

AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Analiza Obrazów

Sprawozdanie nr 1

Laboratoria 2, 3, 4

Łukasz Wajda, 10.11.2021 r.

1. Laboratorium 2

Celem drugich zajęć laboratoryjnych było zapoznanie się z podstawowymi pojęciami i algorytmami wykorzystywanymi w analizie obrazów. Dowiedzieliśmy się, w jaki sposób obraz jest przechowywany w pamięci przy użyciu programu Matlab i jak sprawnie go przekształcać w zależności od tego, co chcemy uzyskać. Poniższy obraz posłużył nam do przeprowadzenia konkretnych operacji.



Rysunek 1.: Obraz żubra, który posłużył do testowania dalszych funkcji

1.1 Przechowywanie obrazu

Obraz został wczytany za pomocą funkcji:

```
im = imread('zubr.jpg'),
```

oraz domyślnie przechowywany jest w postaci macierzy trójwymiarowej **im** o wartościach typu całkowitego. Następnie nastąpiła konwersja poszczególnych wartości w macierzy z Int na Double w zakresie 0-1, a kolejno znormalizowaliśmy wartości dzieląc je przez 255. Tak wczytane i przekształcone dane wyświetliliśmy za pomocą funkcji:

```
imshow(im).
```

1.2 Filtrowanie kanałów

Matlab przechowuje obrazy, jako wielowymiarowe macierze, piksele są poszczególnymi komórkami w macierzy, a kolejne jej wymiary przedstawiają inne właściwości, np. poszczególne kolory składowe. Spróbowaliśmy wyciągnąć warstwę zieloną. Tworzymy kopię oryginalnego obrazu, zerujemy warstwę zieloną, wyodrębniamy ją i wyświetlamy:

```
int = im,  
int(:, :, [2,3]) = 0,  
int = im(:, :, 2),  
imshow(int).
```



Rysunek 2.: Kanał z kolorem zielonym obrazu

1.3 Konwersja do odcieni szarości

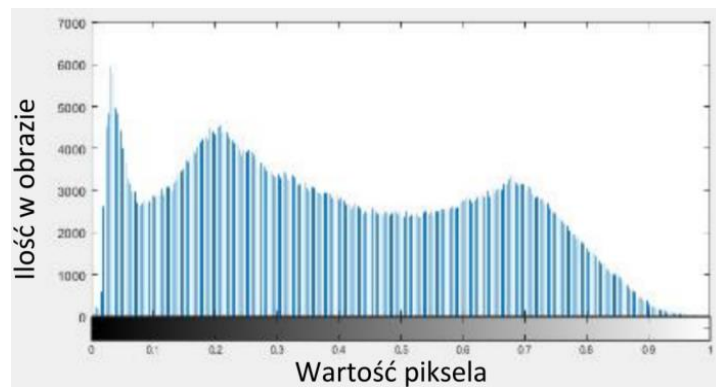
Dla większej wydajności wykonywanych obliczeń, a także dalszej analizy obrazu przekształciliśmy obraz do odcieni szarości za pomocą funkcji akumulacyjnych. Aby nie tracić wiele cennych informacji, wykorzystując jedynie pojedynczy kanał przy konwersji, lepszym sposobem było wykorzystanie średniej arytmetycznej z trzech kanałów, jako wartość jasności piksela w punkcie. Do konwersji kolorowego żubra do odcieni szarości z przedziału $[0, 1]$ posłużyła nam funkcja:

$n = \text{mean}(\text{im},3).$



Rysunek 3.: Żubr w wersji monochromatycznej z wykorzystaniem średniej arytmetycznej.

Przy analizie obrazów istotnym narzędziem jest histogram. Dla powyższego obrazu, histogram przedstawiono poniżej.



Wykres 1.: Histogram wartości pikseli dla żubr w wersji monochromatycznej z wykorzystaniem średniej arytmetycznej.

W dalszej kolejności wykorzystano model YUV, w którym osobna liczba odpowiada za wartość samej jasności, zaś kolory kodowane są w dwóch kanałach chrominancji – U. Konwersji z RGB do YUV możemy dokonać mnożąc odpowiednie piksele przez macierz z podanymi współczynnikami:

$$YUV=[0.299, 0.587, 0.114],$$

$$YUV = \text{permute}(YUV,[1,3,2]),$$

$$n = \text{sum}(im.*w,3).$$

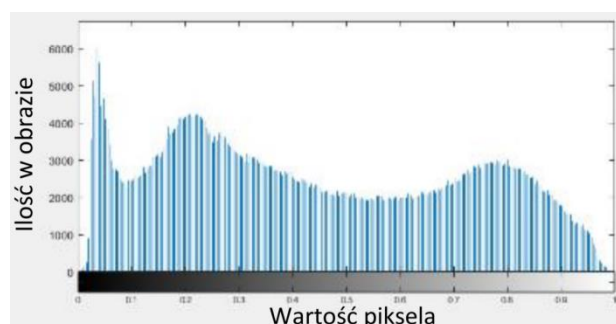
Otrzymaliśmy w ten sposób macierz średniej ważonej sumy kolorów. Powyższe operacje są równoznaczne z wykorzystaniem następującej funkcji:

$$n = \text{rgb2gray}(im).$$

Pietą achillesową wspomnianej funkcji jest występowanie warstw podczerwieni czy ultrafioletu w obrazie, wówczas metoda nie zadziała poprawnie.



Rysunek 4.: Żubr w wersji monochromatycznej z wykorzystaniem średniej ważonej.



Wykres 2.: Histogram wartości pikseli dla żubr w wersji monochromatycznej z wykorzystaniem średniej ważonej.

Dla powyższego wykresu można zaobserwować peak koloru czarnego.

1.4 Inwersja kolorów

Operacja zapewnia, że wartości białe, będą miały kolor czarny i na odwrót. W celu dokonania tego przekształcenia skorzystaliśmy z funkcji:

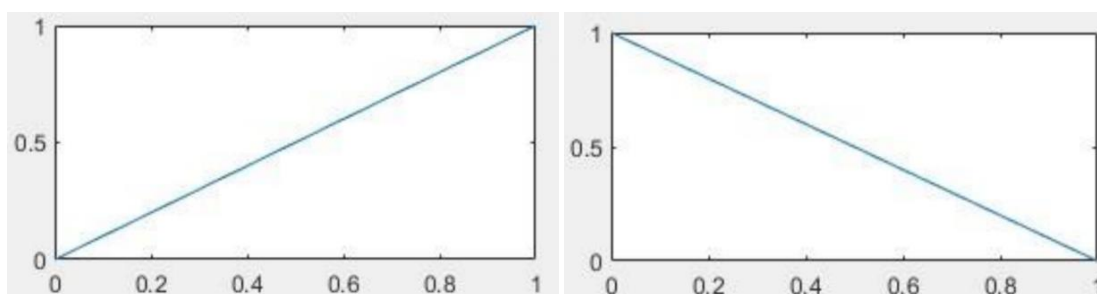
$$n = 1-n.$$

W bardzo wielkim uproszczeniu otrzymaliśmy białego żubra na szarym tle, co prezentuje poniższy rysunek.



Rysunek 5.: Żubr w wersji monochromatycznej po operacji inwersji kolorów.

Wykresy przekształceń lepiej obrazują tę różnicę.



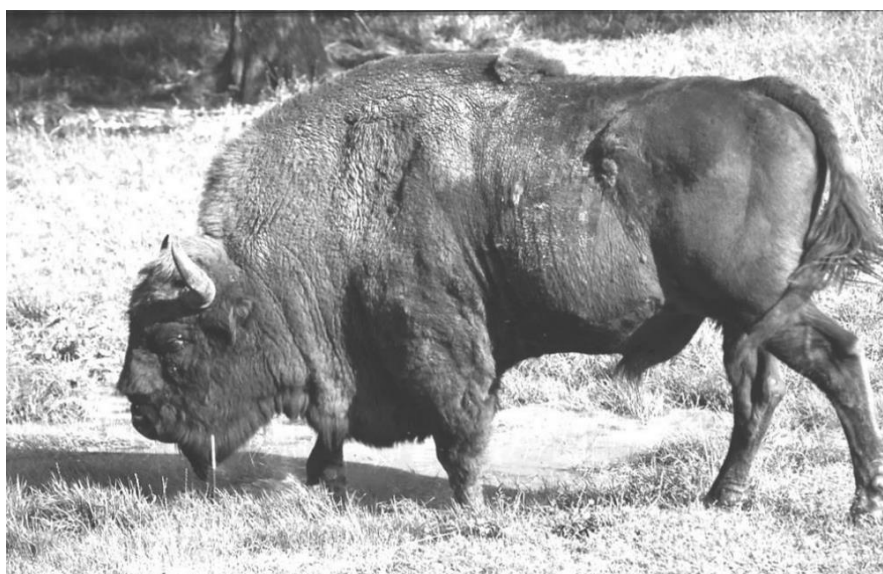
Wykres 3.: Wykres przekształcenia koloru nowego(y) od koloru oryginalnego(x). Po prawej dla oryginalnego obraz, po lewej dla obrazu po inwersji.

1.5 Modyfikacja jasności obrazu

Zmianę jasności przeprowadzono się poprzez:

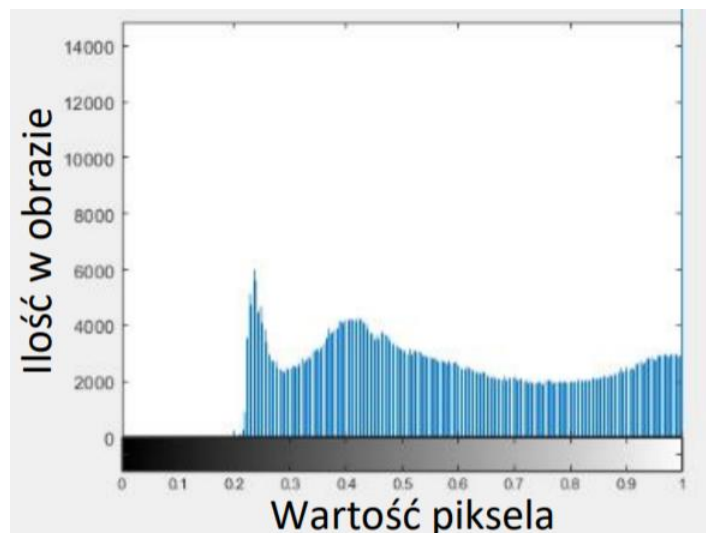
$$\begin{aligned} \text{imb} &= n + b, \\ \text{imb}(\text{imb} < 0) &= 0, \\ \text{imb}(\text{imb} > 1) &= 1. \end{aligned}$$

W wyniku dodawania stałej zaobserwowaliśmy przesunięcie się histogramu oraz zmianę jasności obrazu. Gdy wartość stałej jest poza zakresem $[0, 1]$ następuje obcięcie. Poniżej przedstawiono rezultaty operacji.

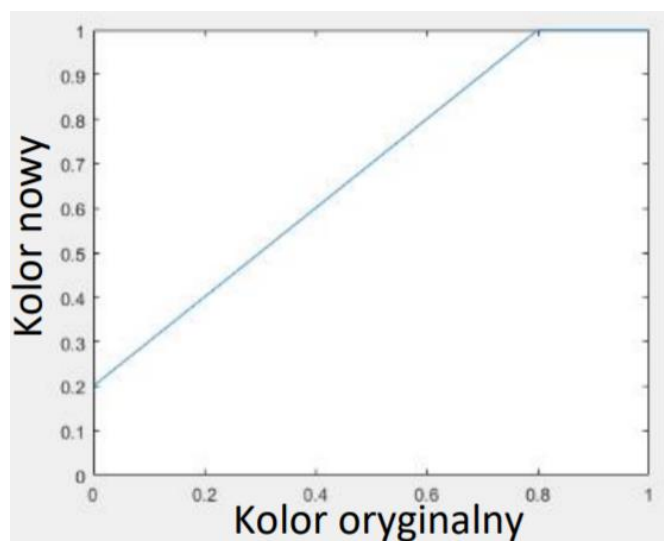


Rysunek 6.: Rozjaśniony żubr po dodaniu wartości 0.2 do pikseli.

Przyjęliśmy $b = 0.2$ i zaobserwowano przesunięcie histogramu w prawo z bardzo dużą ilością pikseli o wartości 1 (wszystkie te, które przekroczyły 1 po przesunięciu).



Wykres 4.: Histogram wartości pikseli dla żubr w wersji monochromatycznej po zwiększeniu jasności.



Wykres 5.: Wykres przekształcenia po dodaniu wartości 0.2 do pikseli.

1.6 Modyfikacja kontrastu obrazu

Zmiana kontrast polega na zmienianiu różnicy pomiędzy największą i najmniejszą wartością. Dla $c=2$:

$$cim = cim * c,$$

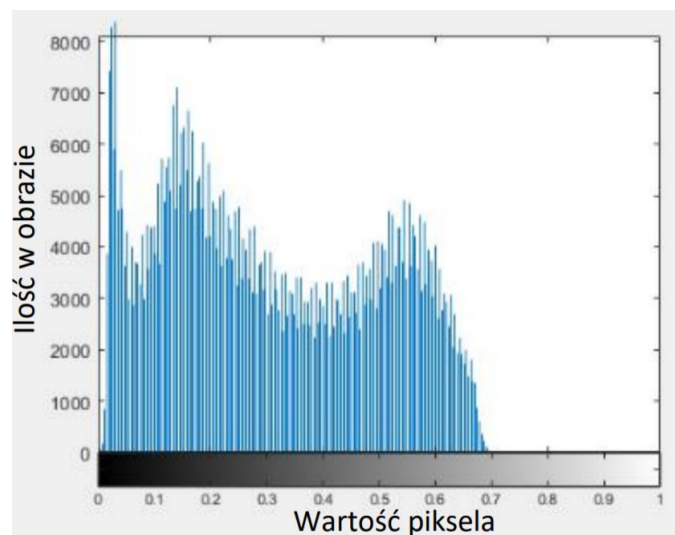
$$\text{cim}(\text{cim} > 1) = 1,$$

$$\text{cim}(\text{cim} < 0) = 0.$$

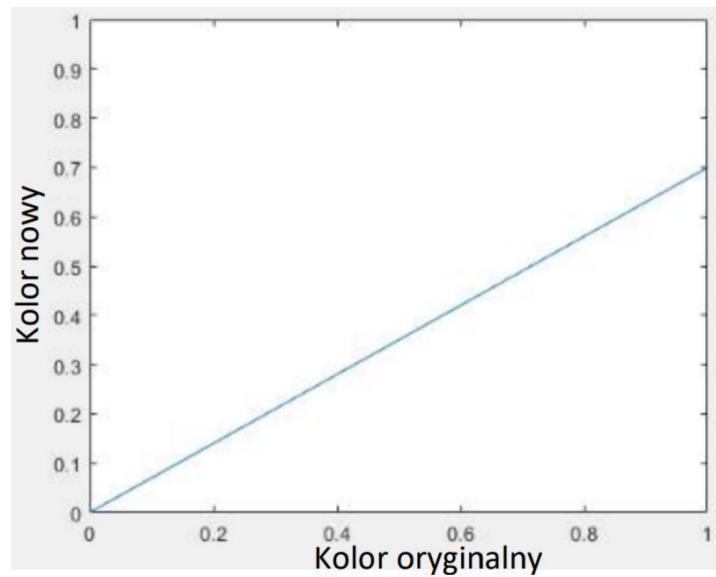
Dla wartości c większych od 1 następuje rozciągnięcie histogramu, uwydatnione zostaną jasne miejsca obrazu, natomiast poniżej jedynki następuje zwężenie histogramu, uwydatnione zostaną ciemne fragmenty.



Rysunek 7.: Wyostrzony żubr po przemnożeniu pikseli przez wartości 0.7.



Wykres 6.: Histogram wartości pikseli dla żubr w wersji monochromatycznej po zwiększeniu kontrastu.



Wykres 7.: Wykres przekształcenia po przemnożeniu pikseli przez wartości 0.7.

1.7 Modyfikacja gamma

Nienilową operację (zmiany kontrastu) gamma realizują się przez potęgowanie wartości pikseli przez stałą, zgodnie ze wzorem:

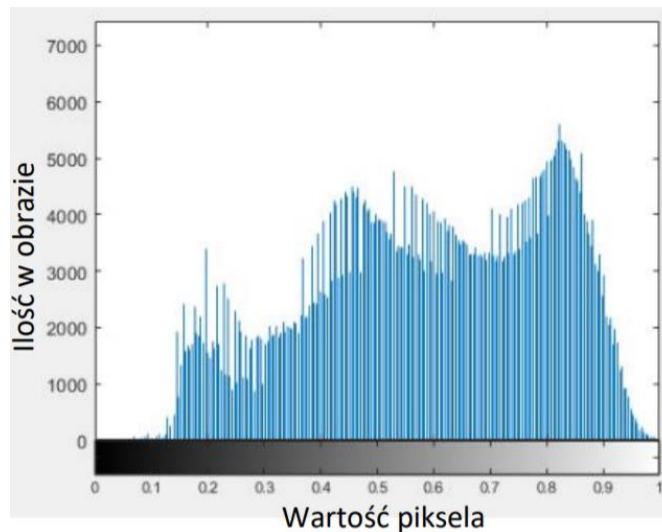
$$gim = n .^ d.$$

Dla znormalizowanego obrazu, im większy parametr gamma od wartości 1 tym jaśniejszy obraz. Korekcja Gamma jest najczęściej stosowana korekcja jasności w przypadku urządzeń elektronicznych.

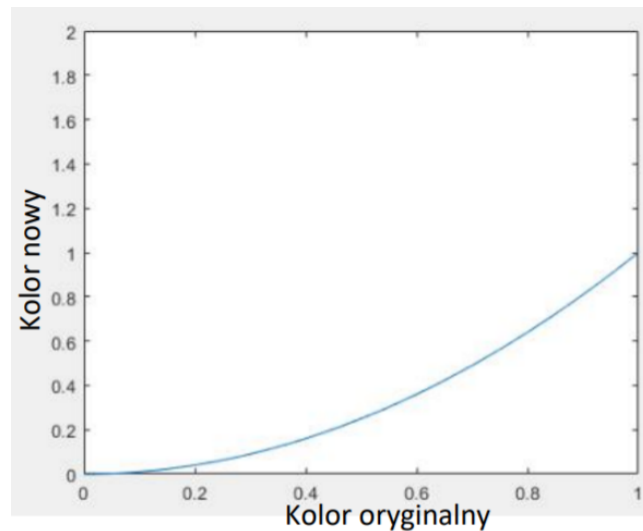


Rysunek 8.: Żubr po podniesienie wartości pikseli do potęgi 0.5.

Przedział histogramu $<0,0.25>$ został rozciągnięty do $<0,0.5>$. Pozostała część została ściśnięta. Efektem jest wyszczególnienie różnic między ciemnymi kolorami i utrata różnic między jasnymi kolorami. Gamma 2 dałaby odwrotny efekt.



Wykres 8.: Histogram wartości pikseli dla żubr w wersji monochromatycznej po podniesienie wartości pikseli do potęgi 0.5.



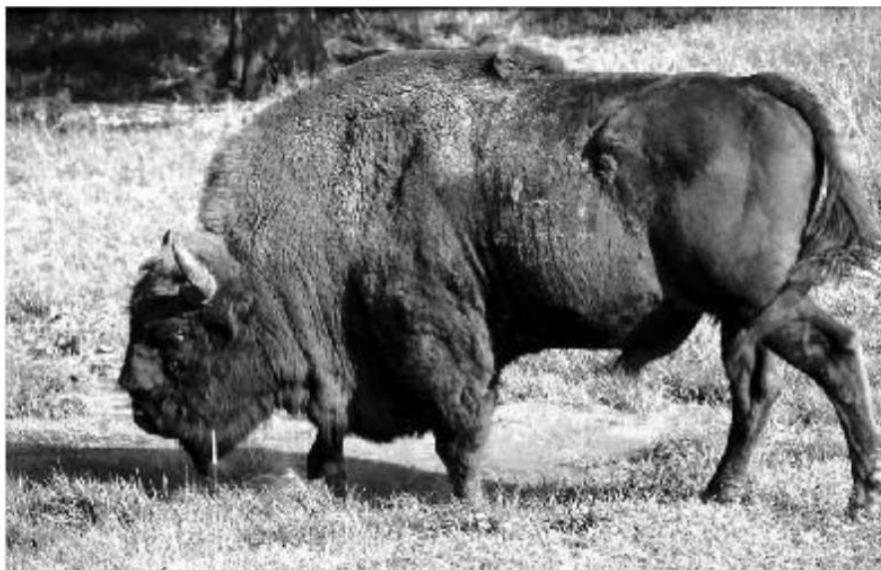
Wykres 9.: Wykres przekształcenia po podniesienie wartości pikseli do potęgi 0.5.

1.8 Normalizacja

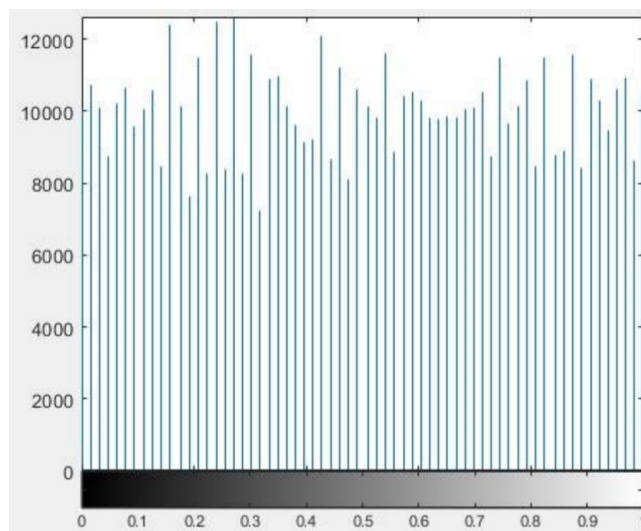
W celu zmniejszenia ilości odcieni szarości na obrazie, a tym samym wzmocnienie kontrastu dokonano ujednolicenie obrazu poprzez wyrównanie histogramu funkcją:

$$\mathbf{imh} = \mathbf{histeq}(\mathbf{im}).$$

Rezultat użycia normalizacji przedstawiono poniżej.



Rysunek 9.: Żubr po normalizacji.



Wykres 10.: Histogram wartości pikseli dla żubr w wersji monochromatycznej po normalizacji.

1.9 Wnioski

- Matlab zapisuje obrazy w formie macierzy wielowymiarowych, składających się z warstw kolorów.
- Za pomocą wbudowanych funkcji możemy manipulować obrazem, uzyskać interesującą nas jakość.

2. Laboratorium 3

Celem następnych zajęć laboratoryjnych było zapoznanie się z szerokim zagadnieniem, jakim są filtry. Rozpoczęliśmy od przyjęcia wspólnych definicji bazowych, jakimi są sąsiedztwo von Neumana i sąsiedztwo Moore'a.

- Sąsiedztwo Moore'a – sąsiedztwo środkowego piksela z ośmioma otaczającymi go pikselami, które graniczą z nim krawędzią, bądź wierzchołkiem.
- Sąsiedztwo von Neumanna – sąsiedztwo centralnego piksela z czterema otaczającymi go pikselami, graniczącymi z tym pikselem jedynie krawędziami.

Na potrzeby naszych zajęć wykorzystywaliśmy sąsiedztwo Moore'a do nakładania filtrów.

2.1 Filtr dolnoprzepustowy

Zaczęliśmy od stworzenia macierzy jedynek 3x3 i znormalizowania jej, aby suma wag wynosiła 1:

$$f = \text{ones}(3),$$
$$f = f/\text{sum}(f, 'all'),$$

Tak przygotowaną macierz przekazaliśmy do funkcji:

$$f_{im} = \text{imfilter}(n, f),$$

gdzie n to obraz żubra. Funkcja ta powoduje, że piksel w środku jest uśredniony według wag w macierzy. Efekt widzimy na poniższym rysunku.



Rysunek 10.: Żubr przefiltrowany przez macierz $f(3)$.

Przy zastosowaniu większej macierzy nastąpił efekt rozmycia.



***Rysunek 11.:** Żubr przefiltrowany przez macierz $f(9)$.*

Poza rozmyciem nasępiło również złagodzenie granic między jasnymi i ciemnymi częściami obrazu. Pojawiła się również ciemna ramka, są to po prostu piksele o wartości 0, które wyszły poza granicę zdjęć.

2.2 Filtr górnoprzepustowy

W celu uzyskania filtru górnoprzepustowego stworzono następującą macierz:

$$\mathbf{f} = \text{-ones}(3),$$

$$\mathbf{f}(2,2) = 9.$$

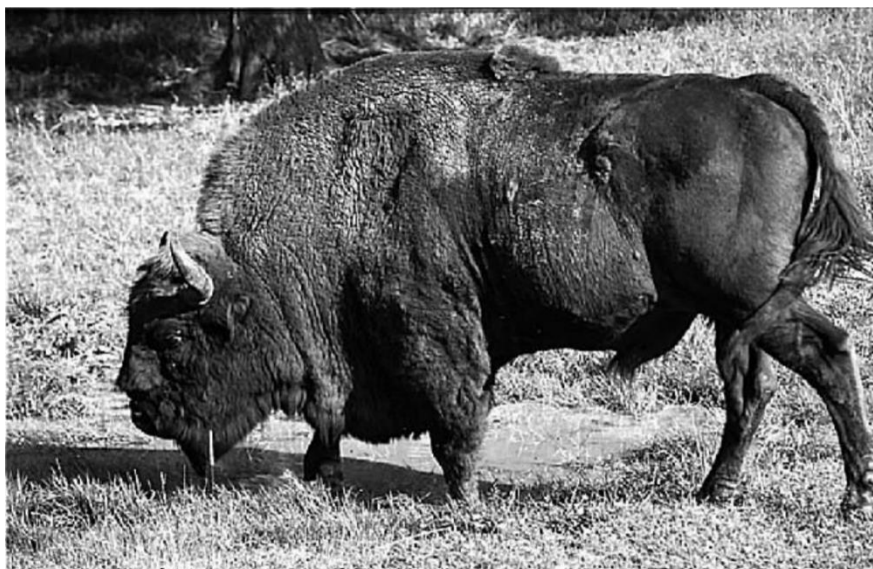


Rysunek 12.: Żubr przefiltrowany filtrem górnoprzepustowym.

Nastąpiło podniesienie kontrastu oraz wyróżnienie szczegółów na obrazie. Wyszczególniliśmy krawędzie obiektów, ale zwiększyły się zakłócenia obrazu. Filtr górnoprzepustowy doprowadził do zwiększenia szumu na obrazie, czyli stłumił niskoczęstotliwościowe elementy obrazu, a wzmocnił elementy o wysokich częstotliwościach.

2.3 Połączenie filtra dolno- i górnoprzepustowego

Filtrując obraz najpierw filtrem dolnoprzepustowym, a kolejno górnoprzepustowym, otrzymaliśmy zmniejszenie szumów i wyostrenie zdjęcia, co sprowadza się do dość szczegółowej korekcji obrazu.



Rysunek 13.: Żubr przefiltrowany filtrem dolno- i górnoprzepustowym.

2.4 Filtr medianowy

Filtr ten podobnie jak filtr dolnoprzepustowy rozmywa oraz zmniejsza szumy obrazu. Na potrzeby zrealizowania filtru medianowego wykorzystaliśmy gotową funkcję:

`imm = medfilt2(im,[3,3]).`

Filtr medianowy świetnie nadaje się do usuwania punktowych zakłóceń, jest przykładem filtru nieliniowego.



Rysunek 14.: Żubr przefiltrowany medianowo.

2.5 Binarizacja

Binarizacja polega w głównej mierze na sprowadzeniu wszystkich pikseli do wartości 0 lub 1 – innymi słowy, sprawieniu, że nasz obraz jest perfekcyjnie czarno-biały. Efekt oddzielenia żubra od tła uzyskaliśmy przy wykorzystaniu funkcji:

$$\begin{aligned}t &= \text{graythresh}(n), \\ \text{imb} &= \text{imbinarize}(n, t).\end{aligned}$$

Próg binaryzacji został ustalony automatycznie metodą Otsu. Wykorzystując binaryzację w ogólności chcemy osiągnąć biały obiekt i czarne tło, lub na odwrót np. przy analizie testu z obrazu. Aby osiągnąć ten efekt użyliśmy funkcji `imbinarize` z wykorzystaniem metody ‘adaptive’, która pozwala na lokalne, adaptacyjne wyznaczenie progu (threshold).



Rysunek 15.: Żubr zbinaryzowany z progiem wybranym metodą Otsu.

2.6 Erozja

Operacja ta zmienia piksele na czarne, jeśli w ich sąsiedztwie znajduje się, choć jeden czarny piksel. Realizowana jest za pomocą funkcji:

$$\text{bim} = \text{imerode}(\text{bim}, \text{ones}(3)).$$

Nieformalnie, dokonuje ona „odchudzenia” obrazu i pogłębienia tak zwanych wąwozów.



Rysunek 16.: Zbinaryzowany żubr po erozji.

2.7 Dylatacja

Zwana również akumulacją. Realizowana jest za pomocą funkcji:

bim = imdilate(bim, ones(3)).

Dylatacja to operacja analogiczna do erozji, czyli zmienia czarny piksel na biały wtedy, gdy sąsiaduje on z przynajmniej jednym pikselem białym. Metoda ta zwiększa granice obiektów dodając warstwę o grubości jednego piksela.



Rysunek 17.: Zbinaryzowany żubr po dylatacji.

2.8 Operacja zamknięcia

Polega na dokonaniu w pierwszej kolejności dylatacji a następnie erozji. Po tych operacjach małe czarne obszary na obiekcie zostają zmienione na białe lub „zamknięte”.



Rysunek 18.: Zbinaryzowany żubr po zamknięciu.

2.9 Operacja otwarcia

Polega na dokonaniu w pierwszej kolejności erozji a następnie dylatacji. Powoduje ona „otwarcie obszarów” blisko krawędzi obrazu, lub też połączenie się dziur wewnątrz obiektu.



Rysunek 19.: Zbinaryzowany żubr po otwarciu.

2.10 Wycinanie obiektu z obrazu.

Obraz oryginalny przekształciliśmy przy pomocy macierzy obrazu po operacji zamknięcia. W celu otrzymania lepszego efektu użyliśmy dylatacji i erozji z większymi wartościami, a następnie połączyliśmy wynik z oryginalnym kolorowym obrazkiem za pomocą funkcji:

$$\text{wynik} = \text{im}.*\text{bim}.$$



Rysunek 20.: Żubr po otwarciu z użyciem większych wartości.



Rysunek 21.: Wycięty żubr.

2.11 Wnioski

- Filtrami można efektywnie usuwać zakłócenia punktowe
- Binaryzacja pozwala na oddzielenie sygnału od szumu i na odnajdywanie konturów obiektów
- Filtry w głównej mierze służą do wyostrażania lub rozmazywania obrazu

3. Laboratorium 4

Kolejne laboratoria były poświęcone transformacie Fouriera, wykorzystywanej do przeprowadzania operacji na obrazach m.in. filtracji. Za obraz testowy posłużyło zdjęcie opery w Sydney wczytane i zmienione na odcienie szarości.



Rysunek 22.: Obraz opery w odcieniach szarości.

3.1 Szybka transformata Fouriera

Swoją analizę obrazu rozpoczęliśmy od szybkiej dwuwymiarowej transformaty Fouriera:

fim = fft2(n).

Funkcja ta utworzyła nam macierz dwuwymiarowej transformaty Fouriera. Liczymy wartości z każdej komórki macierzy oraz fazy:

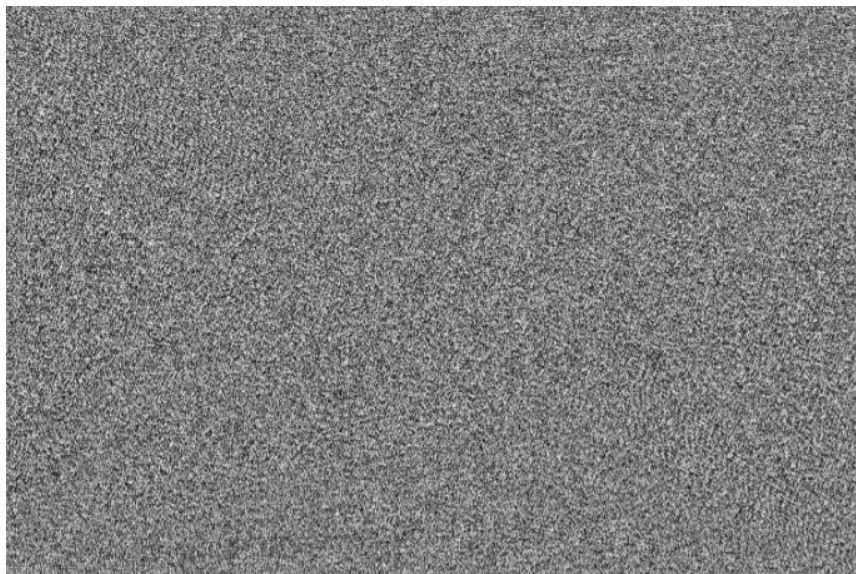
phi = angle(fim),

A = abs(fim),

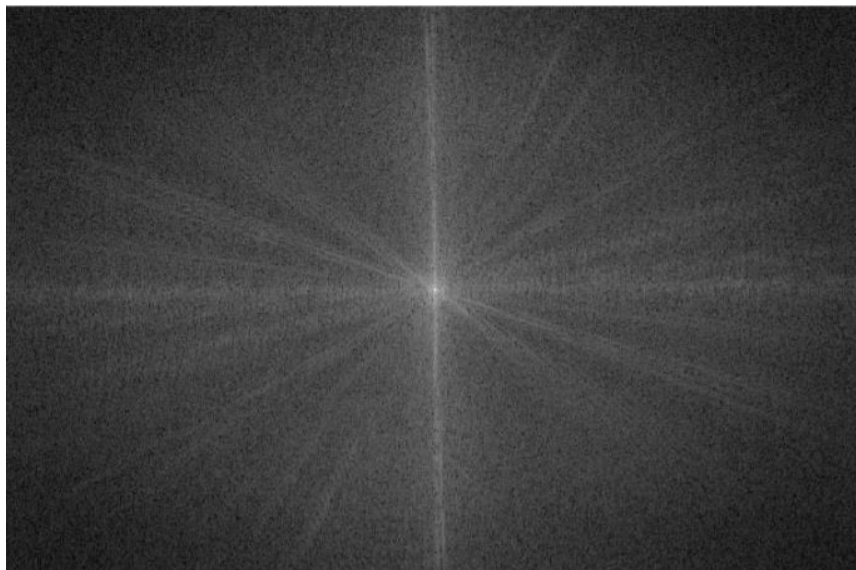
maxA = max(A,[],'all'),

imshow(phi,[-pi;pi]),

imshow(fftshift(log(A)),[0,log(maxA)]).



***Rysunek 22.:** Widmo fazowe opery.*



***Rysunek 23.:** Widmo amplitudowe opery.*

Nieregularność obrazu widma fazowego świadczy o tym, że obraz nie był modyfikowany. Z powyższego obrazu widma fazowego możemy otrzymać kąt oraz fazę, a następnie maksymalną amplitudę. Widmo amplitudowe pomaga zlokalizować i wyodrębnić obszary o pewnej częstotliwości na obrazie.

3.2 Odwrotna transformata Fouriera

W celu przywrócenia obrazu do oryginalnej postaci zastosowano odwrotną transformatę Fouriera. Polega ona w głównej mierze na przywróceniu fazy i amplitudy do postaci kanonicznej. Zrealizowano to przez następujące funkcje:

```
Z = A .* exp(1i * phi),  
im2 = ifft2(Z),  
im2 = abs(im2),  
imshow(im2).
```

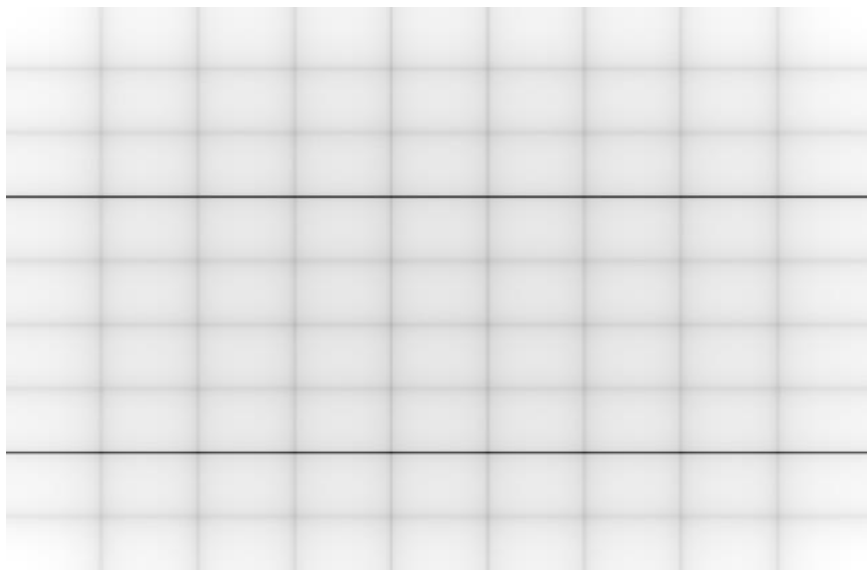


Rysunek 24.: Przywrócona pierwotna wersja.

3.3 Kompresja

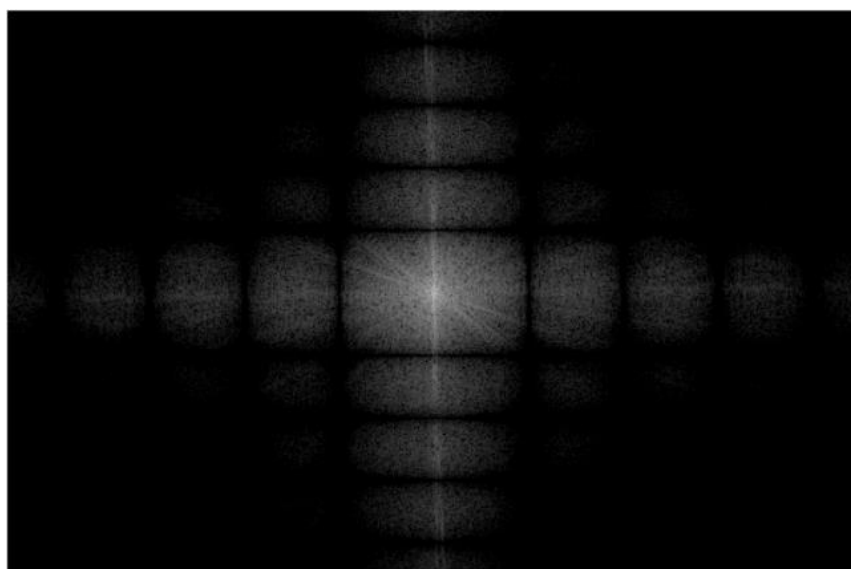
Z pierwotnego obrazu stworzyliśmy maskę po przez nałożenie filtru dolnoprzepustowego. Dokonałmy tego w wyniku transformaty Fouriera. Jest to operacja znacznie szybsza, zapewnia brak czarnej obwódki po odtworzeniu obrazu oraz możliwość

łączenia wielu filtrów ze sobą.



Rysunek 25.: Maska z wykorzystaniem filtru dolnoprzepustowego.

Tak otrzymaną maskę nałożyliśmy na pierwotny obraz i wyświetliliśmy jego widmo amplitudowe pomniejszone o pewne informacje względem pierwotnej wersji.



Rysunek 26.: Widmo amplitudowe obrazu po nałożeniu filtru/maski.

W dalszej kolejności dokonaliśmy kompresji transformatą cosinusową. Ustaliliśmy sobie, jaka część z otrzymanego widma amplitudowego chcemy zachować w pliku po kompresji. Obraz po kompresji stracił na ostrości i szczegółach, ale też zmniejszył się jego rozmiar – został skompresowany.

3.4 Zaburzenia obrazu

W wyniku ingerencji w widmo amplitudowe zaobserwowano nadanie się tzw. „tekstury” na obrazie. Pomimo takich zmian byliśmy w stanie odczytać zawartość obrazu po jego przywróceniu.



Rysunek 25.: Efekt zmiany widma amplitudowego.

3.5 Wnioski

- Transformata Fouriera może posłużyć do utworzenia maski a tym samym do kompresji obrazu
- Niewielkie zmiany w widmie fazowe i amplitudowe niosą za sobą ogromne zmiany przy odtwarzaniu obrazu