



Politechnika
Wrocławska

SPRAWOZDANIE Z LABORATORIUM PODSTAW PRZETWARZANIA OBRAZÓW	
Sprawozdanie nr 1	Temat ćwiczenia:
Data oddania sprawozdania: 24.01.2023r	Wykorzystanie programu Matlab do filtracji obrazów, implementacja filtrów o różnych parametrach
Imię i nazwisko: Filip Ferenc 263578	
Prowadzący : Dr inż. Jan Mazur	Kod grupy: SI0301G

Spis treści

1 Cel ćwiczenia	3
2 Wstęp teoretyczny	3
2.1 Odpowiedź impulsowa filtru	3
2.1.1 Przykład - wyznaczenie odpowiedzi impulsowej dla filtru idealnego . . .	3
2.1.2 Funkcja mask	3
2.1.3 Opis działania funkcji dlt	3
2.2 Okno przestrzenne	5
2.3 Rodzaje filtrów	7
3 Ćwiczenie 1: Usunięcie wybranych harmoniczných z sygnału	9
3.1 Metoda pierwsza - filtra w kształcie walca	10
3.1.1 Filtr dolnoprzepustowy	10
3.1.2 Filtr górnoprzepustowy	14
3.2 Metoda druga - filtr prostokątny	16
3.2.1 Filtr dolnoprzepustowy	16
3.2.2 Filtr górnoprzepustowy	17
4 Ćwiczenie 2: Filtracja szarego obrazka	19
4.1 Filtr dolnoprzepustowy	20
4.2 Filtr górnoprzepustowy	21
4.3 Filtr pasmowoprzepustowy i pasmowozaporowy	22
5 Ćwiczenie 3: Filtracja kolorowego obrazka	23
5.1 Filtr dolnoprzepustowy	24
5.2 Filtr górnoprzepustowy	25
6 Wnioski ogólne	26

1 Cel ćwiczenia

Zapoznanie się z podstawami filtracji obrazków w środowisku Matlab. Poznanie różnych okien czasowych. Zapoznanie się z metodami implementacji filtrów. Poznanie różnych rodzajów filtrów i okien przestrzennych wykorzystywanych do tworzenia filtrów w dziedzinie przetwarzania obrazów.

2 Wstęp teoretyczny

2.1 Odpowiedź impulsowa filtru

Odpowiedź impulsowa [1] filtru odnosi się do reakcji filtru na impuls jednostkowy. Impuls jednostkowy to sygnał, który ma wartość 1 tylko w jednym punkcie czasowym i 0 wszędzie indziej. Wyznaczenie odpowiedzi impulsowej jest bardzo ważnym krokiem w projektowaniu filtrów.

2.1.1 Przykład - wyznaczenie odpowiedzi impulsowej dla filtru idealnego

$$h_d(x_1, x_2) = FT^{-1}\{H_d(\omega_1, \omega_2)\} = \frac{\omega_c}{2\pi\sqrt{x_1^2 + x_2^2}} J_1\left(\omega_c\sqrt{x_1^2 + x_2^2}\right)$$

Rysunek 1: Enter Caption

Odpowiedzi impulsowe filtrów 2D można stworzyć przy pomocy poniższych funkcji:

2.1.2 Funkcja mask

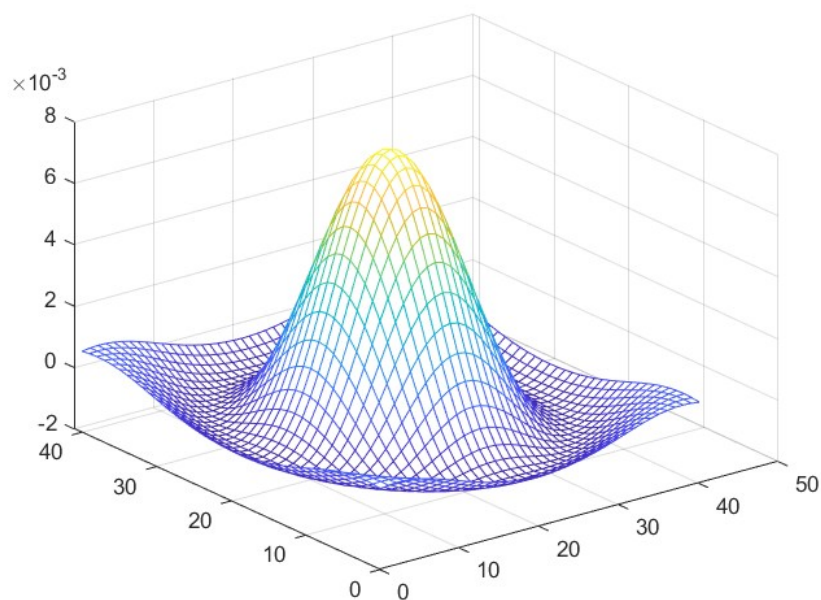
Funkcja mask służy do generowania odpowiedzi impulsowej filtru. Domyślnie używana jest funkcja Bessela (besselj). Można jednak używać różnych rodzajów okien, opisanych w późniejszym punkcie. Na przykład, można użyć okna Hamminga, Hanninga, Blackmana itp., w zależności od wymagań.

```
1 function y=mask(cofffreq,rank) %coffreq - im nizszy tym
    bardziej rozmyty filtr, rank - rzad filtru.
2 rofs = dzd(rank);
3 y = cofffreq*besselj(1,cofffreq*rofs)./(2*pi*rofs);
4 end
```

2.1.3 Opis działania funkcji dlt

Funkcja dlt tworzy deltę kroneckera, dzięki której można stworzyć z filtru dolnoprzepustowego filtr górnoprzepustowy (poprzez odjęcie od delty wyniku funkcji mask).

```
1 function y=dlt(number)
2 orig=ceil(number/2);
3 y = zeros(number);
4 y(orig,orig)=1;
5 end
```



Rysunek 2: Odpowiedź impulsowa wygenerowana przy pomocy funkcji mask.

2.2 Okno przestrzenne

Okno przestrzenne zachowuje się tak jak okno czasowe[2] tylko w przypadku sygnału 2D - jest to funkcja opisująca sposób pobierania próbek z sygnału. Okno wykorzystywane jest w celu 'wycięcia' i zmodyfikowania tylko tej części sygnału która nas interesuje. Domyślnym oknem wykorzystywanym w filtracji jest okno prostokątne. Istnieje wiele różnych rodzajów okien używanych w kontekście przetwarzania sygnałów, obrazów czy danych. Okna te są stosowane w procesie filtracji, a ich celem jest wprowadzenie pewnych właściwości do sygnału wejściowego lub danych. Poniżej przedstawiam kilka popularnych rodzajów okien:

1. Okno prostokątne

- Proste okno o stałej amplitudzie na całej długości.
- Najprostsza forma okna, ale może powodować efekt przejścia sygnału, co prowadzi do zjawiska "wycieku widma".

2. Okno Hamminga

- Charakteryzuje się wygładzeniem brzegów, co pomaga zmniejszyć efekt wycieku widma w porównaniu do okna prostokątnego.
- Wprowadza pewne zmniejszenie amplitudy sygnału na krańcach okna.

3. Okno Hanninga

- Podobne do okna Hamminga, ale ma bardziej wygładzone brzegi
- Efektywne w redukcji wycieku widma.

4. Okno Blackmana

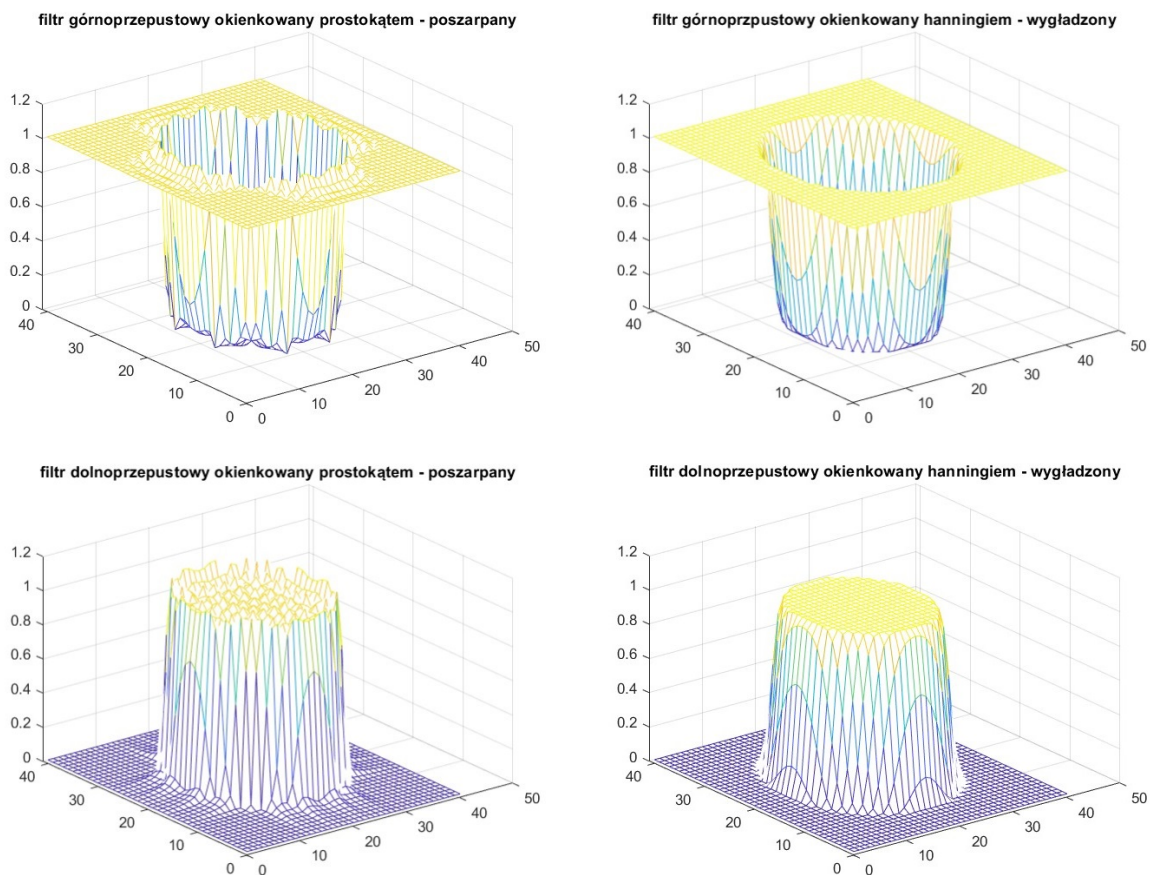
- Zapewnia jeszcze lepszą redukcję wycieku widma niż okna Hamminga i Hanninga, ale kosztem szerszego obszaru brzegowego.

5. Okno Kaiser-Bessela

- Zparametryzowane okno, które pozwala na kontrolę szerokości i poziomu tłumienia brzegów.
- Bardzo elastyczne, ale wymaga dobrania odpowiednich parametrów.

6. Okno Gaussa

- Charakteryzuje się rozkładem Gaussa, co oznacza, że ma gładkie przejście od środka do brzegów.
- Wykorzystywane tam, gdzie istotne jest zachowanie informacji w centrum sygnału.

Przykład zastosowania różnych okien w filtrze:

Rysunek 3: Filtry dolno- i górno- przepustowy z oknem Hanninga i Prostokątnym.

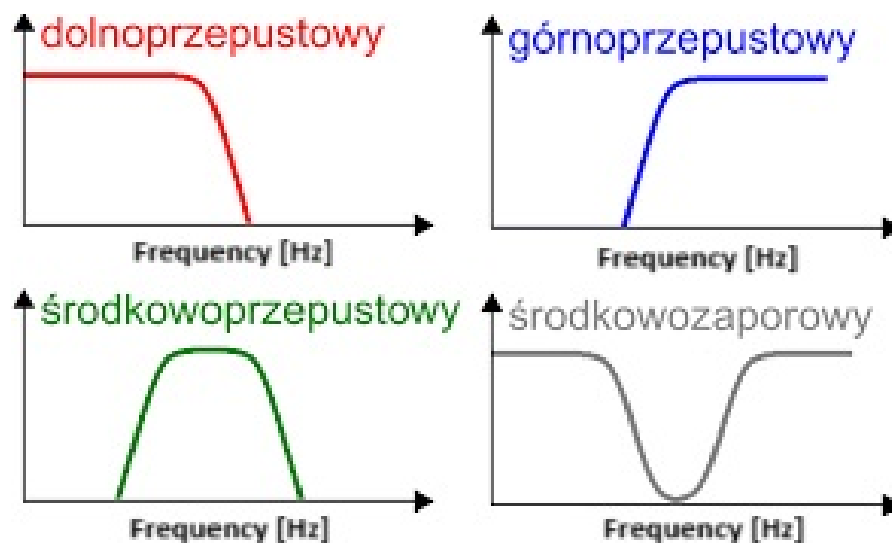
Jak można zauważyć, dzięki oknu Hanninga filtry stały się bardziej wygładzone. Spowodowane jest to tym, że filtr Hanninga ma bardziej gładkie przejścia, co redukuje efekt wycieku częstotliwości. Filtr prostokątny za to ma prostszy kształt, co sprawia, że jest bardziej skuteczny w pewnych zastosowaniach, ale może prowadzić do szerszego wycieku częstotliwości. Wybór między nimi zależy od konkretnego przypadku i celu analizy sygnału.

2.3 Rodzaje filtrów

Istnieje wiele rodzajów filtrów stosowanych w przetwarzaniu sygnałów i obrazów, a ich wybór zależy od konkretnego zadania. Poniżej przedstawiono podstawowe rodzaje filtrów:

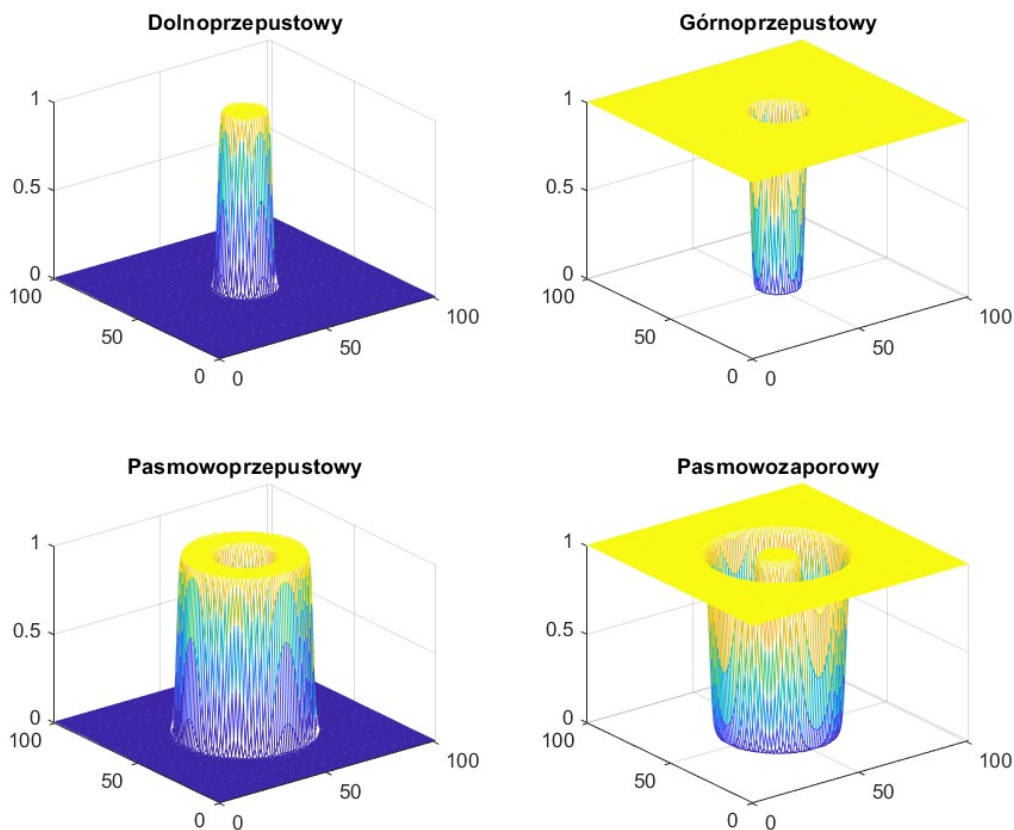
- dolnoprzepustowy
- górnoprzepustowy
- pasmowoprzepustowy (środkowoprzepustowy)
- pasmowozaporowy (środkowozaporowy)

Wykresy filtrów 1D:



Rysunek 4: Rodzaje filtrów 1D.

Jak można zauważyć na powyższym rysunku, filtr dolnoprzepustowy przepuszcza niskie częstotliwości, eliminuje wysokie, filtr górnoprzepustowy przepuszcza wysokie częstotliwości, filtr pasmowoprzepustowy przepuszcza sygnały w określonym pasmie, a filtr pasmowozaporowy blokuje sygnały w określonym pasmie częstotliwości.

Wykresy filtrów 2D:**Rysunek 5:** Rodzaje filtrów 2D.

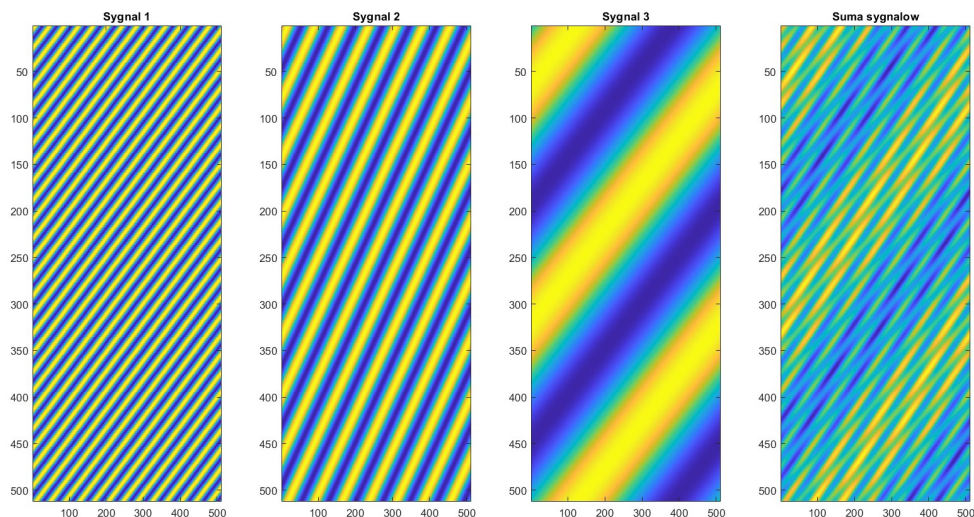
Na powyższym rysunku można zauważyć, że filtry mają kształt walca. Nie jest to jedyny kształt filtra 2D, mogą przyjmować one różne kształty: prostopadłościanu lub graniastosłupów o podstawach z różną ilością kątów. Filtr dolnoprzepustowy wygenerowano poprzez obliczenie transformaty Fouriera odpowiedzi impulsowej wygenerowanej przez funkcję mask. Filtr górnoprzepustowy został wygenerowany poprzez odjęcie od delta Dirack'a filtra dolnoprzepustowego. Filtr pasmowoprzepustowy został wygenerowany poprzez odjęcie od siebie dwóch filtrów dolnoprzepustowych, a filtr pasmowozaporowy poprzez dodanie do delta Dirack'a filtra pasmowoprzepustowego. Na wykresach wyświetlono moduł transformaty Fouriera filtrów po użyciu funkcji *fftshift()* mającej na celu zamianę ćwiartek na osiach poziomych. Dzięki takiemu działaniu łatwiej analizować filtry w programie Matlab.

3 Ćwiczenie 1: Usunięcie wybranych harmoniczných z sygnału

Pierwszym zadaniem było wygenerowanie sygnału składającego się z trzech sinusów o różnych częstotliwościach. Następnie należało przy pomocy wybranej metody usunąć wybrane składowe harmoniczne i wyświetlenie modułu sygnału o jednej częstotliwości.

Do generacji sygnału wykorzystano poniższy kod. Inicjalizowane są zmienne reprezentujące częstotliwości poszczególnych harmoniczných (f_{x1} - f_{x6}), całego sygnału (F_{sx1} , F_{sx2}) oraz okres sygnału (T_{sx1} , T_{sx2}). A także szerokość generowanego obrazka, na którym wyświetli się generowany sygnał. Następnie w pętlach generowane są sinusy: $s1$, $s2$, $s3$. Na samym końcu sinusy dodawane są do siebie i tworzony jest sygnał.

```
1  fx1 = 30;
2  fx2 = 17;
3  fx3 = 16;
4  fx4 = 11;
5  fx5 = 6;
6  fx6 = 3;
7  Fsx1 = 512;
8  Fsx2 = 512;
9  Tsx1 = 1/Fsx1;
10 Tsx2 = 1/Fsx2;
11 N1 = 512;
12 N2 = 512;
13 s = zeros(N1,N2);
14
15 for n1 = 1:N1
16     for n2 = 1:N2
17         s1(n1,n2) = (sin(2*pi*fx1*n1*Tsx1+2*pi*fx2*n2*Tsx2));
18     end
19 end
20
21 for n1 = 1:N1
22     for n2 = 1:N2
23         s2(n1,n2) = (sin(2*pi*fx3*n1*Tsx1+2*pi*fx4*n2*Tsx2));
24     end
25 end
26
27 for n1 = 1:N1
28     for n2 = 1:N2
29         s3(n1,n2) = (sin(2*pi*fx5*n1*Tsx1+2*pi*fx6*n2*Tsx2));
30     end
31 end
32
33 s=s1+s2+s3;
```



Rysunek 6: Sygnały składowe oraz sygnał po dodaniu wszystkich harmonicznych.

3.1 Metoda pierwsza - filtra w kształcie walca

Pierwszą metodą było usunięcie wybranych harmonicznych przy pomocy napisanej na wcześniejszych zajęciach funkcji *"mask"*, a także funkcji *"filter2"*.

3.1.1 Filtr dolnoprzepustowy

Implementacja:

```
1  im_fft=fft2(s);
2
3  shifted_modul = fftshift(im_fft);
4  modul = abs(shifted_modul);
5
6  h=mask(0.1,501);
7
8  ylp=filter2(h,s);
9
10 fft2_filtra = fft2(ylp);
11 fftshift_filtra = fftshift(fft2_filtra);
```

Wyjaśnienie kodu:

1. `im_fft = fft2(s);`

- Wykonuje dwuwymiarową transformatę Fouriera (FFT) na sygnale `s`.

2. `shifted_modul = fftshift(im_fft);`

- Przesuwa zerową częstotliwość sygnału do środka obrazu.

3. `modul = abs(shifted_modul);`

- Oblicza moduł zespolonych liczb zmiennoprzecinkowych uzyskanych po przesunięciu częstotliwości.

4. `h = mask(0.1, 501);`

- Wywołuje funkcję `mask` z parametrami 0.1 i 501,

5. `ylp = filter2(h, s);`

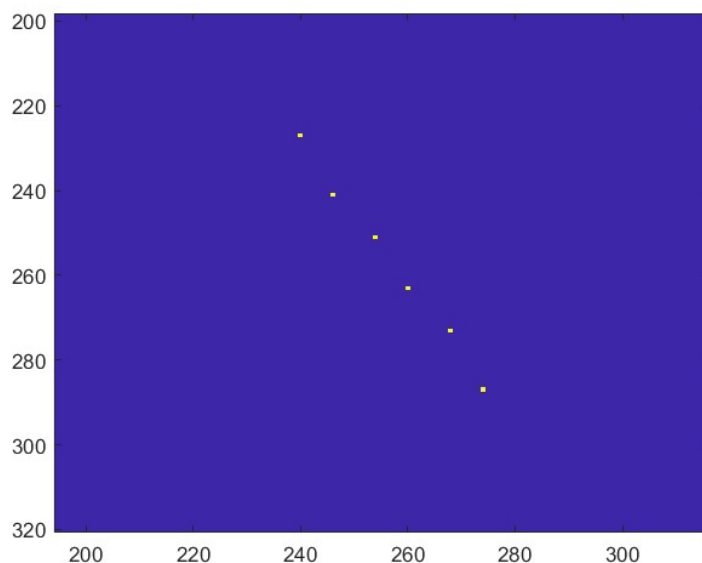
- Stosuje dwuwymiarowy filtr o współczynnikach zdefiniowanych w macierzy `h` do sygnału `s`.

6. `fft2_filtra = fft2(ylp);`

- Ponownie stosuje dwuwymiarową transformatę Fouriera na sygnale `ylp` po zastosowaniu filtracji.

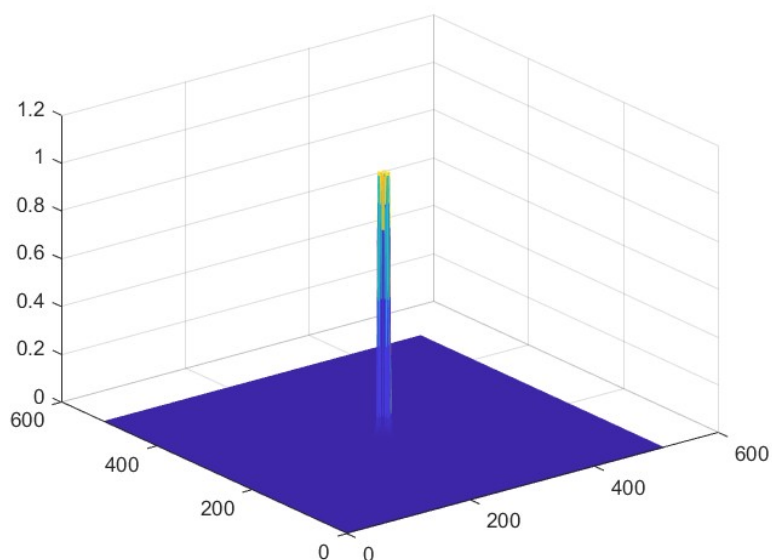
7. `fftshift_filtra = fftshift(fft2_filtra);`

- Ponownie przesuwa zerową częstotliwość sygnału do środka obrazu po zastosowaniu filtracji.

Wynik wywołania kodu:

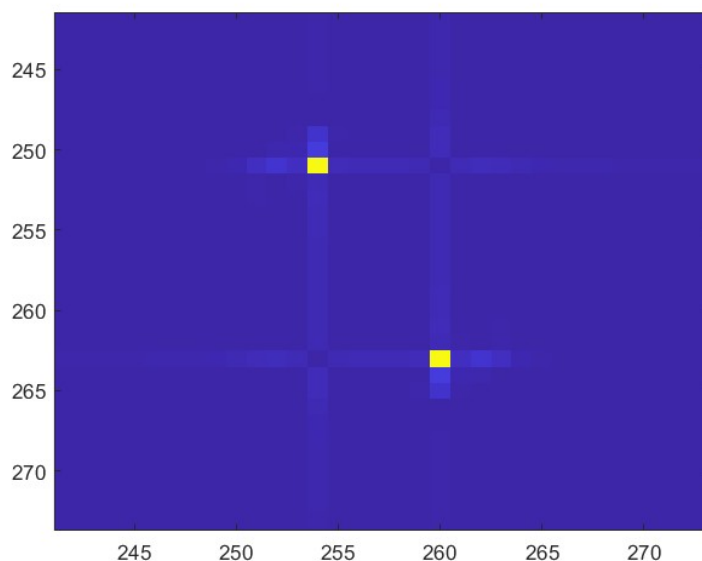
Rysunek 7: Moduł FFT sygnału po użyciu funkcji `fftshift`.

Jak można zauważyć wszystkie harmoniczne sygnału pojawiły się na obrazku. Na każdy sygnał składowy przypadają dwie kropki reprezentujące współrzędne częstotliwości ustalonej dla każdego z sygnałów (spowodowane jest to symetrią transformaty Fouriera). Dzięki funkcji ***fftshift*** można łatwiej badać transformatę fouriera sygnału, ponieważ zamienia on ćwiartki obrazu.

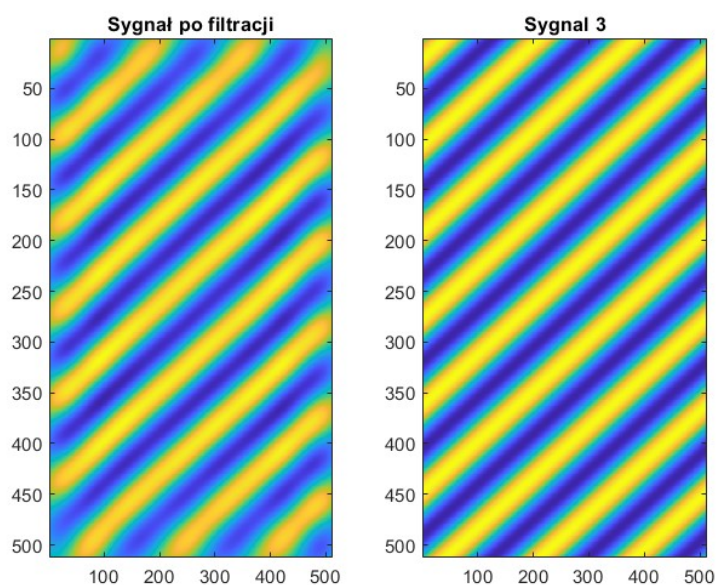


Rysunek 8: Filtr dolnoprzepustowy stworzony przez funkcję `mask`.

Na rysunku 8 zaobserwować można filtr, który wykorzystany został do odcięcia niepotrzebnych harmonicznych z sygnału. Został wyskalowany tak, aby pasmo przepustowe pokrywało się tylko z sygnałem nr 3.



Rysunek 9: Moduł FFT sygnału po filtracji.



Rysunek 10: Wynik filtracji.

Analizując rysunki 9 i 10 można zauważyć, że w module widma sygnału pozostał tylko dwa prążki (czyli jeden sinus). Widać także niewielkie szумы, które mogą być spowodowane złym dopasowaniem parametrów filtra. Mimo to, na rysunku 10 ewidentnie widać podobieństwo sygnału po filtracji i sygnału nr 3. Świadczy to o tym, że filtracja przebiegła pomyślnie.

3.1.2 Filtr górnoprzepustowy

Aby stworzyć filtr górnoprzepustowy należało dodać poniższy kod do programu:

```
1 hhp = dlt(511) - h;  
2 yhp = filter2(hhp, s);
```

Gdzie:

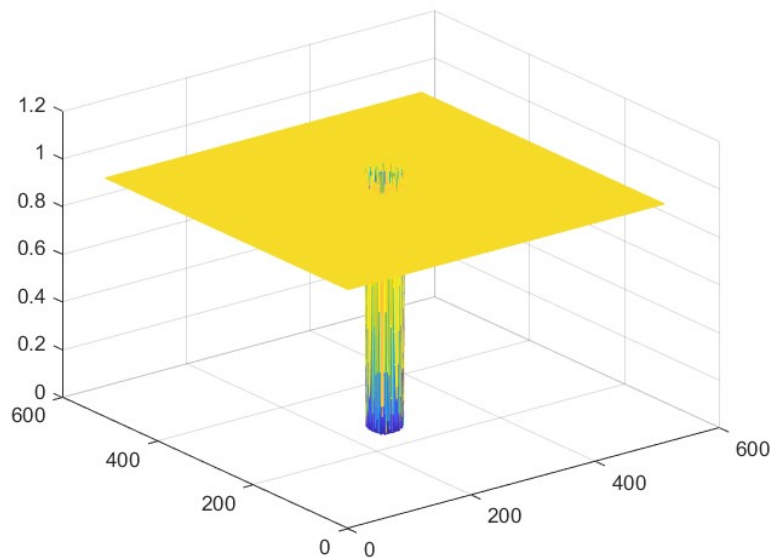
1. `hhp = dlt(511) - h;`

- Tworzy deltę kroneckera o rozmiarze 511, dzięki czemu możliwe jest stworzenie filtra górnoprzepustowego.

2. `yhp = filter2(hhp,s);`

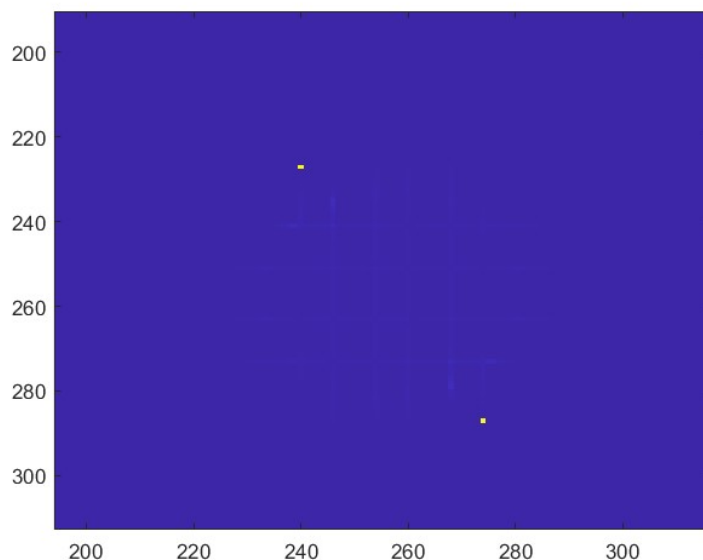
- Tworzy filtr górnoprzepustowy na sygnale 's', przy pomocy funkcji `filter2`.

Wynik wywołania kodu:

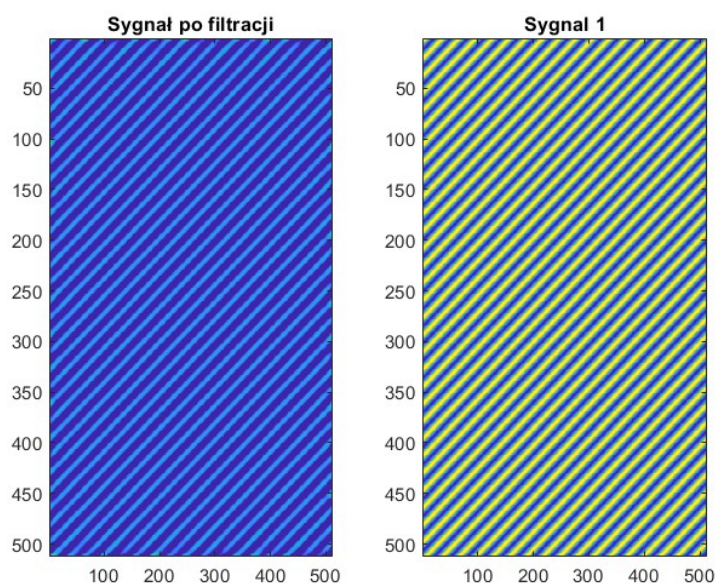


Rysunek 11: Filtr górnoprzepustowy stworzony przez odjęcie maski filtra dolnoprzepustowego od delty Kronecker'a

Widać, że w przypadku filtra górnoprzepustowego wywołanego przez funkcję `mask`, sytuacja jest odwrotna niż wcześniej. Wartość '1' przyjmują wszystkie współrzędne oprócz tych w środku filtra.



Rysunek 12: Moduł FFT sygnału po filtracji.



Rysunek 13: Wynik filtracji.

Obserwując rysunki 12 i 13 można zauważyć, że w module widma sygnału pozostał tylko dwa prążki (czyli jeden sinus). Widać także niewielkie szумы, które mogą być spowodowane złym dopasowaniem parametrów filtra, a także inny kolor sygnału. Mimo to, na rysunku 13 ewidentnie widać podobieństwo sygnału po filtracji do sygnału nr 1. Świadczy to o tym, że filtracja przebiegła pomyślnie.

3.2 Metoda druga - filtr prostokątny

Drugą metodą było przefiltrowanie sygnału przy pomocy dwóch pętli *for*. Czyli wyzerowanie przy ich użyciu obszaru który chcemy aby został wyzerowany w sygnale. Filtrowany był taki sam sygnał jak w metodzie pierwszej.

3.2.1 Filtr dolnoprzepustowy

Aby stworzyć filtr dolnoprzepustowy należało zaimplementować poniższy kod:

```
1 s = s1+s2+s3;
2
3 h_img = fft2(s);
4 inv_h = ifft2(h_img);
5 sh_img = fftshift(h_img);
6
7 sh1_img = sh_img;
8
9 shh = zeros(512, 512, 3);
10
11 for n1 = 250 : 265
12     for n2 = 250 : 265
13         shh(n1,n2) = sh1_img(n1, n2);
14     end
15 end
16
17 sh2_img = abs(fftshift(sh1_img));
18 inverted = ifft(sh2_img);
```

Gdzie:

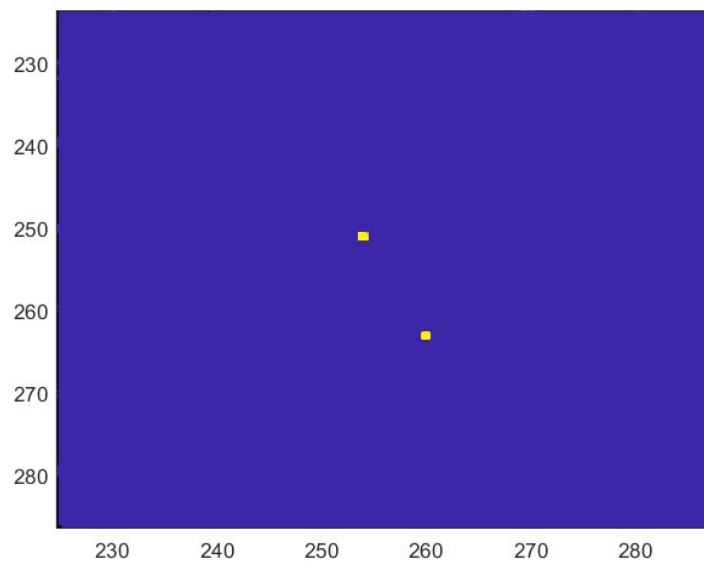
1. `h_img = fft2(s);`
 - Przeprowadza dwuwymiarową transformatę Fouriera.
2. `inv_h = ifft2(h_img);`
 - Przeprowadza odwrotną dwuwymiarową transformatę Fouriera.
3. `sh_img = fftshift(h_img);`
 - Przesuwa wynik transformaty Fouriera `h_img` w taki sposób, aby niskie częstotliwości były w centrum obrazu, a wysokie na zewnątrz.
4. `shh = zeros(512, 512, 3);`
 - Inicjalizuje trójwymiarową macierz zerową o wymiarach 512x512.
5. `shh(n1,n2) = sh1_img(n1, n2);`
 - Przypisuje wartość elementu o indeksach `(n1, n2)` z `sh1_img` do odpowiadającego elementu w macierzy `shh`.

6. `sh2_img = abs(fftshift(sh1_img));`

- Przeprowadza przesuniętą transformatę Fouriera na `sh1_img`, a następnie oblicza wartości bezwzględne wyniku. Wynik przypisywany jest do zmiennej `sh2_img`.

7. `inverted = ifft(sh2_img);`

- Przeprowadza odwrotną transformatę Fouriera na `sh2_img` i przypisuje wynik do zmiennej `inverted`.



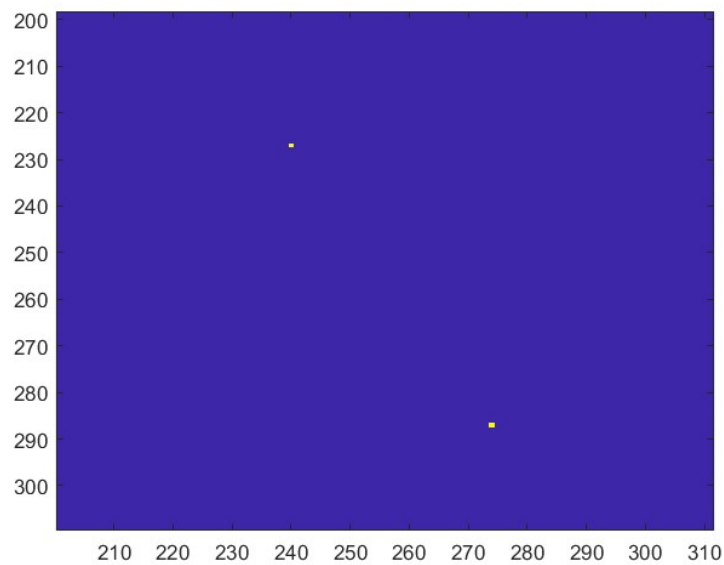
Rysunek 14: Moduł FFT sygnału po filtracji.

3.2.2 Filtr górnoprzepustowy

Aby stworzyć filtr górnoprzepustowy należało zaimplementować poniższy kod:

```
1 h_img = fft2(s);
2 inv_h = ifft2(h_img);
3 sh_img = fftshift(h_img)
4
5 sh1_img = sh_img;
6
7 for n1 = 240:280
8     for n2 = 245:280
9         sh1_img(n1, n2) = 0;
10    end
11 end
12
13 sh2_img = abs(fftshift(sh1_img));
14 inverted = ifft(sh2_img);
```

Inicjalizacja filtru górnoprzepustowego jest bardzo podobna do inicjalizacja dla filtru dolnoprzepustowego, dlatego nie została wyjaśniona.



Rysunek 15: Moduł FFT sygnału po filtracji.

Jak widać, tak jak w metodzie pierwszej filtracja przebiegła pomyślnie i niechciane harmoniczne zostały odcięte. Wybór metody zależy od konkretnego zastosowania, założeń i wymagań projektowych filtru.

4 Ćwiczenie 2: Filtracja szarego obrazka

Kolejnym zadaniem była filtracja szarego obrazka. Tak jak w poprzednim punkcie do wykonania zadania wykorzystano funkcje *mask* i *dlt*.



Rysunek 16: Oryginalny obrazek.



Rysunek 17: Obrazek w skali szarości.

4.1 Filtr dolnoprzepustowy

```
1 h=mask(0.1*pi,41);  
2  
3 x = imread('Lenna.png');  
4 xg = rgb2gray(x);  
5  
6 ylp=filter2(h,xg);
```

Inicjalizacja filtru dla szarego obrazka jest bardzo podobna do tej dla sygnału z pierwszego ćwiczenia.

1. `h=mask(0.1*pi,41);`

- Wywołuje funkcję `mask` z dwoma argumentami: `0.1*pi` i `41`.

2. `x = imread('Lenna.png');`

- Wczytuje obraz o nazwie `'Lenna.png'`.

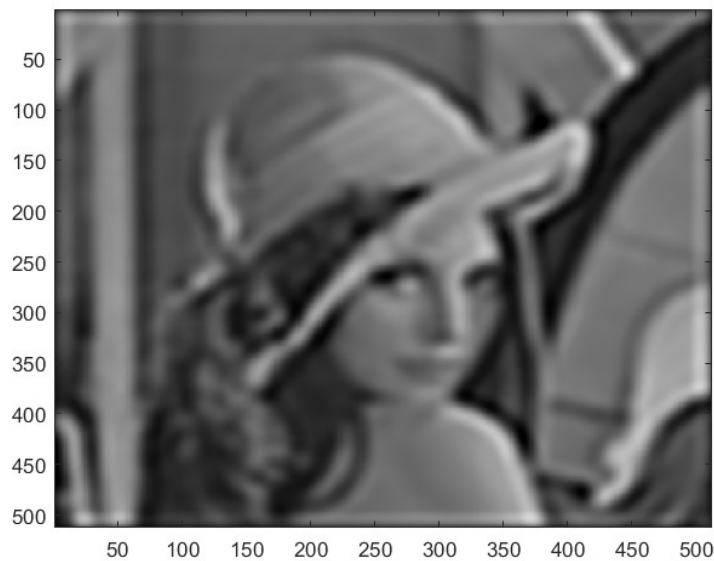
3. `xg = rgb2gray(x);`

- Konwertuje obraz `x` do skali szarości i przypisuje wynik do zmiennej `xg`.

4. `ylp = filter2(h, xg);`

- Przeprowadza dwuwymiarową filtrację obrazu `xg` za pomocą filtra `h` i przypisuje wynik do zmiennej `ylp`.

Do załadowania mapy koloru szarego użyto funkcji `colormap`.



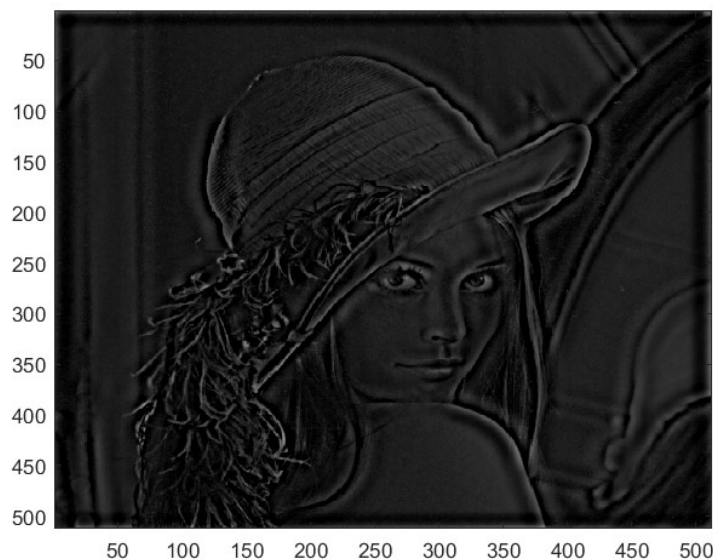
Rysunek 18: Obraz po przefiltrowaniu filtrem dolnoprzepustowym.

Analizując rysunek nr 18 można zauważyć, że obraz stał się rozmyty. Filtracja dolnoprzepustowa może spowodować wygładzenie krawędzi, redukcję szczegółów i usunięcie drobnych detali. W rezultacie obraz staje się bardziej rozmyty. Może być to korzystne w pewnych przypadkach, takich jak redukcja szumów lub poprawa ogólnego wyglądu obrazu.

4.2 Filtr górnoprzepustowy

Do kodu dla filtru dolnoprzepustowego należało dodać poniższe linijki kodu. Są one takie same jak w przypadku filtracji dla sygnału składającego się z sinusów.

```
1 hhp = dlt(41) - h;  
2  
3 yhp = filter2(hhp, xg);
```



Rysunek 19: Obraz po przefiltrowaniu filtrem górnoprzepustowym.

Patrząc na rysunek nr 19 zauważyć można, że krawędzie obrazu stały się bardziej widoczne. Spowodowane jest to tym, że filtracja górnoprzepustowa może zwiększyć kontrast i wyostrzyć krawędzie, podkreślając detale obrazu. Jest to przydatne, gdy chcemy wyróżnić struktury obrazu, takie jak krawędzie obiektów.

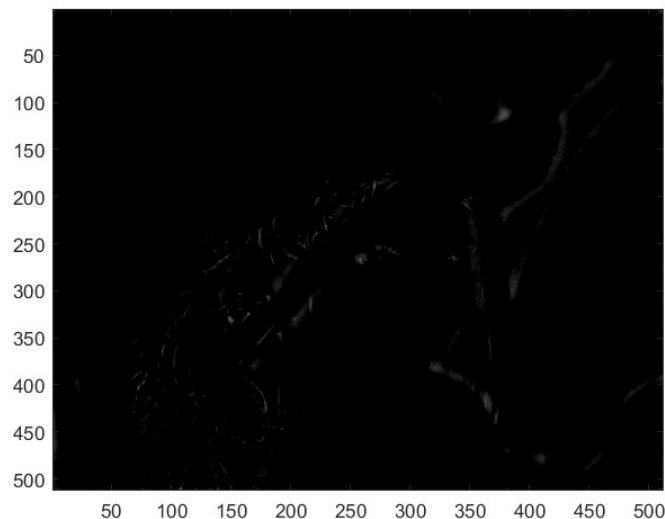
4.3 Filtr pasmowoprzepustowy i pasmowozaporowy

W przypadku filtrów pasmowoprzepustowych i pasmowozaporowych należało odjąć od siebie filtry dolno i górno przepustowe. Aby stworzyć filtr pasmowoprzepustowy odejmujemy filtr górnoprzepustowy od filtra dolnoprzepustowego, a w przypadku filtra pasmowozaporowego sytuację odwracamy.

```
1 ylp_minus_yhp = ylp - yhp; % pasmowoprzepustowy
2
3 yhp_minus_ylp = yhp - ylp; % pasmowozaporowy
```



Rysunek 20: Obraz po przefiltrowaniu filtrem pasmowoprzepustowym.



Rysunek 21: Obraz po przefiltrowaniu filtrem pasmowozaporowym.

Różnica między filtrami polega na tym, czy przepuszczają (pasmowoprzepustowy) czy eliminują (pasmowozaporowy) sygnały w określonym pasmie częstotliwości. Dlatego też dla filtru pasmowozaporowego obraz jest o wiele ciemniejszy, ponieważ po odjęciu od filtra górnoprzepustowego filtra dolnoprzepustowego pozostaje bardzo mały zakres częstotliwości. Oba te filtry są używane w różnych dziedzinach, takich jak przetwarzanie sygnałów, telekomunikacja, czy przetwarzanie obrazów, w zależności od konkretnego zadania i potrzeb.

5 Ćwiczenie 3: Filtracja kolorowego obrazka

Ostatnim zadaniem było wykonanie filtracji na kolorowym obrazie. Podejście do ćwiczenia było bardzo podobne do filtracji dla obrazu szarego, także wykorzystano funkcje *mask* i *dlt*. Jednak należało wykonać filtracje na wszystkich warstwach obrazu i zrobiono to przy pomocy pętli for.



Rysunek 22: Oryginalny obrazek.

5.1 Filtr dolnoprzepustowy

Inicjalizacja:

```
1 h = mask(0.1*pi,511);  
2 dp = zeros(511,511,3);  
3  
4 for i=1:3  
5     dp(:,:,i) = filter2(h,img(:,:,i));  
6 end
```

Gdzie:

1. `h = mask(0.1, 511);`

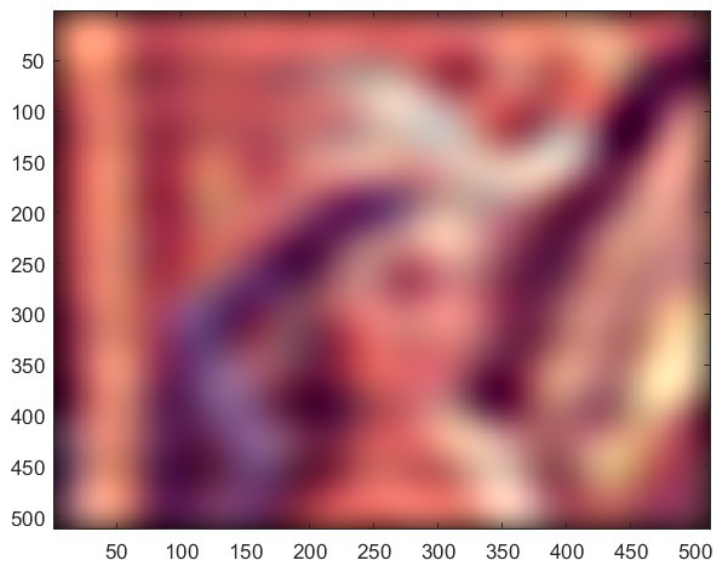
- Wywołuje funkcję `mask` z parametrami 0.1 i 511.

2. `dp = zeros(511, 511, 3);`

- Inicjalizuje trójwymiarową macierz `dp` o wymiarach 511x511x3, w której będą przechowywane wyniki filtracji dla każdej składowej koloru (czerwonej, zielonej i niebieskiej).

3. `for i = 1:3 dp(:, :, i) = filter2(h, img(:, :, i)); end`

- Przechodzi przez każdą składową koloru obrazu (czerwoną, zieloną, niebieską) i stosuje filtr `h` przy użyciu funkcji `filter2`. Wyniki są przypisywane do odpowiednich warstw macierzy `dp`. W efekcie, każda składowa koloru obrazu jest filtrowana za pomocą tego samego filtru `h`.



Rysunek 23: Obraz po przefiltrowaniu filtrem dolnoprzepustowym.

Tak jak w przypadku filtru dolnoprzepustowego dla obrazu szarego zauważyć można, że obraz stał się bardziej rozmyty. Stało się tak, ponieważ dolne częstotliwości odpowiadają za duże obszary obrazu. Mimo to zachował swoje oryginalne kolory. Zgadza się to z założeniami filtracji dolnoprzepustowej i świadczy o poprawnej inicjalizacji filtru.

5.2 Filtr górnoprzepustowy

```
1 ygp = dlt(511) - h;  
2 gp = zeros(511,511,3);  
3  
4 for i=1:3  
5     gp(:, :, i) = filter2(ygp, img(:, :, i));  
6 end
```

Gdzie:

1. `ygp = dlt(511) - h;`

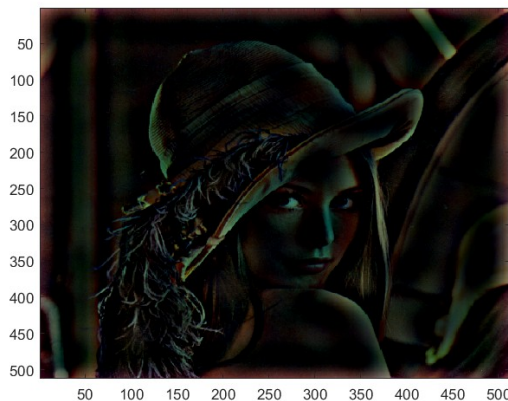
- Wywołuje funkcję `dlt` (tworzy deltę Kroneckera), a następnie odejmuje od niej maskę `'h'`.

2. `gp = zeros(511, 511, 3);`

- Inicjalizuje trójwymiarową macierz `gp` o wymiarach `511x511x3`, w której będą przechowywane wyniki filtracji dla każdej składowej koloru (czerwonej, zielonej i niebieskiej).

3. `for i = 1:3 gp(:, :, i) = filter2(ygp, img(:, :, i)); end`

- Przechodzi przez każdą składową koloru obrazu (czerwoną, zieloną, niebieską) i stosuje filtr `ygp` przy użyciu funkcji `filter2`. Wyniki są przypisywane do odpowiednich warstw macierzy `gp`.



Rysunek 24: Obraz po przefiltrowaniu filtrem górnoprzepustowym.

Można zauważyć podobieństwo do szarego obrazu przefiltrowanego przez filtr górnoprzepustowy. Krawędzie stały się ostrzejsze, podczas gdy reszta obrazu stała się ciemna. Obraz zachował także swoje kolory, widoczne szczególnie w obszarach krawędzi. Zgadza się to z założeniami filtracji górnoprzepustowej i świadczy o poprawnej inicjalizacji filtru.

6 Wnioski ogólne

Sprawozdanie dotyczące filtrów i filtracji skupiało się na analizie różnych aspektów przetwarzania obrazów. W ramach eksperymentów zastosowano różne rodzaje filtrów, a także okna przestrzenne, by zbadać ich wpływ na obrazy.

Pierwszym istotnym elementem w przetwarzaniu obrazów są okna przestrzenne, które pełnią kluczową rolę w procesie filtracji. Wyznaczają one obszar, na którym operuje filtr, i ich wybór wpływa na ostateczny efekt filtracji. Poszczególne rodzaje okien posiadają różne charakterystyki, co skutkuje zróżnicowanymi efektami w procesie przetwarzania.

W dalszej części sprawozdania skoncentrowano się na różnych rodzajach filtrów, w tym dolnoprzepustowych, górnoprzepustowych, pasmowoprzepustowych i pasmowozaporowych. Filtry te mają zastosowanie w wyodrębnianiu, redukowaniu lub przepuszczaniu określonych zakresów częstotliwości, co zostało zilustrowane na rysunku nr 5.

W ćwiczeniu pierwszym przeprowadzono filtrację sygnału sinusoidalnego 2D, wykorzystując filtry walcowe i prostokątne. To pozwoliło na wycięcie lub wzmocnienie poszczególnych harmonicznych, co ilustruje różnice między charakterystykami filtrów.

W kolejnym ćwiczeniu skoncentrowano się na filtracji obrazów szarych, wykorzystując różne filtry (dolnoprzepustowy, górnoprzepustowy, pasmowoprzepustowy, pasmowozaporowy). Proces ten pozwolił na manipulację zawartością częstotliwościową obrazu, co jest istotne w kontekście eliminacji szumów, wyostrzania krawędzi czy wygładzania tekstur.

W ostatnim ćwiczeniu przeniesiono koncepcję filtracji na obrazy kolorowe, gdzie zastosowano filtry dla każdej warstwy kolorów (RGB). To umożliwiło kontrolowanie częstotliwości w każdym kanale kolorów niezależnie, co ma znaczenie przy przetwarzaniu obrazów wielokolorowych.

Podsumowując, eksperymenty z filtrami i filtracją potwierdziły ich wszechstronne zastosowanie w przetwarzaniu obrazów. Okna przestrzenne oraz różne rodzaje filtrów pozwalają na precyzyjne dostosowanie procesu filtracji do konkretnych potrzeb, co stanowi istotny element w dziedzinie przetwarzania obrazów.

Literatura

- [1] Odpowiedź impulsowa [Online] -<https://sound.eti.pg.gda.pl/greg/dsp/02-FIR.html>
- [2] Okno czasowe [Online] - https://pl.wikipedia.org/wiki/Okno_czasowe