### Informatyka, studia dzienne, mgr II st. semestr I

## Przetwarzanie obrazu i dźwięku 2020/2021

### Prowadzący: dr inż. Bartłomiej Stasiak poniedziałek, 12:15

### Data oddania: Ocena:

Aleksander Dominiak 234049

Dominik Andrzejczak 234038

# Zadanie 2:

# Filtracja w dziedzinie częstotliwości i segmentacja obrazu.

## 1. Cel

Celem niniejszego zadania było zaimplementowanie takich operacji jak: szybkie i odwrotne przekształcenie Fouriera, filtry (dolnoprzepustowy, górnoprzepustowy, pasmowoprzepustowy, pasmozaporowy, z detekcją krawędzi i modyfikujący fazę widma transformaty Fouriera) oraz metoda segmentacji oparta o rozrost obszarów.

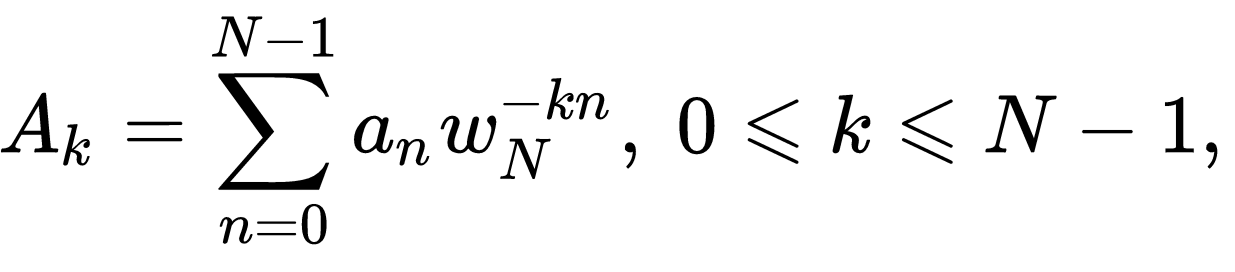
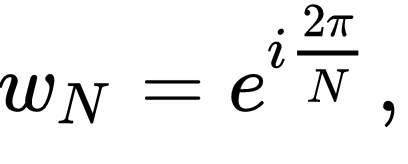
## 2. Wprowadzenie

Wykonanie zadania wiązało się z implementacją operacji przetwarzania obrazu, które oparte są na konkretnej wiedzy i algorytmach. Każda z nich wykorzystuje piksele, z których złożony jest obraz, do jego modyfikacji. W celu realizacji drugiego zadania wykorzystano aplikację stworzoną na potrzeby zadania pierwszego, a następnie ją rozbudowano. Umożliwia ona użytkownikowi zastosowanie wyżej wymienionych operacji.

1. **FFT**

Szybkie przekształcenie Fouriera (FFT) to bardzo wydajny algorytm implementujący dyskretne przekształcenie Fouriera (DFT).

DFT to transformata Fouriera wyznaczona dla sygnału próbkowanego, a więc dyskretnego. Przekształca ona skończony ciąg próbek sygnału (*a0, a1, ..., aN-1*), *ai*∈R w ciąg harmoniczny (*A0, A1, ..., AN-1*), *Ai*∈C, zgodnie ze wzorem:

gdzie:

*i* – jednostka urojona,

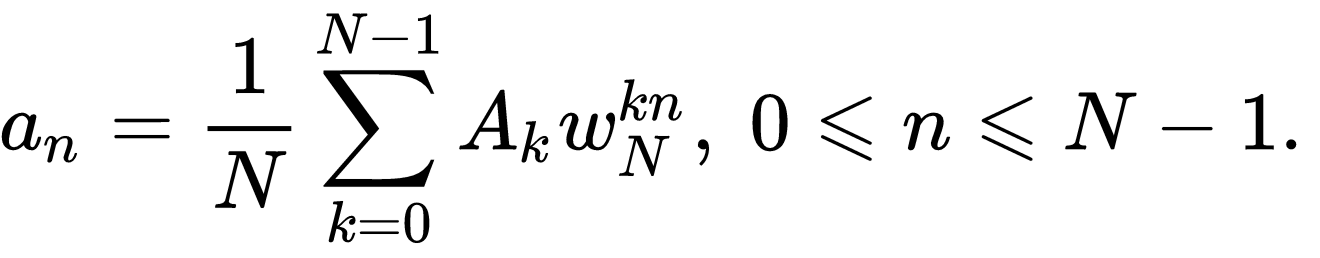
*k* – numer harmoniczny,

*n* – numer próbki sygnału,

*an* – wartość próbki sygnału,

*N* – liczba próbek

Przekształcenie odwrotne wyraża się wzorem:



FFT pozwala na szybsze obliczenie DFT, najpopularniejszym wariantem FFT jest ten o podstawie 2, wymagający, by rozmiar DFT był całkowitą potęgą liczby dwa.

DFT do obliczenia wymaga wykonania N2 mnożeń. Dodatkowym problemem jest fakt, że są to mnożenia liczb zespolonych, znacznie zużywające moc obliczeniową procesora. W przypadku FFT liczba mnożeń spada do około N/2 \* log2N, przy czym część z nich wymaga jedynie zmiany znaku.

Algorytm FFT korzysta z podanych heurystyk budowania szybkich algorytmów:

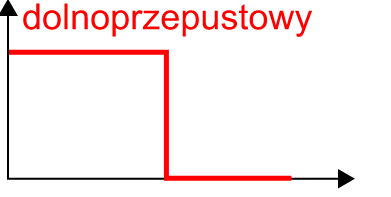
* „dziel i rządź” – podział zadania na mniejsze fragmenty,
* rekurencja,
* programowanie dynamiczne.

1. **Filtr dolnoprzepustowy**

Filtry dolnoprzepustowe przepuszczają elementy obrazu o małej częstotliwości. Elementy o wysokiej częstotliwości (szumy, drobne szczegóły) są natomiast tłumione a nawet blokowane. Wynikiem działania tych filtrów jest zredukowanie szumów, ale również wygładzenie i rozmycie obrazu.

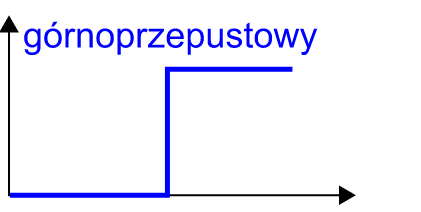
Podstawową wersją filtru dolnoprzepustowego jest filtr uśredniający, którego wynikiem jest uśrednienie wartości każdego piksela z uwzględnieniem wartości jego sąsiadów.

Idealny filtr dolnoprzepustowy polega na odcięciu składowych częstotliwości w oparciu o ich odległość od centrum transformaty.



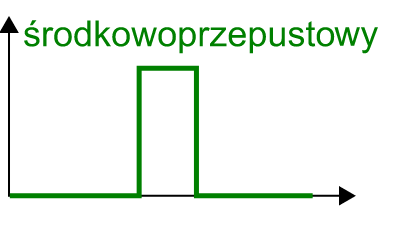
1. **Filtr górnoprzepustowy**

Filtry górnoprzepustowe działają na zasadzie przeciwnej do filtrów dolnoprzepustowych, wzmacniając takie elementy obrazu jak szumy, szczegóły lub krawędzie.



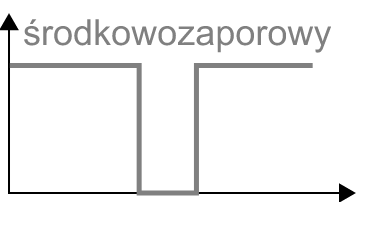
1. **Filtr pasmowoprzepustowy**

Filtry pasmowoprzepustowe (środkowoprzepustowe) przepuszczają składowe częstotliwości znajdujące się wyłącznie w danym przedziale. Pasmo przepustowe filtru definiuje się jako przedział znajdujący się między określonym częstotliwościami granicznymi lub przedział leżący wokół określonej częstotliwości środkowej.



1. **Filtr pasmozaporowy**

Filtry pasmozaporowe (środkowozaporowe) nie przepuszczają składowych częstotliwości znajdujących się w danym przedziale. Jest to odwrotność działania filtra pasmowoprzepustowego.

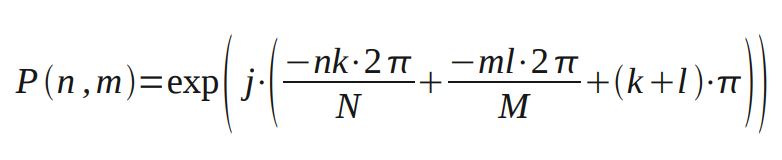


1. **Filtr z detekcją krawędzi**

Filtr wykrywający krawędź opiera się w swoim działaniu na filtrze górnoprzepustowym. Jest on używany do usuwania pikseli o niskiej częstotliwości. Wyostrza on krawędzie położone w konkretnym, wybranym, kierunku (np. NS, EW).

1. **Filtr modyfikujący fazę widma**

Filtr ten pozwala na zmodyfikowanie fazy widma obrazu, tzn. przesunięciu obrazu po podzieleniu go na 4 części, których rozmiar wynika z podanych wartości (*k* i *l* w poniższym wzorze). Przesunięcie wynika z pomnożenia każdego elementu widma przez:

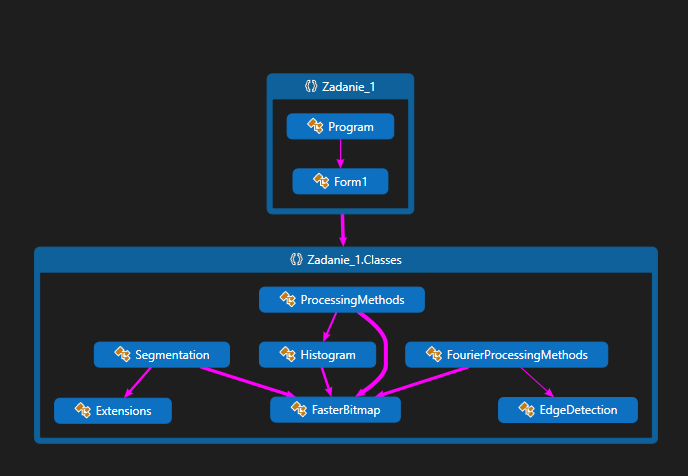


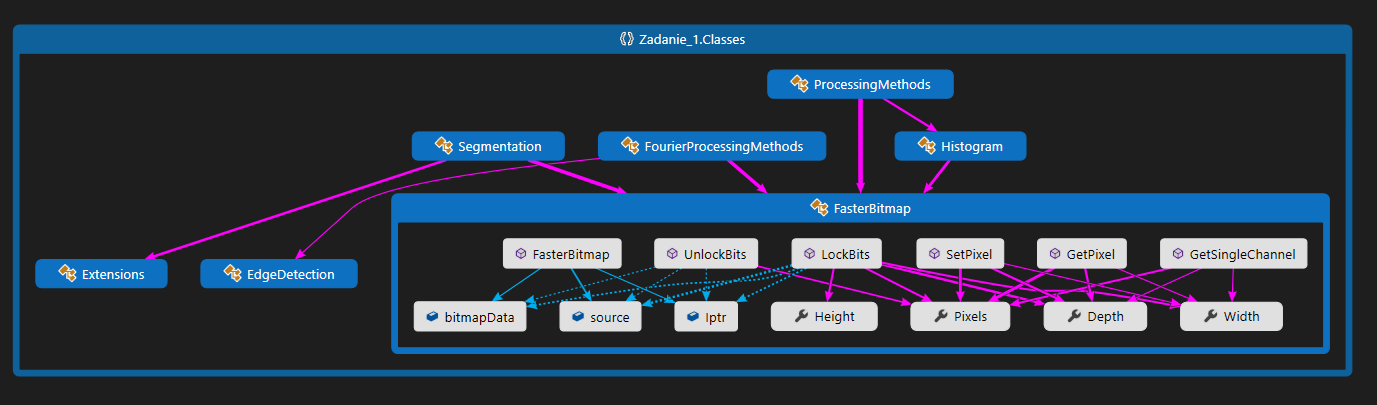
1. **Segmentacja – metoda rozrostu obrazu**

Segmentacja obrazu oznacza podzielenie go na zbiór regionów określonych jednakowymi wartościami wcześniej parametrów. Metoda rozrostu obrazu polega na wybraniu pewnych pikseli (ziaren). Następnie ich wartość jest porównywana z wartościami sąsiednich pikseli. Jeżeli ich podobieństwo znajduje się w określonych granicach, sąsiad jest włączany do tworzonego regionu. Rozrost obszaru trwa do momentu znalezienia wszystkich sąsiadów spełniających założenia.

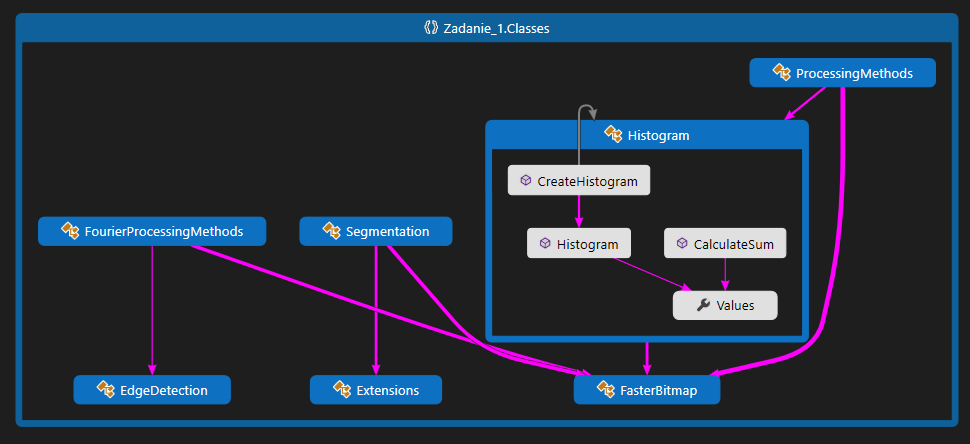
## 3. Opis implementacji

Aplikacja została napisana w języku C# z użyciem frameworka Windows Forma.

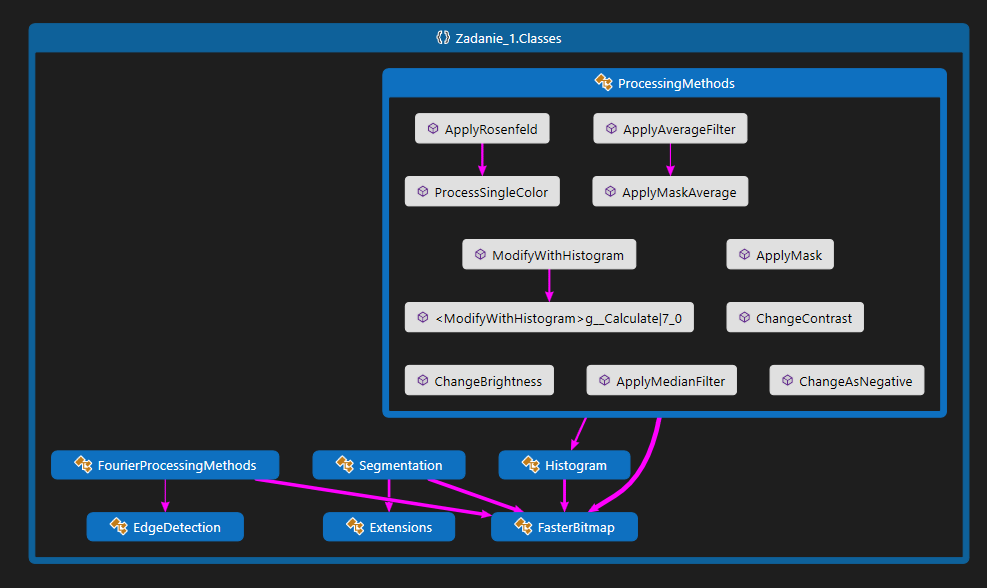




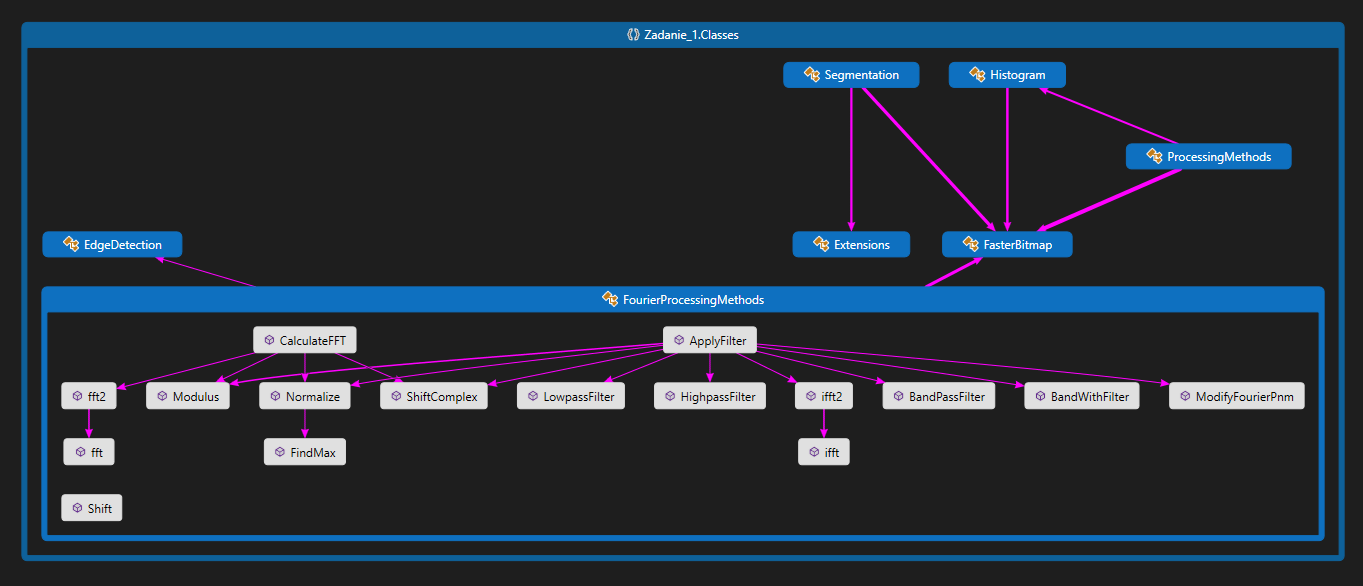
Klasa FasterBitmap odpowiada za przyśpieszenie edycji wbudowanej klasy bitmapy. Pozwala ona na pobranie i ustawienie wartości danego piksela lub w przypadku obrazów kolorowych konkretnego kanału.



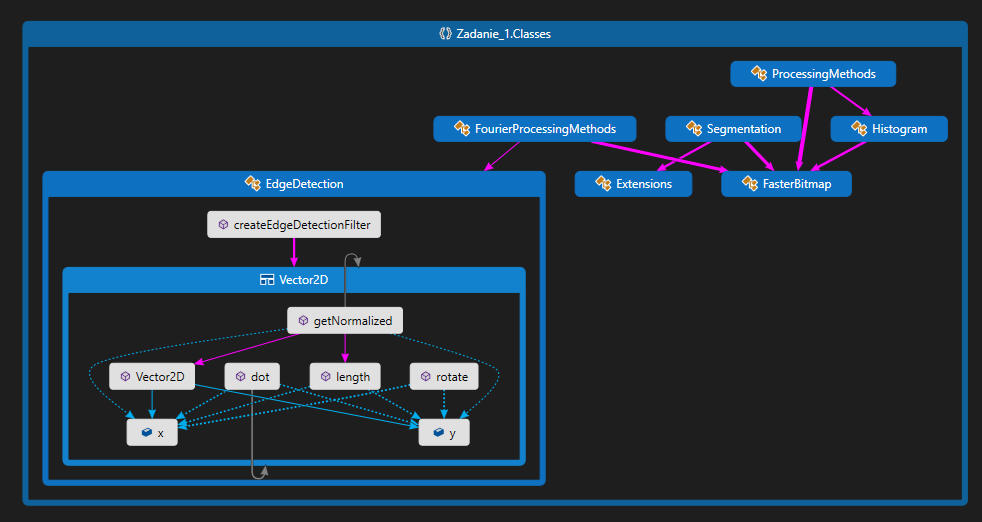
Klasa Histogram oblicza i przechowuje wartości potrzebne do stworzenia wykresu.



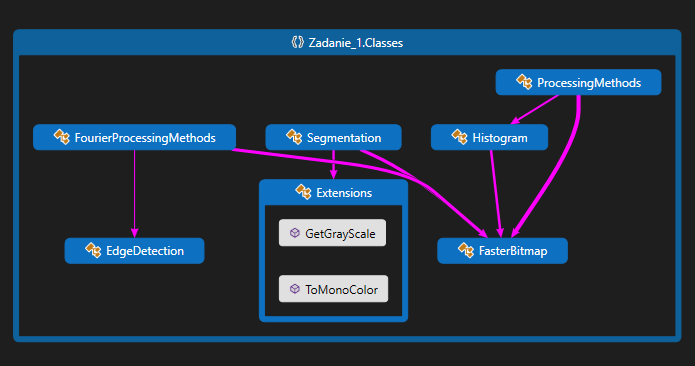
Statyczna klasa ProcessingMethods to zbiór metod modyfikujących bitmapę w oparciu o przekazane wartości.



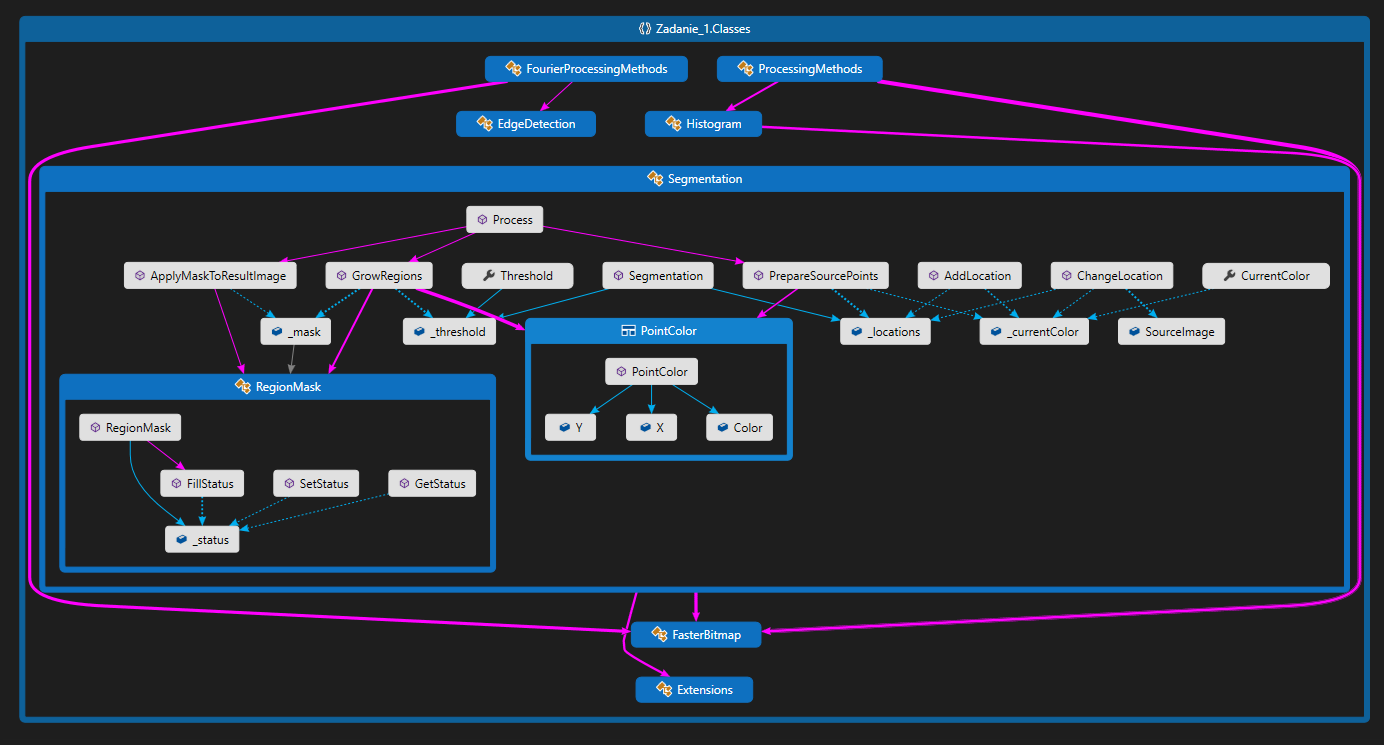
Statyczna klasa FourierProcessingMethods zawiera metody implementujące filtry i pomocnicze operacje opisane wyżej.



Statyczna klasa EdgeDetection to klasa pomocnicza, pozwalająca skorzystać z filtru wykrywającego krawędź. Do swojego działania wykorzystuje zagnieżdżoną strukturę Vector2D.



Statyczna klasa Extensions zawiera metody pomocnicze, wykorzystane przy obróbce kolorów – obliczenie skali szarości z kolorowego piksela i zamianę pojedynczej wartości w kolor.



Klasa Segmentation odpowiada za segmentację obrazu. Zagnieżdżona klasa RegionMask ułatwia tworzenie i używanie maski używanej przy segmentacji. Zagnieżdżona struktura PointColor przechowuje informację na temat ziaren – ich położenie i kolor.

## 4. Materiały i metody

Z wykorzystaniem stworzonej aplikacji przeprowadzono badania dla wybranych obrazów (a-h) w celu zweryfikowania poprawności ich działania. Zastosowano wszystkie zaimplementowane operacje i filtry do wszystkich poniższych wariantów:

|  |  |
| --- | --- |
|  |  |
| a |  |
|  |  |
| b |  |
| C:\Users\dmnk0\AppData\Local\Microsoft\Windows\INetCache\Content.Word\mandril.bmp |  |
| c |  |

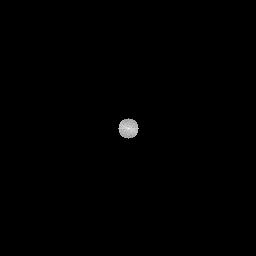
Metody:

1. Filtry poza wykrywanie krawędzi i modyfikacją widma – b i c.
2. Wykrywanie krawędzi i modyfikacja widma – b.
3. Segmentacja – a i b.

|  |  |
| --- | --- |
| Filtr | Wartości |
| dolnoprzepustowy | poziom: 10, 100 |
| górnoprzepustowy | poziom: 10, 100 |
| pasmozaporowy | przedziały: 10- 20, 75-150 |
| pasmoprzepustowy | przedziały: 10- 20, 75-150 |
| detekcja krawędzi | poziom: 10, 30  kąt: 90, 180  fi: 0.05, 0.5 |
| modyfikacja widma | przedziały: 10-20, 100-200 |
| segmentacja | próg: 10, 50 |

## 5. Wyniki

1. **dolnoprzepustowy**
   1. **10**

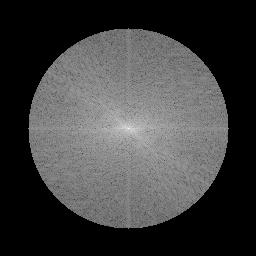


rys. A1a



rys. A1

* 1. **100**

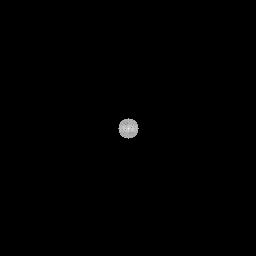


rys. A2a



rys. A2

* 1. **10 – II**

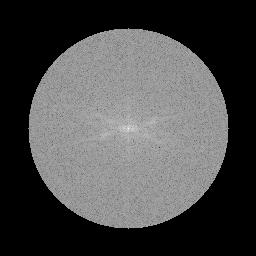


rys. A3a



rys. A3

* 1. **100 – II**

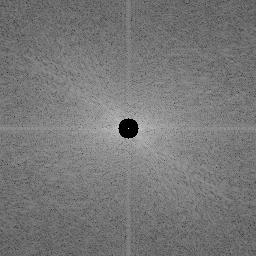


rys. A4a



rys. A4

1. **górnoprzepustowy**
   1. **10**

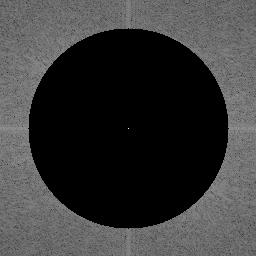


rys. B1a

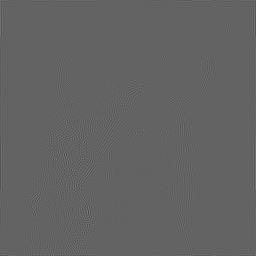


rys. B1

* 1. **100**

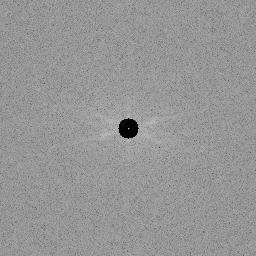


rys. B2a

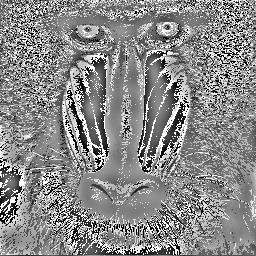


rys. B2

* 1. **10 – II**

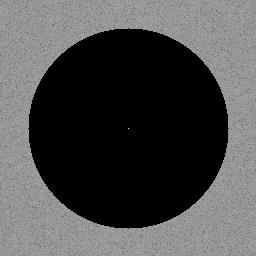


rys. B3a

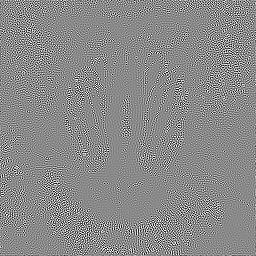


rys. B3

* 1. **100 – II**

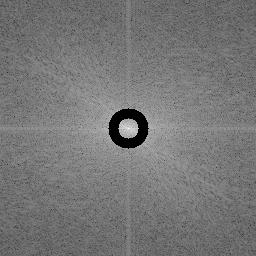


rys. B4a



rys. B4

1. **pasmowozaporowy**
   1. **10-20**

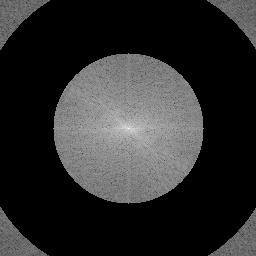


rys. C1a



rys. C1

* 1. **75-150**

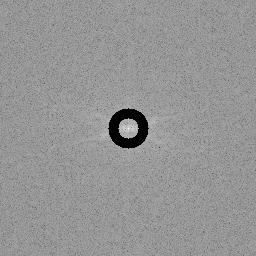


rys. C2a

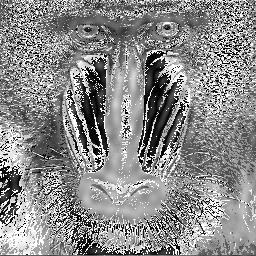


rys. C2

* 1. **10-20 – II**

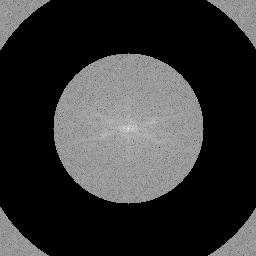


rys. C3a



rys. C3

* 1. **75-150 – II**

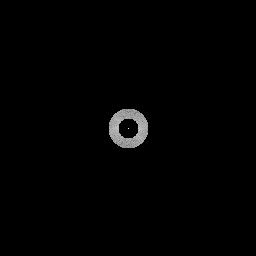


rys. C4a

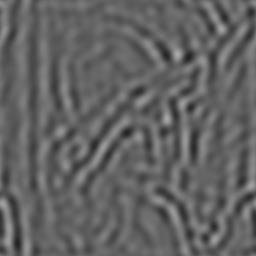


rys. C4

1. **pasmowoprzepustowy**
   1. **10- 20**

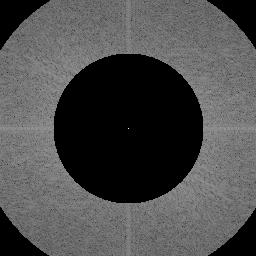


rys. D1a

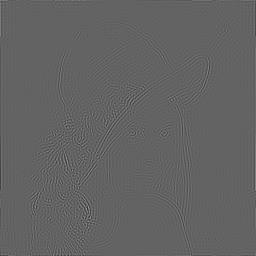


rys. D1

* 1. **75- 150**

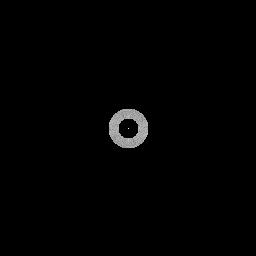


rys. D2a



rys. D2

* 1. **10- 20 – II**

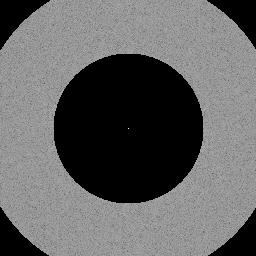


rys. D3a

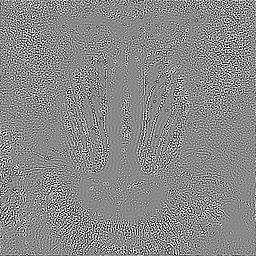


rys. D3

* 1. **75- 150 – II**

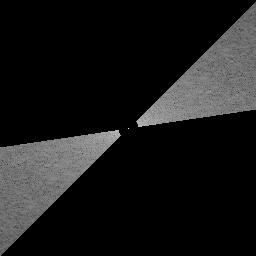


rys. D4a

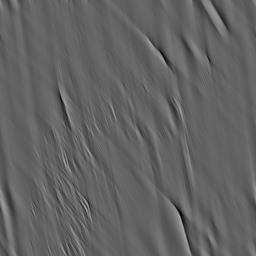


rys. D4

1. **Wykrywanie krawędzi**
   1. **10, kąt: 90, fi: 0.05**

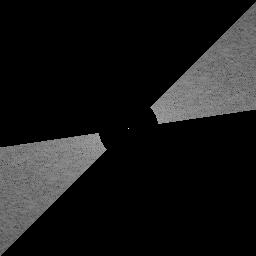


rys. E1a

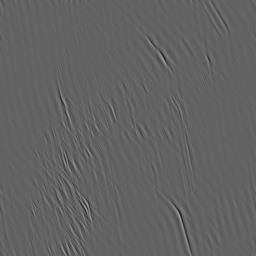


rys. E1

* 1. **30, kąt: 90, fi: 0.05**

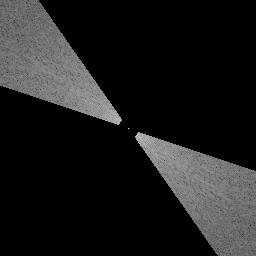


rys. E2a

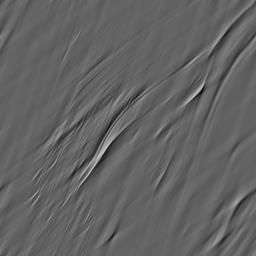


rys. E2

* 1. **10, kąt: 180, fi: 0.05**

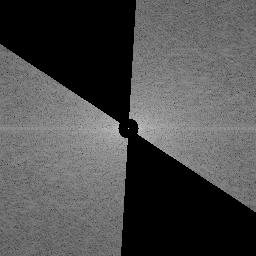


rys. E3a



rys. E3

* 1. **10, kąt: 90, fi: 0.5**

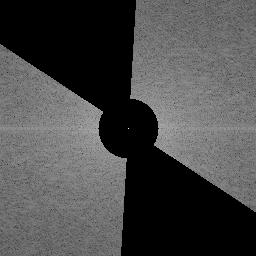


rys. E4a

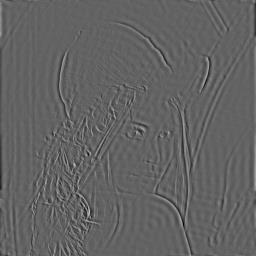


rys. E4

* 1. **30, kąt: 90, fi: 0.5**

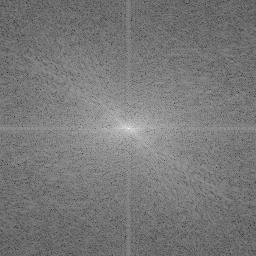


rys. E5a



rys. E5

1. **Modyfikacja widma**
   1. **10-20**

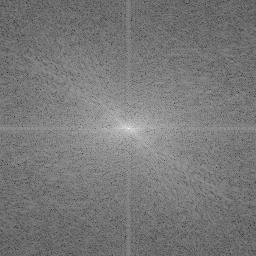


rys. F1a



rys. F1

* 1. **100-200**



rys. F2a



rys. F2

1. **Segmentacja**
   1. **próg: 10, odcienie szarości**



rys. G1a



rys. G1

* 1. **próg: 50, odcienie szarości**



rys. G2a



rys. G2

* 1. **próg: 10, kolorowe**



rys. G3a



rys. G3

* 1. **próg: 50, kolorowe**



rys. G4a



rys. G4

## 6. Dyskusja

1. dolnoprzepustowy –jak widać na rysunkach z tego podpunktu uzyskane efekty są zgodne z oczekiwaniami, filtr usuwa wszystkie składowe częstotliwości poniżej zadanych progów.
2. górnoprzepustowy – podobnie jak wyżej, rezultaty zgadzają się z oczekiwaniami.
3. pasmowozaporowy – filtr skutecznie usuwa składowe częstotliwości pomiędzy ustalonymi zakresami.
4. pasmowoprzepustowy – filtr skutecznie usuwa składowe częstotliwości znajdujące się poza ustalonym zakresem.
5. wykrywanie krawędzi – odpowiednie ustawienie parametrów umożliwia wykrycie krawędzi na rysunku, przy innych ustawieniach możliwe jest uzyskanie innych interesujących efektów. Zmiana kąta wpływa na wykryte krawędzie.
6. modyfikacja widma – ten filtr skutecznie dzieli obraz na cztery części zgodnie z podanymi współczynnikami, a następnie odpowiednio przesuwa uzyskane części.
7. segmentacja – maska jest tworzona w oparciu o podane ziarna, rezultat jest zgodny z oczekiwaniami. Zmiana progu wpływa na wielkość uzyskanych regionów.

## 7. Wnioski

1. Szybkość działania wszystkich metod jest zależna od ich skomplikowania, i przede wszystkim wielkości obrazów.
2. Zastosowanie formaty Fouriera umożliwia modyfikowanie częstotliwości obrazu, co pozwala uzyskać interesujące efekty, ale ma także zastosowanie praktyczne – wykrywanie szczegółów, odszumianie.
3. Podobnie jest w przypadku segmentacji obrazu.
4. Transformata Fouriera może zastąpić niektóre z metod użytych   
   w zadaniu 1.

## Literatura

1. <https://ftims.edu.p.lodz.pl/pluginfile.php/141357/mod_resource/content/1/PO_i_DZ_19_20_W6.pdf>
2. <https://akshaysin.github.io/fourier_transform.html>
3. <http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html>
4. <http://mstrzel.eletel.p.lodz.pl/mstrzel/pattern_rec/filtering.pdf>
5. <https://universe.bits-pilani.ac.in/uploads/JNKDUBAI/ImageProcessing7-FrequencyFiltering.pdf>