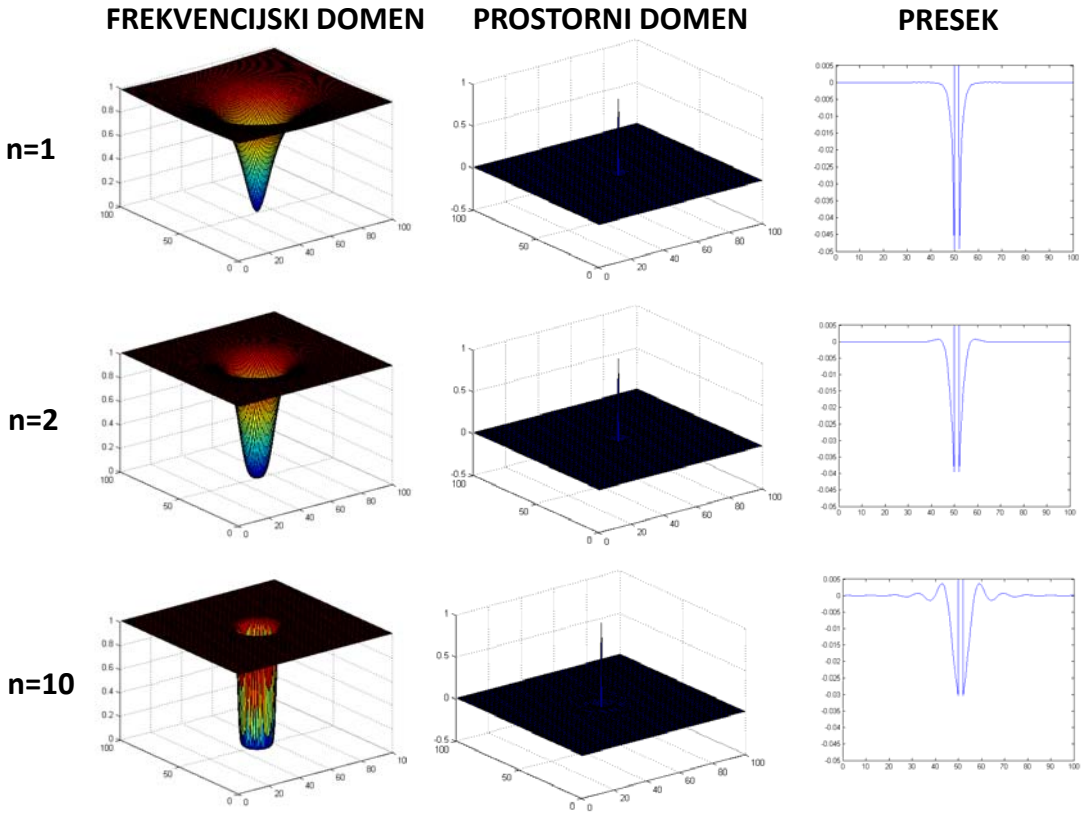
Idealni visokofrekventni filtar

```
H = hpfilter('ideal', 100, 100, 5);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-8e-3, 2e-3]);

H = hpfilter('ideal', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-31e-3, 5e-3]);

H = hpfilter('ideal', 100, 100, 20);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-11e-2, 2e-2]);
```



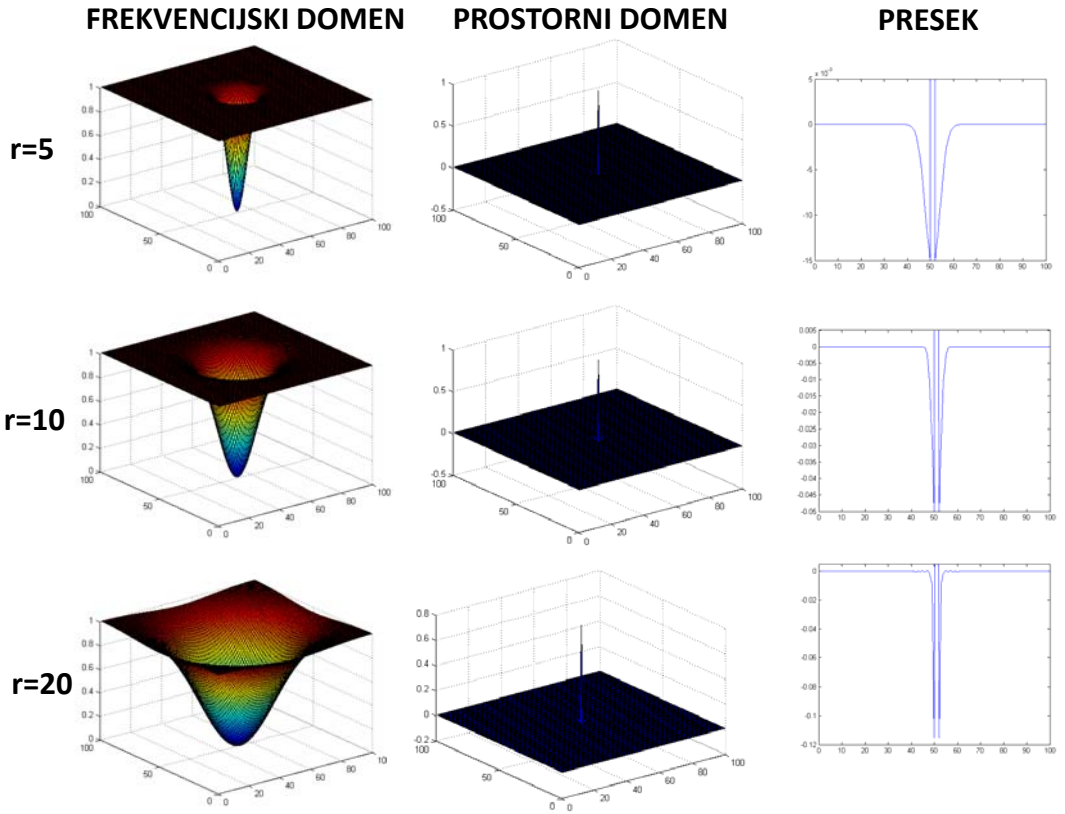
Batervortov visokofrekventni filtar

```
H = hpfilter('btw', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-5e-2, 5e-3])

H = hpfilter('btw', 100, 100, 10, 2);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-5e-2, 5e-3])

H = hpfilter('btw', 100, 100, 10, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-5e-2, 5e-3])
```



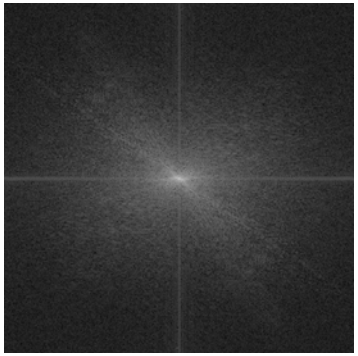
Gausov visokofrekventni filtar

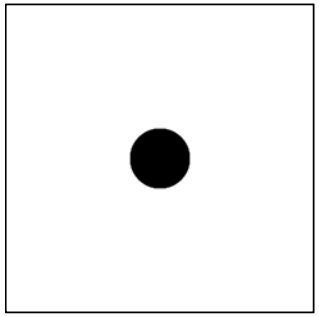
```
H = hpfilter('gaussian', 100, 100, 5);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-15e-3, 5e-3])

H = hpfilter('gaussian', 100, 100, 10);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-5e-2, 5e-3])

H = hpfilter('gaussian', 100, 100, 20);
figure; surf(fftshift(H));
h = ifftshift(ifft2(H));
figure; surf(h);
figure; plot(h(51,:)); ylim([-12e-2, 5e-3])
```

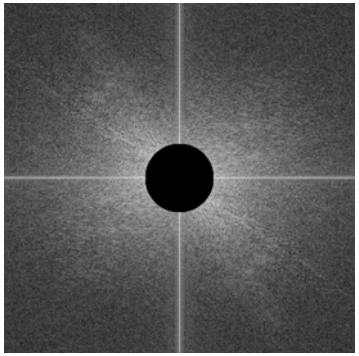


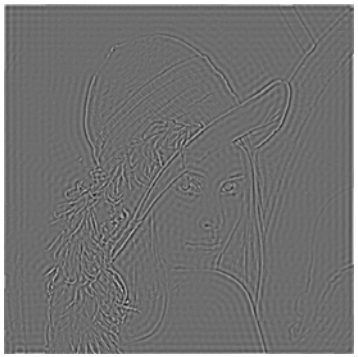
Visokofrekventni filtar - primer

```
f = im2double(imread('lena.tif'));
[M, N] = size(f);
P = 2*M-1; Q = 2*N-1;

Fp = fftshift(fft2(f, P, Q));

H = hpfilter('ideal', P, Q, 100);
H = fftshift(H);
figure; imshow(log(1+abs(H)), []);

Gp = Fp.*H;
figure; imshow(log(1+abs(Gp)), []);

gp = ifft2(ifftshift(Gp));
g = gp(1:M, 1:N);
figure; imshow(g, []);
```









ISPROBATI RAZLIČITE PARAMETRE I TIPOVE FILTARA!

Izoštravanje isticanjem visokih učestanosti

```
f = im2double(imread('lena.tif'));
[M, N] = size(f);
P = 2*M-1; Q = 2*N-1;

Fp = fftshift(fft2(f, P, Q));

H = 1 + 2*hpfilter('gaussian', P, Q, 100);
H = fftshift(H);
figure; imshow(log(1+abs(H)), []);

Gp = Fp.*H;
figure; imshow(log(1+abs(Gp)), []);

gp = ifft2(ifftshift(Gp));
g = gp(1:M, 1:N);
figure; imshow(g);
```

Selektivno filtriranje

PROPUSNIK OPSEGA

$$H_{PO}(u,v) = H_{NF1}(u,v) - H_{NF2}(u,v)$$

NEPROPUSNIK OPSEGA

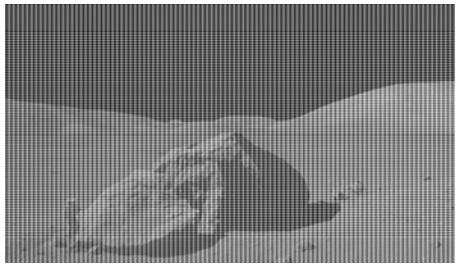
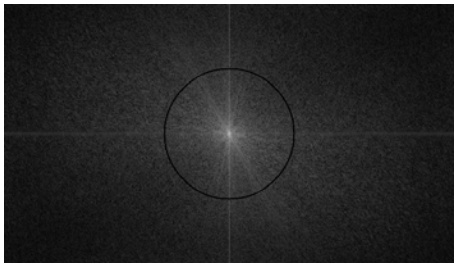
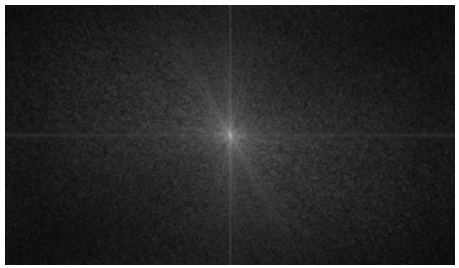
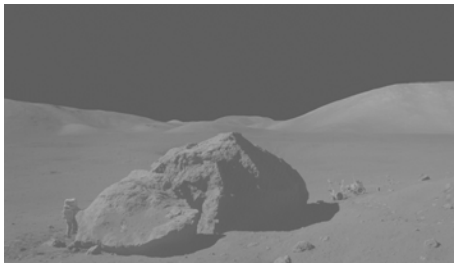
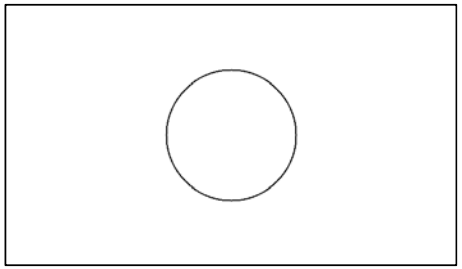
$$H_{NPO}(u,v) = 1 - H_{PO}(u,v)$$

NOČ FILTAR

$$H_{NR}(u,v) = \prod_{i=1}^{Q} H_{VFi}(u,v) \cdot H_{VF-i}(u,v)$$

Filtriranje periodičnog šuma

```
f = im2double(imread('appolo17.tif'));
[M, N] = size(f);
figure; imshow(f);

F = fftshift(fft2(f));
figure; imshow(log(1+abs(F)), []);

Hlp1 = lpfilter('ideal', M, N, 101);
Hlp2 = lpfilter('ideal', M, N, 99);
H = fftshift(1 - Hlp1 + Hlp2);
figure; imshow(log(1+abs(H)), []);
G = F.*H;
figure; imshow(log(1+abs(G)), []);
g = ifft2(ifftshift(G));
figure; imshow(g);
g1 = imadjust(g, stretchlim(g), [0, 1]);
figure; imshow(g1);
```


Selektivno filtriranje

Nije deo MTALAB IP toolbox-a

‘reject’, ‘pass’

Selektivnost filtara

```
H = cnotch(type, notch, M, N, C, r, n)
```

Red Batervortovog filtra

Tip filtra koji se koristi: *‘ideal’, ‘btw’, ‘gaussian’*

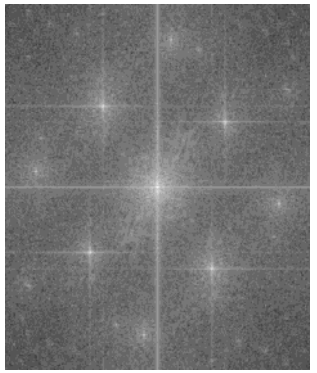
Koordinate centara svakog filtra, uzete sa centriranog spektra.

Prilikom selektivnog filtriranja uglavnom se ne radi proširivanje slike!!!

Selektivno filtriranje

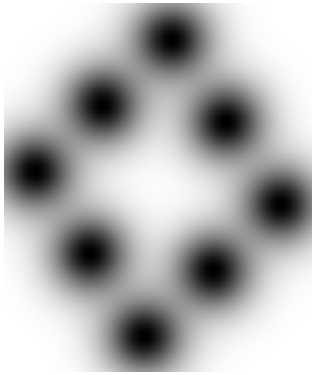
```
f = imread('tiger_ht.jpg');
f = double(rgb2gray(f));
figure; imshow(uint8(f));
[M, N] = size(f);
F = fft2(f);
figure; imshow(log(1 + abs(fftshift(F))),[]);

C = [113 21; 69 67; 25 112; 80 148];
H = cnotch('gaussian', 'reject', M, N, C, 20);
figure; imshow(log(1 + abs(fftshift(H))),[]);

G = H.*F;
figure; imshow(log(1 + abs(fftshift(G))),[]);
g = ifft2(G);
figure; imshow(uint8(g));
```

f

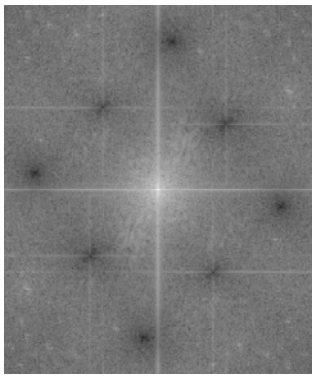


F

H



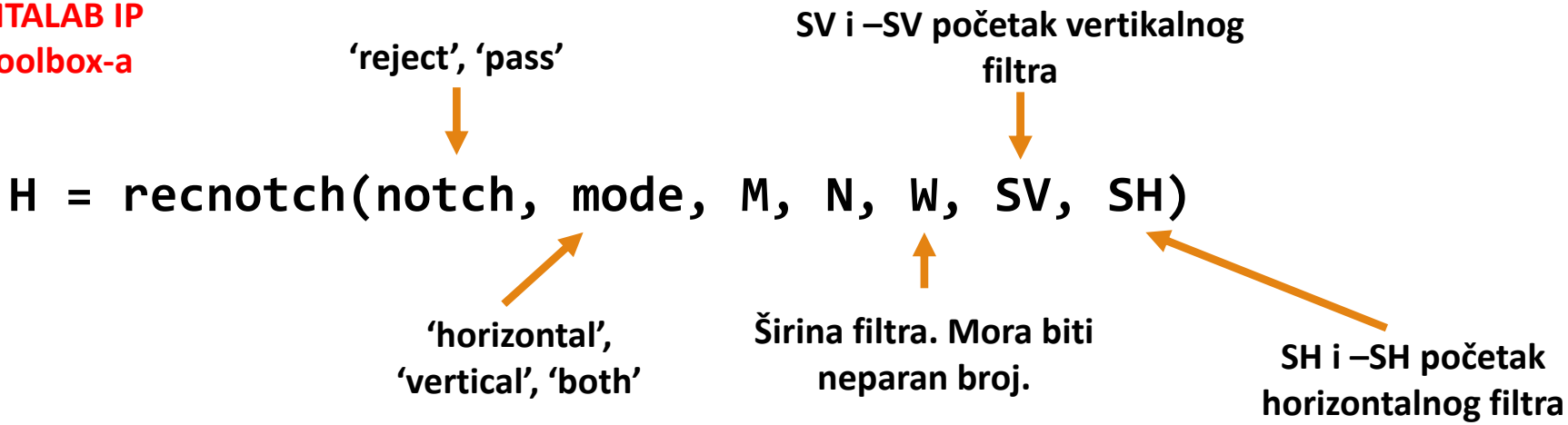
G



g

Selektivno filtriranje

Nije deo MTALAB IP toolbox-a

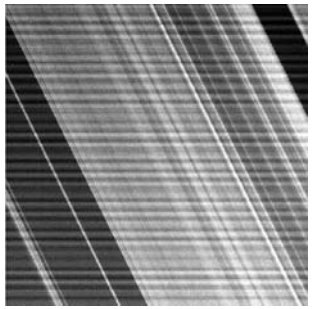


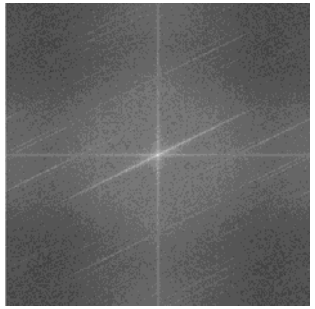
Selektivno filtriranje

```
f = double(imread('cassini.tif'));
figure; imshow(uint8(f));
[M, N] = size(f);
F = fft2(f, M, N);
figure; imshow(log(1 + abs(fftshift(F))),[]);

H = recnotch('reject', 'vertical', M, N, 3, 15, 15);
figure; imshow(log(1 + abs(fftshift(H))),[]);

G = H.*F;
figure; imshow(log(1 + abs(fftshift(G))),[]);
g = ifft2(G);
figure; imshow(uint8(g));
```

Nije urađeno proširenje slike, koristi se implicitna periodičnost

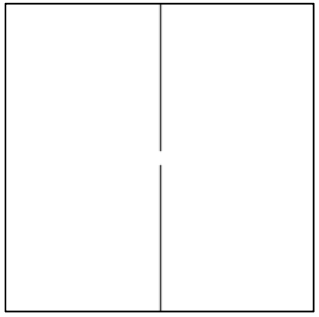
f



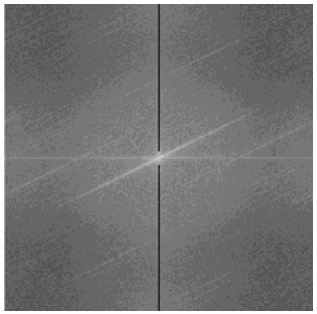


F

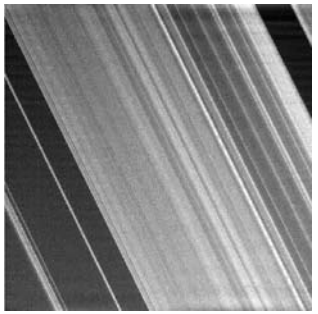
H



G

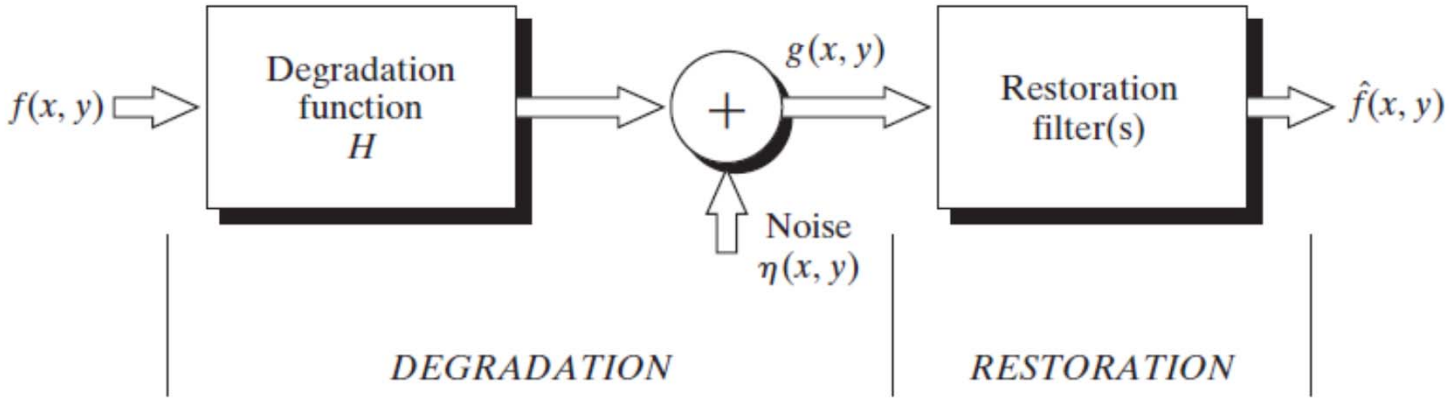


g



Restauracija slike

Model degradacije slike



$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Dodavanje šuma

imnoise

gaussian

salt & pepper

localvar

```
f = im2double(imread('lena.tif'));
figure; imshow(f);

gn = imnoise(f, 'gaussian', 0, 0.001);
figure; imshow(gn);

gn = imnoise(f, 'gaussian', 0, 0.01);
figure; imshow(gn);

gn = imnoise(f, 'salt & pepper', 0.2);
figure; imshow(gn);

gn = imnoise(f, 'localvar', f/20);
figure; imshow(gn);

gn = imnoise(f, 'localvar', [0:255]/255,
(1/20)*[0:255]/255);
figure; imshow(gn);
```

gausov šum

impulsni šum

varijansa zavisna od intenziteta piksela

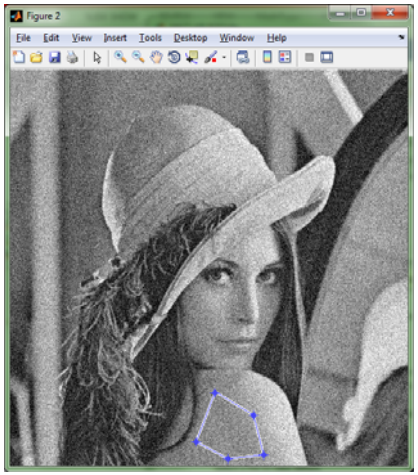
Dodavanje degradacije

```
f = im2double(imread('road.tif'));
figure; imshow(f);
[M, N] = size(f);

F = fftshift(fft2(F));

H_blur = fftshift(lpfilter('gaussian', M, N, 40));
h_motion = fspecial('motion', 20, 135);
H_motion = fftshift(fft2(h_motion, M, N));

G_blur = H_blur.*F;
g_blur = ifft2(ifftshift(G_blur));
G_motion = H_motion.*F;
g_motion = ifft2(ifftshift(G_motion));

figure; imshow(g_blur);
figure; imshow(g_motion);
```

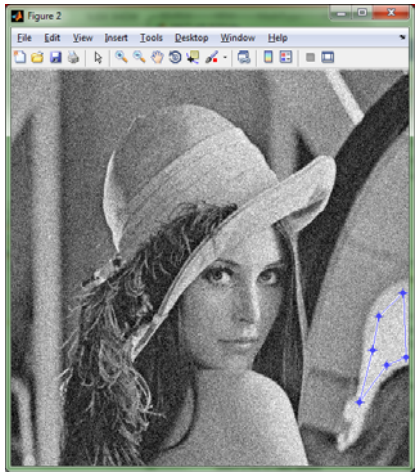
Estimacija parametara šuma

```
f = im2double(imread('lena.tif'));
gn = imnoise(f, 'gaussian', 0, 0.01);
figure; imshow(gn);

R = roipoly(gn);
var(gn(R))
```

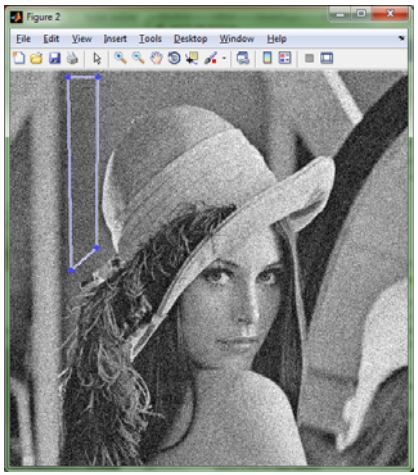
Ideja: Odabrati što uniformniji deo slike i odrediti varijansu. Na osnovu oblika histograma se može odrediti raspodela šuma.



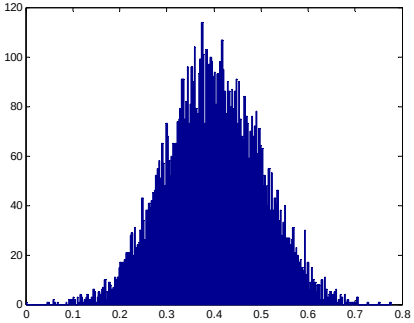
var = 0.0104



var = 0.0099



var = 0.0102



Adaptivno lokalno usrednjavanje

```
f = im2double(imread('lena.tif'));
figure; imshow(f);

nvar = 0.01;
g = imnoise(f, 'gaussian', 0, nvar);
figure; imshow(g);

havg = fspecial('average', [7,7]);
gavg = imfilter(g, havg, 'replicate');
gsqr_avg = imfilter(g.^2, havg, 'replicate');
gvar = gsqr_avg - gavg.^2;

w = nvar./gvar;
w(w>1)=1;
f_est = g + w.*(gavg - g);

figure; imshow(gavg); figure; imshow(w);
figure; imshow(f_est);
```

Adaptivni medijan – sa fiksnim prozorom

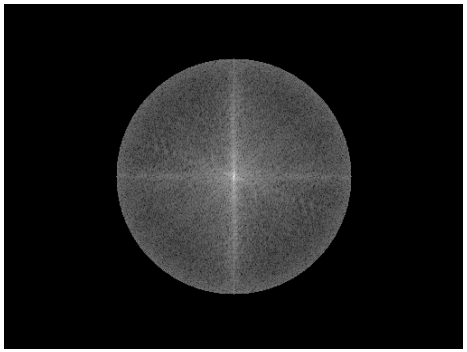
```
f = im2double(imread('lena.tif'));
g = imnoise(f, 'salt & pepper', 0.4);
figure; imshow(g);

gmed = medfilt2(g, [7,7], 'symmetric');
figure; imshow(gmed);

gmax = ordfilt2(g, 49, ones(7,7));
gmin = ordfilt2(g, 1, ones(7,7));

fest = g;

nind = (g == gmin) | (g == gmax);
fest(nind) = gmed(nind);
figure; imshow(fest);
```

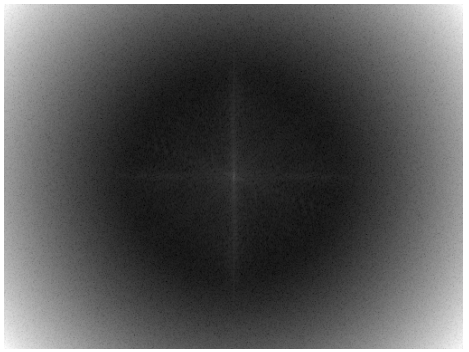


Ideja: Zameniti medijanom samo one piksele koji predstavljaju ekstremne vrednosti u okviru lokalnog susedstva. Ostale piksele ostaviti nepromenjenim. Na taj način se sprečava degradacija piksela koji nisu pogođeni šumom. Unapređenje je adaptivna promena veličine lokalnog susedstva tako da se uvek koristi najmanji mogući prozor oko centralnog piksela.

Inverzno filtriranje uz ograničenje opsega

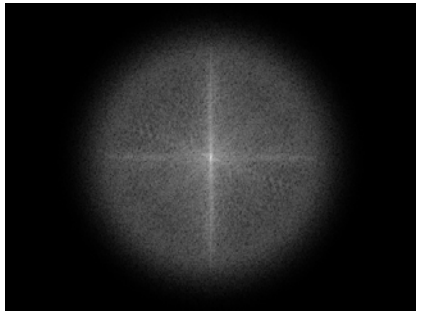
```
g = im2double(imread('road_blur.tif'));
figure; imshow(g);
[M, N] = size(g);

G = fftshift(fft2(g));
load H_blur

Fest = G./H_blur;
figure; imshow(log(1+abs(Fest)), []);

Hlp = fftshift(lpfilter('ideal', M, N, 130));
figure; imshow(log(1+abs(Hlp)), []);

Fest = (G./H_blur).*Hlp;
figure; imshow(log(1+abs(Fest)), []);

fest = ifft2(ifftshift(Fest));
figure; imshow(fest);
```





Vinerov filtar

```
g = im2double(imread('road_blur.tif'));
figure; imshow(g);
[M, N] = size(g);

G = fftshift(fft2(g));
load H_blur

W = (abs(H_blur).^2)./(abs(H_blur).^2 + 1e-4);
Fest = (G./H_blur).*W;
figure; imshow(log(1+abs(Fest)), []);

fest = ifft2(ifftshift(Fest));
figure; imshow(fest);
```



Uklanjanje degradacije usled pokreta

```
g = im2double(imread('road_motion.tif'));
figure; imshow(g);
[M, N] = size(g);

G = fftshift(fft2(g));
load H_motion

W = (abs(H_motion).^2)./(abs(H_motion).^2 + 1e-4);
Fest = (G./H_motion).*W;
figure; imshow(log(1+abs(Fest)), []);

fest = ifft2(ifftshift(Fest));
figure; imshow(fest);
```