

Gliwice, 25.10.2022

# Przetwarzanie Obrazów Cyfrowych

## Transformata Fouriera



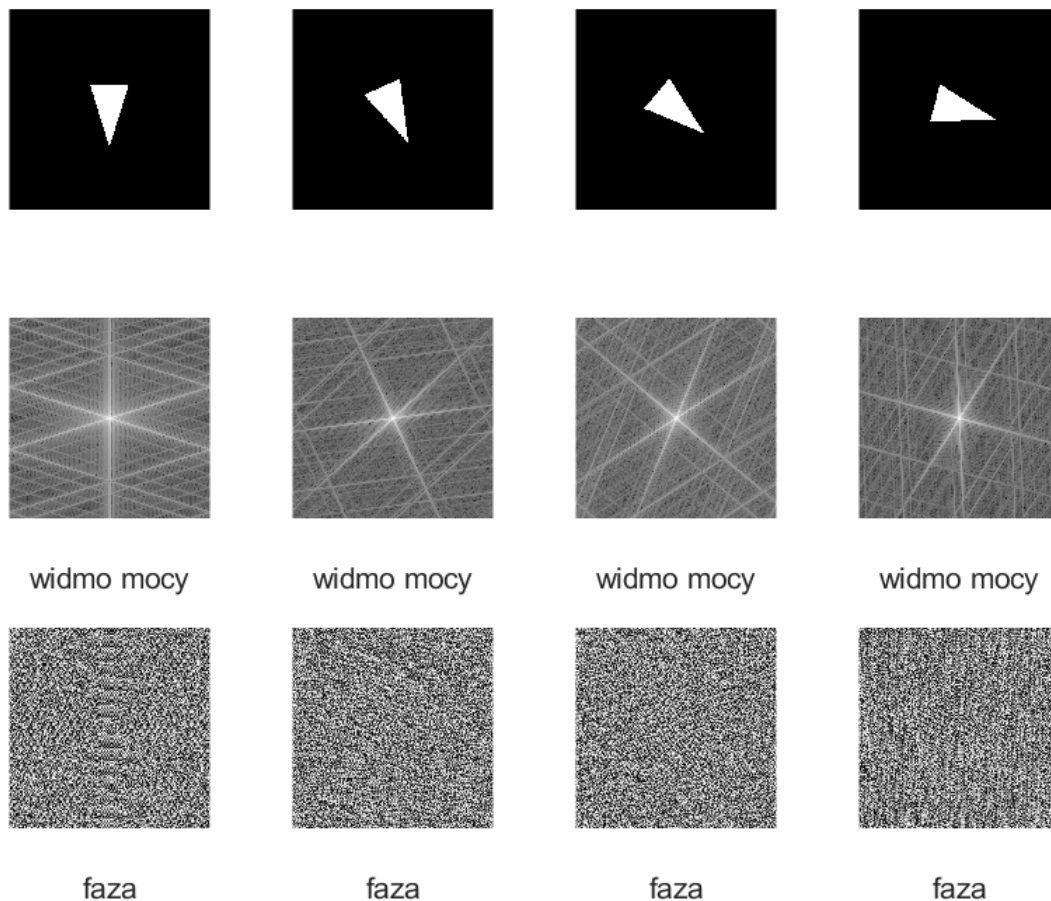
**Politechnika  
Śląska**

Jakub Zeifert

# 1 Przygotowanie obrazów testowych

(Zaliczone w trakcie zajęć)

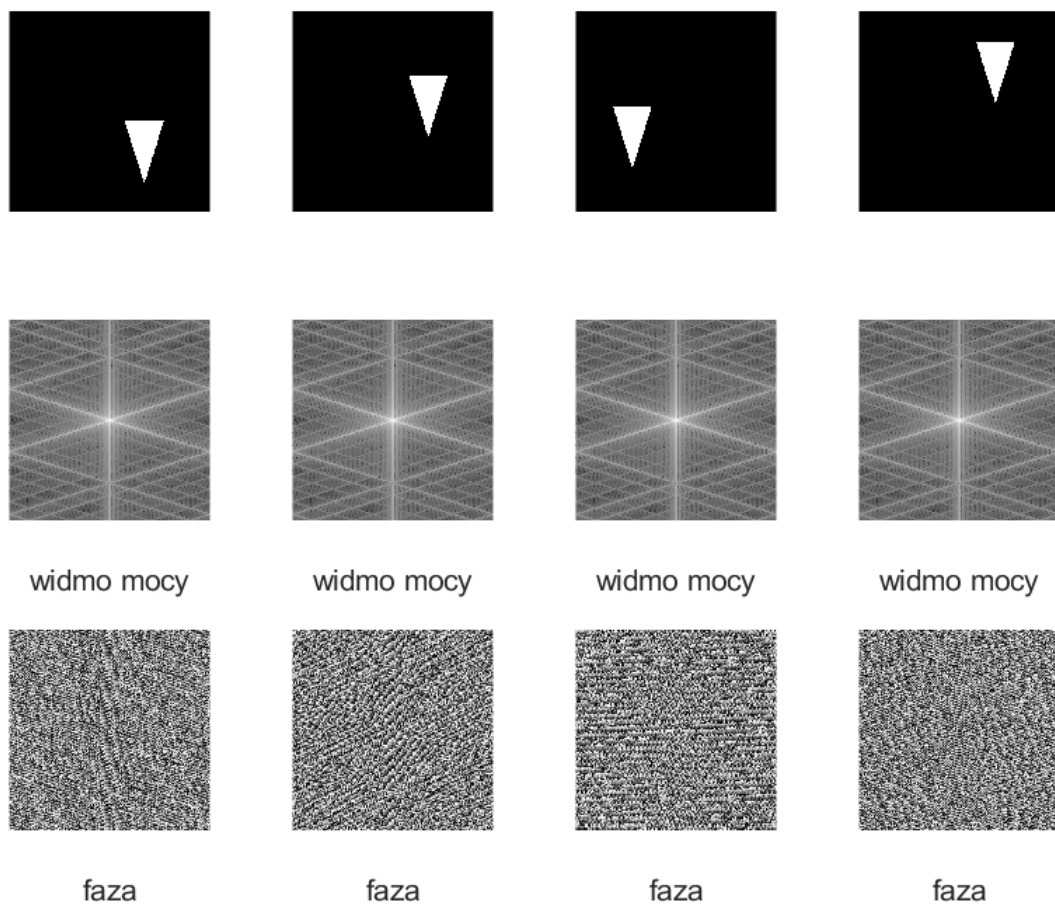
1. trójkąty różnej orientacji (jeden trójkąt, na jednym obrazie) (imrotate)



Rysunek 1: trójkąty różnej orientacji

Przy rotacji możemy zauważyć, że widmo mocy nie zmienia kształu, lecz wraz z obrotem widmo jest również zrotowane. Najjaśniejsze miejsca wyznaczają granicę przejścia z białego w czerni, a więc krawędzie trójkąta.

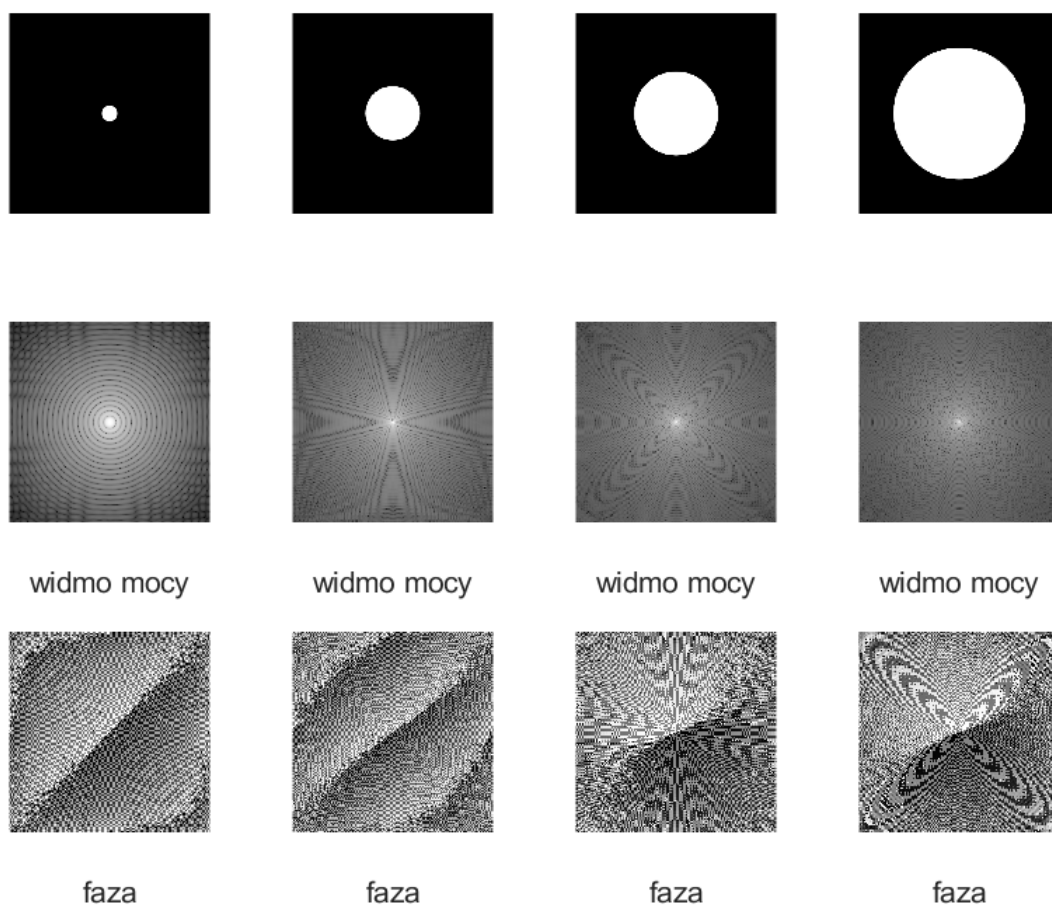
## 2. trójkąty o różnym położeniu (imtranslate)



Rysunek 2: trójkąty o różnym położeniu

Przy translacji wyniki w widmie mocy jest takie samo, co wynika z wykorzystania funkcji *fftshift*. Powoduje ona umiejscowienie najniższych częstotliwości na środek. A więc nie ma znaczenia czy obraz jest przesunięty czy nie. Póki nie występuje rotacja albo zanik pewnych informacji poprzez przycięcie, to widmo nie powinno mieć różnic.

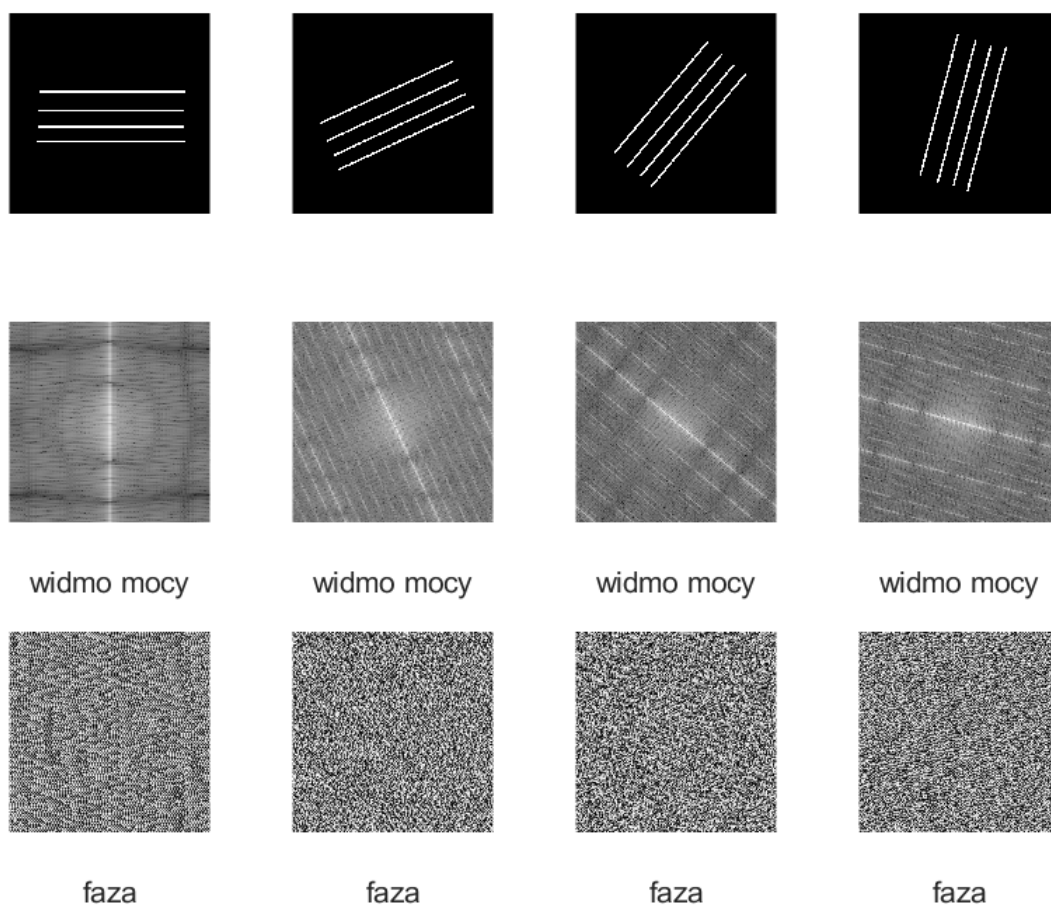
### 3. koła różnej wielkości (imresize)



Rysunek 3: koła różnej wielkości

Tutaj możemy zaobserwować niewielkie różnice w wyglądzie widm, jednakże przeskoków między kolorami jest podobna ilość. Niewielkie zmiany wynikają z faktu, że koło po przybliżeniu okazuje się być lekko "postrzępionym".

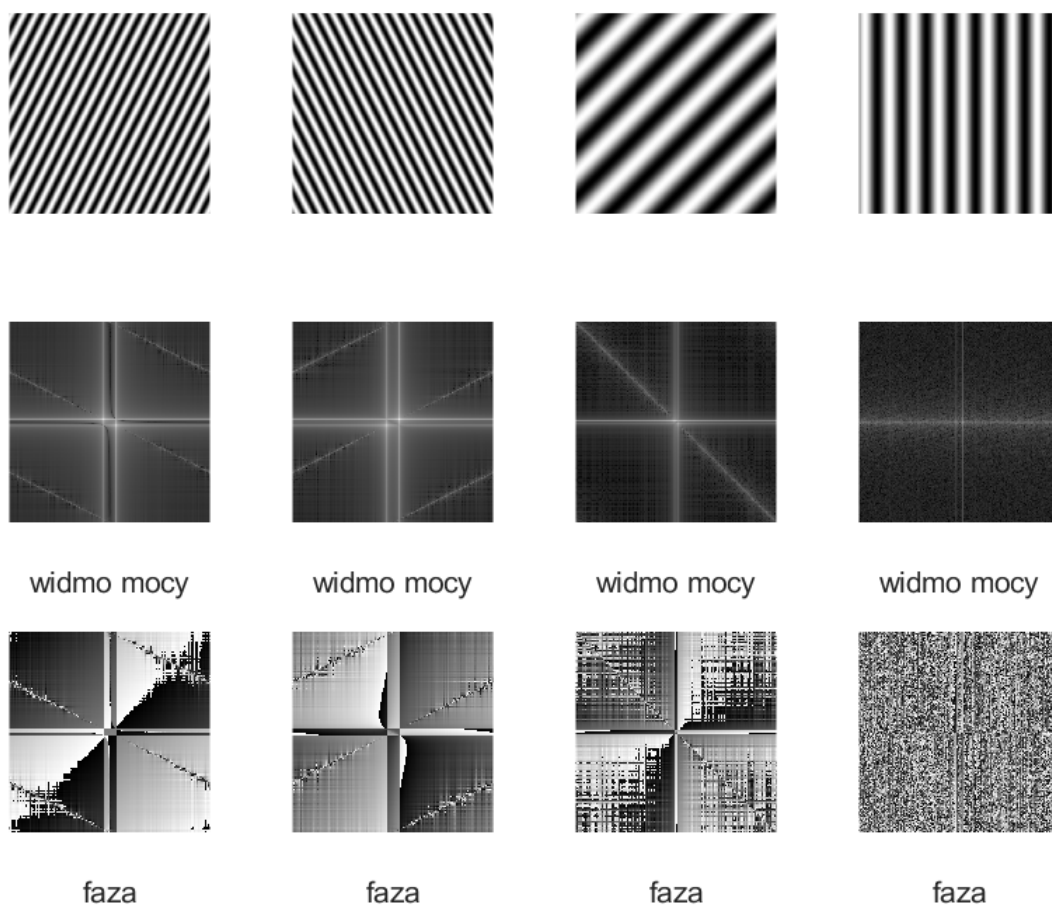
#### 4. zestawy równoległych linii o różnym nachyleniu



Rysunek 4: zestaw równoległych linii o różnym nachyleniu

Za pomocą linii możemy zaobserwować wpływ rotacji na widmo mocy. Widmo nieco przypomina to z trójkąta lecz znacznie uboższe ze względu na występowanie tylko linii do siebie równoległych. Stąd też nie obserwujemy żadnych przecięć na widmie.

5. sinusoidy o różnej częstotliwości i kierunkach.



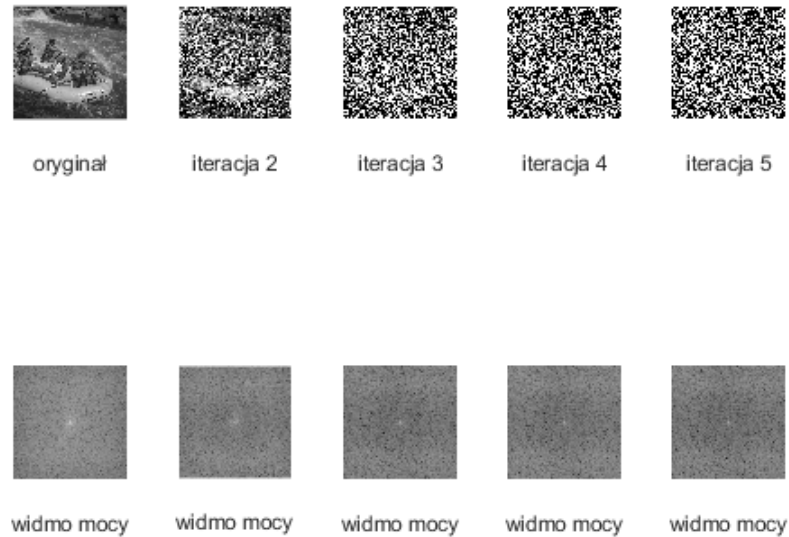
Rysunek 5: sinusoidy o różnej częstotliwości i kierunkach

Przy sinusoidach widzimy, że im większa częstotliwość lini oraz większy kontrast pomiędzy odcinkami, to będzie to bardziej wyraźne na widmie. Wraz z obrotem uwidoczniają się linie poprzeczne których nie widać na czwartym obrazie.

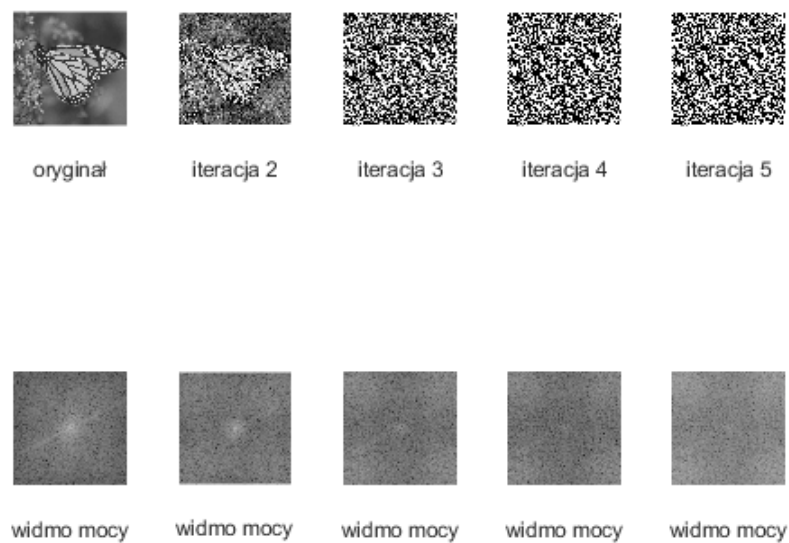
## 2 Wpływ filtracji w dziedzinie przestrzennej na charakterystyki częstotliwościowe

(Zaliczone w trakcie zajęć)

### 1. filtracja górnoprzepustowa



Rysunek 6: filtracja górnoprzepustowa



Rysunek 7: filtracja górnoprzepustowa

## 2. filtracja dolnoprzepustowa

Wraz ze wzrostem wyostżenia na obrazie powstawał większy szum co powodował widoczne zakłócenia w widmie mocy. Zaczął również zanikać wyraźny środek widma, który odpowiada za najniższe częstotliwości. Przy ostatecznej iteracji obraz nie ma niemalże nic wspólnego z oryginałem i jego widmo ukazuje bardzo mocne szumy, lecz nic ciekawego poza tym.



oryginał



iteracja 1



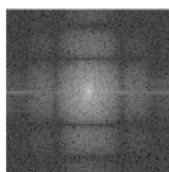
iteracja 2



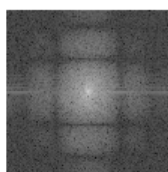
iteracja 5



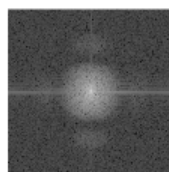
iteracja 10



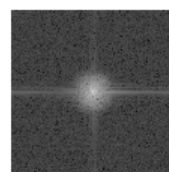
widmo mocy



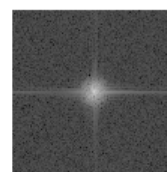
widmo mocy



widmo mocy



widmo mocy



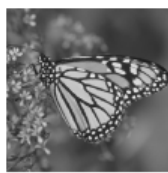
widmo mocy

Rysunek 8: filtracja dolnoprzepustowa





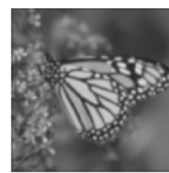
oryginał



iteracja 1



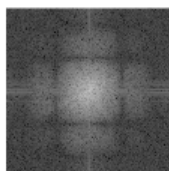
iteracja 2



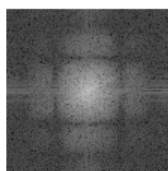
iteracja 5



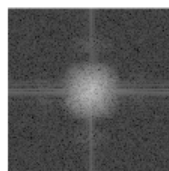
iteracja 10



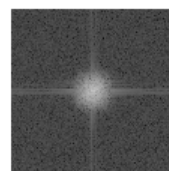
widmo mocy



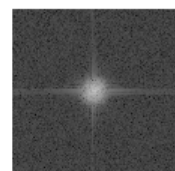
widmo mocy



widmo mocy



widmo mocy



widmo mocy

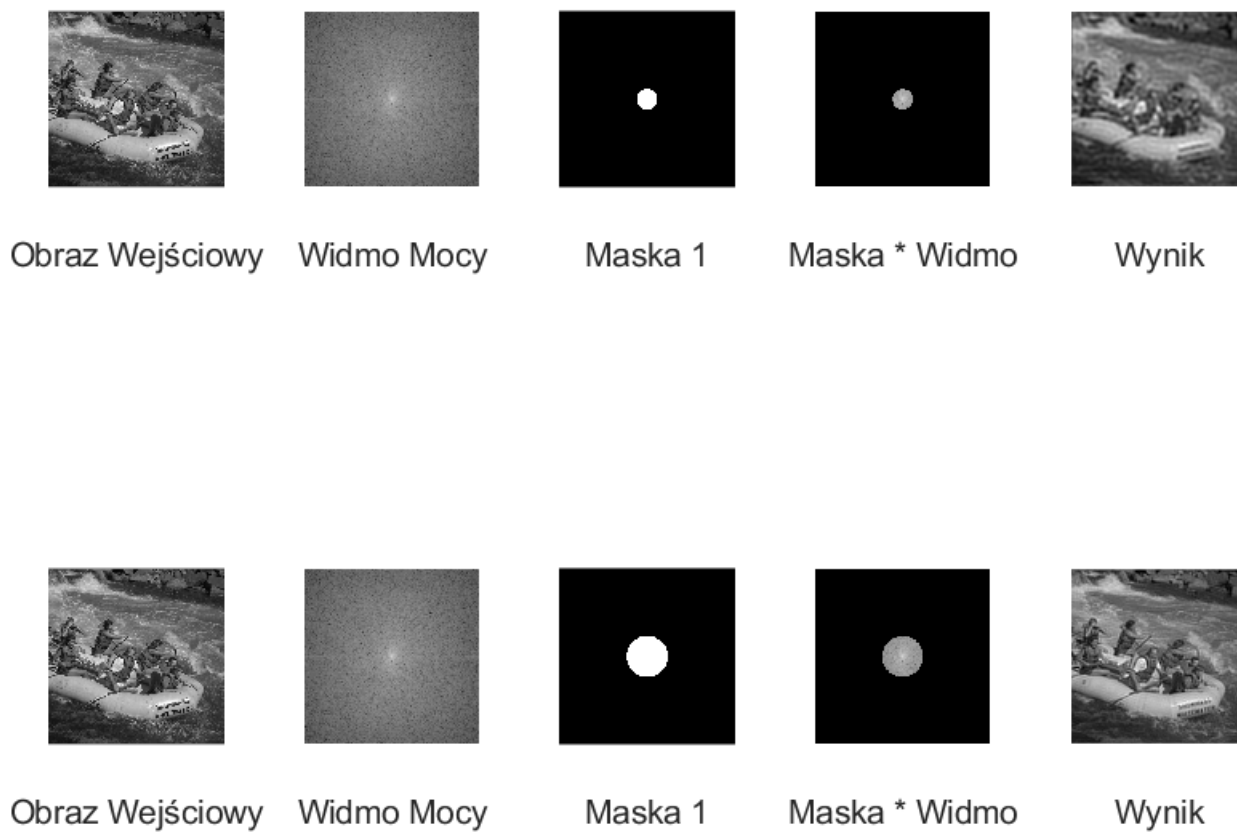
Rysunek 9: filtracja dolnoprzepustowa

Wraz ze wzrostem rozmycia obrazu widać znaczące zmniejszenie się skupiska widma mocy na środku. Jest one mniej zaszumione, co wynika z wyłagodzenia się przejść między krawędziami obrazu. Widmo mocy jest bardziej skupione w środku, a dookoła występuje mniej zmian.

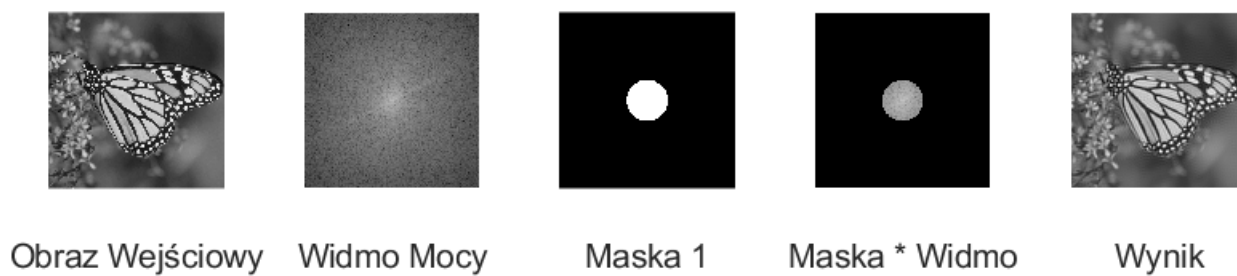
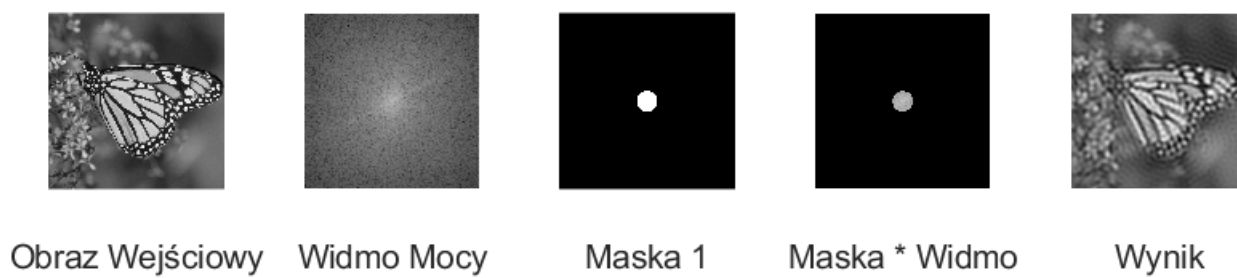
### 3 Filtracja w dziedzinie częstotliwości

(Zaliczone w trakcie zajęć)

#### 1. Filtracja dolnoprzepustowa



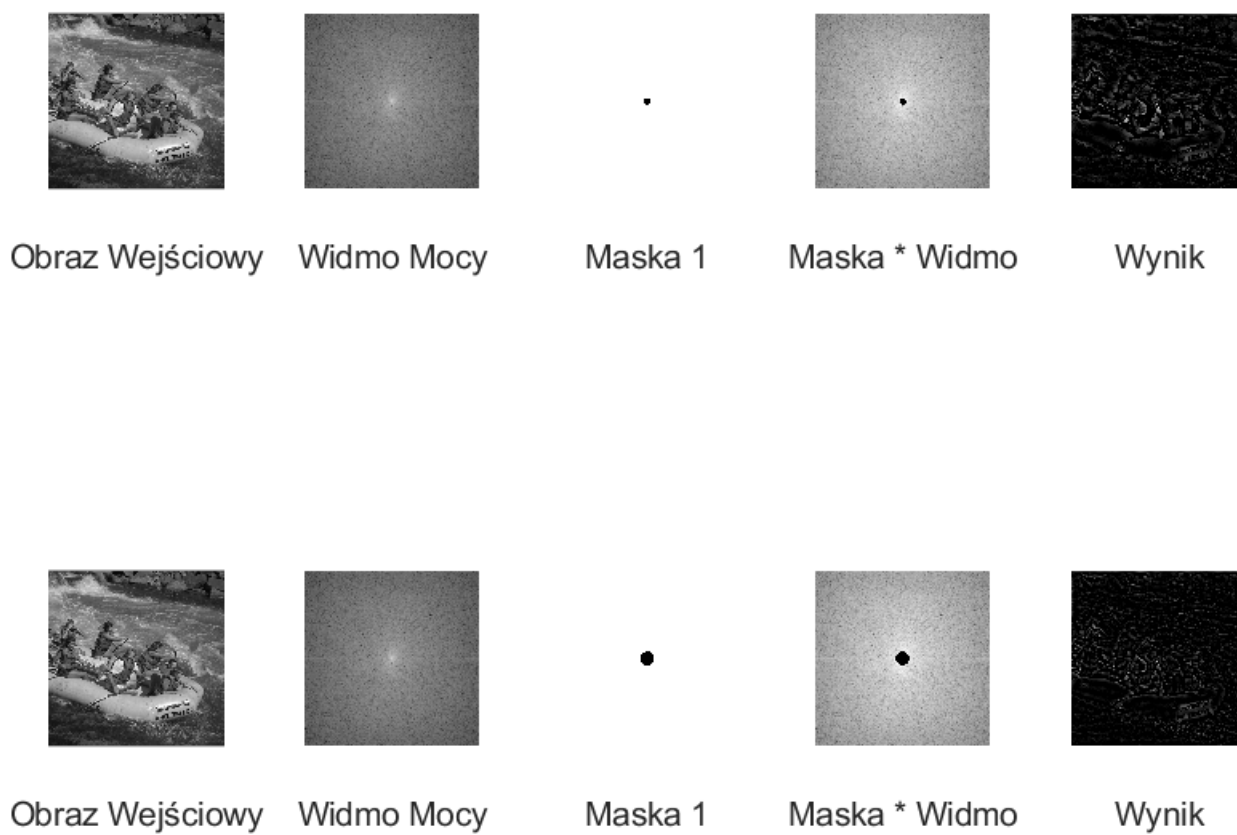
Rysunek 10: filtracja dolnoprzepustowa



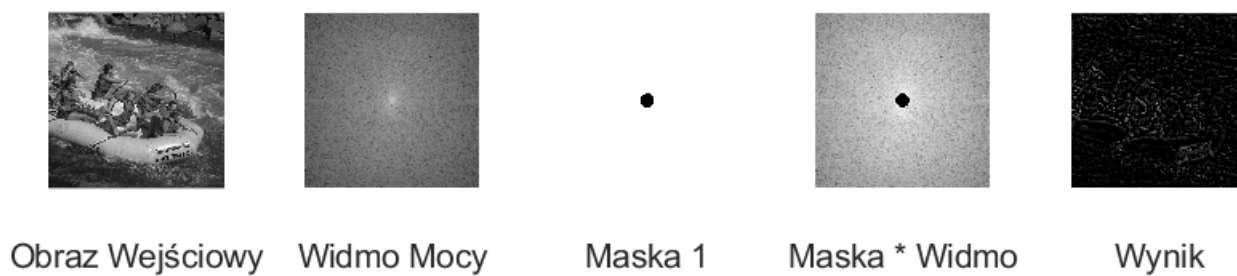
Rysunek 11: filtracja dolnoprzepustowa

Zastosowanie większej maski skutkuje mniejszym rozmyciem obrazu. Przy wykorzystaniu takiej maski bardzo widoczne jest efekt pierścieni, którego można się pozbyć za pomocą użycia filtru dolnoprzepustowego na samej masce.

## 2. Filtra gónoprzepustowa



Rysunek 12: filtracja gónoprzepustowa



Rysunek 13: filtracja górnoprzepustowa

Na obrazach możemy zaobserwować w widoczności kontur wraz ze zmianą maski, gdzie czarny środek jest mniejszy. Mniejsze pole zachowuje na wyniku większą ilość mocnych zmian kontrastu, a więc krawędzi w obrazie. Dzięki temu na wyniku znacznie bardziej widoczne są krawędzie.

Użyty kod do zadań został zaimplementowany w języku MATLAB.

### Przygotowanie obrazów oraz ich zapis

```
1  clc ;
2
3  %% Obracanie  Obraz w
4
5  t1=imread( 'triangle.png' );
6  j=0;
7  for  i=1:25:91
8  j=j+1;
9  t1r=imrotate(t1,i-1,'crop' );
10 L1=fftshift(fft2(t1r));
11 subplot(2,4,j)
12 s1=strcat(' k t=',num2str(i-1));
13 imshow(t1r,[])
14 % imwrite(t1r,sprintf("triangle_rotated_%d.png",j));
15 title(s1);
16 end
17 subplot(2,4,1)
18 ylabel(' Tr j k ty rotacja ');
19 sc= int8(-150 + 250.*rand(4,2));
20 for  i=1:4
21 j=j+1;
22 t1t=imtranslate(t1,sc(i,:));
23 L1=fftshift(fft2(t1t));
24 subplot(2,4,j)
25 imshow(t1t,[])
26 imwrite(t1t,sprintf("triangle_translated_%d.png",j));
27 end
28 subplot(2,4,5)
29 ylabel(' Tr j k ty translacja ');
```

przygotowanie\_obrazow.m

## kod do zadania 2

```

1  clc;
2  clear;
3
4  %%
5  % [file ,path]=uigetfile({'*.png';'*.BMP'}, 'Select an image');
6  % img = imread([path, file]);
7
8  % result = fft2_and_shift(img,'widmo');
9  % imshow(result,[]);
10
11 error = plotting('lines','lines_rotated');
12 error = plotting('circle','circle');
13 error = plotting('triangle','triangle_rotated');
14 error = plotting('triangle','triangle_translated');
15 error = plotting('triangle_grey','triangle_rotated');
16 error = plotting('triangle_grey','triangle_translated');
17 error = plotting('sinus','sinus');
18
19
20
21
22
23 %% Functions
24
25 function error = plotting(folder_name,figure_name)
26     figure;
27     k = 1;
28     label_info = "";
29     for i=1:1:3
30         for j=1:1:4
31             file_name = append(folder_name,'/',figure_name,sprintf('_%d.png',j));
32             if isfile(file_name)
33                 % If the file exists, display it.
34                 image_data = imread(file_name);
35                 subplot(3,4,k);
36                 k = 1 + k;
37                 switch i
38                     case 1
39                         imshow(image_data);
40                     case 2
41                         image_data = fft2_and_shift(image_data,'widmo');
42                         label_info = "widmo mocy";
43                         imshow(image_data,[]);
44                     case 3
45                         image_data = fft2_and_shift(image_data,'faza');
46                         label_info = "faza";
47                         imshow(image_data,[]);
48                 end
49                 xlabel(label_info);
50                 error = 0;
51             else
52                 % Print alert for those files that don't exist.
53                 fprintf('File not found: "%s".\n', file_name)
54                 error = 1;
55             end
56         end
57     end
58 end

```

### kod do zadania 3

```

1  clc;
2  clear;
3
4  %%
5  % [file ,path]=uigetfile({'*.BMP'; '*.png'}, 'Select an image');
6  % img = imread([path, file]);
7
8  % error = img_prepare(img,10,"motyl","blur");
9  % error = img_prepare(img,10,"motyl","sharpen");
10
11 % error = img_prepare(img,10,"boat","blur");
12 % error = img_prepare(img,10,"boat","sharpen");
13
14 error = plotting('motyl','motyl','blur');
15 error = plotting('motyl','motyl','sharpen');
16 error = plotting('boat','boat','blur');
17 error = plotting('boat','boat','sharpen');
18
19
20
21
22
23
24 %% Functions
25
26 function error = img_prepare (img,iterations ,folder_name ,mode)
27     k = 1;
28     high_pass_kernel = [-1 -1 -1;-1 9 -1 ; -1 -1 -1];
29     img = rgb2gray(img);
30     imwrite(img,append(folder_name,"/",folder_name,".png"));
31     if mode == "blur"
32         windowSize = 5;
33         avg3 = ones(windowSize) / windowSize^2;
34     elseif mode == "sharpen"
35         avg3 = high_pass_kernel;
36     else
37         disp('ERROR! WRONG MODE INPUT!!!');
38         return;
39     end
40     for i=1:iterations
41         if mode == "blur"
42             img = imfilter(img,avg3,'conv');
43             imshow(img,[]);
44         else
45             img = imfilter(img,avg3);
46             % img = imsharpen(img);
47             imshow(img,[]);
48         end
49         if (i==1||i==2||i==5||i==10) && mode == "blur"
50             k = k+1;
51             imwrite(img,append(folder_name,"/",folder_name,'_',mode,sprintf("_%d.png",k
52             )));
53         elseif mode == "sharpen"
54             imwrite(img,append(folder_name,"/",folder_name,'_',mode,sprintf("_%d.png",i
55             )));
56         end
57         error = 1;
58     end
59 end
60
61 function error = plotting(folder_name ,figure_name ,mode)
62     figure;

```



```

63 k =1;
64 blur_iteration = [1,2,5,10];
65 label_info = "";
66 for i=1:1:2
67     for j=1:1:5
68         if k == 1
69             file_name = append(folder_name, '/', figure_name, '.png');
70         else
71             file_name = append(folder_name, '/', figure_name, '_', mode, sprintf('_%d.
                png', j));
72         end
73         if isfile(file_name)
74             % If the file exists, display it.
75             image_data = imread(file_name);
76             subplot(2,5,k);
77             k = 1 + k;
78             switch i
79                 case 1
80                     imshow(image_data);
81                     if mode == "blur" && j ~= 1
82                         label_info = append("iteracja ", sprintf("%d",
                            blur_iteration(j-1)));
83                     elseif j == 1
84                         label_info = "orygina Ć";
85                     else
86                         label_info = append("iteracja ", sprintf("%d", j));
87                     end
88                 case 2
89                     image_data = fft2_and_shift(image_data, 'widmo');
90                     label_info = "widmo mocy";
91                     imshow(image_data, []);
92             end
93             xlabel(label_info);
94             error = 0;
95         else
96             % Print alert for those files that don't exist.
97             fprintf('File not found: "%s".\n', file_name)
98             error = 1;
99         end
100     end
101 end
102 end

```

zad3.m

kod do zadanie 4

```

1 clc;
2 clear;
3
4 %%
5 [file, path] = uigetfile({'*.BMP'; '*.png'}, 'Select an image');
6 img = imread([path, file]);
7 % img_1 = imread('boat.bmp');
8 % img_2 = imread("motyl.bmp");
9 [X,Y,DIM] = size(img);
10 if DIM > 1
11     img = rgb2gray(img);
12 end
13
14 [low_1, high_1, widmo_1] = low_high_pass_filter(img, 1);
15 [low_2, high_2, widmo_2] = low_high_pass_filter(img, 0.5);
16
17
18 inverted_img_high_1 = uint8(real(ifft2(ifftshift(widmo_1.*high_1))));
19 inverted_img_low_1 = uint8(real(ifft2(ifftshift(widmo_1.*low_1))));
20 inverted_img_high_2 = uint8(real(ifft2(ifftshift(widmo_2.*high_2))));

```

```

21 inverted_img_low_2 = uint8(real(ifft2(ifftshift(widmo_2.*low_2))));
22 widmo_1 = log(1+abs(widmo_1));
23 widmo_2 = log(1+abs(widmo_2));
24
25 img_pack_1 = {img, widmo_1, low_1, widmo_1.*low_1, inverted_img_low_1, widmo_2, low_2, widmo_2
    .*low_2, inverted_img_low_2};
26 img_pack_2 = {img, widmo_1, high_1, widmo_1.*high_1, inverted_img_high_1, widmo_2, high_2,
    widmo_2.*high_2, inverted_img_high_2};
27
28 error = plot_imgs(img_pack_1);
29 error = plot_imgs(img_pack_2);
30
31
32
33 %% Functions
34
35 function [lowpass_img, highpass_img, widmo] = low_high_pass_filter(img, scale)
36     widmo = fftshift(fft2(img));
37     [X,Y] = size(img);
38     x_diff = 0:(X-1);
39     y_diff = 0:(Y-1);
40     idx = find(x_diff > X/2);
41     x_diff(idx) = x_diff(idx) - X;
42     idy = find(y_diff > Y/2);
43     y_diff(idy) = y_diff(idy) - Y;
44     [V, U] = meshgrid(y_diff, x_diff);
45     D0=30;
46     D = scale*sqrt(U.^2 + V.^2);
47     lowpass_img =abs(ifftshift(double(D <=D0)));
48     % Highpass
49     D0=10;
50     H =ifftshift(double(D <=D0));
51     highpass_img=1-H;
52 end
53
54 %% Plotting
55 function error = plot_imgs(img_pack)
56 figure;
57 subplot(2,5,1);
58 imshow(img_pack{1},[]);
59 xlabel("Obraz Wej Żciowy");
60 subplot(2,5,2);
61 imshow(img_pack{2},[]);
62 xlabel("Widmo Moc");
63 subplot(2,5,3);
64 imshow(img_pack{3},[]);
65 xlabel("Maska 1");
66 subplot(2,5,4);
67 imshow(img_pack{4},[]);
68 xlabel("Maska * Widmo");
69 subplot(2,5,5);
70 imshow(img_pack{5},[]);
71 xlabel("Wynik");
72 subplot(2,5,6);
73 imshow(img_pack{1},[]);
74 xlabel("Obraz Wej Żciowy");
75 subplot(2,5,7);
76 imshow(img_pack{6},[]);
77 xlabel("Widmo Moc");
78 subplot(2,5,8);
79 imshow(img_pack{7},[]);
80 xlabel("Maska 1");
81 subplot(2,5,9);
82 imshow(img_pack{8},[]);
83 xlabel("Maska * Widmo");

```

```
84 subplot(2,5,10);
85 imshow(img_pack{9},[]);
86 xlabel("Wynik");
87 error =1;
88 end
```

zad4.m

funkcja pomocnicza

```
1 function result = fft2_and_shift(img,mode)
2     [X,Y,DIM] = size(img);
3     if DIM > 1
4         img = rgb2gray(img);
5     end
6     img = double(img);
7     if mode == "widmo"
8         result = log(1+abs(fftshift(fft2(img))));
9     elseif mode == "faza"
10        result = angle(fftshift(fft2(img)));
11    else
12        disp('ERROR! ');
13    end
14 end
```

fft2\_and\_shift.m