

DSZOB, cvičenie 9.

Zadanie:

Úloha 1 Odstránenie šumu z 2D signálu (obraz) v priestorovej doméne

Z obrázka, ktorý obsahuje šum typu “salt and pepper”, odstráňte tento šum. Použite pri tom konvolúciu s týmito filtračnými jadarmi:

- Medián
- Gauss
- Mean

Zodpovedzte nasledujúce otázky:

Ktorý z týchto filtrov je najvhodnejší na odstránenie šumu typu “salt and pepper” a prečo ?

Ktorý z týchto filtrov najviac zachováva hrany v obrázku, a ktorý najmenej? **Zdôvodnite !**

Postup vhodne dokumentuje (Code/Text bloky)!

Riešenie:

```
% Riesenie / Solution
img = imread('salt_and_pepper.png');
grayImg = rgb2gray(img);
figure
imshow(grayImg)
title("Original image")
```



```
% Z obrázka, ktorý obsahuje šum typu "salt and pepper", sme odstránili tento typ  
šumu (po ďalšej časti).  
% Použitím konvolúciu s týmito filtračnými jadarmi:  
imgSize = size(grayImg)
```

```
imgSize = 1x2  
391    583
```

```
medianImg = medfilt2(grayImg);  
gaussImg = conv2(grayImg, fspecial("gaussian", [7 7], 6));  
averageImg = conv2(grayImg, fspecial("average", [7 7]));  
figure  
imshow(medianImg, [])  
title("Median filtered image")
```

Median filtered image



```
figure  
imshow(gaussImg, [])  
title("Gauss filtered image")
```

Gauss filtered image



```
figure  
imshow(averageImg, [])  
title("Average filtered image")
```

Average filtered image



Ktorý z týchto filtrov je najvhodnejší na odstránenie šumu typu “salt and pepper” a prečo ?

1. Median, pretože po filtrovaní s ním bola odstranena značná časť šumu (typu salt and pepper).

Ktorý z týchto filtrov najviac zachováva hrany v obrázku, a ktorý najmenej? **Zdôvodnite !**

1. Najviditeľnejšie hrany sú pri median filtry, to je jednak z dôvodu jeho najväčšom rozšírení (nezáleží ako ostre vydíma hrany na ostatných obrazkov keď neviem do čoho sa zapájaju/do akého celku). V porovnaní s týmto filtrom gausov filter vyhľazoval hrany, respektíve mierne rozmažával obrazok, tým size čiastočne mizol "salt and pepper" šum, však toto malo plošný efekt stratenia i ostatných detailov, vrátanie hrán. Podobne ako gausov filter sa správal mean filter.

Úloha 2 Odstránenie šumu z 2D signálu (obraz) v spektre

Vizualujte spektrum obrazu s periodickým šumom. Navrhnite podľa pozície dominantných vrcholov v spektri filter, pomocou ktorého vo výsledku odstráňte periodický šum z obrazu. Filtráciu vykonajte pre všetky obrázky na dokumentovom serveri s periodickým šumom. Spekturm umožnite filtrovať interaktívne - používateľ bude mať možnosť kliknutím vybrať miesto v spektre, ktoré má byť vyfiltrované.

Filtrujte nasledovnými spôsobmi:

Umožnite zo spektra vymazať (vynulovať) okolie miesta, na ktoré bolo kliknuté (ľavé stlačenie myši). Využite symetriu, aby ste odstránili rovnaké miesto aj vo zvyšných kvadrantoch. Vymažte zo spektra kružnicu, ktorá bude mať stred v strede spektra a polomer určený vzdialenosťou od stredu k miestu, na ktoré bolo kliknuté (pravé stlačenie myši) ako na obrázku nižšie.

Kružnicu vytvorte tak, aby bola typu BandPass podobná Gaussovmu filtrovi. Vytvoríte ju napríklad ako ideálny BandPass filter a následnou konvolúciou s Gaussovym filtrom ju rozmažete.

Zodpovedzte nasledujúce otázky:

Prečo je vhodné používateľovi umožniť zo spektra vymazať celú kružnicu ? (pomôcka: symetria).

```
% Riesenie / Solution
grayImgPer1 = imread('house.jpg');
grayImgPer2 = imread('periodic_noise1.jpg');
grayImgPer3 = imread('periodic_noise2.jpg');
grayImgPer4 = imread('periodic_noise3.jpg');
figure
imshow(grayImgPer1)
title("Original image")
```



```
% Vizualizovali sme spektrum obrazu s periodickým šumom.  
shiftFFTIImg1 = fftshift(fft2(grayImgPer1));  
shiftFFTIImg2 = fftshift(fft2(grayImgPer2));  
shiftFFTIImg3 = fftshift(fft2(grayImgPer3));  
shiftFFTIImg4 = fftshift(fft2(grayImgPer4));
```

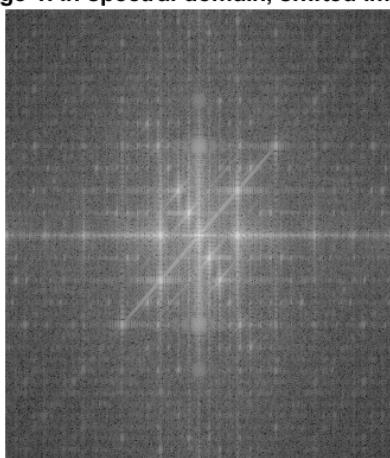
Poznamka: v nasledujucich častiach nasledujú obrazky, začinajuce s originalnym obrazkom so zeleným titulom. A dalšie páry obrazkov začinajuce s pozmeneným spektralnym obrazkom o filter a filtrovaným výslednym obrazkom.

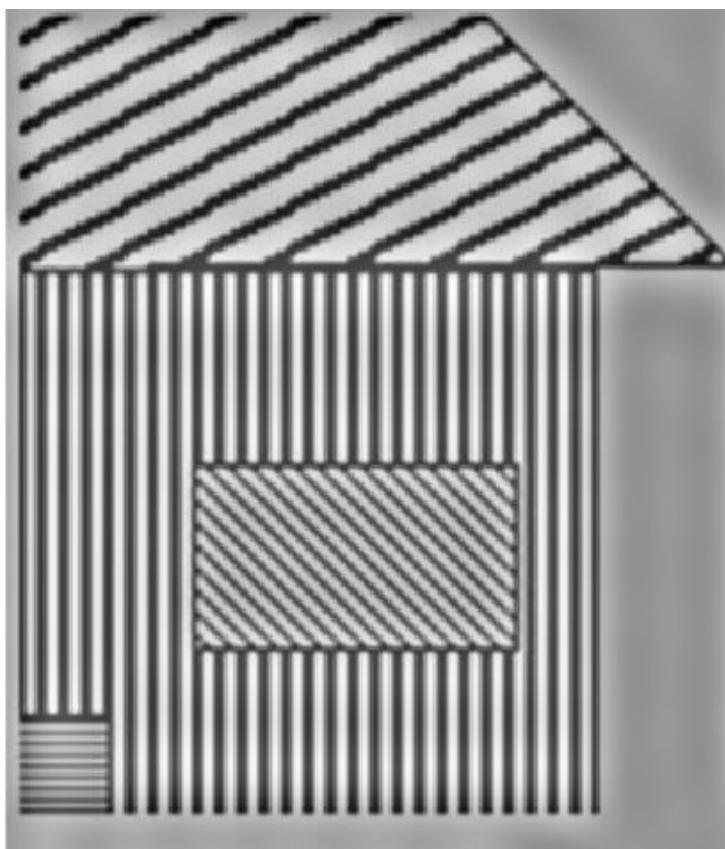
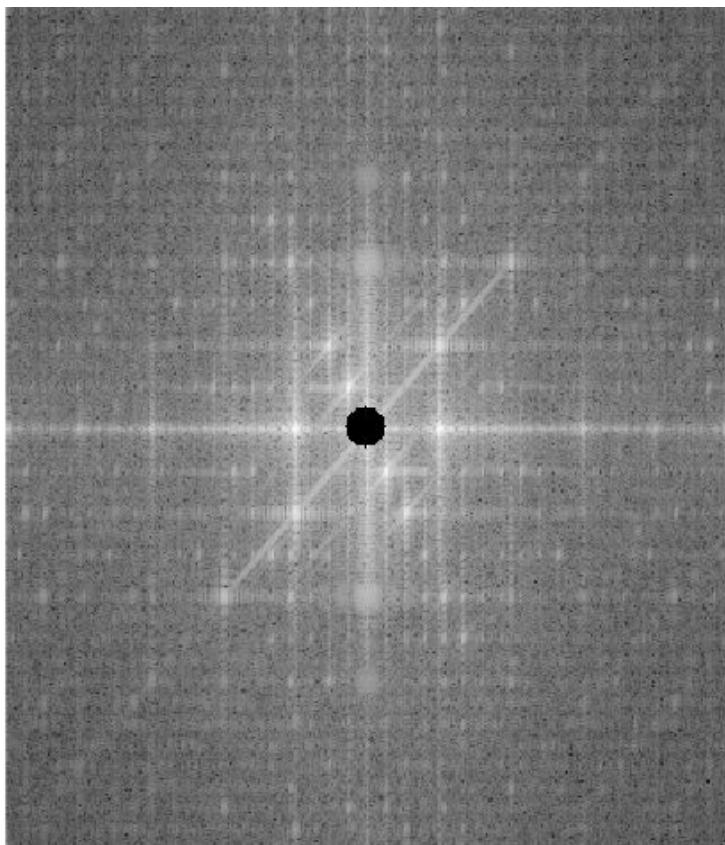
```
figure  
imshow(grayImgPer1, [])  
title("First origin image", Color="green")
```

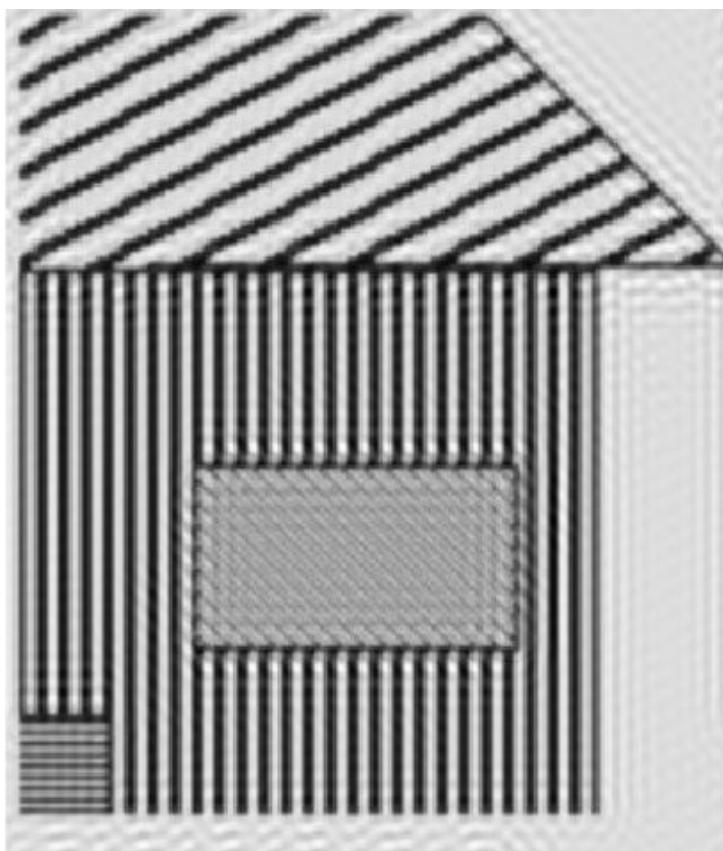
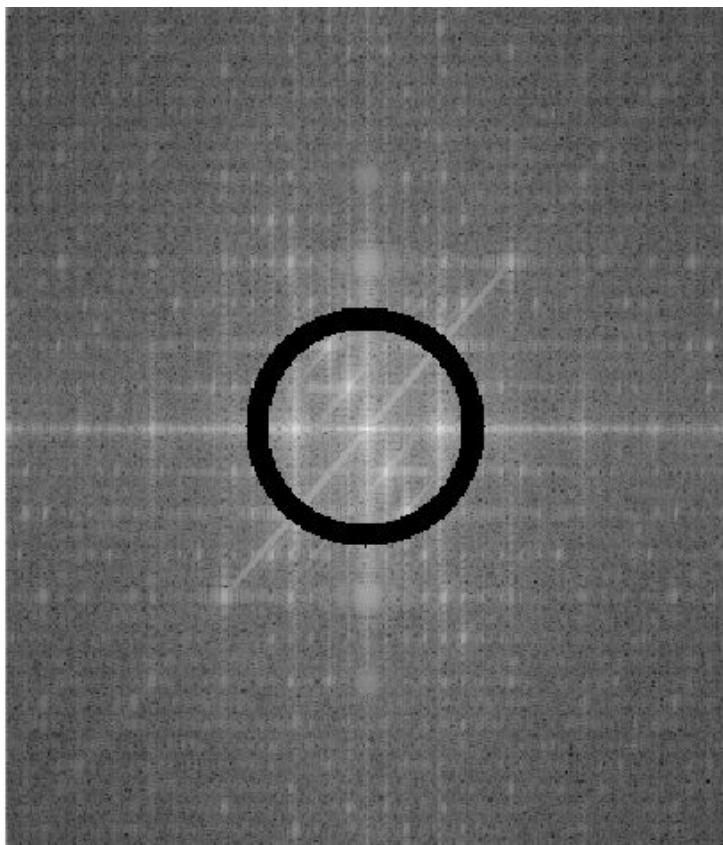


```
processSpectImage(shiftFFTIimg1, "Image 1. in spectral domain, shifted  
image.")
```

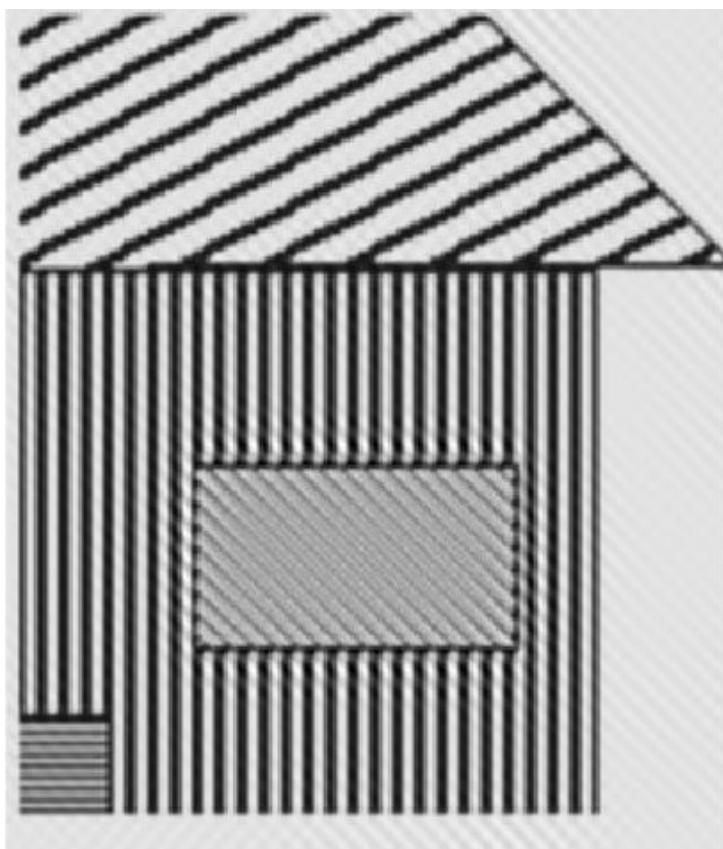
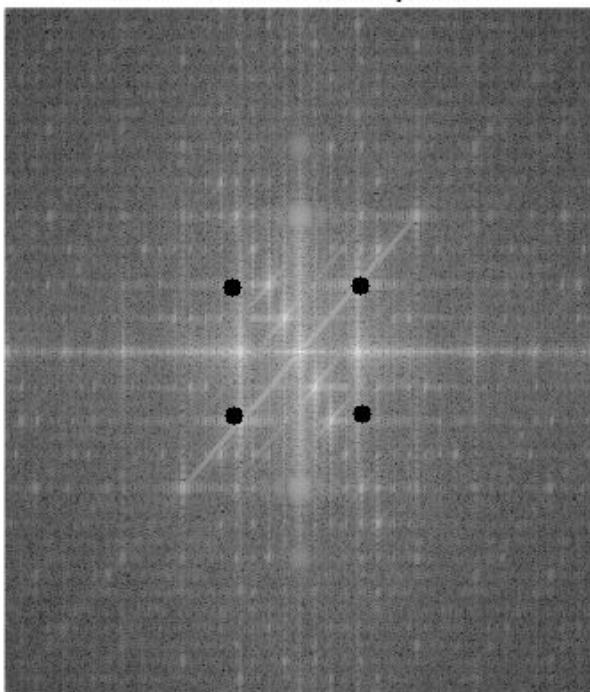
Image 1. in spectral domain, shifted image.

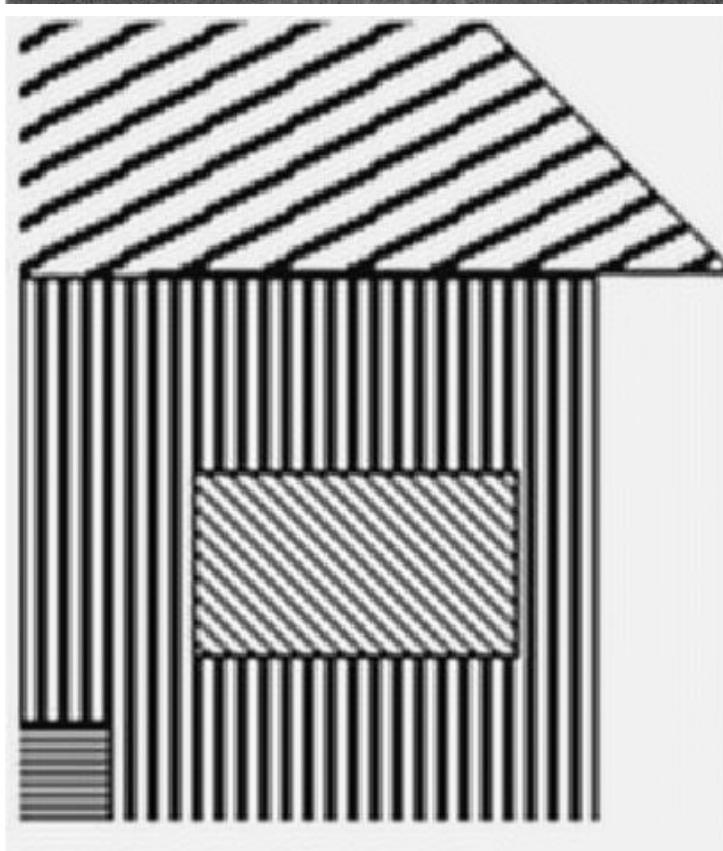
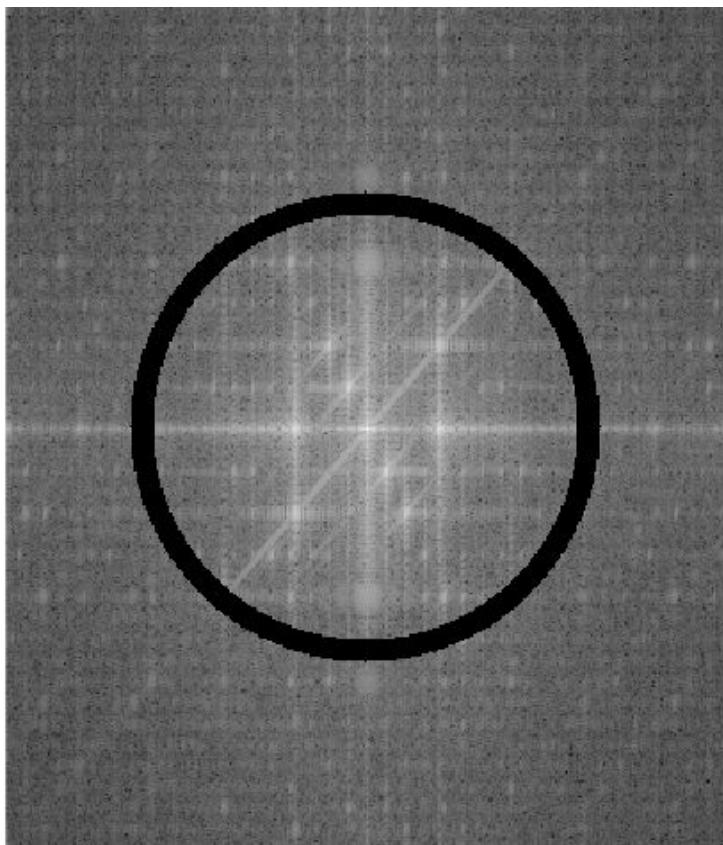




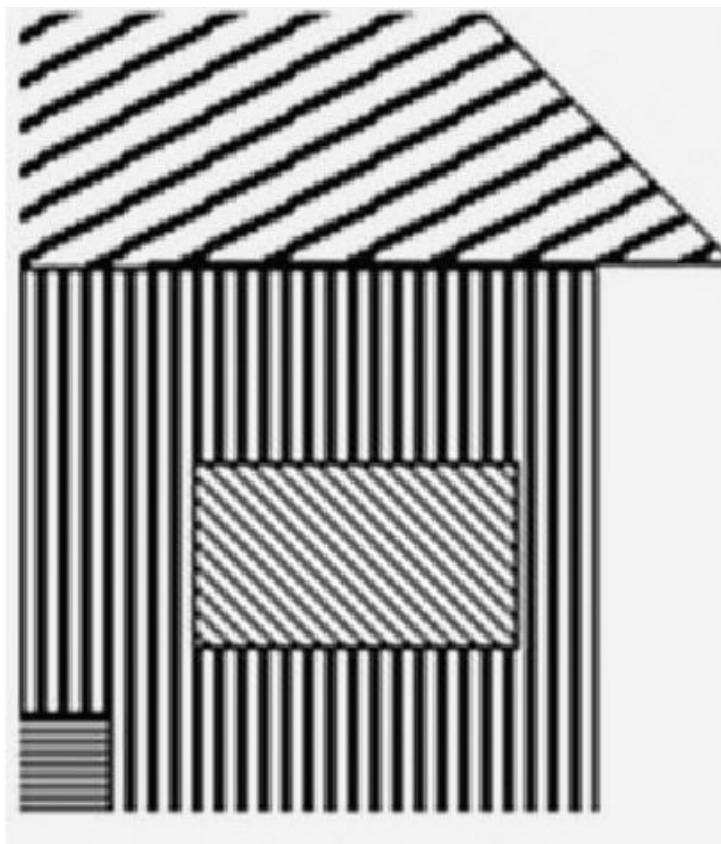
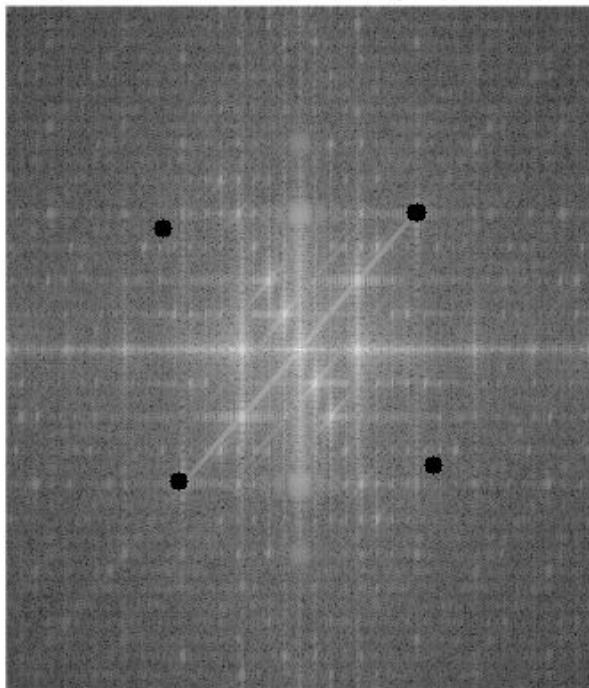


Point filter in shifted spectrum





Point filter in shifted spectrum



```
figure  
imshow(grayImgPer2, [])
```

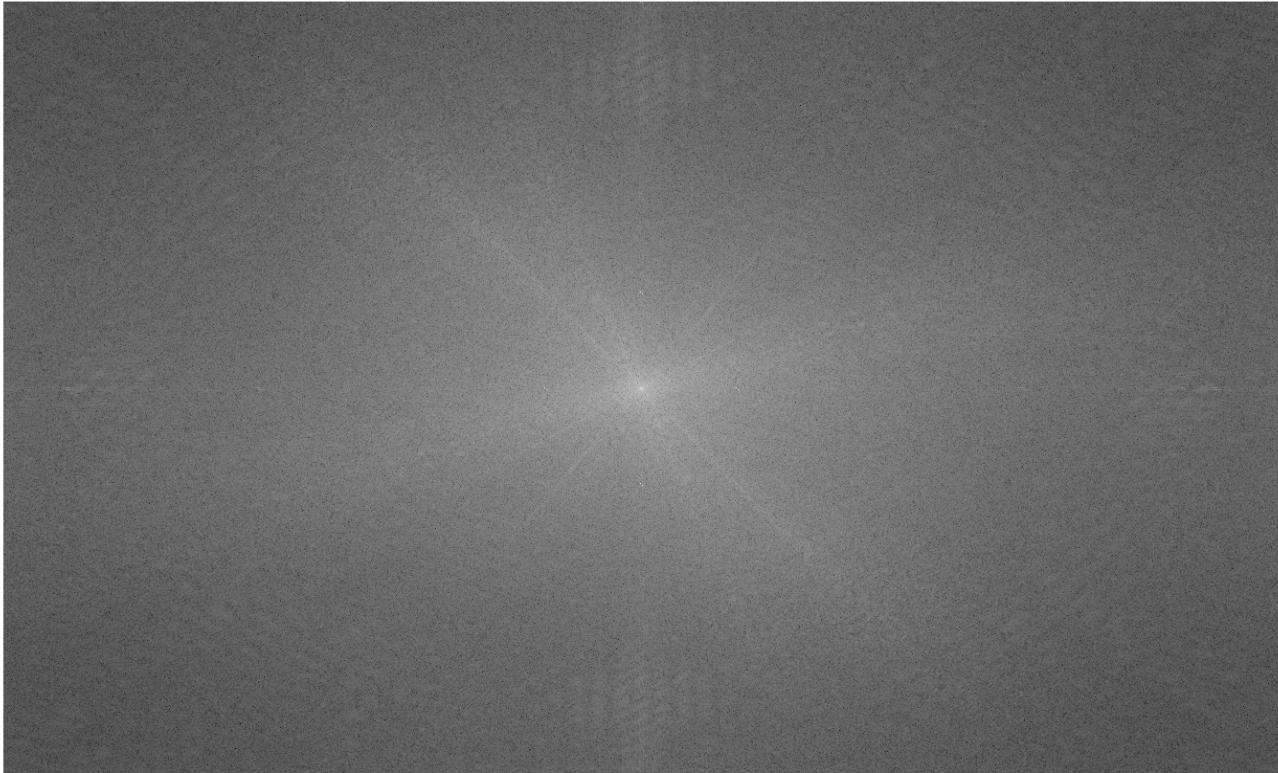
```
title("Second origin image", Color="green")
```

Second origin image

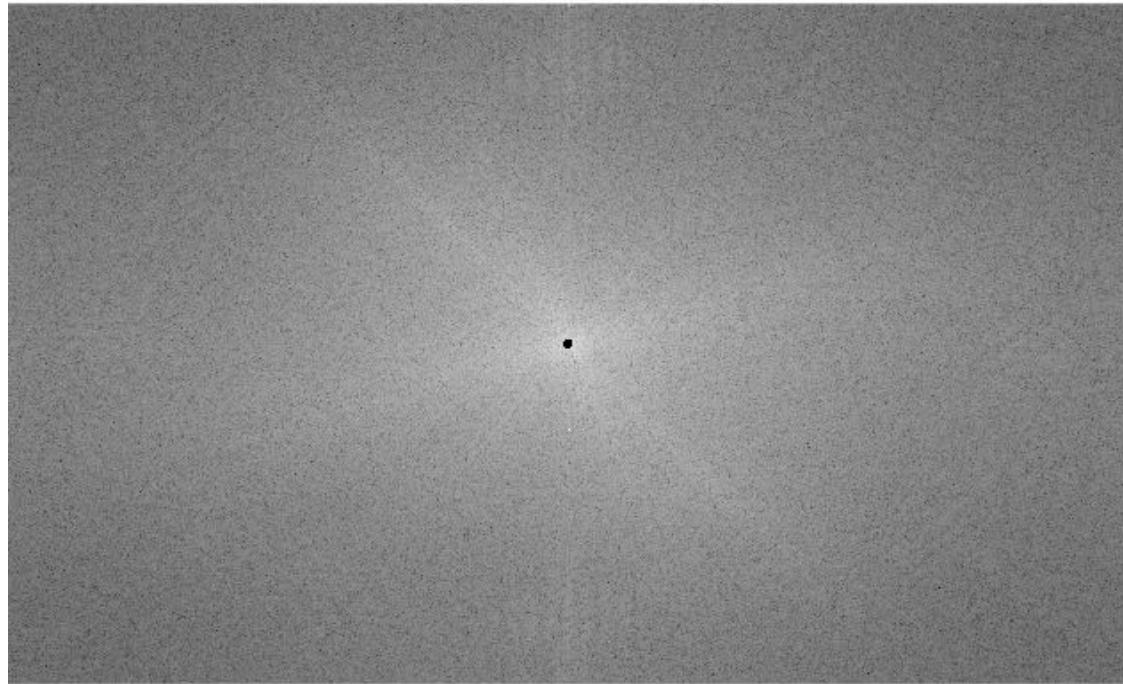


```
processSpectImgage(shiftFFTImg2, "Image 2. in spectral  
domain, shifted image.")
```

Image 2. in spectral domain, shifted image.



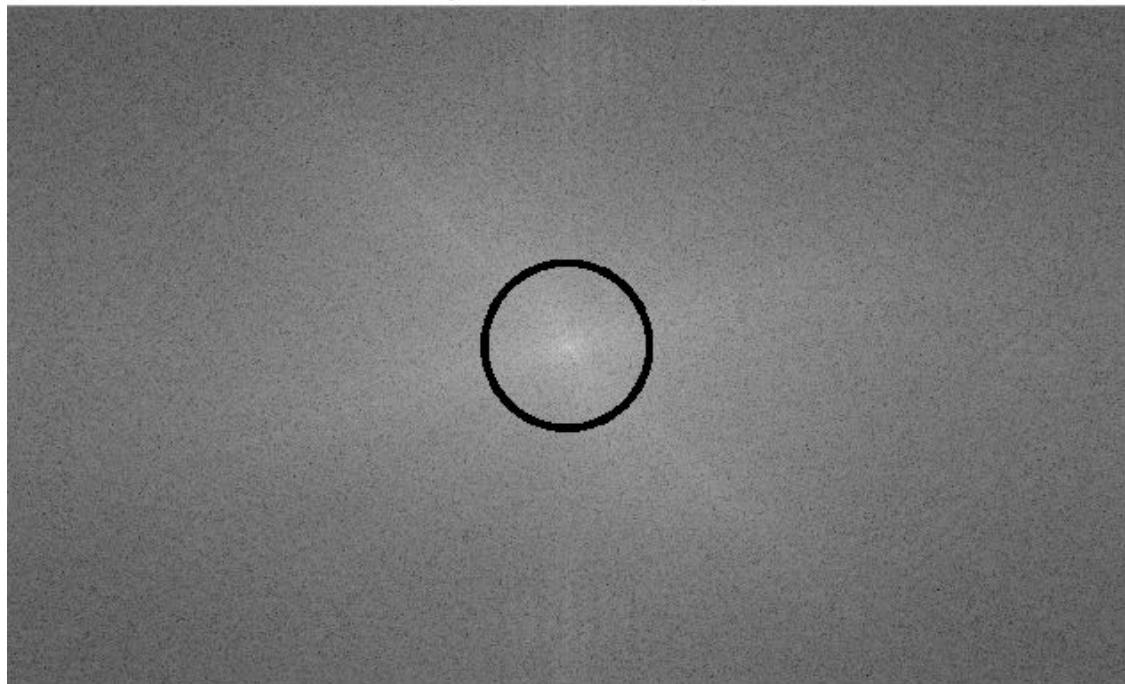
Band-stop filter in shifted spectrum



Img after filtration



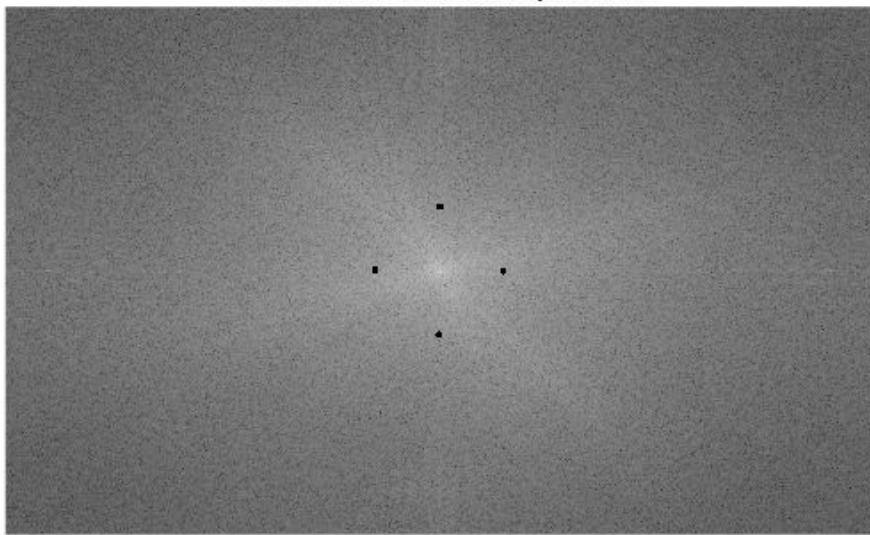
Band-stop filter in shifted spectrum



Img after filtration



Point filter in shifted spectrum

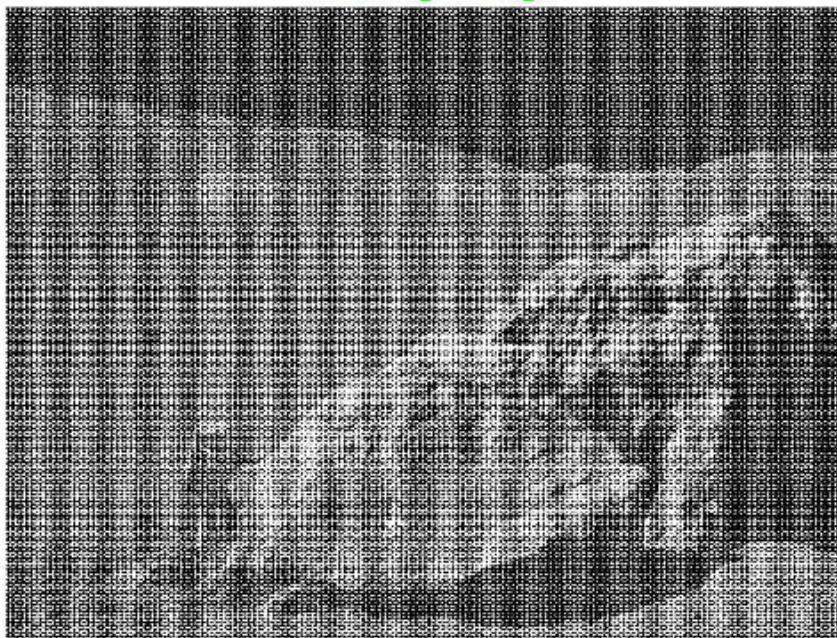


Img after filtration



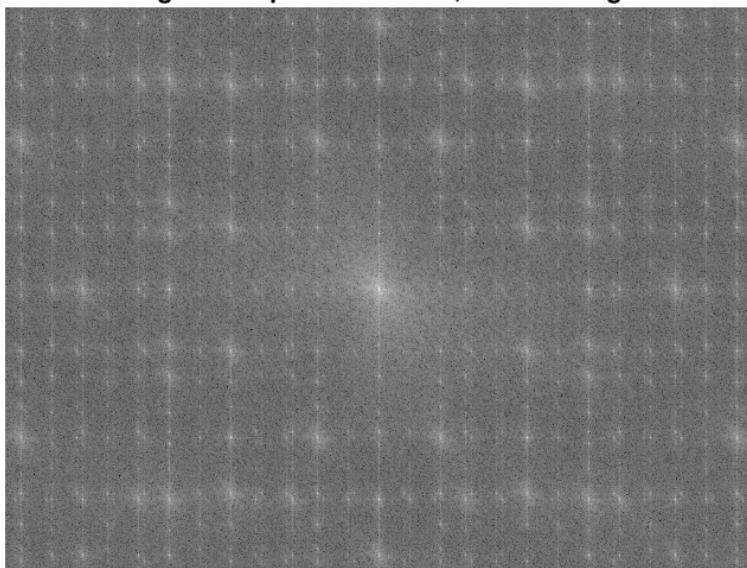
```
figure  
    imshow(grayImgPer3, [])  
    title("Third origin image", Color="green")
```

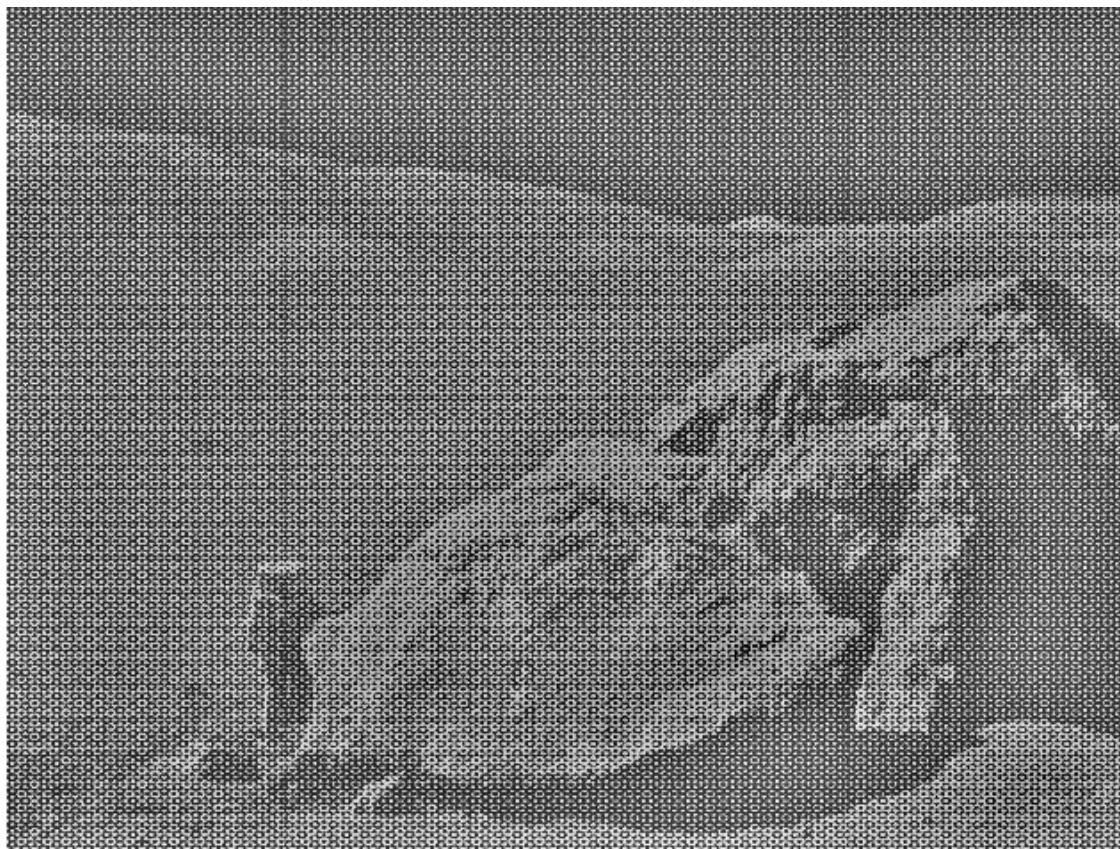
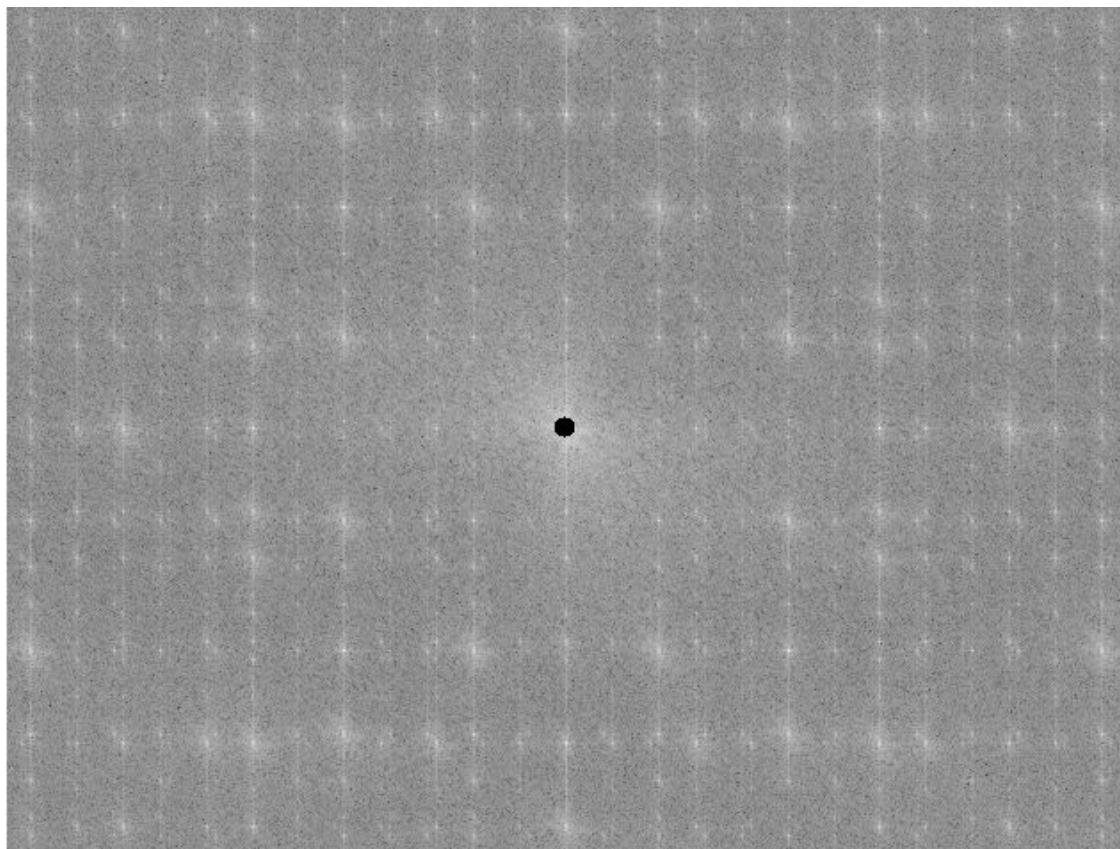
Third origin image

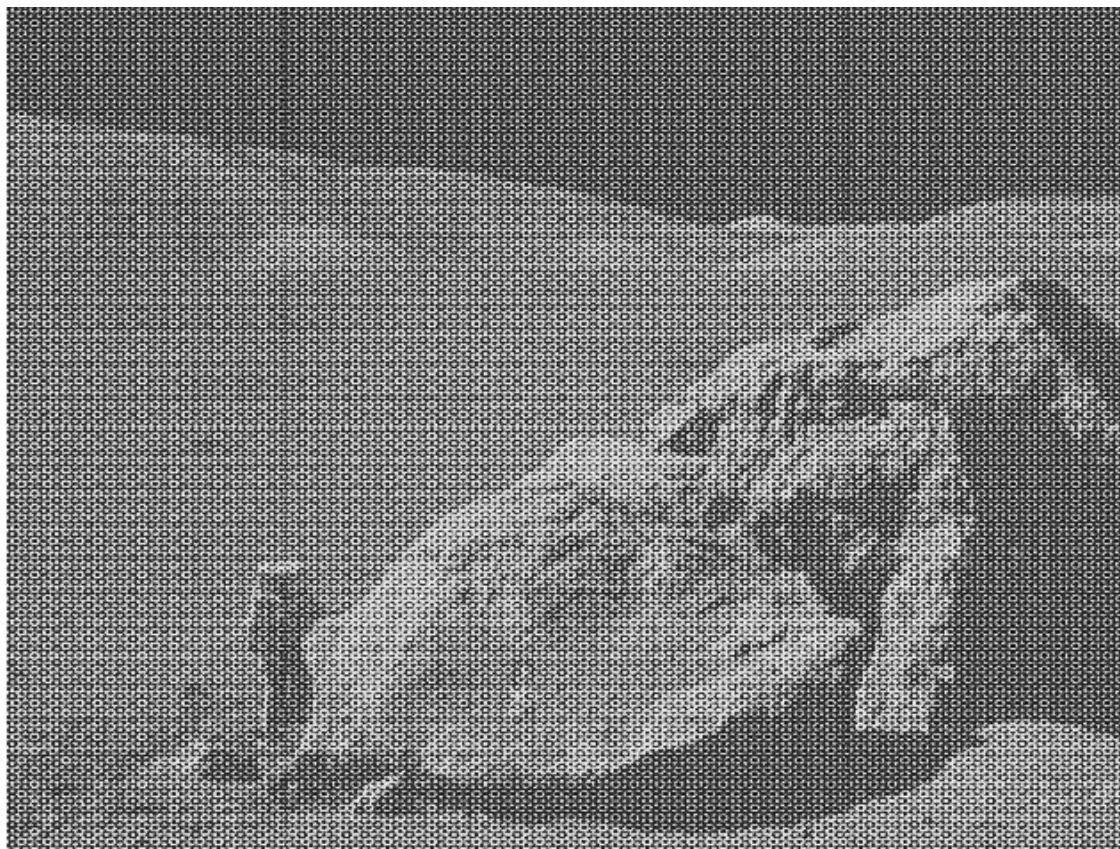
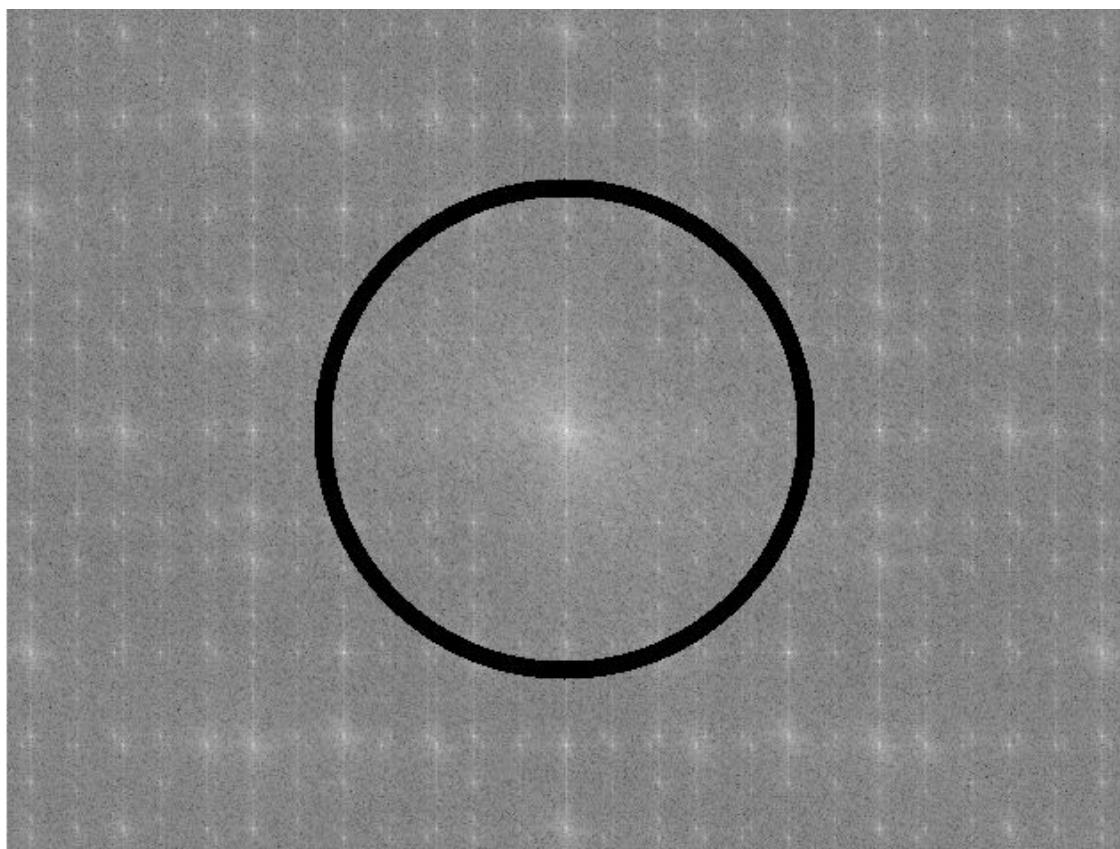


```
processSpectImgage(shiftFFTIimg3, "Image 3. in spectral  
domain, shifted image.")
```

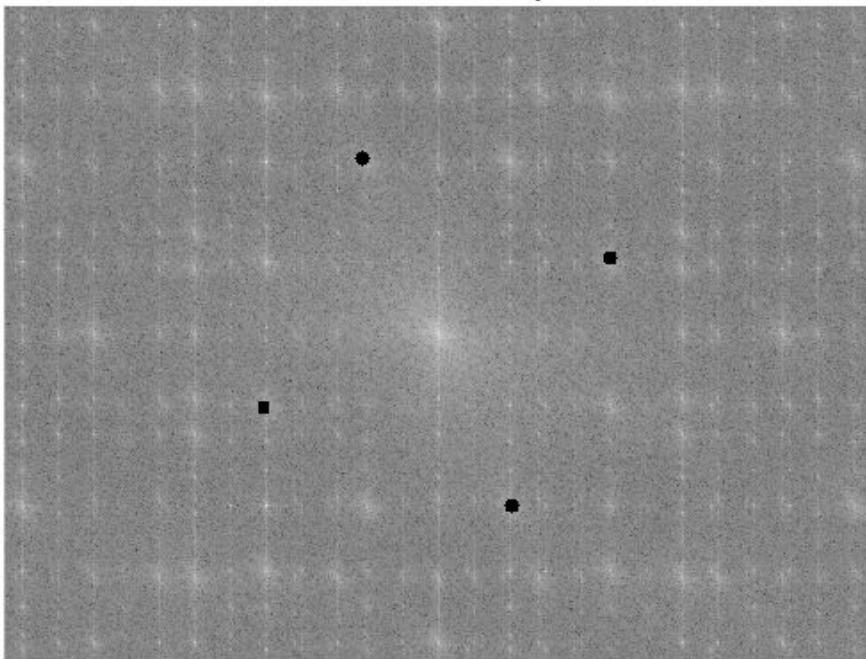
Image 3. in spectral domain, shifted image.

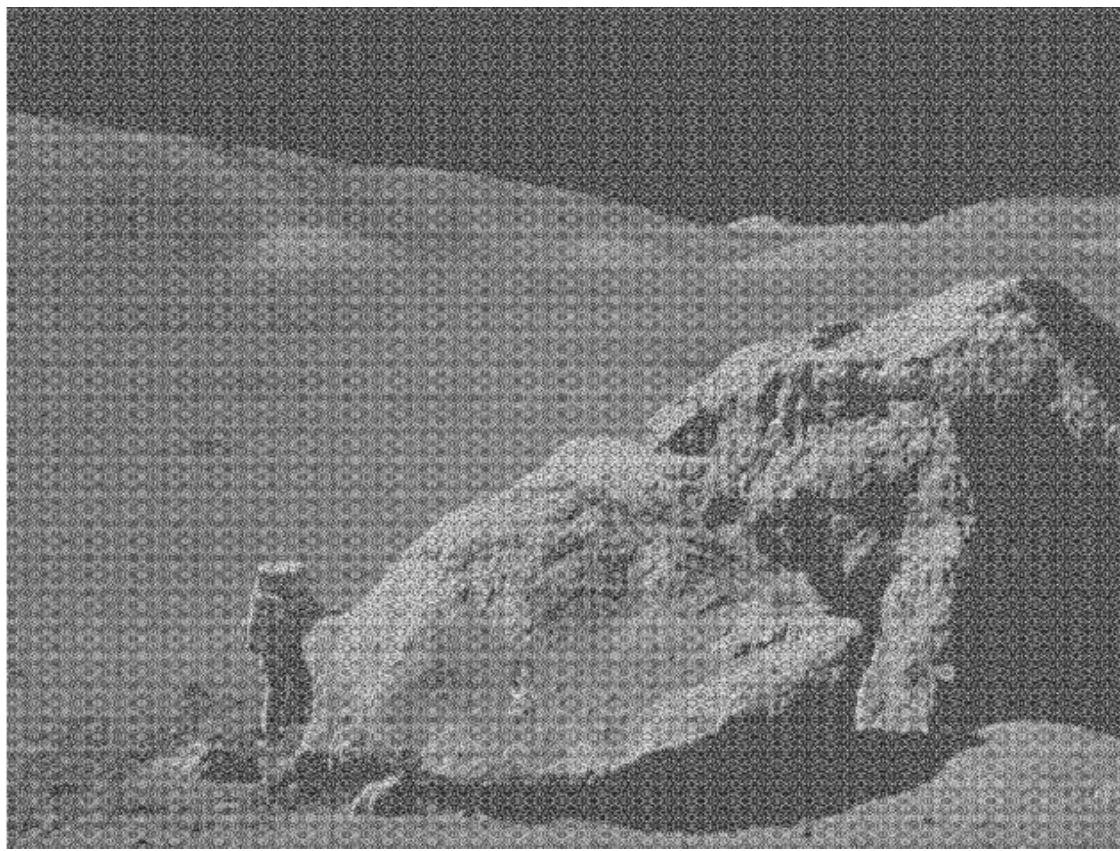
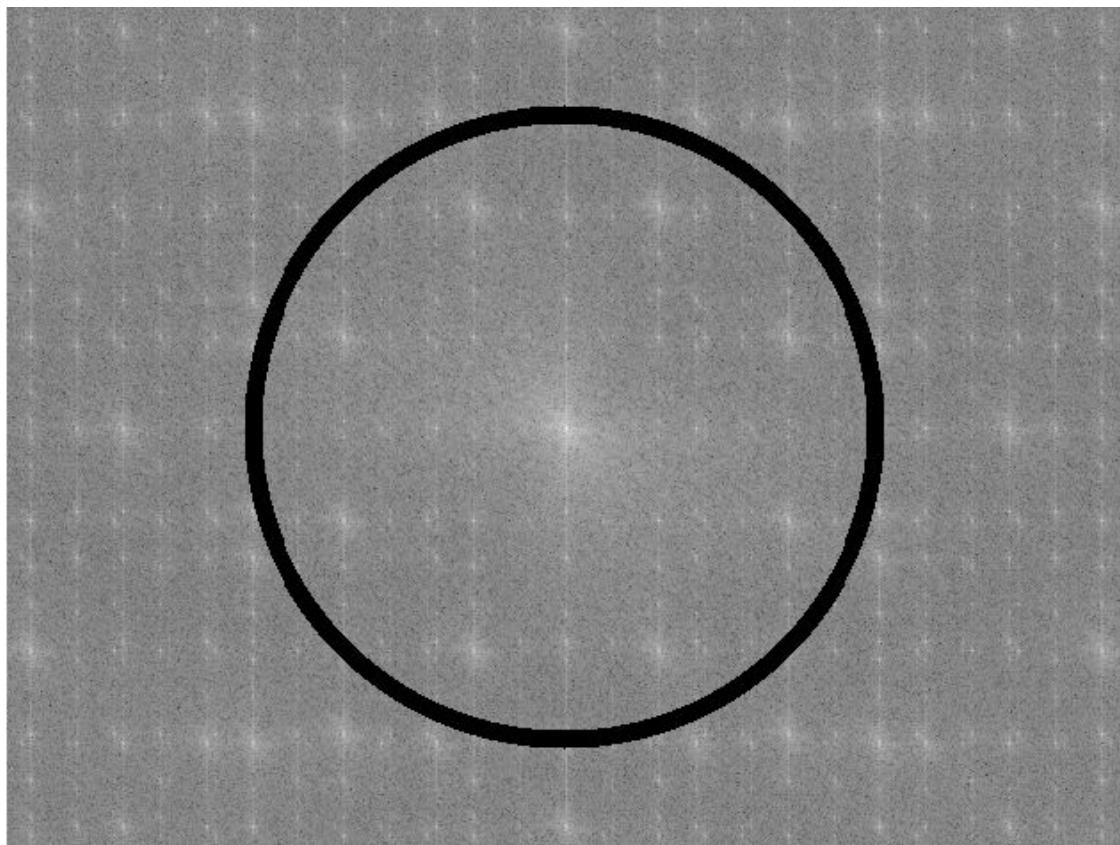




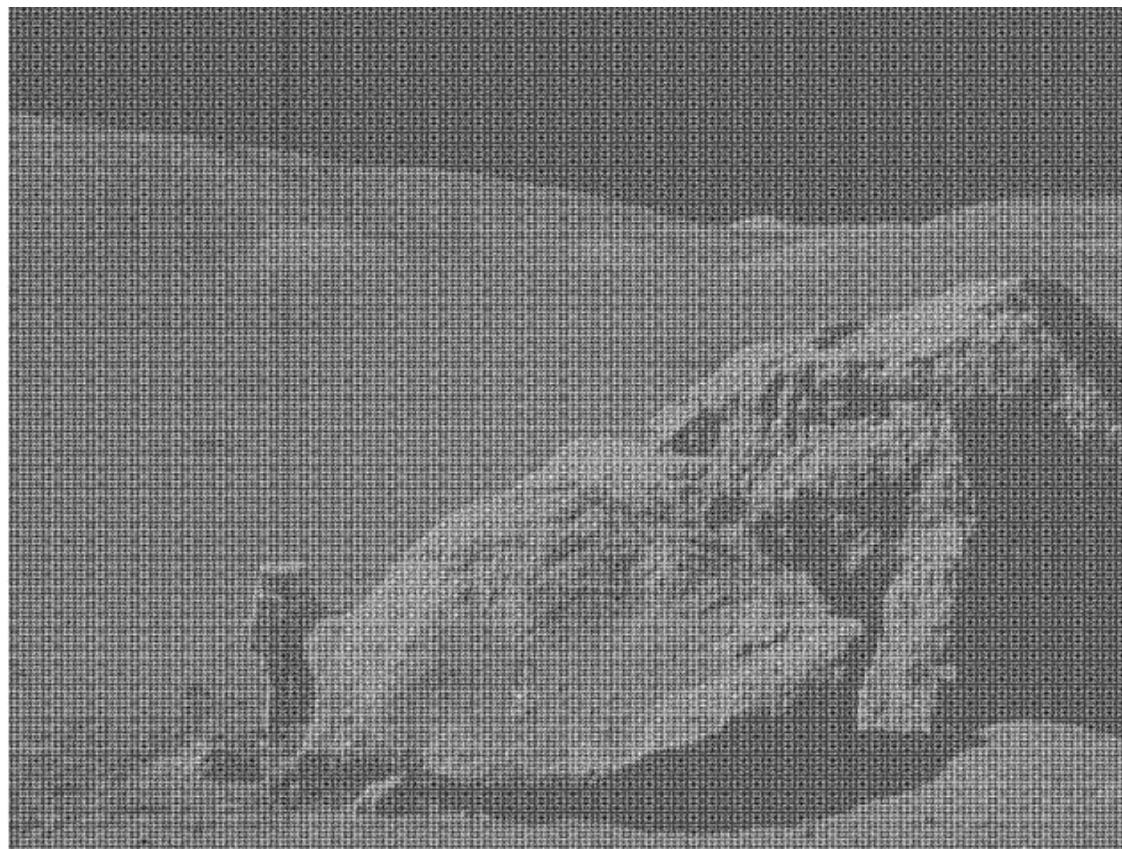
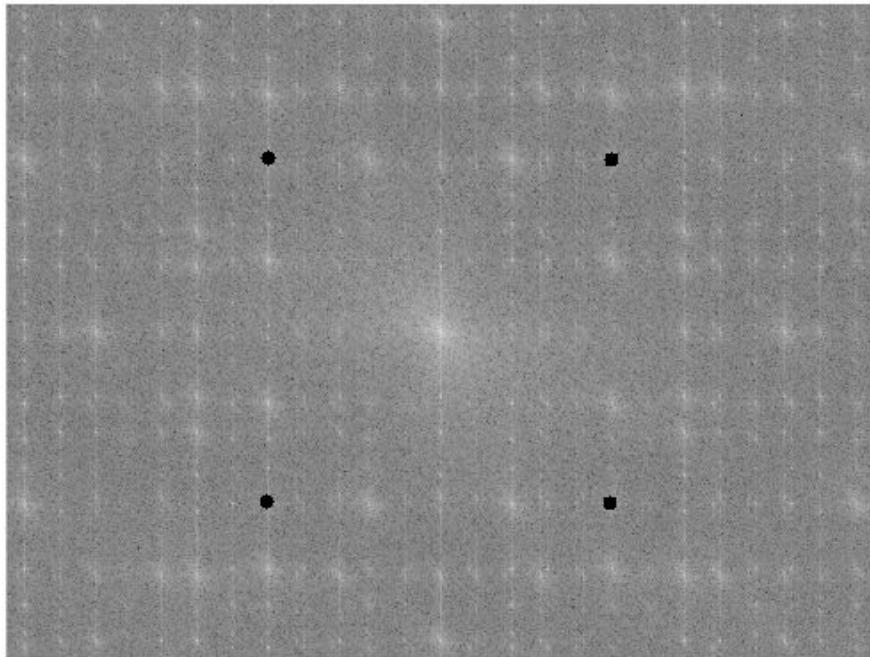


Point filter in shifted spectrum





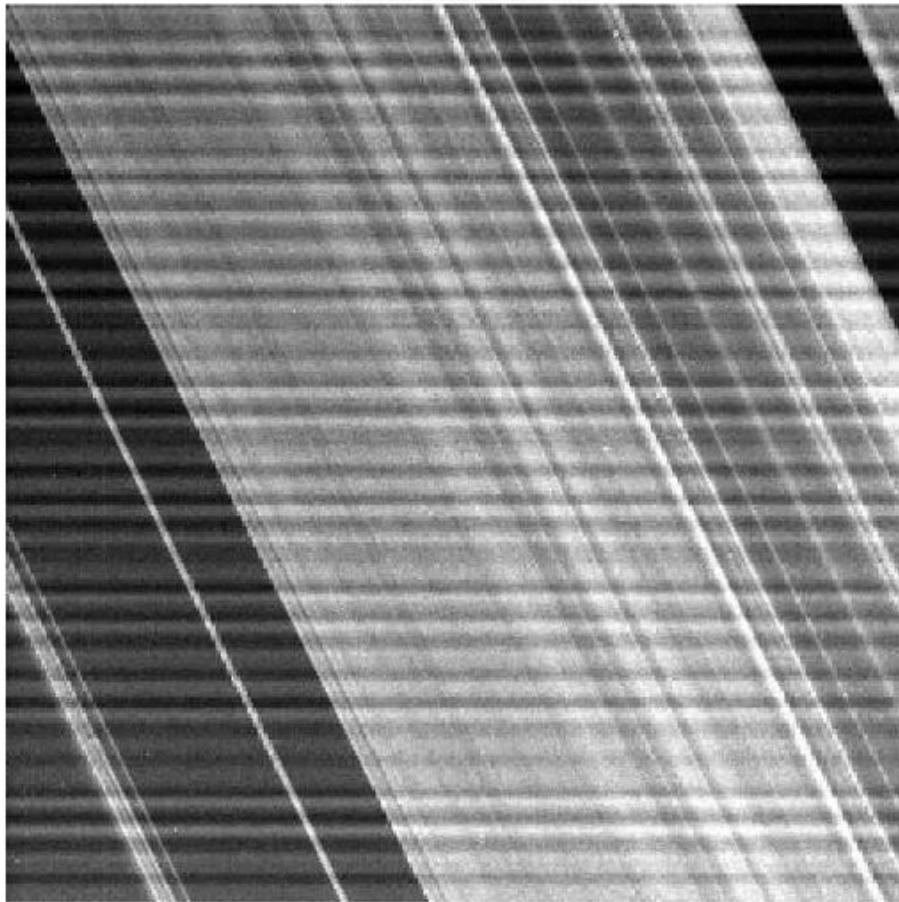
Point filter in shifted spectrum



```
figure  
imshow(grayImgPer4, [])
```

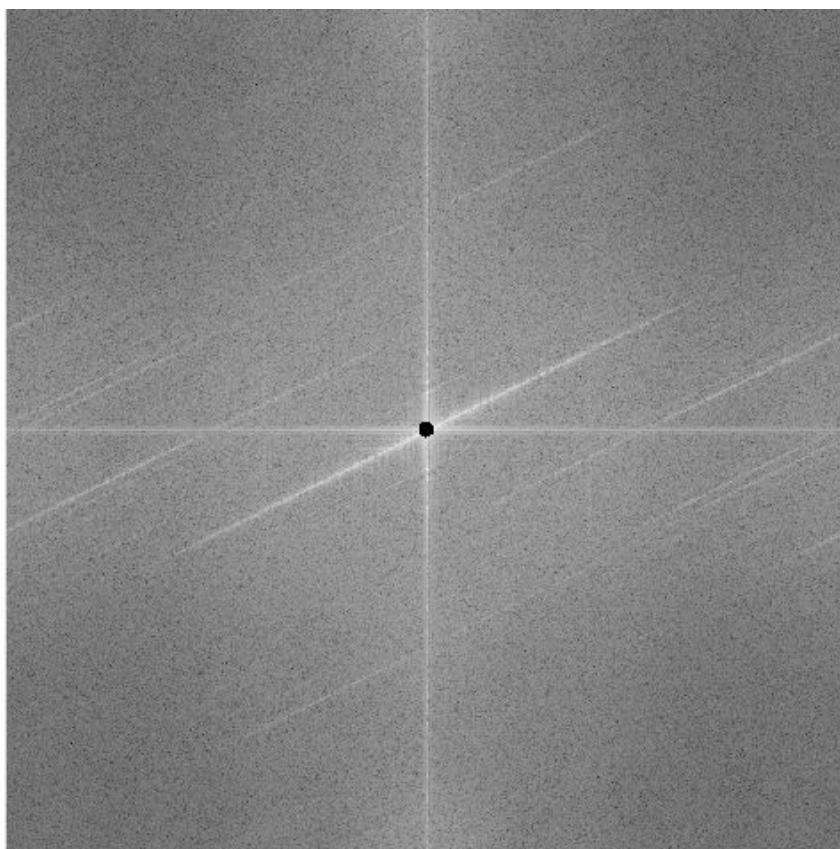
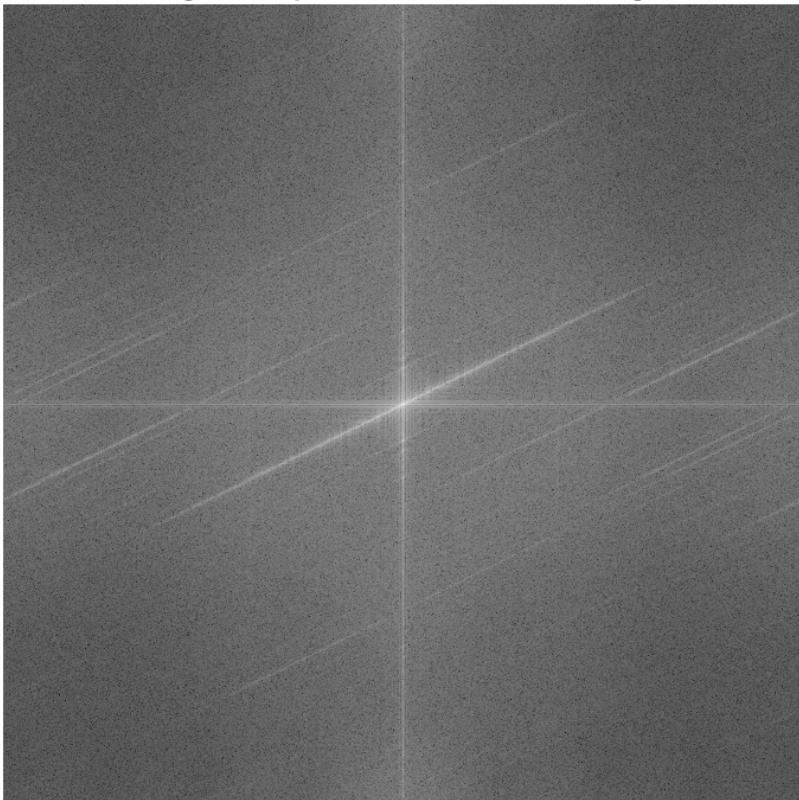
```
title("Fourth origin image", Color="green")
```

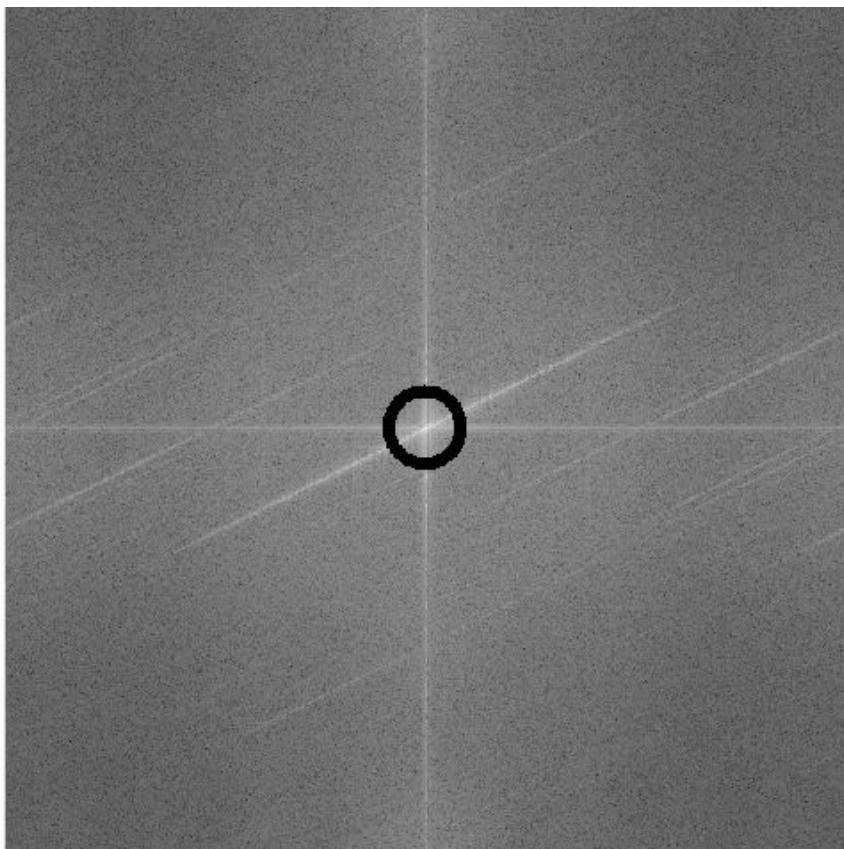
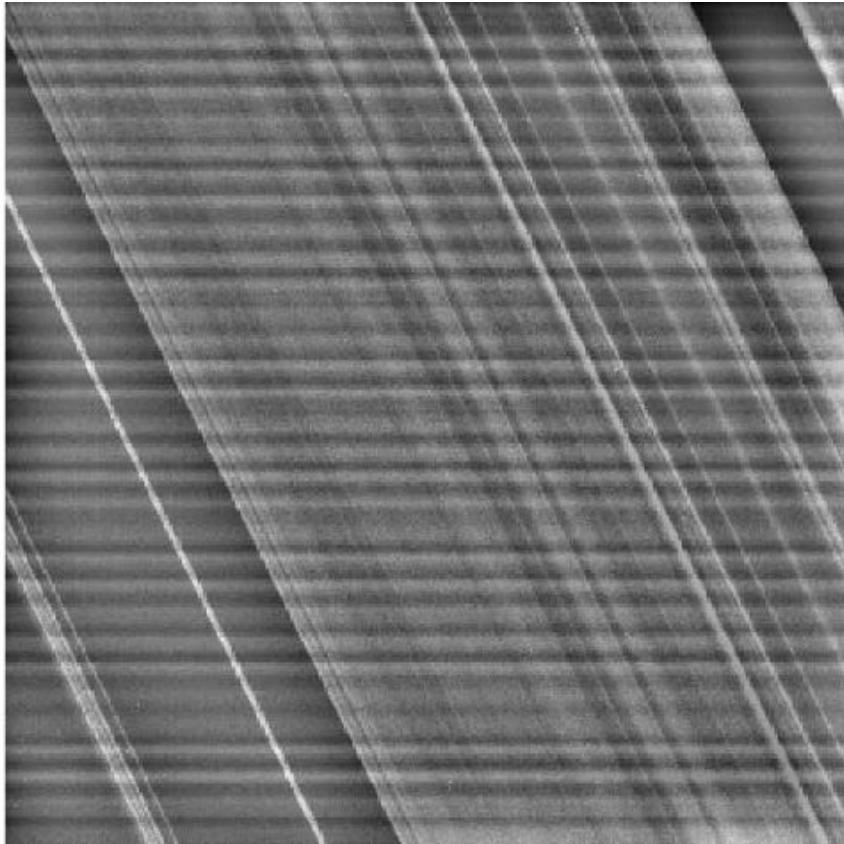
Fourth origin image

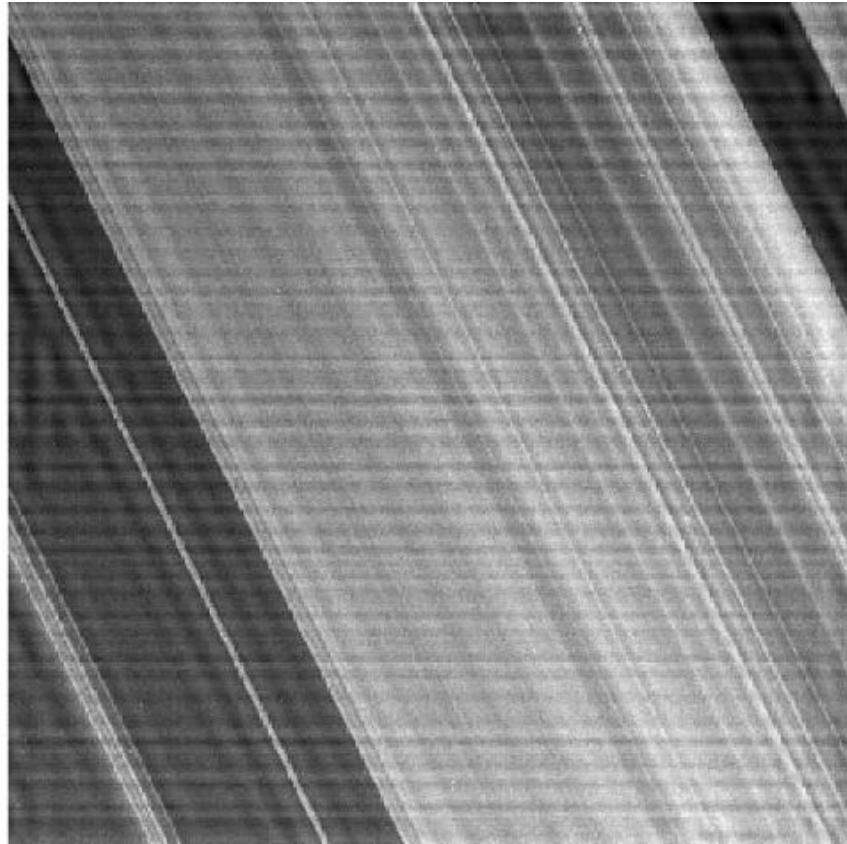


```
processSpectImgage(shiftFFTIimg4, "Image 4. in spectral  
domain, shifted image.")
```

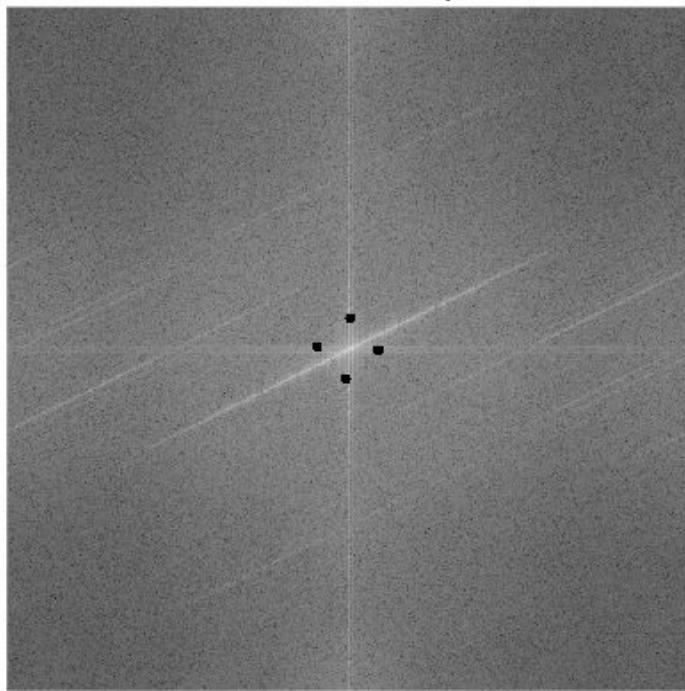
Image 4. in spectral domain, shifted image.

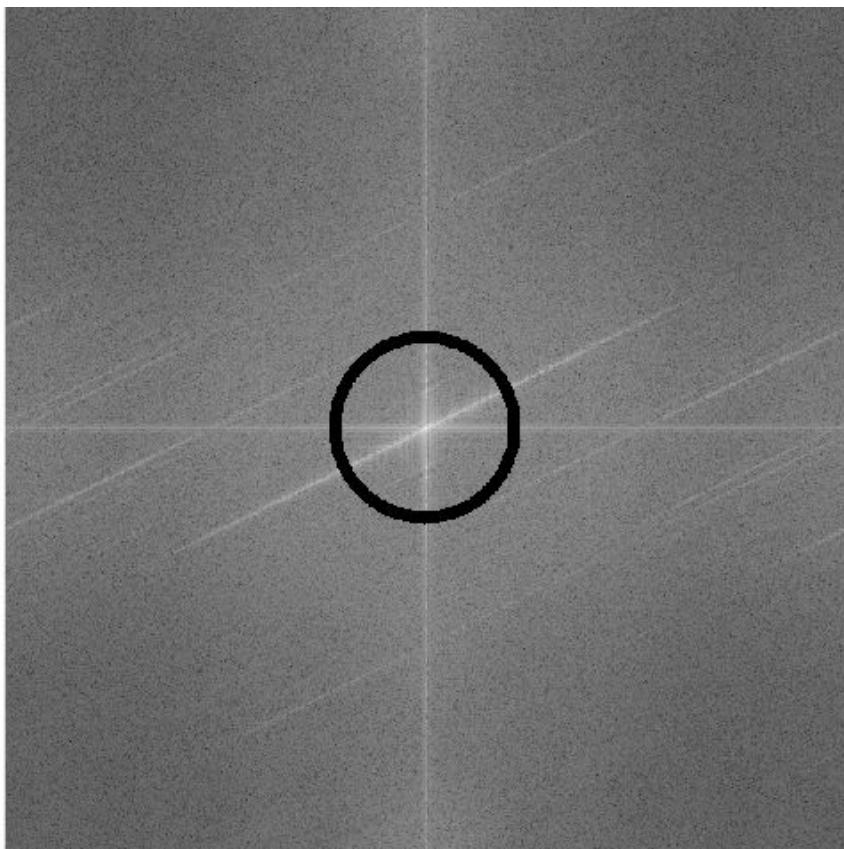
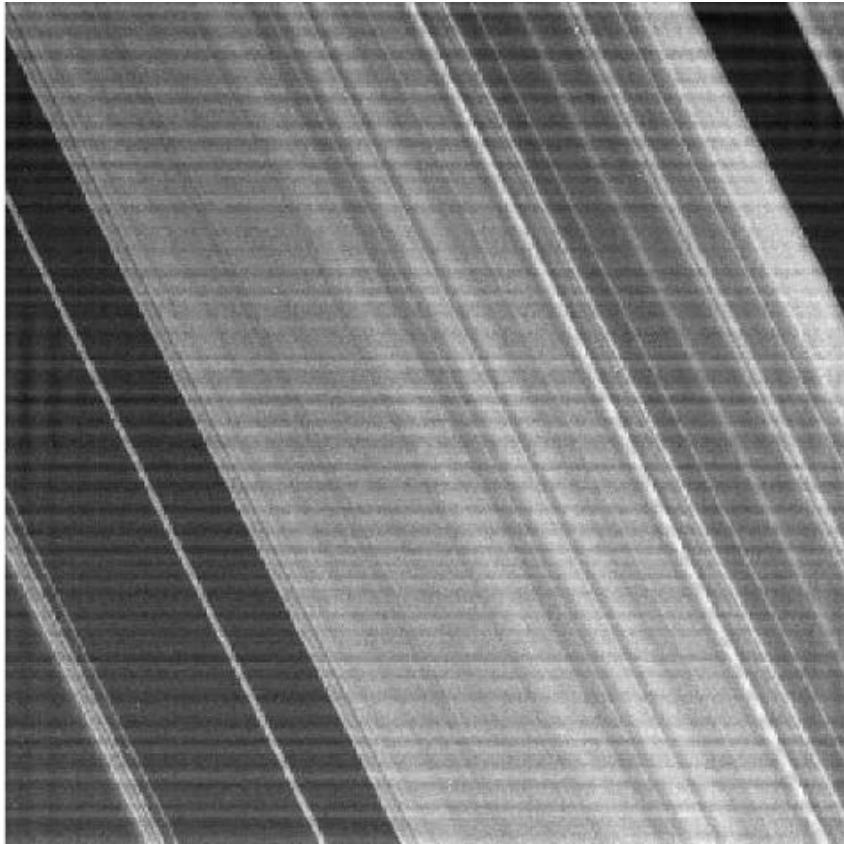


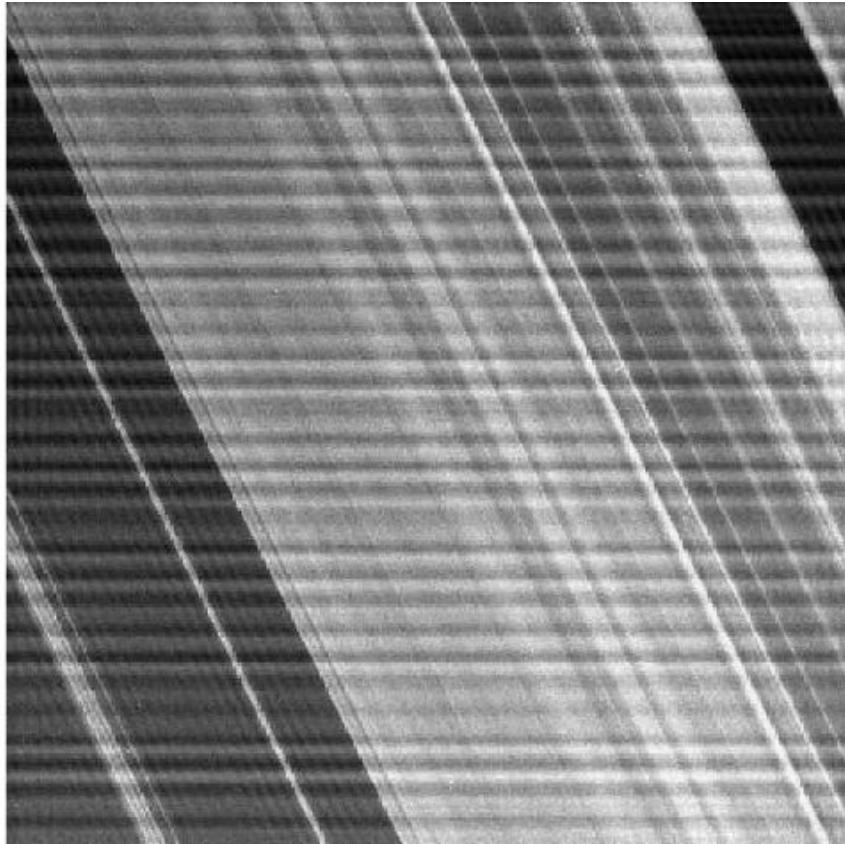




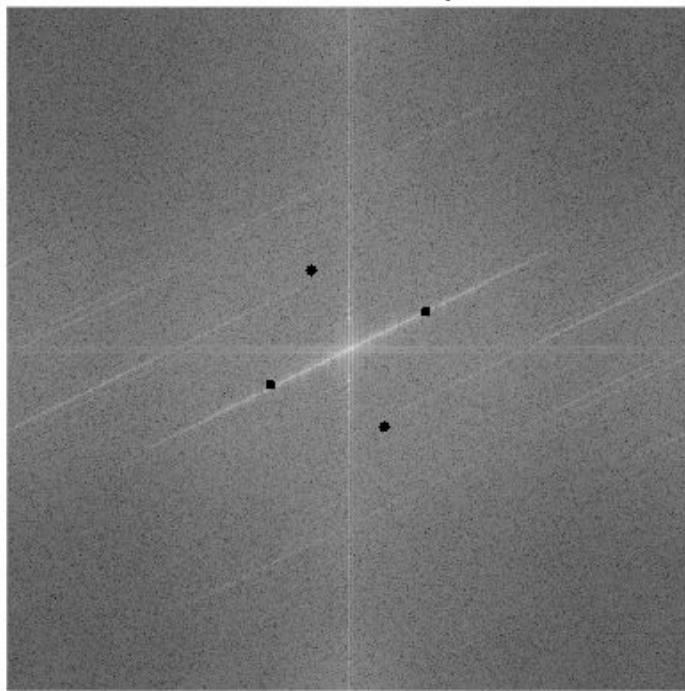
Point filter in shifted spectrum

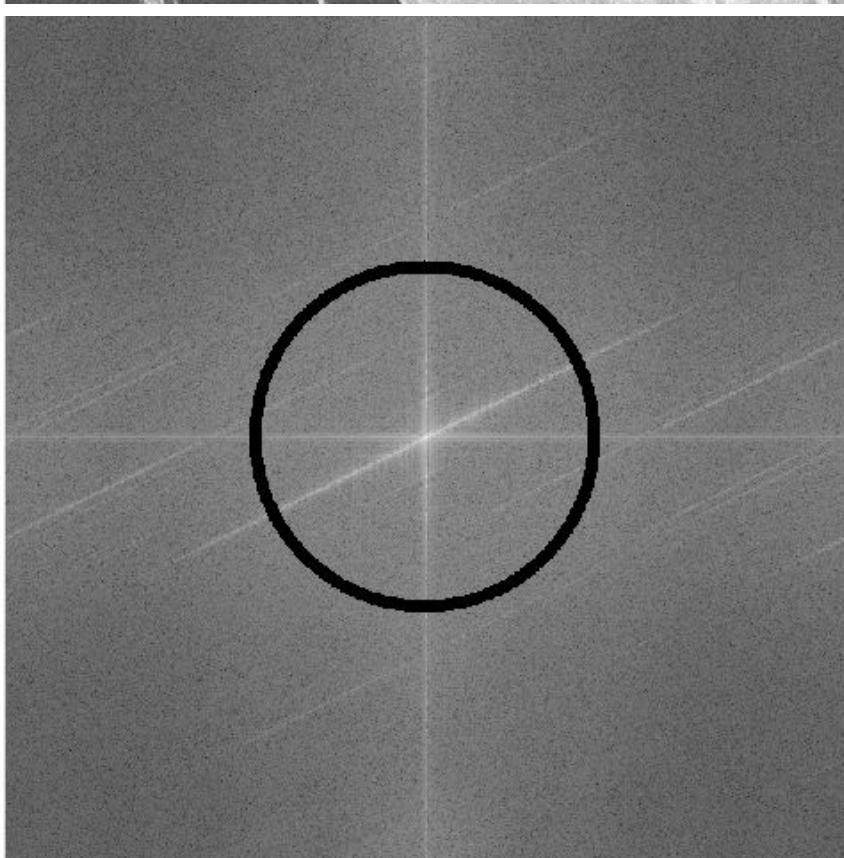
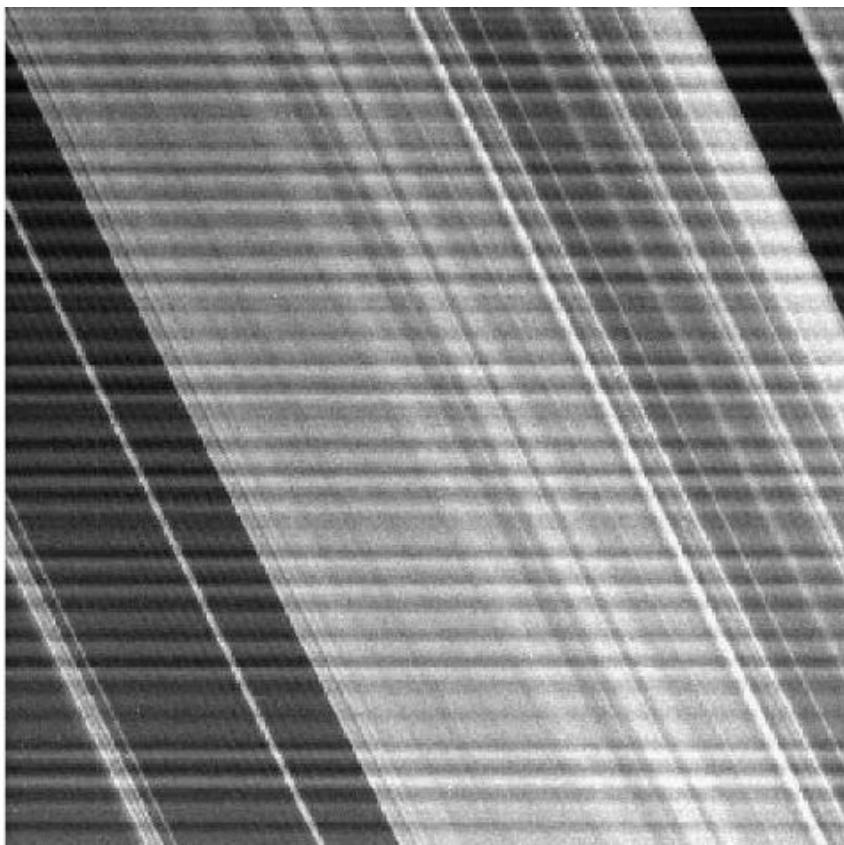


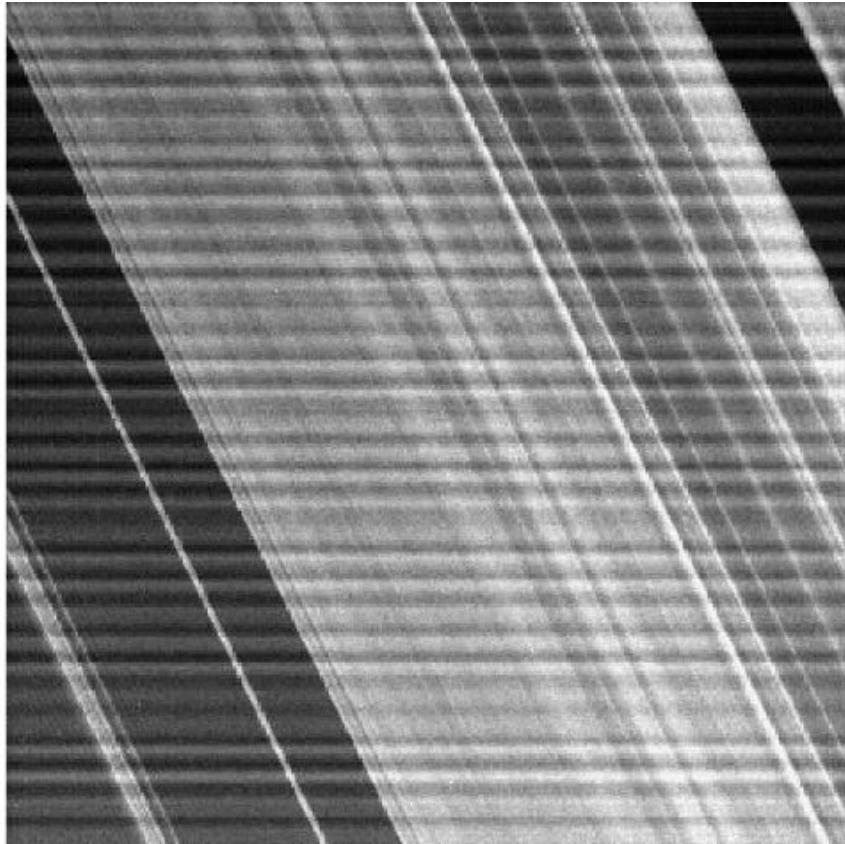




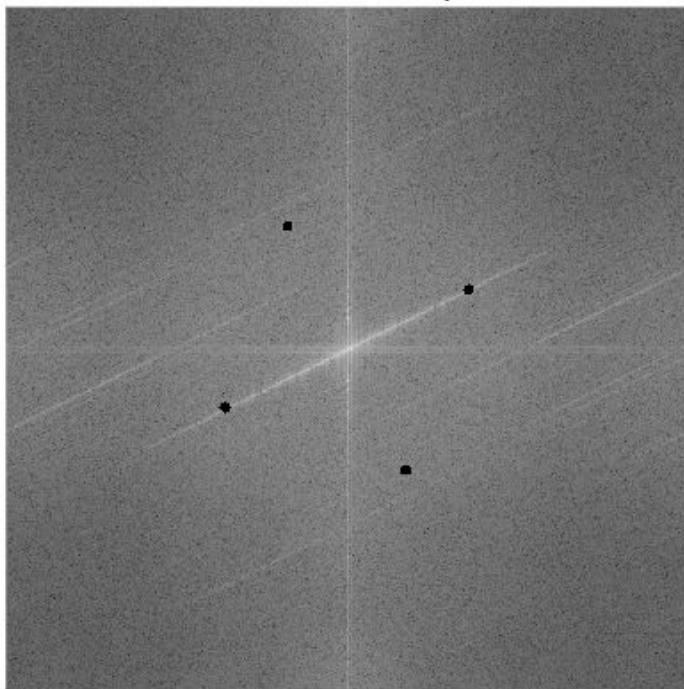
Point filter in shifted spectrum

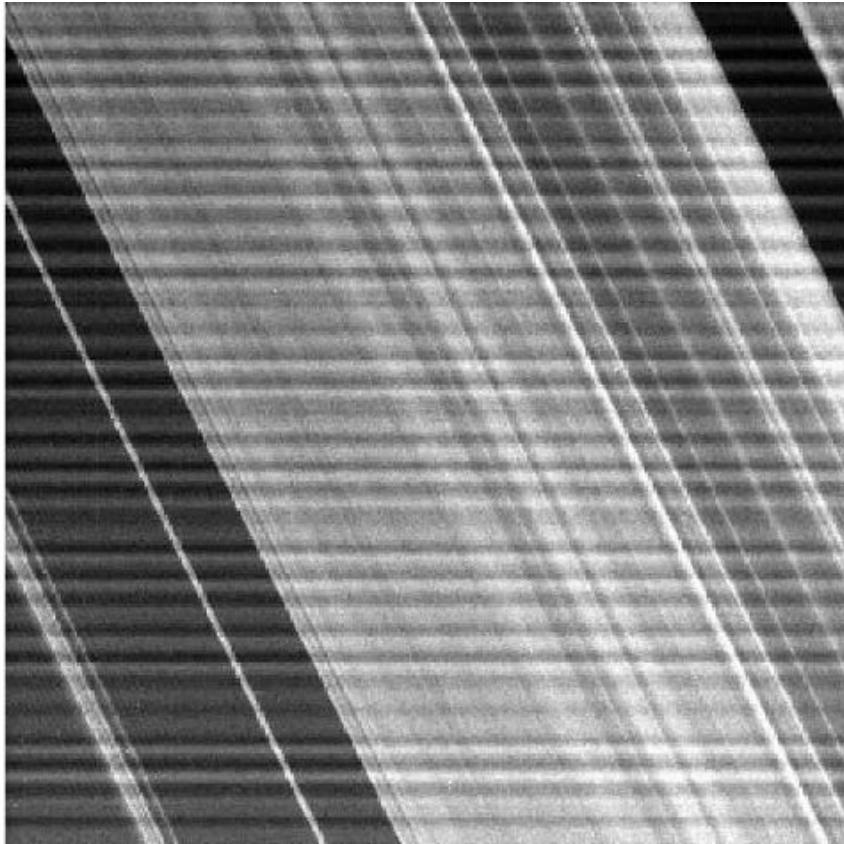






Point filter in shifted spectrum





Prečo je vhodné používateľovi umožniť zo spektra vymazať celú kružnicu ? (pomôcka: symetria):

1. Podla pozorovaných obrazkov, vymazaním celej kružnice vykazovalo rovnomernejši vysledok naprieč obrazkom, ako filter symetrických bodiek (vid v filtrovaná 1. obrazku). Avšak nebolo to až tak presne ako symetricky filter bodiek (viditeľné v filtrovaní obrazku).

Funkcie:

Vymazali sme zo spektra kružnicu, ktorá bola vytvorená pomocou inputu používateľa (lavé tlačidlo) ktoré bolo kliknuté (pravé stlačenie myši) ako na obrázku nižšie

```
function [imgOut, filtOut] = filtCircleImg(imgIn, x_, y_, width_)

imgSize = size(imgIn);
%4.1 Ideálny LP filter
offset = width_/2;
center = round(imgSize / 2);
radius = round(norm(center - [y_,x_]));
circlefilter = ones(imgSize);

for y = (center(1) - (radius + width_)): (center(1) + (radius + width_)) %
s predpokladom toho ze vzdialenosť od stredu by nemala presahovať radius bodov od stredu
    for x = (center(2) - (radius + width_)): (center(2) + (radius + width_))
```

```

        dist = norm(center - [y,x]);
        if(dist >= (radius - offset) && dist <= (radius + offset))
            % [y, x]
            circlefilter(y,x) = 0;
        end
    end
end
filtOut = circlefilter;
imgOut = ifft2(ifftshift(imgIn.*conv2(circlefilter, fspecial("gaussian"),
"same")));

```



```

figure;
axes('Units', 'normalized', 'Position', [0 0 1 1]);
imshow(log(abs(imgIn)).*circlefilter, [])
title("Band-stop filter in shifted spectrum")
figure;
axes('Units', 'normalized', 'Position', [0 0 1 1]);
imshow(real(imgOut), [])
title("Img after filtration")
end

```

Vymazali sme zo spektra simetrické body, ktoré boli vytvorené pomocou inputu používateľa (1 klik pravým tlačidlom miši).

```

% Využite symetriu, aby ste odstránili rovnaké miesto aj vo zvyšných kvadrantoch.
function [imgOut, filtOut] = filtPointImg(imgIn, x_, y_, width_)
imgSize = size(imgIn);
%4.1 Ideálny LP filter
pointFilter = ones(imgSize);
pointFilter = fillPoint(pointFilter,x_,y_, width_);

filtOut = pointFilter;
imgOut = ifft2(ifftshift(imgIn.*pointFilter));

figure;
axes('Units', 'normalized', 'Position', [0 0 1 1]);
nexttile
imshow(log(abs(imgIn)).*pointFilter, [])
title("Point filter in shifted spectrum")

figure;
axes('Units', 'normalized', 'Position', [0 0 1 1]);
imshow(real(imgOut), [])
title("Img after filtration")
% set(gcf,'position',[10,10,2000,2000])

end

function [imgOut] = fillPoint(imgIn, x_, y_, width_)

```

```

offset = width_/2;
center = round(size(imgIn) / 2);

for y = (y_ - width_):(y_ + width_) % s predpokladom toho ze vzdialenosť
od stredu by nemala presahovať radius bodov od stredu
    for x = (x_ - width_):(x_ + width_)
        dist = norm([y_,x_] - [y,x]);
        if(dist <= offset)
            diff = center - [y,x];

            imgIn(center(1) + diff(1), center(2) + diff(2)) = 0;
            imgIn(center(1) - diff(1), center(2) - diff(2)) = 0;

            imgIn(center(1) + diff(2), center(2) - diff(1)) = 0;
            imgIn(center(1) - diff(2), center(2) + diff(1)) = 0;

        end
    end
end
imgOut = imgIn;
end

```

Umožnili sme výmaz zo spektra okolie miesta, na ktoré bolo kliknuté (ľavé stlačenie myši), respektíve určité spektrum frekvencií. Toto bolo dokázané použitím vytvorením 2 typov filtrov (prisluchajúcich 2 tlačidlam myši ktoré inicializujú ich vytvorenie a aplikáciu).

Lavé tlačidlo myši inicializuje band stop filter, ktorý sa prejaví vo vizualizácii ako čierny kruh na bielom pozadí, napäťom tomu pravé tlačidlo myši vytvára filter ktorý obsahuje symetrické body kliku, symetria bola dosiahnutá ziskania absolutného vzdialenosťného vectora diff (vzdialenosť bodu od stredu), variaciami znamienok jeho prvkov sme dosiahly "našu" symetriu bodov.

```

function [imgOut] = processSpectImage(imgIn, img_text)
fig1 = figure;
imshow(log(abs(imgIn)), []);
title(img_text);
button1 = 1;

while button1 == 1 || button1 == 3 % Spekturm sme umožníme filtrovať
interaktívne - používateľ má možnosť kliknutím vybrať miesto v spektre, ktoré má
byť vyfiltrované.
% To sme urobili ----> Navrhli sme podľa pozície dominantných vrcholov v
spektre filter, pomocou ktorého by sme vo výsledku vedeli odstrániť periodický šum
z obrazu.
set(0, 'CurrentFigure', fig1) % Make fig current
[x1,y1,button1] = ginput(1);
x1 = round(x1);
y1 = round(y1);
if button1 == 1
    [outCircleImg1, circleFilt1] = filtCircleImg(imgIn, x1, y1, 10);

```

```
elseif button1 == 3
    [outpointImg1, pointFilt1] = filtPointImg(imgIn, x1, y1, 10);
end
end
```